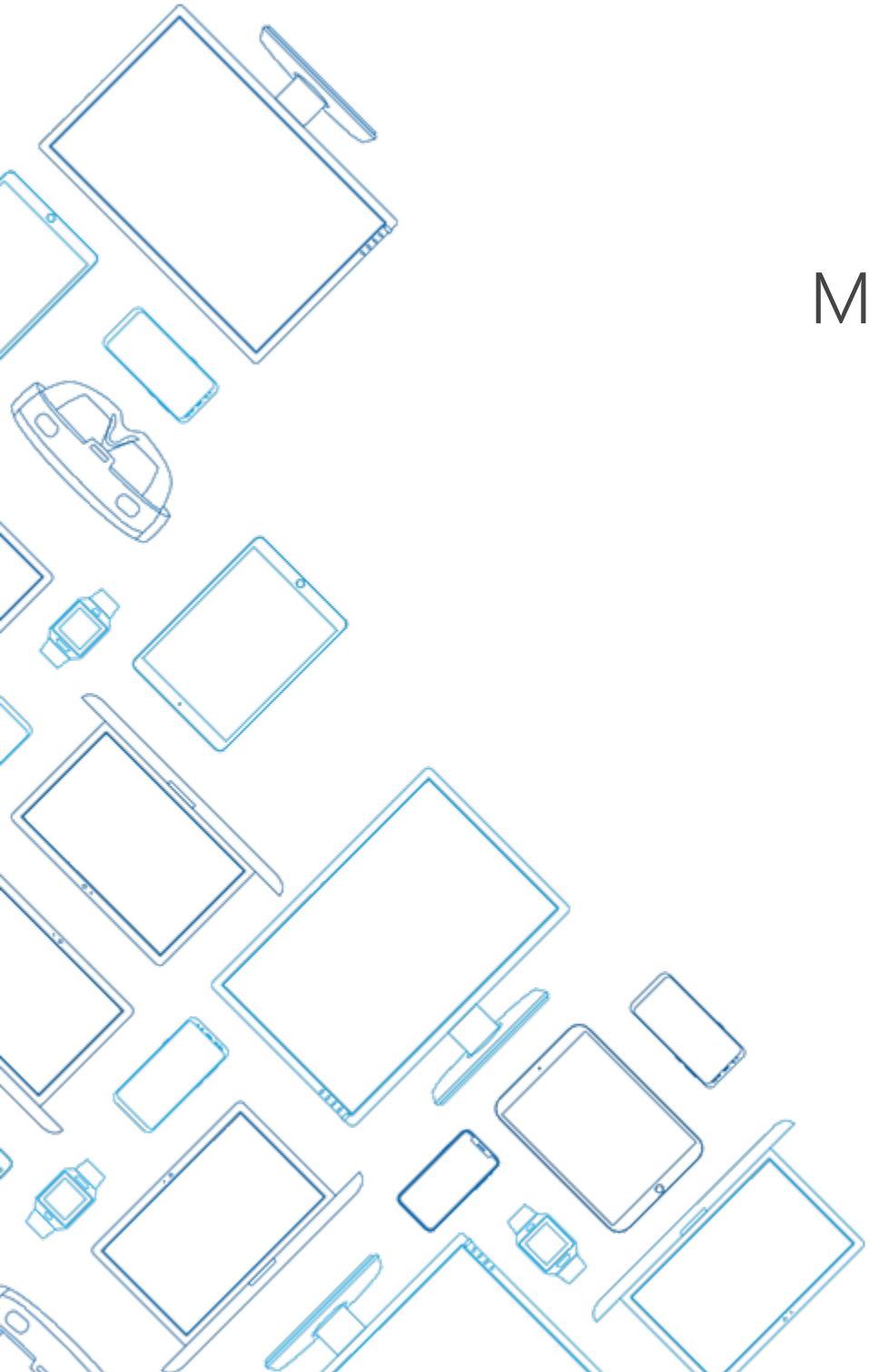
ICONICS

Make the Invisible Visible™

Microsoft Azure



Monitoring Building Air Quality

An ICONICS Whitepaper

February 2021

Contents

1 Introduction 4
2 Architecture 5
3 Air Quality data 6
4 Azure infrastructure 7
4.1 Azure Resource Group 7
4.2 Azure IoT Hub 8
4.3 SQL Server 8
5 Getting air quality data 10
5.1 Installing ICONICS IoTWorX 10
5.2 Configuring BACnet communications to the air quality meter 10
5.3 Identifying data points on the air quality sensor 11
5.4 Specifying objects to push to Azure 12
5.5 Testing device connectivity locally with Data Explorer 13
6 Publishing and storing data 13
6.1 Publishing from IoTWorX to Azure IoT Hub 13
6.2 Testing data reception at IoT Hub with Azure IoT explorer 14
6.3 Pushing from IoT Hub to SQL Server with Azure Stream Analytics 16
6.4 Pulling from IoT Hub into ICONICS GENESIS64 18
7 Displaying air quality data and trends 19
7.1 Designing a report in Power BI Desktop 19
7.2 Publishing to PowerBI.com 20
7.3 Embedding a Power BI report in a website 20
8 Available partner solutions 22
8.1 ICONICS GENESIS64 22

© 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
- Caleb Johnson, Program Manager, Critical Systems & Infrastructure, CBRE

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

1 Introduction

Monitoring the quality of the air inside or outside of buildings, and in particular the levels of carbon dioxide (CO2) and particulates (PM), is important for understanding the environment and for enabling building owners to provide a healthier place for people to live and work. The impact of particulate matter on health is well documented, for example in this article from Kaiterra:

https://learn.kaiterra.com/en/air-academy/particulate-matter-pm

The World Air Quality Index project, a non-profit project whose mission is to promote air pollution awareness for citizens, publishes the following summary of the health implications of PM2.5 particulates:

AQI	Air Pollution Level	Health Implications	Cautionary Statement (for PM2.5)
0 - 50	Good	Air quality is considered satisfactory, and air pollution poses little or no risk	None
51 -100	Moderate	Air quality is acceptable; however, for some pollutants there may be a moderate health concern for a very small number of people who are unusually sensitive to air pollution.	Active children and adults, and people with respiratory disease, such as asthma, should limit prolonged outdoor exertion.
101-150	Unhealthy for Sensitive Groups	Members of sensitive groups may experience health effects. The general public is not likely to be affected.	Active children and adults, and people with respiratory disease, such as asthma, should limit prolonged outdoor exertion.
151-200	Unhealthy	Everyone may begin to experience health effects; members of sensitive groups may experience more serious health effects	Active children and adults, and people with respiratory disease, such as asthma, should avoid prolonged outdoor exertion; everyone else, especially children, should limit prolonged outdoor exertion
201-300	Very Unhealthy	Health warnings of emergency conditions. The entire population is more likely to be affected.	Active children and adults, and people with respiratory disease, such as asthma, should avoid all outdoor exertion; everyone else, especially children, should limit outdoor exertion.
300+	Hazardous	Health alert: everyone may experience more serious health effects	Everyone should avoid all outdoor exertion

Figure 1: From https://aqicn.org/faq/2013-09-09/revised-pm25-aqi-breakpoints/

With this in mind, look at the historical data for an area near where Microsoft has a large number of buildings in the U.S:

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

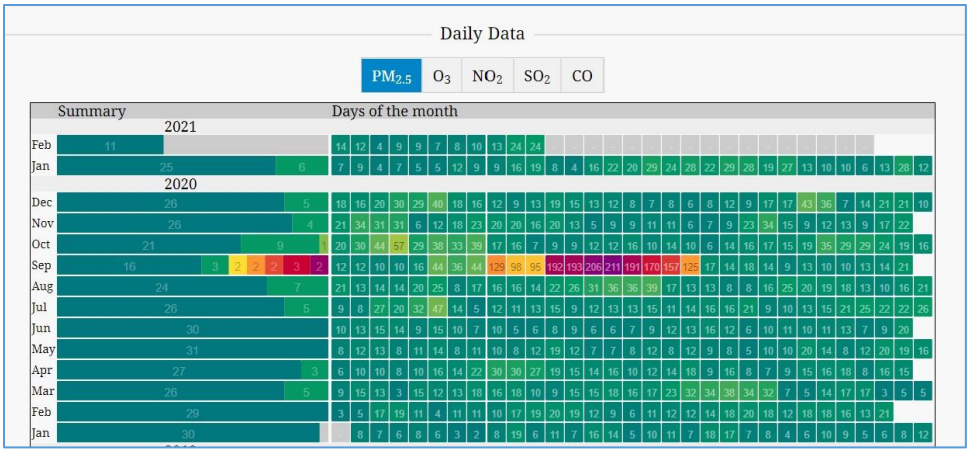


Figure 2: From https://aqicn.org/historical#!city:usa/washington/bellevue/se-12th-st

For most of 2020, the air around the Microsoft offices was rated 'Good', but in September there were a lot of wildfires East of the offices, and the smoke from those fires raised the pollution level to "Very Unhealthy". Clearly, monitoring the air quality in and around the buildings is extremely important to understand what measures need to be taken to protect the health of the people working in them.

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 air quality. 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.

2 Architecture

This article documents a simple infrastructure to monitor the air quality in or outside of a building by connecting to a sensor that supports data acquisition over the BACnet protocol. The infrastructure comprises several distinct pieces:

1. An on-premises computer running software to communicate with the sensor 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.

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

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

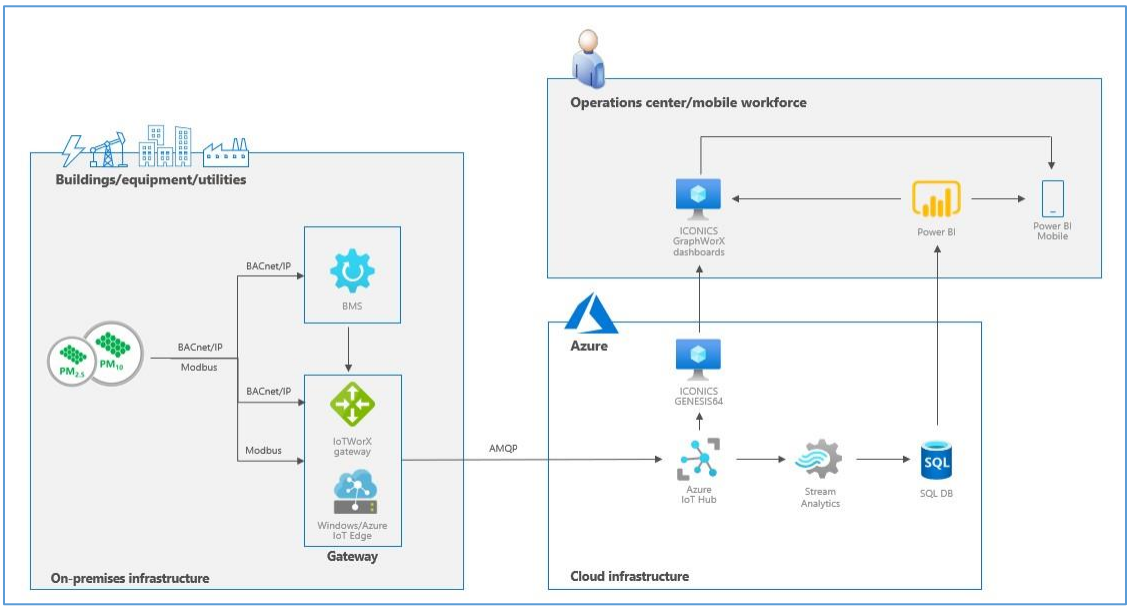


Figure 3: Architecture

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*
- Monitoring Building Power Consumption, referred to as *Power Monitoring*

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

3 Air Quality data

Air Quality sensors vary in functionality and are available from multiple manufacturers. For this article we chose to monitor the air quality using a Kaiterra Sensedge device installed in a Microsoft's Operations Center on its Redmond, Washington campus:

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



Figure 4: Kaiterra Sensedge

The Sensedge device exposes data about the environment using the BACnet™ protocol, and individual metrics can be read by any application that supports BACnet. For example, we can read the CO2 or PM2.5 data using the ICONICS IoTWorX gateway.

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 **AirQualityMonitoring** when creating this configuration:

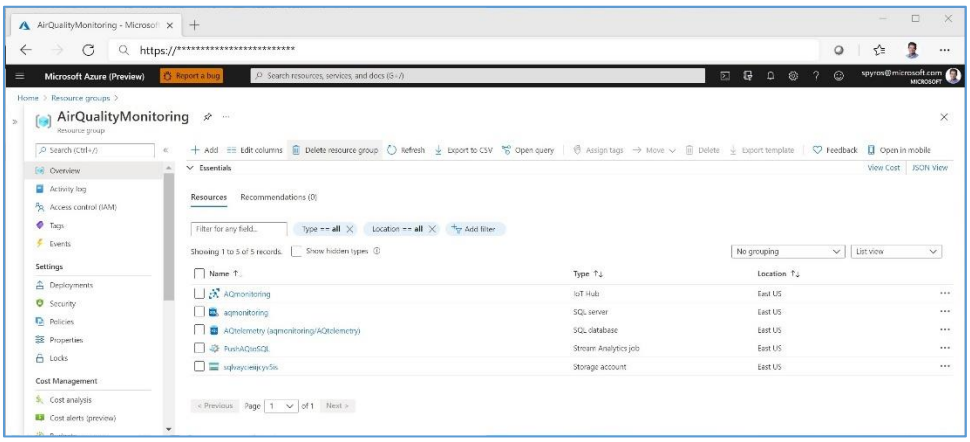


Figure 5: Resource Group with Azure services used for monitoring air quality sensor

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

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

Resource	Type	Function
AQMonitoring	Azure IoT Hub	Receive data from IoTWorX gateway
AQTelemetry	SQL database	Store data received from IoTWorX gateway
aqmonitoring	SQL server	Holds SQL database
PushAQtoSQL	Stream Analytics job	Push data from IoT Hub to SQL
sqlvaycieiijcyv5is	Storage account	Used by resources to house configurations

Figure 6: Azure services used

4.2 Azure IoT Hub

Prior to installing the on-premises components necessary to read the air quality 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 air quality example, we used the following values when creating this configuration:

Azure Service	Value
Azure IoT Hub	AQMonitoring
IoT Device	Sensedge

Figure 7: IoT Hub values

The completed deployment is shown here:

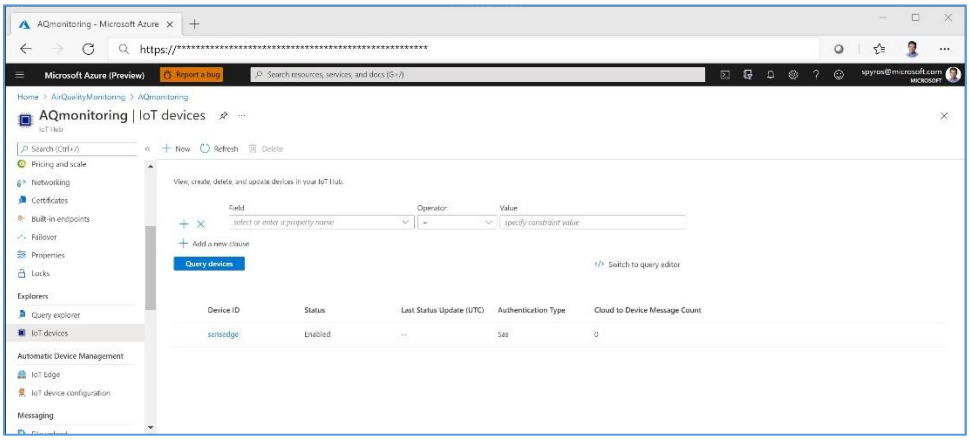


Figure 8: IoT Hub

The ICONICS IoTWorX software on the on-premises gateway will be configured to send air quality data to the **Sensedge** 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:

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

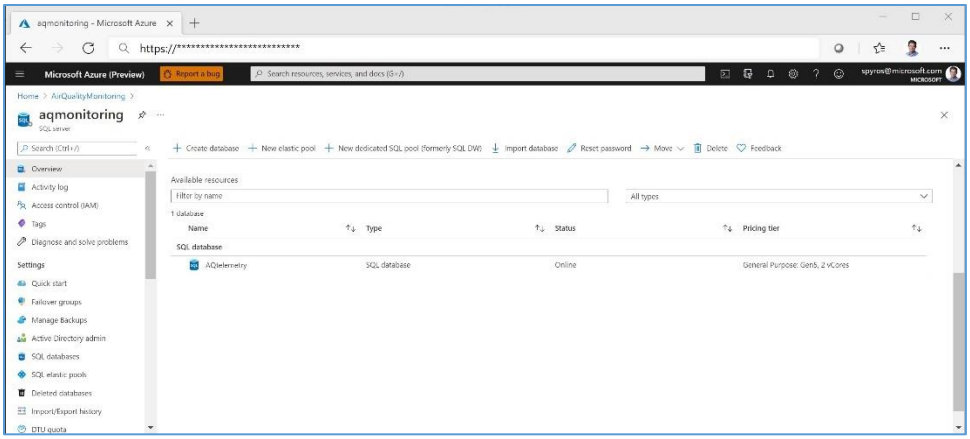


Figure 9: SQL Server overview

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

Azure Service	Value
SQL Server	aqmonitoring.database.windows.net
SQL Database	AQTelemetry
SQL Table	dbo.RawTelemetry
SQL View	dbo.v_AQ_Pivot

Figure 10: SQL Server values

The **RawTelemetry** table will hold the data as it comes from the air quality sensor, storing each measurement in a separate record. The view **dbo.v_AQ_Pivot** will display the same data, but all the measurements at a specific time will be stored in a single record. 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 11: RawTelemetry table schema

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

Finally, the view definition is as follows:

```
CREATE VIEW [dbo].[v_AQ_Pivot] AS
SELECT *
FROM
  (SELECT PVTSI.Building,
          PVTSI.TimeStamp,
          PVTSI.CO2,
          PVTSI.PM25 AS [PM2.5],
          PVTSI.PM10,
          PVTSI.Humidity,
          PVTSI.Temperature
   FROM
     (SELECT PVTS.*
      FROM
        (SELECT CONVERT(date, TIMESTAMP) AS Date,
                Building,
                TimeStamp,
                [CO2],
                [PM25],
                [PM10],
                [Humidity],
                [Temperature]
         FROM
           (SELECT t1.*
            FROM RawTelemetry t1) AS SourceTable PIVOT(MAX(Value)
            FOR PARAMETER IN([CO2], [PM25], [PM10], [Humidity], [Temperature])) AS PivotTable) AS PVTS
      WHERE [CO2] IS NOT NULL AND [PM25] IS NOT NULL AND [Temperature] IS NOT NULL) AS PVTSI) AS PVTSIP
```

Figure 12: v_AQ_Pivot

5 Getting air quality 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 BACnet communications to the air quality meter

Support for BACnet access to the Kaiterra Sensedge device is described in the following article:

https://learn.kaiterra.com/en/resources/introducing-bacnet/ip-integration-with-kaiterra-sensedge

Here we will describe how we did this with IoTWorX. First, we created a BACnet port in Workbench:

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

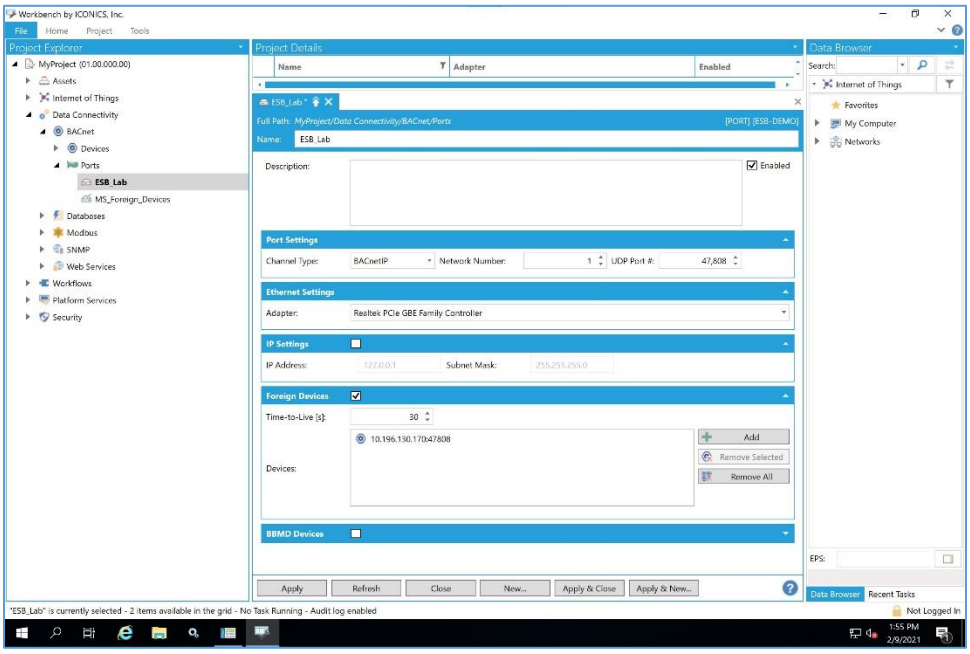


Figure 13: BACnet port

5.3 Identifying data points on the air quality sensor

Having specified how to communicate with the air quality sensor meter, we need to identify what data it collects from the air. To do this, first we initiate a BACnet device discovery in Workbench, by right-clicking Devices under the Data Connectivity/BACnet folder and selecting Network Discovery:

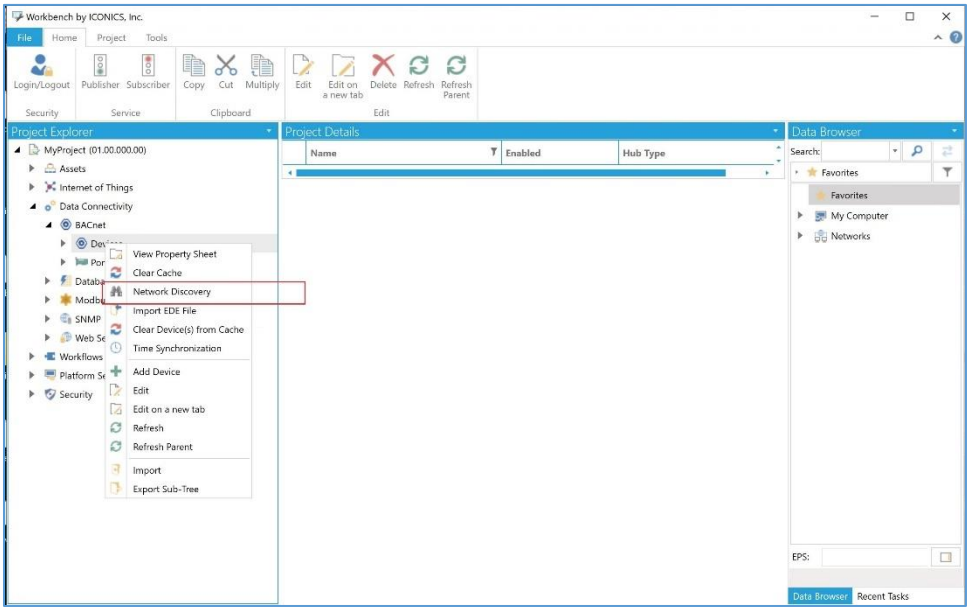


Figure 14: Initiating a BACnet network discovery

In this case, the discovery identified the Sensedge device, and a series of metrics it measures:

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

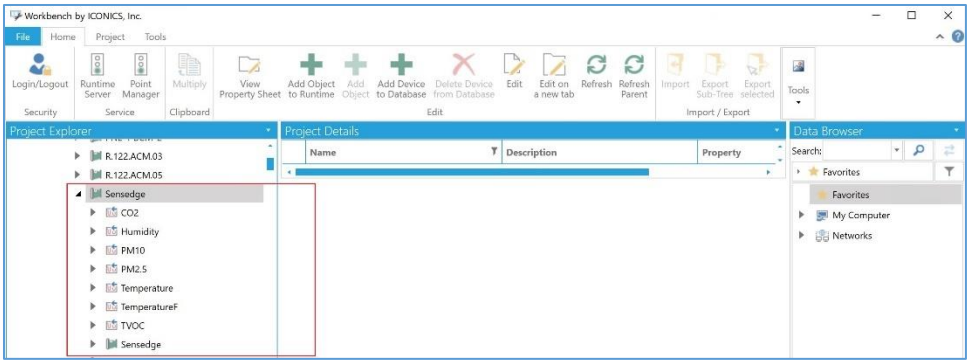


Figure 15: Sensedge device and objects measured

5.4 Specifying objects to push to Azure

At this point, IoTWorX is configured to get data from the air quality sensor. 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 **AirQuality_PublishList** when creating this example:

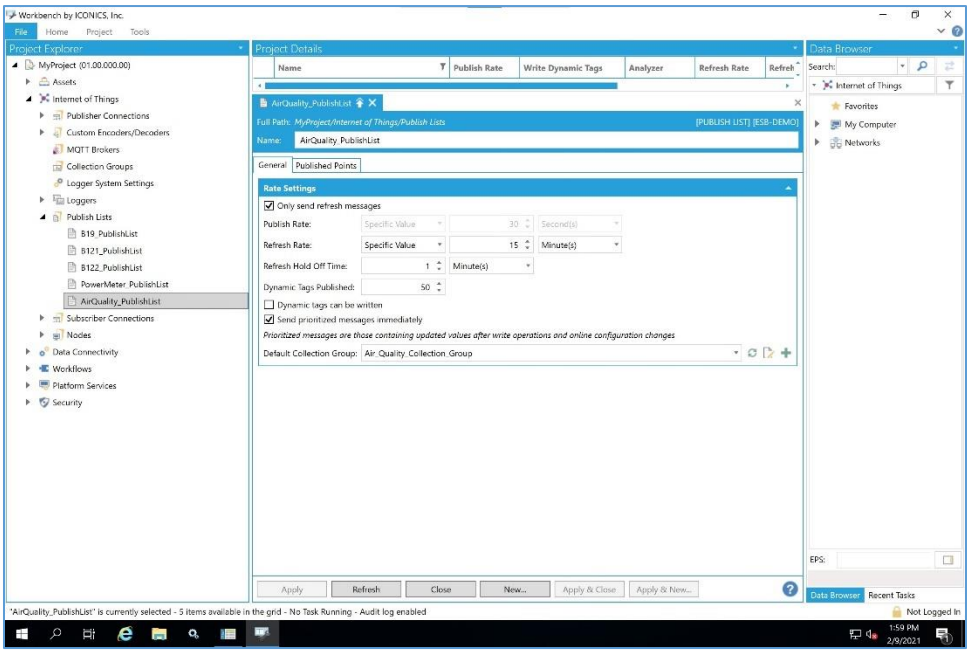


Figure 16: IoTWorX Publish List

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

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

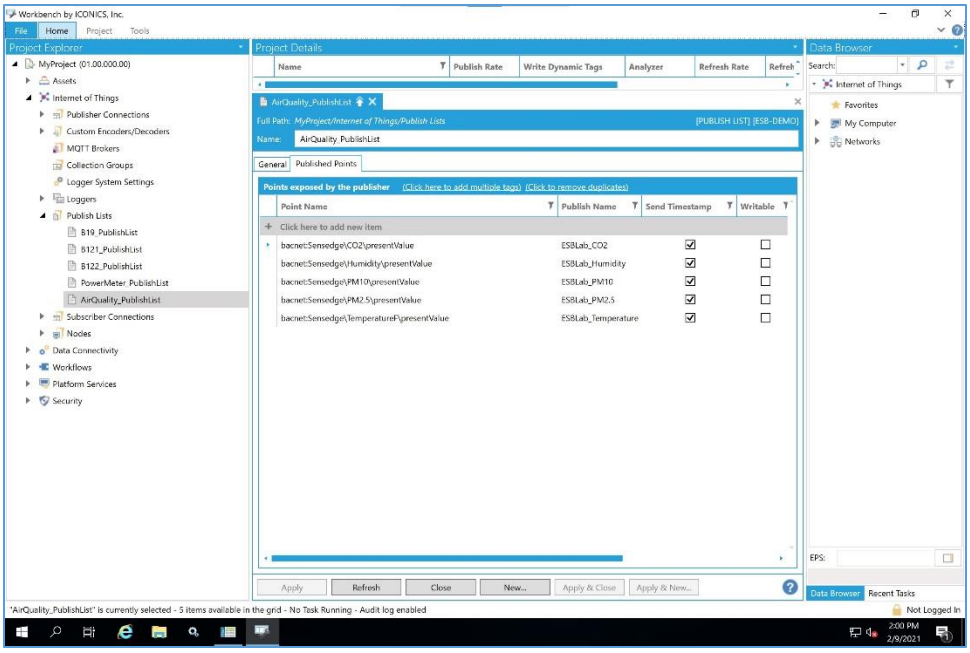


Figure 17: IoTWorX Published Points

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 → BACnet → Sensedge** and double click on **CO2**. You should see the data values similar to the following:

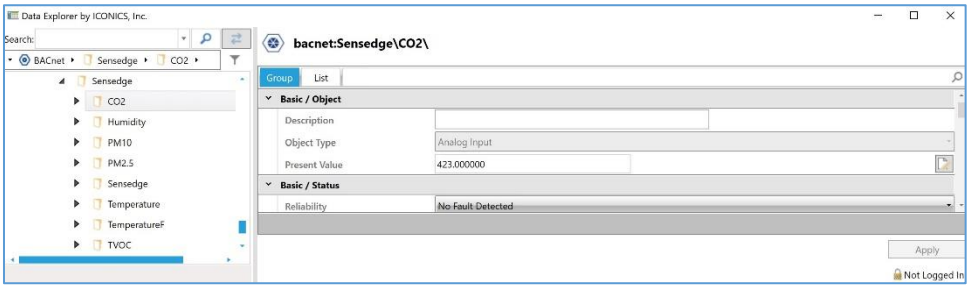


Figure 18: Viewing data collected by IoTWorX using ICONICS Data Explorer on the gateway machine

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:

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

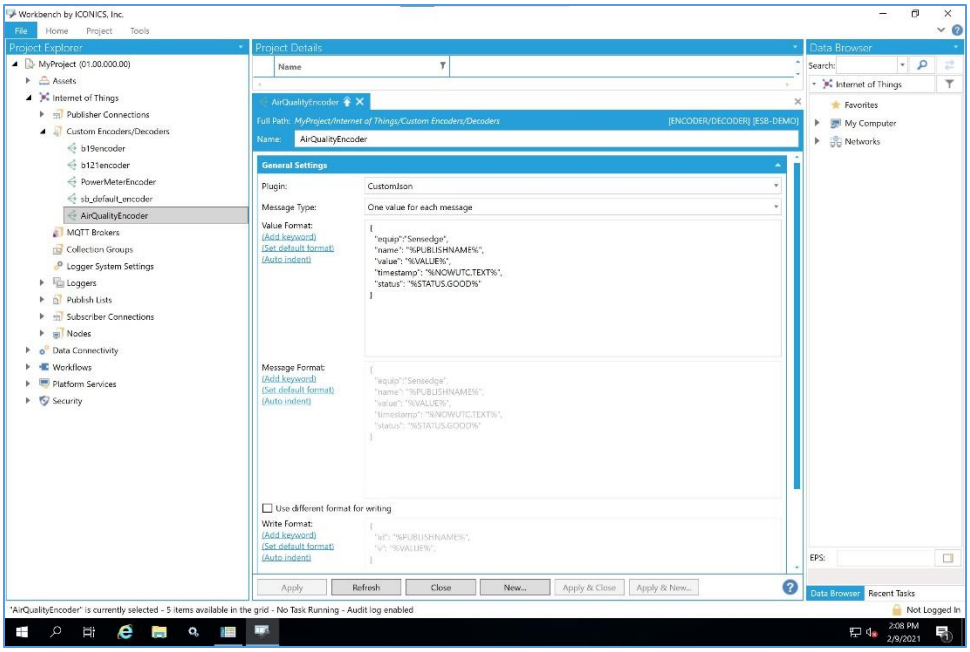


Figure 19: IoTWorX Custom Encoder

The Publisher Connection is shown below, using the IoT Connection String for the **Sensedge** device as shown in Figure 8: IoT Hub above:

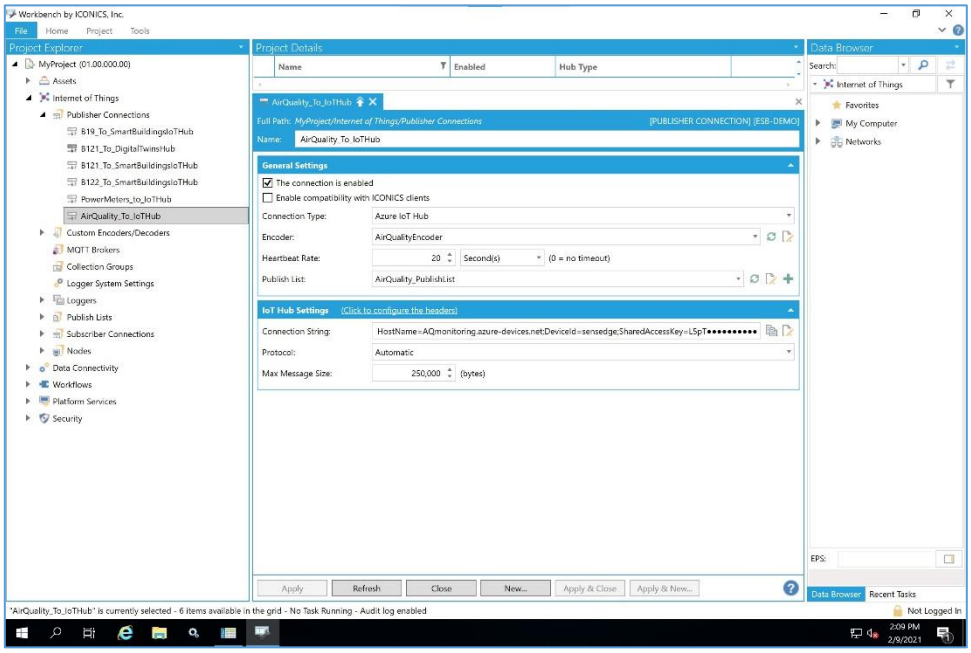


Figure 20: IoTWorX Publisher Connection

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:

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

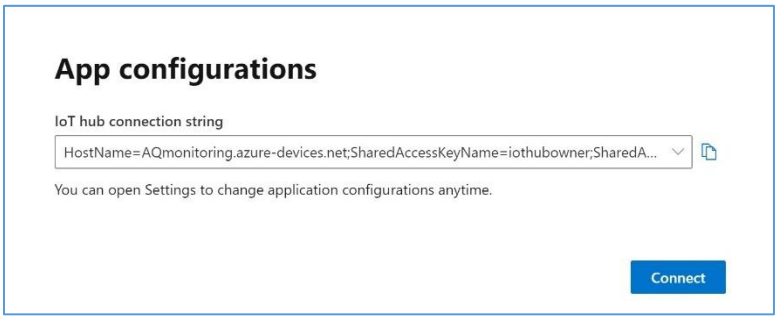


Figure 21: Azure IoT explorer configuration

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

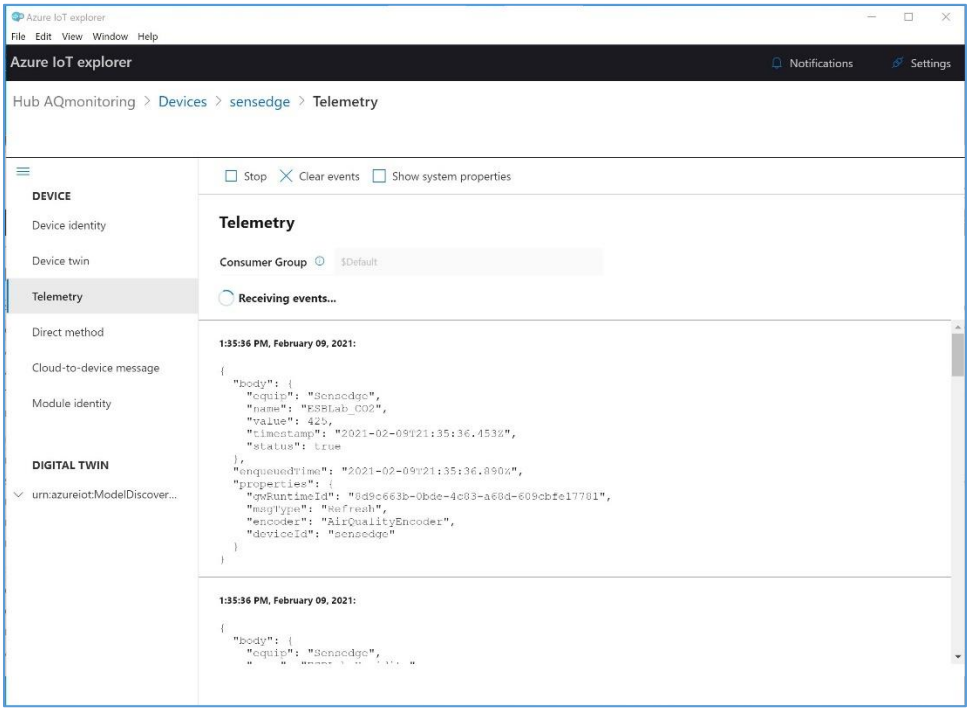


Figure 22: Viewing data received by IoT Hub

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

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

IoTWorX Encoder	Value shown above
Equip	Sensedge
Name	ESBLab_CO2
Value	425
TimeStamp	2021-02-09T21:35:36.453Z
Status	True

Figure 23: Data received

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 **PushAQtoSQL**:

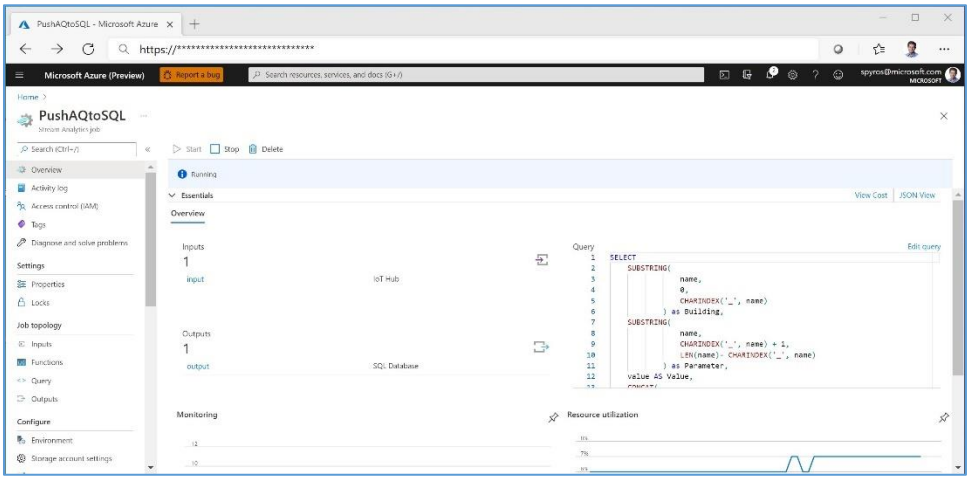


Figure 24: Azure Stream Analytics job

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 8: IoT Hub. For the output, we specified the SQL Server table shown in Figure 9: SQL Server overview. Then, for the query we entered the following:

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

```
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 25: Azure Stream Analytics Query

The first two SUBSTRING clauses parse the name (e.g. ESBLab_PM2.5) into two fields, Building (e.g. ESBLab) and Parameter (e.g. PM2.5). 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** table in the SQL Server database. We can see that it has been parsed and received correctly using a simple query:

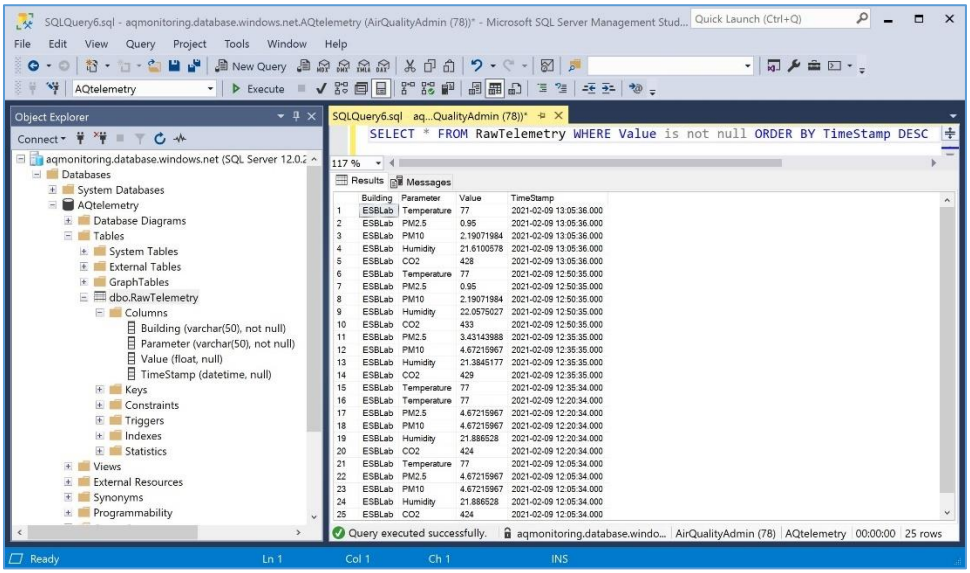


Figure 26: SQL Server RawTelemetry listing

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

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

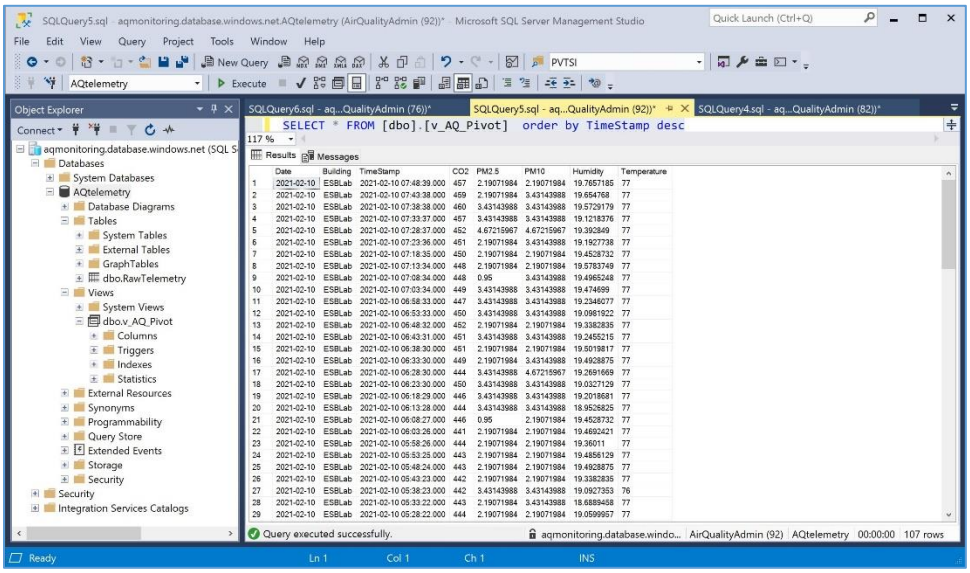


Figure 27: SQL view listing

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 19: 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 **AQMonitoring**.

To visualize the data being ingested by the subscriber service, bring up the ICONICS Data Explorer from the Start Menu in the ICONICS VM. Navigate to My Computer → Assets → Environmental and double click on Sensedge to see the data received by ICONICS in the VM:

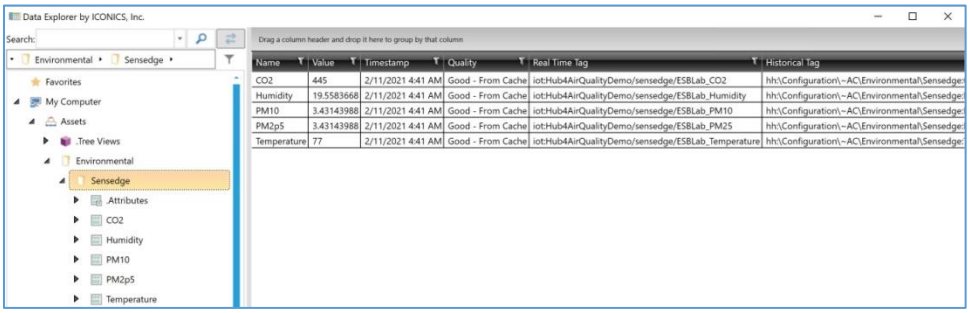


Figure 28: Viewing data received by GENESIS64 using Data Explorer in the ICONICS Azure VM

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

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

Data values correspond to those seen in the Azure IoT explorer, shown in Figure 18: Viewing data collected by IoTWorX using ICONICS Data Explorer on the gateway machine.

7 Displaying air quality data and trends

7.1 Designing a report in Power BI Desktop

Figure 29: Power BI Desktop shows a Power BI report pulling the data from the SQL table:



Figure 29: 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 **Import Data from SQL Server**
3. Enter **aqmonitoring.database.windows.net** for the SQL Server, **AQTelemetry** 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 air quality sensor
    d. Matrix visuals for the grids
    e. A Line Chart for the trends

© 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:

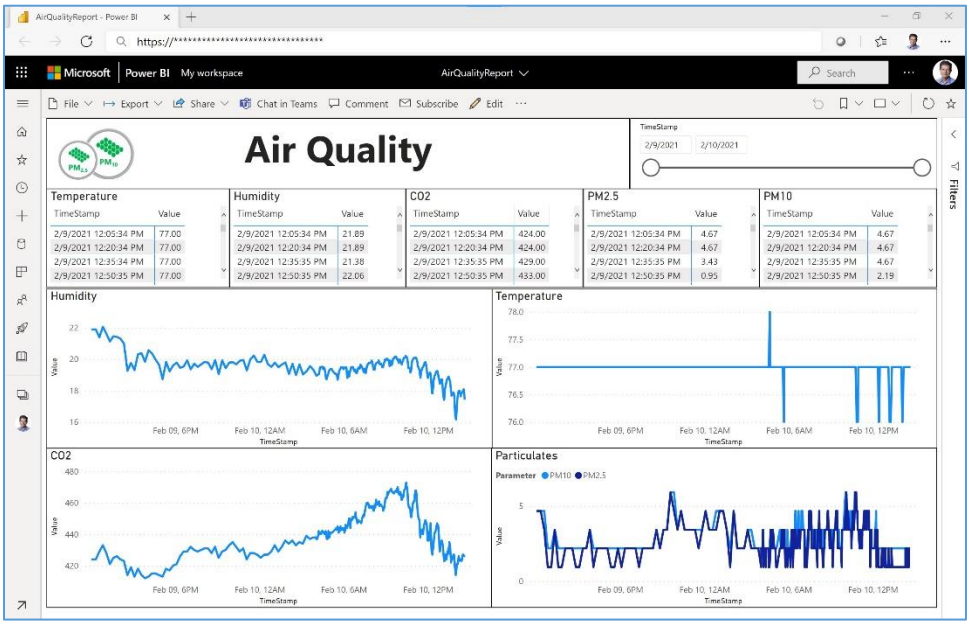


Figure 30: PowerBI.com

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.

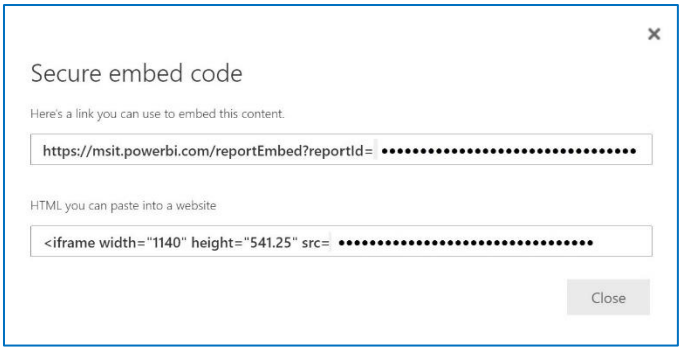


Figure 31: Getting HTML code to embed in gated website

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:

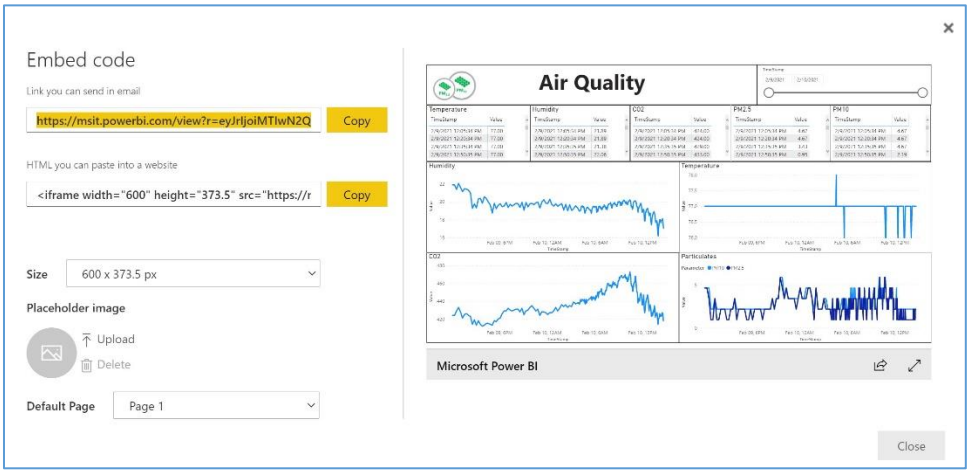


Figure 32: Getting HTML code to embed in public website

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:

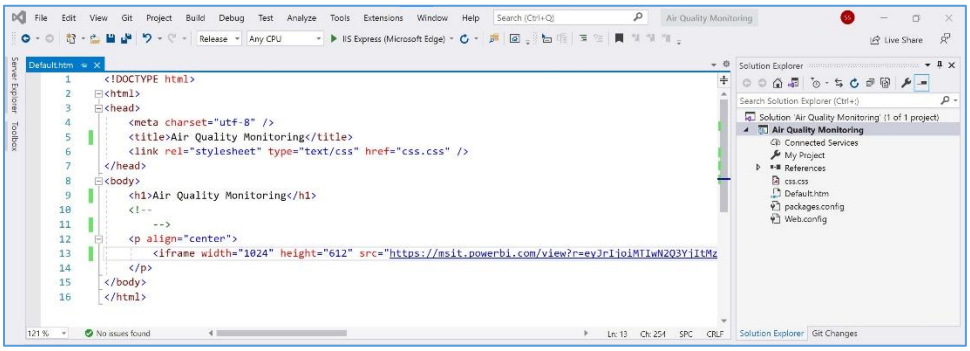


Figure 33: Visual Studio website

Having created the page, right click on the website in the Solution Explorer and select **Publish.** Enter values and credentials as requested and select **Next**. For this example, we called the website **AirQualityMonitoring**. When publishing completes, select the link to view the website. On the website, the report looks as follows.

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

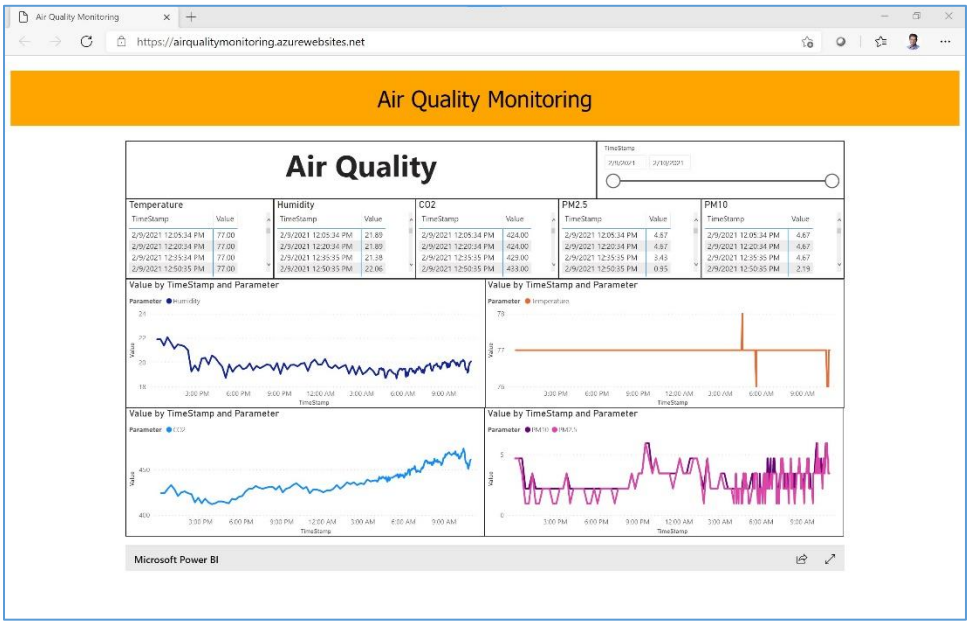


Figure 34: Power BI report embedded in web page

8 Available partner solutions

8.1 ICONICS GENESIS64

An ICONICS GENESIS64 based solution is ideal for enterprise level air quality monitoring applications, where it could be coupled with alerting functionality to identify when air quality is such that certain actions need to be taken. For example, outside air quality can be used to determine whether to incorporate outside air for heating or cooling a building, and inside air quality can be used to determine whether to activate air purifiers.

An ICONICS GENESIS64 air quality monitoring dashboard could look something like this:

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

Monitoring Air Quality

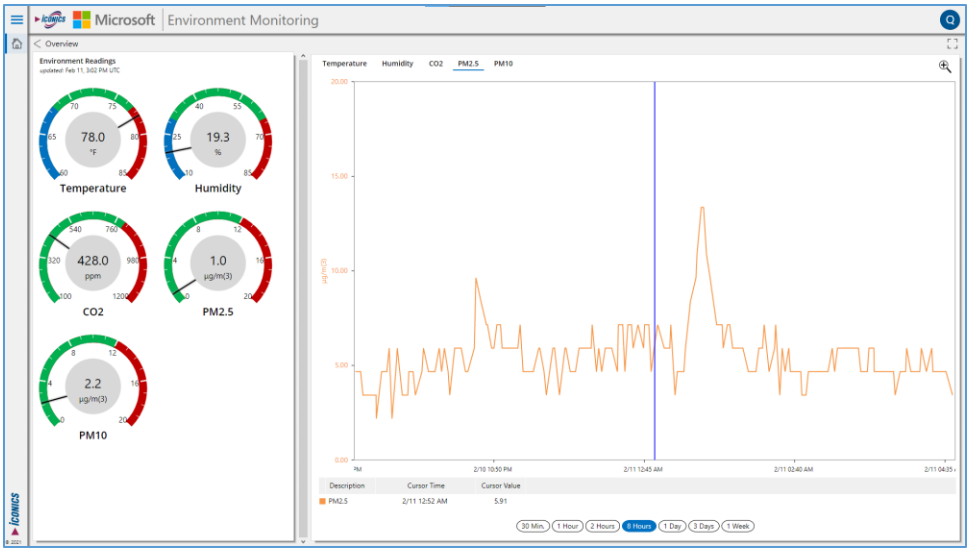


Figure 35: ICONICS air quality monitoring dashboard

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." The monitoring dashboard shown above can then be easily created with the graphic templates provided in the ICONICS Suite, and you can use AssetWorX™ to organize the incoming data and easily log the data with Hyper Historian™.

© 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, IoTWorX, KPIWorX, MobileHMI, WebHMI and their respective modules, OPC-to-the-Core, Make the Invisible Visible, and ICONICS company logo, 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