LANTRONIX®

Application Note:

Integrating xPico 200 Series with Microsoft Azure

Part Number APP-0042
Revision C June 2023

Intellectual Property

Contacts

Lantronix, Inc.
48 Discovery, Suite 250
Irvine, CA 92618, USA
Toll Free: 800-526-8766
Phone: 949-453-3990
Fax: 949-453-3995

Technical Support
Online: www.lantronix.com/support

Sales Offices
For a current list of our domestic and international sales offices, go to the Lantronix web site at www.lantronix.com/about/contact

Disclaimer

All information contained herein is provided “AS IS.” Lantronix undertakes no obligation to update the information in this publication. Lantronix does not make, and specifically disclaims, all warranties of any kind (express, implied or otherwise) regarding title, non-infringement, fitness, quality, accuracy, completeness, usefulness, suitability or performance of the information provided herein. Lantronix shall have no liability whatsoever to any user for any damages, losses and causes of action (whether in contract or in tort or otherwise) in connection with the user’s access or usage of any of the information or content contained herein. The information and specifications contained in this document are subject to change without notice.

Revision History

Date	Rev.	Comments
June 2020	A	Initial document.
June 2021	B	Update openssl command to generate a private key.
June 2023	C	Fixed misspelling

For the latest revision of this product document, please check our online documentation at www.lantronix.com/support/documentation.

Contents

1 Overview 5
Prerequisites 5

2 Creating an IoT Hub in Microsoft Azure 6

3 Adding an xPico 200 Series Device to Azure IoT Hub 8
Adding a device using a SAS Token 8
Adding a device using a self-signed X.509 certificate 16
Adding a device using a CA-signed X.509 certificate 29

4 Calling Methods on an xPico 200 Series Device 41
Examples 41
ltrx_import_xml_config 41
ltrx_trusted_import_xml_config 43
ltrx_read_xml_status 44
ltrx_read_xml_config 45
ltrx_trusted_import_xml_config 46

1 Overview

This application note describes how to integrate an xPico 200 series device with Microsoft Azure. The Microsoft Azure service and the Azure portal may change. This document serves as an example but may not be up to date. Refer to the Microsoft Azure documentation for updated Azure instructions.

You can add a device to an IoT Hub in Azure using either a SAS Token, a self-signed X.509 certificate, or a CA-signed X.509 certificate. This document describes how to create an IoT Hub in Azure, the methods for adding a device, and examples for calling methods on an xPico 200 series device using the Device Explorer tool from the Azure IoT SDK.

Prerequisites

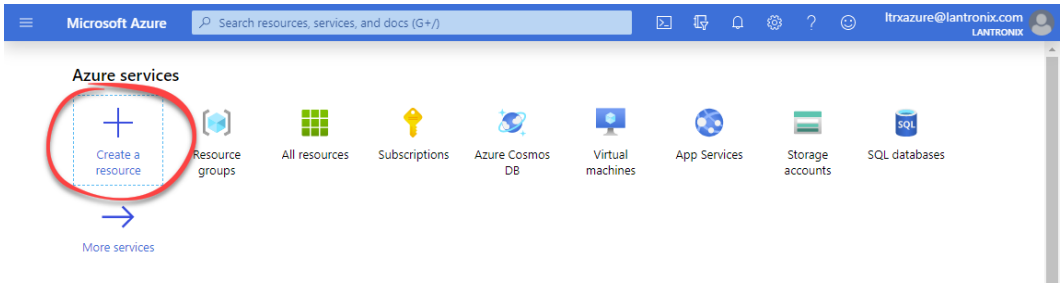
You will need the following software:

- Visual Studio Code: https://code.visualstudio.com/download
- Azure IoT Tools for Visual Studio Code: https://marketplace.visualstudio.com/items?itemName=vsciot-vscode.azure-iot-tools
- Device Explorer: https://github.com/Azure/azure-iot-sdks/releases
- D-TRUST.pem, DigiCert.pem, and baltimore-ca.pem: https://github.com/Azure/azure-iot-sdk-c/tree/master/certs (for the self-signed X.509 method)
- Certificate Authority files: https://github.com/Azure/azure-iot-sdk-c/tree/master/tools/CACertificates (for the CA-signed X.509 method)

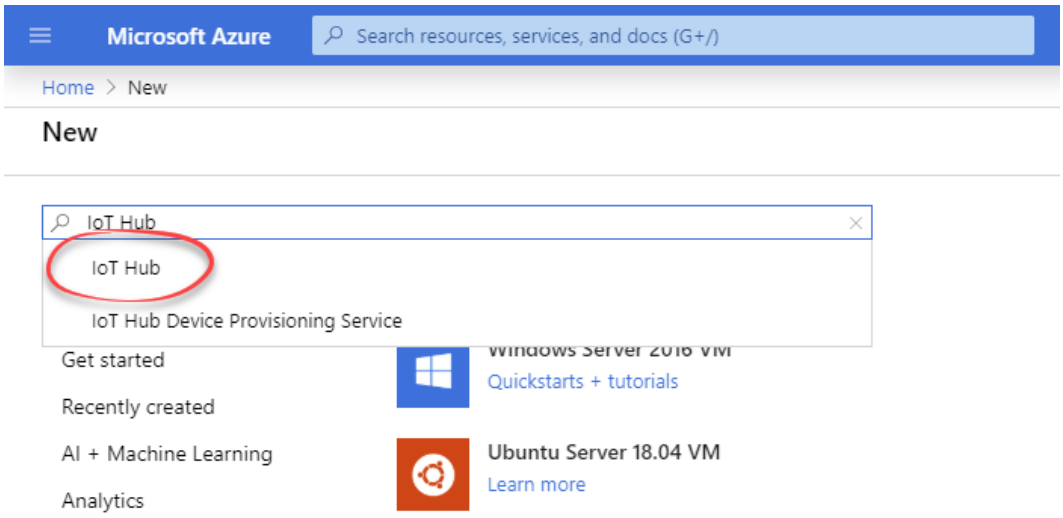
2 Creating an IoT Hub in Microsoft Azure

To create an IoT Hub in Microsoft Azure:

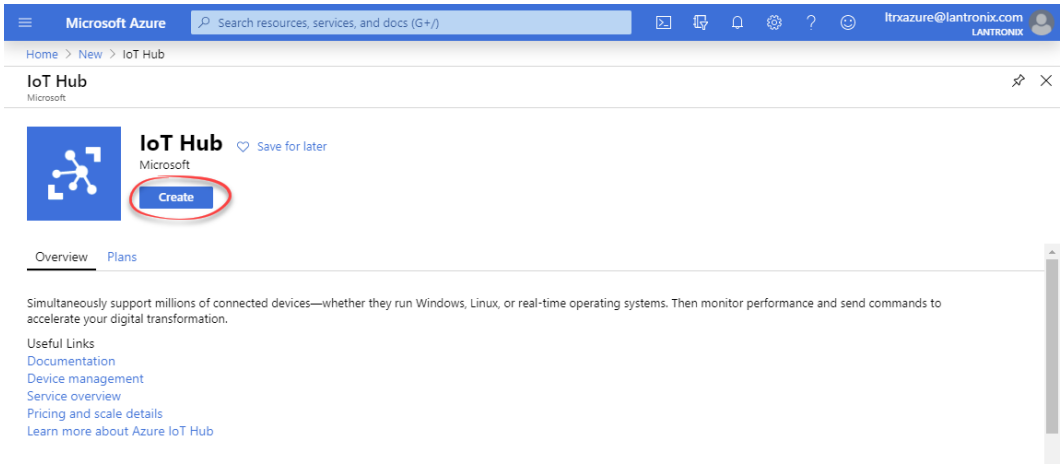
1. Sign in to the Azure portal at https://portal.azure.com/.
2. Click **Create a resource**.



3. In the **Search the Marketplace** field, type “IoT Hub” and select it from the results.



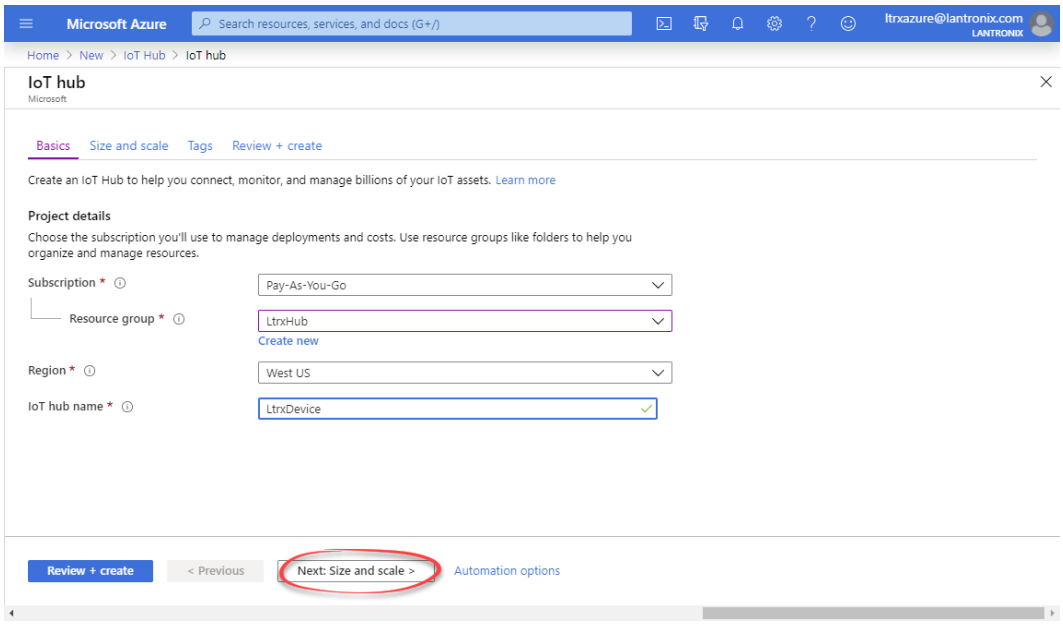
4. Click **Create**.



5. On the **Basics** tab, select a **Subscription**, select a **Resource Group** or create a

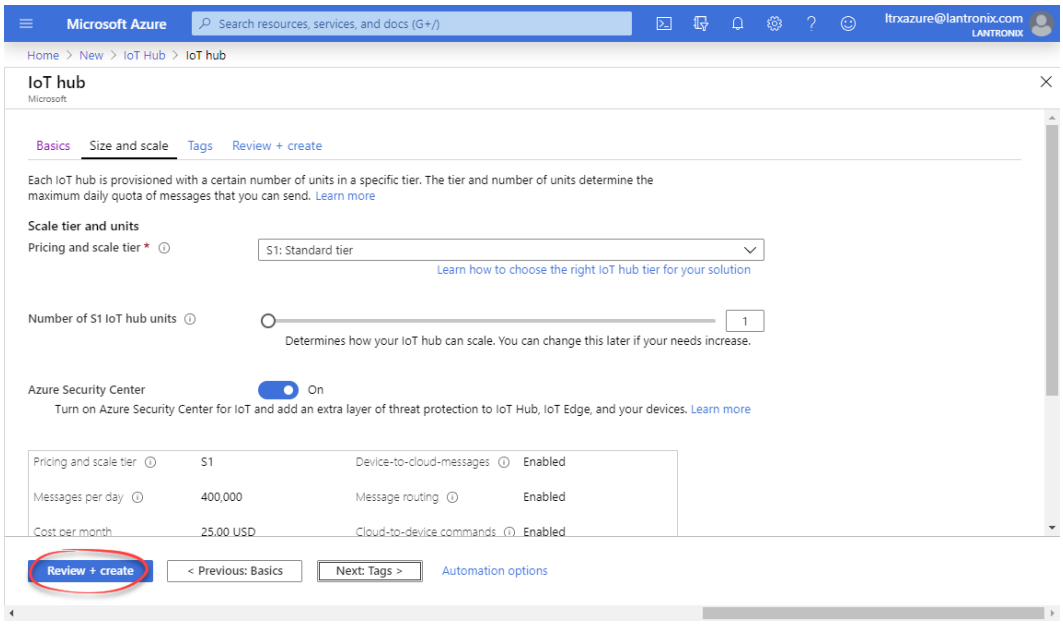
new one by clicking **Create new** and typing a name, select a **Region**, and enter a globally unique **IoT Hub Name**. This information will be publicly available, so do not user private information.

6. Click **Next: Size and scale**.



7. On the **Size and scale** tab, select a **Pricing and scale tier**, specify the **IoT Hub units**, and under **Advanced Settings**, choose the number of **Device-to-cloud partitions** (most likely four).

8. Click **Review + create**.



3 Adding an xPico 200 Series Device to Azure IoT Hub

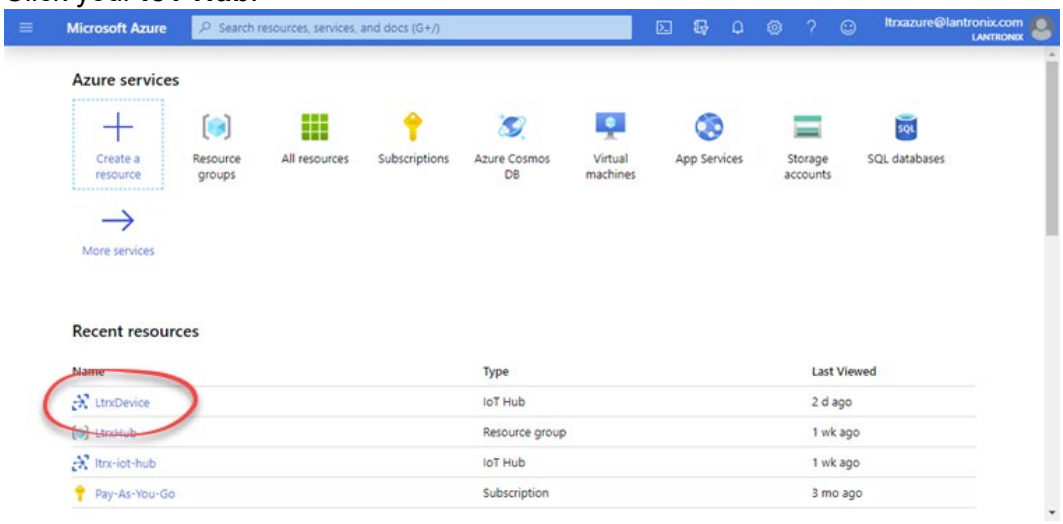
Use one of the methods described in this section to add the xPico 200 device to your IoT Hub in Microsoft Azure.

Adding a device using a SAS Token

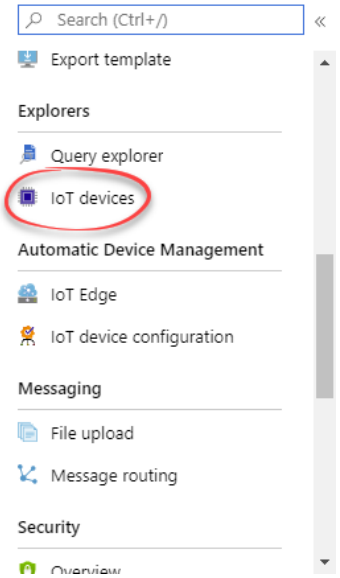
To add a device:

STEP 1. Create a device in your IoT Hub

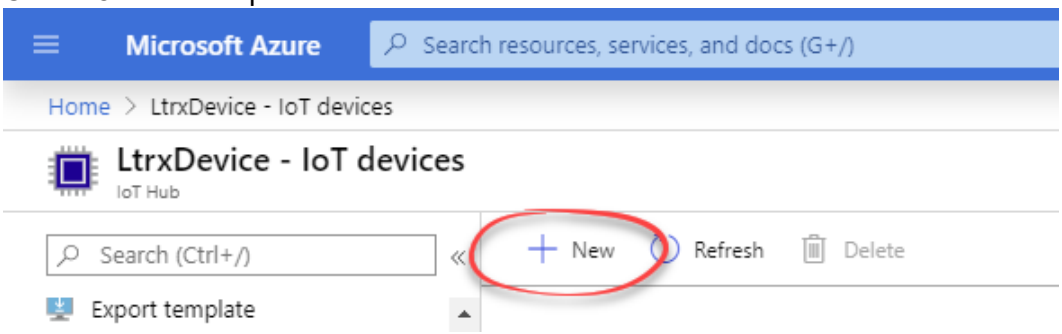
1. Log in to https://portal.azure.com.
2. Click **Home** if you are not already at the home page.
3. Click your **IoT Hub**.



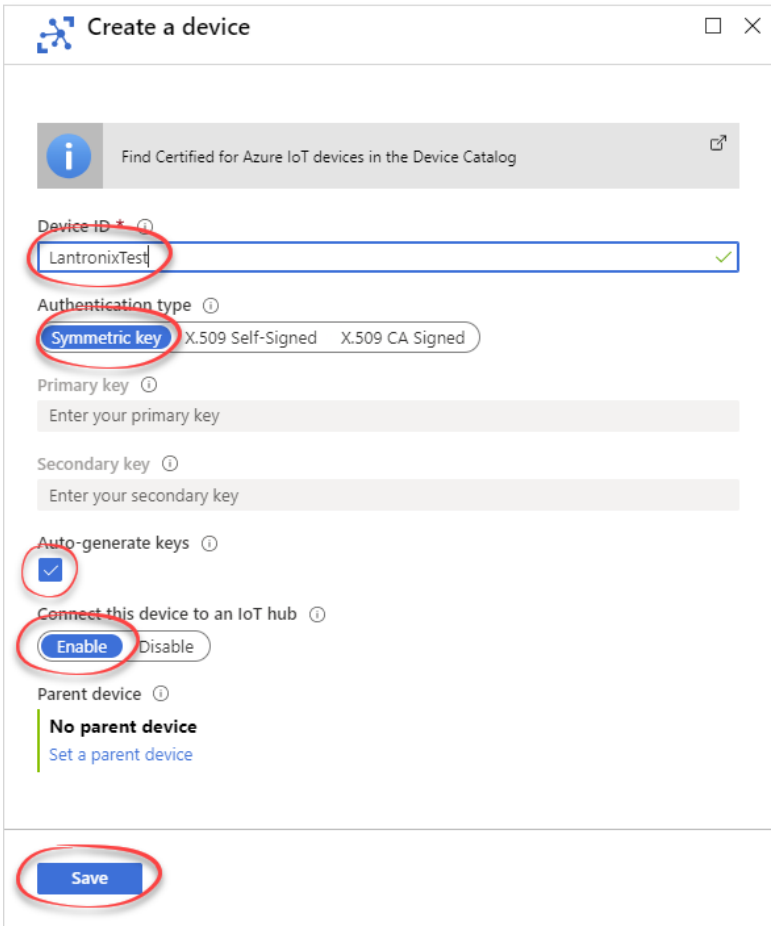
4. On the left, click **IoT devices**.



5. Click **New** at the top.

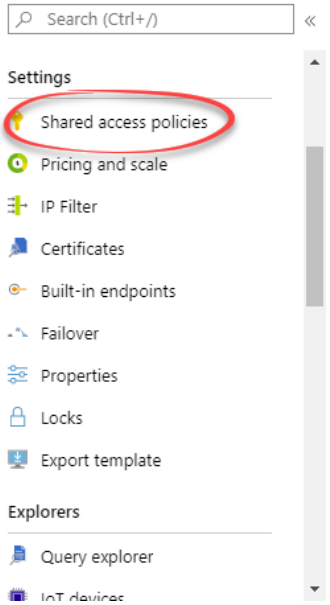


6. Enter a unique name for the device in **Device ID**. Under **Authentication type**, select **Symmetric key**. Leave **Auto-generate keys** checked. Leave **Connect this device to an IoT hub** set to **Enable**. Click **Save**.

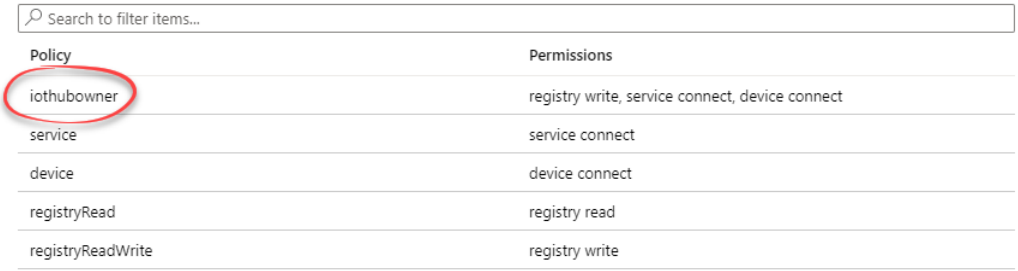


7. Click **Home**.
8. Click your **IoT Hub**.

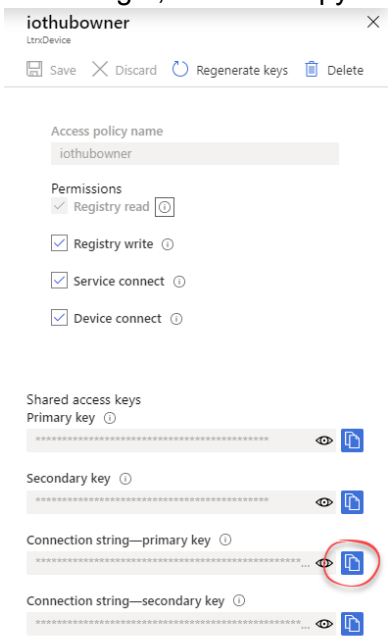
9. Click **Shared access policies** on the left.



10. Under **Policy**, click **iothubowner**.

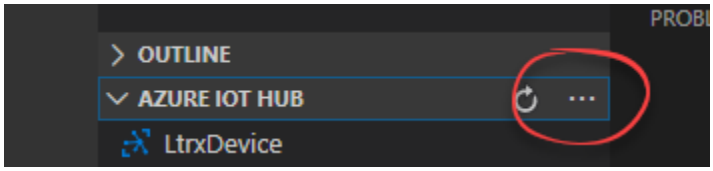


11. On the right, click the copy button next to **Connection string – primary key**.

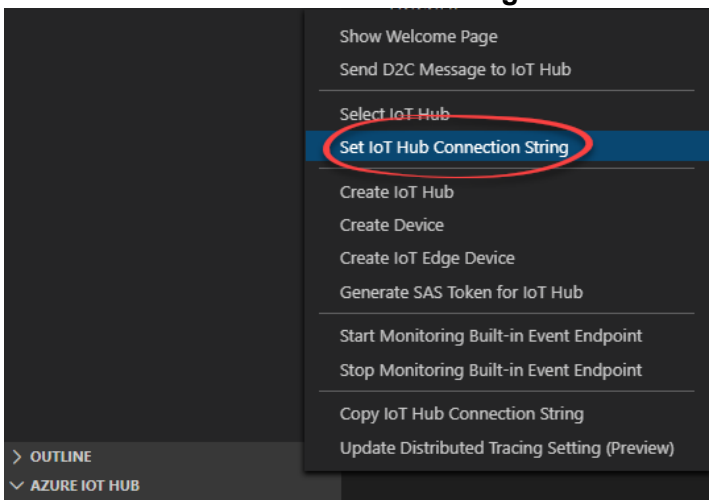


STEP 2. Generate a SAS Token for the device.

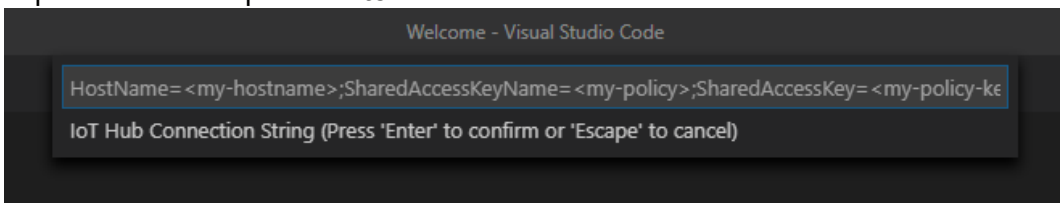
1. Open **Visual Studio Code** with Azure IoT Tools installed.
2. If you are not signed in to Azure, you will be prompted to sign in to Azure and select a subscription.
3. In the bottom-left, click **AZURE IOT HUB** and then click the menu button.



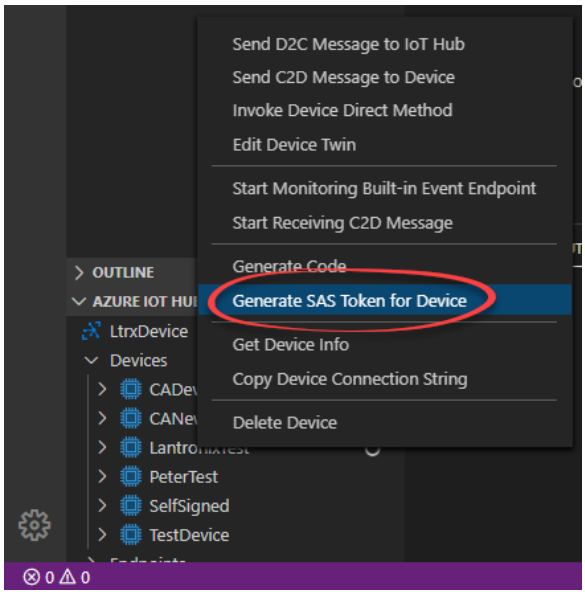
4. Select **Set IoT Hub Connection String**.



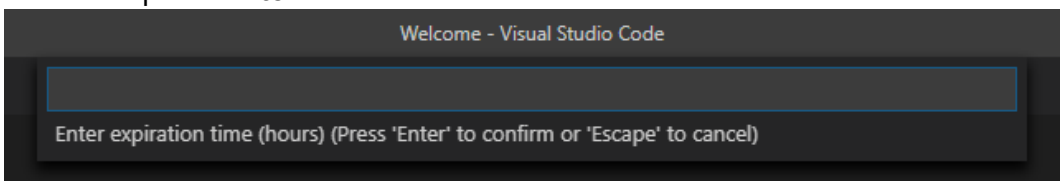
5. A prompt will appear at the top of the window. Enter the connection string you copied earlier and press **Enter**.



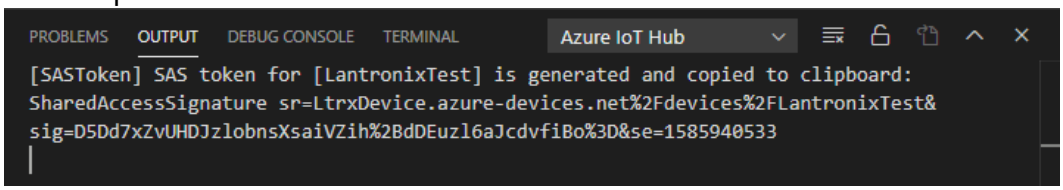
6. A device list will be shown under AZURE IOT HUB. Right-click the device you created and select **Generate SAS Token for Device**.



7. A prompt will appear at the top of the window. Specify the expiration time in hours and press **Enter**.



8. The SAS Token will be generated, shown in the output, and automatically copied to the clipboard.

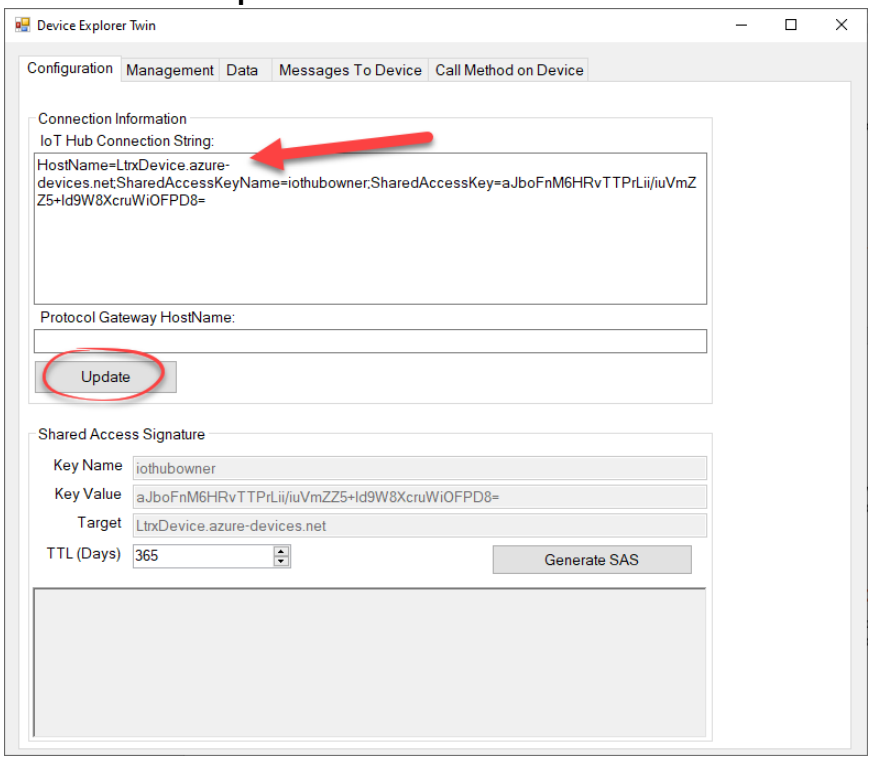


STEP 3. Configure the xPico 200 series device via Web Manager or CLI

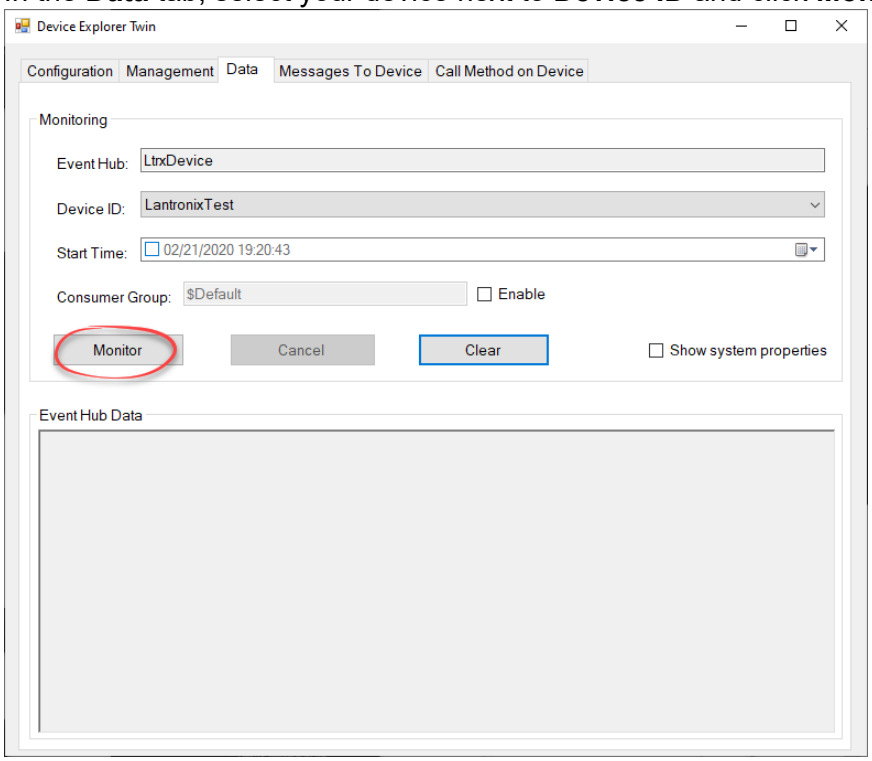
1. Set the **Line 1 protocol** to Azure IoT.
2. Set the Azure Configuration as follows:
   State: Enabled
   Hub Name: The name of the IoT Hub in Azure
   Device ID: The Device ID set in Azure when the device was created
   Security: Security Keys
   SAS Token: The SAS Token generated in Visual Studio Code
   MQTT Local Port: <Random>

STEP 4. Test the integration

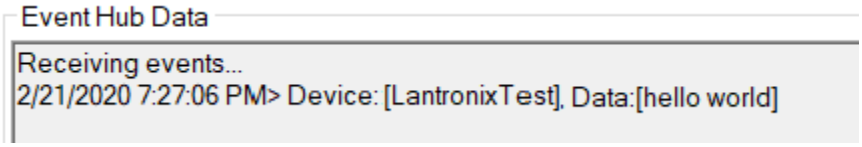
1. Using TeraTerm, connect to the xPico 200 series device via serial connection on line 1.
2. Click **Setup** > **Terminal**.
3. Under **New-line**, set **Receive** and **Transmit** to **LF** and click **OK**.
4. Open **Device Explorer**.
5. Under **IoT Hub Connection String**, paste the connection string you obtained earlier and click **Update**.



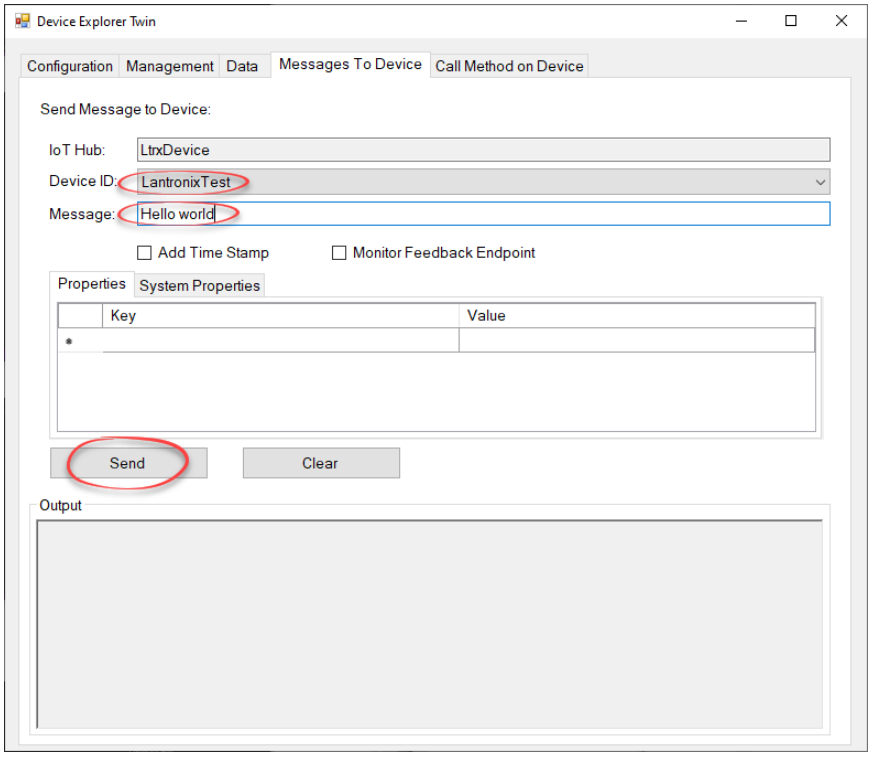
6. In the **Data** tab, select your device next to **Device ID** and click **Monitor**.



7. In TeraTerm, type some characters and hit **Enter**. The data will appear under **Event Hub Data** in Device Explorer.



8. In the **Messages To Device** tab, select your device next to **Device ID**, type some characters next to **Message**, and click **Send**. The data will appear in TeraTerm.



Adding a device using a self-signed X.509 certificate

To add a device:

STEP 1. Create a self-signed certificate

1. Open a Linux, CygWin, or MinGW terminal.
2. Generate an openssl private key.

```
$ openssl req -x509 -newkey rsa:4096 -nodes -keyout key.pem -out cert.pem -
days 365

Generating a 4096 bit RSA private key
.....++
.....++
writing new private key to 'key.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
```

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

3. Generate a certificate signing request (CSR).
   $ openssl req -new -key server.key -out server.csr
   Enter pass phrase for server.key:
   You are about to be asked to enter information that will be incorporated
   into your certificate request.
   What you are about to enter is what is called a Distinguished Name or a DN.
   There are quite a few fields but you can leave some blank
   For some fields there will be a default value,
   If you enter '.', the field will be left blank.

   Country Name (2 letter code) [AU]:US
   State or Province Name (full name) [Some-State]:CA
   Locality Name (eg, city) []:Irvine
   Organization Name (eg, company) [Internet Widgits Pty Ltd]:Lantronix
   Organizational Unit Name (eg, section) []:Engineering
   Common Name (e.g. server FQDN or YOUR name) []:<device IP address>
   Email Address []:.

   Please enter the following 'extra' attributes
   to be sent with your certificate request
   A challenge password []:<secret>
   An optional company name []:

4. Remove the passphrase from the key.
   $ cp server.key server.key.org

   $ openssl rsa -in server.key.org -out server.key
   Enter pass phrase for server.key.org:
   writing RSA key

5. Generate a self-signed certificate.
   $ openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.pem
   Signature ok
   subject=/C=US/ST=CA/L=Irvine/O=Lantronix/OU=Engineering/CN=<device IP address>
   Getting Private key

STEP 2. Configure the xPico 200 series device via Web Manager or CLI

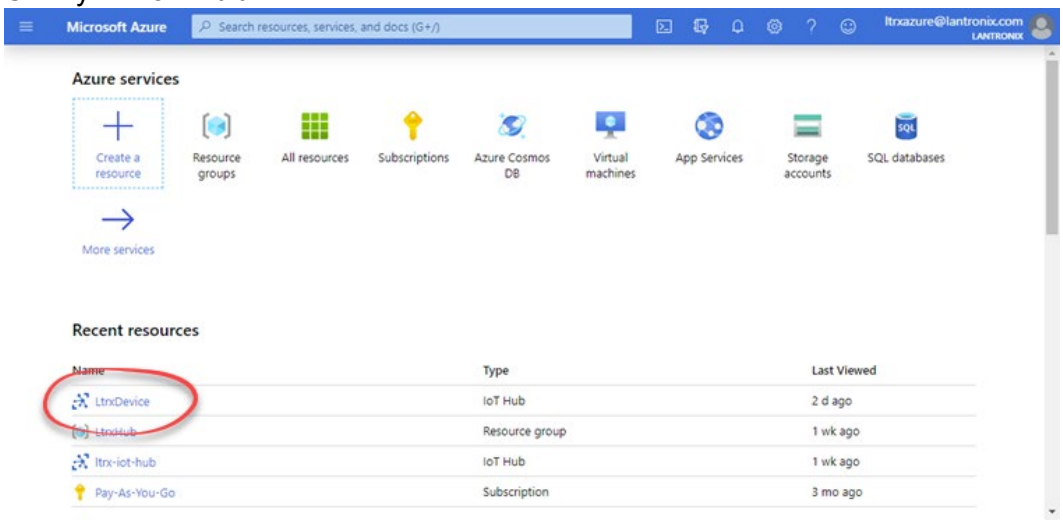
1. Create a new TLS Credential.
2. Set the **Private Key** to the contents of server.key, which can be opened with a text editor such as Notepad. Include both the beginning and ending lines as well:
   -----BEGIN RSA PRIVATE KEY-----

<Your base64 encoded private key will be in here.>
-----END RSA PRIVATE KEY-----

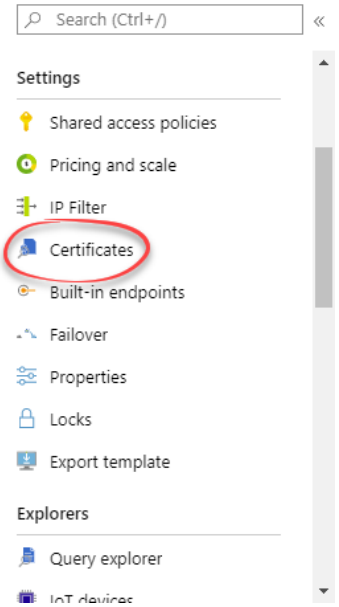
3. Set the **Certificate** to the contents of server.pem, which can be opened with a text editor such as Notepad. Include both the beginning and ending lines as well:
-----BEGIN CERTIFICATE-----
<Your base64 encoded certificate will be in here.>
-----END CERTIFICATE-----

STEP 3. Register the X.509 self-signed certificates to your IoT Hub

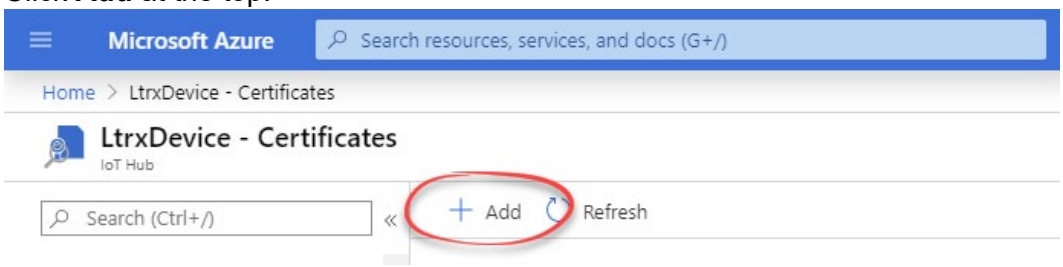
1. Log in to https://portal.azure.com.
2. Click **Home** if you are not already at the home page.
3. Click your **IoT Hub**.



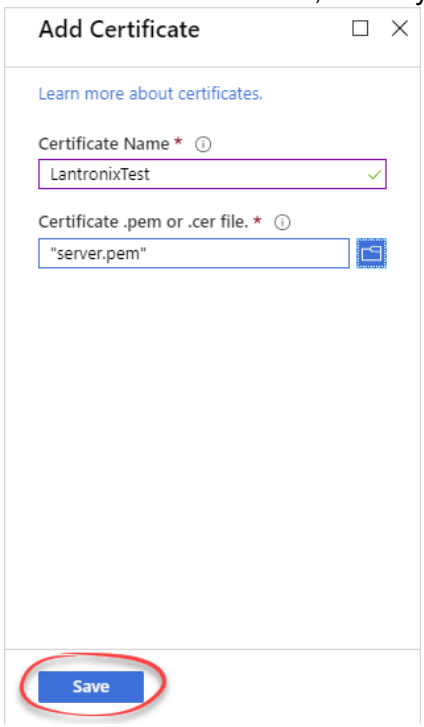
4. Click **Certificates** on the left.



5. Click **Add** at the top.

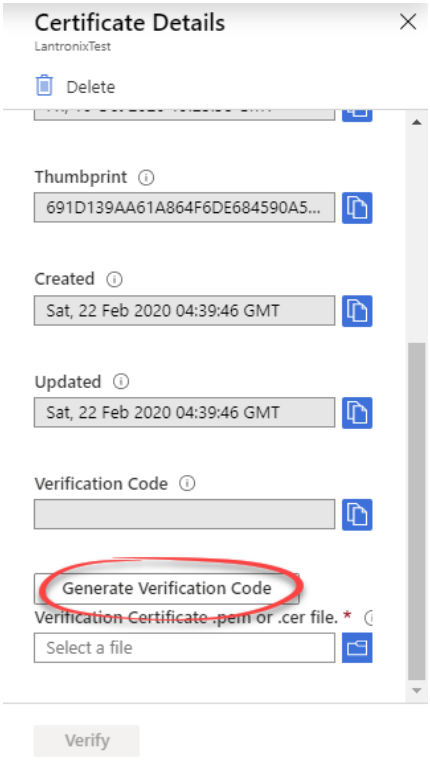


6. Enter a certificate name, select your **server.pem** file, and click **Save**.

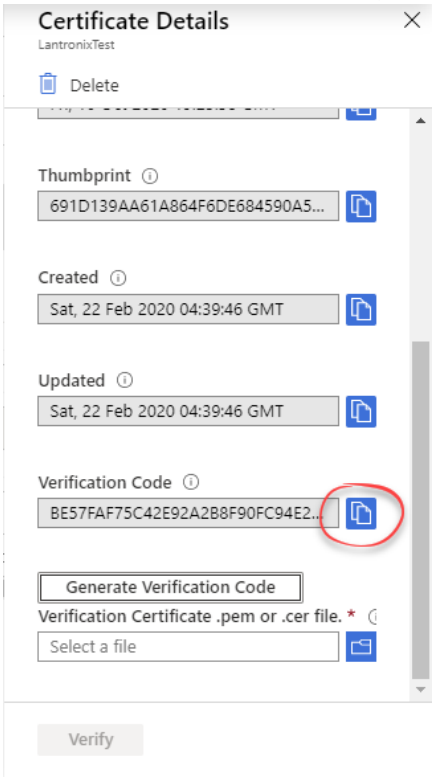


7. Click the certificate you created to open Certificate Details on the right.

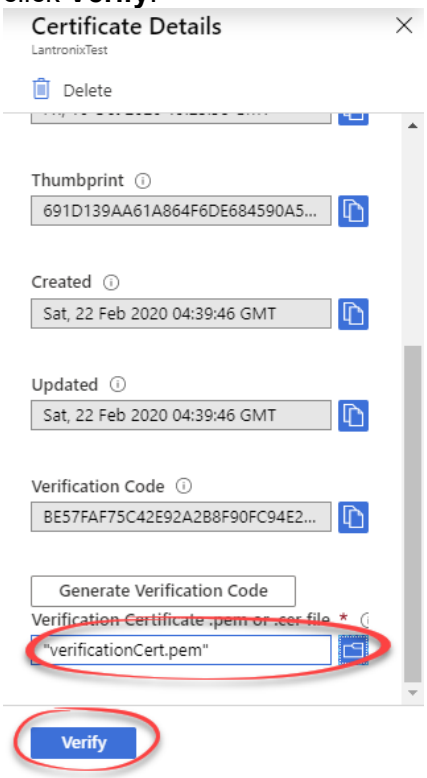
8. Under **Verification Code**, click **Generate Verification Code** to generate a Verification Code.



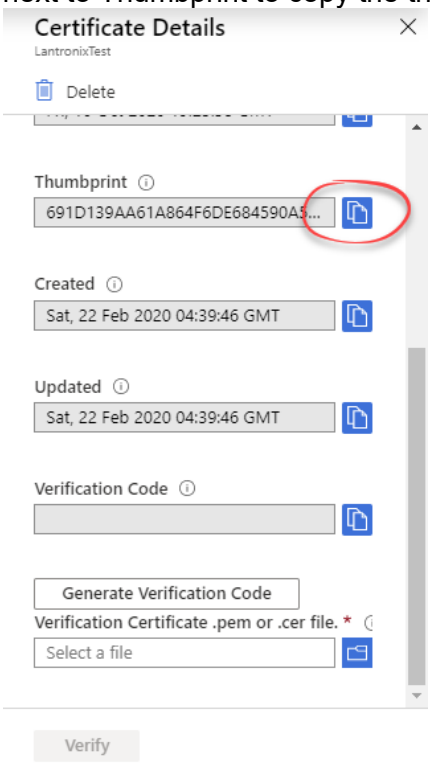
9. Next to the Verification Code, click the **blue copy button** to copy the code.



10. In a Linux, CygWin, or MinGW terminal, create the verification key.
    $ openssl genrsa -out verification.key 2048
11. Create the verification certificate using the verification key.
    $ openssl req -new -key verification.key -out verification.csr
12. Specify the verification code copied previously when prompted.
13. Create the proof of possession certificate using the verification certificate.
    $ openssl x509 -req -in verification.csr -CA server.pem -CAkey server.key -CAcreateserial -out verificationCert.pem -days 1024 -sha256
14. In the Certificate Details panel in Azure Portal, upload **verificationCert.pem** and click **Verify**.

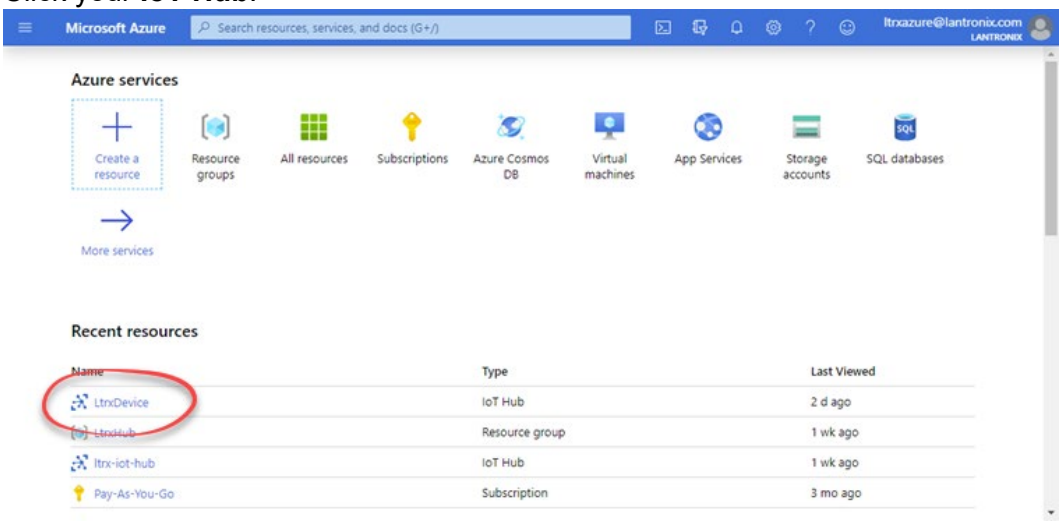


15. Click the certificate you created. In Certificate Details, click the **blue copy button** next to Thumbprint to copy the thumbprint.

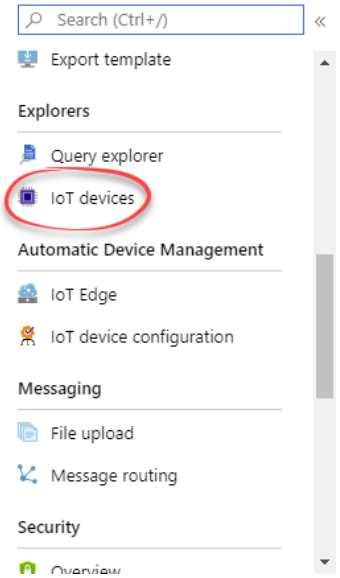


STEP 4. Create a device in your IoT Hub

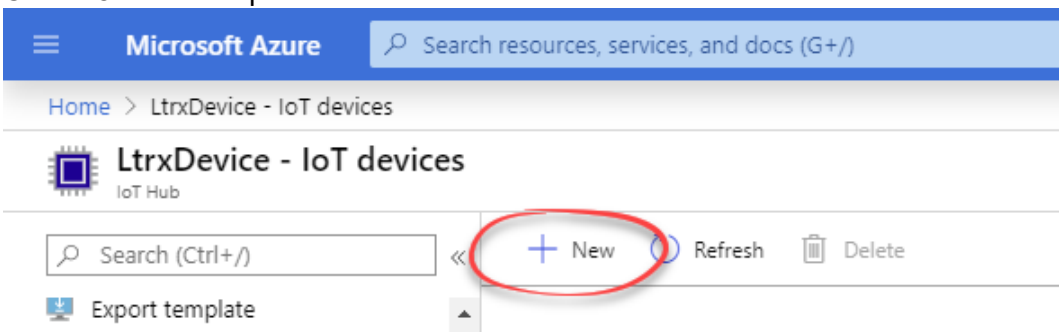
1. Click **Home** if you are not already at the home page.
2. Click your **IoT Hub**.



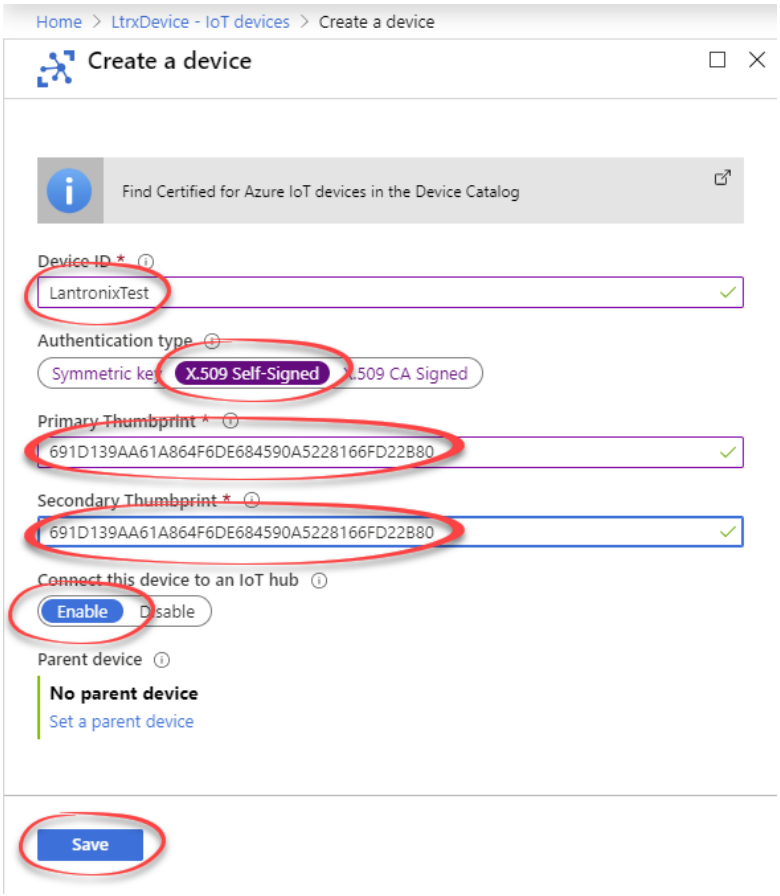
3. On the left, click **IoT devices**.



4. Click **New** at the top.



5. Enter a unique name for the device in **Device ID**. Under **Authentication type**, select **X.509 Self-Signed**. Paste the thumbprint you copied earlier under **Primary Thumbprint** and **Secondary Thumbprint**. Leave **Connect this device to an IoT hub** set to **Enable**. Click **Save**.



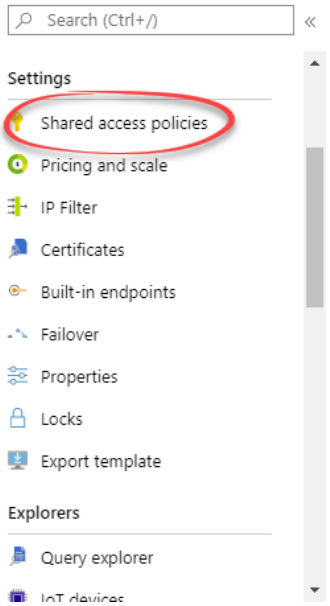
STEP 5. Configure the xPico 200 series device via Web Manager or CLI

1. Create a new TLS Credential.
2. Set the **Private Key** to the contents of server.key, which can be opened with a text editor such as Notepad.
3. Set the **Certificate** to the contents of server.pem, which can be opened with a text editor such as Notepad.
4. Set **Trusted Authority 1** to the contents of D-TRUST.PEM, **Trusted Authority 2** to the contents of DigiCert.pem, and **Trusted Authority 3** to the contents of baltimore-ca.pem.
5. Set the **Line 1 protocol** to Azure IoT.
6. Set the Azure Configuration as follows:
   State: Enabled
   Hub Name: The name of the IoT Hub in Azure
   Device ID: The Device ID set in Azure when the device was created

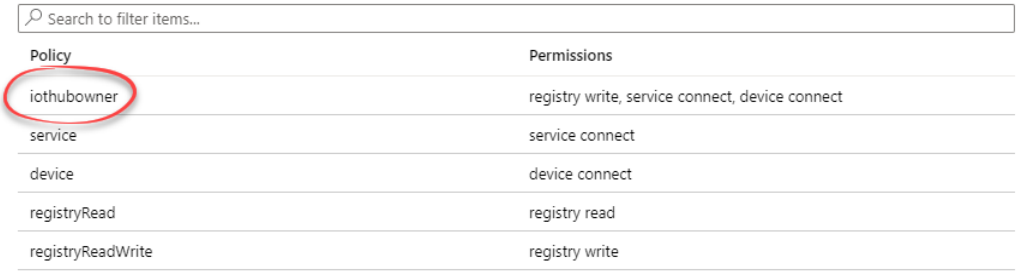
Security: X.509
Credential Name: <the name of the TLS credential>
MQTT Local Port: <Random>

STEP 6. Obtain the connection string

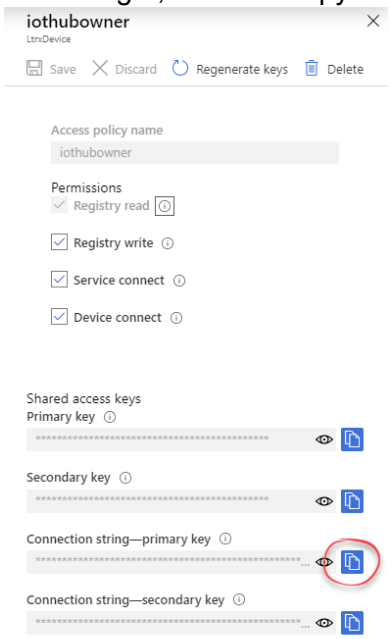
1. In Azure Portal, click **Home**.
2. Click your **IoT Hub**.
3. Click **Shared access policies** on the left.



4. Under **Policy**, click **iothubowner**.



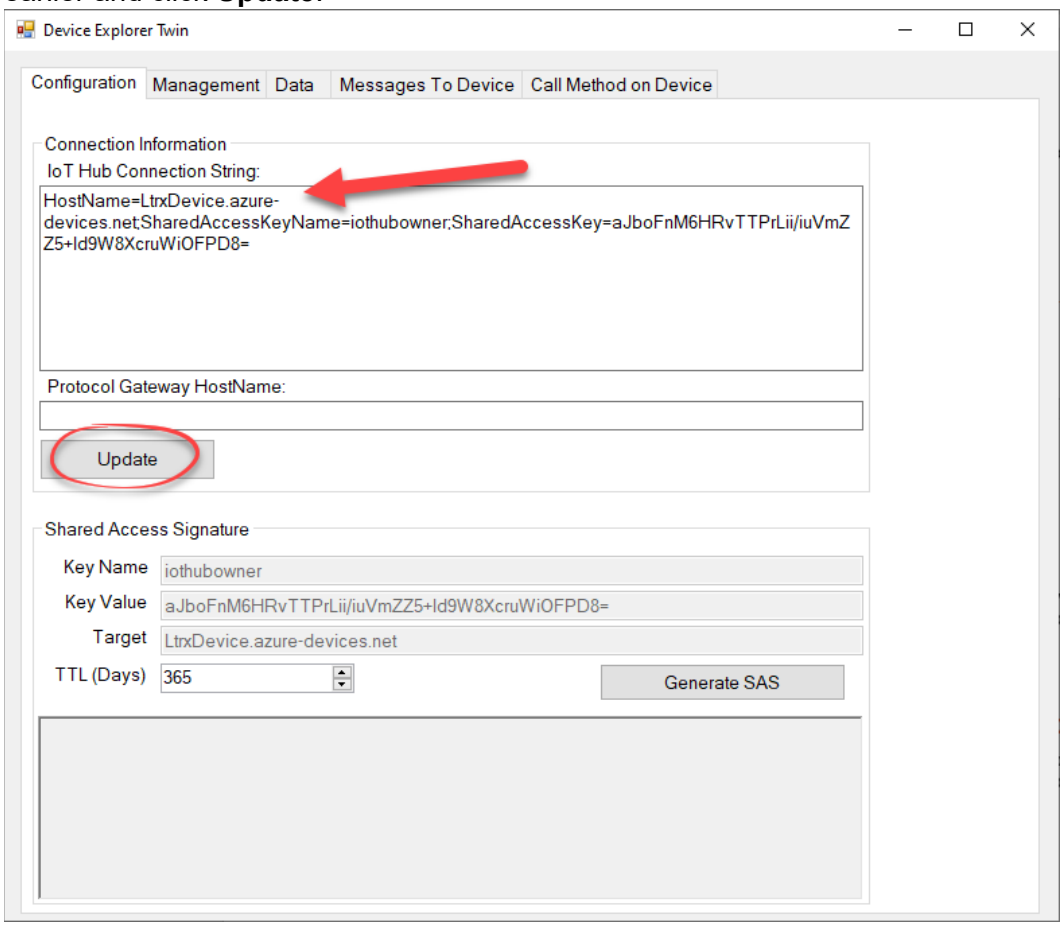
5. On the right, click the copy button next to **Connection string – primary key**



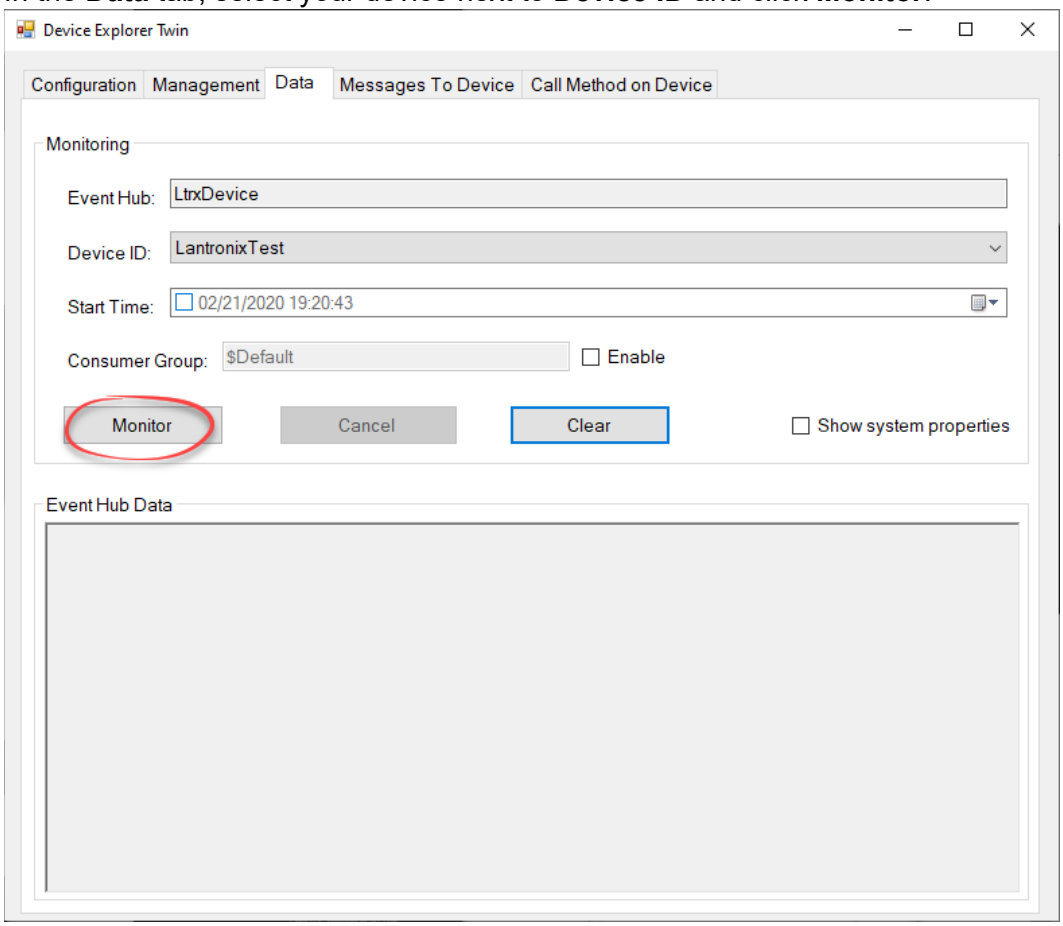
STEP 7. Test the integration

1. Using TeraTerm, connect to the xPico 200 series device via serial connection on line 1.
2. Click **Setup** > **Terminal**.
3. Under **New-line**, set **Receive** and **Transmit** to **LF** and click **OK**.
4. Open **Device Explorer**.

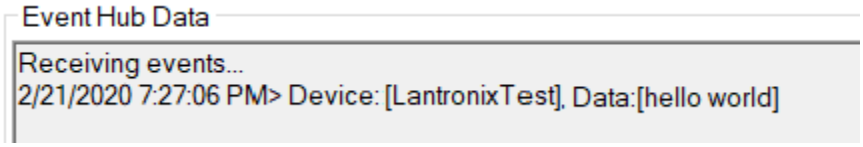
5. Under **IoT Hub Connection String**, paste the connection string you obtained earlier and click **Update**.



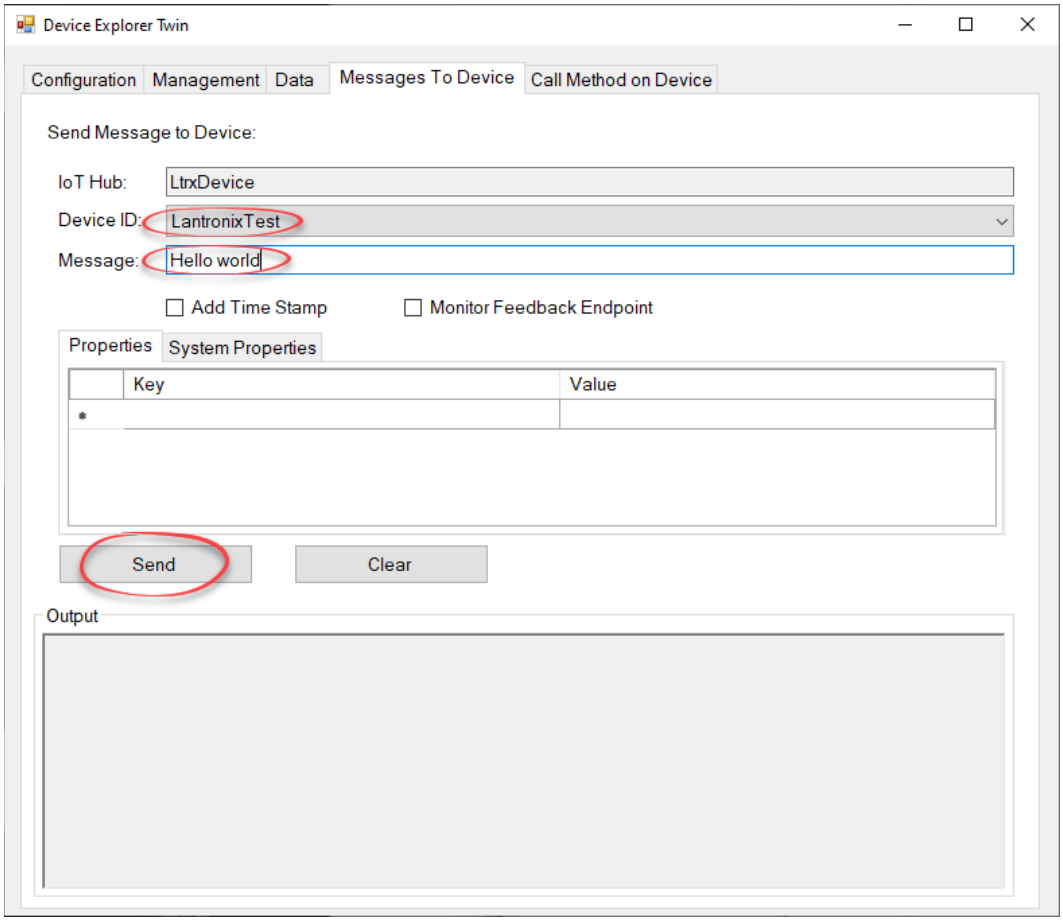
6. In the **Data** tab, select your device next to **Device ID** and click **Monitor**.



7. In TeraTerm, type some characters and hit **Enter**. The data will appear under **Event Hub Data** in Device Explorer.



8. In the **Messages To Device** tab, select your device next to **Device ID**, type some characters next to **Message**, and click **Send**. The data will appear in TeraTerm.



Adding a device using a CA-signed X.509 certificate

To add a device:

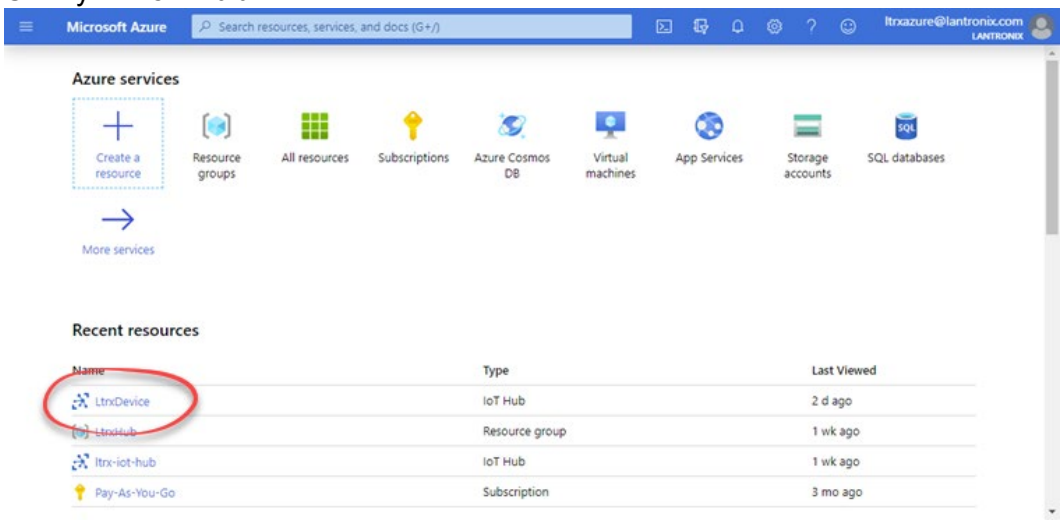
STEP 1. Generate the root and intermediate certificates

1. Download all files from https://github.com/Azure/azure-iot-sdk-c/tree/master/tools/CACertificates.
2. Open a Linux, CygWin, or MinGW terminal.
3. Go to the directory storing the files that you downloaded.
4. Set certGen.sh as an executable.
   chmod +x certGen.sh
5. Create the root and intermediate certificates using the following command:
   ./certGen.sh create_root_and_intermediate

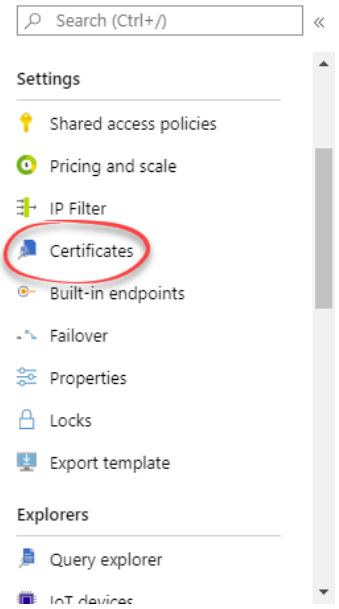
STEP 2. Register the certificates to your IoT Hub

1. Log in to https://portal.azure.com.

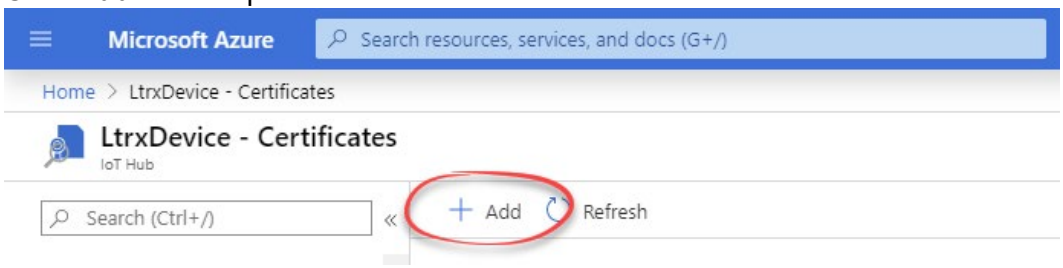
2. Click **Home** if you are not already at the home page.
3. Click your **IoT Hub**.



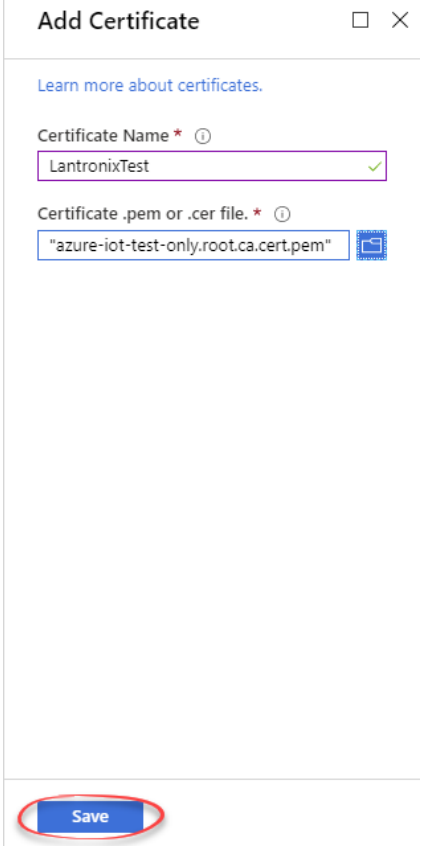
4. Click **Certificates** on the left.



5. Click **Add** at the top.

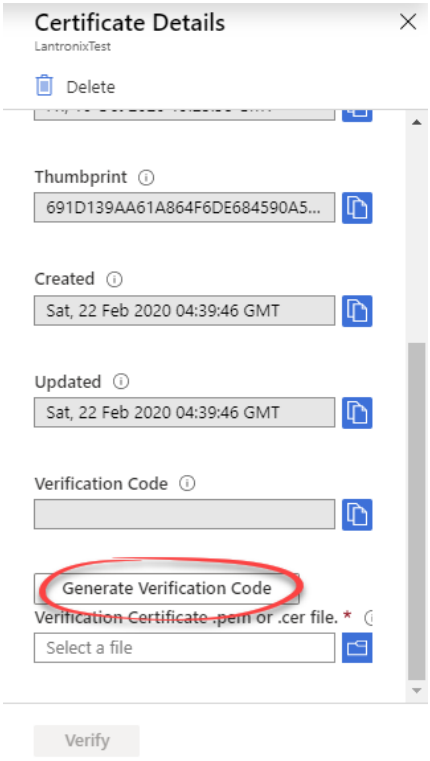


6. Enter a certificate name, select your **azure-iot-test-only.root.ca.cert.pem** file, and click **Save**.

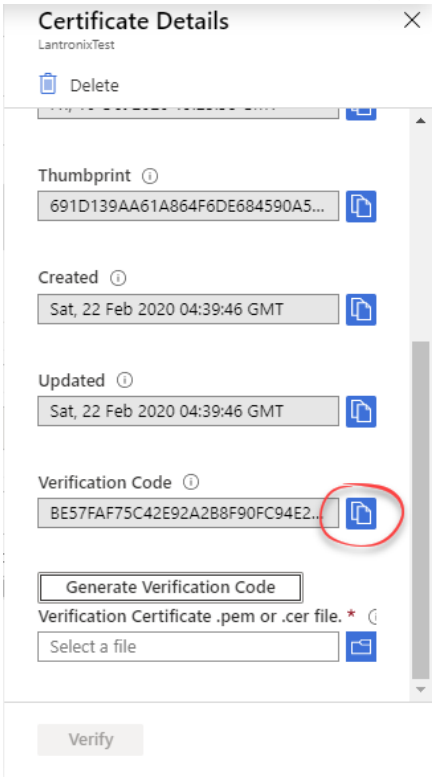


7. Click the certificate you created to open Certificate Details on the right.

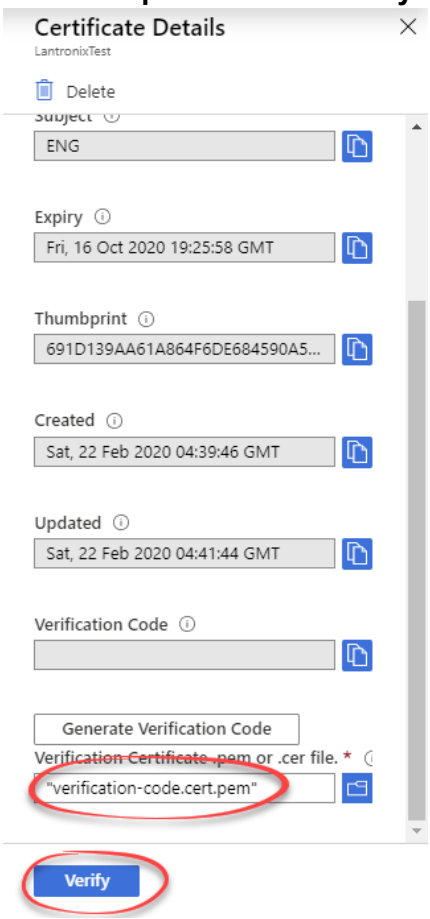
8. Under **Verification Code**, click **Generate Verification Code** to generate a Verification Code.



9. Next to the Verification Code, click the **blue copy button** to copy the code.



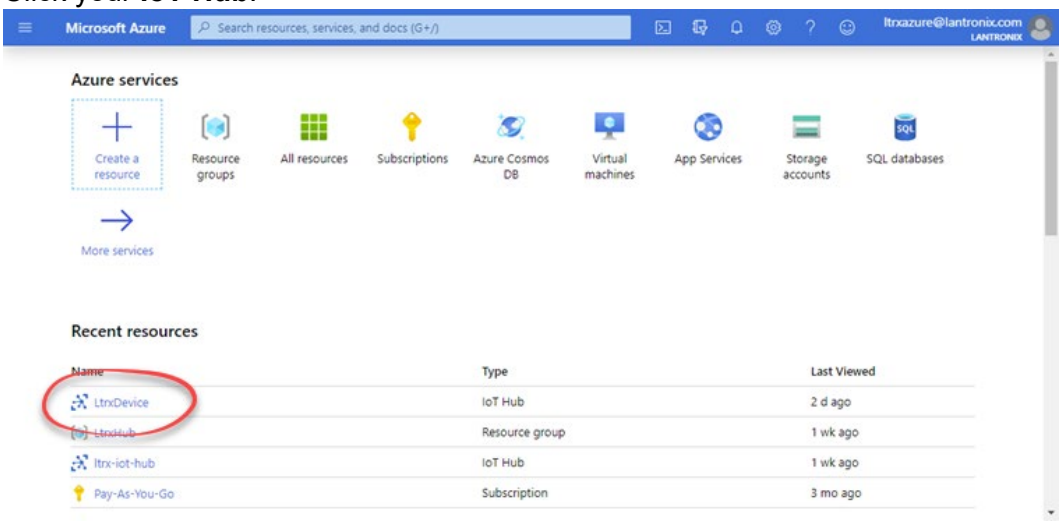
10. From the terminal used earlier, generate the verification certificate. ./certGen.sh create_verification_certificate <Verification Code copied in the previous step>
11. In the Certificate Details panel in Azure Portal, upload **verification-code.cert.pem** and click **Verify**.



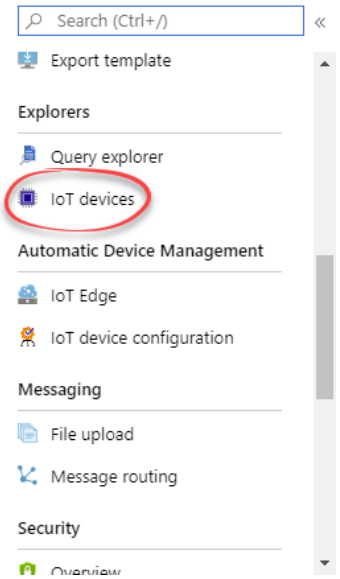
STEP 3. Create a device in your IoT Hub

1. Click **Home** if you are not already at the home page.

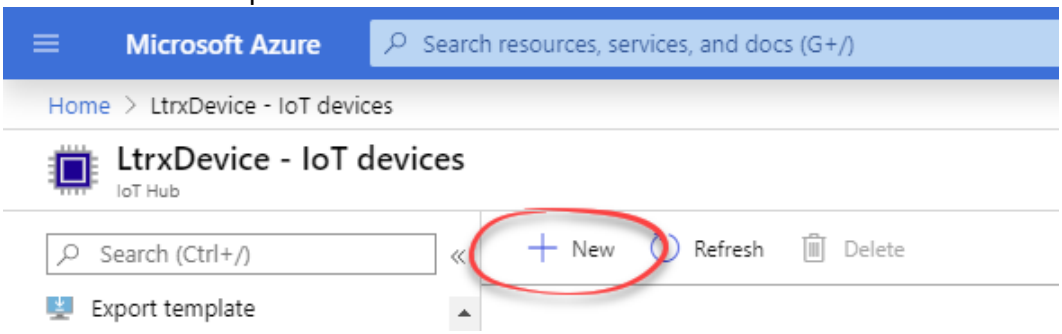
2. Click your **IoT Hub**.



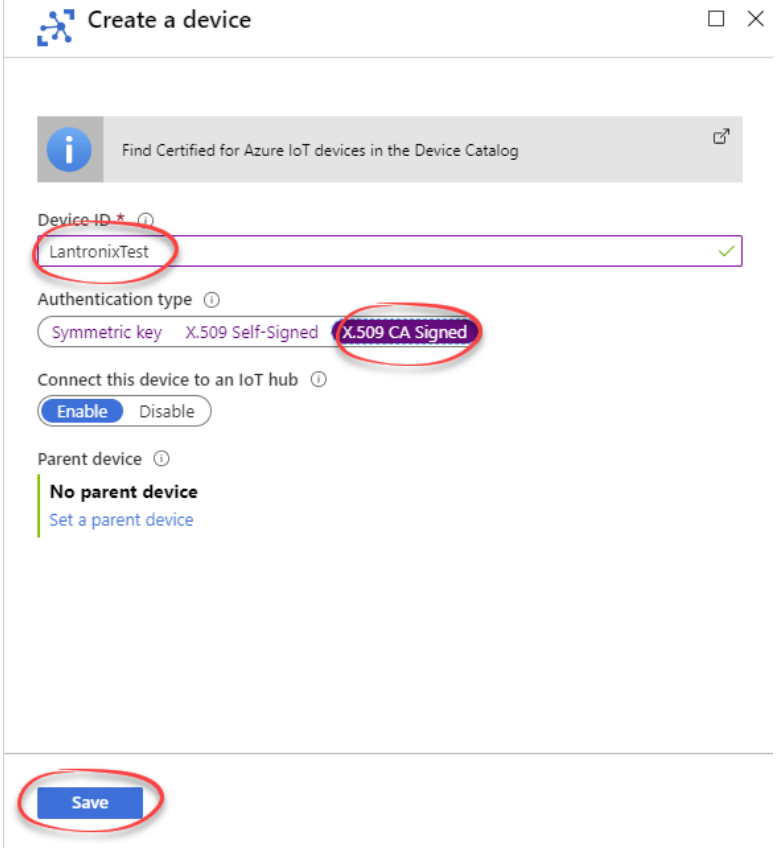
3. On the left, click **IoT devices**.



4. Click **New** at the top.



5. Enter a unique name for the device in **Device ID**. Under **Authentication type**, select **X.509 CA Signed**. Click **Save**.



6. From the terminal used earlier, generate the device certificate (new-device.key.pem and device.cert.pem).
   ./certGen.sh create_device_certificate <Device ID>

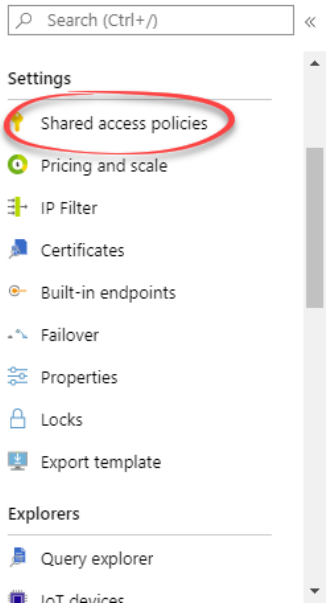
STEP 4. Configure the xPico 200 series device via Web Manager or CLI

1. Create a new TLS Credential.
2. Set the **Private Key** to the contents of new-device.key.pem, which can be opened with a text editor such as Notepad.
3. Set the **Certificate** to the contents of new-device.cert.pem, which can be opened with a text editor such as Notepad.
4. Set **Higher Authority 1** to the contents of azure-iot-test-only.root.ca.cert.pem.
5. Set **Trusted Authority 1** to the contents of D-TRUST.PEM, **Trusted Authority 2** to the contents of DigiCert.pem, and **Trusted Authority 3** to the contents of baltimore-ca.pem.
6. Set the **Line 1 protocol** to Azure IoT.
7. Set the Azure Configuration as follows:
   State: Enabled
   Hub Name: The name of the IoT Hub in Azure
   Device ID: The Device ID set in Azure when the device was created

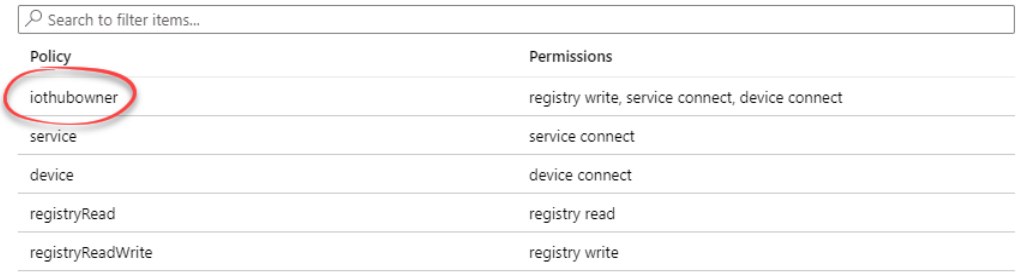
Security: X.509
Credential Name: <the name of the TLS credential>
MQTT Local Port: <Random>

STEP 5. Obtain the connection string

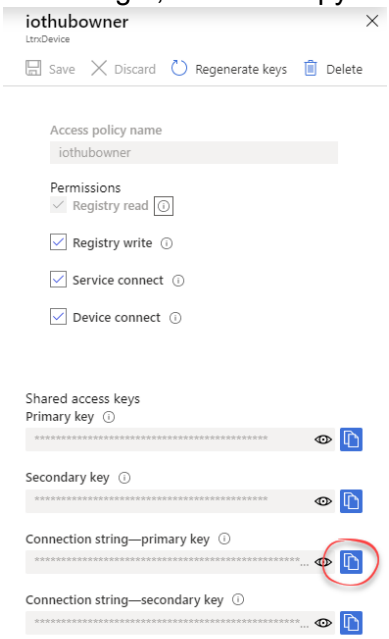
1. In Azure Portal, click **Home**.
2. Click your **IoT Hub**.
3. Click **Shared access policies** on the left.



4. Under **Policy**, click **iothubowner**.



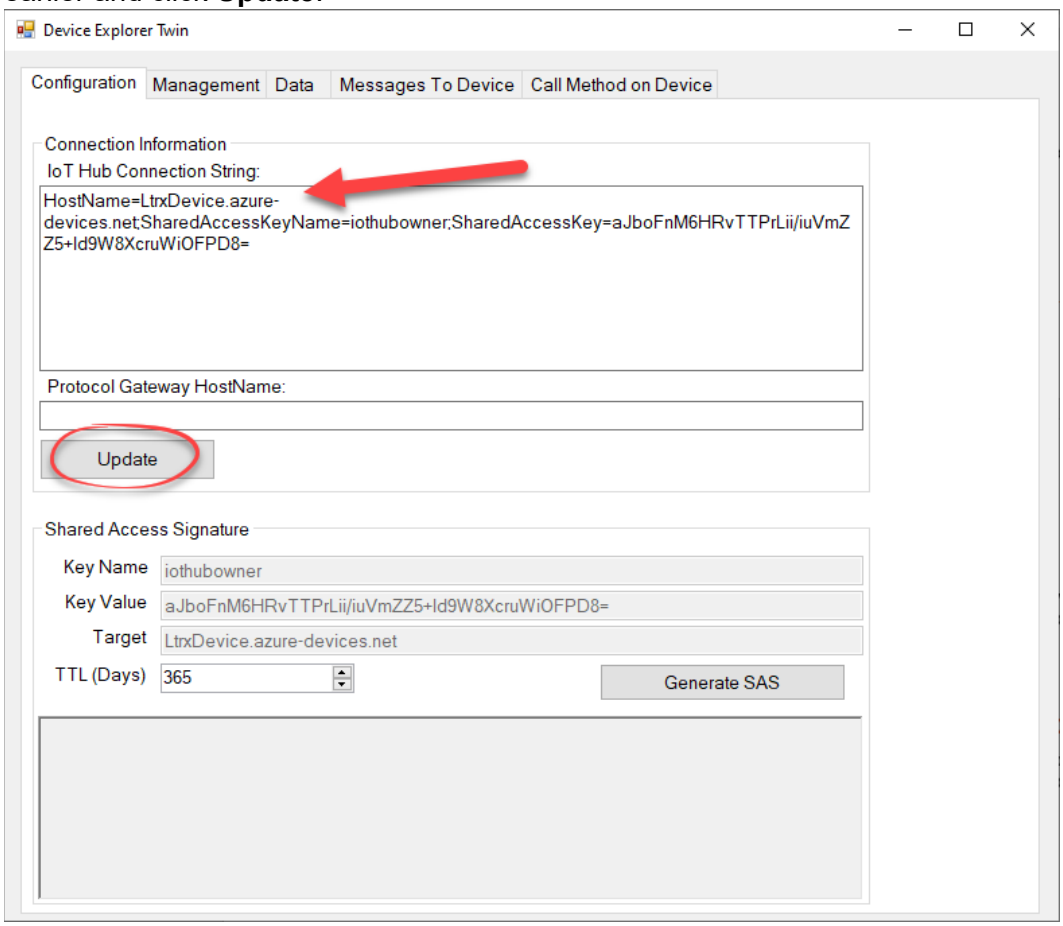
5. On the right, click the copy button next to **Connection string – primary key**



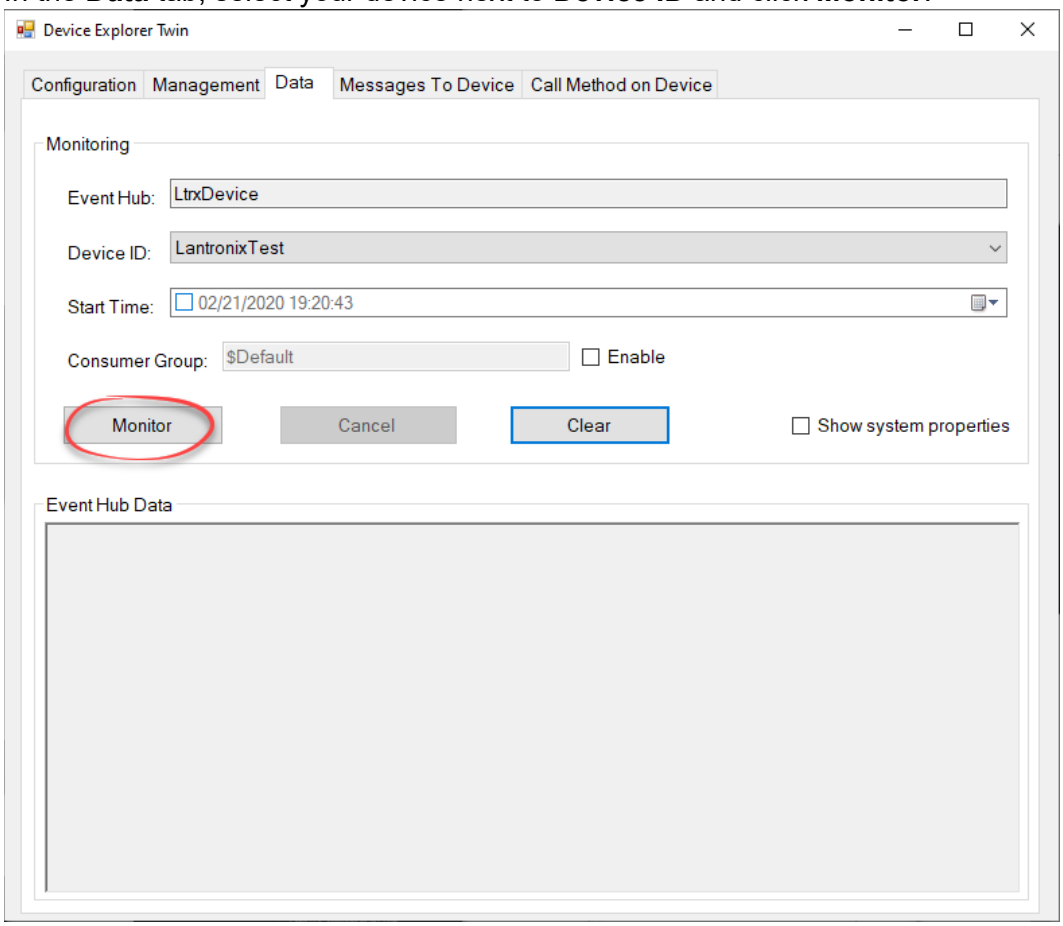
STEP 6. Test the integration

1. Using TeraTerm, connect to the xPico 200 series device via serial connection on line 1.
2. Click **Setup** > **Terminal**.
3. Under **New-line**, set **Receive** and **Transmit** to **LF** and click **OK**.
4. Open **Device Explorer**.

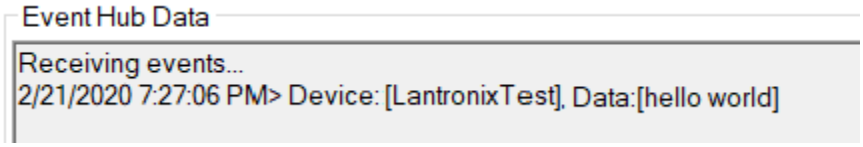
5. Under **IoT Hub Connection String**, paste the connection string you obtained earlier and click **Update**.



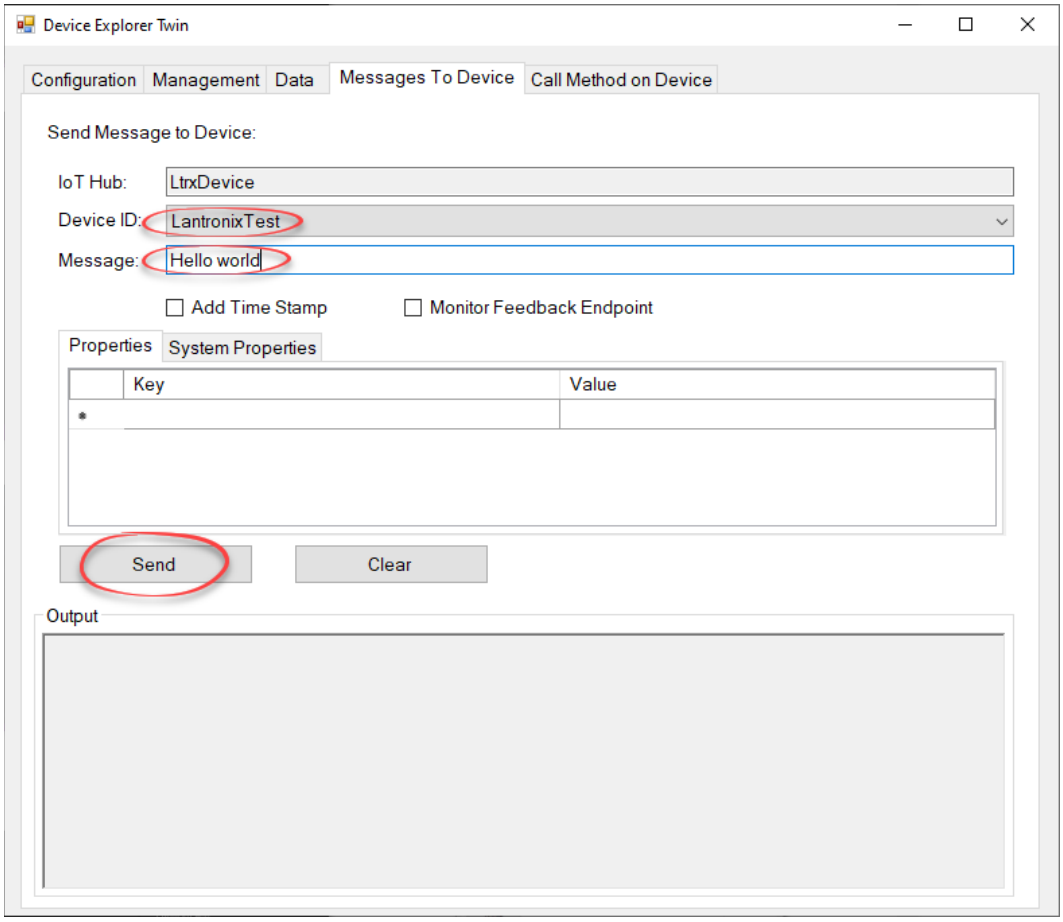
6. In the **Data** tab, select your device next to **Device ID** and click **Monitor**.



7. In TeraTerm, type some characters and hit **Enter**. The data will appear under **Event Hub Data** in Device Explorer.



8. In the **Messages To Device** tab, select your device next to **Device ID**, type some characters next to **Message**, and click **Send**. The data will appear in TeraTerm.



4 Calling Methods on an xPico 200 Series Device

You can call the following methods directly in Device Explorer. The line protocol on the device does not need to be set to Azure IoT to call methods.

- ltrx_import_xml_config – imports XML configuration or performs action
- ltrx_trusted_import_xml_config – imports XML configuration or performs action. Compared to ltrx_import_xml_config, ltrx_trused_import_xml_config is intended for programmers who have already tested their XML. The XML header must be left out. It has less error checking than ltrx_import_xml_config and eliminates multiple passes. It does not need to cache the XML in flash memory and is faster. While ltrx_import_xml_config saves the configuration if successful, ltrx_trused_import_xml_config does not, like a CLI command.
- ltrx_read_xml_status – reads XML status
- ltrx_read_xml_config – reads XML configuration

The following examples show how to use these methods. In all examples, the device must be added to an IoT Hub in Microsoft Azure, and Azure State must be Up.

Examples

ltrx_import_xml_config

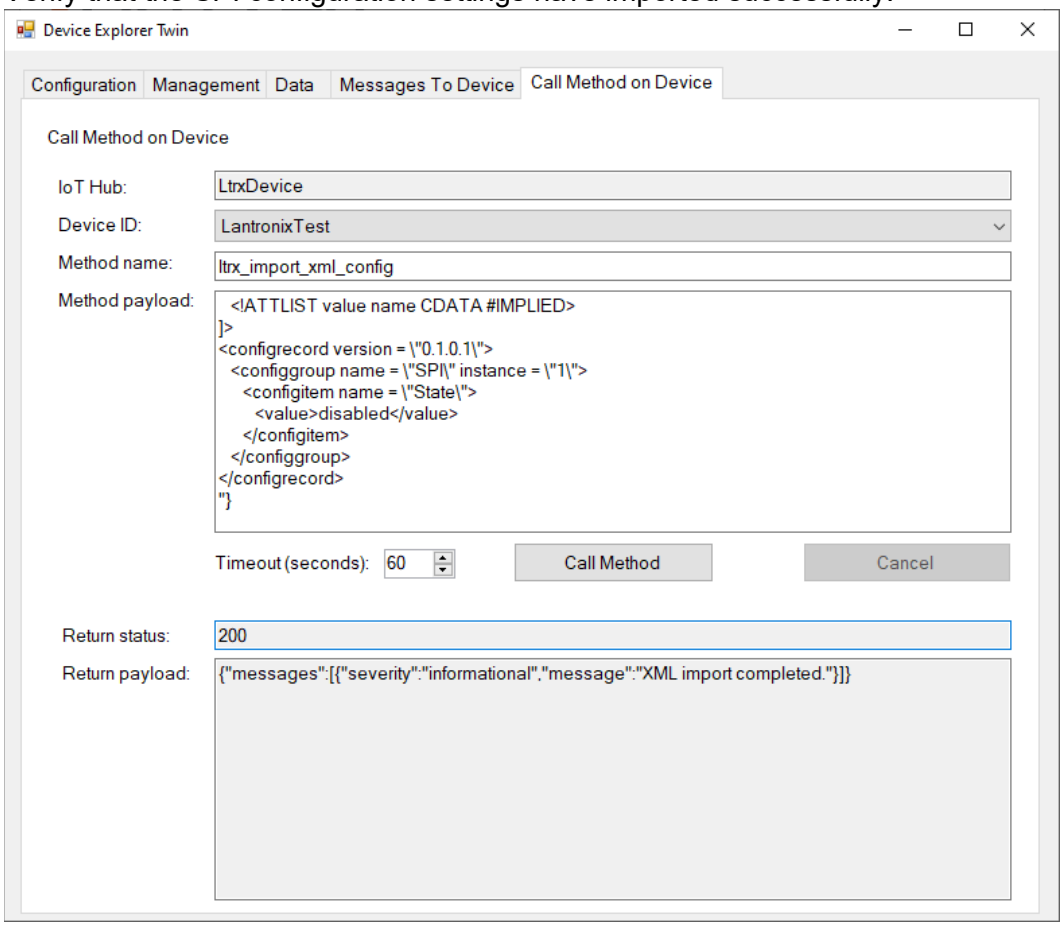
Description: Import XML (configgroup: SPI)

1. Open Device Explorer and click the **Call Method on Device** tab.
2. Select your device next to **Device ID**.
3. Type **ltrx_import_xml_config** next to **Method name**.
4. Next to **Method payload**, copy and paste the below XML and then click **Call Method**.

```
{"param":"
<?xml version=\"1.0\" standalone=\"yes\"?>
<!-- Automatically generated XML -->
<!DOCTYPE configrecord [
   <!ELEMENT configrecord (configgroup+)>
   <!ELEMENT configgroup (configitem+)>
   <!ELEMENT configitem (value+)>
   <!ELEMENT value (#PCDATA)>
   <!ATTLIST configrecord version CDATA #IMPLIED>
   <!ATTLIST configgroup name CDATA #IMPLIED>
   <!ATTLIST configgroup instance CDATA #IMPLIED>
   <!ATTLIST configitem name CDATA #IMPLIED>
   <!ATTLIST configitem instance CDATA #IMPLIED>
   <!ATTLIST value name CDATA #IMPLIED>
```

```
]>
<configrecord version = \"0.1.0.1\">
   <configgroup name = \"SPI\" instance = \"1\">
      <configitem name = \"State\">
         <value>disabled</value>
      </configitem>
   </configgroup>
</configrecord>
"}
```

5. Verify that the SPI configuration settings have imported successfully.



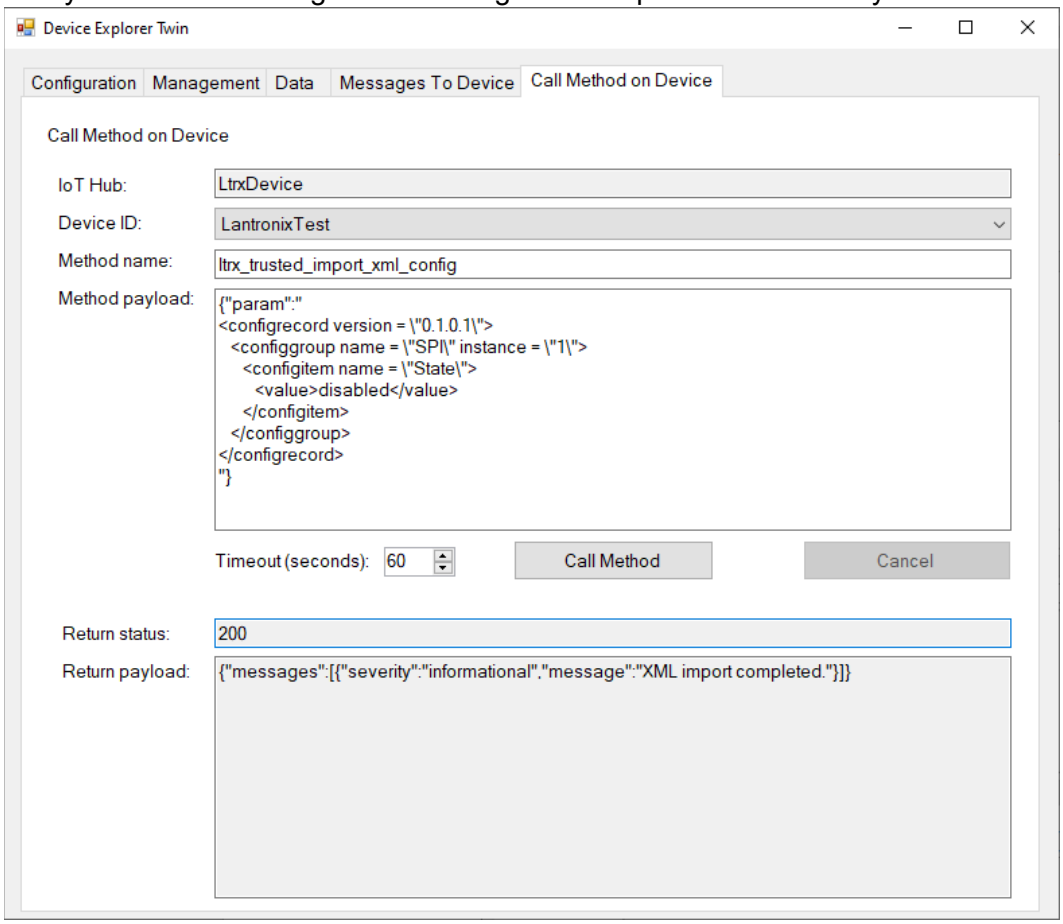
ltrx_trusted_import_xml_config

Description: Trusted import XML (configgroup: SPI)

1. Open Device Explorer and click the **Call Method on Device** tab.
2. Select your device next to **Device ID**.
3. Type **ltrx_trusted_import_xml_config** next to **Method name**.
4. Next to **Method payload**, copy and paste the below XML and then click **Call Method**.

```
{"param":"
<configrecord version = \"0.1.0.1\">
   <configgroup name = \"SPI\" instance = \"1\">
      <configitem name = \"State\">
         <value>disabled</value>
      </configitem>
   </configgroup>
</configrecord>
"}
```

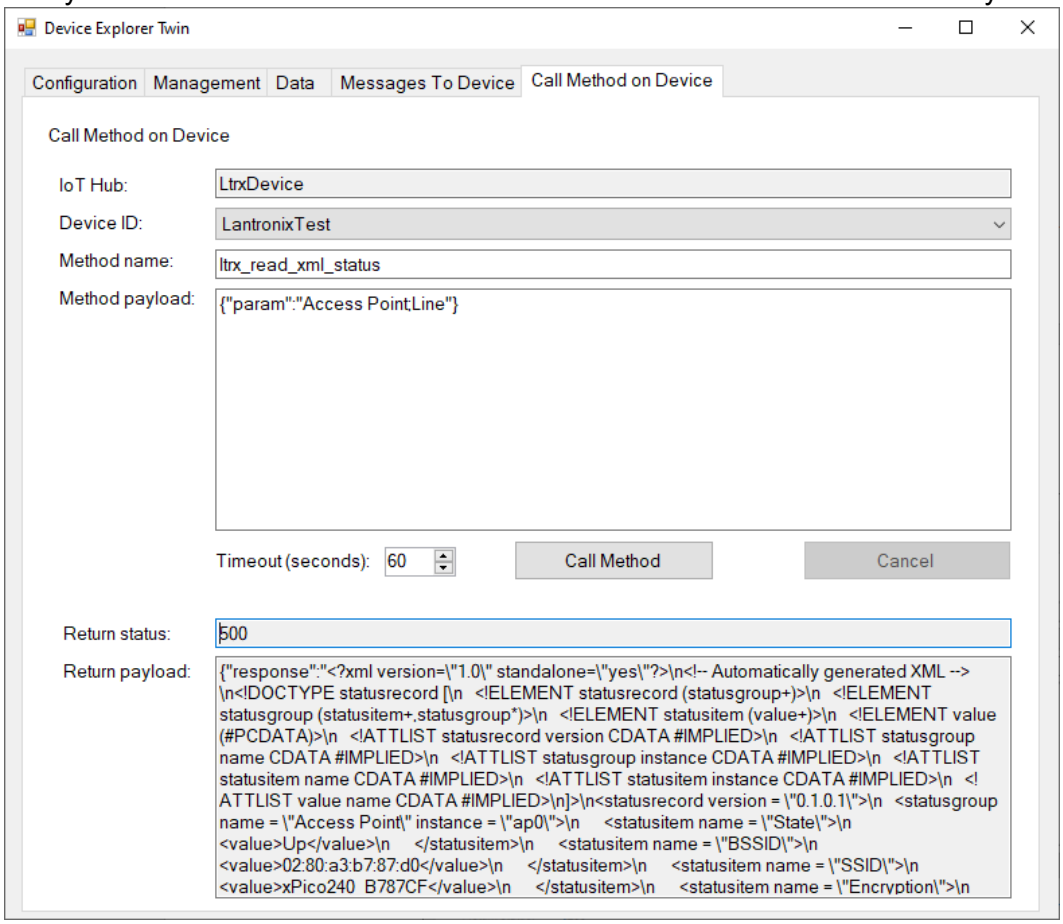
5. Verify that the SPI configuration settings have imported successfully.



ltrx_read_xml_status

Description: Read XML status

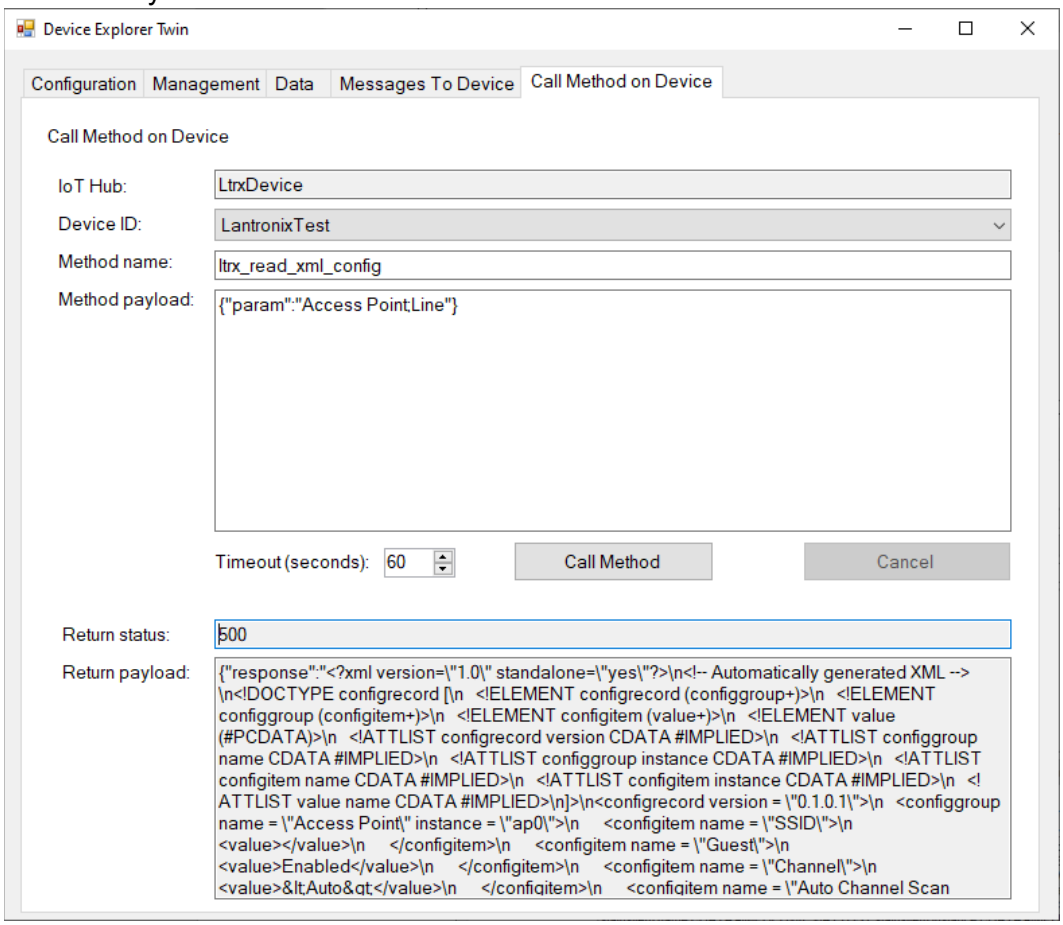
1. Open Device Explorer and click the **Call Method on Device** tab.
2. Select your device next to **Device ID**.
3. Type **ltrx_read_xml_status** next to **Method name**.
4. Next to **Method payload**, copy and paste the below XML and then click **Call Method**.
   `{"param":"Access Point;Line"}`
5. Verify that the Access Point and Line status have been returned successfully.



ltrx_read_xml_config

Description: Read XML configuration

1. Open Device Explorer and click the **Call Method on Device** tab.
2. Select your device next to **Device ID**.
3. Type **ltrx_read_xml_config** next to **Method name**.
4. Next to **Method payload**, copy and paste the below XML and then click **Call Method**.
   `{"param":"Access Point;Line"}`
5. Verify that the Access Point and Line configuration have been returned successfully.



ltrx_trusted_import_xml_config

Description: Status Action

1. Open Device Explorer and click the **Call Method on Device** tab.
2. Select your device next to **Device ID**.
3. Type **ltrx_trusted_import_xml_config** next to **Method name**.
4. Next to **Method payload**, copy and paste the below XML and then click **Call Method**.

```
{"param":"
<configrecord version = \"0.1.0.1\">
   <configgroup name = \"xml import control\">
      <configitem name = \"action\">
         <value name = \"group\">interface</value>
         <value name = \"optional group
instance\">wlan0</value>
         <value name = \"optional item\"></value>
         <value name = \"optional item instance\"></value>
         <value name = \"name\">renew</value>
      </configitem>
   </configgroup>
</configrecord>
"}
```

5. Verify that the DHCP IP address has been renewed successfully.