



Design the next-gen user experience with Simulink and Qt Design Studio

MathWorks EXPO global conference 2021

Mahmoud Badri

May 2021

Next gen Automotive (**CASE**)

- › **C**onnected
- › **A**utonomous
- › **S**hared
- › **E**lectric



Industry trend for next gen UX in Automotive

- › Digital cockpits.
- › HMI / Voice are an integral part of modern cars.
- › Cars are becoming more connected.
- › Automakers are competing to bring the best UX to their cars' HMIs.
- › In-car systems that are as effective and user-centric as a consumer's mobile device.
- › Automotive UX influence buying decision as much as safety, comfort, ...etc.



Autonomous

Electric

Connected

Shared

Example automotive application built with Qt



Harman





Mercedes-Benz' MBUX

Rimac concept Supercar



Qt Design Studio

- › A unique design tool that allows designers to import designs into Qt world, and turn them into QML code.
- › Designers can edit QML visually.
- › Developers can collaborate seamlessly on editing the same QML code.
- › Live preview on the target hardware.
- › Rapid prototyping and fast iterations.
- › Open Source / Commercial licenses.



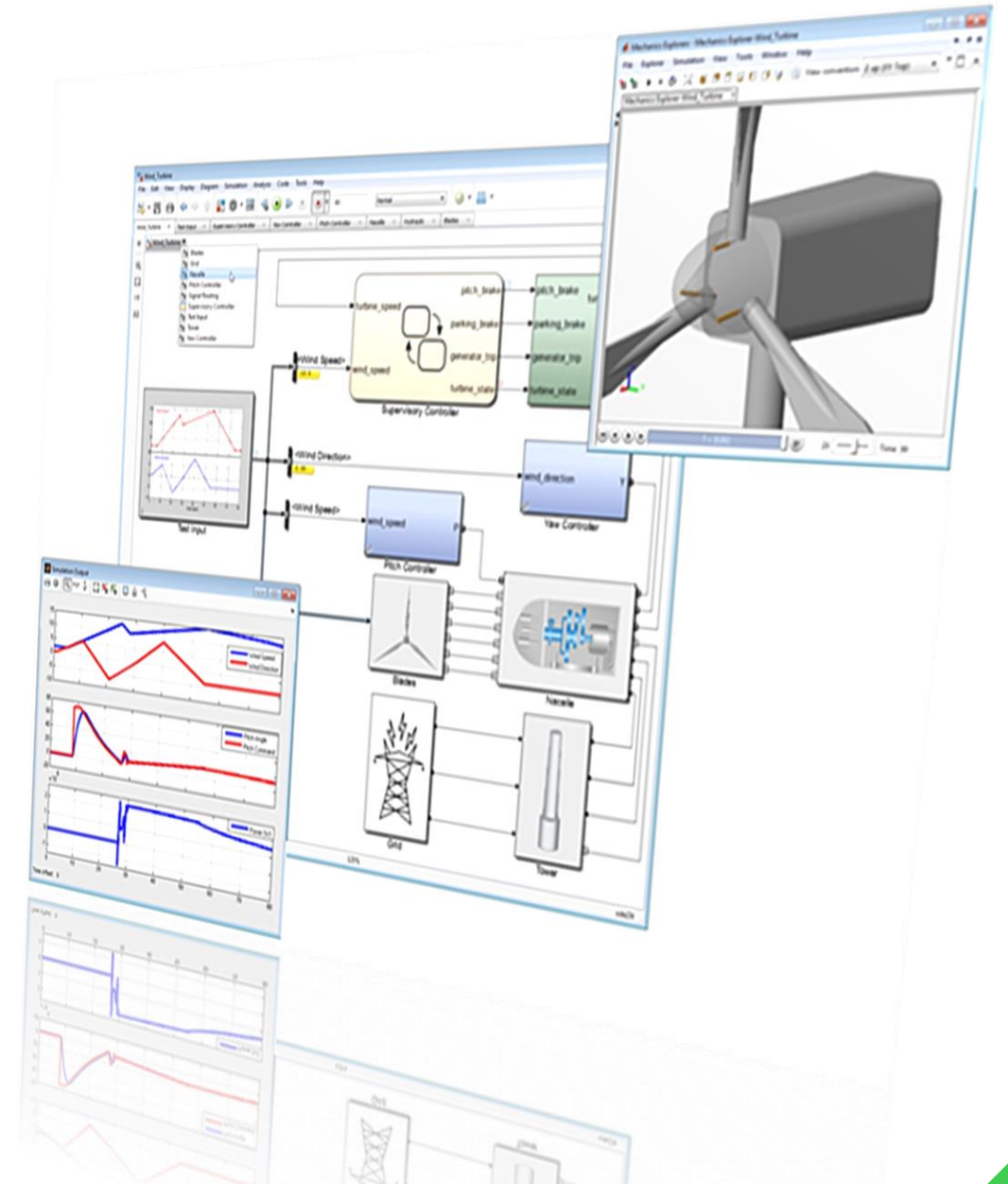
Qt Design Studio Key features

- › Visual 2D Editor.
- › 3D Scene Editor.
- › Code Editor.
- › Timeline Animations.
- › Libraries of presets of UI components.
- › And much more (States, Transition Editor, Connection Editor, Binding Editor, ...).
- › Simulink Support.



Simulink for HMI development

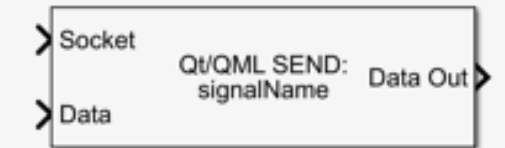
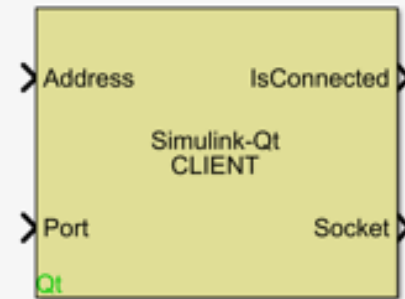
- › Rich sets of Simulink blocks to design and simulate complex HMI logics using blocks and connections.
- › Abstract complex logic by converting it to a Subsystem.
- › Stateflow charts to design state machines and flow charts.



Simulink Qt blockset

- › Contains all the Simulink blocks needed to establish a bi-directional data transfer between a Simulink model and a Qt Design Studio application.
- › Requires an existing Simulink installation. The blockset installer can be downloaded from this repo: <https://git.qt.io/qt-design-studio/simulink-plugin-dependencies>

Simulink - Qt Block Library



Address



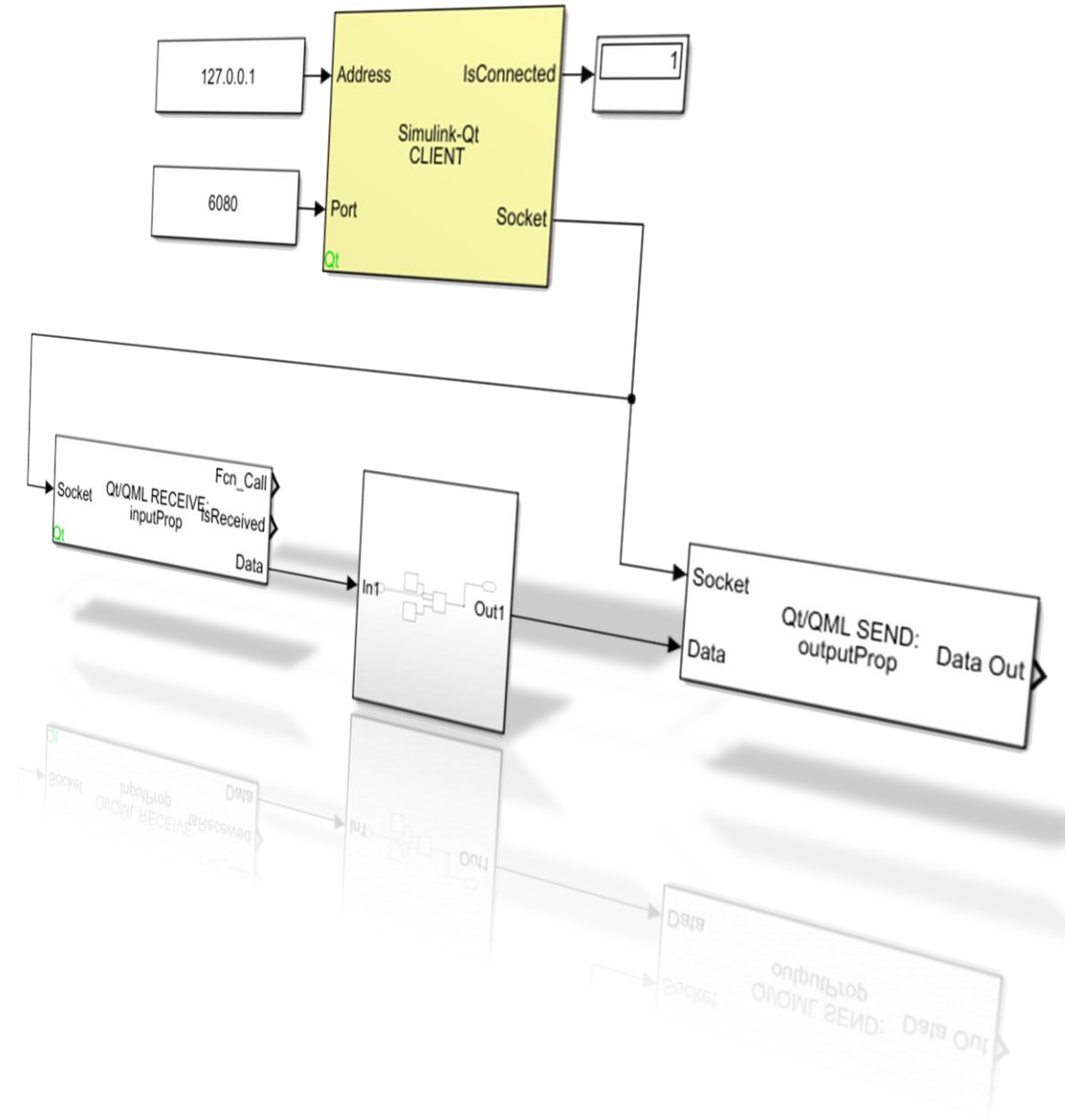
Port

Simulink – Qt Design Studio workflow



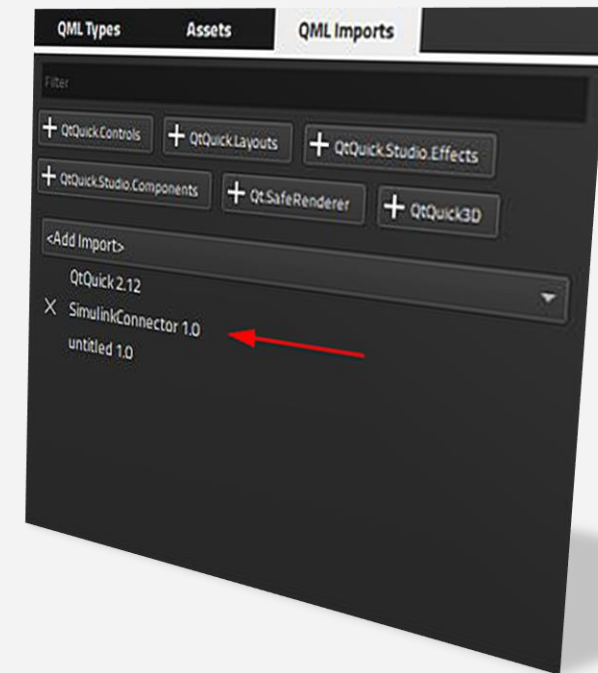
Simulink side

- › Open the Qt blockset.
- › Add Client, IP and Port blocks.
- › Add send and/or receive blocks for each data that needs to be sent to QML.
- › Give the send and receive blocks names matching the properties names in QML side.
- › Connect the blocks like in the picture.



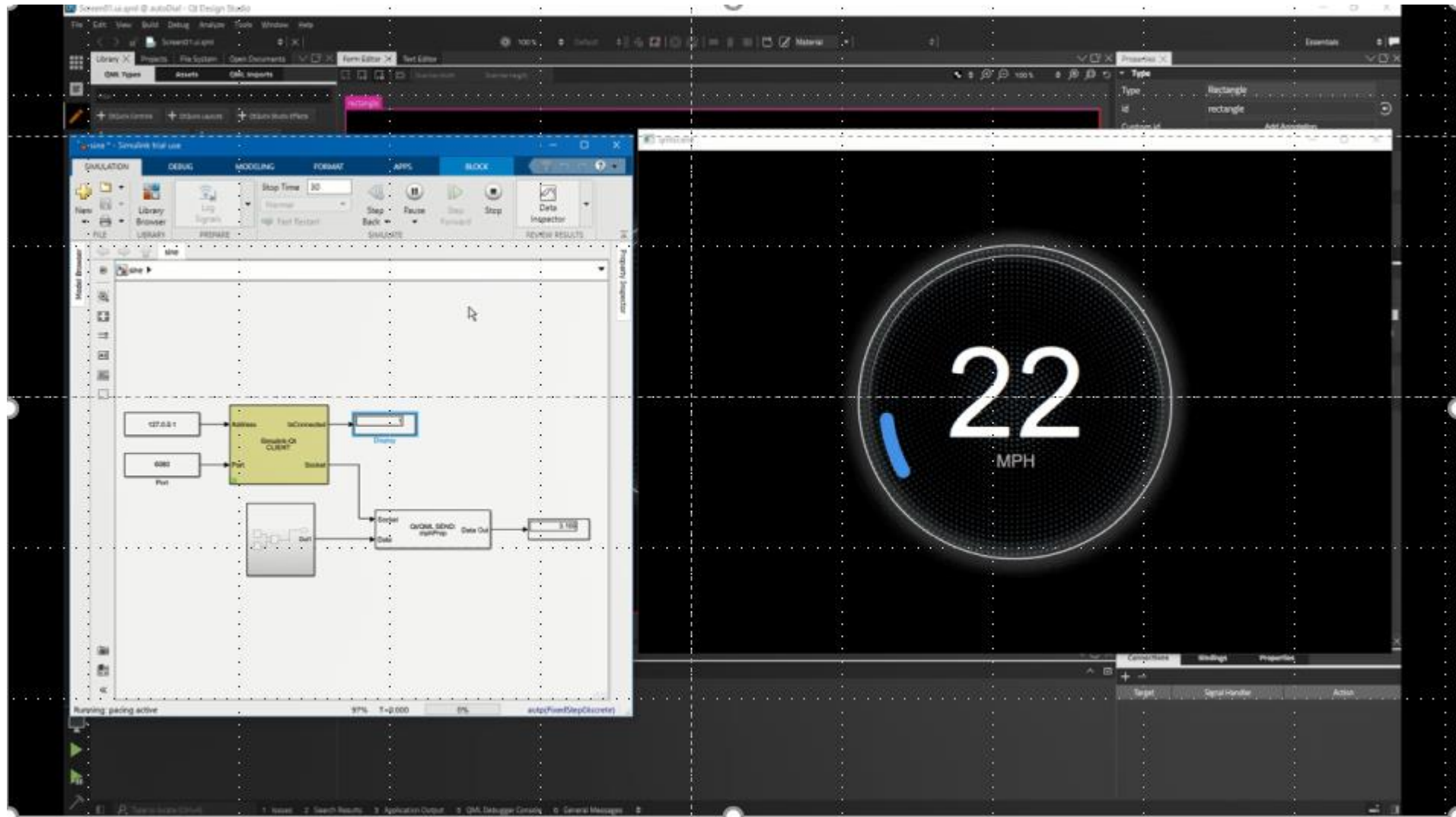
Qt Design Studio side

- › Qt Design Studio 1.6+ (commercial version) comes with Simulink connectivity built-in.
- › Add SimulinkConnector module to the project.
- › Create properties on the QML root object, matching the properties on Simulink side that will to be sent/received to/from Qt Design Studio.
- › Bind the created properties to the desired QML components properties.



Item	Property	Property Type	Property Value
Rectangle	outputProp	real	0
Rectangle	inputProp	real	0

Demo



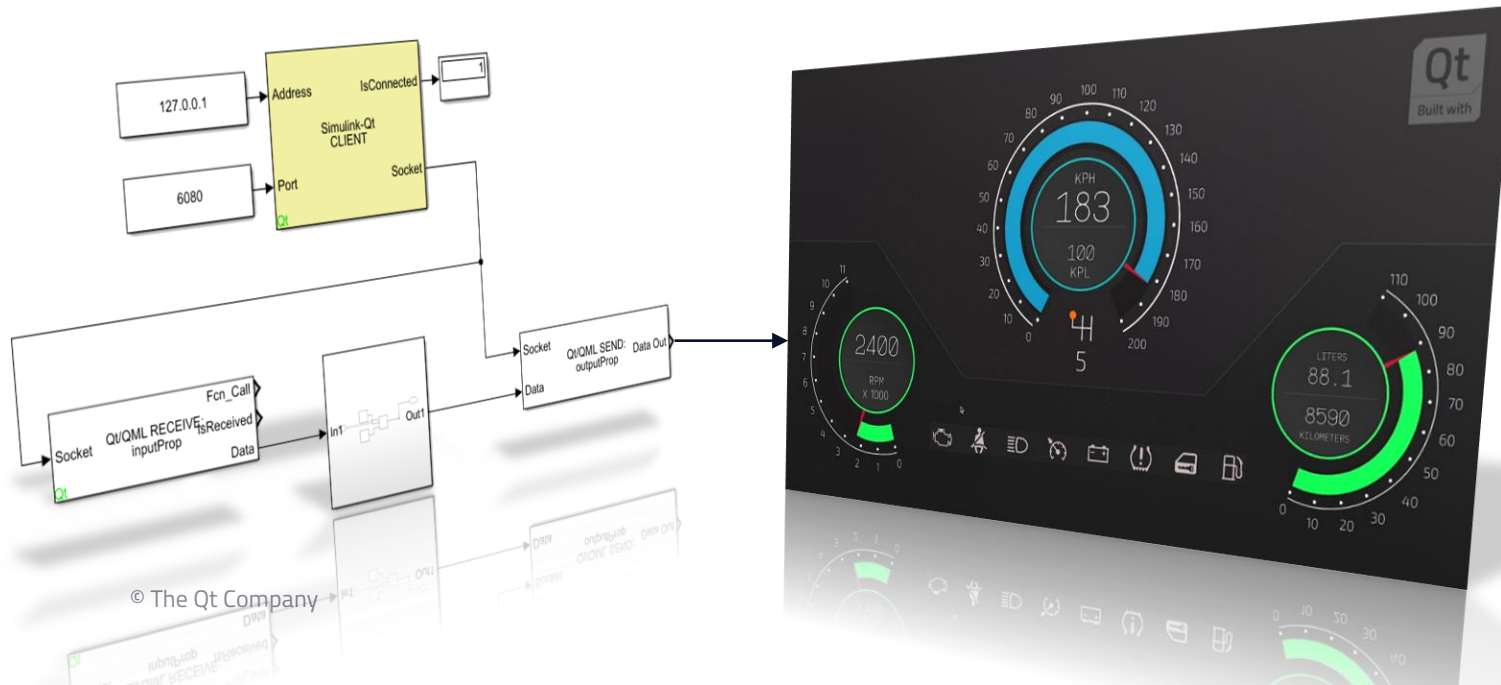
Deployment on hardware

- › Simulink can export a model as C/C++ code using [Embedded Coder](#). Embedded Coder generates fast and optimized code that is efficient for running on embedded processors. The generated code can be integrated both with a Qt Quick based HMI as well as a Qt Quick Ultralite based HMI.



Benefits for customers

- › Leverage the power of Qt and QML to build advanced UI mixing 2D and 3D designs.
- › Easily connect Simulink simulation model to QML UI.
- › Simple and intuitive workflow.
- › Quick iteration cycles: easy to test modifications both in the simulation or UI sides.
- › Collaborative workflow where designers and developers can work together more productively.



MathWorks – Qt collaboration

- › MathWorks developed the underlying interface for Simulink – Qt communication.
- › MathWorks Consulting is helping Qt to extend the workflow, extending it to code generation.



Thank you

