ICONICS

Make the Invisible Visible™

Microsoft Azure



Monitoring Building Power Consumption

An ICONICS Whitepaper

January 2021

Contents

1 Introduction 4
2 Architecture 4
3 Power Meter data 5
4 Azure infrastructure 6
4.1 Azure Resource Group 6
4.2 Azure IoT Hub 7
4.3 SQL Server 8
5 Getting power meter data 11
5.1 Installing ICONICS IoTWorX 11
5.2 Configuring Modbus communications to the power meters 11
5.3 Identifying data points on the power meter 14
5.4 Specifying objects to push to Azure 15
5.5 Testing device connectivity locally with Data Explorer 16
6 Publishing and storing data 17
6.1 Publishing from IoTWorX to Azure IoT Hub 17
6.2 Testing data reception at IoT Hub with Azure IoT explorer 18
6.3 Pushing from IoT Hub to SQL Server with Azure Stream Analytics 20
6.4 Pulling from IoT Hub into ICONICS GENESIS64 22
7 Displaying power consumption trends and ratios 23
7.1 Designing a report in Power BI Desktop 23
7.2 Publishing to PowerBI.com 24
7.3 Embedding a Power BI report in a website 25
8 Available partner solutions 29
8.1 ICONICS GENESIS64 ™ 29
9 Insights 30
9.1 Load profiling 30
9.2 Phase balancing 31

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

Copyright and Confidentiality

By accessing and using the installation instructions (the “instructions”) you acknowledge and agree, on your behalf and on behalf of the person, entity or other organization on whose behalf you are accessing the instructions, that neither Microsoft or ICONICS, nor any of its service providers, including, without limitation, any system integrator or independent software vendor: (1) makes any representations or warranties of any kind, either express, implied, statuary or otherwise with respect to the instructions, including the accuracy, completeness or usefulness thereof; and (2) shall be liable for damages of any kind, under any legal theory, arising out of or in connection with your election to follow or use, or inability to follow or use, the instructions, including any direct, indirect, incidental, special, punitive or consequential damages, or for loss of use, loss of profits, loss of data, loss of business, or loss of privacy or security, even if foreseeable, arising out of or in connection with your election to follow or use, or inability to follow or use, the instructions. You further acknowledge and agree that your use of the instructions, whether directly or indirectly, is at your own risk and that you expressly assume all risk in connection with your use of the instructions. If you do not agree to the foregoing, you may not access or use the instructions.

Copyright © 2021, Microsoft Corporation and ICONICS, Inc. All rights reserved.

Authors

- Spyros Sakellariadis, Principal Program Manager, Industry Innovation, Microsoft Corporation
- Maksym Mushkin, Industry Innovation, Microsoft Corporation
- Zhi Wei Li, Director of Innovation and Incubation Solutions, ICONICS
- Tearle Whitson, Director, Critical Systems and Infrastructure, CBRE
- Sean Kim, ESB Chief Engineer, CBRE
- Francis Yuen, System Integrator, CBRE
- Roger Killam, CISA, Senior Systems Engineer, CBRE

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

1 Introduction

Monitoring power consumption of buildings or equipment is an important component in understanding and therefore reducing operational costs. It has other benefits as well, for example reducing maintenance costs and increasing sustainability. Often power monitoring is a functionality that is included in a larger software solution from the local utility company or manufacturer of the power meters used, but it can be useful to view and analyze power consumption data outside of these domain-specific applications. In this article we describe an example of a simple data collection and visualization workflow, ultimately storing the data in a cloud database that can be queried by any 3rd party application. In essence this workflow demonstrates one way of monitoring power consumption. The workflow is not a production-ready enterprise solution; however, the elements described in this article could certainly be the basis for a larger implementation.

If the reader plans to replicate a similar installation to that described here, they would need access to a power meter and have purchased Microsoft Azure and Power BI subscriptions, as well as licensed any 3rd party products used in the example described.

2 Architecture

This article documents a simple infrastructure to monitor the electrical consumption of a building by connecting to the power meter installed on the building by the local electrical utility. The infrastructure comprises several distinct pieces:

1. An on-premises computer running software to communicate with the power meter and push the data to the Microsoft Azure cloud
2. A cloud infrastructure with a number of cloud services to ingest, parse, and store the incoming data
3. A number of user interfaces to display the data

The infrastructure described in this article uses:

1. ICONICS IoTWorX™ for the on-premises gateway software
2. Azure IoT Hub, Azure Stream Analytics and SQL Server for the services to ingest, parse, and store the incoming data
3. Power BI to display the data

In a later section, we also discuss using ICONICS GENESIS64™ as a cloud solution which can be incorporated into the infrastructure for real-time monitoring.

A high-level architecture showing these components is shown below:

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

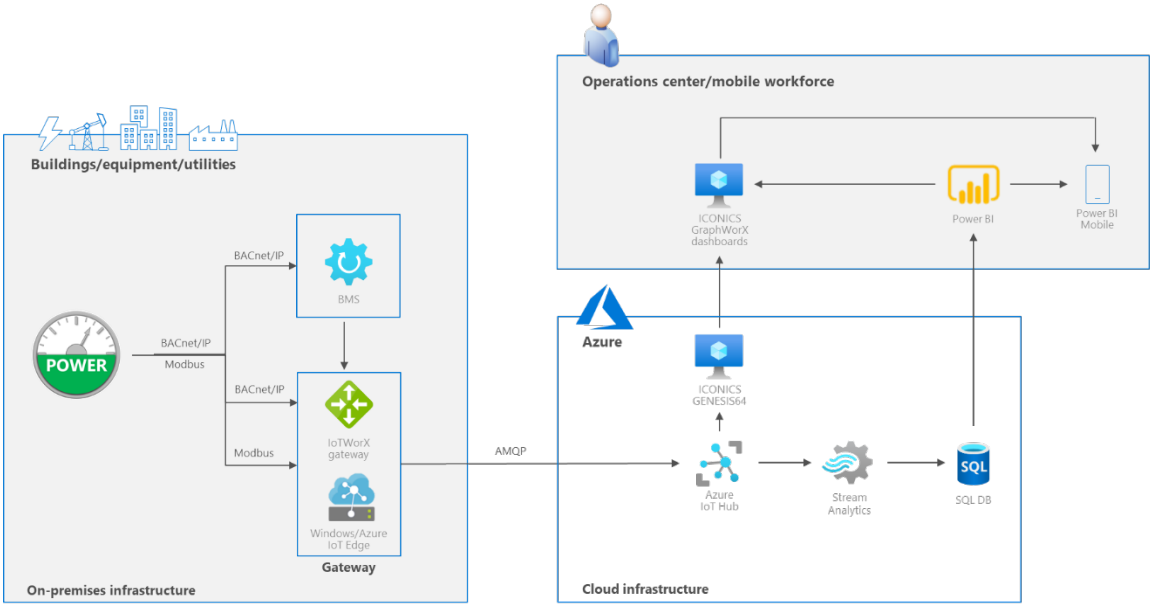


Figure 1: Architecture

We have documented elsewhere suggested step-by-step guidance for installing IoTWorX, so there is no need to repeat that information in this article. We will refer to those instructions where appropriate, which are published at the following locations:

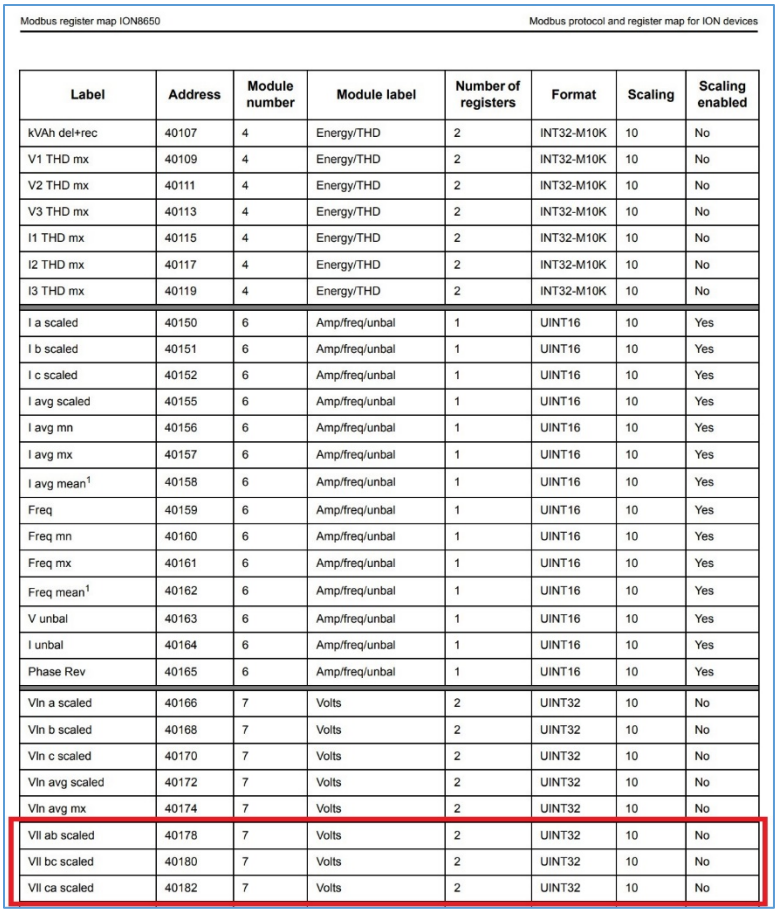
- Using IoTWorX as a Gateway in Microsoft Azure Deployments, and will be referred to in this article as *Using IoTWorX*
- Installing IoTWorX on IoT Edge, referred to as *IoTWorX on IoT Edge*
- Connecting a Delta VFD to Microsoft Azure using ICONICS IoTWorX, referred to as *Connecting a VFD*

3 Power Meter data

Power meters vary in size and functionality and are available from multiple manufacturers. For this article we chose to monitor the power data from the power meters installed in Microsoft's Buildings 121 and 122 on its Redmond, Washington campus. They are PowerLogic™ ION8650 meters from Schneider Electric, installed by the local utility provider Puget Sound Energy. These large commercial power meters have many capabilities, but for the purposes of this article we will focus only on communicating with the power meters to read the current values for the power consumed by the building. Note that the Building 121 and 122 power meters had Ethernet cable run to them at the time of installation. Depending upon the location this can present a high hurdle to integration in some instances.

The power meters expose the data about the power consumption using the Modbus™ protocol. Individual power readings are published in what are called Modbus registers, which can be read by any application that supports Modbus. The protocol document published on the Schneider Electric download page summarizes the default Modbus protocol and the register maps. The important section for the purposes of monitoring the power consumption is shown below:

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.



Modbus register map ION8650 — Modbus protocol and register map for ION devices

Label	Address	Module number	Module label	Number of registers	Format	Scaling	Scaling enabled
kVAh del+rec	40107	4	Energy/THD	2	INT32-M10K	10	No
V1 THD mx	40109	4	Energy/THD	2	INT32-M10K	10	No
V2 THD mx	40111	4	Energy/THD	2	INT32-M10K	10	No
V3 THD mx	40113	4	Energy/THD	2	INT32-M10K	10	No
I1 THD mx	40115	4	Energy/THD	2	INT32-M10K	10	No
I2 THD mx	40117	4	Energy/THD	2	INT32-M10K	10	No
I3 THD mx	40119	4	Energy/THD	2	INT32-M10K	10	No
I a scaled	40150	6	Amp/freq/unbal	1	UINT16	10	Yes
I b scaled	40151	6	Amp/freq/unbal	1	UINT16	10	Yes
I c scaled	40152	6	Amp/freq/unbal	1	UINT16	10	Yes
I avg scaled	40155	6	Amp/freq/unbal	1	UINT16	10	Yes
I avg mn	40156	6	Amp/freq/unbal	1	UINT16	10	Yes
I avg mx	40157	6	Amp/freq/unbal	1	UINT16	10	Yes
I avg mean[1]	40158	6	Amp/freq/unbal	1	UINT16	10	Yes
Freq	40159	6	Amp/freq/unbal	1	UINT16	10	Yes
Freq mn	40160	6	Amp/freq/unbal	1	UINT16	10	Yes
Freq mx	40161	6	Amp/freq/unbal	1	UINT16	10	Yes
Freq mean[1]	40162	6	Amp/freq/unbal	1	UINT16	10	Yes
V unbal	40163	6	Amp/freq/unbal	1	UINT16	10	Yes
I unbal	40164	6	Amp/freq/unbal	1	UINT16	10	Yes
Phase Rev	40165	6	Amp/freq/unbal	1	UINT16	10	Yes
Vln a scaled	40166	7	Volts	2	UINT32	10	No
Vln b scaled	40168	7	Volts	2	UINT32	10	No
Vln c scaled	40170	7	Volts	2	UINT32	10	No
Vln avg scaled	40172	7	Volts	2	UINT32	10	No
Vln avg mx	40174	7	Volts	2	UINT32	10	No
Vll ab scaled	40178	7	Volts	2	UINT32	10	No
Vll bc scaled	40180	7	Volts	2	UINT32	10	No
Vll ca scaled	40182	7	Volts	2	UINT32	10	No

Figure 2: Modbus registers

For example, we can read the voltage difference between lines A and B at address 40178.

4 Azure infrastructure

4.1 Azure Resource Group

This article assumes the reader has basic knowledge of Microsoft's cloud products and services and understands how to create and configure resources. Consequently, only descriptions or diagrams of the final configuration will be included, not step-by-step instructions.

The example described here uses six Azure services, deployed in a single resource group shown below. We called the resource group **PowerMeterMonitoring** when creating this configuration:

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

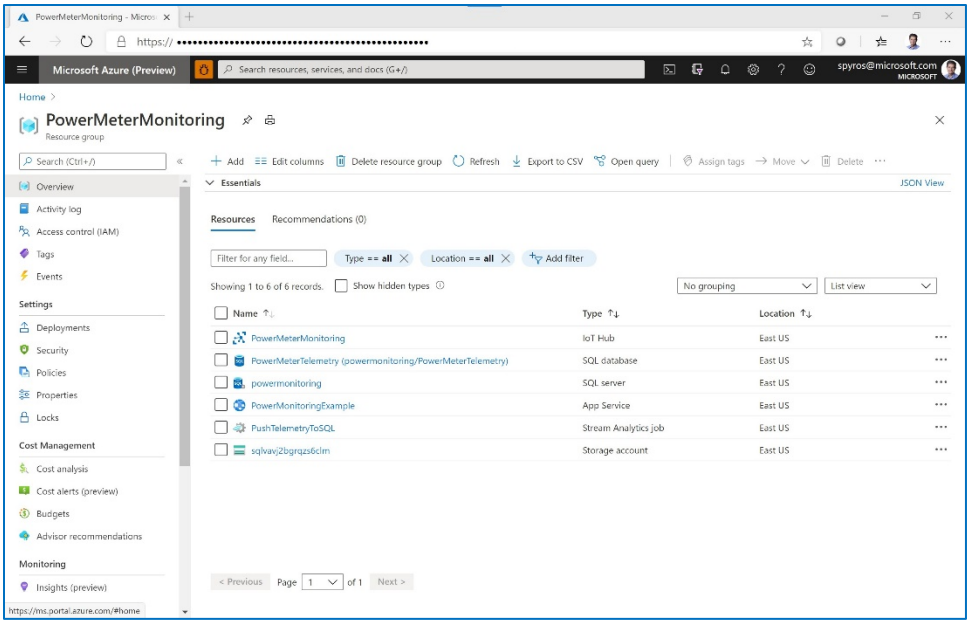


Figure 3: Resource Group with Azure services used for monitoring power meters

Briefly, the function of each of these services is as follows:

Resource	Type	Function
PowerMeterMonitoring	Azure IoT Hub	Receive data from IoTWorX gateway
PowerMeterTelemetry	SQL database	Store data received from IoTWorX gateway
powermonitoring	SQL server	Holds SQL database
PowerMonitoringExample	App Service	Website for displaying power consumption
PushTelemetryToSQL	Stream Analytics job	Push data from IoT Hub to SQL
sqlvavj2bgrqzs6clm	Storage account	Used by resources to house configurations

Figure 4: Azure services used

4.2 Azure IoT Hub

Prior to installing the on-premises components necessary to read the power meter data, we created an Azure IoT Hub to receive the data. See *Using IoTWorX,* Section 2, “Setting up IoT Hub”, for step-by-step instructions for creating an IoT Hub instance.

In the power meter example, we used the following values when creating this configuration:

Azure Service	Value
Azure IoT Hub	PowerMeterMonitoring
IoT Device	IoTWorX

Figure 5: IoT Hub values

The completed deployment is shown here:

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

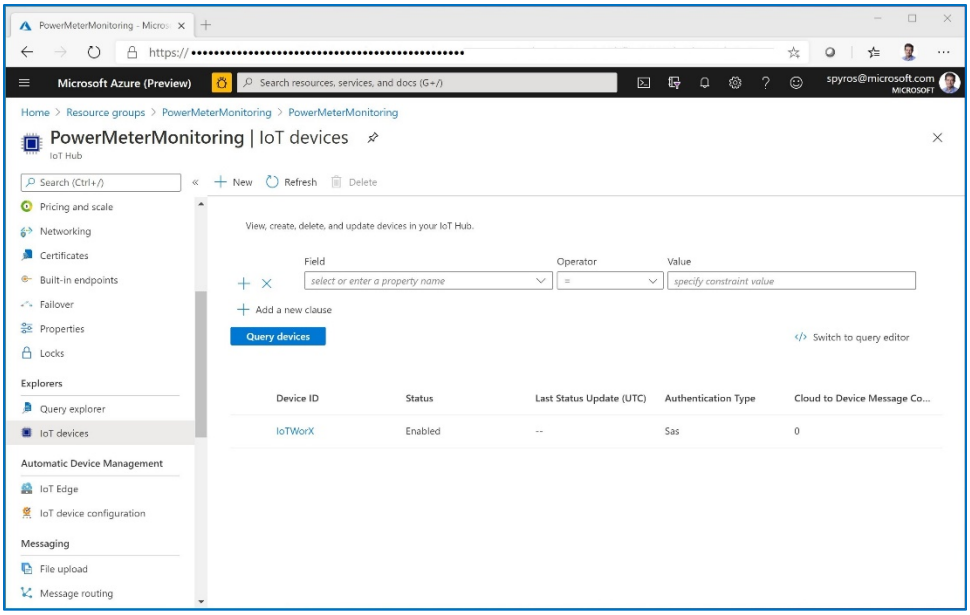


Figure 6: IoT Hub

The ICONICS IoTWorX software on the on-premises gateway will be configured to send power meter data to the **IoTWorX** device ID in the IoT Hub.

4.3 SQL Server

Prior to installing the on-premises components, we also created a SQL database and tables to store the data. The completed deployment is shown here:

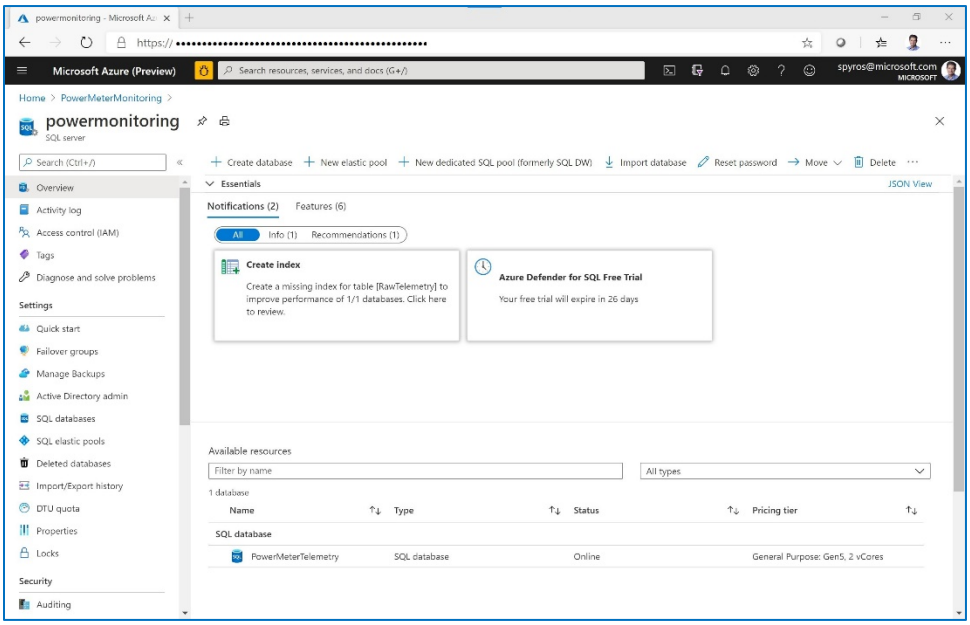


Figure 7: SQL Server overview

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

In this example, we used the following values when creating this configuration:

Azure Service	Value
SQL Server	PowerMonitoring
SQL Database	PowerMeterTelemetry
SQL Table	RawTelemetry
SQL View	v_PowerMeter_Pivot

Figure 8: SQL Server values

The **RawTelemetry** table will hold the data as it comes from the power meter, storing each measurement in a separate record. The **v_PowerMeter_Pivot** will display the same data, but all the measurements at a specific time will be stored in a single record, with an additional column that determines the imbalance rating of the Line-to-Line voltage. The table in the SQL database has the following structures:

```
CREATE TABLE [dbo].[RawTelemetry](
	[Building] [varchar](50) NOT NULL,
	[Parameter] [varchar](50) NOT NULL,
	[Value] [float] NULL,
	[TimeStamp] [datetime] NULL
) ON [PRIMARY]
GO
```

Figure 9: RawTelemetry table schema

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

Finally, the view definition is as follows:

```
CREATE VIEW [dbo].[v_PowerMeter_Pivot] AS
SELECT * ,
       CASE
           WHEN Imbalance_Max_Pct < 0.01 THEN 'Good'
           WHEN Imbalance_Max_Pct >= 0.01
                AND Imbalance_Max_Pct < 0.04 THEN 'Alert'
           ELSE 'Urgent'
       END AS Imbalance_Rating
FROM
  (SELECT PVTSI.* ,
          CASE
              WHEN Imbalance_AB >= Imbalance_AC
                   AND Imbalance_AB >= Imbalance_BC THEN ROUND(Imbalance_AB / VLL_AVG, 4)
              WHEN Imbalance_AC >= Imbalance_AB
                   AND Imbalance_AC >= Imbalance_BC THEN ROUND(Imbalance_AC / VLL_AVG, 4)
              ELSE ROUND(Imbalance_BC / VLL_AVG, 4)
          END AS Imbalance_Max_Pct
   FROM
     (SELECT PVTS.* ,
             ABS(ROUND(VLL_AB-VLL_AVG, 2)) AS 'Imbalance_AB' ,
             ABS(ROUND(VLL_AC - VLL_AVG, 2)) AS 'Imbalance_AC' ,
             ABS(ROUND(VLL_BC - VLL_AVG, 2)) AS 'Imbalance_BC'
      FROM
        (SELECT CONVERT(date, TIMESTAMP) AS Date --      , CONVERT(time, TimeStamp) AS
'Time UTC'
 ,
                Building,
                TimeStamp,
                kW_total AS kW,
                           Voltage_AB AS VLL_AB,
                Voltage_CA AS VLL_AC,
                Voltage_BC AS VLL_BC,
                ROUND((Voltage_AB + Voltage_BC + Voltage_CA)/3, 2) AS VLL_AVG
         FROM
           (SELECT t1.*
            FROM RawTelemetry t1) AS SourceTable PIVOT(MAX(Value)
                                                       FOR PARAMETER IN(Voltage_CA,
Voltage_BC, Voltage_AB, kW_total)) AS PivotTable) AS PVTS
      WHERE VLL_AC IS NOT NULL
        AND VLL_BC IS NOT NULL
        AND VLL_AB IS NOT NULL) AS PVTSI) AS PVTSIP
GO
```

Figure 10: v_PowerMeter_Pivot view

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

5 Getting power meter data

5.1 Installing ICONICS IoTWorX

See *Using IoTWorX,* Section 1, “Installing GENESIS64 with custom options”, for step-by-step instructions for installing IoTWorX.

5.2 Configuring Modbus communications to the power meters

To communicate with the power meters using IoTWorX, we defined two Modbus channels in the Modbus configuration branch in Workbench. One channel was defined for the power meter attached to Building 121, and another for the power meter attached to Building 122. The channels identify the IP addresses (10.135.x.x) and Modbus TCP Port (502) for the power meters:

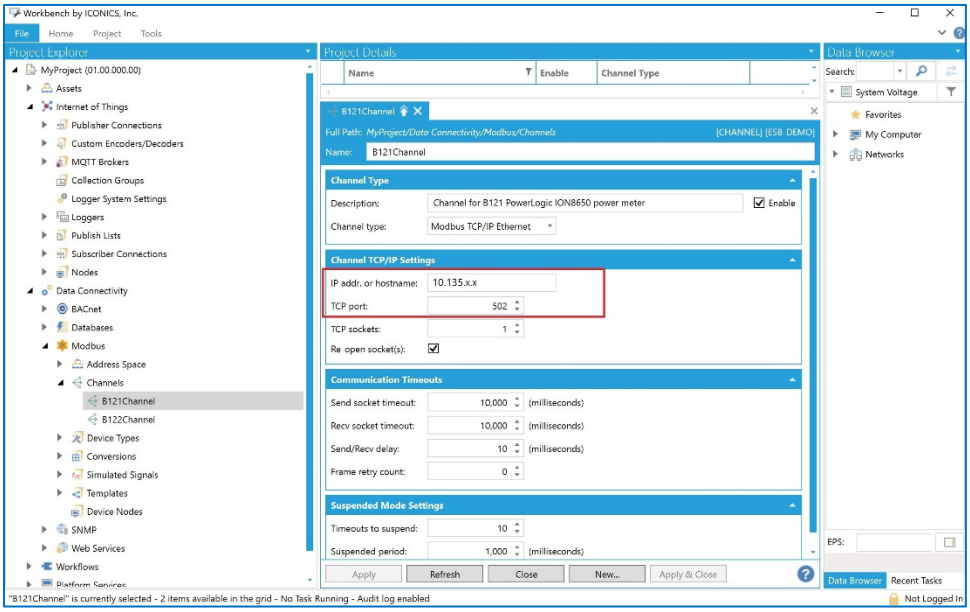


Figure 11: Modbus channel configuration

In addition to creating the Modbus channels, we need to specify the method used to convert the raw data when it is received by IoTWorX into something meaningful. Since each type of data requires its own conversion method, we created a conversion definition for kW and another for Line-to-Line voltage difference.

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

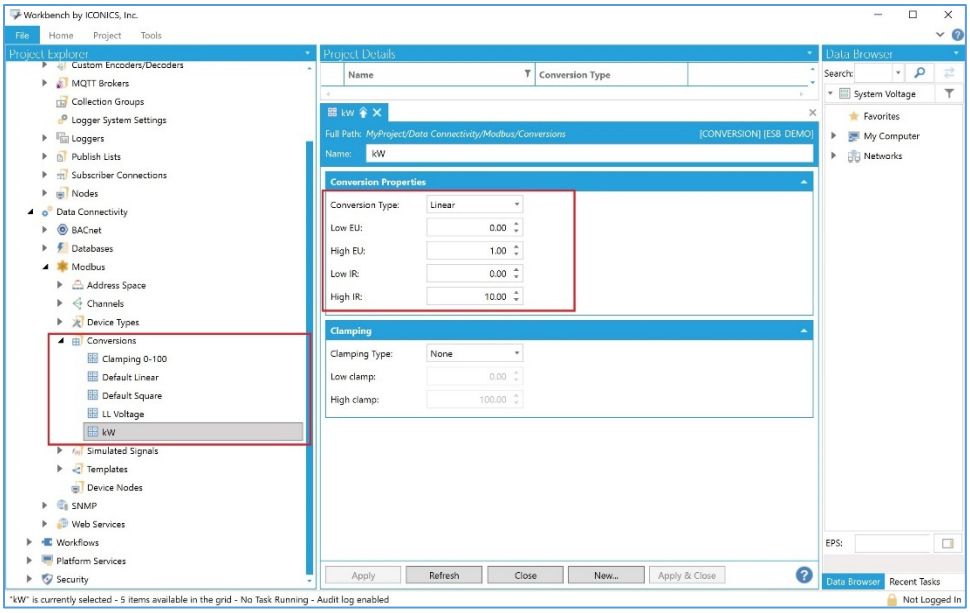


Figure 12: Modbus Conversion - kW

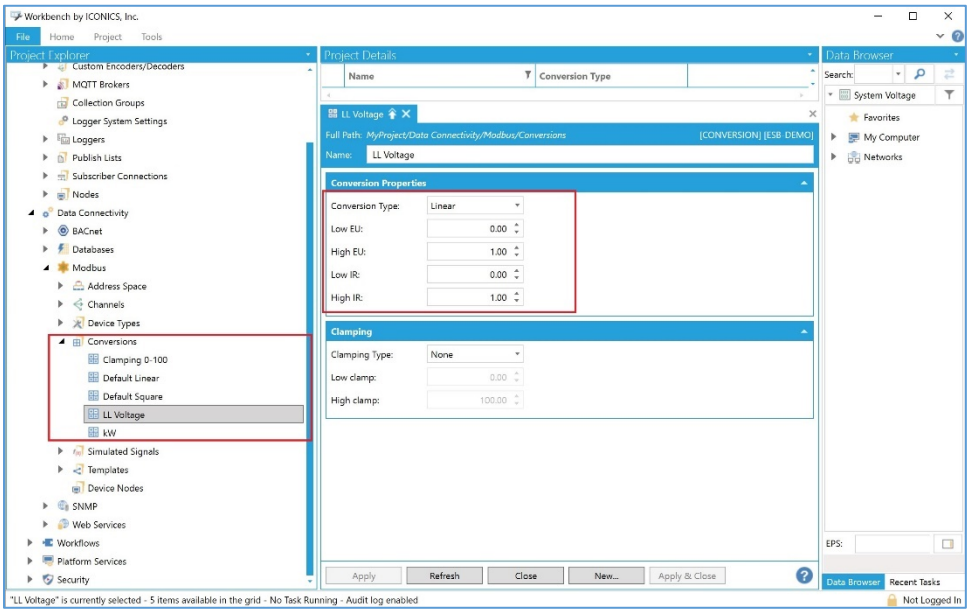


Figure 13: Modbus Conversion - LL Voltage

Note that the conversion for kW and Line-to-Line voltage differ in the Hi IR, the Instrument Range. The protocol document, referenced in Figure 2: Modbus registers, shows scaling enabled, with a scaling factor of 10 for kW (p.44), but scaling disabled for VLL (p.46):

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

Label	Address	Module Number	Module label	Number of registers	Format	Scaling	Scaling enabled
kW tot scaled	40033	2	Power/PF	2	INT32	10	Yes
VLL AB scaled	40178	7	Volts	2	UINT32	10	No
VLL BC scaled	40180	7	Volts	2	UINT32	10	No
VLL CA scaled	40182	7	Volts	2	UINT32	10	No

Figure 14: Protocol document scaling values

To comply with this, we set the Hi IR for the kW Conversion to 10.0.

Finally, we need to define a Device Type for the ION8650 power meter that specifies how the data is represented in the power meter register. Note specifically the **Swap words inside dwords** setting, without which the data received will be unintelligible.

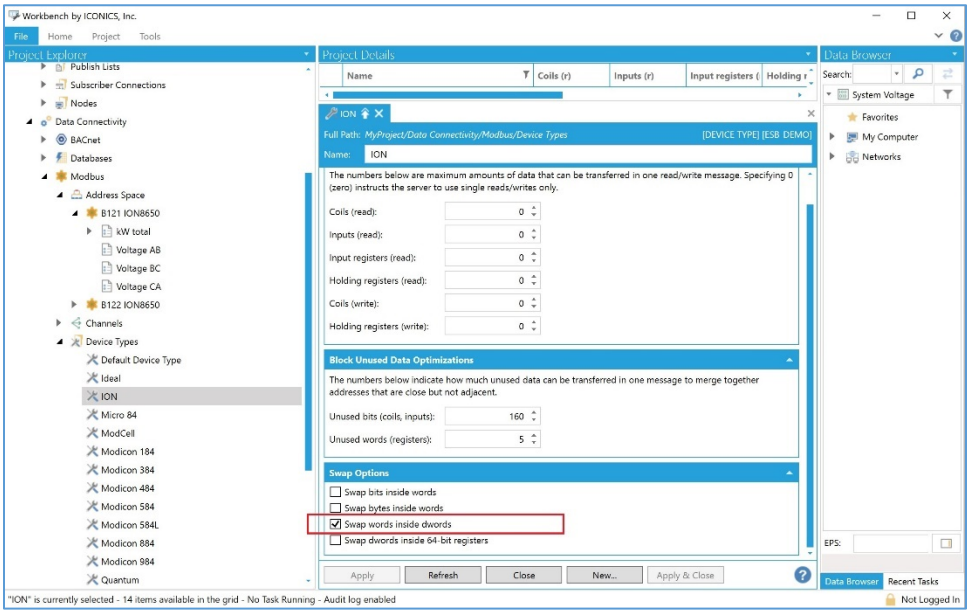


Figure 15: Modbus Device Type

On page 10 of the protocol documentation, under “32-bit integer format”, it states that “In signed and unsigned 32-bit integer formats, each ION input register to the module corresponds to two 16-bit Modbus holding register outputs.” Typically the first Modbus register of the pair represents the high order word of the 32-bit value, while the second Modbus register represents the low order word. However, IoTWorX treats two consecutive words oppositely to the way the device does, so the swap is necessary.

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

5.3 Identifying data points on the power meter

Having specified how to communicate with the power meter, we need to specify the individual power meters and the registers we wish to poll. To do this, first we create an object that represents the power meter:

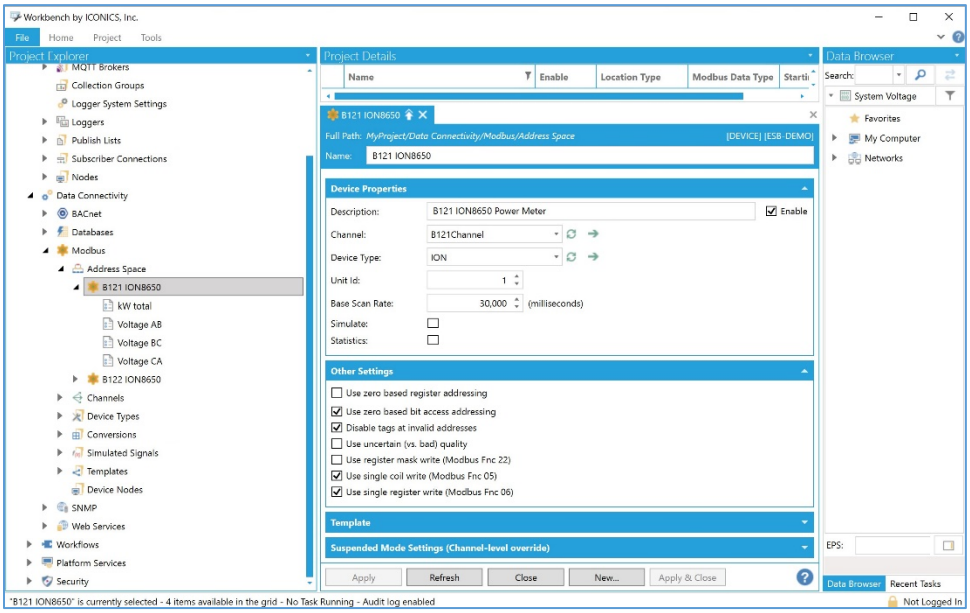


Figure 16: B121 power meter

Then, under that power meter, we create objects for each of the registers to be polled. We get the starting address and Modbus data type from the Power Meter protocol document shown in Figure 2: Modbus registers, above:

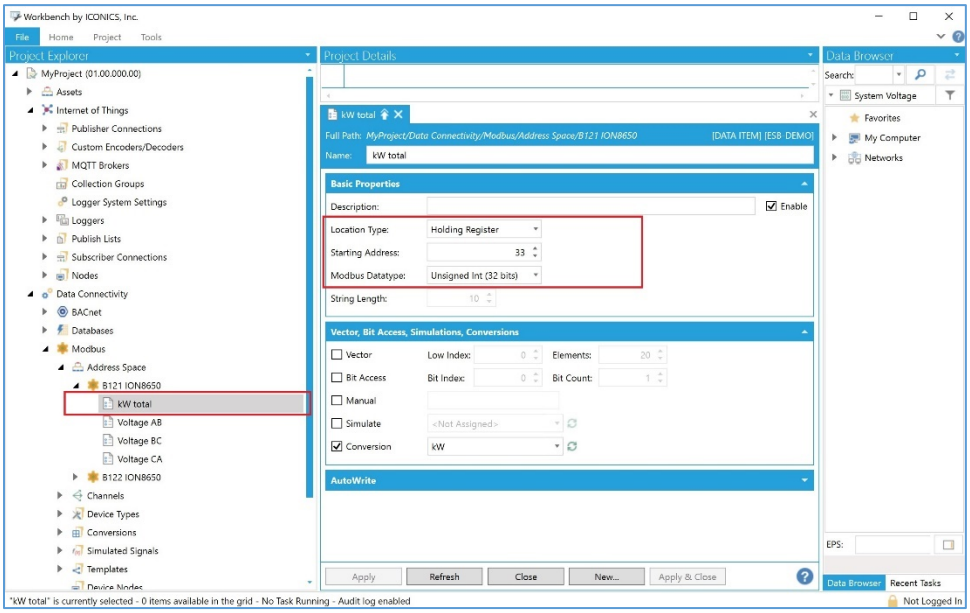


Figure 17: Power meter kW register

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

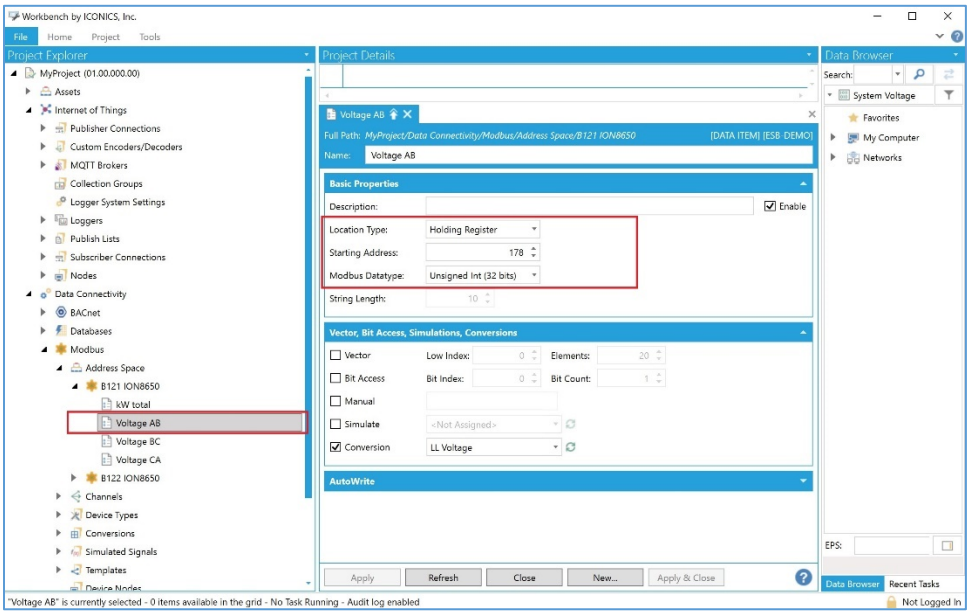


Figure 18: Power meter VLL AB register

5.4 Specifying objects to push to Azure

At this point, IoTWorX is configured to get data from the power meter registers. We have the option to publish some or all of them to Azure, and to do this we create a Publish List in IoTWorX. We called the Publish List **PowerMeter PublishList** when creating this example:

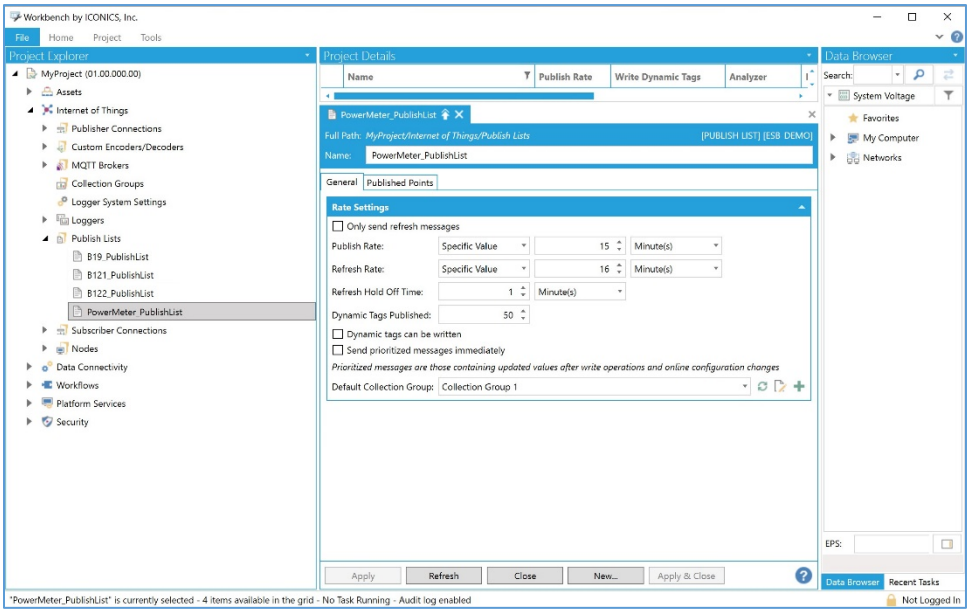


Figure 19: IoTWorX Publish List

Having created the Publish List object, we need to populate it with the subset of points that we collected and now want to publish. See *Using IoTWorX,* Section 3.3, “Create a publish list”, for step-by-step

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

instructions for adding points to a Publish List. The completed list with the Modbus power meter points is shown here:

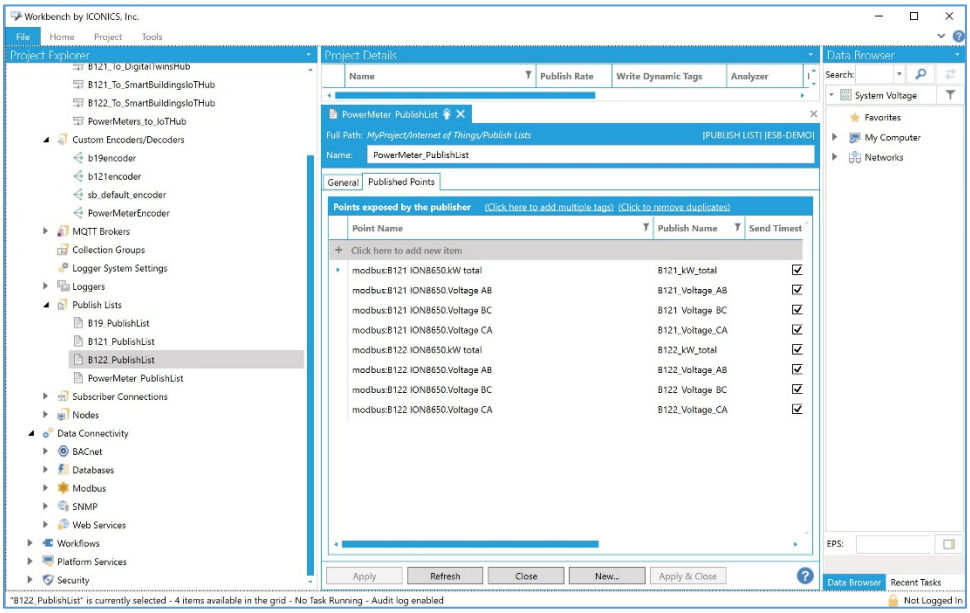


Figure 20: IoTWorX Published Points

5.5 Testing device connectivity locally with Data Explorer

To visualize the data being sent by IoTWorX, launch the Data Explorer application in the ICONICS Tools folder in the computer's Start Menu. Navigate to **My Computer → Data Connectivity → Modbus** and double click on the power meter defined. You should see the data values similar to those shown in Figure 12: Modbus Conversion - kW and Figure 13: Modbus Conversion - LL Voltage:

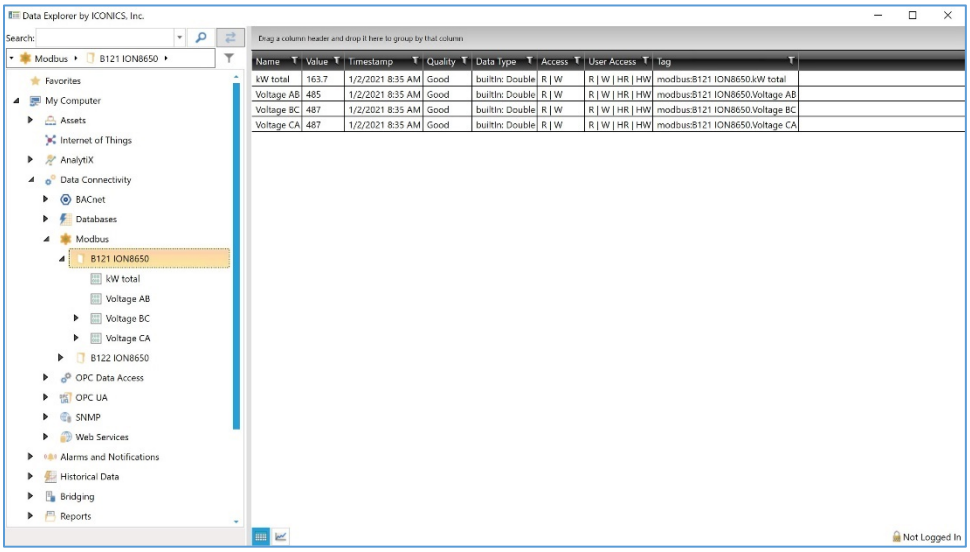


Figure 21: Viewing data collected by IoTWorX using ICONICS Data Explorer on the gateway machine

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

6 Publishing and storing data

6.1 Publishing from IoTWorX to Azure IoT Hub

See *Using IoTWorX,* Section 3.4 "Create a custom encoder", for step-by-step instructions for creating an encoder for the messages sent by IoTWorX, and Section 3.5, "Create a publisher connection", for steps to identify the Azure target for the date. In the example used for this article, the encoder is shown below:

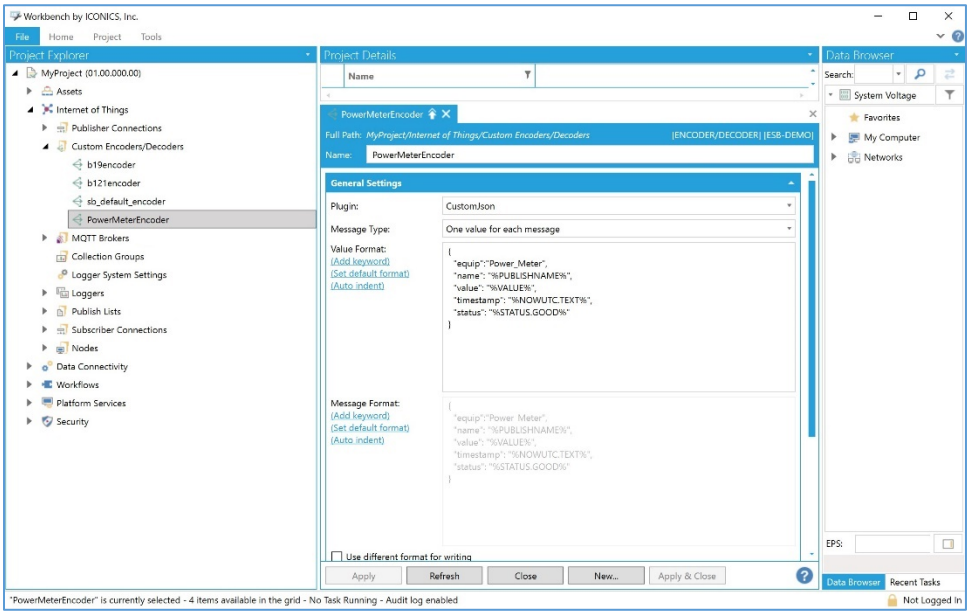


Figure 22: IoTWorX Custom Encoder

The Publisher Connection is shown below, using the IoT Connection String for the **IoTWorX** device as shown in Figure 6: IoT Hub above:

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

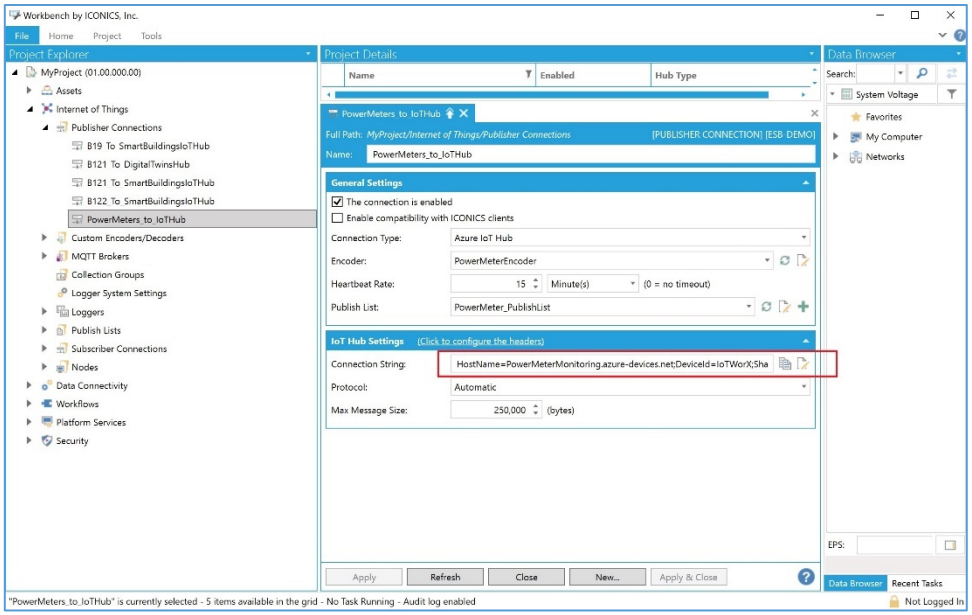


Figure 23: IoTWorX Publisher Connection

6.2 Testing data reception at IoT Hub with Azure IoT explorer

See Install and use Azure IoT explorer for step-by-step instructions for using the Azure IoT explorer tool to monitor incoming data. Upon launching Azure IoT Explorer, enter the IoT Hub connection string for the IoT Hub used. In the example used for this article, the configuration is shown below:

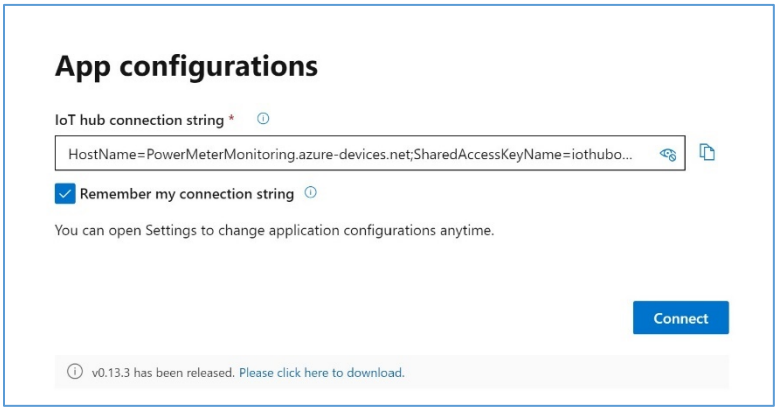


Figure 24: Azure IoT explorer configuration

If IoTWorX and IoT Hub are configured as described in this article, after navigating to PowerMeterMonitoring → Devices → IoTWorX → Telemetry and clicking Start, data should be seen in the main window:

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

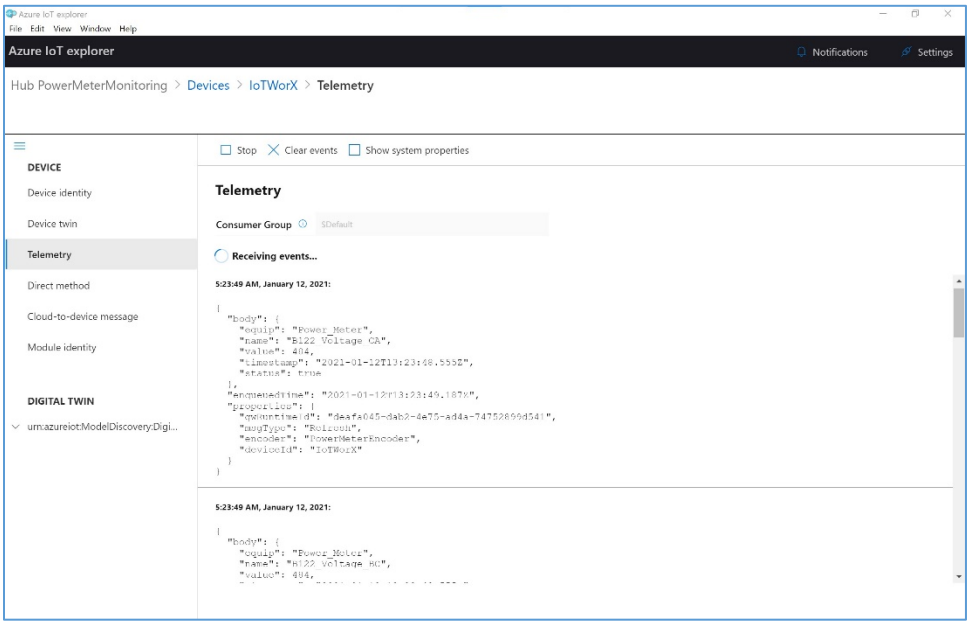


Figure 25: Viewing data received by IoT Hub

Note that the format of the data corresponds to the IoTWorX Custom Encoder shown in Section 6.1 above.

IoTWorX Encoder	Value shown above
Equip	Power_Meter
Name	B122 Voltage CA
Value	482.0
TimeStamp	2021-01-02T14:26:35.085Z
Status	True

Figure 26: Data received

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

6.3 Pushing from IoT Hub to SQL Server with Azure Stream Analytics

To write the streaming data to SQL Server, we created and ran an Azure Stream Analytics job in the Azure portal which we named PushTelemetryToSQL:

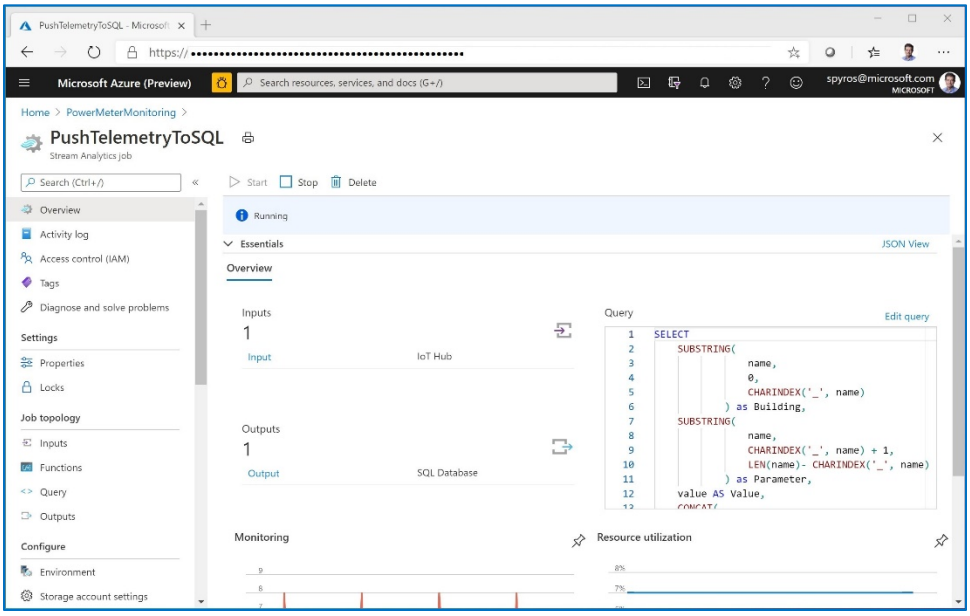


Figure 27: Azure Stream Analytics job

A Stream Analytics job contains three parts – an input, an output, and a query. For the input, we specified the IoT Hub shown in Figure 6: IoT Hub. For the output, we specified the SQL Server table shown in Figure 7: SQL Server overview. Then, for the query we entered the following:

```
SELECT
    SUBSTRING(
                name,
                0,
                CHARINDEX('_', name)
            ) as Building,
    SUBSTRING(
                name,
                CHARINDEX('_', name) + 1,
                LEN(name)- CHARINDEX('_', name)
            ) as Parameter,
    value AS Value,
    CONCAT(
        cast(DATEPART(year,DATEADD(hour,-8,timestamp)) as nvarchar(max)),'/',
        cast(DATEPART(month,DATEADD(hour,-8,timestamp)) as nvarchar(max)),'/',
        cast(DATEPART(day,DATEADD(hour,-8,timestamp)) as nvarchar(max)),' ',
        cast(DATEPART(hour,DATEADD(hour,-8,timestamp)) as nvarchar(max)),':',
        cast(DATEPART(minute,DATEADD(hour,-8,timestamp)) as nvarchar(max)),':',
        cast(DATEPART(second,DATEADD(hour,-8,timestamp)) as nvarchar(max))
    ) as TimeStamp
INTO
    Output
FROM
    Input
```

Figure 28: Azure Stream Analytics Query

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

The first two SUBSTRING clauses parse the name (e.g. B121_Voltage_AB) into two fields, Building (e.g. B121) and Parameter (e.g. Voltage_AB). The CONCAT clause reformats the incoming UTC TimeStamp field into a SQL date format and converts into the PST time zone by subtracting 8 hours. Of course, the query depicted in Figure 28 is only applicable when the US observes Pacific Standard Time (PST) since PST is 8 hours behind Coordinated Universal Time (UTC).

The Azure Stream Analytics job continuously writes the incoming data to the **RawTelemetry** file in the SQL Server database. We can see that it has been parsed and received correctly using a simple query:

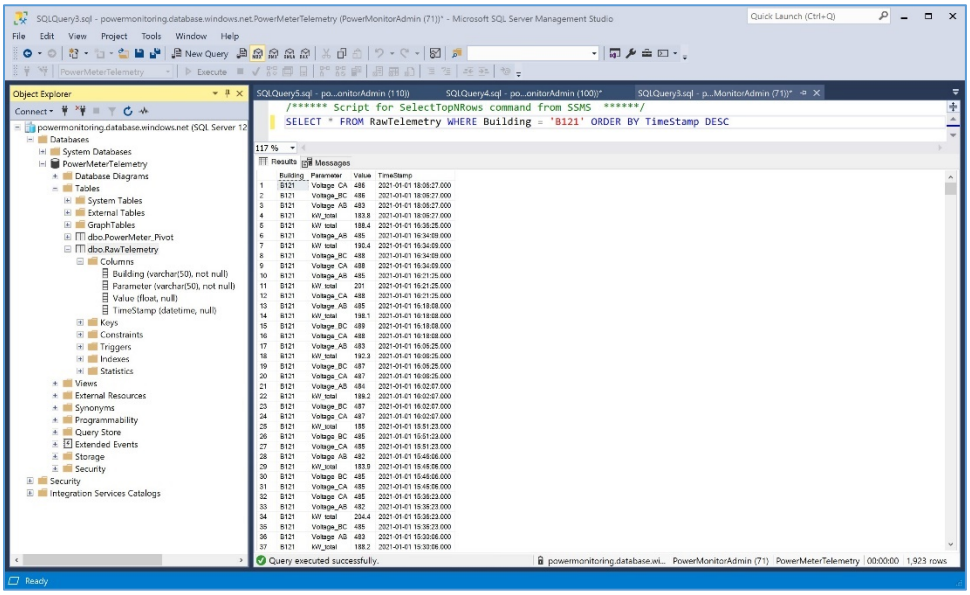


Figure 29: SQL Server RawTelemetry listing

Similarly, we can see the data in the pivot view:

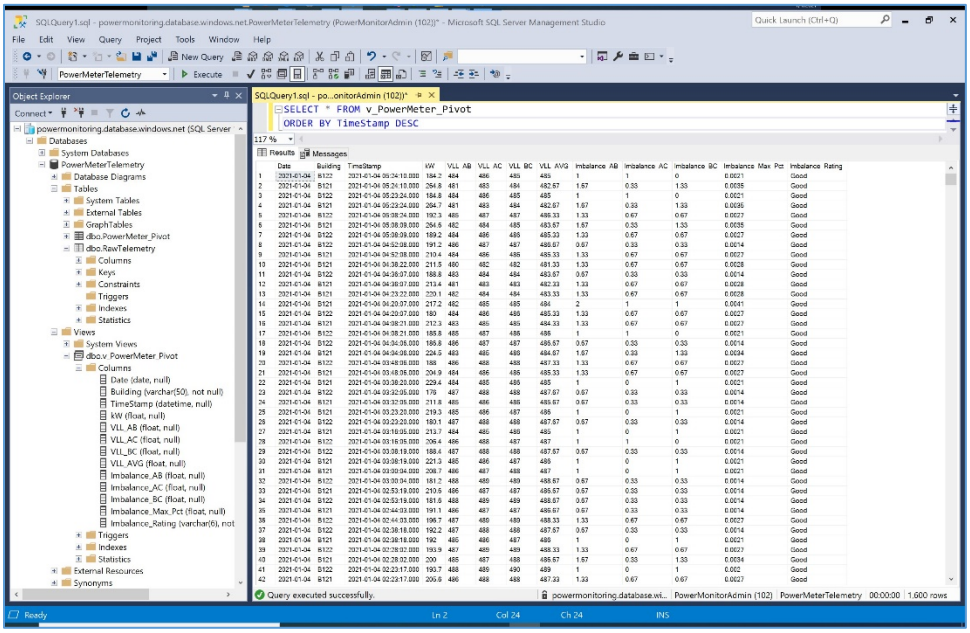


Figure 30: SQL view listing

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

6.4 Pulling from IoT Hub into ICONICS GENESIS64

In the previous section we described a workflow using Azure Stream Analytics and SQL Server. Since the data is being received in IoT Hub, we can also connect to it with 3rd party clients. In our example we created a second workflow using ICONICS GENESIS64. For step-by-step instructions on installing and configuring GENESIS64, see *Connecting a VFD,* Section 6, "Using GENESIS64 as a no code client". Differences from those instructions are outlined below.

- For the custom encoder, described in *Connecting a VFD,* Section 6.1, create one exactly the same as shown in Figure 22: IoTWorX Custom Encoder, above. Do not check the box labelled "Use different format for writing."
- For the subscriber connection, described in *Connecting a VFD,* Section 6.2, use the IoT Hub owner connection string and Event-Hub compatible connection string for IoT Hub **PowerMeterMonitoring**.

To visualize the data being ingested by the subscriber service, bring up the ICONICS Data Explorer from the Start Menu in the ICONICS VM. Browse under the Internet of Things folder to the *Published Data via Static List* folder:

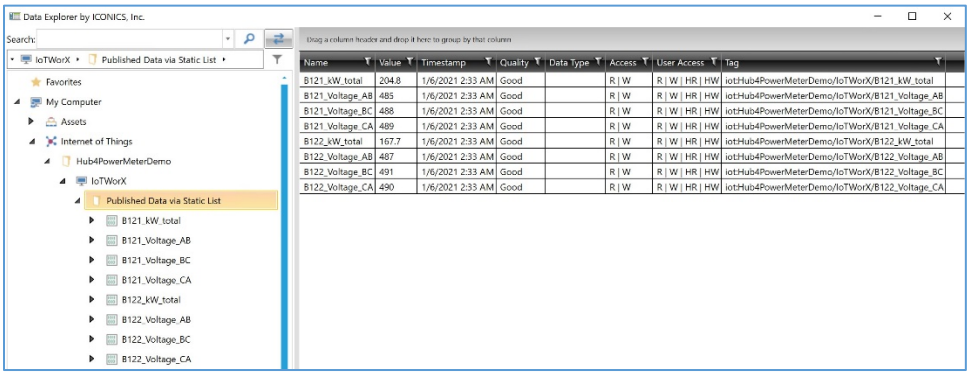


Figure 31: Viewing data received by GENESIS64 using Data Explorer in the ICONICS Azure VM

Data values correspond to those seen in the Azure IoT explorer, shown in Figure 21: Viewing data collected by IoTWorX using ICONICS Data Explorer on the gateway machine.

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

7 Displaying power consumption trends and ratios

7.1 Designing a report in Power BI Desktop

Figure 32: Power BI Desktop shows a Power BI report pulling the data from the SQL table:

u

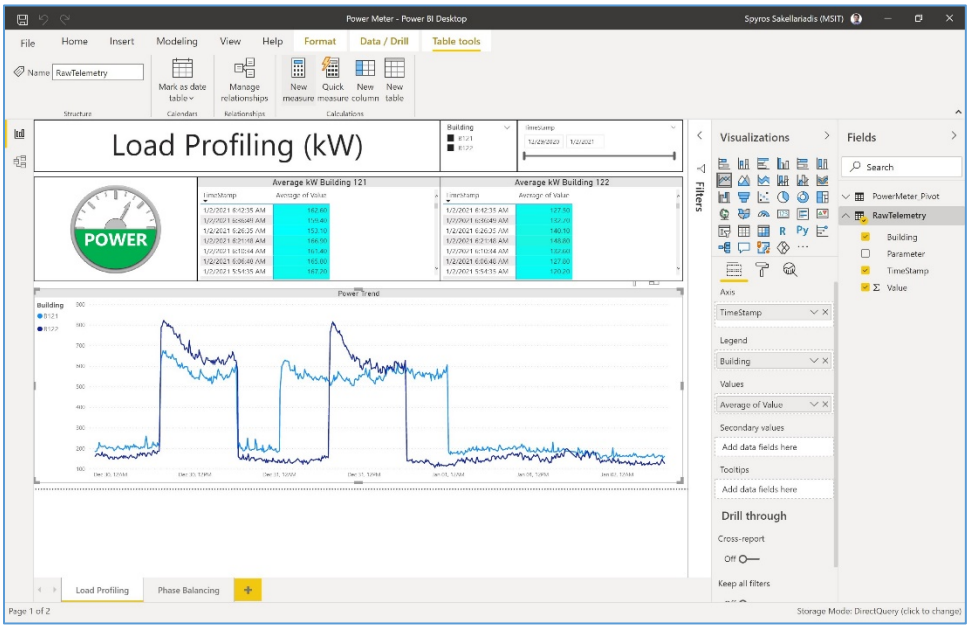


Figure 32: Power BI Desktop

To create this dashboard use the [Power BI desktop application] and take the following steps:

1. Create a new report
2. From the **Home** page select **SQL Server**
3. Enter **powermonitoring.database.windows.net** for the SQL Server, **PowerMeterTelemetry** for the SQL database, and select **Direct Query**
4. Specify the chart visuals. For the Load Balancing tab use the **dbo.RawTelemetry** file and add:
    a. A text box for the title
    b. Two Slicer visuals for the Building and TimeStamp filters
    c. An image for the power meter JPG
    d. Two Matrix visuals for the grids
    e. A Line Chart for the trends
5. From the **View** page, select **Mobile Layout** to add visuals for a mobile phone display
6. Add a second page
7. From the **Home** page select **SQL Server**, specify the SQL database, select **Direct Query,** and click **Advanced Options**
8. Add the query **SELECT * FROM dbo.v_PowerMeter_Pivot**
9. For the Phase Balancing report use the **dbo.v_PowerMeter_Pivot** view and add visuals as in step 3 above

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

There are multiple tutorials on the internet on setting up Power BI dashboards and reports, which you can find easily with a simple search.

7.2 Publishing to PowerBI.com

Upon completing the design of the reports, select **Publish** from the **File** menu to publish the data in a PowerBI.com workspace to which you have access. On PowerBI.com, the report looks as follows:

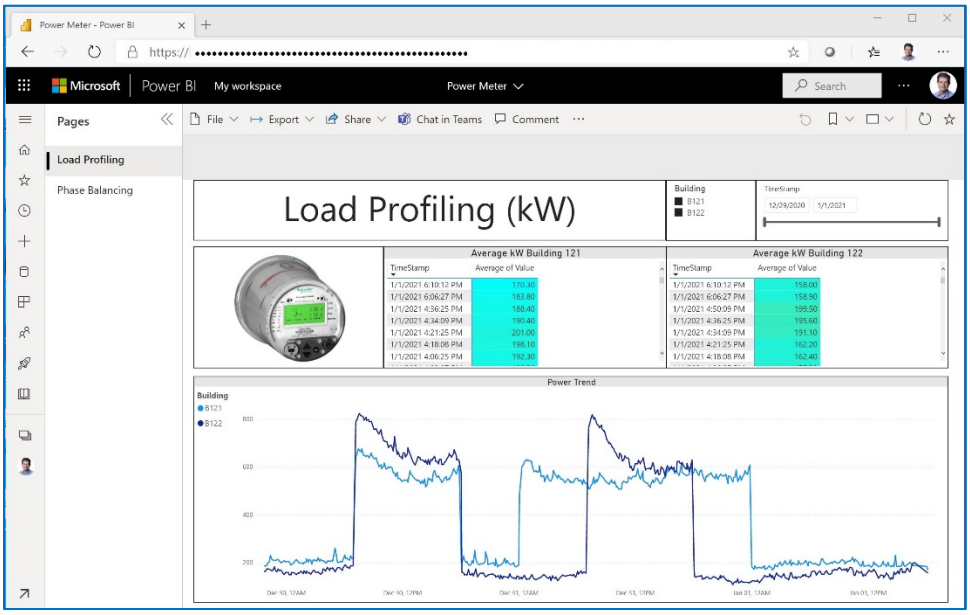


Figure 33: PowerBI.Com - Load Profiling

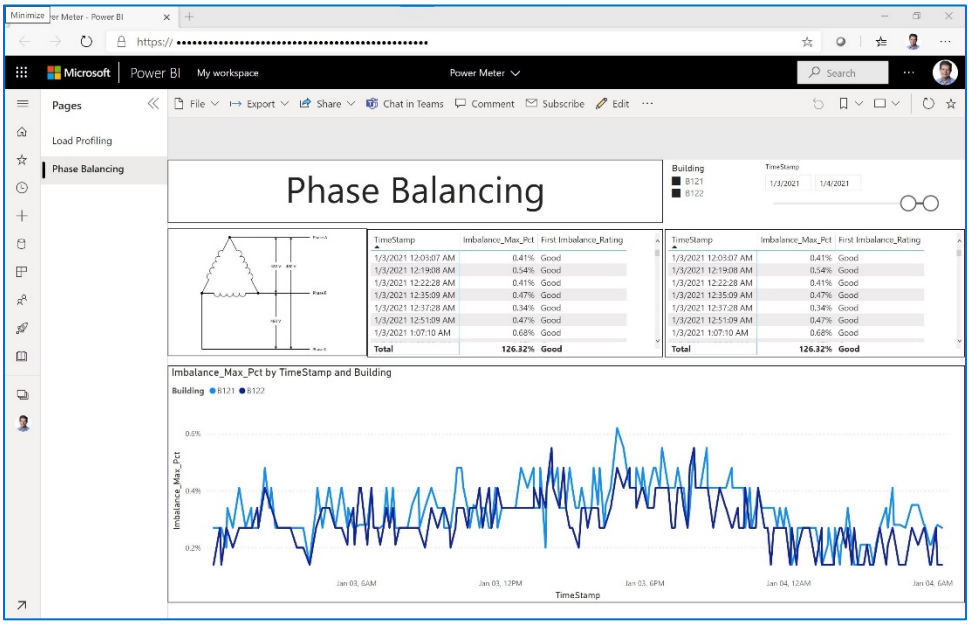


Figure 34: Power BI Phase Balancing

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

Note the pages that were accessed as tabs at the bottom of the page in Power BI Desktop are now menu options on the left in PowerBI.com.

In the Power BI Desktop application we also created a mobile phone report. Install and launch the Power BI application on a mobile phone, and open the mobile phone report:

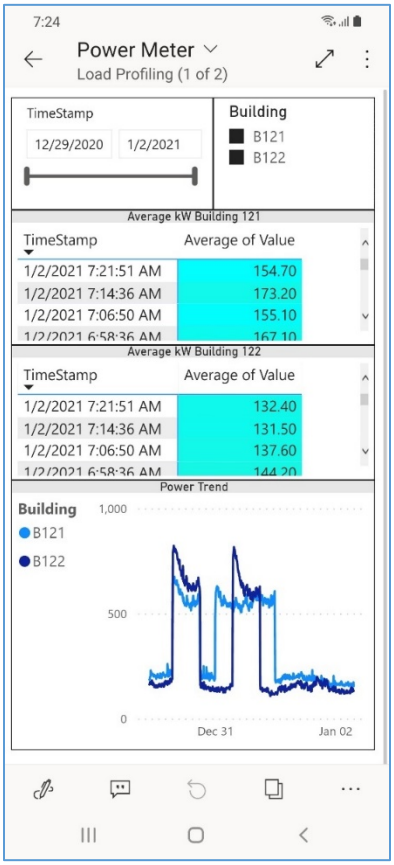


Figure 35: Power BI report on mobile phone

In order for users to be able to access these reports in PowerBI.com, they need to be given explicit access. To do this, select the **Share** option in the report on PowerBI.com and enter the email addresses of the individuals that should be granted access. In order to view those reports, those individuals will need to log in to access the reports.

7.3 Embedding a Power BI report in a website

Instead of sharing with named individuals and directing them to a report in PowerBI.com, we can embed the Power BI report in a standard website.

There are several steps required to publishing a Power BI report to a website. First, from the top menu in PowerBI.com select **Share.** From the dropdown select **Embed report,** then **Website or portal,** to get the HTML code to embed in any website.

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

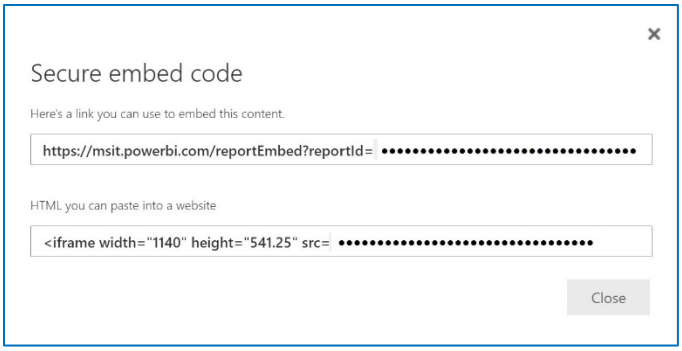


Figure 36: Getting HTML code to embed in gated website

If your Power BI policy permits publishing anonymously to a web page, from the dropdown select **Embed report,** then **Publish to web (public)**, in which case the dialog looks like this:

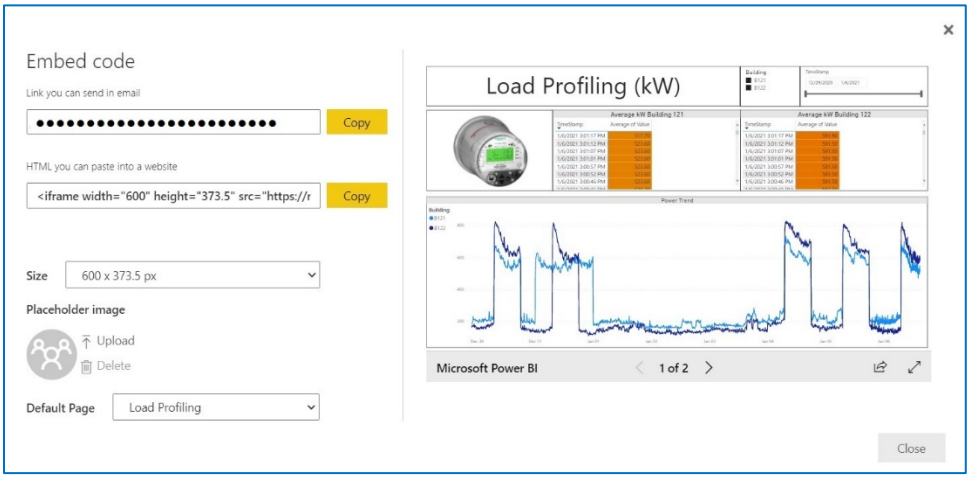


Figure 37: Getting HTML code to embed in public website

Next, create a simple website in Azure using any editing tool. For this example we used Visual Studio to create a single web page and copied the HTML code above into the page:

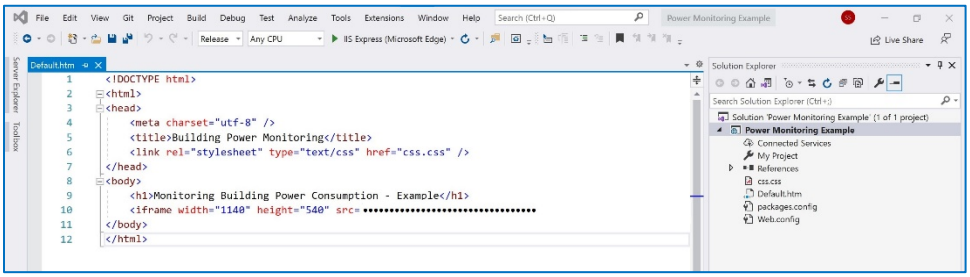


Figure 38: Visual Studio website

Having created the page, right click on the website in the Solution Explorer and select **Publish** to bring up the publishing options:

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

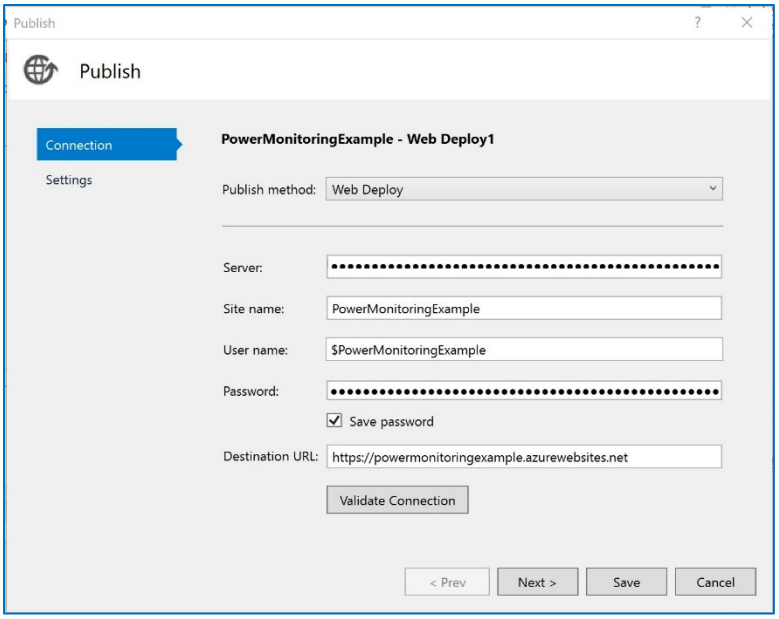


Figure 39: Visual Studio publishing options

Enter values as requested. For this example, we called the website **PowerMonitoringExample**. Click **Next** to publish. When publishing completes, select the link to view the website. On the website, the report looks as follows.

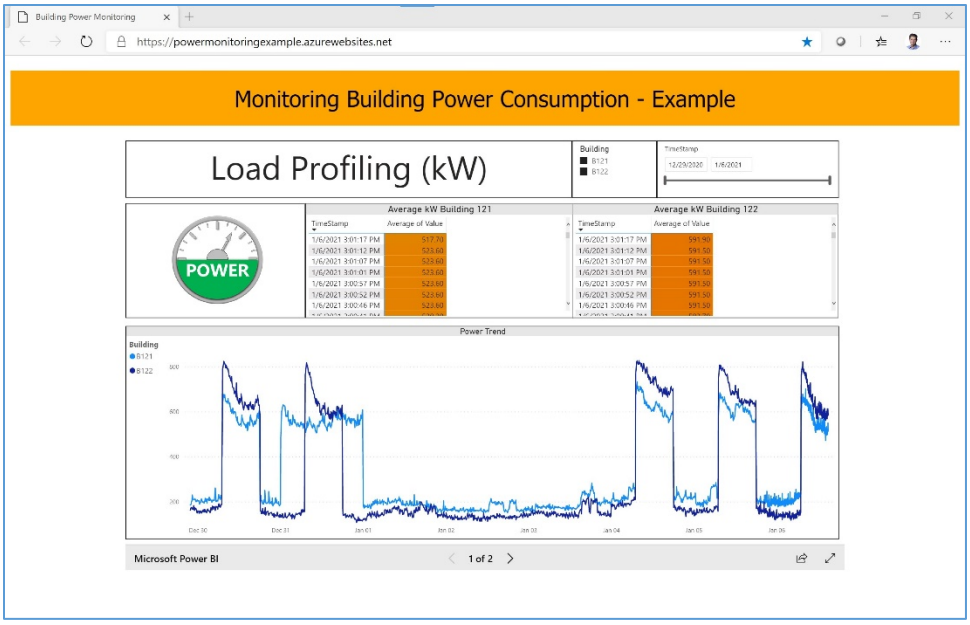


Figure 40: Power BI report embedded in web page – Load Profiling tab

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

Monitoring Building Power Consumption

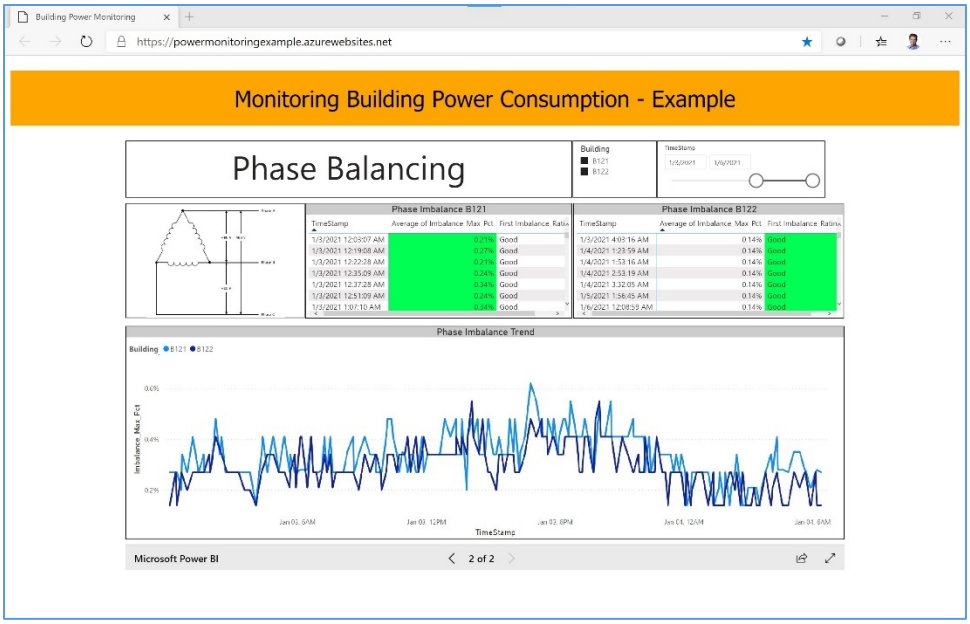


Figure 41: Power BI report embedded in web page - Phase Balancing tab

Note the pages that were accessed as menu options on the left in PowerBI.com are now tabs at the bottom of the page again.

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

8 Available partner solutions

8.1 ICONICS GENESIS64™

An ICONICS GENESIS64™ based solution is ideal for enterprise level power monitoring applications. Coupled with its fault detection capabilities, energy efficiency offenders can be quickly detected and addressed.

An ICONICS GENESIS64 power monitoring dashboard wi could look something like this:

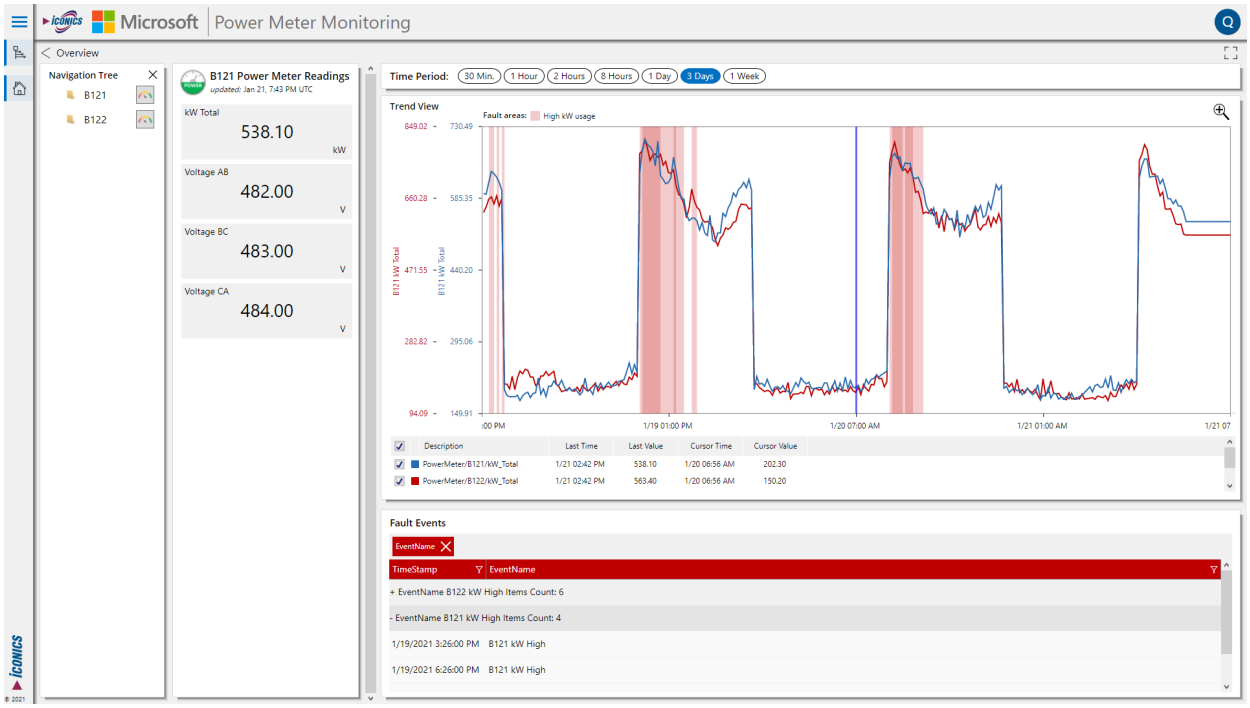


Figure 42: ICONICS power monitoring solution with fault detection of high kW usage

To explore the possibilities with ICONICS GENESIS64, simply provision the latest ICONICS Suite VM from Azure Marketplace and configure the IoT Subscriber connection as described in Section 6.4, “Pulling from IoT Hub into ICONICS GENESIS64.”

It is also recommended to leverage AssetWorX™ to organize the incoming data and easily log the data with Hyper Historian™.

The monitoring dashboard shown in Figure 42: ICONICS solution, can then be easily created with the graphic templates provided in the ICONICS Suite.

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

9 Insights

9.1 Load profiling

Typically, one of the first uses of load profiling relates to the building schedules – whether a building or group of buildings is showing a different profile than another. In the timeframe shown in our example, buildings B121 and B122 both 'turn on' at 7:00 AM and 'turn off' at 6:00 PM on Dec 30th, but B121 is 'on' from midnight to midnight on Dec 31. This warrants further investigation – was the schedule changed on purpose to support work being done in B121 on the 31st (or maybe there was a New Year's Eve event scheduled in the building)? In addition to anomalies, the schedule can be used to determine whether there are optimizations that would lead to reduced energy consumption.

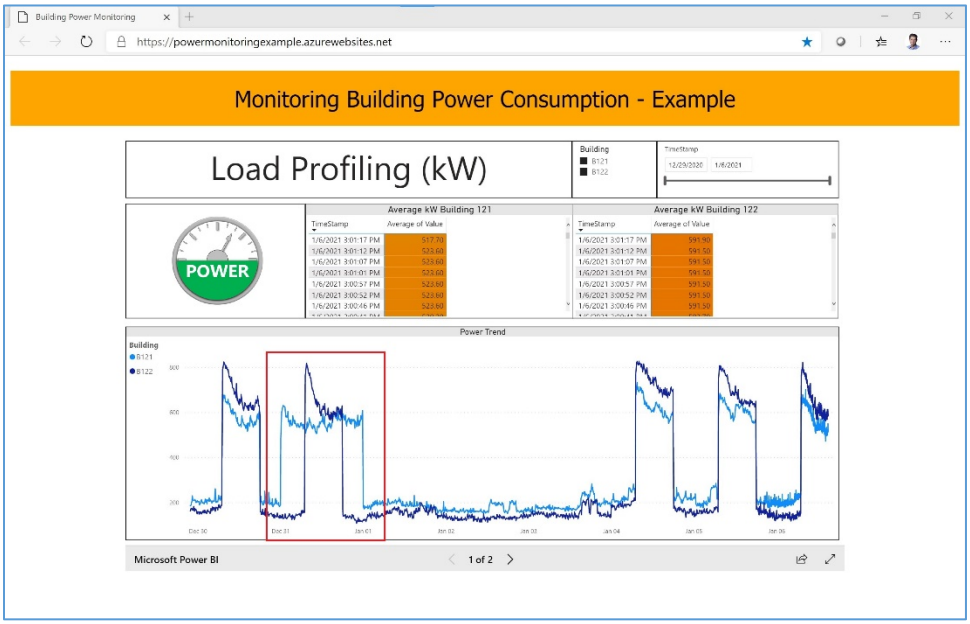


Figure 43: Building schedules

Another common use of load profiling is to reduce peak demand charges incurred when buildings come online. In our example you can see a surge we refer to as "sharking" at 7:00 AM, which flattens out as the day progresses:

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

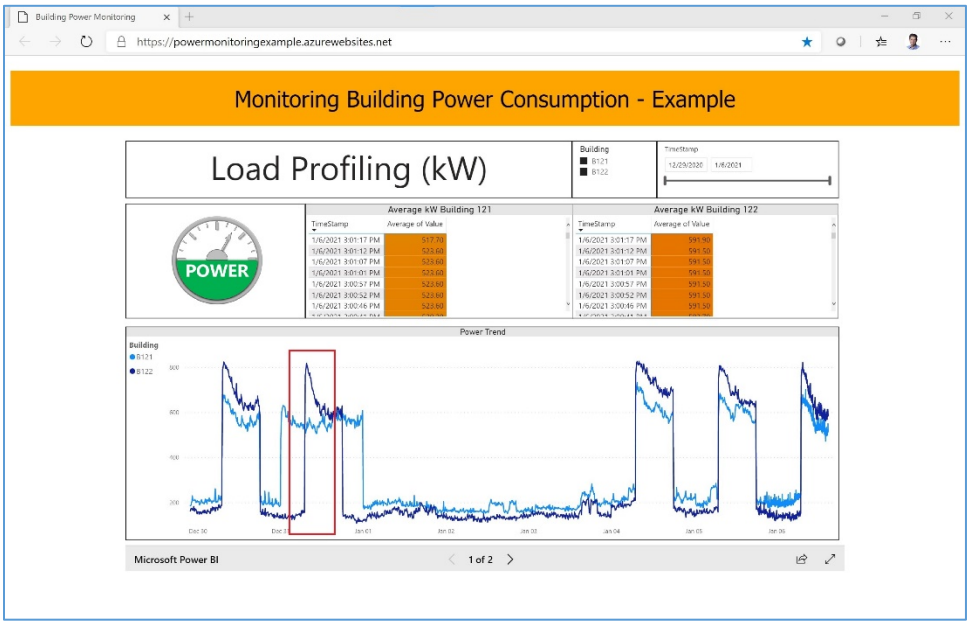


Figure 44: Demand surge ("sharking")

A reasonable goal for the operations staff would be to find ways to reduce that 7:00 AM surge without increasing the overall electrical consumption. (You could eliminate the surge by leaving the building online 24x7, but that would increase the overall consumption.). There are various AI/ML solutions available from smart building software vendors to analyze demand and make recommendations, and the impact of implementing any changes would be seen here.

9.2 Phase balancing

Line-to-Line voltage (**Voltage_AB**, for example) measurement is important to check to assess power quality. Unbalanced voltage will eventually cause overheating, low efficiency, or result in a shorter life span of equipment such as three phase fans or pumps. The U.S. Department of Energy makes the case clearly for motors:

> "Voltage unbalance degrades the performance and shortens the life of a three-phase motor. Voltage unbalance at the motor terminals can cause current unbalance that is far out of proportion to the voltage unbalance. Unbalanced currents lead to torque pulsations, increased vibrations and mechanical stresses, increased losses resulting in lower efficiency, and motor overheating, which reduces winding insulation life."
> (https://www.energy.gov/sites/prod/files/2014/04/f15/eliminate_voltage_unbalanced_motor_systemts7.pdf, accessed 1/4/21)

In our example, we found that the phase imbalance over the time period scanned maxed out at 0.6%, but averaged around 0.3%:

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.

Monitoring Building Power Consumption

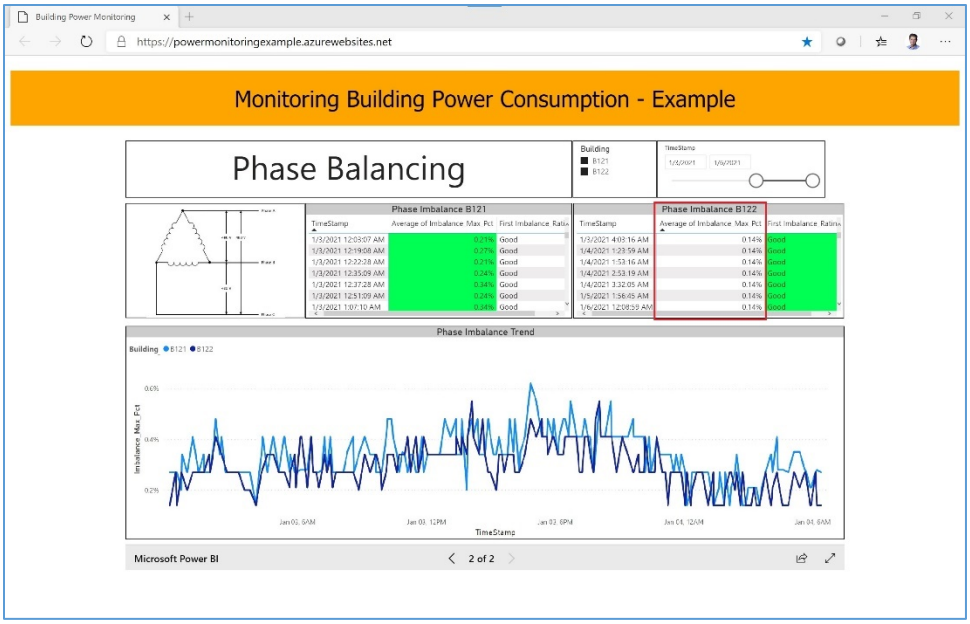


Figure 45: Phase balancing

The U.S. Department of Energy recommends voltage imbalances between the phases not exceed 1%, and any number above 4% is generally regarded as indicative of a serious problem. In our case, the phase imbalance is within acceptable limits.

© 2021, Microsoft Corporation and ICONICS, Inc.
All rights reserved.



Founded in 1986, ICONICS is an award-winning independent software provider offering real-time visualization, HMI/SCADA, energy management, fault detection, manufacturing intelligence, MES, and a suite of analytics solutions for operational excellence. ICONICS solutions are installed in 70 percent of the Global 500 companies around the world, helping customers to be more profitable, agile and efficient, to improve quality, and to be more sustainable.

ICONICS is leading the way in cloud-based solutions with its HMI/SCADA, analytics, mobile and data historian to help its customers embrace the Internet of Things (IoT). ICONICS products are used in manufacturing, building automation, oil and gas, renewable energy, utilities, water and wastewater, pharmaceuticals, automotive, and many other industries. ICONICS' advanced visualization, productivity, and sustainability solutions are built on its flagship products: GENESIS64™ HMI/SCADA, Hyper Historian™ plant historian, AnalytiX® solution suite, and MobileHMI™ mobile apps. Delivering information anytime, anywhere, ICONICS' solutions scale from the smallest standalone embedded projects to the largest enterprise applications.

ICONICS promotes an international culture of innovation, creativity, and excellence in product design, development, technical support, training, sales, and consulting services for end users, systems integrators, OEMs, and channel partners. ICONICS has over 375,000 applications installed in multiple industries worldwide.

ICONICS Sales Offices

World Headquarters

100 Foxborough Blvd.
Foxborough, MA, USA, 02035
+1 508 543 8600
us@iconics.com

European Headquarters

Netherlands
+31 252 228 588
holland@iconics.com

Australia
+61 2 9605 1333
australia@iconics.com

China
+86 10 8494 2570
china@iconics.com

Czech Republic
+420 377 183 420
czech@iconics.com

France
+33 4 50 19 11 80
france@iconics.com

Germany
+49 2241 16 508 0
germany@iconics.com

India
+91 265 6700821
india@iconics.com

Italy
+39 010 46 0626
italy@iconics.com

Singapore
+65 6667 8295
singapore@iconics.com

UK
+44 1384 246 700
uk@iconics.com

For more, visit iconics.com

© 2021 ICONICS, Inc. All rights reserved. Specifications are subject to change without notice. AnalytiX and its respective modules are registered trademarks of ICONICS, Inc. GENESIS64, GENESIS32, Hyper Historian, BizViz, PortalWorX, MobileHMI and their respective modules, OPC-to-the-Core, and Visualize Your Enterprise are trademarks of ICONICS, Inc. Other product and company names mentioned herein may be trademarks of their respective owners.

Gold
Microsoft Partner
Six-time Partner of the Year Winner

VISUALIZE HISTORIZE MOBILIZE ANALYZE IoT CLOUD