

Real-Time Data Logging and Visualization using KEPServerEX, SQL Server, and Power BI

A Comprehensive Approach to Real-Time Data Monitoring
(Guide)

By: Atharva Gondhali

Guided by: Mr. Shreyas Mokashi

Date:

03/07/2024

Index

1. Introduction
 - Overview of the Project
 - Technologies Used
2. Block Diagram
3. KEPServerEX Configuration
 - Connecting to Data Sources
 - Configuring Data Logging
4. SQL Server Express Configuration
5. Power BI Dashboard Development
 - Connecting Power BI to SQL Server
 - Creating Visuals
6. Creating and Configuring Dashboards in Power BI
 - Designing Visuals and Reports
 - Setting Up Real-Time Data Refresh
7. Conclusion
8. References

Introduction

Overview of the project:

This project aims to develop a real-time data logging and visualization system using KEPServerEX, SQL Server Express, and Power BI. The primary objective is to seamlessly integrate industrial data sources, enabling real-time monitoring and insights.

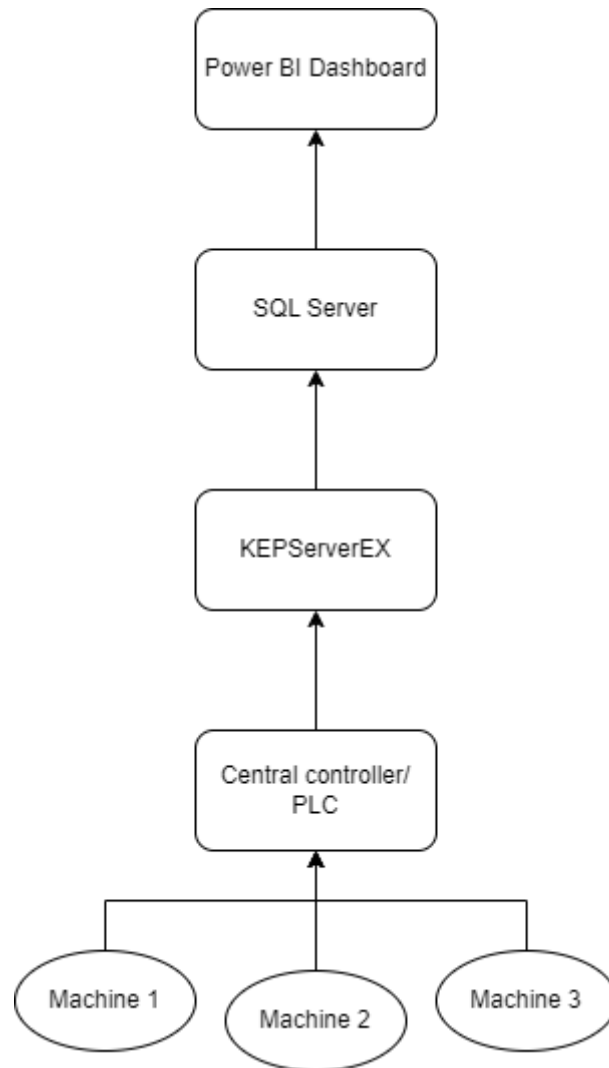
Objectives:

1. **Data Logging:** Configure KEPServerEX to log data from various industrial sources directly into SQL Server Express.
2. **Database Management:** Set up SQL Server Express to efficiently store and manage the logged data.
3. **Real-Time Visualization:** Develop a Power BI dashboard that fetches data from SQL Server using the DirectQuery method, ensuring real-time data visualization and insights.

Technologies Used:

- **KEPServerEX:** Kepware KEPServerEX is an industrial connectivity platform that allows you to connect to various devices which comprise of PLC, OPC Server, RTU, Database, as well as Sensor & Actuator like Modbus.
- **SQL Server Express:** A relational database management system used to store and manage the logged data.
- **Power BI:** A business analytics tool used to create interactive visualizations and real-time dashboards.

Block Diagram



SQL Server Express Configuration

While installing SQL server Express make sure you choose authentication type as mix ie. Windows System + User and password. The username for admin would be 'sa'. Keep the services of SQL server as run as automatic for easier server detection. After installing sql server check is ODBC data sources 32 bit is installed. After that install Microsoft SQL Server Management Studio or simply SSMS. After completing all the installations open SSMS and connect to database engine providing all the login information.

Create a new database for KEPServerEX. Creation of tables is not required as it is done automatically by KEPServerEX.

KEPServerEX Configuration

Connecting to data sources:

As for the demo and implementation purposes KEPServerEX is configured to generate random machine data.

Steps to generate simulated values:

1. Create a demo channel with default or case related properties.
2. After creation of channel create a demo device with driver as 'Simulator' and model as '16 Bit Device', keeping the rest default of as per given case.

Property Editor - Demo_Channel1.FD1

Property Groups

- General
- Scan Mode

Identification

Name	FD1
Description	
Driver	Simulator
Model	16 Bit Device
Channel Assignment	Demo_Channel1
ID Format	Decimal
ID	1

Operating Mode

Data Collection	Enable
-----------------	--------

Tag Counts

Static Tags	10
-------------	----

Name
Specify the identity of this object.

Defaults OK Cancel Apply Help

3. After the demo device is created create demo tags as per requirements as the address as 'RANDOM()' function. It is a function of simulator driver. To know more you can navigate to Help->Server help->Simulator(contents). You can keep the rest of the properties as per your requirement.

Tag Name	Address	Data Type	Scan Rate
CO	RANDOM (3000, 23, 28)	Short	5000
CO2	RANDOM (3000, 22, 28)	Short	5000
Cooling Temperature	RANDOM (2000, 25, 30)	Short	2000
Load	RANDOM (10000, 75, 80)	Short	5000
NOX	RANDOM (3000, 21, 28)	Short	5000

4. After creating tags you can test the project by launching quick client from toolbar and cross check the tag values.

Configuring data logging:

To log the tag data to database you will need to create a system DSN.

Steps to create a System DSN:

1. Open ODBC Data sources 32bit. This should be installed along with SQL server.
2. Navigate to System DSN and select Add option.
3. Select 'SQL Server' as driver and provide with database connection information such as server name, user name, password etc.
4. Test the connection and save it.

After creating a DSN, open KEPServerEX and follow the below steps:

1. Under Data Logger create a new Log group.
2. Provide the previously connected DSN details.
3. Set Table properties as 'Create a new table once and always append to it..'
4. Provide a table name and set the format as Wide.
5. The rest settings can be kept default
6. Under the log group now create new log item with unique Numeric ID and map the Item ID with corresponding Tags by browsing the selection menu

Name	ItemID	Description
CO	Demo_Channel1.FD1.CO	
CO2	Demo_Channel1.FD1.CO2	
Cooling	Demo_Channel1.FD1.Cooling Tempera...	
Load	Demo_Channel1.FD1.Load	
NOX	Demo_Channel1.FD1.NOX	
O2	Demo_Channel1.FD1.O2	
Fill	Demo_Channel1.FD1.Percent_fill	
Power	Demo_Channel1.FD1.Power	
Pressure	Demo_Channel1.FD1.Pressure	
Temperature	Demo_Channel1.FD1.Temperature	

7. After creating log items, you can now modify column mappings and trigger properties as per requirements.

Power BI Dashboard Development

Connecting Power BI to SQL Server:

Open Microsoft Power BI and follow the steps:

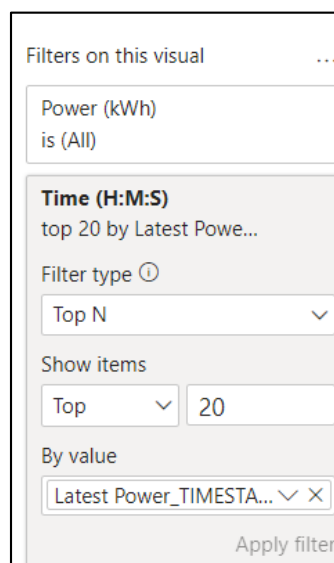
1. Create a blank report, under the data section select SQL Server.
2. Provide with the server name and database name and select Direct Query data connectivity mode (important).
3. Transform the data as per requirements and save it.

Creating Visuals:

Create visuals and design the dashboard as per your liking.

To add data to visuals, follow the steps:

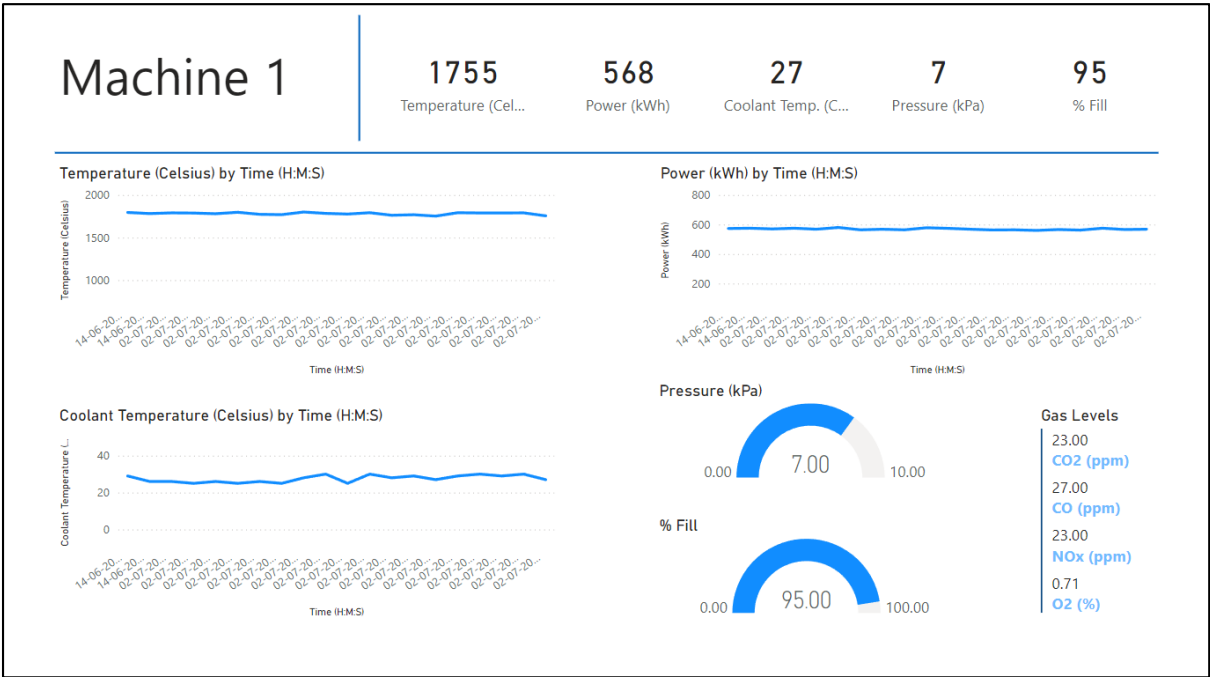
1. You can simply drag the data column from table into the fields of visuals.
2. Set the value as 'average' and rename the field as per requirement.
3. Depending upon the visual if graph, keep the timestamp on x axis and make the y axis type categorical. Keep the actual values on y axis and set them as average and not sum.
4. Head to filters, for the specific visual as its timestamp column from table to the filter field. Set the filter type as Top N, show items as Top, set the number of items as per visual, again add timestamp column to by value field and set it to latest.
5. Apply the filter



6. Apply the data and same logic to all the other visuals to keep them updated by time.

Head over to format your report page and enable the page refresh. Set the refresh time as per requirements.

Demo dashboard created by me.



Save the report and enable the data logging through KEPServerEX, with auto refresh enabled the dashboard in Power BI should now update on its own with new values.

You can now publish this report with lower refresh rate with premium account or with higher with free account.

Conclusion

This project demonstrates the effective integration of KEPServerEX, SQL Server Express, and Power BI to create a real-time data logging and visualization system, embodying the principles of the Internet of Things (IoT). By seamlessly connecting industrial devices for continuous data capture, efficient storage, and dynamic visualization, the project enables proactive monitoring and data-driven decision-making. The real-time dashboards in Power BI provide instant insights into industrial processes, highlighting the transformative power of IoT in modern industrial applications. Future enhancements could include advanced analytics, improved scalability, and enhanced security measures to further leverage the potential of IoT in industrial data management.

References

- KEPServerEX documentation: <https://support.ptc.com/help/kepware/kepserverex/en/>
- SQL Server documentation: <https://learn.microsoft.com/en-us/sql/sql-server/what-is-sql-server?view=sql-server-ver16>
- Power BI documentation: <https://learn.microsoft.com/en-us/sql/sql-server/what-is-sql-server?view=sql-server-ver16>
- KEPServerEX data logging to SQL Server tutorial: <https://www.youtube.com/watch?v=SYe3wM6y8LQ>