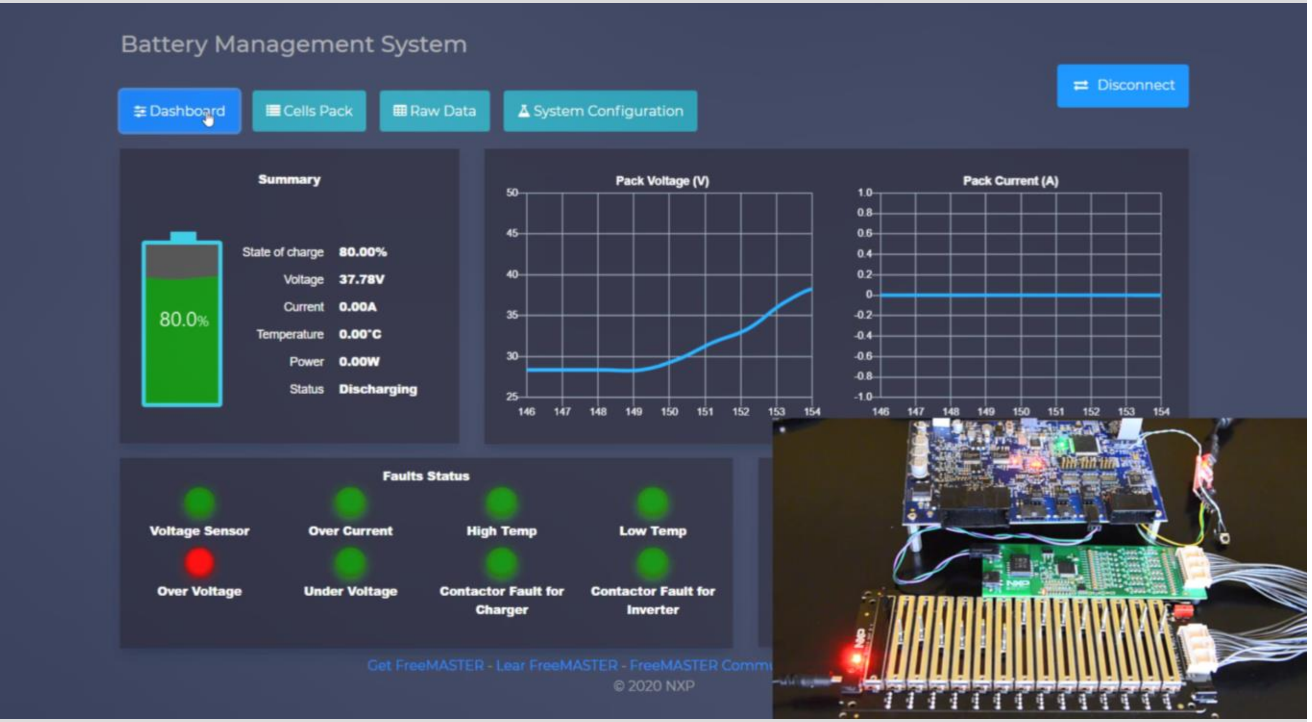
SECURE CONNECTIONS
FOR A SMARTER WORLD

Rapid Prototyping Embedded Designs using NXP Model-Based Design Toolbox: A Battery Management System Application

Irina Costachescu

Marius-Lucian Andrei

MATLAB EXPO



AGENDA

1. **NXP Ecosystem**

2. NXP's **Model-Based Design Toolbox** Introduction

3. Model-Based Design Toolbox for **Battery Management Systems**

NXP SEMICONDUCTORS WORLDWIDE

Together with our valued customers, we’re not just advancing technology, we’re advancing society.

60 years of combined experience and expertise

Operations in more than **30 countries** worldwide

Approximately **31,000 employees**

Headquarters in The Netherlands – **Eindhoven**

AUTOMOTIVE

Enabling carmakers to develop smarter solutions for complex autonomy, connectivity, and electrification challenges

Accelerating the shift to greater mobility

INDUSTRIAL

Reducing wasted time, money, and effort by helping business run more efficiently.

Enabling more efficient data processing

MOBILE

Giving wearable and mobile devices easier access to the services that make modern life more convenient without compromising security and safety.

Transforming how people and devices connect

SMART HOME

Solutions that listen, learn, and adapt into the places we call home for more comfort, affordability, safety, and convenience.

Powering the intelligence behind the technologies

SMART CITY

Simplifying how people access and interact with local services to achieve new standards of sustainability, efficiency, mobility, and economic growth.

Anticipating the demands of tomorrow

COMMUNICATION INFRASTRUCTURE

Powering insights and inspiring performance with hardware solutions for handling 5G connectivity across the emerging communications spectrum.

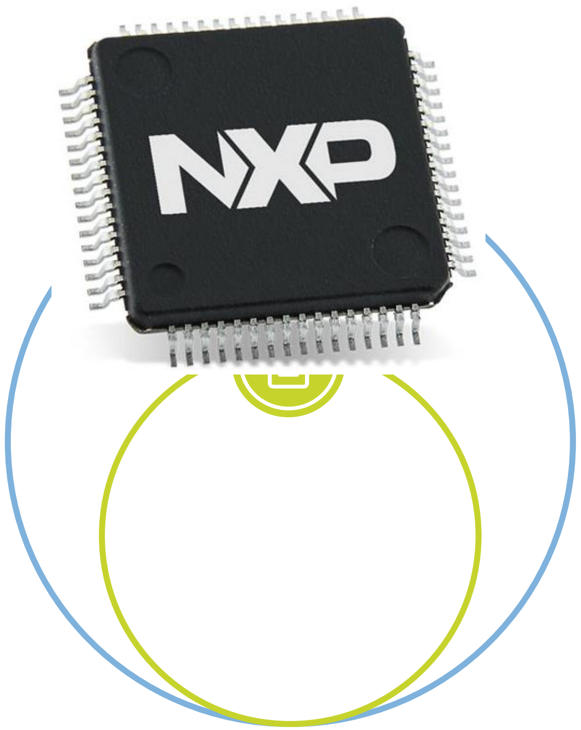
Delivering real-time responsiveness at the speed of 5G

NXP ECOSYSTEM

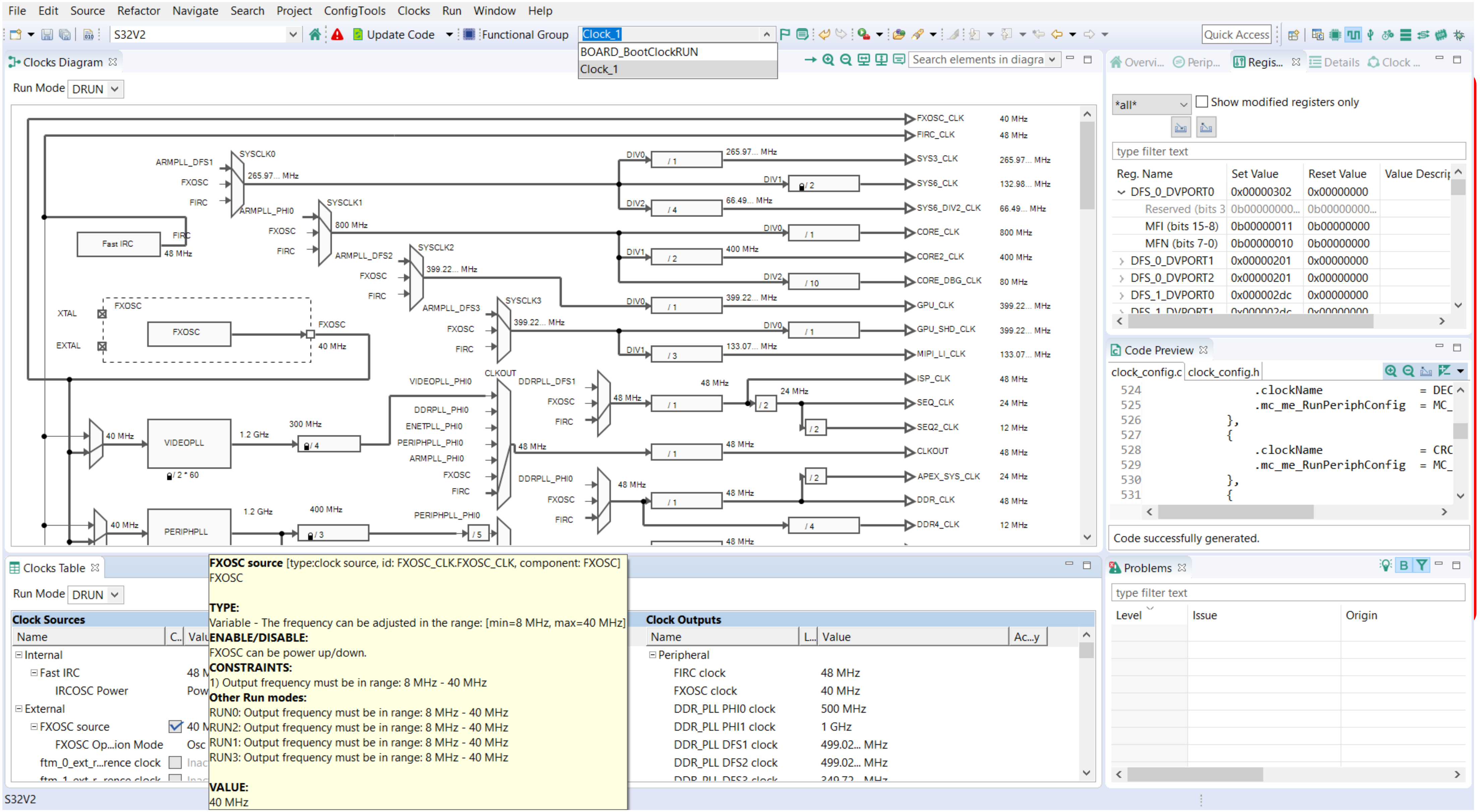
DRIVERS, MIDDLEWARE, LIBRARIES

- Simplify hardware access by using hardware optimized software

BUILD, DEBUG, CONFIGURATION TOOLS



BUILD, DEBUG, CONFIGURATION TOOLS



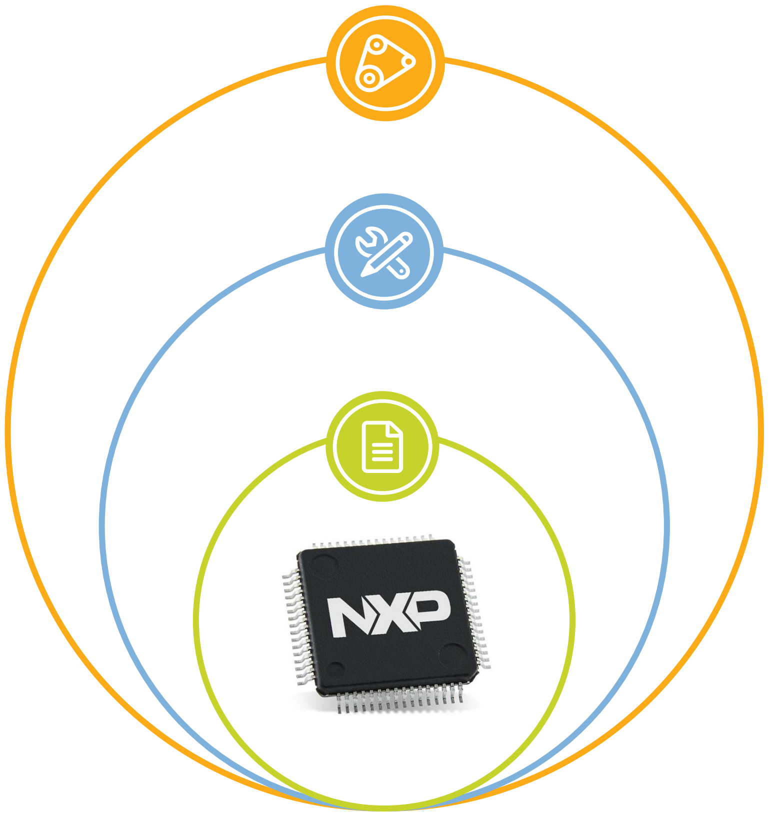
NXP ECOSYSTEM

DRIVERS, MIDDLEWARE, LIBRARIES

- Simplify hardware access by using hardware optimized software

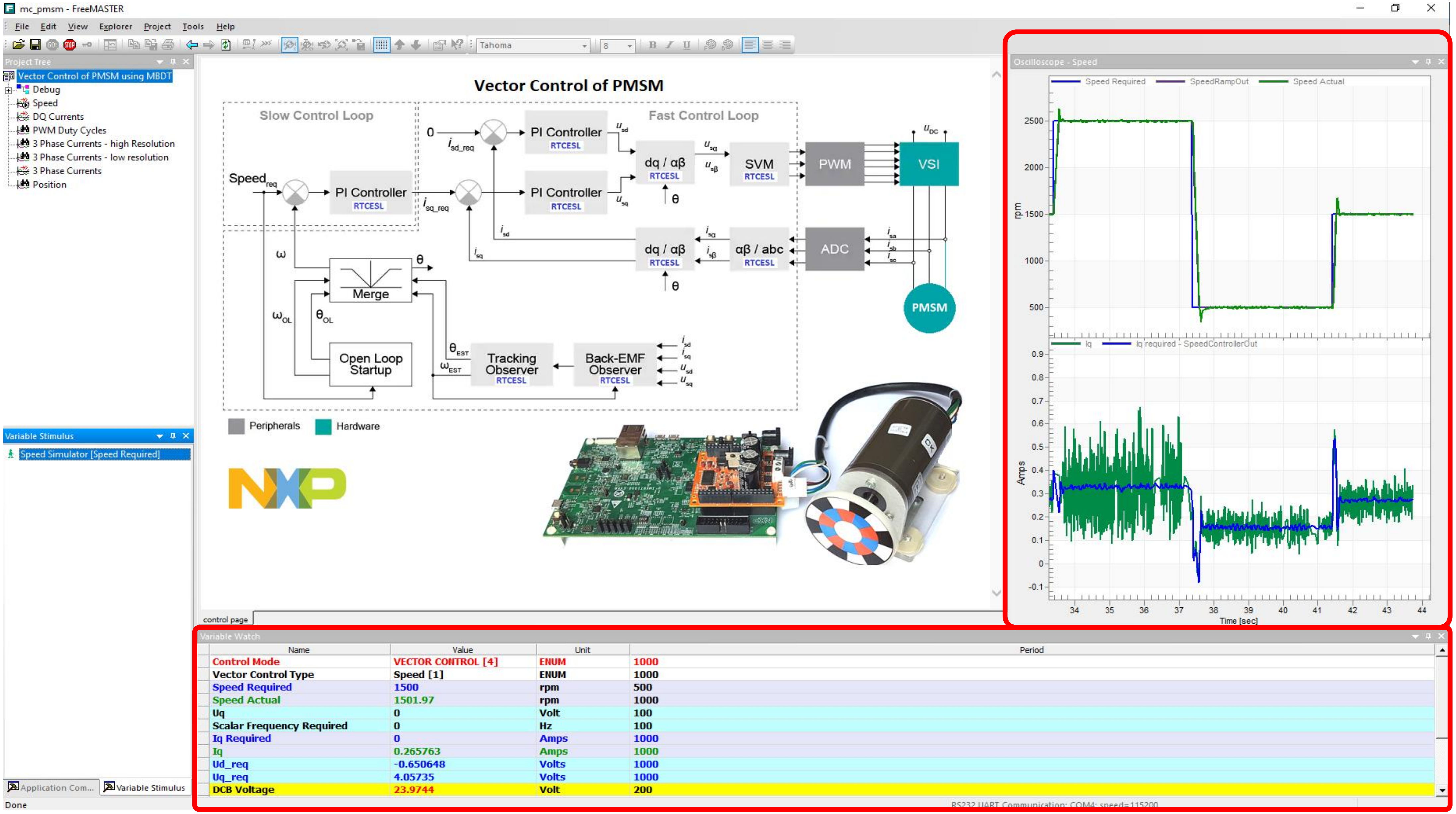
BUILD, DEBUG, CONFIGURATION TOOLS

- Application development inside an IDE
- Build Tools, Debug Tools and Configuration Tools integrated within the IDE
- Drivers, Middleware and Libraries configuration and initialization in a graphical environment



REAL TIME MONITOR, DEMO TOOLS

REAL TIME MONITOR, DEMO TOOLS



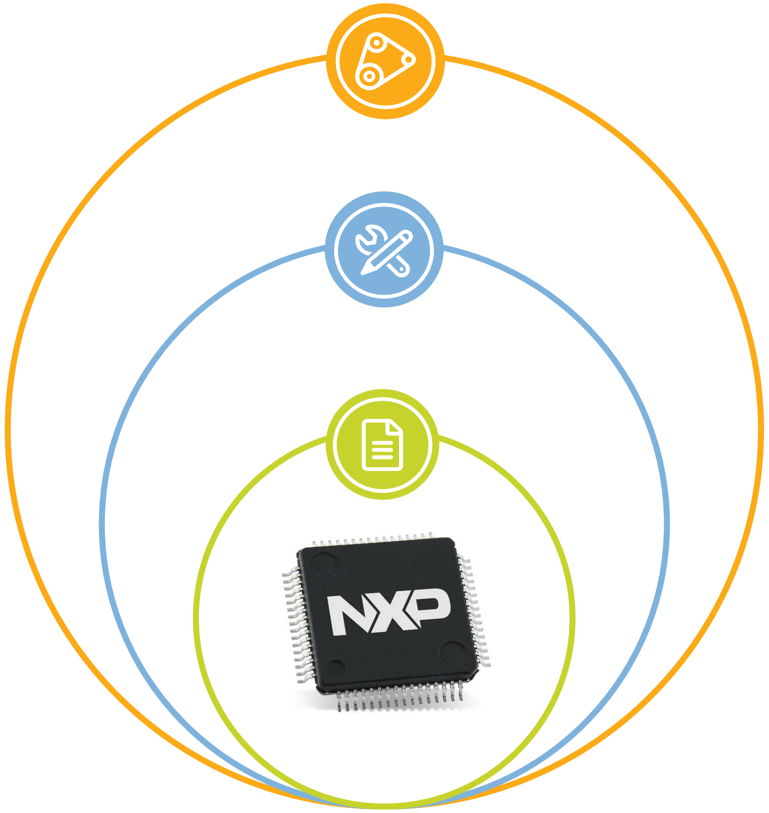
NXP ECOSYSTEM

DRIVERS, MIDDLEWARE, LIBRARIES

- Simplify hardware access by using hardware optimized software

BUILD, DEBUG, CONFIGURATION TOOLS

- Application development inside an IDE
- Build Tools, Debug Tools and Configuration Tools integrated within the IDE
- Drivers, Middleware and Libraries configuration and initialization in a graphical environment



REAL TIME MONITOR, DEMO TOOLS

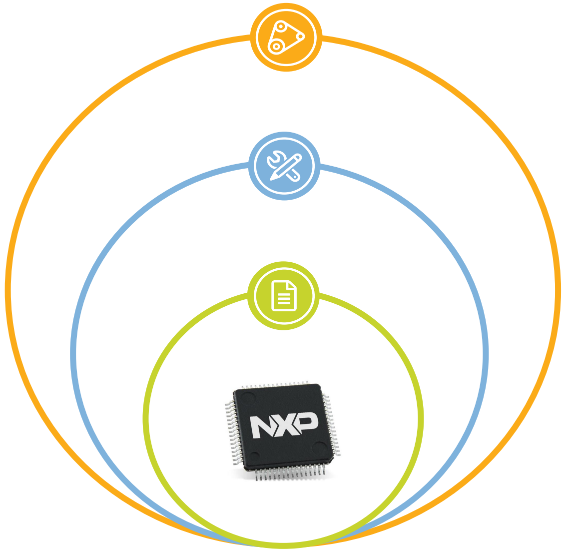
- Check the status of the running on target application in real time using **FreeMASTER**
- Write and read variables, registers, memory locations
- Monitor signals on the embedded target
- Fast demo design

DRIVERS, MIDDLEWARE, LIBRARIES

- Simplify hardware access by using hardware optimized software

BUILD, DEBUG, CONFIGURATION TOOLS

- Application development inside an IDE
- Build Tools, Debug Tools and Configuration Tools integrated within the IDE
- Drivers, Middleware and Libraries configuration and initialization in a graphical environment



REAL TIME MONITOR, DEMO TOOLS

- Check the status of the running on target application in real time using **FreeMASTER**
- Write and read variables, registers, memory locations
- Monitor signals on the embedded target
- Fast demo design

MODEL-BASED DESIGN

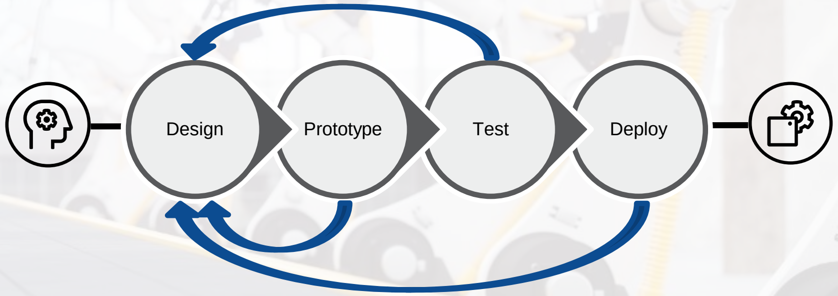
MODEL-BASED DESIGN TOOLBOX

- Collection of Drivers, Libraries and Tools
- Embedded systems design and deployment on NXP MCUs directly from **Simulink**

MATHWORKS ECOSYSTEM MATLAB/SIMULINK

- Model-Based Design
- Simulation
- Automatic Code Generation
- Verification and Validation

FROM IDEA TO APPLICATION



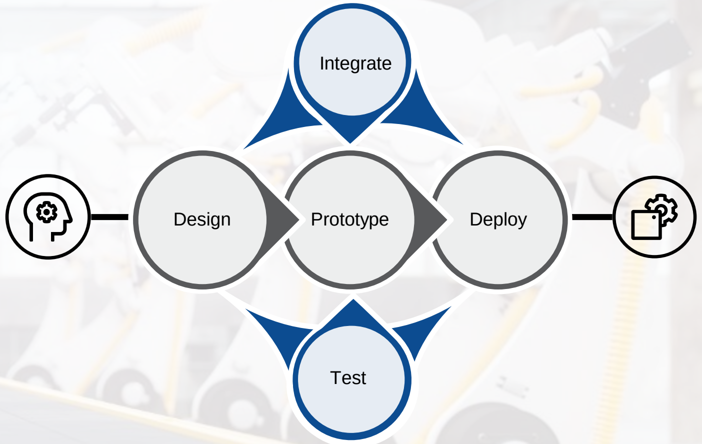
MODEL-BASED DESIGN TOOLBOX

- Collection of Drivers, Libraries and Tools
- Embedded systems design and deployment on NXP MCUs directly from **Simulink**

MATHWORKS ECOSYSTEM MATLAB/SIMULINK

- Model-Based Design
- Simulation
- Automatic Code Generation
- Verification and Validation

FROM IDEA TO APPLICATION



- ✓ **FAST** – Time To Market
- ✓ Hardware **independent** simulations
- ✓ Easy To **Use-Reuse**

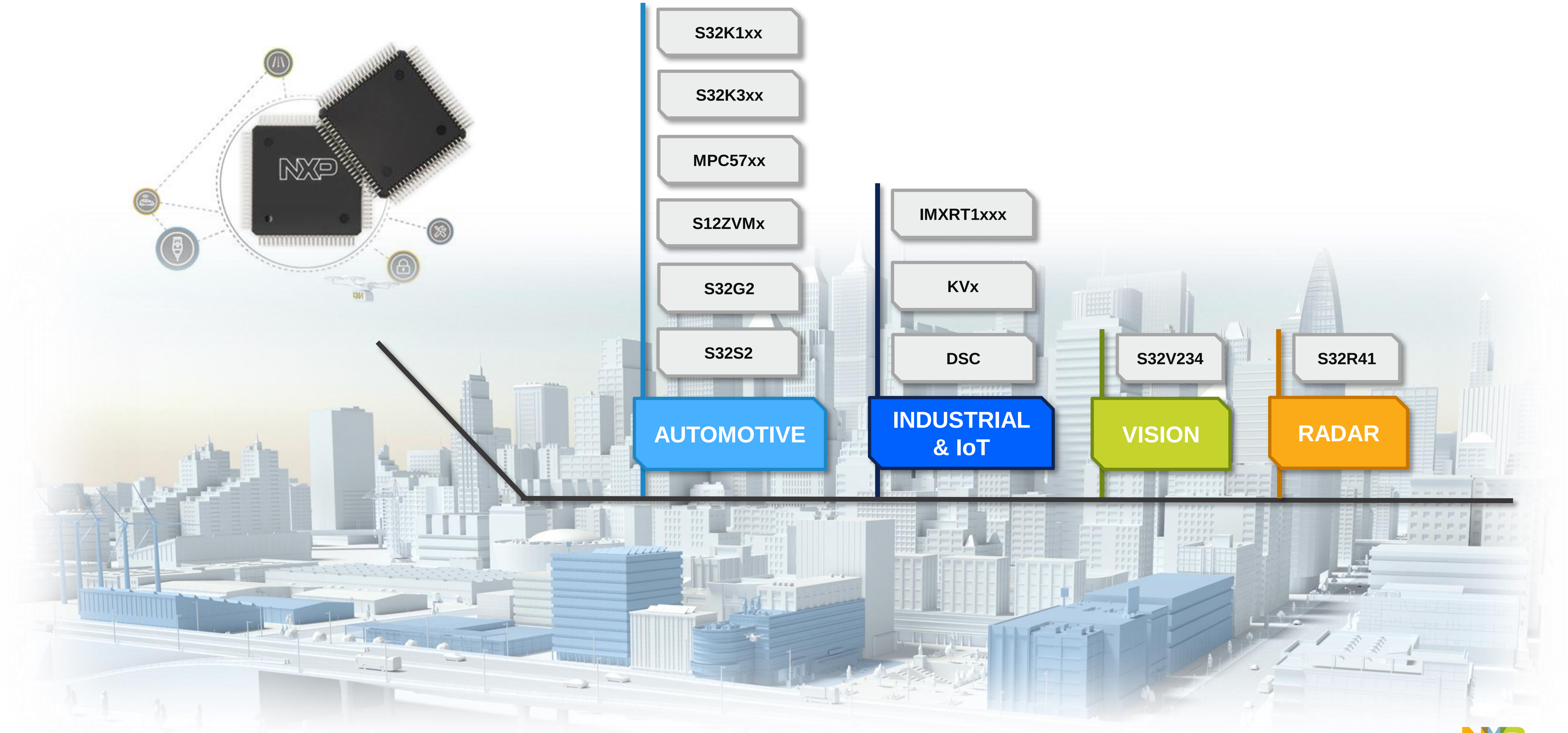
MODEL-BASED DESIGN TOOLBOX

- Collection of Drivers, Libraries and Tools
- Embedded systems design and deployment on NXP MCUs directly from **Simulink**

MATHWORKS ECOSYSTEM MATLAB/SIMULINK

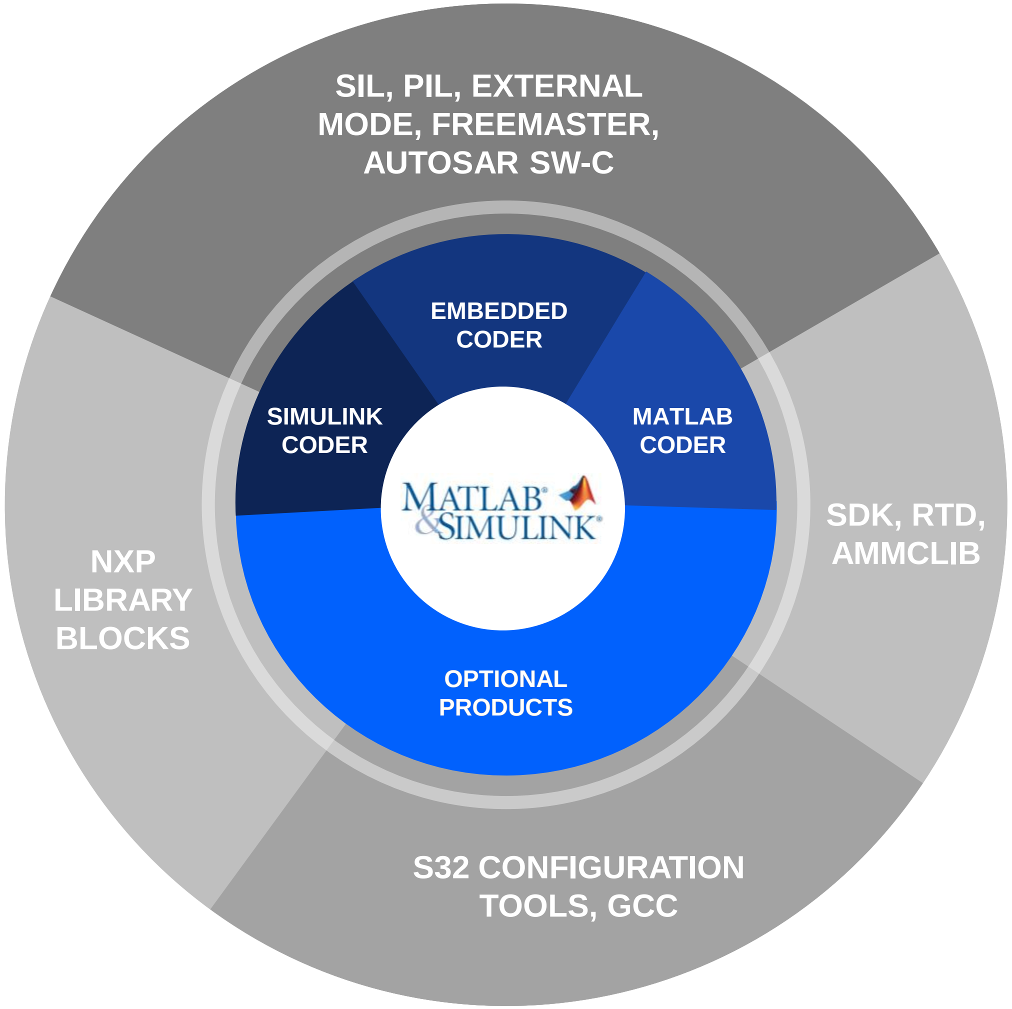
- Model-Based Design
- Simulation
- Automatic Code Generation
- Verification and Validation

MODEL-BASED DESIGN TOOLBOX – SUPPORTED PLATFORMS

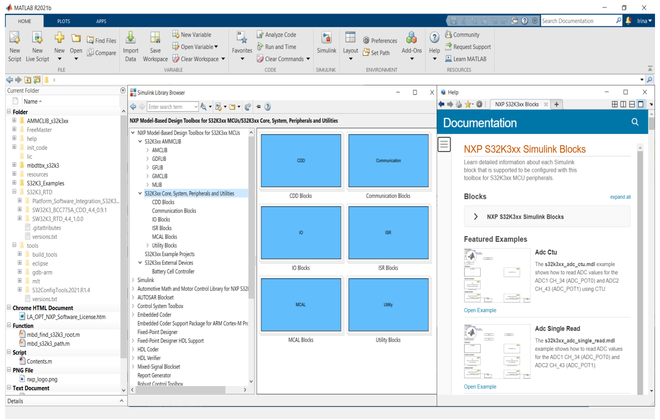


MODEL-BASED DESIGN TOOLBOX – SOFTWARE DEVELOPMENT ENVIRONMENT AND MODULES

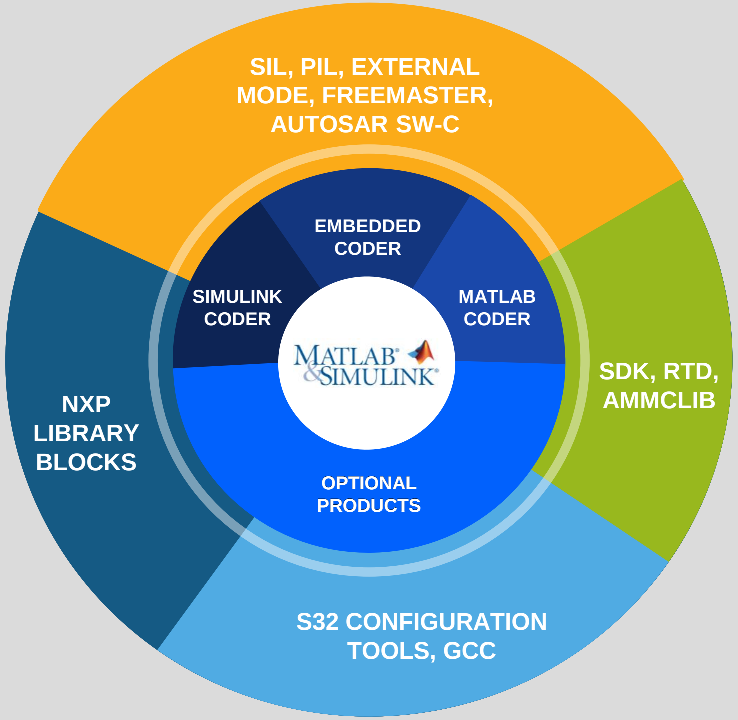
- Stateflow
- Simscape
- Motor Control Blockset
- AUTOSAR Blockset
- Deep Learning Toolbox
- Vehicle Network Toolbox

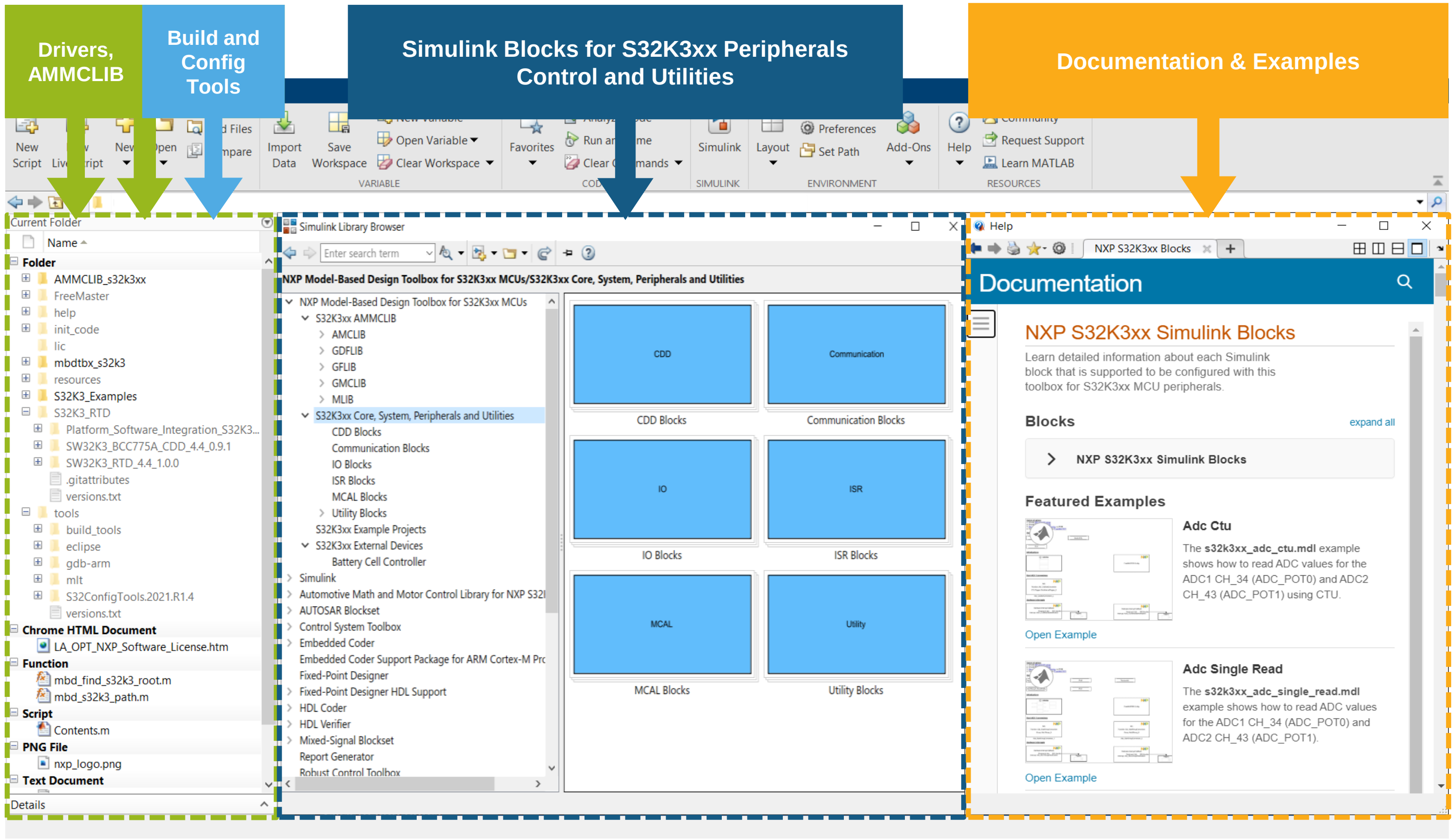


MODEL-BASED DESIGN TOOLBOX OVERVIEW



MODEL-BASED DESIGN TOOLBOX - OVERVIEW





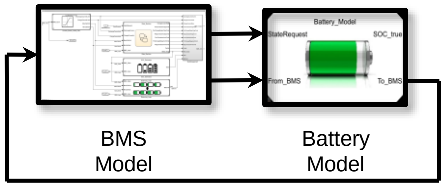
- ✓ MCU Peripherals **Configuration & Control** using NXP's **Real-Time Drivers** (RTD) / **Software Development Kit** (SDK) APIs
- ✓ **External Tools** integration for **peripherals, pins and clocks configuration; Build Tools**
- ✓ **MBDT blocks generate code on top of RTD/SDK**
- ✓ **Example applications** covering all the toolbox **features and functionalities**

* AMMCLIB – Automotive Math and Motor Control Library

BATTERY MANAGEMENT SYSTEM – DEVELOPMENT FLOW WITH MATHWORKS AND NXP

MODEL-BASED DESIGN DEVELOPMENT FLOW

Idea Incubation

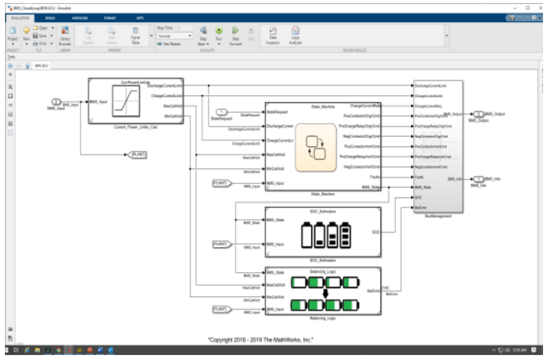


PC Environment

Step 1 – System Requirements

Model-in-the-Loop

- Software requirements
- Control system requirements
- Overall application control strategy

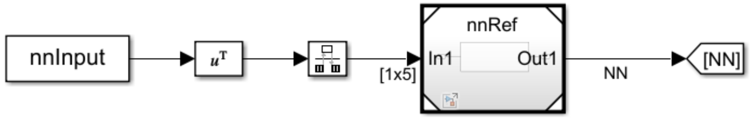


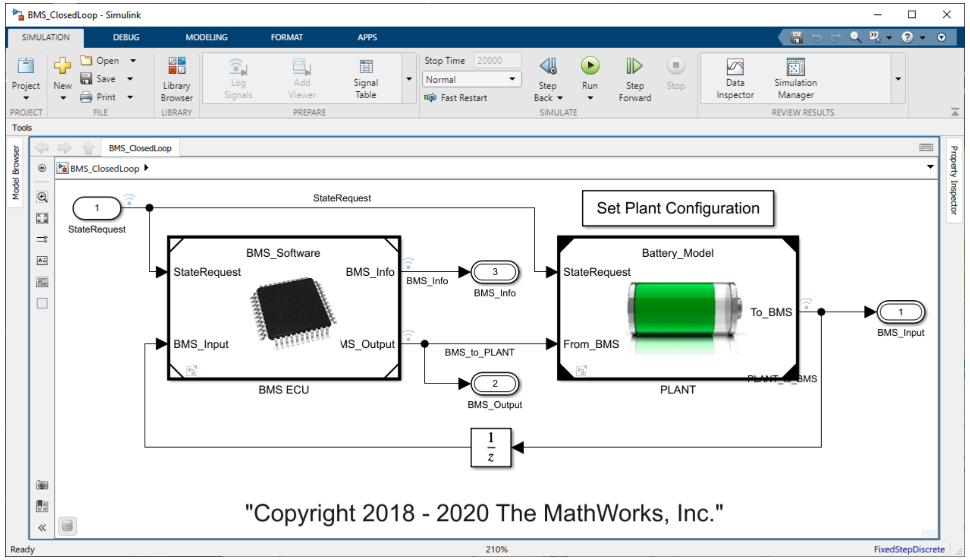
IDEA INCUBATION
MODEL-IN-THE-LOOP

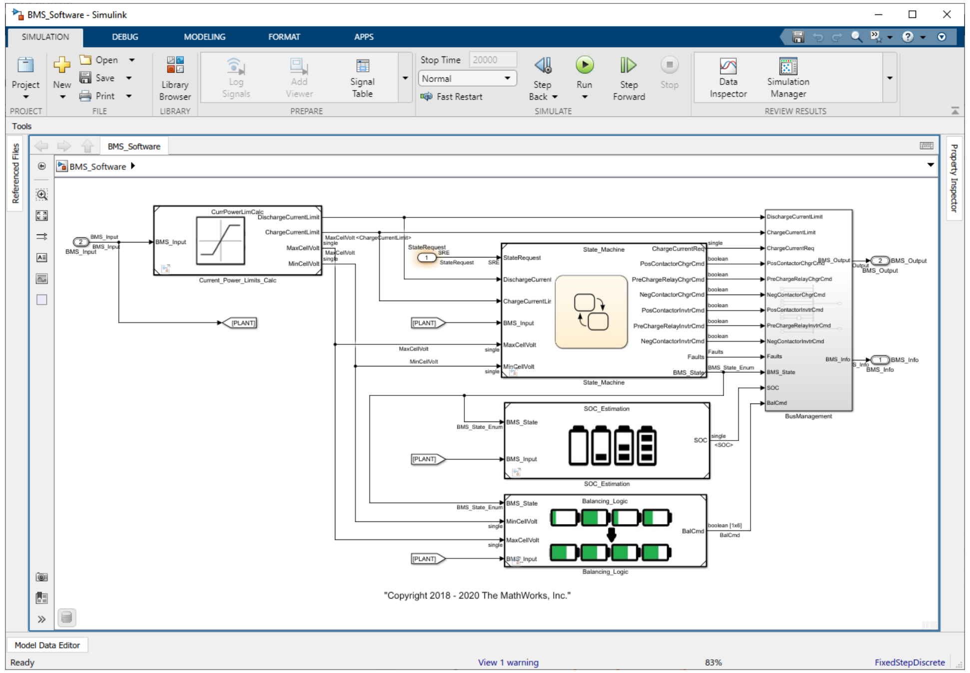
MIL

Model-in-the-Loop

- Software requirements
- Control system requirements
- Overall application control strategy
- Model testing



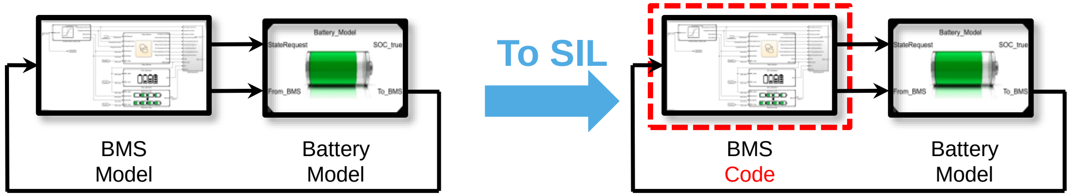




MODEL-BASED DESIGN DEVELOPMENT FLOW

Idea Incubation

Automatic Code Generation



PC Environment

PC Environment

Step 1 – System Requirements

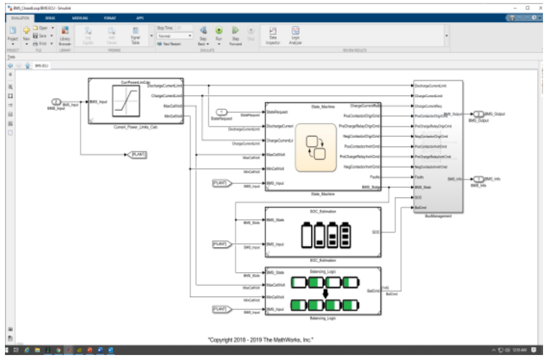
Model-in-the-Loop

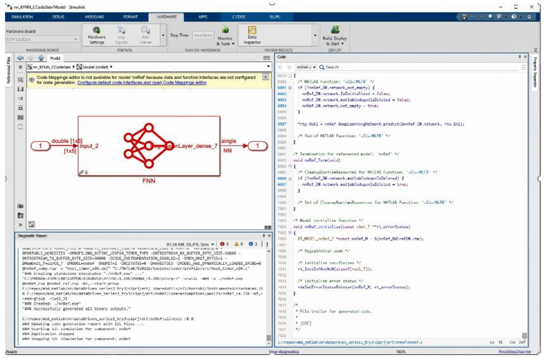
- Software requirements
- Control system requirements
- Overall application control strategy

Step 2 – Modeling/Simulation

Software-in-the-Loop

- Control algorithm design
- Code generation preparation
- Control system design
- Start testing implementation approach





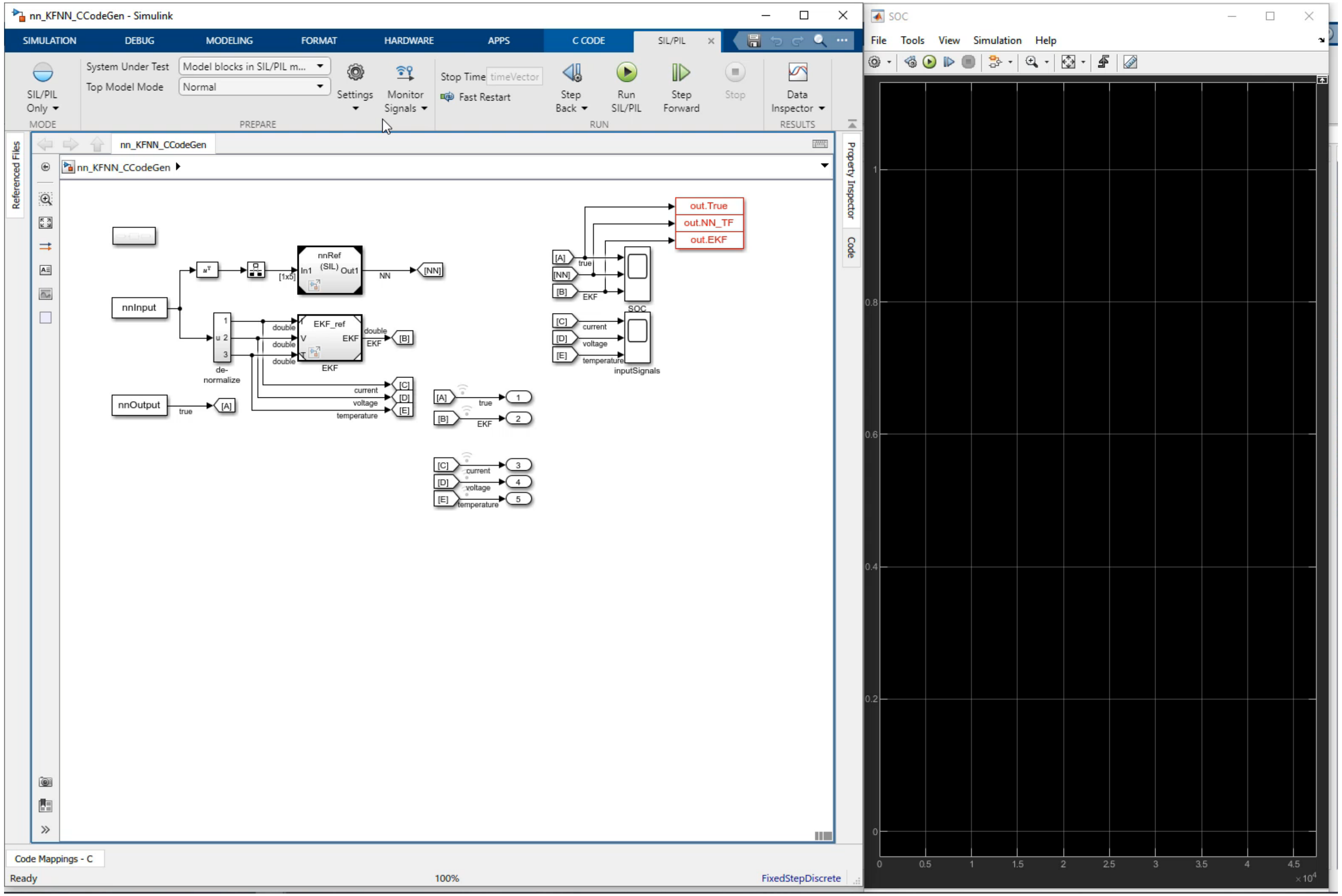
AUTOMATIC CODE GENERATION

SOFTWARE-IN-THE-LOOP

SIL

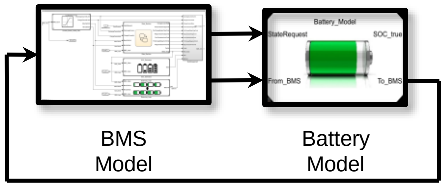
Software-in-the-Loop

- Control algorithm design
- Code generation preparation
- Control system design
- Start testing implementation approach



MODEL-BASED DESIGN DEVELOPMENT FLOW

Idea Incubation

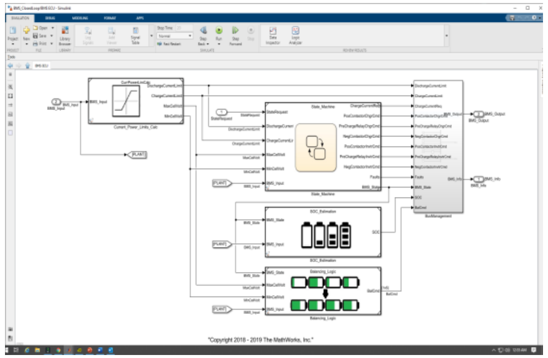


PC Environment

[Step 1 – System Requirements]

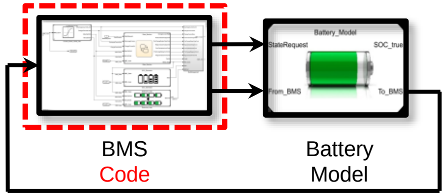
Model-in-the-Loop

- Software requirements
- Control system requirements
- Overall application control strategy



To SIL

Automatic Code Generation

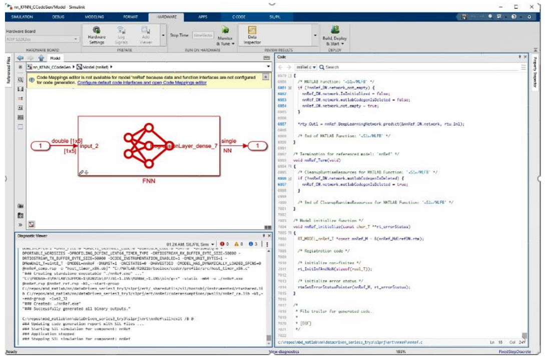


PC Environment

[Step 2 – Modeling/Simulation]

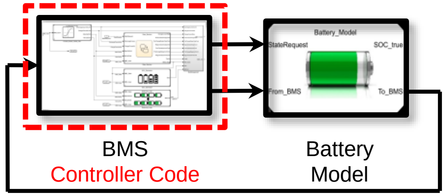
Software-in-the-Loop

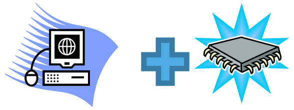
- Control algorithm design
- Code generation preparation
- Control system design
- Start testing implementation approach



To PIL

Code Validation



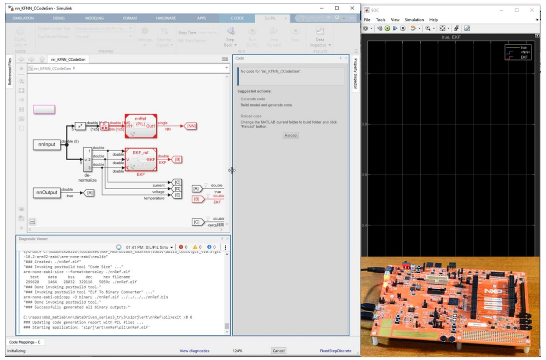


PC Environment
+ MCU

[Step 3 – Rapid Prototype]

Processor-in-the-Loop

- Controller code generation
- Determine execution time on MCU
- Verify algorithm on MCU
- Check memory/stack usage on MCU

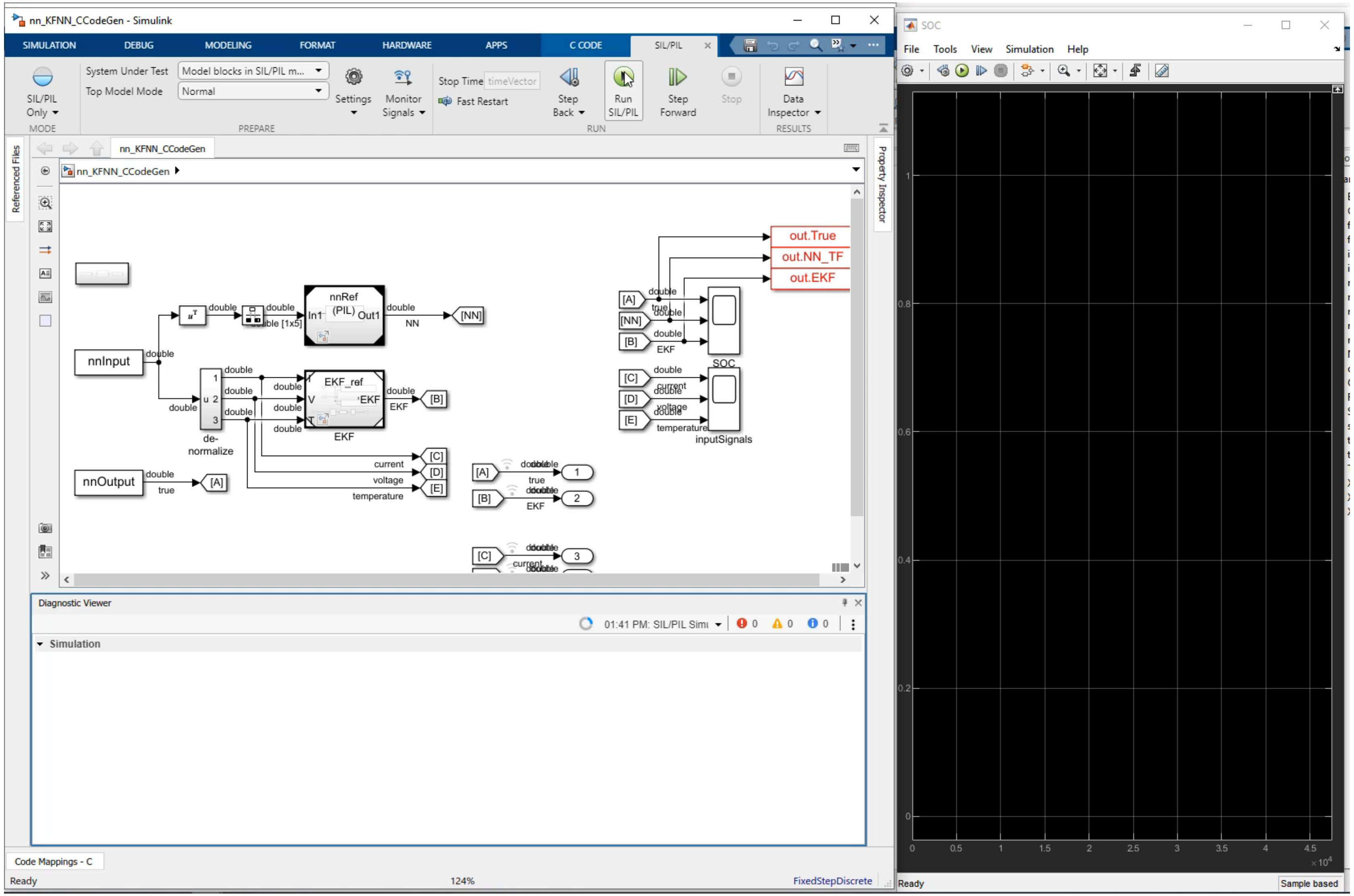


AUTOMATIC CODE GENERATION
PROCESSOR-IN-THE-LOOP

PIL

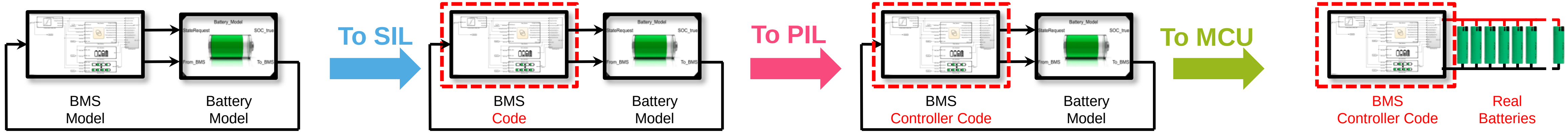
Processor-in-the-Loop

- Controller code generation
- Determine execution time on MCU
- Verify Algorithm on MCU
- Check memory/stack usage on MCU



MODEL-BASED DESIGN DEVELOPMENT FLOW

Idea Incubation	Automatic Code Generation	Code Validation	Prototype



PC Environment

PC Environment

PC Environment + MCU

MCU with Embedded Control Module (ECM)

Step 1 – System Requirements

Model-in-the-Loop

- Software requirements
- Control system requirements
- Overall application control strategy

Step 2 – Modeling/Simulation

Software-in-the-Loop

- Control algorithm design
- Code generation preparation
- Control system design
- Start testing implementation approach

Step 3 – Rapid Prototype

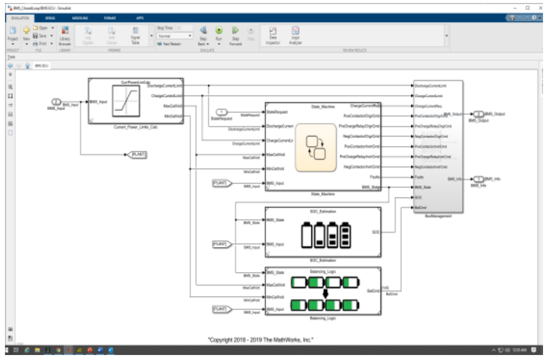
Processor-in-the-Loop

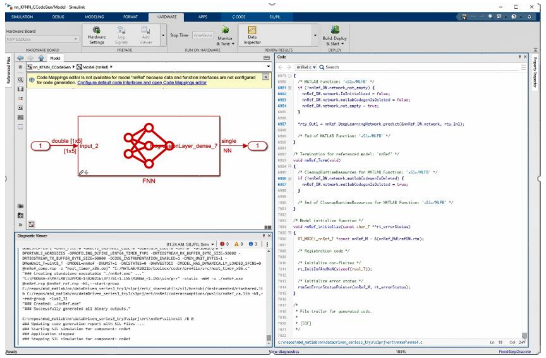
- Controller code generation
- Determine execution time on MCU
- Verify algorithm on MCU
- Check memory/stack usage on MCU

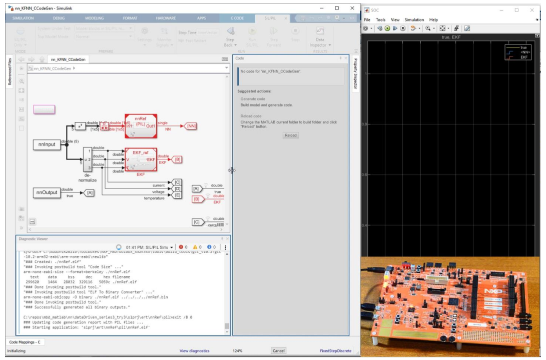
Step 4 – Target MCU Implementation

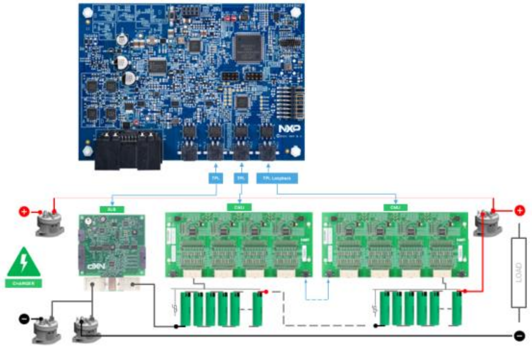
MCU Final Application

- Validation/verification phase
- Controller code generation
- Test system in target environment using tools for data logging and parameter tuning



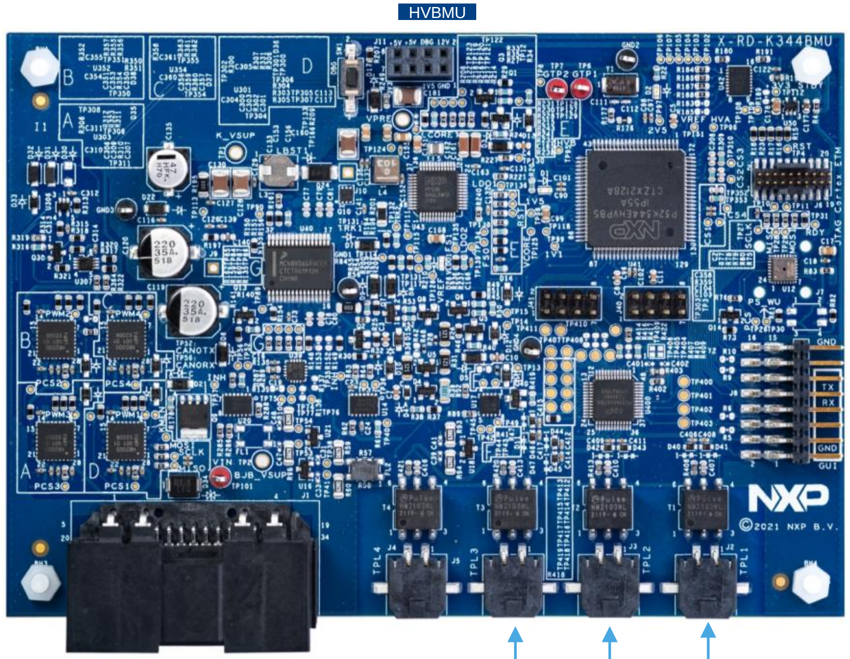


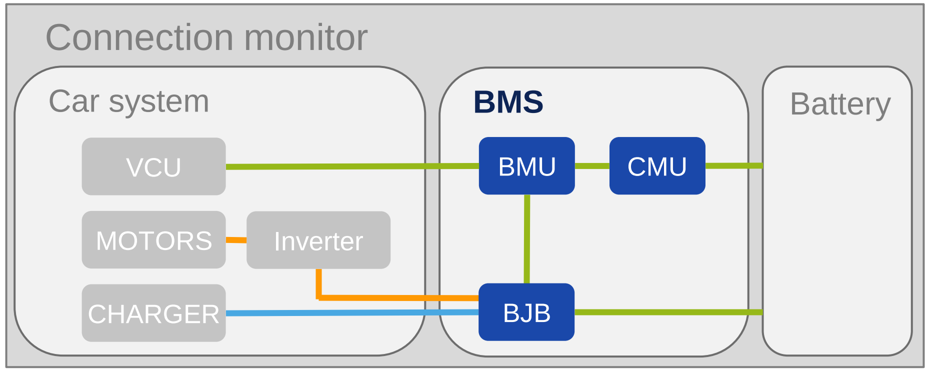


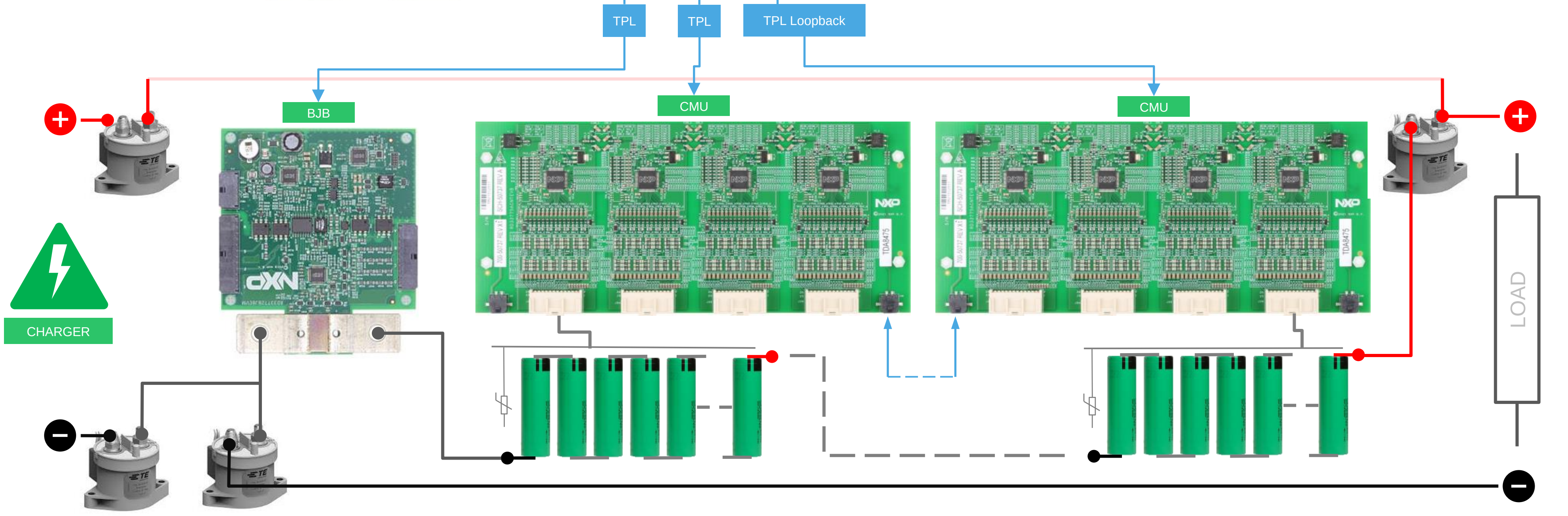


HIGH VOLTAGE BMS DESIGN

Prototype

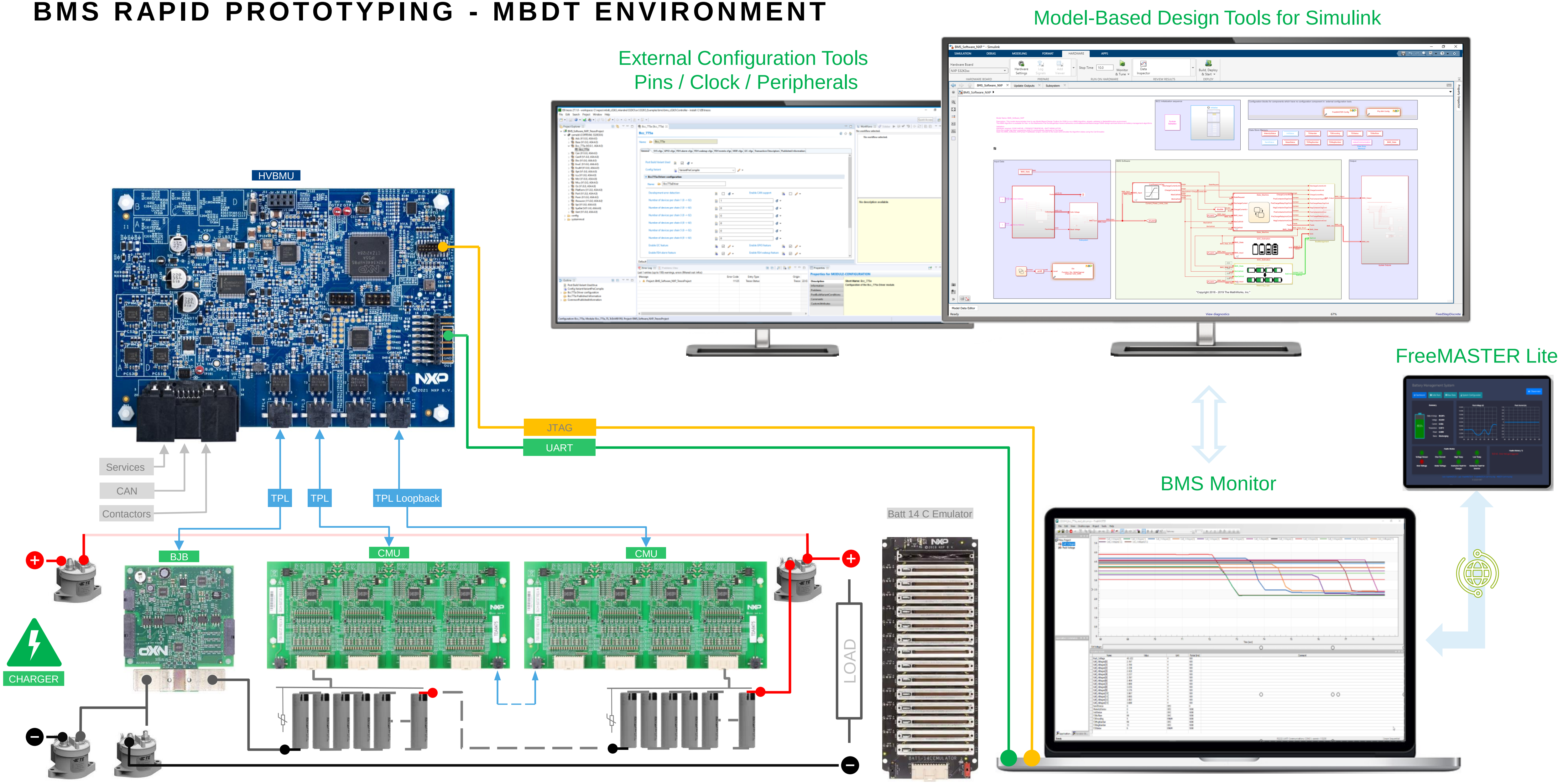






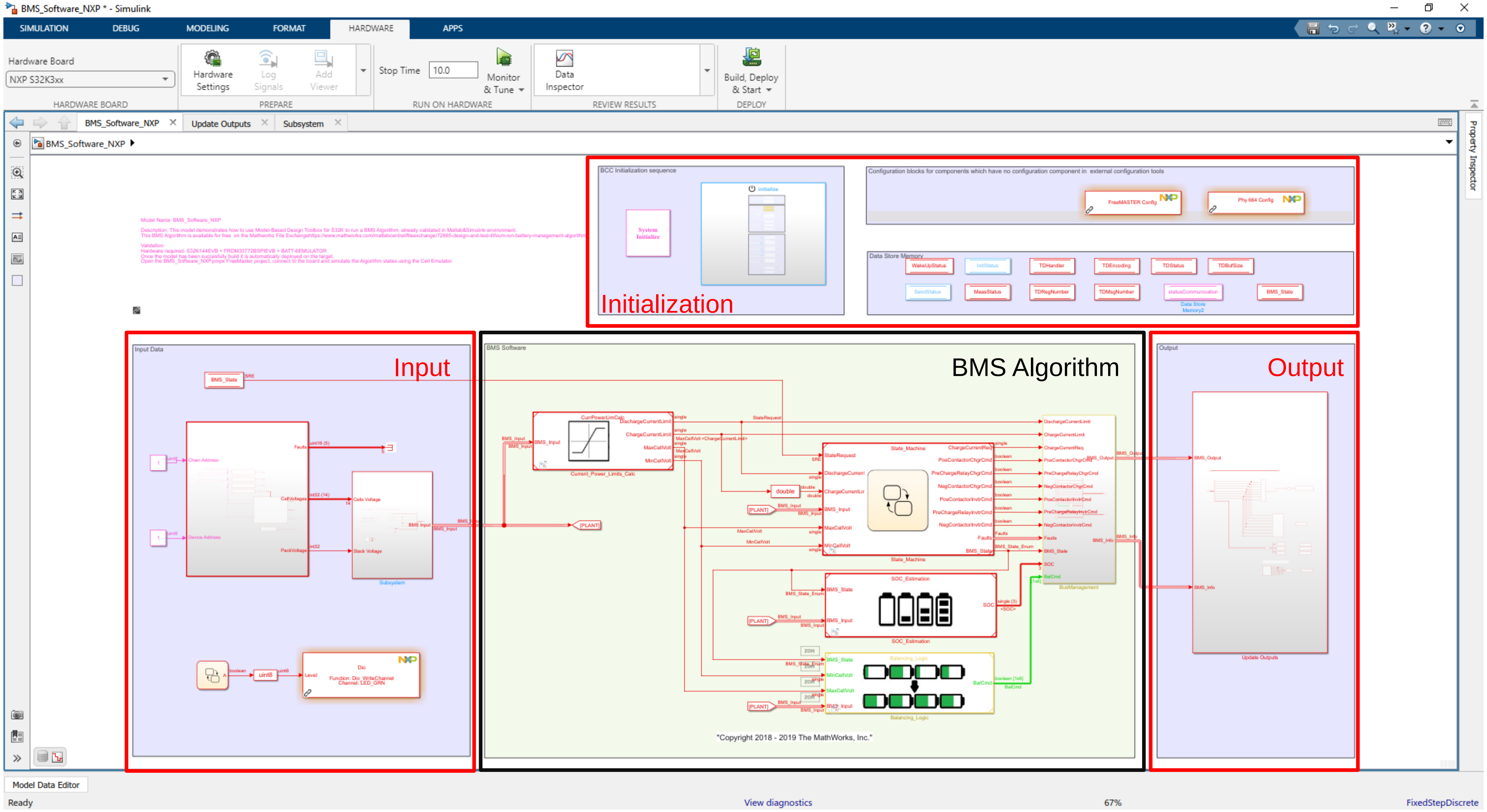
Images presented are for illustration purposes only and may not be an exact representation of the product – their purpose is just exemplification of a concept.

BMS RAPID PROTOTYPING - MBDT ENVIRONMENT



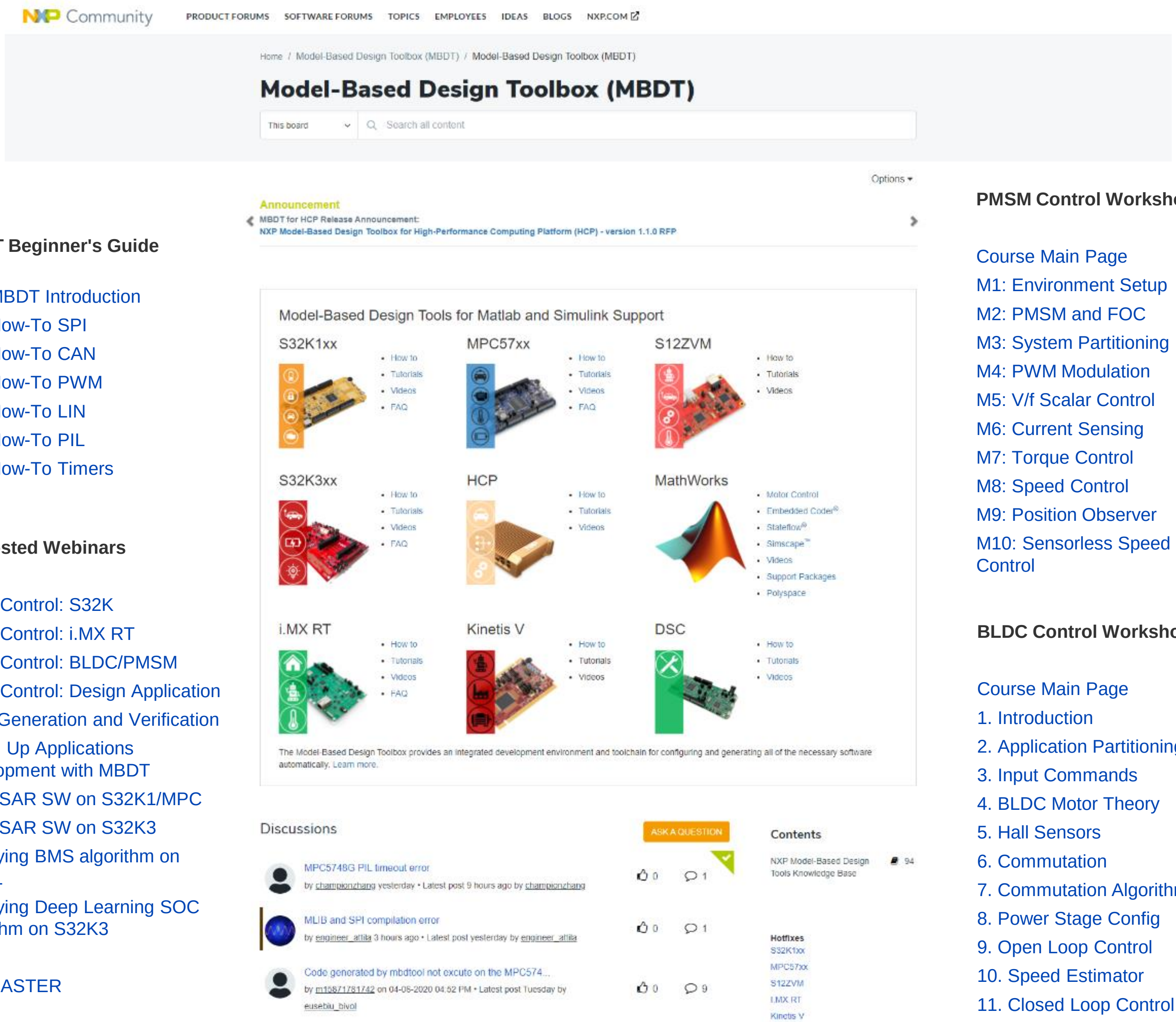
Images presented are for illustration purposes only and may not be an exact representation of the product – their purpose is just exemplification of a concept.

BMS MODEL ON MBDT



FREEMASTER DEMO INTERFACE

ADDITIONAL RESOURCES & SUPPORT



MBDT Beginner's Guide

W1: MBDT Introduction
W2: How-To SPI
W3: How-To CAN
W4: How-To PWM
W5: How-To LIN
W6: How-To PIL
W7: How-To Timers

Co-hosted Webinars

Motor Control: S32K
Motor Control: i.MX RT
Motor Control: BLDC/PMSM
Motor Control: Design Application
Code Generation and Verification
Speed Up Applications Development with MBDT
AUTOSAR SW on S32K1/MPC
AUTOSAR SW on S32K3
Deploying BMS algorithm on S32K1
Deploying Deep Learning SOC algorithm on S32K3
Vision
FreeMASTER

PMSM Control Workshop

Course Main Page
M1: Environment Setup
M2: PMSM and FOC
M3: System Partitioning
M4: PWM Modulation
M5: V/f Scalar Control
M6: Current Sensing
M7: Torque Control
M8: Speed Control
M9: Position Observer
M10: Sensorless Speed Control

BLDC Control Workshop

Course Main Page
1. Introduction
2. Application Partitioning
3. Input Commands
4. BLDC Motor Theory
5. Hall Sensors
6. Commutation
7. Commutation Algorithm
8. Power Stage Config
9. Open Loop Control
10. Speed Estimator
11. Closed Loop Control
12. Motor Control System

MATLAB EXPO

Thank you

MathWorks®