Benefits of Using the Right Development Tools for IoT Device Data Management

Sasan Montaseri, CEO, ITTIA IoT Factory

Ganesh Balamitran, Product Marketing Manager, Industrial Business Unit, Renesas Electronics Corp.

October 2018

Abstract

Renesas selected ITTIA to independently demonstrate the benefits of using the right software and hardware technologies for Internet of Things (IoT) data management to realize the future and full potential of data. After several months of research, investigation, and collaboration, our results are reflected and shared in this white paper.

We share the data management and connectivity challenges and options faced by manufacturers of embedded IoT edge devices. Our approach for suggesting a general solution is broad-based, and we developed a software demonstration to verify and share the process and results.

Introduction

The Internet of Things (IoT) represents a new approach to hardware and software design where physical devices connect to exchange data and adapt to real-world problems. Developers and device manufacturers continuously search for ways to apply IoT for diverse applications such as smart energy, manufacturing, and transportation. Integrated sensors produce a wealth of IoT data, but managing data and extracting value is a challenge when information is distributed across different devices. To achieve distributed intelligence, developers and manufacturers must use integrated technologies with proven interoperability.

A verified IoT platform includes all the components required for a group of devices to collect data, share data, make decisions, and act. Decisions made by direct machine-to-machine interaction or through a human-machine interface should be supported by real-world data and machine learning. To keep up with the amount of data available and maintain privacy requires a data management solution that can store and query information directly on IoT edge devices. Manufacturers building industrial automation, medical devices, power grid, and transportation systems all need to find ways to harness the incredible power of device data management and connected computing. But this new era introduces a hidden risk. The challenge is that many of the component technologies focus on a limited aspect of a general platform.

This white paper will introduce an integrated and verified solution for software and hardware development of IoT edge devices. The goal of ITTIA's IoT Data Factory is to help customers benefit from the existing verified solution for IoT and embedded system development.

IoT Device Data Management Challenges

Internet of Things data is changing the way we live our lives. From vehicles and buildings, to roads and cities, everything around us is increasingly becoming controlled by IoT devices and sensors. Sophisticated sensors and computer chips are embedded in objects that each collect and transmit valuable data. Whether monitoring temperature, moisture, pressure, light, or other data, sensors continuously produce new data. But how do you identify the best integrated technologies that will work together and leverage large quantities of data?

For example, an automobile has countless components that can be monitored by IoT sensors. With access to this data, machine learning and similar analysis techniques help to identify potential problems with the engine or electrical system and schedule preventative maintenance. Automotive IoT enables the vehicle to perform real-time updates and perform diagnoses to recognize the seriousness of a problem, contact a service center for an inspection, and share analytical data with the automobile manufacturer to improve the design and production of thousands of other vehicles at the factory.

In another example, an industrial production line must coordinate the actions of multiple pieces of machinery, converting raw materials into a finished product. Industrial IoT uses distributed intelligence to improve traceability and efficiency by tracking production at a high level of detail. An edge gateway device collects data from nearby sensors, filters it if necessary, and the most relevant information is stored and shared with other systems. Collected data is then used to precisely identify the source of raw materials in each individual manufactured good.

In the electric power industry, it is important to balance power consumption with power generation to minimize energy costs for consumers, encourage use of sustainable energy sources, and achieve high reliability. Monitoring energy usage throughout the power grid to make efficient use of energy storage solutions is an essential goal.

Building an embedded system depends on in-house expertise and an awareness of available solutions. Whether you are building large deployment systems or a few hundred expensive systems using COTS (commercial-off-the-shelf), reinventing data management from scratch is too great a risk. Even with internal database expertise, time and budget are generally limited. Accessing experience and an available verified solution will save many months of development and integration. As a result, manufacturers effectivity optimize the total cost of ownership and bring products to market on time.

In all these scenarios, a common IoT data management platform is required to store and query/analyze data produced by sensors, without relying on centralized storage to bring diverse information together. A connectivity solution must enable devices to publish current information to any number of subscribers with a variety of network configurations. The core operating systems, programming languages, and other development tools provide the foundation for distributed computing. Hardware for IoT devices ranges from low-powered microcontrollers to fully-functional embedded computers. In some cases, a dedicated human machine interface is required to report the status of other IoT devices, provide detailed insight, and offer controls.

Exploring Solutions for IoT Device Data Management

To understand how IoT data management technologies are applied to solve real-world problems, consider the electric power grid. The grid connects homes, businesses, and other buildings to central power sources, which allows us to use electronics, appliances, and heating/cooling systems. But when parts of the grid fail or need to be repaired, everyone in the area is affected. Energy costs are high during peak times of the day. Microgrids help to mitigate these problems by decentralizing energy generation.

A microgrid is a local energy grid that can be disconnected from the traditional grid and operate autonomously. When functioning as an island, a microgrid can be powered by renewable resources like solar panels, wind turbines, energy storage systems such as batteries, or fuel-based generators. This

makes it possible for a home or business to operate using its own local energy generation, both in times of crisis, such as power outages, and to reduce energy costs.

Like other potential applications of IoT technology, microgrids use a variety of physical devices that can collect data and communicate with each other, such as power meters, solar generators, and batteries. When connected to a traditional power grid, it is also useful for sharing information with the utility company, such as the best times to use battery power. The data management process starts with a device that facilitates communication with the common Internet of Things data management platform and makes all software components work together.

To create a microgrid, software must be developed to monitor power load and ensure that energy production meets current demand. Energy storage systems, such as batteries, should be used intelligently to maintain a stable power supply. This requires an awareness of how energy is consumed, both time of day and category, in the same way that an automobile must monitor wear and tear on various parts or a factory must monitor production rates. A system is required that can store and query data across multiple dimensions.

Edge Devices: Components of a Complete IoT System

Consider a smart home microgrid that oversees all components inside a house and offers serious potential energy and cost savings. It may also be extended to offer other important information to the owner of the house, such as when an appliance is out of warranty. However, we will leave this feature to other developers to add as we make the source code for the demo available. The demo implements the following components:

- Solar power generator
- Battery energy storage system
- Heating/cooling energy meter
- Appliance energy meter
- Electric vehicle charging energy meter
- Power history recording system
- A human machine interface (HMI) for monitoring and control
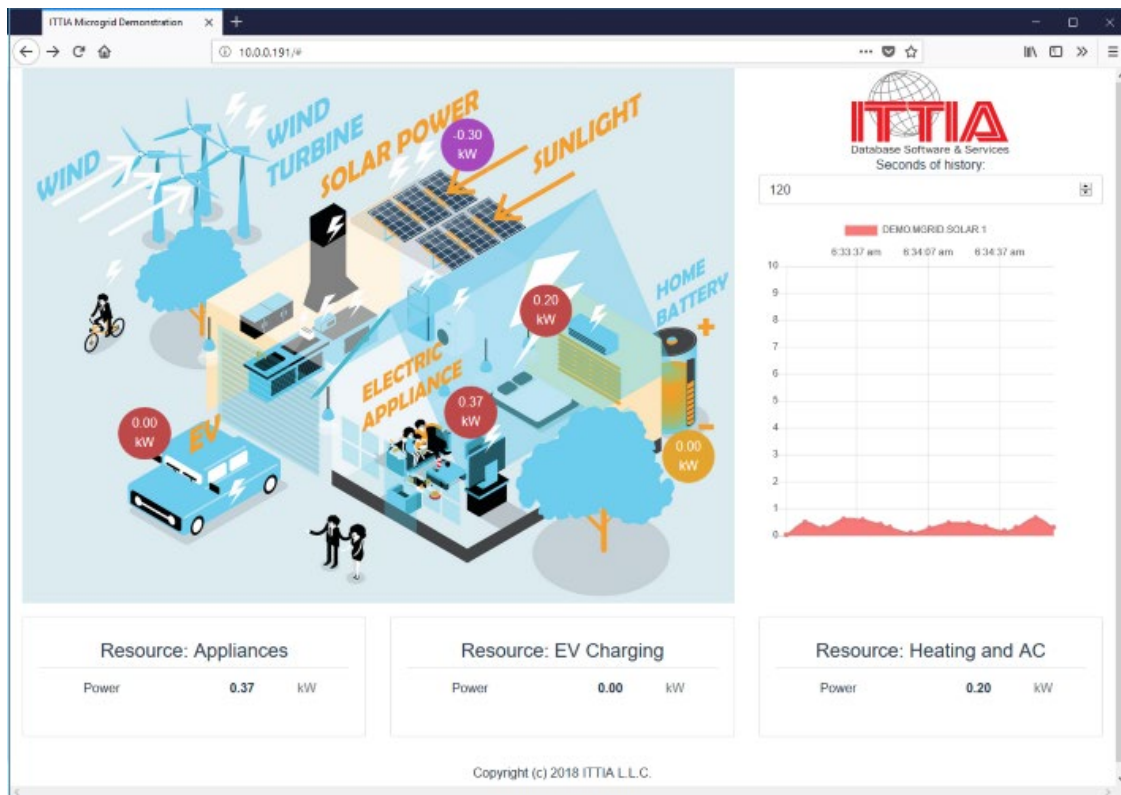
ITTIA Data Management Microgrid Demonstration

In this scenario, the solar power generator supplies energy during the day, and the battery supplies energy at night. Meters measure each different category of power load, which is continuously recorded in a historical database. The status of the system and history is summarized and displayed through the HMI component.

To implement this scenario, you would need multiple devices capable of power measurements, a wired or wireless network, and a touch display. The ITTIA Data Management Microgrid Demonstration is a collection of C++ programs that implement a simulation for each of these components. This demonstration uses:

- The Renesas RZ/G high-processing-performance dual-core Arm® Cortex®-A7 CPUs, with 3D graphics and video codec engine
- ITTIA DB SQL embedded database software
- RTI Connext DDS connectivity software
- Embedded REST web server
- Javascript web client
- iWave SOM Development Platform
- The RZ/G Linux® Platform
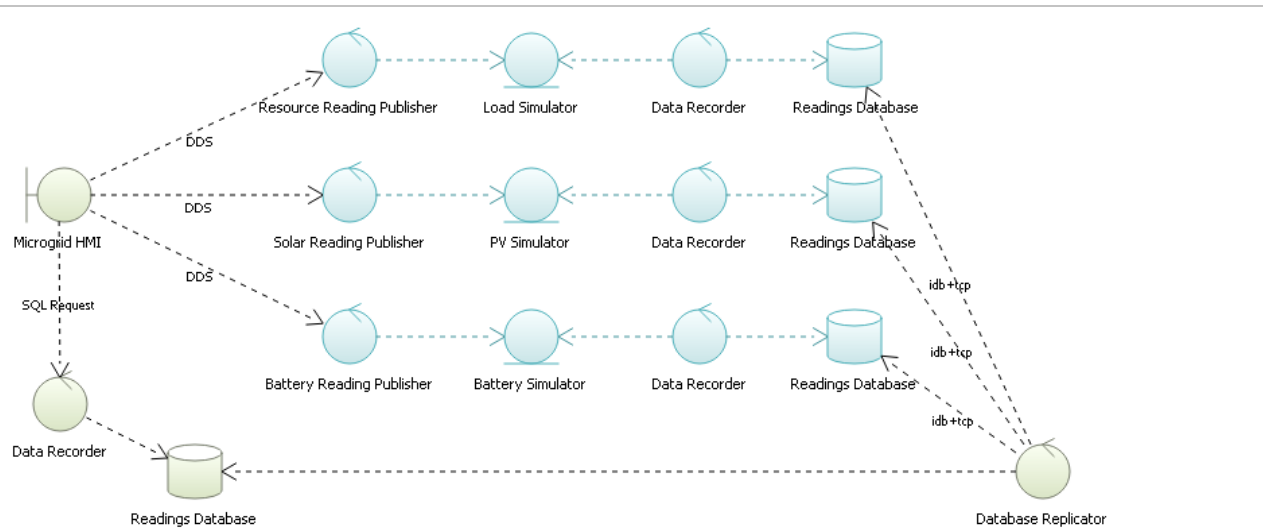
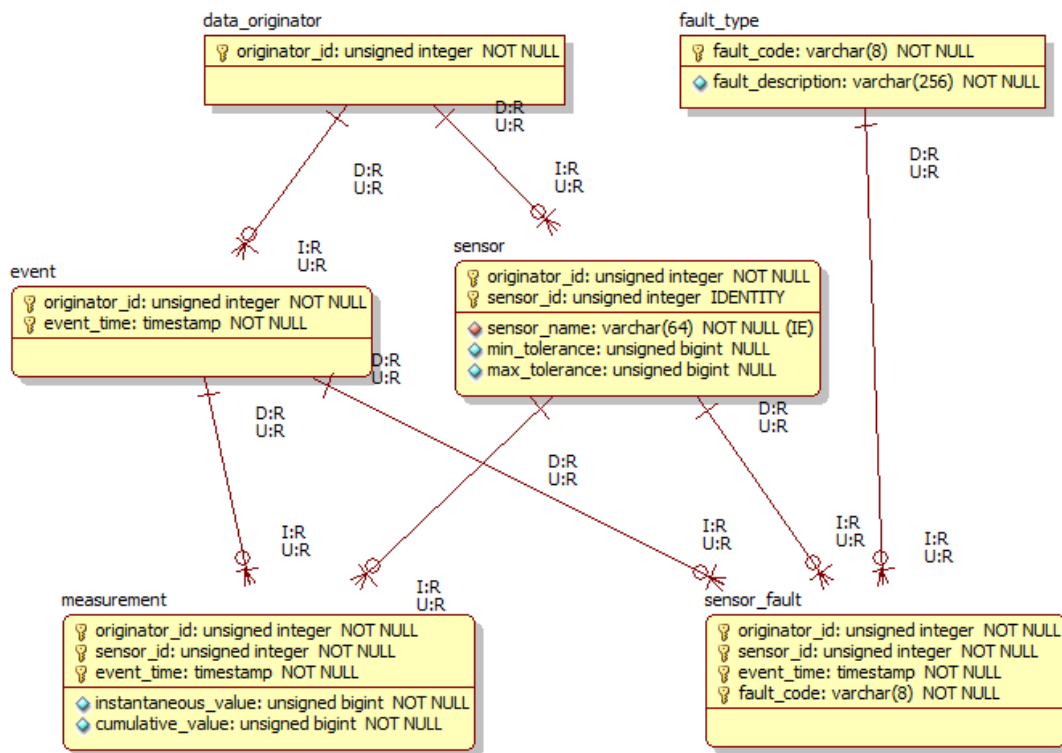*ITTIA Data Management Microgrid System Design*

Capturing IoT Sensor Data

The microgrid demonstration simulates data for a solar power generator and three energy-consuming resources: heating/cooling, appliances, and an electric vehicle charging station. Each simulated meter produces a power value measured in kilowatts. Negative values are used for power generation, while positive values represent an energy-consuming load.

The standard set by the Data Distribution Service (DDS) for Real-Time Systems enables scalable, real-time, dependable, high-performance, and interoperable data exchanges using a publish–subscribe pattern. To communicate with other components, each meter publishes the current power value to a DDS topic. Other components on the same network only need to subscribe to this topic to receive updates whenever the power value changes. This data-centric approach to communication allows each meter to be deployed to a different device without the need to configure a central server or broker to receive the data.

Recording Data with Machine-to-Machine

To record a history of energy production and consumption data, the ITTIA Data Management Microgrid Demonstration includes a power recorder program. A relational SQL embedded database system, tightly integrated with the microgrid demo application, stores a large volume of data on flash storage media. The database is hidden from the end-user and requires no ongoing maintenance or database administration.

The power recorder subscribes to the DDS topics for the solar generator and resources, periodically saving the current power value for each active meter in a database table. Data that rarely changes, such as the name and description of each meter, is only saved once. The recorder also computes and stores a cumulative value for each measurement that can be used to find the average value between any two timestamps.

data_originator

- originator_id: unsigned integer NOT NULL

fault_type

- fault_code: varchar(8) NOT NULL
- fault_description: varchar(256) NOT NULL

D:R
U:R

D:R
U:R

D:R
U:R

I:R
U:R

I:R
U:R

event

- originator_id: unsigned integer NOT NULL
- event_time: timestamp NOT NULL

sensor

- originator_id: unsigned integer NOT NULL
- sensor_id: unsigned integer IDENTITY
- sensor_name: varchar(64) NOT NULL (IE)
- min_tolerance: unsigned bigint NULL
- max_tolerance: unsigned bigint NULL

D:R
U:R

D:R
U:R

D:R
U:R

D:R
U:R

I:R
U:R

I:R
U:R

I:R
U:R

I:R
U:R

I:R
U:R

measurement

- originator_id: unsigned integer NOT NULL
- sensor_id: unsigned integer NOT NULL
- event_time: timestamp NOT NULL
- instantaneous_value: unsigned bigint NOT NULL
- cumulative_value: unsigned bigint NOT NULL

sensor_fault

- originator_id: unsigned integer NOT NULL
- sensor_id: unsigned integer NOT NULL
- event_time: timestamp NOT NULL
- fault_code: varchar(8) NOT NULL

Sensor History Database Schema

The power recorder can be deployed on the same device as the power meters or on a separate device with a reliable network connection. By storing a complete history of power measurements in a local database, this system reliably maintains a complete historical record with high accuracy. By querying the standard database, it is possible to extract a small number of records, at specific times, for analysis or data sharing.

Recording data to a text file or other flat file format would limit the quantity of data that can be stored and retrieved. With a large flat file, it is only possible to communicate the entire file, in whole or divided into parts. Relational database tables are indexed, so they can be filtered and sorted efficiently.

Recording data to a central database on a different network is also a risk because sensor data is lost when the microgrid is disconnected. Recording to a local database ensures that meter values are still collected even when the microgrid is completely disconnected from the outside world. A local database keeps data private by limiting the amount of information that is shared with others. This is especially important in other applications where personal data is collected by IoT devices.

Interfacing Humans and Machines

The Human Machine Interface (HMI) provides a user interface for interacting with machines. The ITTIA Data Management Microgrid Demo implements a web-based HMI that combines the current real-time status of the microgrid from DDS topics with the full history from the power recorder.
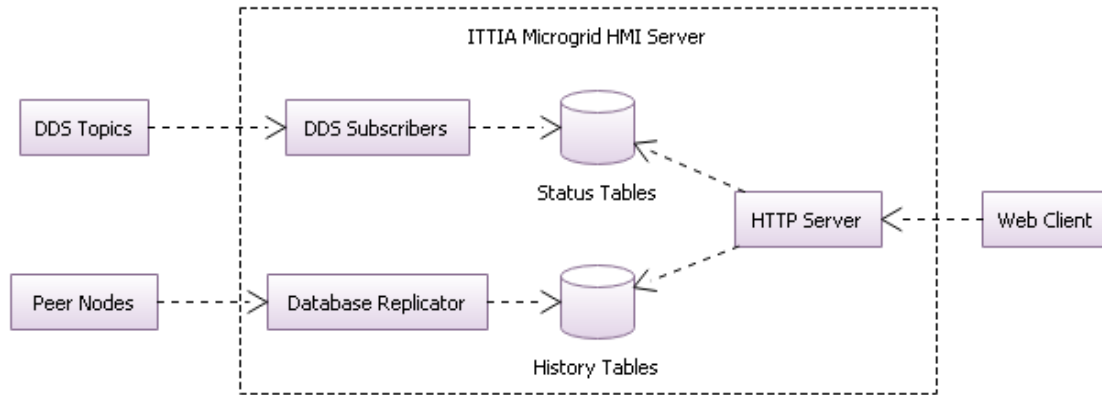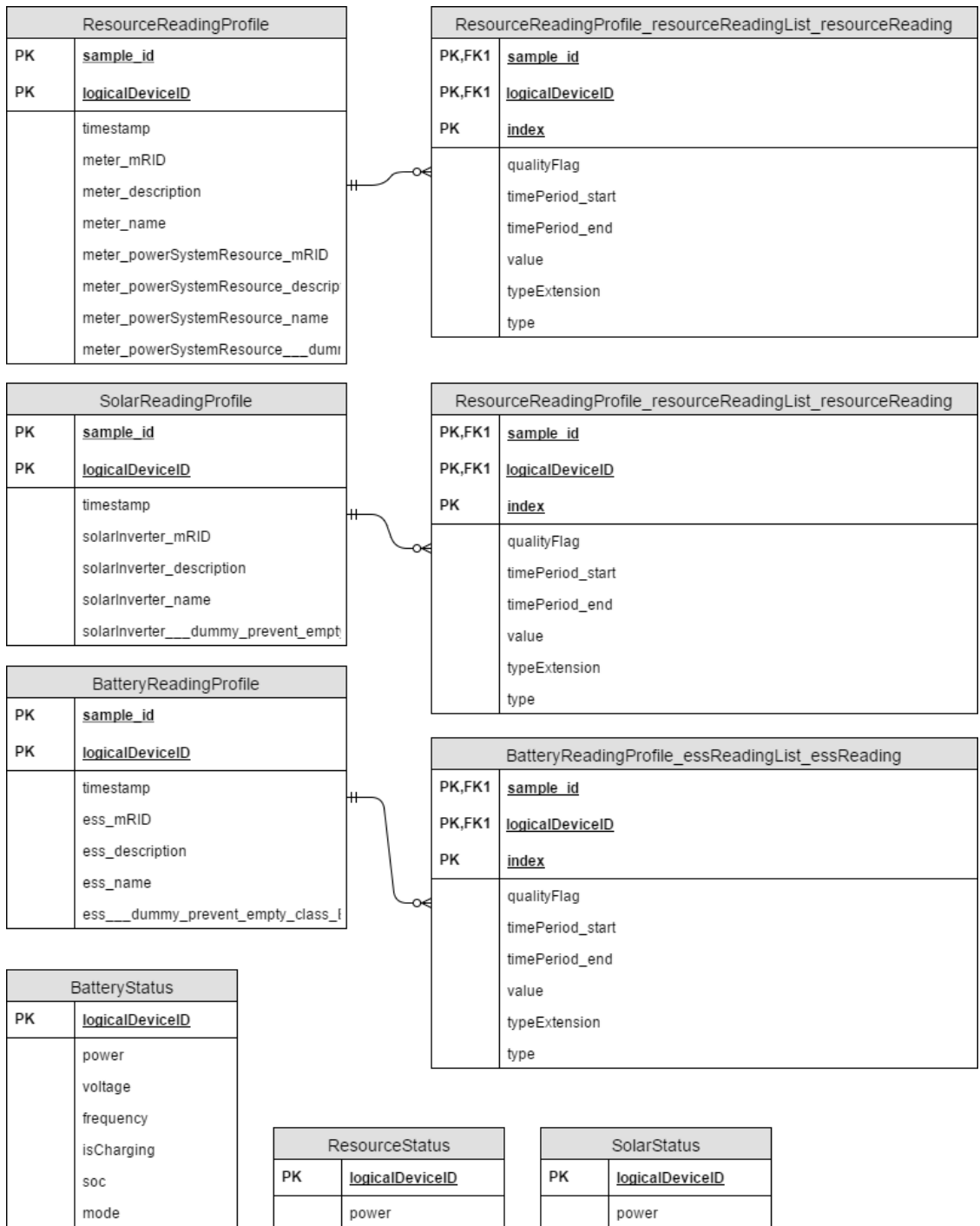
HMI Components

Status data is used to show which components are connected to the grid, how much power is being used, and other details. The HMI should always have access to the status of other devices to show an accurate picture of the microgrid. With ITTIA DB SQL, the HMI server application queries the database with local SQL queries. These can:

•   Search for power values over a time range
•   Aggregate power usage by time of day

The HMI component uses an embedded REST web server to respond to requests for current data from a web client. Requests, which can be received at any time, are handled by a pool of worker threads that must synchronize their access to the current data status. At the same time, the HMI component must subscribe to DDS topics and handle updated power values as they become available. These separate tasks need a place to store the current values that is thread-safe and easy to query. For that reason, the DDS subscriber updates the current sensor values stored in memory tables, which are concurrently queried by REST requests.

ResourceReadingProfile

PK	sample_id
PK	logicalDeviceID
	timestamp
	meter_mRID
	meter_description
	meter_name
	meter_powerSystemResource_mRID
	meter_powerSystemResource_descrip
	meter_powerSystemResource_name
	meter_powerSystemResource___dumi

ResourceReadingProfile_resourceReadingList_resourceReading

PK,FK1	sample_id
PK,FK1	logicalDeviceID
PK	index
	qualityFlag
	timePeriod_start
	timePeriod_end
	value
	typeExtension
	type

SolarReadingProfile

PK	sample_id
PK	logicalDeviceID
	timestamp
	solarInverter_mRID
	solarInverter_description
	solarInverter_name
	solarInverter___dummy_prevent_empty

ResourceReadingProfile_resourceReadingList_resourceReading

PK,FK1	sample_id
PK,FK1	logicalDeviceID
PK	index
	qualityFlag
	timePeriod_start
	timePeriod_end
	value
	typeExtension
	type

BatteryReadingProfile

PK	sample_id
PK	logicalDeviceID
	timestamp
	ess_mRID
	ess_description
	ess_name
	ess___dummy_prevent_empty_class_l

BatteryReadingProfile_essReadingList_essReading

PK,FK1	sample_id
PK,FK1	logicalDeviceID
PK	index
	qualityFlag
	timePeriod_start
	timePeriod_end
	value
	typeExtension
	type

BatteryStatus

PK	logicalDeviceID
	power
	voltage
	frequency
	isCharging
	soc
	mode

ResourceStatus

PK	logicalDeviceID
	power

SolarStatus

PK	logicalDeviceID
	power

Microgrid Status Database Schema

In this schema, the ResourceReadingProfile stores samples for a heating/cooling system, appliances, and electric vehicle charging. The SolarReadingProfile stores samples for the solar power generator. The BatteryReadingProfile stores samples for the battery power usage, voltage, and state of charge.

The Microgrid HMI is also able to query the full history of sensor measurements collected by the power recorder component. So that the HMI and power recorder can operate independently, each program keeps its own local copy of the history tables. Database replication is used to send data from the power

recorder to the Microgrid HMI. This ensures that the power recorder is never overwhelmed by a large number of requests, which could impact its ability to record new data. At the same time, the HMI can accumulate data from multiple power recorders, enabling this solution to scale up to support a greater number of power meters.

ITTIA DB SQL is an embedded database library, so there is no separate program to deploy for the database. A database replication server is run in the HMI to accept connections from the power recorder program.

Benefits of the Right Software Development Tools

The complexity of IoT integration often surprises integration managers as costs and development budget rapidly spiral out of control. To prevent disappointment, developers and managers can address the diverse integration requirements for IoT data management by selecting verified tools. There are important factors behind the IoT device data management platform, including interoperability, scalability, security, and standards offered by software technologies selected to build IoT products.

ITTIA DB SQL is a relational database management software library for embedded systems and intelligent IoT devices. It is a full-featured, robust data management technology that scales down to meet the constrained resource requirements of embedded systems. It simplifies data management for developers with SQL features not typically available on an embedded device. Elegant APIs, tools to ease development, and smooth upgrades make it fun to build applications, independent of an operating system and hardware.

ITTIA DB SQL offers benefits in its modular architecture that enable customers to create highly optimized and reliable systems with low total cost of ownership. With ITTIA DB SQL, manufacturers build robust, high-value products, with no database administrator, to store, manage, analyze, connect, and distribute data. Libraries support C, C++, and various scripting languages
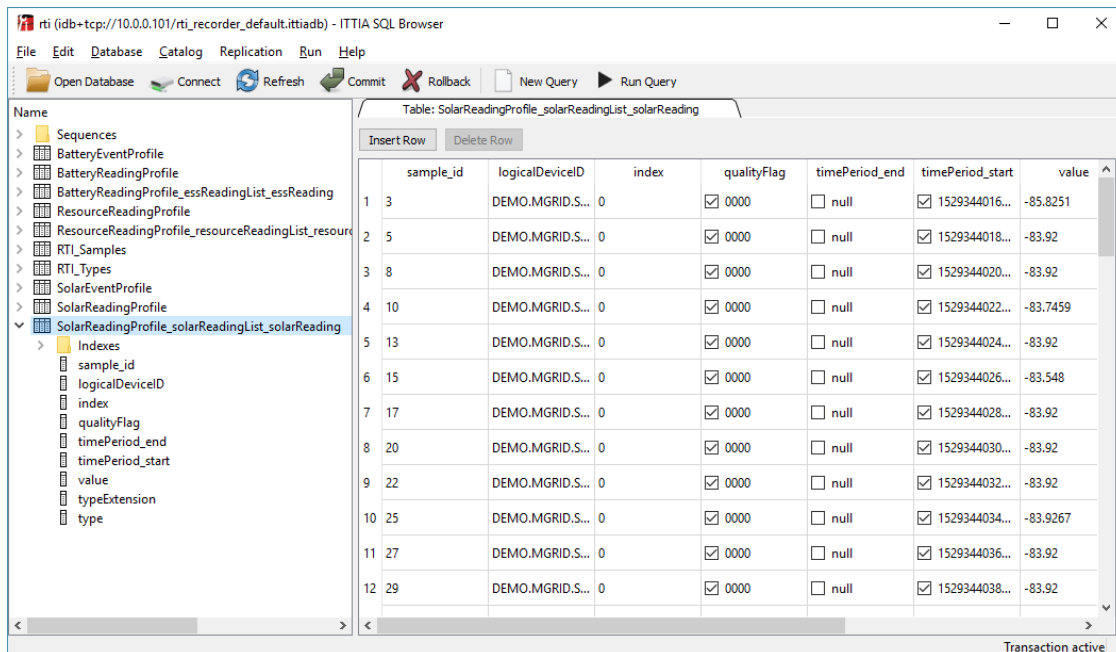
ITTIA SQL Browser

RTI Connext DDS software includes an implementation of the DDS standard. DDS is an open standard for messaging that supports the unique needs of real-time systems. Its open interfaces and advanced integration capabilities slash costs across a system's lifecycle, from initial development and integration

through on-going maintenance and upgrades. RTI Connext® DDS is a reliable connectivity framework designed for the demanding requirements of the Industrial Internet of Things (IIoT).
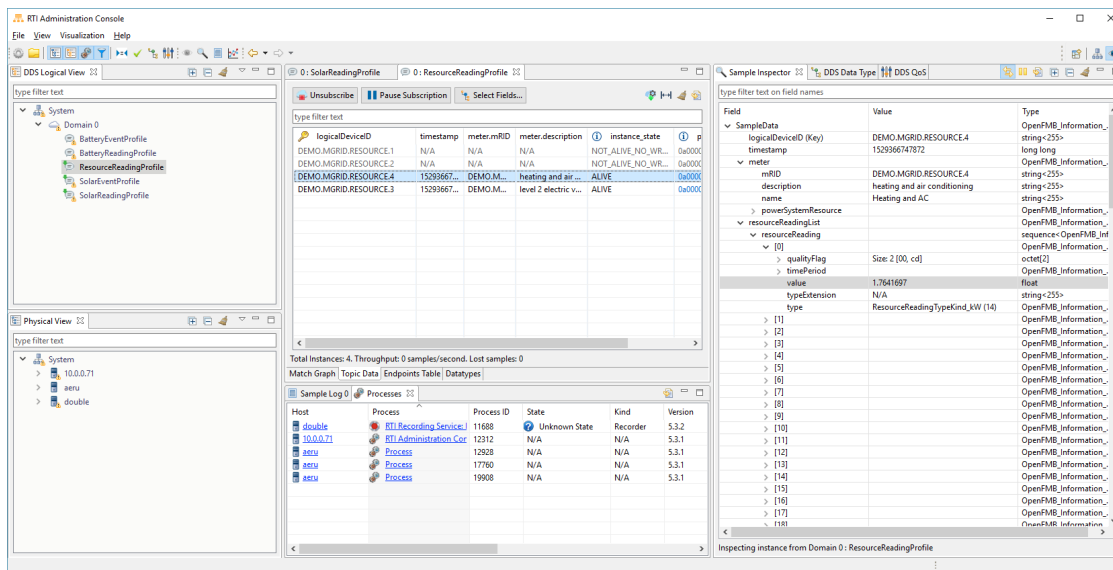
RTI Administration Console

The Renesas RZ Family of ARM®-based microprocessors (MPUs) enables manufacturers to implement high-resolution human machine interfaces (HMI), embedded vision, and real-time Industrial Ethernet connectivity. The RZ Family is easy to use and designed to accelerate time to market, reduce total cost of ownership, and remove many of the obstacles engineers face when designing Internet of Things (IoT) and other embedded applications. Software development tools that accompany the RZ Family include operating system, database, which gets silently embedded into the application, and data distribution software. For more information on Renesas RZ microprocessors, please visit https://www.renesas.com/us/en/products/microcontrollers-microprocessors/rz.html

Renesas RZ/G Series microprocessors support the Civil Infrastructure Platform® (CIP) Linux from the Linux Foundation. CIP Linux provides excellent long-term support for specific Linux kernels. This long-term support extends for 10-15 years, which is ideal for industrial platforms with very long lives. For more information on CIP Linux, please visit https://www.cip-project.org/ For information on the RZ/G Linux Platform, visit https://www.renesas.com/us/en/products/rzg-linux-platform.html

iWave's SOMs offer unique advantages for processor scalability and flexibility. iWave SOMs using Renesas RZ/G processors offer CIP Linux. The operating system, database storage, and connectivity software work together on compatible hardware to manage sensor data, make decisions, and present information. While the format of the data may change, using a common data model through each component greatly simplifies the design and development of an IoT system. ITTIA DB SQL and RTI Connext DDS both take a data-centric approach, making it easy to store and connect data at the same time in embedded C and C++ applications. For more information on iWave SOMs based on Renesas RZ/G, please visit https://www.iwavesystems.com/product/cpu-modules/renesas-rz-g-som-modules-sbc.html

Conclusion

In this paper, how selected technologies work together to solve microgrid problems were described. One can see how these technologies could be easily adapted to any IoT scenario where there is data to be

managed. The ITTIA Microgrid Demo solved a specific problem, but the techniques can be applied generally to any application.

When it comes to building new products, most of the data management and the analysis cycle are spent on developing and setting up hardware and software components. Information from various market studies confirms that most projects do not release on time because the integration task is unexpectedly complex and manufacturers experience long delays. The majority of time is spent on prototyping, rather than designing, testing, and debugging. In addition, when the project nears its end, problems such as performance, lack of data integration, and waiting to receive support from a community can cause additional delay. Therefore, manufacturers face complex system integration challenges to market products on time.

While the new era of IoT is revolutionizing innovation, ITTIA IoT Factory is a combination of qualified third-party technologies that are already tested and verified. Consequently, developers save time and can quickly start development. The third-party solution includes RTOS, middleware, connectivity software, user interface, processors, controllers, and more.

 Access ITTIA IoT Data Factory to support this integration and as a first point of contact to answer your questions. http://www.ittia.com/iot-data-factory

Technical Takeaways

- Integration of software and hardware solutions, standards, and approaches for the IoT data device journey includes collecting, managing, and analyzing data locally. When there is no connection, devices must still gain insights and make decisions and prepare for data distribution.
- With the right verified software and hardware technologies, prototyping IoT devices to manage, share, and distribute data offers tremendous development time, cost, and resource savings. With mature and proven solutions, the device data management complexity is reduced.
- A data management framework is essential. Embedded systems work with data from various origins, and each system needs the right hardware and software to manage and share information.

Business Takeaways

- Meeting business IoT initiatives should include a coordinated hardware, data management, data integration, and operating system vendor strategy.
- Data, data management, and data integration directly affect bottom lines, which is a critical aspect of any business venture. Utilizing available commercial solutions tremendously reduces development costs.
- Complex IoT systems consist of many individual products, hardware, and software. Although each of them works separately, a total system may behave in unexpected ways. Verified solutions can help identify problems early on and validate the performance of the overall system by analyzing the different layers of technology needed for successful deployment. This approach offers tremendous savings in the integration and verification of each technology.

Accessing ITTIA Data Management Microgrid Demo Source Code

The source code and Getting Started Guide for this demonstration are available at https://www.ittia.com/demos

Legal Disclaimers

ITTIA Disclaimer

Information in this document is provided solely to enable system and software implementers. No express or implied copyright license is granted hereunder to design or implement any database management system software based on the information in this document.

Renesas Electronics Disclaimer