MicroZed
Industrial Internet of Things Starter Kit
Getting Started Guide
Version 1.1
Prior Version History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>30 May 2016</td>
<td>Initial MiniZed Hardware User Guide</td>
</tr>
<tr>
<td>1.1</td>
<td>04 Feb 2018</td>
<td>Updated to comply with new Avnet branding</td>
</tr>
</tbody>
</table>
Contents

1 Getting Started with the MicroZed IIoT Starter Kit ................................................. 4

2 What’s Inside the Box? .......................................................... 5
   2.1 MicroZed IIoT Starter Kit ..................................................... 5

3 What’s on the Web? ............................................................. 6
   3.1 Official Documentation: ....................................................... 6
   3.2 Tutorials and Reference Designs: ........................................... 6
   3.3 Trainings and Videos: ......................................................... 6
   3.4 MicroZed Documentation: .................................................... 6
   3.5 Cloud Solutions: .............................................................. 7
   3.6 Application Source Code: .................................................... 7
   3.7 Wind River Pulsar Linux: ..................................................... 7
   3.8 Available through Avnet FAE: ............................................. 7

4 MicroZed IIoT Key Features .................................................. 8

5 MicroZed IIoT Starter Kit Basic Setup and Operation ............................................. 9
   5.1 MicroZed SOM Boot Mode .................................................... 9
   5.2 Install Silicon Labs CP210x USB-to-UART Drivers on a PC .................. 11
   5.3 MicroZed IIoT Hardware Setup ............................................. 11
   5.4 Pulsar Login and Firmware Upgrade ...................................... 12

6 Running the Applications ...................................................... 13
   6.1 Run the SPI Application ...................................................... 13
   6.2 Run the I2C Application ...................................................... 14
   6.3 Run the IIoT Quickstart Demonstration .................................... 15
   6.4 Run the IIoT Bluemix Demonstration ..................................... 18
   6.5 Wind River Helix App Cloud ................................................. 20

7 Getting Help and Support ...................................................... 22
   7.1 Avnet Support .............................................................. 22
   7.2 Xilinx Support ............................................................ 23

8 Appendix I – Create Your Bluemix Service ............................................ 24
   8.1 Create a Starter Cloud Application ........................................ 24
   8.2 Create a Cloud Service ...................................................... 27
   8.3 Register a Device ............................................................ 29
   8.4 Complete the Bluemix Application ........................................ 34
   8.5 Clean up ............................................................................ 38
1 Getting Started with the MicroZed IIoT Starter Kit

The Avnet MicroZed IIoT Starter Kit enables hardware and software developers to create and evaluate Zynq™-7000 All Programmable SoC designs, using on-board sensors to gather information from the immediate surroundings and transmit the data to a Cloud environment. Cloud services can be used for real-time data analysis and decision-making, data storage, post-analysis, real-time or historical display and an ever-growing list of domain-specific functions.

The kit includes a standalone MicroZed System-on-Module (SoM) containing a hard-wired Zynq Processing System (PS) with peripherals as well as enabling the Zynq Programmable Logic (PL) fabric. This PS system includes DDR3 memory, Flash memory, gigabit Ethernet, USB 2.0 Host, and a UART.

The MicroZed SoM is attached to an Arduino™ Carrier Card. The carrier brings both Shield expansion and Peripheral Module (2x6 pin connectors) expansion to the MicroZed System-On-Module. The carrier card is ideal for building quick prototypes that leverage the large number of Arduino-compatible Shields and Pmod-compatible expansion boards.

The following sensors are included with the kit:

- ST Microelectronics motion MEMS and environmental sensor Shield
  - 3-axis accelerometer + 3-axis gyroscope
  - 3-axis magnetometer
  - Humidity and temperature
  - Pressure
- Maxim Integrated thermocouple-to-digital peripheral module
  - Connects to 2x6 peripheral module
  - SPI-based interface
  - Measures temperatures from -270°C to +1800°C
  - Includes K-type thermocouple

This Getting Started Guide outlines the steps to set up the hardware, load a sample hardware design, and execute Linux applications running on the ARM® dual-core Cortex™-A9 MPCore™ Processing System. Readings are taken from the on-board sensors and transmitted to an IBM Watson IoT Bluemix™ application, where the data is analyzed and resulting commands are sent back to the target. You will need to set up an IBM Watson IoT account to access the Bluemix services. A free trial and free online training is available through Avnet by filling out the registration at:

http://artofthepossibility.com/bluemix-free-trial
2 What’s Inside the Box?

2.1 MicroZed IIoT Starter Kit

- MicroZed 7010 SoM
- Arduino Carrier Card + (4) stand-offs and screws
- 2 USB-A to microUSB-B cables
- Ethernet cable
- 8GB microSD card + microSD to SD card adapter
- ST Microelectronics motion MEMS and environmental sensor Shield
- Maxim Integrated thermocouple-to-digital peripheral module
- Wind River Pulsar Linux
- Documentation
  - MZ IIoT Kit Quick Start Card
  - Wind River Pulsar Quick Start Card
3 What’s on the Web?

MicroZed IIoT is a community-oriented kit, with all materials being made available through the www.microzed.org community website. Documentation specific to the Arduino carrier can be found here:

http://microzed.org/product/arduino-cc

3.1 Official Documentation:
- Schematics (MicroZed 7010 and Arduino Carrier Card)
- Bill of materials
- Layout
- Hardware manual
- Board definition files

3.2 Tutorials and Reference Designs:
http://microzed.org/support/design/13591/
- MicroZed 7010 SPI and I2C Vivado Design Example
- MicroZed 7010 SPI Peripheral Application with Wind River Pulsar Linux
- MicroZed 7010 I2C Peripheral Application with Wind River Pulsar Linux

3.3 Trainings and Videos:
http://microzed.org/support/trainings-and-videos
- Overview of MicroZed
- Introduction to Zynq-7000 All Programmable SoC
- Developing Zynq-7000 All Programmable SoC Software
- Developing Zynq-7000 All Programmable SoC Hardware
- Wind River Pulsar Linux for Avnet Designed SoMs

3.4 MicroZed Documentation:
http://microzed.org/support/documentation/1519
- MicroZed Getting Started Guide
- Getting Started with Pulsar Linux on Avnet Zynq Boards
- Wind River Pulsar Linux Overview

http://microzed.org/support/documentation/9031
- MZCC_ARD Getting Started Guide
3.5 Cloud Solutions:
https://acm.avnetcloudsolutions.com/catalog/
- IBM Bluemix
- Amazon Web Services

3.6 Application Source Code:
https://github.com/Avnet/software

3.7 Wind River Pulsar Linux:
http://www.windriver.com/products/operating-systems/pulsar/

3.8 Available through Avnet FAE:
- Altium source database for schematic and layout
4 MicroZed IIoT Key Features

MicroZed 7010 SoM

- Zynq™-7000 AP SoC XC7Z010-CLG400-1 Processor
- Memory
  - 1 GB DDR3
  - 128 Mb Quad-SPI Flash
  - 4 GB microSD card
- Communication
  - 10/100/1000 Ethernet
  - USB Host 2.0 and USB-UART
- Expansion connectors
  - 2 Micro-Header connectors (108 single-ended, 48 differential pairs, Agile Mixed Signaling (AMS))
  - Digilent Pmod™ Compatible header (8 MIO)
- Clocking
  - 33.33333 MHz clock source for PS
- Configuration and Debug
  - Xilinx Platform Cable JTAG connector
- General Purpose I/O
  - 1 user LED
  - 1 push button

Arduino Carrier Card

- Full Arduino-compatible Shield site with connection to MicroZed PL I/O
- Two 2x6 Pmod™-compatible peripheral interfaces for MicroZed
- Supports interface for PL based SDIO/SPI
- Supports interface for I2C/UART
- One 2x6 Pmod™-compatible peripheral interface to low-power MCU connector

ST Microelectronics motion MEMS and environmental sensor Shield

- Six sensor solutions connect on Shield sockets via I2C AXI soft IP
  - 3-axis accelerometer and 3-axis gyroscope
  - 3-axis magnetometer
  - Humidity and temperature
  - Pressure
- General Purpose I/O
- 1 user LED

Maxim Integrated thermocouple-to-digital peripheral module

- Connects 2x6 peripheral module via SPI AXI soft IP
- Dual temperatures from on-board sensor and thermocouple sensor
  - Thermocouple temperature range -270°C to +1800°C
5 MicroZed IIoT Starter Kit Basic Setup and Operation

The MicroZed IIoT Starter Kit includes an 8 GB microSD card preloaded with a certified Wind River Pulsar Linux image, augmented with SPI and I2C drivers for the sensor peripherals and a bitstream to provide hardware support. The image also contains the following pre-built applications:

1. HTS221 Temperature and Humidity sense example
2. Maxim 31855 Dual Temperature sense example
3. Cloud-connectivity example using IBM Watson IoT and Bluemix services

Wind River Pulsar Linux is an application-ready development and production platform, allowing for on-target compilation and remote upgrade through the Smart Package Manager. The platform also offers a connection to a Cloud-based SDK and visual debug environment on the Wind River Helix App Cloud. This free service allows development on a remote target from any Web browser. You may register here:

https://app.cloud.windriver.com/

For more details on Wind River Pulsar Linux, consult the Wind River Pulsar Quick Start Card included in your kit, or visit:

http://www.windriver.com/products/operating-systems/pulsar/

5.1 MicroZed SOM Boot Mode

Before assembling the kit items, please take a moment to locate the key components on the MicroZed SoM and ensure it is in the microSD boot configuration. MicroZed SoMs are available standalone, and have the Boot MODE jumpers configured at the factory to boot from QSPI flash. In QSPI flash mode, a memory-resident Linux kernel and file system will boot but will not provide the software/hardware elements necessary to complete the procedures in this guide.
MicroZed SoM Component Locations and Factory Boot Setting

Please check the Boot MODE jumpers and ensure they are in the position shown below. This will allow the boot to source the image from the supplied microSD card (JP1 in the top position, JP2 and JP3 in the bottom position).

MicroZed SoM microSD Boot Mode Jumper Settings
5.2 Install Silicon Labs CP210x USB-to-UART Drivers on a PC

The USB-to-UART port on the MicroZed SoM is required to display serial output generated on the target to a console on a host PC. To use a standard USB port on a host PC as a serial device, you must first install drivers from Silicon Labs. The drivers will allow the USB connection to be enumerated as a serial port, which can then be used to communicate with the target via a serial communications program such as Tera Term Pro.

If not previously installed on your host system, visit:

http://microzed.org/support/documentation/1519

to download and install the drivers referenced in Avnet’s CP210x USB-to-UART Setup Guide.

5.3 MicroZed IIoT Hardware Setup

1. Attach the ST Micro X-NUCLEO-IKS01A1 Shield to the Arduino Carrier Card.

2. Plug the thermocouple into the Maxim 31855 Pmod and attach the assembly to the connector (J3) top row at the end of the Arduino Carrier Card.

3. Insert the supplied Pulsar microSD card in the cage beneath the MicroZed SoM.

4. Attach the MicroZed SoM to the Arduino Carrier Card FMC connectors. Set the boot mode jumpers for microSD-boot (see MicroZed SoM Boot Mode section).

5. Plug a DHCP-Server-connected Ethernet cable to the MicroZed SoM RJ-45. In order to connect to the Watson IoT Bluemix service via Ethernet, the kit must be able to obtain a LAN IP address from a DHCP server, and a gateway address that allows unfettered Internet access.

6. Connect one microUSB cable between the AC/DC power supply and Arduino power port (J7). The blue DONE LED adjacent to SW2 on the MicroZed SoM will illuminate as the board boots Wind River Pulsar Linux.

Connect the USB-UART port on the MicroZed SoM (J2) to a free USB port on your PC. Use the PC Device Manager to determine the COM port and set the console parameters to 115200,n,8,1. Optionally, reset the processor by pressing and releasing the RST button on the MicroZed SoM (SW2) to view the entire boot.
5.4 Pulsar Login and Firmware Upgrade

Once power has been applied to the kit, the boot sequence will initialize the Programmable Logic with the supplied hardware system. Completion of this step is indicated by the illumination of the blue DONE LED on the MicroZed SoM.

Once hardware initialization is complete, the Pulsar Linux software will boot from the microSD card. On completion of the boot, the serial console on the host will display a login prompt, as shown here:

```
Pulsar Linux 7.0.0.11 cube-28-03-16-cube-server ttyPS0
Pulsar Linux 7.0.0.11 cube-28-03-16-cube-server login:  
```

1. Log in at the command prompt with user/pw:

   **root/incendia**

2. To upgrade the device firmware, at the command prompt enter the following:

   **smart update**

   This procedure is typically done only on the first-time boot, but it can be repeated as desired to manually check for updated repository contents.

The Smart Package Manager will automatically contact the Wind River public repository to compare the contents on the microSD card with the current packages available. If necessary, newer packages will be downloaded and installed automatically.
6 Running the Applications

6.1 Run the SPI Application

The SPI application reads temperature data from the Maxim 31855 Pmod sensors. There are two sensors on the module, one providing ambient temperature at the module location and the second from the end of the cable attached to the thermocouple.

![Thermocouple and Cable](image)

**Pmod On-board Temperature Sensor Location**

The sample application reads data from both sensors and displays the result on the PC host serial console. On the target, the SPI interface is accessed from the application code using the built-in SPIDEV driver in the Pulsar Linux kernel.

At the command prompt, enter the following command:

```
./maxim31855_sensor
```

The temperatures are reported once on the console as shown below:

```
root@cube-28-03-16-cube-server:~# ./maxim31855_sensor
spi mode: 1
bits per word: 8
max speed: 3125000 Hz (3125 KHz)
MAX31855 Internal Temp = 21.6 deg C or 70.81 deg F
MAX31855 Thermocouple Temp = 20.5 deg C or 68.90 deg F
```

**Note:** The thermocouple must be inserted into the Pmod, or the application will detect an open circuit and exit without displaying temperature values.
6.2 Run the I2C Application

The I2C application reads temperature and humidity data from the ST Microelectronics HTS221 sensor, located on the Arduino Shield.

HTS221
Temperature/Humidity Sensor (U3)

The sample application reads data values from the sensor and outputs the values continuously to the PC host serial console.

On the target, an I2C driver in the Pulsar Linux uses the sysfs virtual file system to export information about the I2C kernel subsystem to user space.

At the command prompt, enter the following:

```
./hts221_sensor
```

The temperature and humidity values are reported on the console as shown below:

```
root@cube-28-03-16-cube-server:~# ./hts221_sensor
Humidity = 39.898%  Temperature = 21.548 °C or 70.786 °F
Humidity = 41.646%  Temperature = 21.663 °C or 70.993 °F
Humidity = 41.784%  Temperature = 21.576 °C or 70.837 °F
Humidity = 40.488%  Temperature = 21.663 °C or 70.993 °F
CSigINT received.
```

To end data display, enter <CTRL>-C in the console.
6.3 Run the IIoT Quickstart Demonstration

This demonstration reads data from selected onboard sensors and publishes (transmits) to the Cloud-resident IBM Watson IoT public Quickstart service. Incoming values are displayed graphically as they arrive. Your IIoT kit must be connected to the Internet for this demonstration to operate correctly, but a Bluemix account is not required.

By default, data is read from the HTS221 temperature sensor on the ST Microelectronics Arduino Shield and published to the cloud service. The application can be executed from the command prompt using the format:

```
./IIoT_QuickStart_Demo <uniqueID>
```

where `<uniqueID>` is any alphanumeric string you choose. This string is simply used to isolate your specific instance from other simultaneous access requests to the site by any number of concurrent users. It is valid only for the duration of the application run, but can be reused or changed in each subsequent execution. In the unlikely event that your `<uniqueID>` is refused, simply choose another string that is less likely to be picked by another user.

The example below uses a `<uniqueID>` of **abc123**:

```
root@cube-28-03-16-cube-server:~# ./IIoT_QuickStart_Demo abc123
Connection Successful. Press Ctrl+C to quit
View the visualization at https://quickstart.internetofthings.ibmcloud.com/#/device/abc123
Humidity = 39.441% Temperature = 22.712 C or 72.882 F
----Sensor reading from HTS221 Temperature
JSON = {"d" : {"temp" : 22.71 }}
Publishing the event stat with rc 0. Type <CTRL>-C to terminate...
Humidity = 40.206% Temperature = 22.741 C or 72.934 F
-----Sensor reading from HTS221 Temperature
JSON = {"d" : {"temp" : 22.74 }}
Publishing the event stat with rc 0. Type <CTRL>-C to terminate...
```

Notice that the live data transmission can be seen at the link shown above, in the format:

https://quickstart.internetofthings.ibmcloud.com/#/device/
with <uniqueID> appended to the end. You may click the link above to go to the website and enter your <uniqueID> in the text box provided, or copy the link directly from the serial console and paste it into the browser of any Internet-connected device.

![Quickstart](image)

HTS221 Temperature Data in IBM Watson IoT QuickStart

Type <CTRL>-C in the IIoT Kit console to terminate the application instance.

You may read data from other sensors by supplying an optional selection parameter following the <uniqueID>. For a list of the sensors currently available from the application, at the command prompt type:

`.IIoT_Quickstart_Demo abc123 -help`
root@cube-29-03-16-cube-server:~$ ./IIoT_QuickStart_Demo abc123 -help

Default no argument - read from HTS221 Temperature
  0 - read from HTS221 Temperature
  1 - read from HTS221 Humidity
  2 - read from MAX31855 Internal Temperature
  3 - read from MAX31855 Thermocouple Temperature
  4 - read from No Sensor - generated data
6.4 Run the IIoT Bluemix Demonstration

This demonstration builds on the Quickstart application, adding these features:

- Published data requires security credentials to connect to a Bluemix application
- The Bluemix application processes published data and communicates analytic results back to the IIoT Kit through a callback registration mechanism
- Incoming results from Bluemix are received by the application and the User LED on the MicroZed is accessed to convey status information

Your IIoT kit must be connected to the Internet for this demonstration to operate correctly, and you must have registered with IBM Watson IoT to set up your own Bluemix account. If you have not done so, to access your Bluemix trial and obtain a promotional code to access Avnet online training at no charge, please visit:

http://artofthepossibility.com/bluemix-free-trial

Once you have your Bluemix account, you will need to configure the Watson IoT broker and bind it to a running Bluemix application. For first time operation, see Appendix I for step-by-step instructions.

1. Ensure that your Cloud (Bluemix) application is running, as described in Appendix I.

2. To run the IIoT Bluemix demonstration, at the command prompt for the IIoT Kit enter:

   ./Iiot_Bluemix_Demo

The application will read temperature and humidity values from the HTS221 sensor on the Shield and transmit data to your Bluemix service.

```
./Iiot_Bluemix_Demo
Connecting to registered service with org 5frn3
Humidity = 44.98% Temperature = 20.958 C or 69.724 F
---Sensor reading from HTS221 Temperature
JSON = ({"d": {"temp": 2095})
Publishing the event stat with rc 0
---The command received :: Status
format: JSON
Payload is: Temperature (20.95) within safe limits
-----------------
Humidity = 44.28% Temperature = 20.872 C or 69.570 F
---Sensor reading from HTS221 Temperature
JSON = ({"d": {"temp": 2087})
Publishing the event stat with rc 0
---The command received :: Status
format: JSON
Payload is: Temperature (20.87) within safe limits
```
Additional sensor options for the sample demonstration program can be displayed by requesting help at the command line.

```
./IoT_Bluemix_Demo -help
```

Default no argument - read from HTS221 Temperature
0 - read from HTS221 Temperature
1 - read from HTS221 Humidity
2 - read from MAX31855 Internal Temperature
3 - read from MAX31855 Thermocouple Temperature
4 - read from No Sensor - generated data
6.5 Wind River Helix App Cloud

Pulsar Linux provides an application-ready development environment on the MicroZed SoM, allowing you to create and build programs in the command line environment directly on the target. For details on this and other features of your software platform, consult the Wind River Pulsar Quickstart Card included in the kit.

However, with Avnet Zynq SoMs you may also access a cloud-based Software Development Kit. Using Wind River Helix App Cloud (HAC) allows development from any Internet-enabled device with a web browser. Application source can be uploaded and downloaded easily from a host system to the Cloud, or between the IIoT Kit and the Cloud. The HAC provides a full-screen, color-coded editor and a compatible tool-chain for your target, as well as complete visual debugger. With your IIoT Kit connected to the Internet, you can debug applications seamlessly on your target, regardless of the proximity of the device to your physical location.

To access the HAC, follow these steps:

1. Ensure the IIoT Kit is connected to the Internet and at the Pulsar command prompt enter:

   ```
   registerTarget -n <RandomCode>
   ```

   ```
   root@cube-28-03-16-cube-server:~ # registerTarget -n myUniqueID123
   Created target [ myUniqueID123 ] on server [ app.cloud.windriver.com ]
   Device registration status: pending
   Device registration expires in 19.9666666666667 (mins)
   Device Registration Key: 61d96_ 
   ```

2. You will receive a Registration Key valid for approximately 20 minutes. Direct your host-based browser to:

   ```
   https://app.cloud.windriver.com/
   ```

3. Sign in with your Wind River Support credentials¹ and click the New Device button.

   ![My Devices New Device](image)

¹ If you do not already have a Wind River Account, you may create one by clicking the Register Now button on this site.
4. Click on “Register an existing device…” and follow the on-screen prompts to create the cloud-based SDK workspace for your IIoT Kit.

![Register an existing device using its Registration ID](image)

5. Once the IIoT kit is registered, you can create a new application project by clicking the Create new project button.

![Create new project](image)

For further details on getting started developing with the Wind River Helix Application Cloud on the MicroZed IIoT Kit, view the online video Developing IIoT Applications on Wind River Helix App Cloud at:

7  Getting Help and Support

7.1  Avnet Support

Technical support is offered online through the MicroZed.org website support forums. MicroZed IIoT Starter Kit users are encouraged to participate in the forums and offer help to others when possible.

For questions regarding the MicroZed community website, please direct questions to:

MicroZed.org Web Master – webmaster@microzed.org

To access the most current collateral for the MicroZed IIoT Starter Kit, visit the community support page (www.microzed.org/content/support) and click one of the icons shown below:

Documentation

Reference Designs

Scroll to the Industrial IoT Starter Kit and click the View button.

Industrial IoT Starter Kit

The Avnet MicroZed Industrial IoT Starter Kit supports designers’ edge-to-

Technical questions for the MicroZed IIoT Starter Kit should be directed to the MicroZed Support forums. Click the Support Forums link:

Support Forums

Enter questions under either MicroZed Hardware Design or Software Application Development.
7.2 Xilinx Support

For questions regarding products within the Product Entitlement Account, send an email message to the Customer Service Representative in your region:

Canada, USA and South America - isscs_cases@xilinx.com
Europe, Middle East, and Africa - eucases@xilinx.com
Asia Pacific including Japan - apaccase@xilinx.com

For technical support, including the installation and use of the product license file, contact Xilinx Online Technical Support at [www.xilinx.com/support](http://www.xilinx.com/support). The following assistance resources are also available on the website:

- Software, IP and documentation updates
- Access to technical support Web tools
- Searchable answer database with over 4,000 solutions
- User forums
8 Appendix I – Create Your Bluemix Service

8.1 Create a Starter Cloud Application

These instructions assume you have already registered for an IBM Watson IoT account.

1. From your host computer, log into the Bluemix site, using the credentials you established in the lab pre-work. The website is: https://console.ng.bluemix.net.

2. From the Bluemix main menu, select CATALOG.

3. From the Starters list on the left, select Boilerplates.

4. Click the Internet of Things Platform Starter.

5. In the panel on the right, enter a unique application name. The name must consist entirely of alphanumeric characters and/or the dash character.
6. Click the **CREATE** button. You will see the staging message at the top of the page as follows:

![Staging message](image)

Your application is staging. [http://lloTKitCloudApp-01.mybluemix.net](http://lloTKitCloudApp-01.mybluemix.net)

After a few minutes, the message changes to indicate the application has started running, as shown below. Depending on the volume of traffic to the site, starting the application can take anywhere from 2 to 10 minutes.

![Running message](image)

Your app is running. [http://lloTKitCloudApp-01.mybluemix.net](http://lloTKitCloudApp-01.mybluemix.net)
7. Enter the following URL in your browser (or click the link in the app-running message):

   http://<ApplicationName>.mybluemix.net

8. Click **Go to your Node-RED flow editor**.

This opens a new tab in your browser showing the Node-RED cloud application development SDK.

The default application shown below is made up of different types of nodes, each with different operational characteristics, and linked together to perform a logical circuit.

In this case, information arrives at the left in the blue input node, and is interpreted as temperature data. In the lower chain, the temperature is evaluated based on a threshold and a status message printed.

This is the Bluemix equivalent to a helloWorld program, and could be used with the Quickstart service to create an unsecured application connection from the IIoT Kit. However, the goal is to create a secure connection, so for the moment we will leave the application and create an IoT Service. Leave this browser tab open, as you will return to it later.
8.2 Create a Cloud Service

Quickstart is a public service that allows anyone to make an unsecured connection to the sample application. However, now that we have a custom application, we want to provide a more secure access point, through the IBM Watson IoT broker service. Once the service is created, we can bind our application to it, so incoming data will be validated and then routed to your application.

1. Return to the Dashboard tab in your browser.

2. From the Bluemix main menu, select CATALOG.

3. From the Services map on the left, select Internet of Things.

4. Click Internet of Things Platform.

5. In the rightmost panel, select the application to bind to the service from the App drop-down (the application you created in earlier will be listed) and click the CREATE button.
6. Accept the offer to restage the application by clicking the **RESTAGE** button when presented. Once the application is running, you will have a new Bluemix option for authentication, which can be used with the IIoT_Bluemx_Demo on the target.

   As before, this operation will take a few minutes (2-10) to complete.
8.3 Register a Device

Now we have an IoT Cloud application, bound to an IoT Service, and we are ready to establish credentials to register an edge device.

1. You should still be in the Dashboard tab in the browser. Select it if you are not.

2. Click the Back to Dashboard link in the left panel.

3. Your application displays a new icon representing the IoT service bound previously. Click the IoT icon in your application to display the Internet of Things Platform page.

4. In the Connect your Devices column, click the Launch Dashboard button.
5. In the left panel, click Devices, then click the Add Device button.

6. You can add an existing device type (one created previously) or add a new device type. If this is your first time registering a device, click the Create device type button.

   If you are creating a device type, you will need to enter various parameters to associate with the device type. For example, the manufacturer (Avnet) or the serial number which is located on a white sticker on the bottom of the board.

7. Click the Create device type button (again) on the updated browser page.

8. Fill in the Name and Description of the new device type. You can follow the example below or enter information of your choosing. Click the Next button at the bottom right.

9. In the Create Device Type panel, click in the check box next to Manufacturer to select it. Click the Next button.
10. In the *Create Device Type* panel, enter the *Manufacturer* as *Avnet*. Click the *Next* button.

11. Click the *Create* button to skip the optional Metadata entry.

12. In the *Add Device – Choose Device Type* panel, select the device type you just created and click the *Next* button.

13. In the *Add Device – Device Info* panel, enter the unique Device ID specified earlier in the input node of your Bluemix application. Note that the *Manufacturer* field is automatically filled in. If you expand the *Additional fields*, you can optionally enter further device information. Click the *Next* button.

14. Click the *Next* button to skip the optional *Metadata* entry.

15. In the *Add Device - Security* panel, enter a simple token in the *Provide a token (optional)* field. This will be easier to remember and enter later in the on-target configuration file. However, In a production situation, a more complex value can be used or generated. Make a note of the token you entered, and click *Next*.  

Page 31
16. In the **Add Device – Summary** panel, review the information and click the **ADD** button. The system will display your device credentials. For example:

![Add Device – Summary panel](image)

Keep the authentication information handy, as you will need it to configure the edge application so it can connect to the running Bluemix application.

**TIP:** Use the Windows snipping tool to save it temporarily to the desktop.

17. To use the new service, you need to create/update the configuration file with the new authentication information. This is done on the target (MZ IIoT Kit) device.

   a. Return to the terminal console (or reboot your target and log in) and ensure you are still in the `/root` directory.

   ```
   root@cube-28-03-16-cube-server:~# pwd
   /root
   ```

   b. Launch the `vi` editor on the target to create a new file called `device.cfg`.

   ```
   vi device.cfg
   ```

   ```
   root@cube-28-03-16-cube-server:~# vi device.cfg
   ```

   You will be creating a file exactly as show, replacing the highlighted text to substitute the credentials you generated for your device in the previous step.
c. Press the i keyboard key to enter **Insert** mode.

d. Use the **arrow** keys as needed to move around the file. Use the **backspace** key to erase the text you wish to replace. Type in the information you saved for your device credentials, replacing each line as indicated with the true information. For example, using the sample credentials shown in Step 16, your file contents would appear as shown here:

```
org=8a8rej
type=MicroZed
id=MZ-1402120-7010
auth-method=token
auth-token=SecurityToken99
```

e. When you have completed entering all the information, return to command mode by hitting the <ESC> key followed by a colon (ii:). The cursor will go to the bottom of the screen.

f. Enter **wq** (for write-quit). At this point you should be back at the command line.

18. List your file with the **cat** instruction. It should appear exactly as you entered it. If not, repeat the editing process until it is correct.

```
cat device.cfg
```
Using the example above, the listing appears as shown here:

```
root@cube-28-03-16-cube-server:~# cat device.cfg
org=8a8rej
type=MicroZed
id=MZ-1402120-7010
auth-method=token
auth-token=SecurityToken99
```

The *IIoT_Bluemix_Demo* program automatically looks in its current directory for a configuration file, so when it is launched, it will adopt your configuration credentials. These could be hard-coded in an application, but it is more convenient to use a configuration file.

### 8.4 Complete the Bluemix Application

1. Return to the Node-RED editor tab in your host browser, and double click the *IBM IoT App In* node to edit the node.
   a. Select the *Bluemix Service* as the *Authentication* method.
   b. Uncheck the *All* box to the right of *Device Type* and enter the value you created in the prior section (ie: MicroZed).
   c. Uncheck the *All* box to the right of *Device Id* and enter the ID you selected for the IIoT Kit.
   d. Click the *OK* button.
2. In the left-hand panel, scroll down to the output nodes and drag and drop the ibmiot node onto Flow 1. Position the node to the right and above the safe node.

3. Click the bubble on the right side of the safe node and with the left mouse button pressed, drag the cursor over to the left bubble on your new IBM IoT node to make a connection between them. Repeat the connection using the right bubble on the danger node.

4. Double-click the temp thresh node. Update the values in both boxes to 2500 (integer representation of 25 degrees Celsius to two decimal places). Click the OK button.

5. Double-click the danger node. Add 10 spaces after critical as shown below. This will correct an overwrite issue on the output console when switching between the safe and danger nodes. Click the OK button.

6. Double-click the new IBM IoT node to open its properties. Fill the parameters as shown, but substitute your own Device Type and Device ID for the example values:
Make sure you capitalize JSON in the Format box. Click the OK button.

7. Click the Deploy button at the top right.

8. Return to the IIoT Kit console and run the IIoT_Bluemix_Demo application. You can monitor the connection in the debug panel of the Node-RED flow editor, and you will see new messages on the target console that indicate the Cloud is communicating back to the target.

In the target console, enter: 

```
./IIoT_Bluemix_Demo
```

root@cubex-28-03-16-cube-server:~# ./IIoT_Bluemix_Demo
In the *debug* tab in your Node-RED application, you will see data start to flow:

![debug_tab](image)

Note: Clear the debug window by pressing the garbage can icon at the upper right.

**TIP:** If you do not see an Info/Debug tab, click the right hamburger menu icon (three horizontal lines) and select View | *Show Sidebar* (check the box).

On the IIoT Kit console, you will see corresponding transmissions (JSON string above each *Publishing* line) and the incoming response from the Bluemix application.

```json
Humidity = 46.427%  Temperature = 22.913 C or 73.243 F
--------Sensor reading from HTS221 Temperature
JSON = {"d": {"temp": 2291 }}
Publishing the event stat with rc 0

The command received :: Status
format : JSON
Payload is : Temperature (2291) within safe limits

Humidity = 46.666%  Temperature = 22.956 C or 73.321 F
--------Sensor reading from HTS221 Temperature
JSON = {"d": {"temp": 2295 }}
Publishing the event stat with rc 0

The command received :: Status
format : JSON
Payload is : Temperature (2295) within safe limits
```

If you do not see the return messages on your console, the local credentials in the *device.cfg* file on your target do not match those entered above in the Bluemix *IBMiot* output node properties (step 6). Also make sure you did not forget to capitalize *JSON* in the output node properties.
8.5 Clean up

Once you are finished working with the Bluemix application and Watson IoT services, you should close the Bluemix application and stop the service. This will prevent charges from accumulating in the event you are not using a Free service.

1. Close the Node-Red tab in your browser by clicking the X icon.

2. Close the Watson IoT Platform tab by clicking the X icon.


4. Click the gear icon at the upper right of your running application.
5. Select **Stop App** from the pop-up menu. Click the **STOP** button in the subsequent dialog to confirm.

![Stop Application Dialog]

6. Once the application has stopped, you may log out from your Bluemix account by clicking on the icon in the upper right of the page.

![Logout Icon]

7. Click on the **Log Out** link that appears. Click the **Log Out** button in the new dialog to confirm.

![Confirm Logout Dialog]

8. You may now close the browser.

9. If you still have the target application running, enter `<CTRL>-C` in the console.

10. To ensure a safe power down of the IIoT Kit, enter `poweroff` at the console prompt.

    ```
    root@cube-28-03-16-cube-server:~# poweroff
    ```

11. You may now disconnect the serial microUSB cable from the MicroZed SoM and the power microUSB cable from the Arduino carrier.

12. Close the IIoT Kit serial console.