Automotive Smart Data Access Demonstration Technical Information

Link to Press Release (January 5, 2022)

Contents

Introduction and market motivation

Data will fuel future vehicle innovations as the automotive industry shifts focus from horsepower to compute power. This transformation has the potential to deliver valuable vehicle insights and drive new data-driven service revenue. Expanded access to real-time, vehicle-wide data, secure connectivity to cloud services and streamlined machine learning (ML) can accelerate the shift and enable intelligent vehicles that improve over their lifetime through remote updates.

NXP's S32 Automotive Platform family of processors provides the safety, security and reliability required by demanding in-vehicle applications. Two of NXP's flagship platforms, the S32G GoldBox Service-oriented Gateway reference design and the S32S GreenBox II xEV Propulsion Domain Controller development platform are leveraged in this demonstration. The S32G GoldBox supports secure in-vehicle networking as well as connectivity to the private / public cloud domains. The S32S GreenBox supports hard-real time and math intensive computation for managing safe vehicle 'start, stop, and steer' while maximizing vehicle range.

In addition, NXP brings key partners into the demonstration with AWS providing in-vehicle and cloud based services with AWS IoT Greengrass and AWS IoT Core respectively, aicas providing Smart Data Access functionality with their EdgeSuite products, and MathWorks providing model-based design technologies to run vehicle simulation and create range extension algorithms.

System overview

The demonstration system is shown below featuring the NXP GreenBox II and GoldBox platforms.

Figure 1 S32 Automotive Processors – Smart Data Access Solution

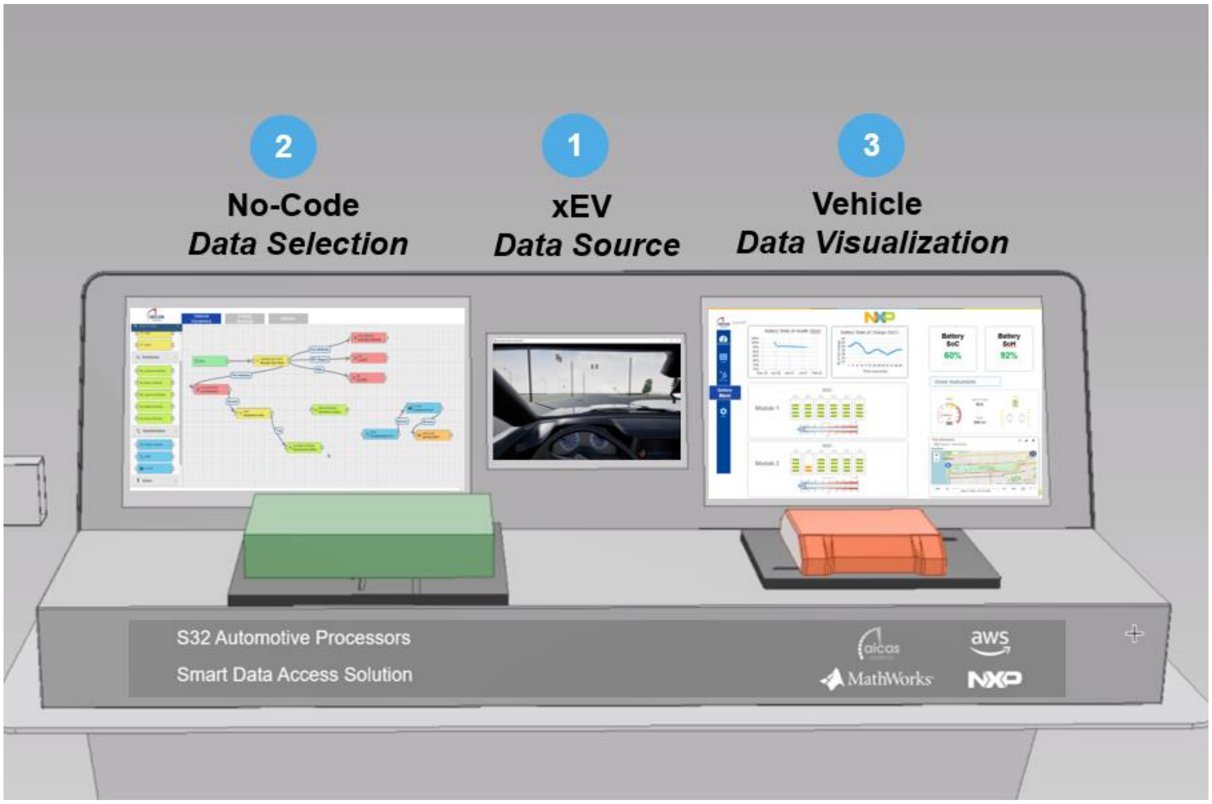


Figure 2 Smart Data Access with NXP S32S GreenBox and S32G GoldBox

The demo visualizes Smart Data Access to real-time vehicle data on the backside of the car with focus on Connected Propulsion Domain Control. Two screens will be used to *select* desired data for Vehicle Dynamics, Battery Management and Energy Management (Display 1) and to *visualize* the selected real-time data (Display 2). Stored driving simulation data will be used to show the ability to access selected real-time vehicle data remotely from connected vehicles using AWS IoT Greengrass edge runtime and aicas EdgeSuite for control and visualization allowing Smart Data Access without having to write software. A third display shows an animation of our test car driving around the Mcity test track using MathWorks model-based design software.

A high level schematic of the system is shown below, both with a conceptual view of the GreenBox → GoldBox → Cloud operation as well as a system level connectivity diagram.

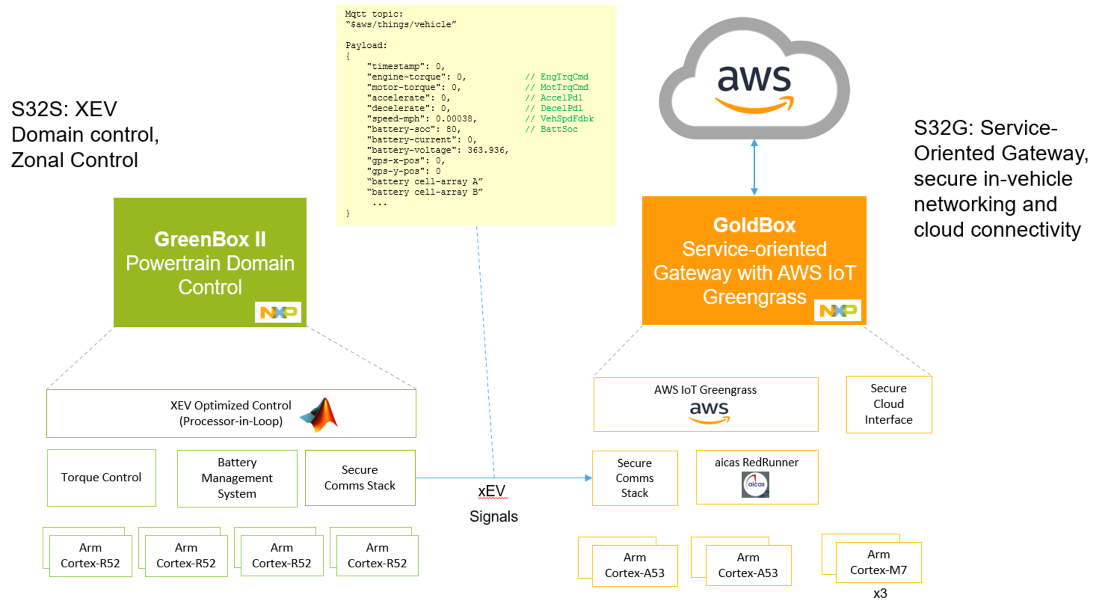
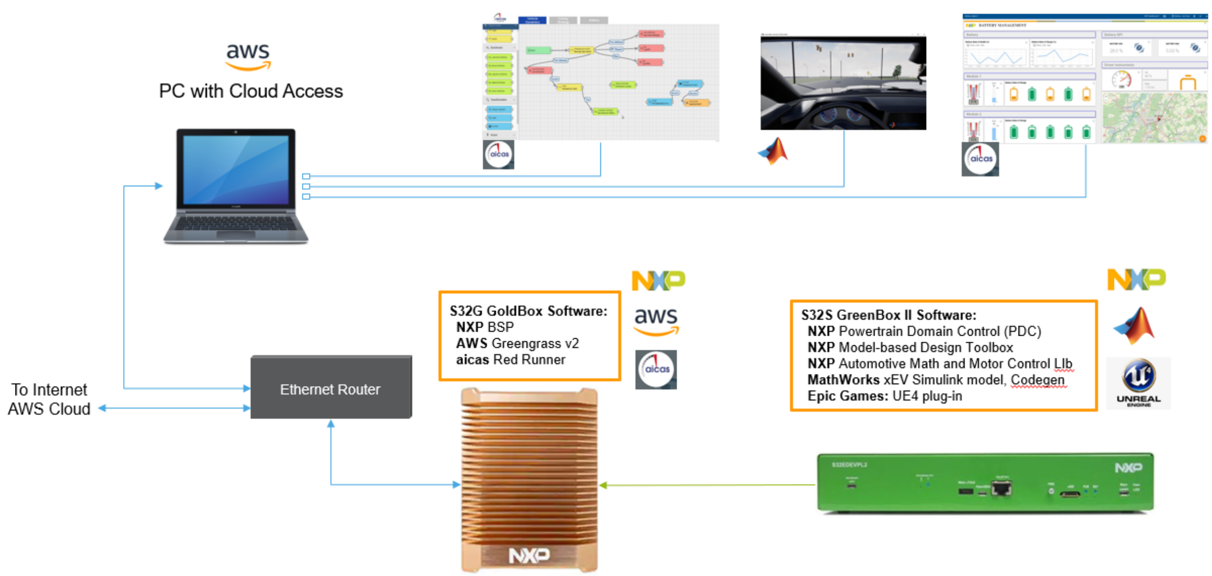


Figure 3 conceptual view of the GreenBox-to -GoldBox- to-Cloud operation



Signal walkthrough

S32S GreenBox II xEV Powertrain Domain Control (PDC)

A processor in the loop operation using MathWorks model-based design tools for vehicle simulation consumes vehicle sensor data and produces optimized vehicle control actions. Driver inputs such as acceleration, braking, and steering as well as vehicle model inputs such as speed, battery state of charge, and battery current are all utilized to feed an Optimizing Energy Management algorithm.

The key model-based design technologies utilized from the MathWorks include the Vehicle Dynamics Blockset, Powertrain Blockset, HEV P4 Simulink Model, and Simulink Coder. The NXP Model-Based Design Toolbox Add-On provides a development flow to easily migrate software to the NXP S32S MCU inside GreenBox II.

Vehicle data, available on Arm Cortex-R52 Core 2 is published to Arm Cortex-R52 core 0 in the S32S MCU. Software running on Core 0 consumes the data published via shared memory (shmem) and encapsulates it into MQTT packets for transport to the S32G GoldBox Service-oriented Gateway. The following diagram shows software running on the S32S GreenBox II. Notice the shared memory shmem channel between Core 2 and Core 0. The figure below shows program execution and data flow inside GreenBox II.

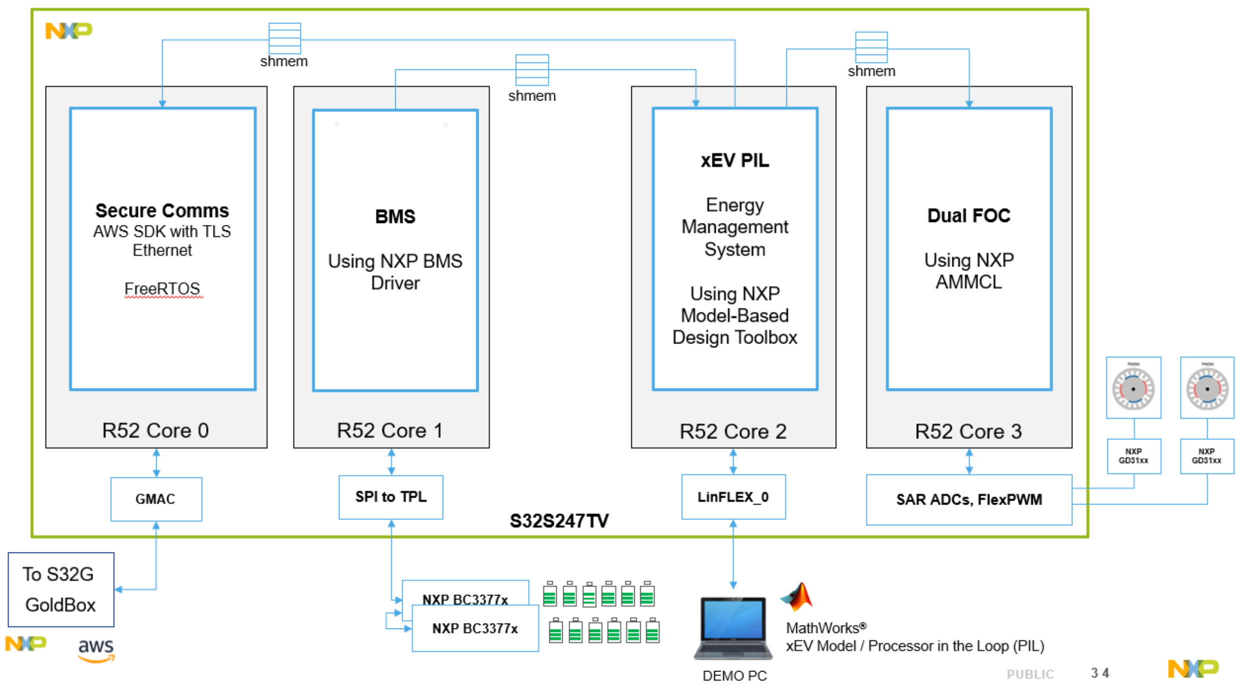


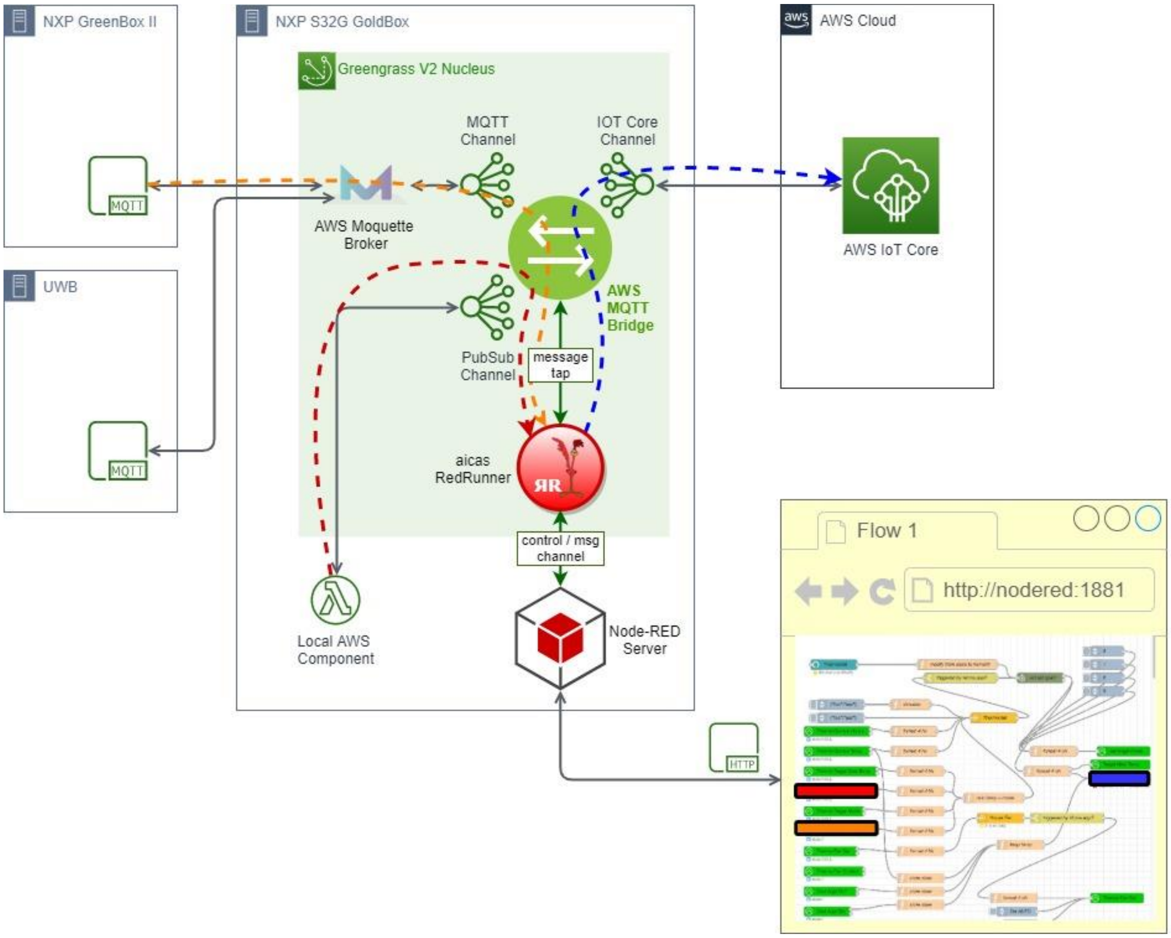
Figure 4 Powertrain Domain Control (PDC) software execution and data flow inside GreenBox II

S32G GoldBox Service-oriented Gateway

The S32G GoldBox Service-oriented Gateway provides secure in-vehicle connectivity, high-performance compute using multiple Arm A-class cores, real time embedded operation using Arm M-Class cores, and a wide array of vehicle connectivity.

Utilizing the high-performance compute capability of the Arm Cortex-A53 cores on the GoldBox, Greengrass V2 running on NXP's Linux BSP handles the flow of vehicle data from the GreenBox up to the AWS Cloud. Through the insertion of aicas' RedRunner into the AWS MQTT Bridge and Node-RED, the Node-RED server that is concurrently running in Linux BSP is able to collect messages from the GreenBox and insert them into the deployed Node-RED flow.

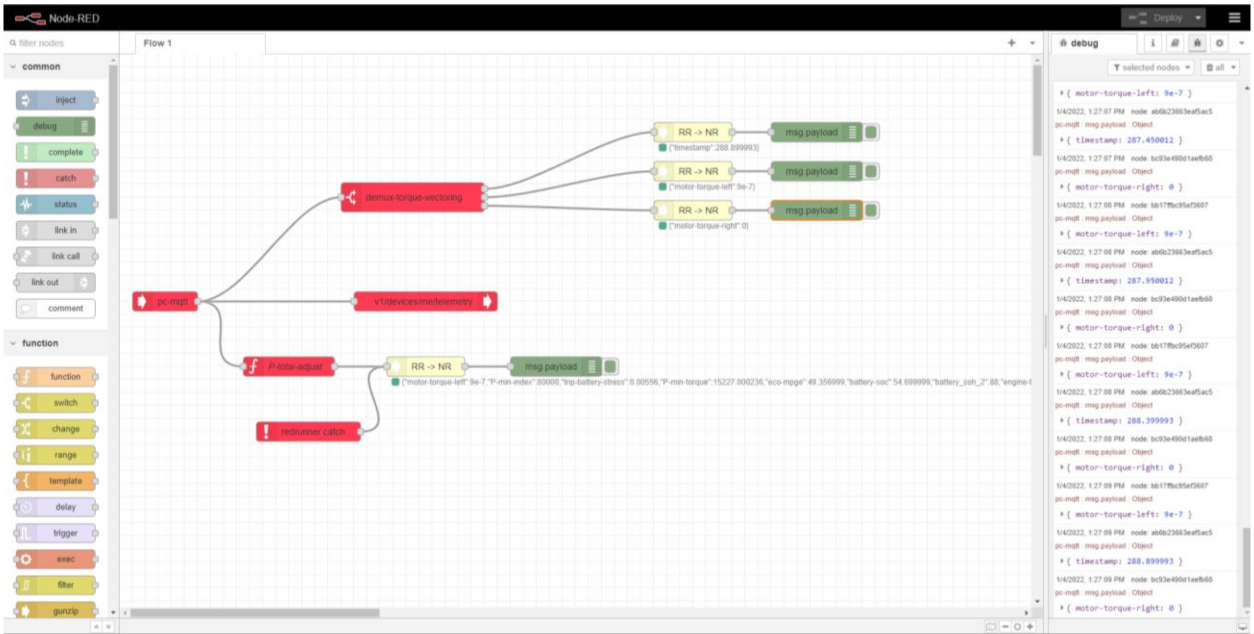
Once the data is processed in the user-defined Node-RED flow, it is then forwarded up to the AWS Cloud by using the same RedRunner message tap functionality that collected the original messages from the GreenBox in the first place. The figure below provides an overview of the data flow discussed where the GoldBox encapsulates the IoT Gateway.



Edge processing and signal selection

With the GoldBox acting as the central gateway between the in-vehicle's powertrain data and the AWS Cloud, edge processing through Node-RED and aicas' RedRunner provide controls to the user over what data is forwarded up to the cloud and when it is forwarded. By providing these controls, users can further reduce their data transmission and cloud costs and easily visualize their data on-device before it reaches the cloud.

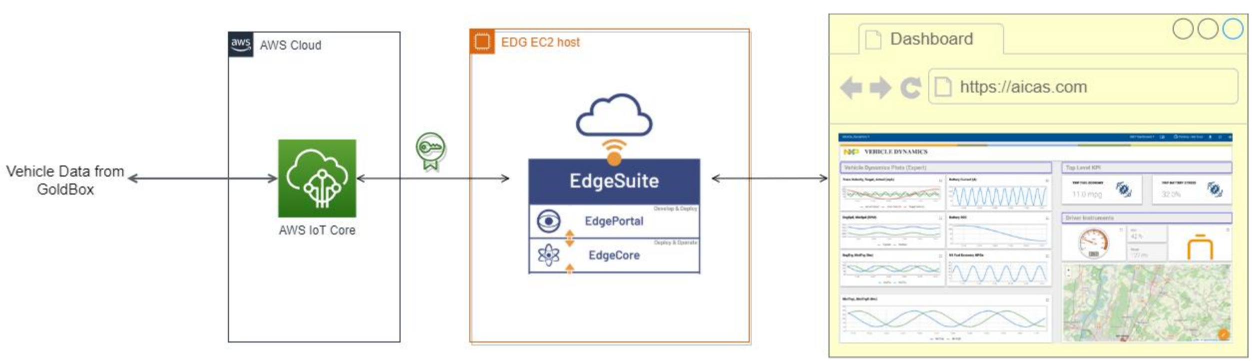
Utilizing the user-defined flows in Node-RED, a user is able to visually see how the data is processed on-device through connected nodes or blocks as shown below. This type of solution requires no code and allows the user to deploy new flows and configurations on-the-fly. Through the inclusion of aicas' RedRunner solution, additional nodes are added to Node-RED that complement the existing standard nodes by providing Greengrass support and allowing the user to define conditions that must be met before data is forwarded to the cloud.



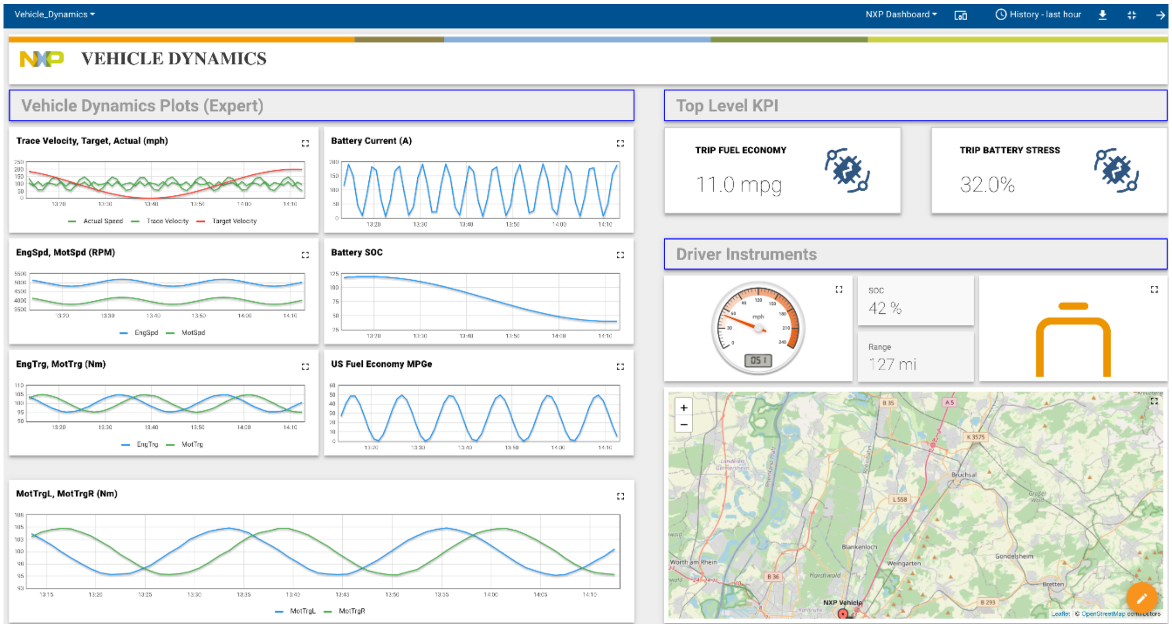
Cloud data movement

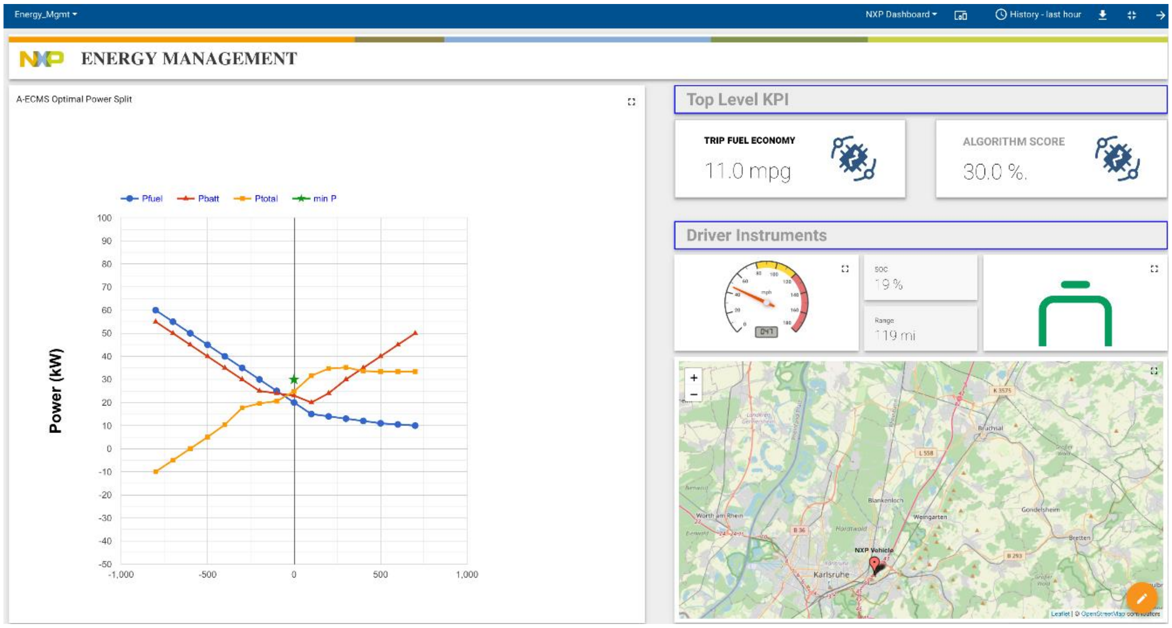
The EdgeSuite solution from aicas provides dashboards for viewing the real-time data of the vehicle. As a result of being hosted on an EC2 instance of AWS Cloud, the EdgeSuite dashboards can be accessed from anywhere in the world, allowing users to effectively monitor data from the vehicle.

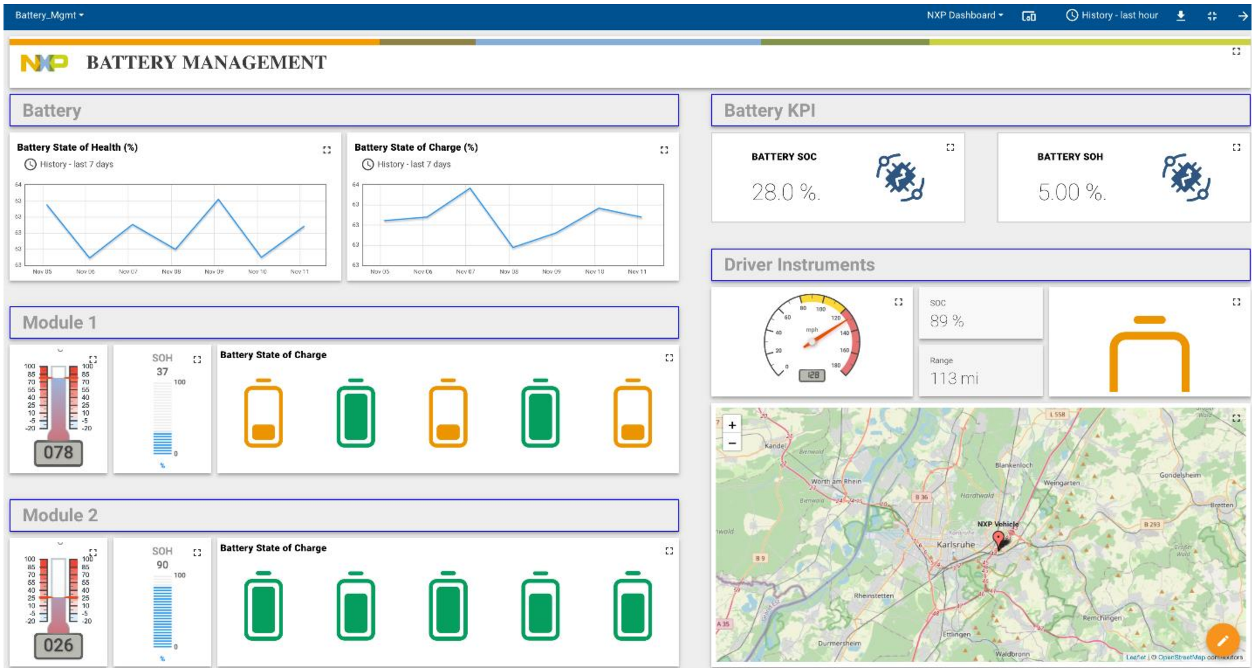
After data is published to the cloud by the GoldBox, EdgeSuite utilizes the endpoints provided by AWS IoT Core in order to subscribe to the data received from the vehicle as shown in the below diagram. Through this method, the data remains encrypted until it is received by EdgeSuite due to the secure MQTT connection. Once received, EdgeSuite will update the dynamic dashboards with the new data with limited latency in order to provide real-time diagnostic information about the vehicle.



Real-time data dashboards






Battery_Mgmt
NXP Dashboard
History - last hour
BATTERY MANAGEMENT
Battery
Battery State of Health (%)
History - last 7 days
Battery State of Charge (%)
History - last 7 days
Battery KPI
BATTERY SOC
28.0 %
BATTERY SOH
5.00 %
Driver Instruments
SOC
89 %
Range
113 mi
Module 1
SOH
37
078
Battery State of Charge
Module 2
SOH
90
026
Battery State of Charge
Karlsruhe
NXP Vehicle

Product list

aicas contributions

- EdgeSuite products

  aicas EdgeSuite simplifies how to develop, deploy and operate cloud-to-edge solutions. The family of products connects devices via edge to the cloud. It provides developers and data analysts full access and management of logic and data. It enables to manage full lifecycle of large fleets of devices or different IoT edge systems with ONE single, comprehensive while open system unifying your fleet of devices into a single environment.

- aicas JamaicaCAR
  JamaicaCAR enables automakers to efficiently deploy and manage in-vehicle applications for intelligent functions and connected services. Decoupling of hardware from software allows for lifecycle management as well reuse of applications across the entire fleet of brands and models of any automotive OEM. JamaicaCAR is one of many edge device runtimes of the aicas EdgeSuite.

AWS contributions

- AWS IoT Greengrass

  AWS IoT Greengrass is an open-source edge runtime and cloud service for building, deploying, and managing device software.

- AWS IoT Core
  AWS IoT Core is a managed cloud service that enables connected devices to securely interact with cloud applications and other devices.

NXP contributions

- S32G GoldBox
  The NXP GoldBox is a compact, highly optimized and integrated board engineered for vehicle service-oriented gateway (SoG), domain control applications, high-performance processing, safety and security applications.

- S32G GreenBox
  The NXP GreenBox is a development platform providing advanced performance, peripherals, and multi-core Arm® ecosystem for engineers to begin development on NXP's next generation of Electric Vehicle (xEV) and internal combustion engine microcontrollers.

MathWorks contributions

- Vehicle Dynamics Blockset
  Vehicle Dynamics Blockset™ provides fully assembled reference application models that simulate driving maneuvers in a 3D environment.

- Powertrain Blockset
  Powertrain Blockset™ provides fully assembled reference application models of automotive powertrains, including gasoline, diesel, hybrid, and electric systems.

- Simulink Coder
  Simulink Coder™ generates and executes C and C++ code from Simulink® models, Stateflow® charts, and MATLAB® functions.

Conclusion

aicas, AWS, MathWorks, and NXP unlock a new world of data-driven opportunities to provide easy access to deep vehicle data. The experts offer a solution for engineering, data scientists, product managers, fleet operations, mobility services, marketing, and many more is just around the corner.

The future of the automotive industry lies in software-defined vehicles. Vehicle data plays a crucial role for the success of future product development, operations, and business models, but it has not been readily available for the automotive industry to realize its full potential. Especially for data-driven vehicles, the challenges vary from having access to the vehicle-wide data, selecting the right data, having the ability to combine and process data in the vehicle and the cloud, easy access to specific data from any remote location, up to selecting and processing the data efficiently without needing programming skills.

The amount of vehicle data is enormous and continues to grow amid more sensors and software-defined vehicle data generation. Vehicle data is also extremely diverse. It is not feasible to send all the data from every vehicle to the cloud. The transmission and storage costs are too high.

aicas, AWS, MathWorks, and NXP have collaborated to realize the vision of having access to deep vehicle insights from across the fleet and being able to extend this to production vehicles. With this combination of expertise, the companies are able to showcase that this vision can be realized with a visual, no-code approach.

Appendix A: Signal table

A summary of the available signals is shown below. Each of the signals can be accessed by the S32G GoldBox Smart Data Access solution utilizing AWS Greengrass and the aicas RedRunner products.

To Do: add energy management signals.

Signal Group	Signal Name
Vehicle Dynamics Data	speed-mph
	engine-torque
	motor-torque
	battery-current
	battery-soc
	accelerate
	decelerate
Torque Vectoring Data	motor-torque-left
	motor-torque-right
Battery Data	battery_emulator_p1c1
	battery_emulator_p1c2
	battery_emulator_p1c3
	battery_emulator_p1c4
	battery_emulator_p1c5
	battery_emulator_p1c6
	battery_emulator_p2c1
	battery_emulator_p2c2
	battery_emulator_p2c3
	battery_emulator_p2c4
	battery_emulator_p2c5
	battery_emulator_p2c6
	battery-voltage
	battery_soc
	battery_soh_total
	battery_soh_1
	battery_soh_2
	module_temp_1
	module_temp_2
Map Data	gps-x-pos
	gps-y-pos
	Odometer

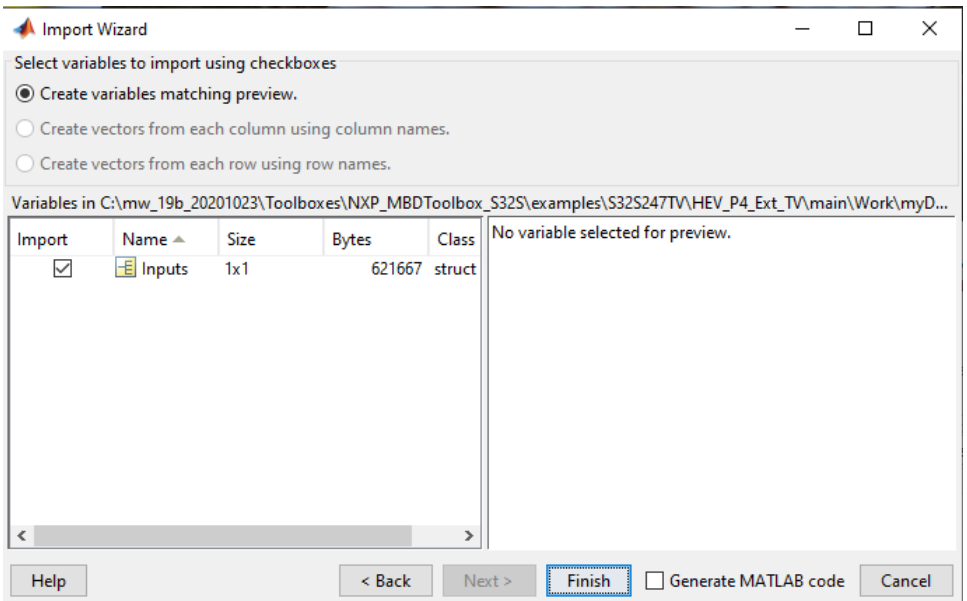
Appendix B: Simulink data capture information

When the test run completes:

Select "myDataInputs.mat", right click with the mouse

Select Import Data...

When the Import Wizard opens, select Finish



Overwrite the existing Inputs variable (click on OK).

Select the Parser.m script and execute it. It will start at the 101st timestamp which is 5.00 seconds into the simulation data capture (100 * 0.050 ms = 5 s).

Verify the Input_data.h file was created. Rename it to something specific for your test trial number.