MATLAB EXPO

Deploying Motor Control Algorithms to a TI C2000™ Dual-core Microcontroller

Shang-Chuan Lee, MathWorks

Peeyush Kumar, MathWorks

MathWorks®

MathWorks
@MathWorks

Share the EXPO experience
#MATLABEXPO

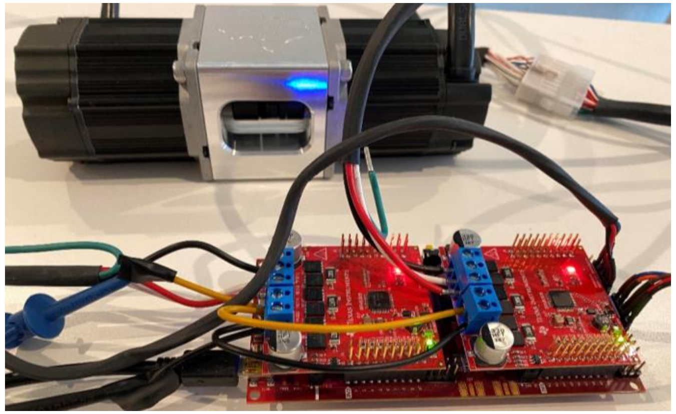
Poll Question

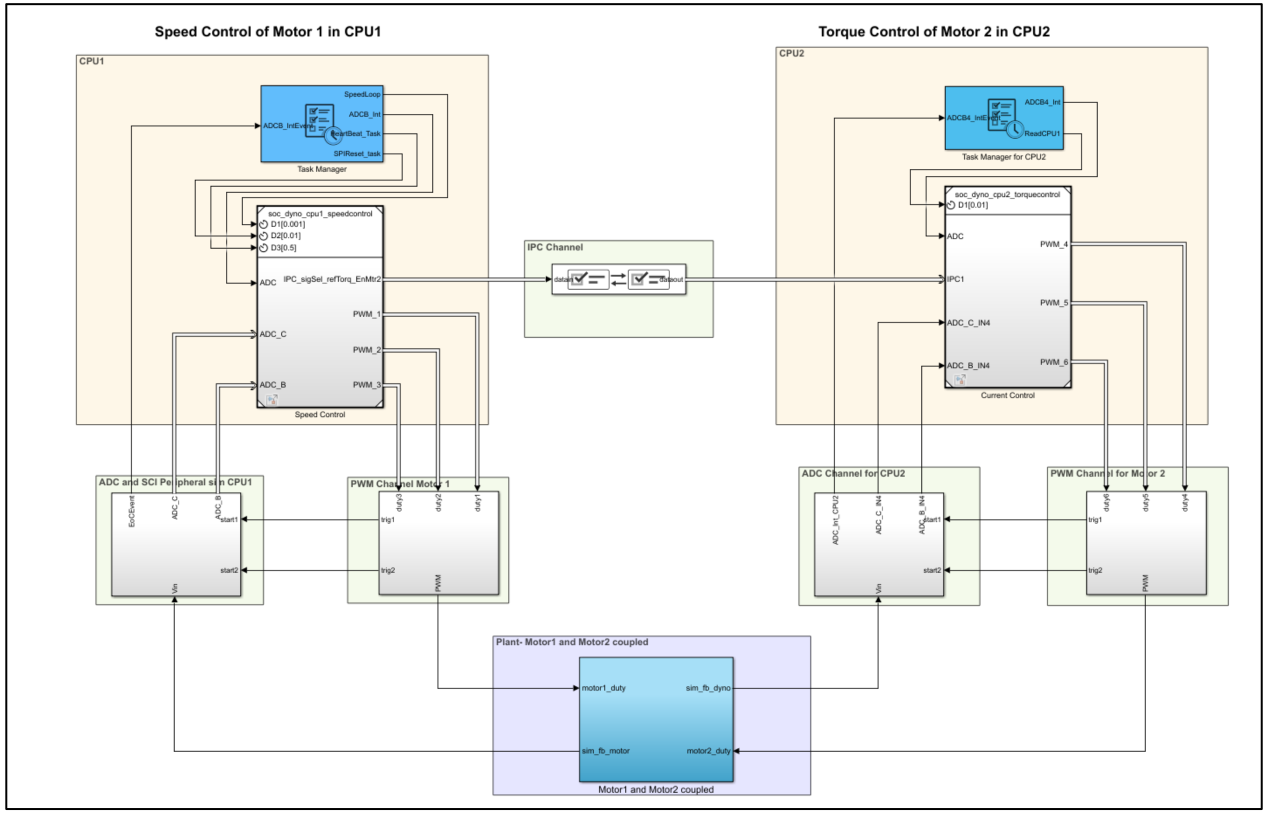
What is the most challenging part in your Motor Control Development process?

1. Motor control algorithm design
2. Write efficient code for embedded processors
3. Identify hardware and software architecture for multicore processors

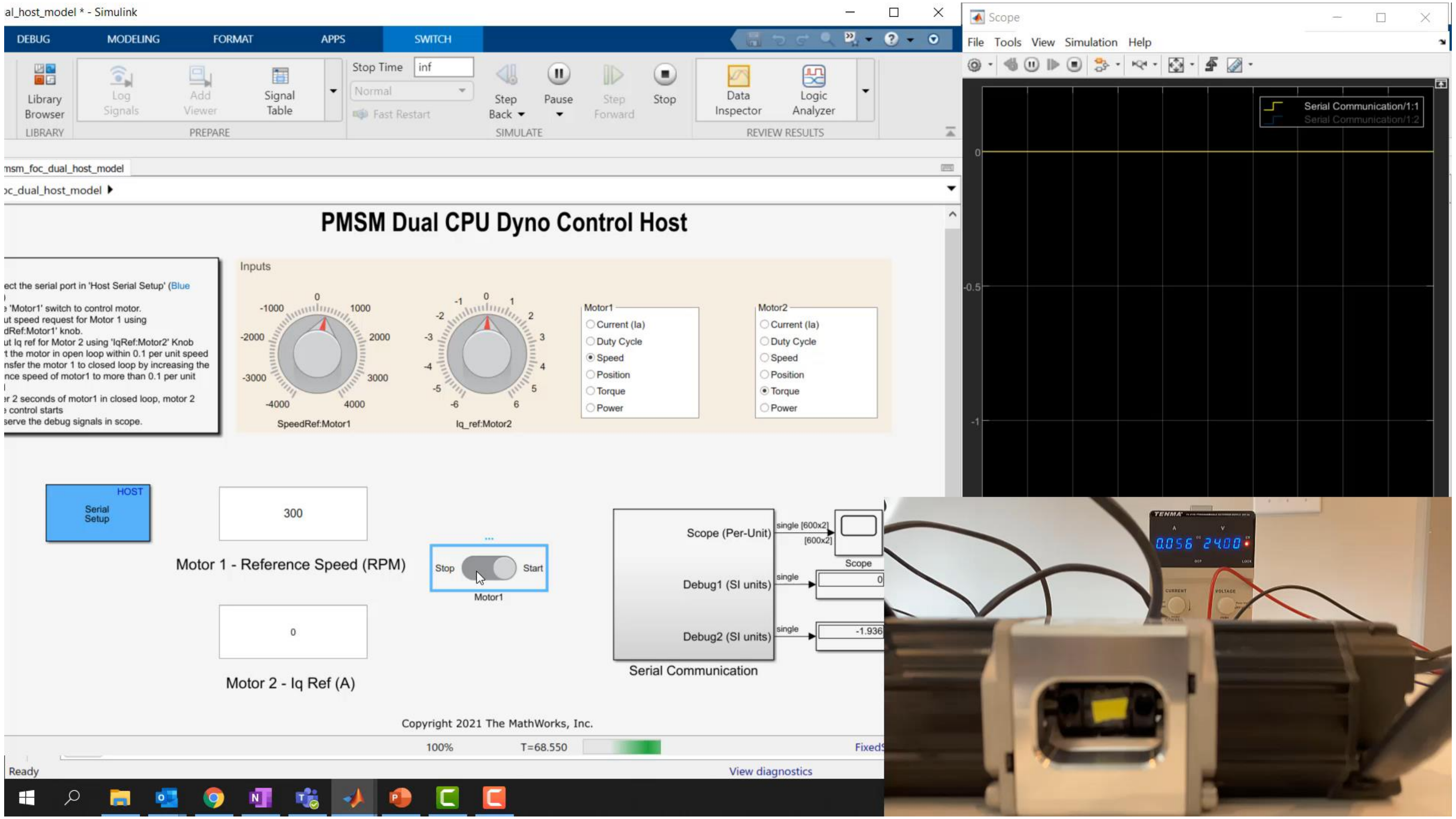
Key Takeaways

- Simulate sensorless field-oriented control (FOC) on a dyno setup
- Complete Model-Based Design workflow for multicore microcontroller
- Hardware component and device driver behavior simulation
  - Enhanced on-device profiling



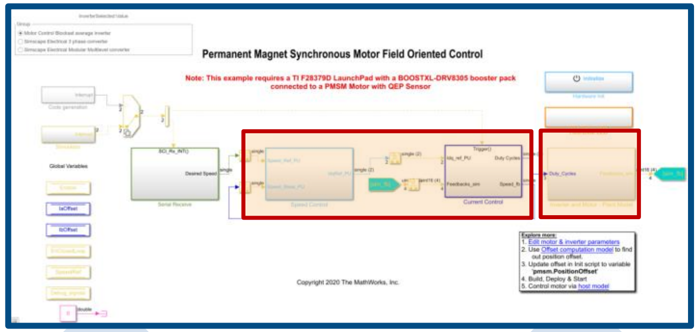


Dual CPUs PMSM Dyno Testing

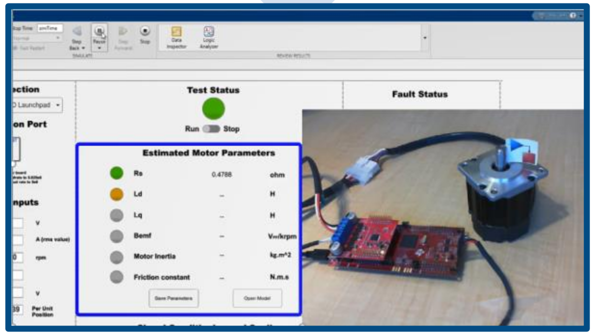


Workflow for Implementing Field-Oriented Control

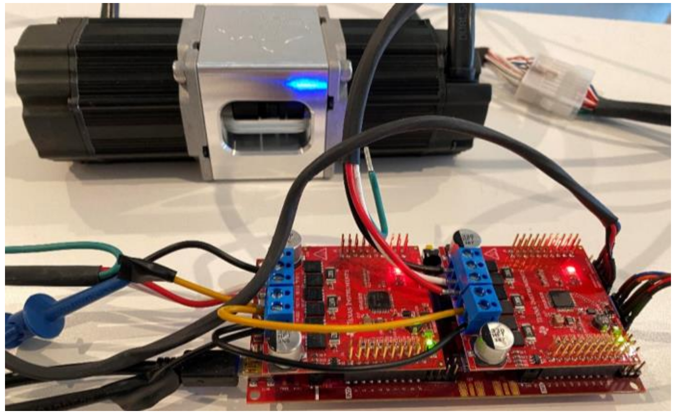
- Motor Control Blockset

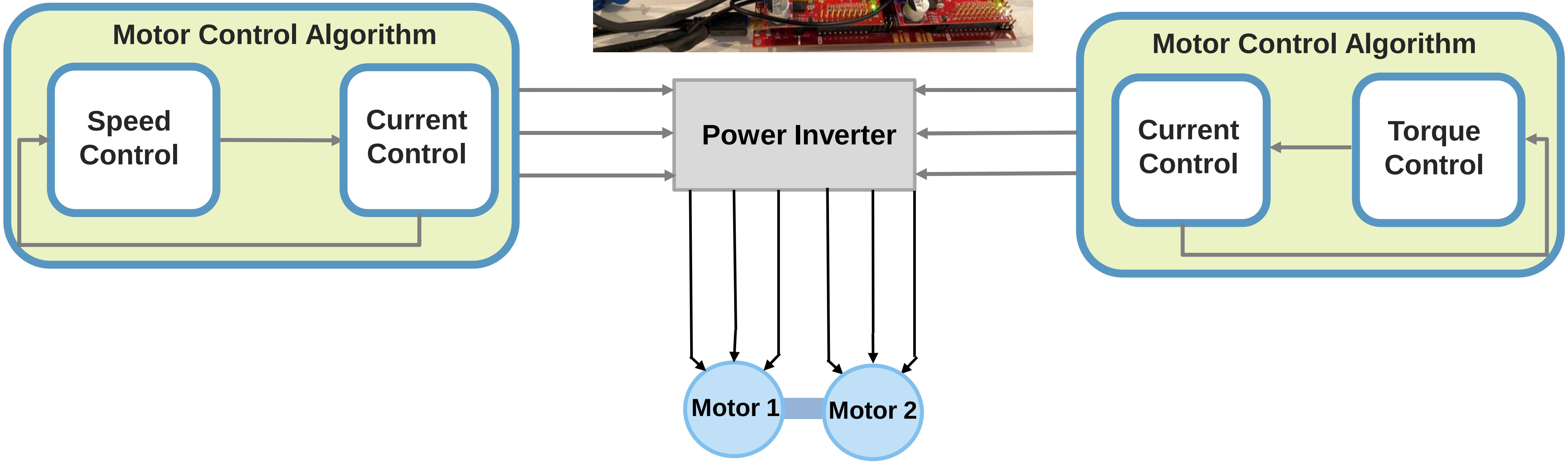


Calibrate Sensors → Estimate Motor Parameters → Model Motor & Inverter → Design Control Algorithm → Deploy & Validate

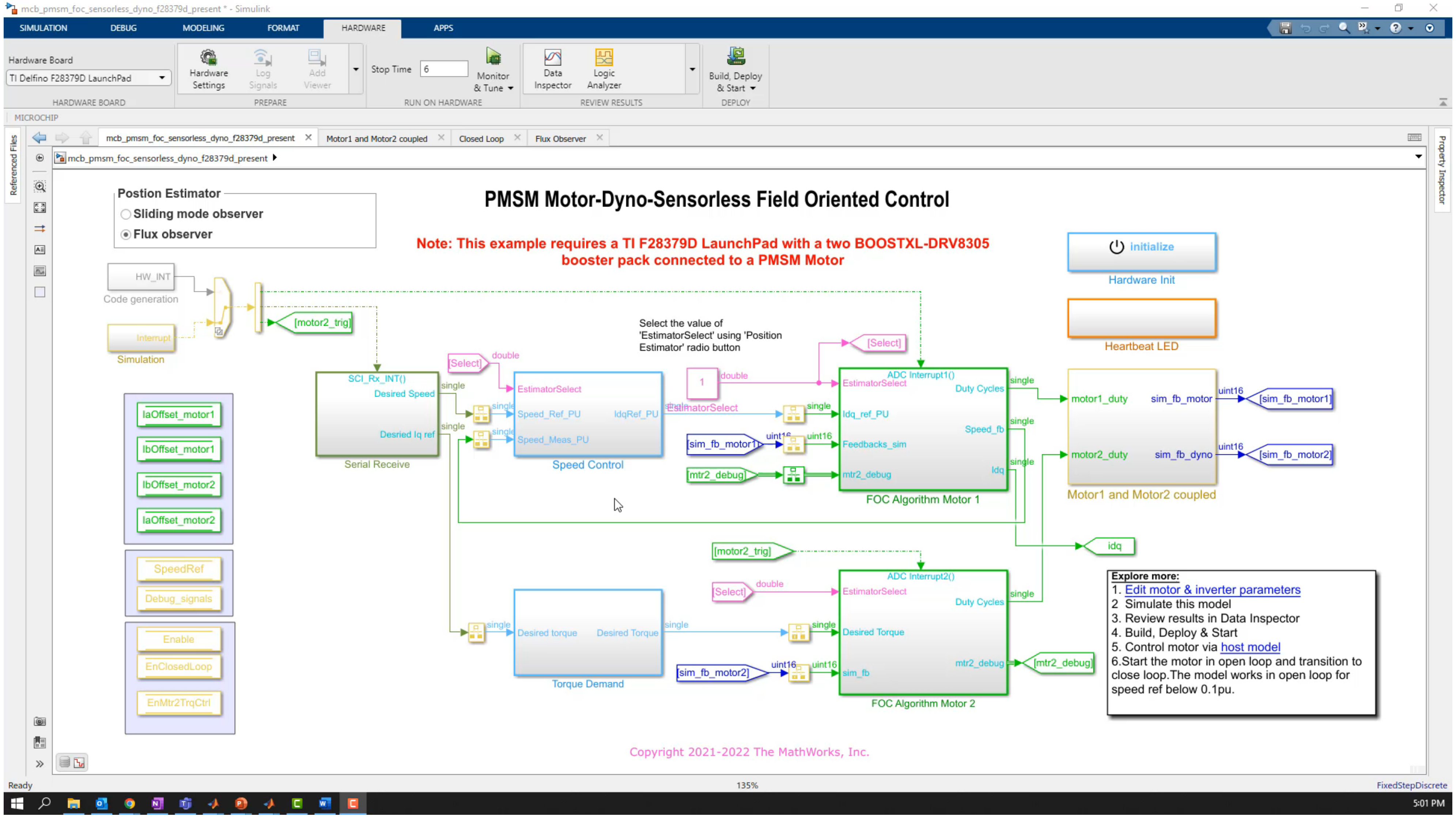


Simulate Motor Control System





Demo – PMSMs Dyno Model in FOC Sensorless Control



Challenges of Deployment on the Embedded Systems..

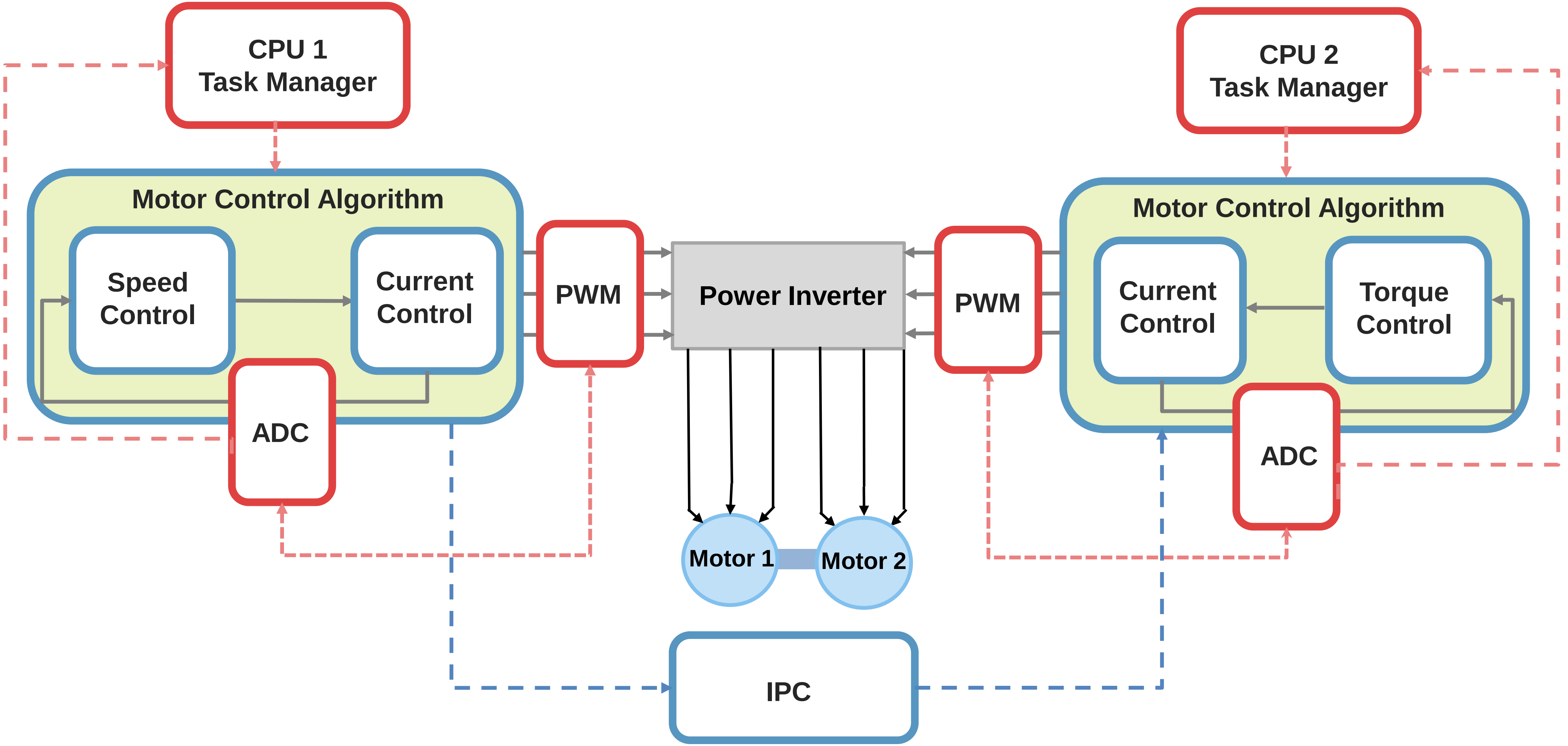
Model

- System requirement
  - TI C2000 dual-core processors
- Controller sample rate is 20kHz
  - Field-oriented control (FOC)
  - Sensorless control
  - Dyno setup (2 motors)
- No sensor delays in my model
- ADC-PWM synchronization

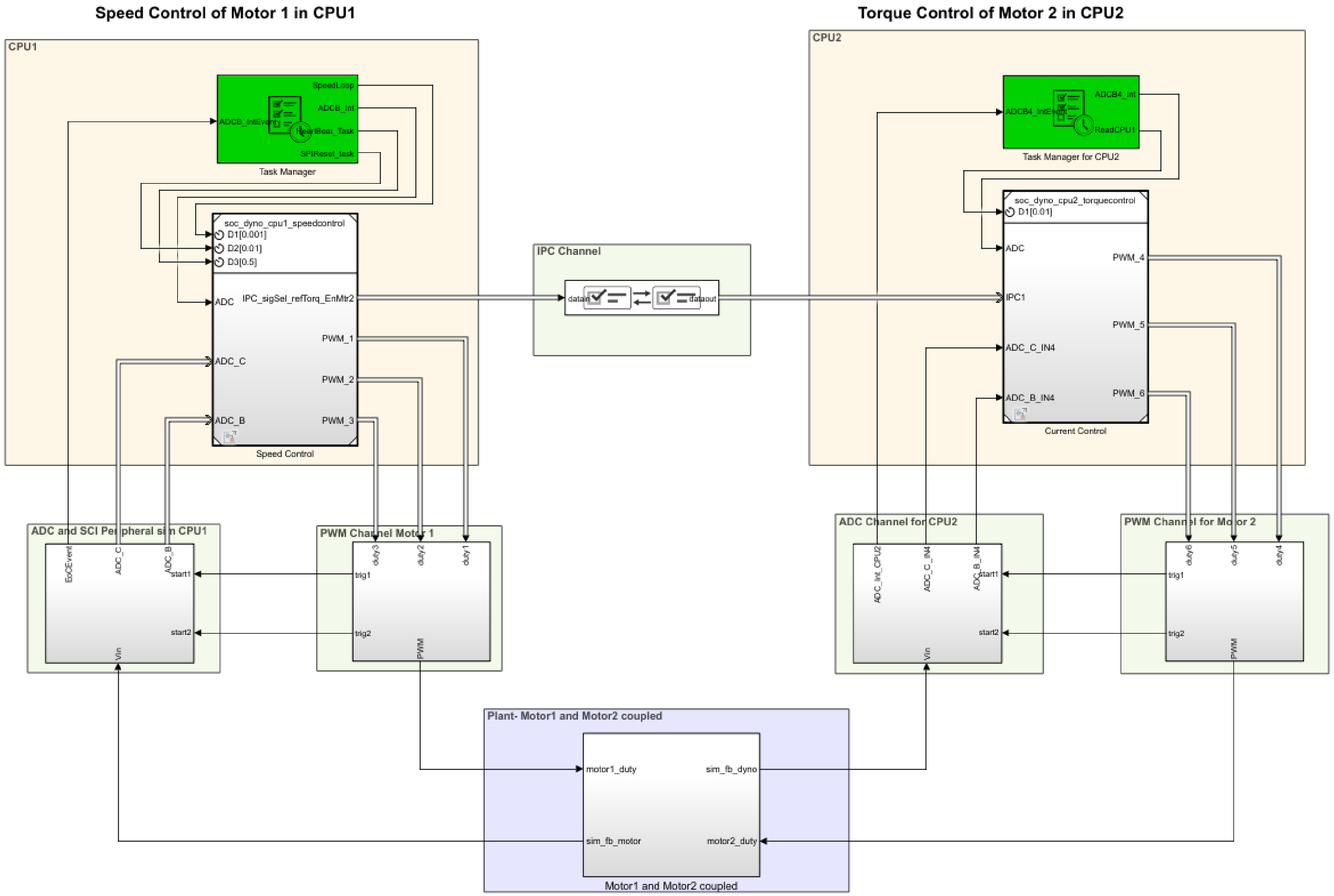
Multicore Processor

- How to implement and partition controls into two separated cores?
- How to communicate between CPU1 and CPU2?
- How to make sure task execution meets software requirement?

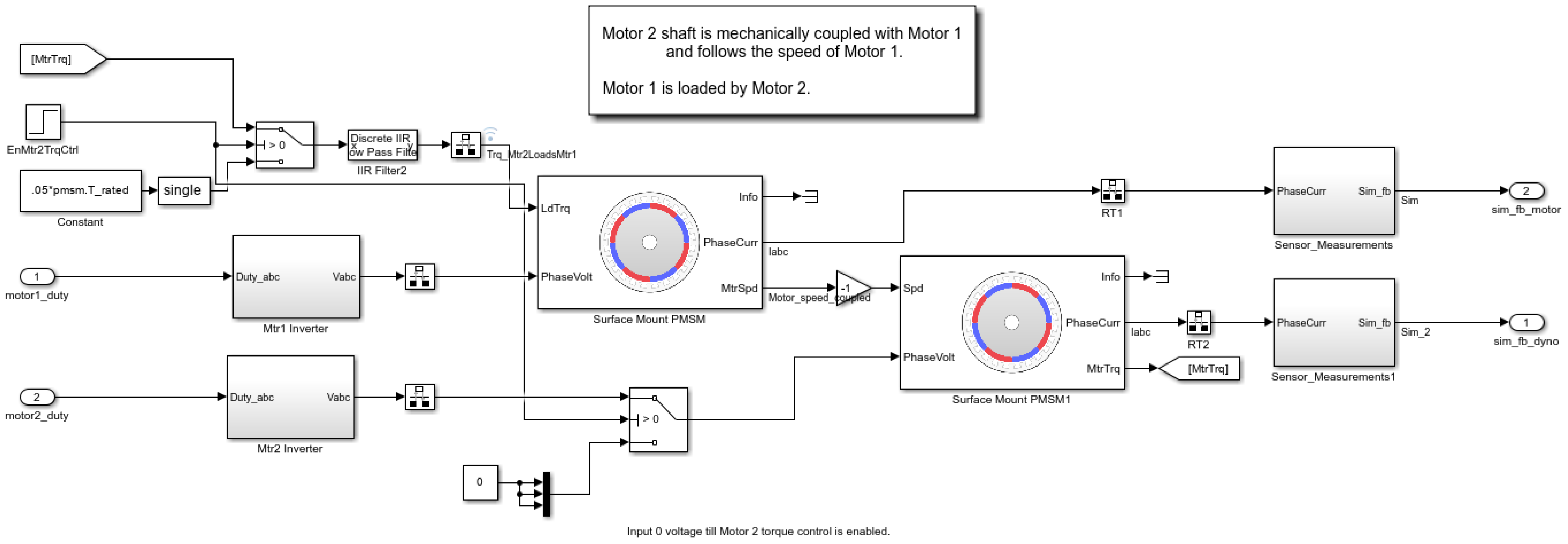
Simulate Motor Control System with Peripherals and Task Execution



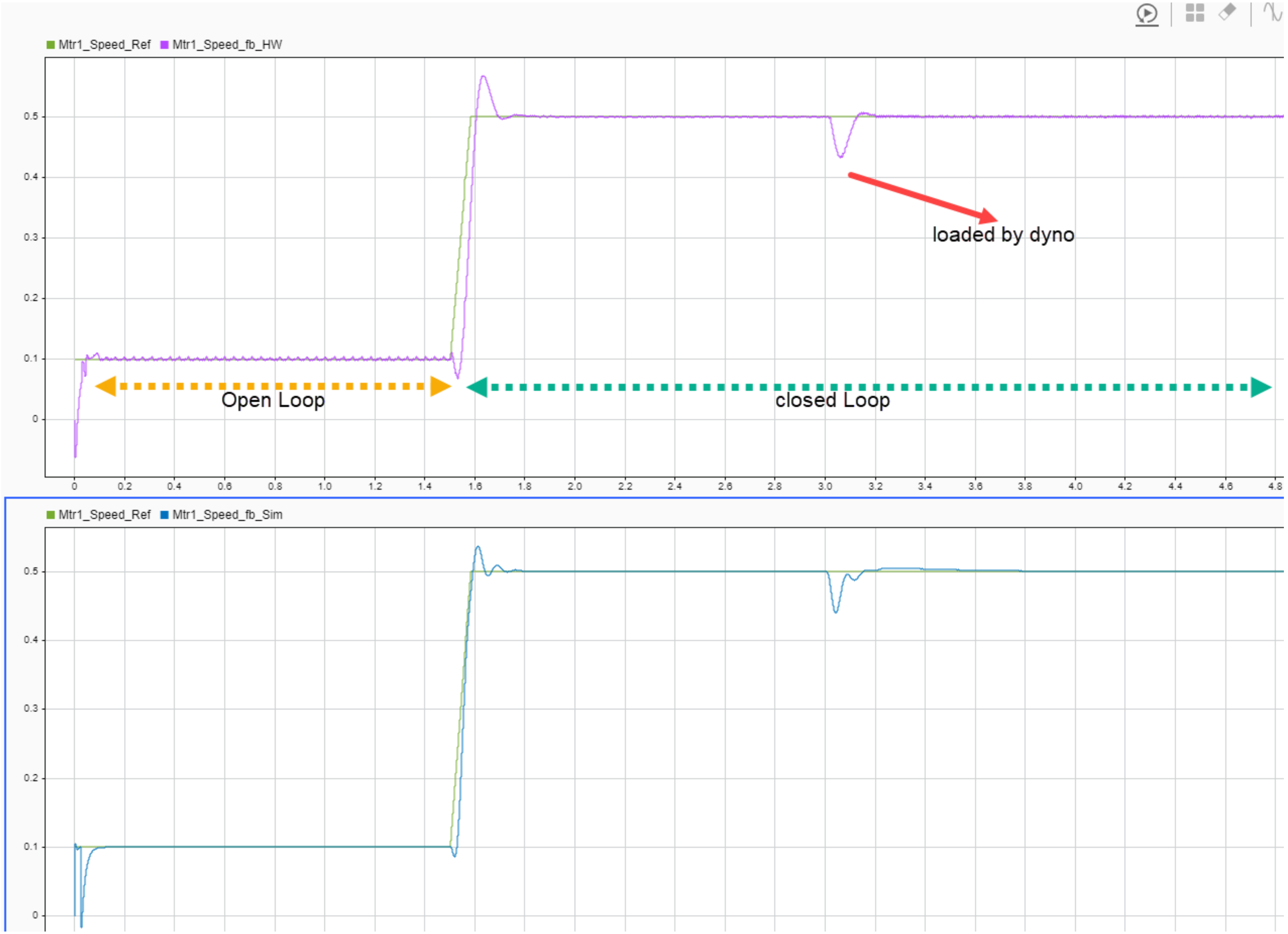
Model Multicore application Using SoC Blockset



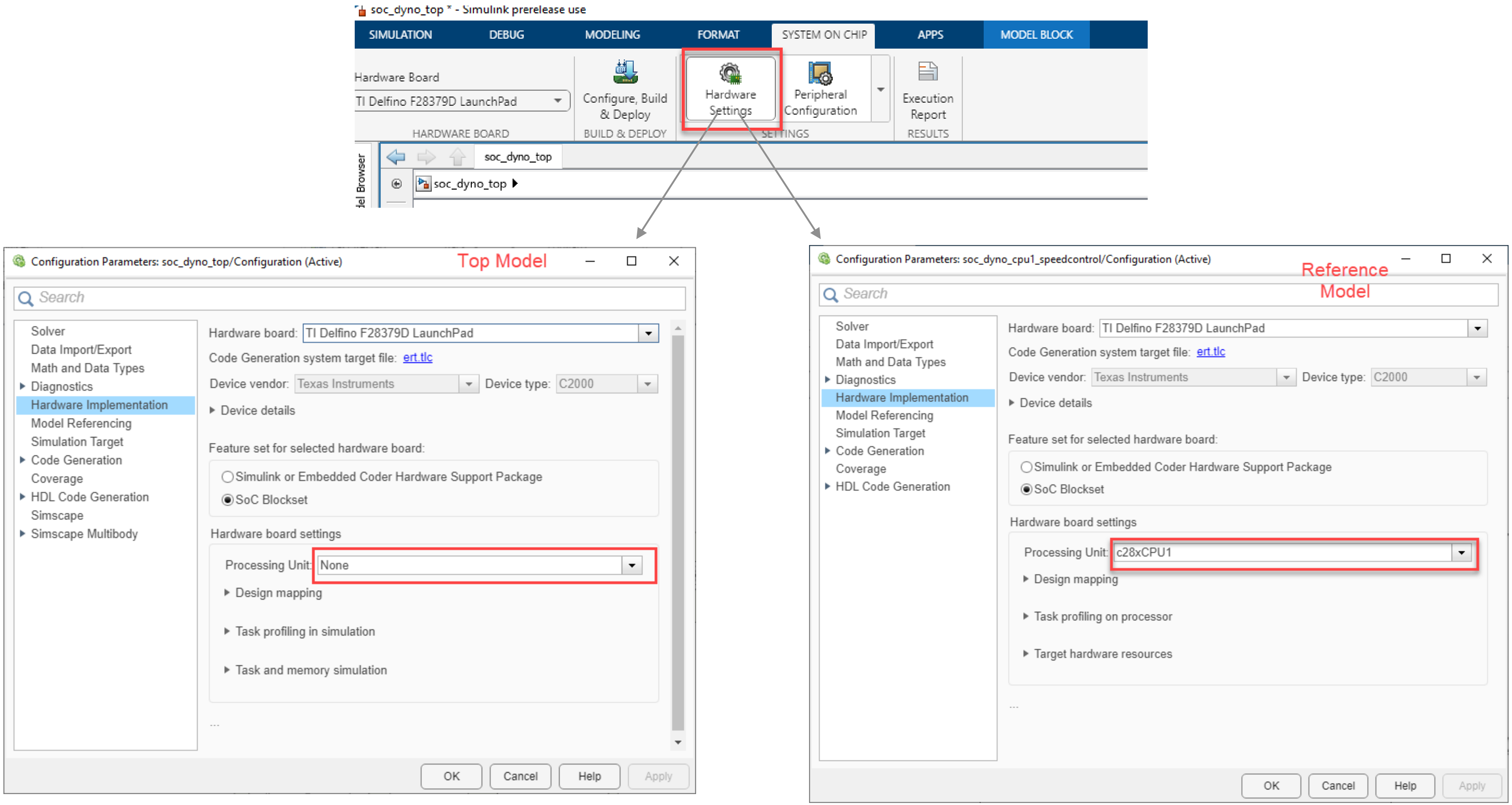
Plant Subsystem



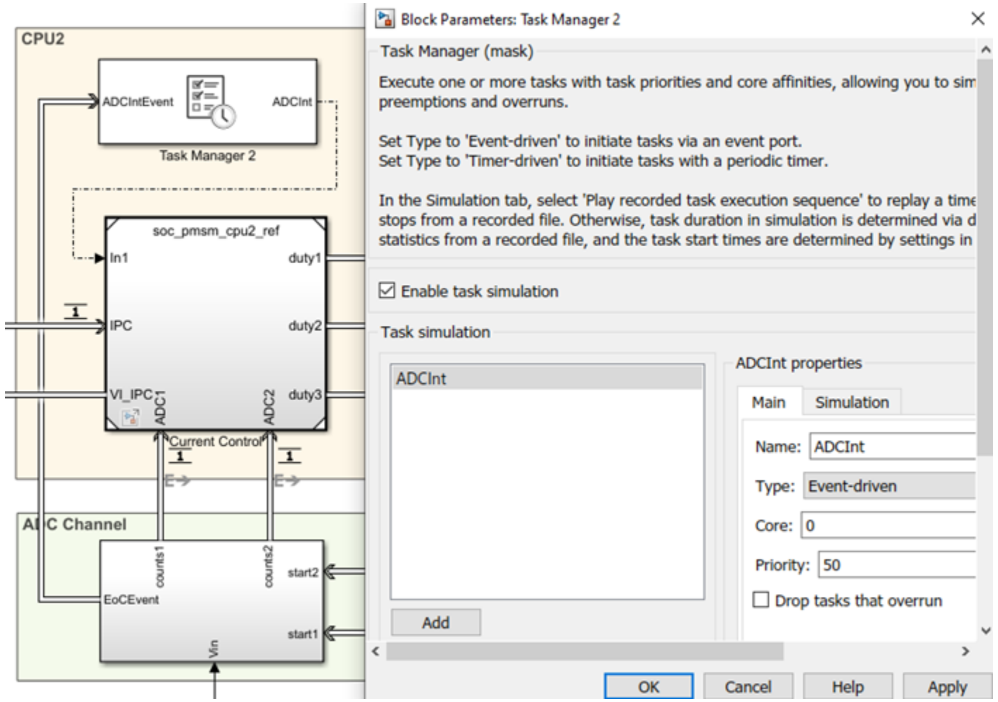
Hardware vs Simulation Analysis

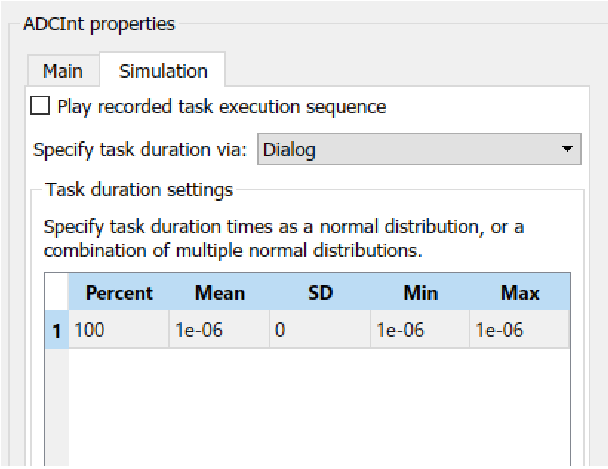


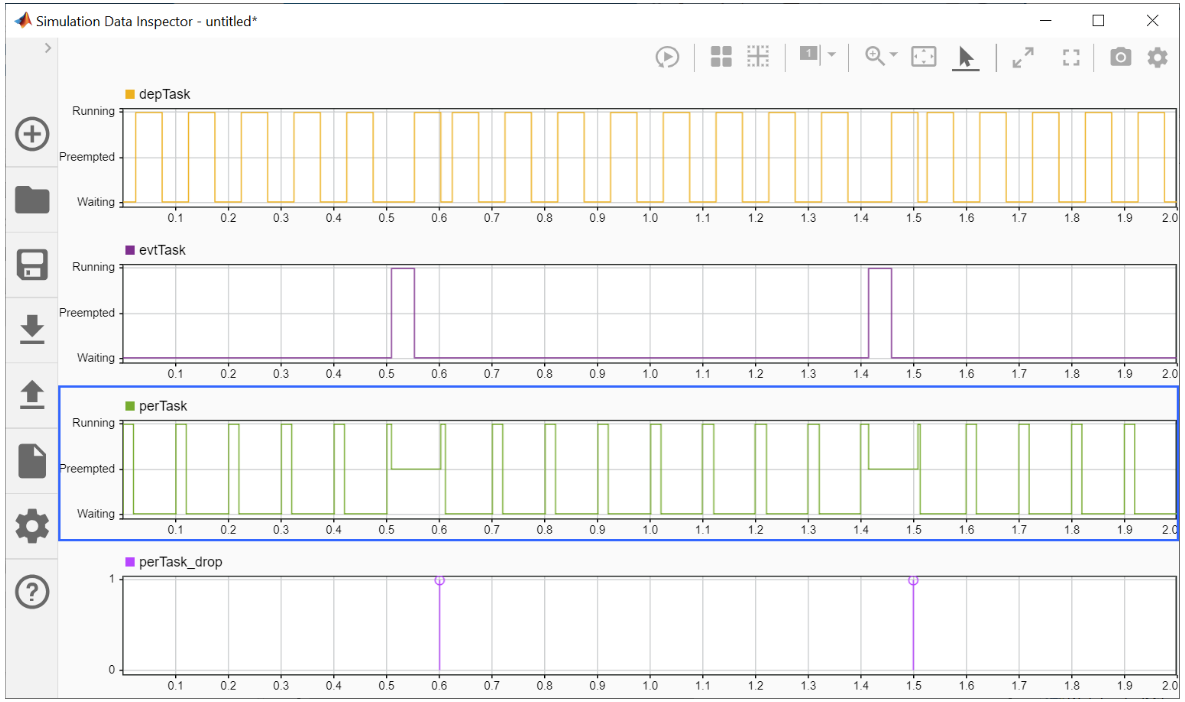
Model Configuration



Task Manager



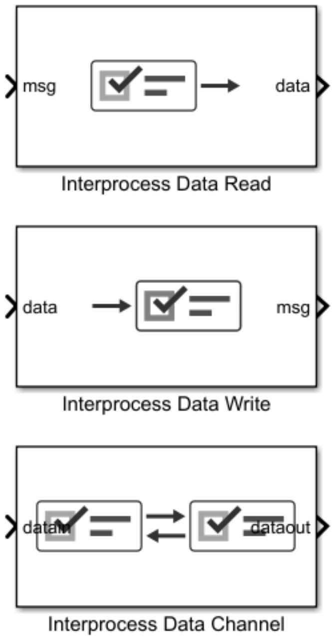


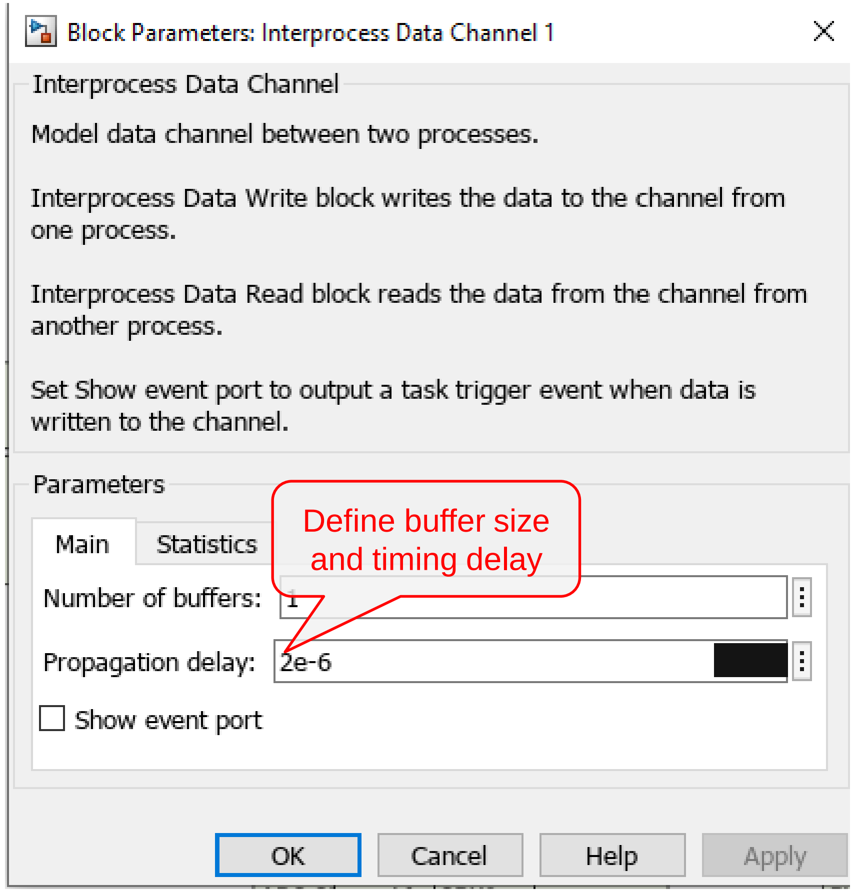


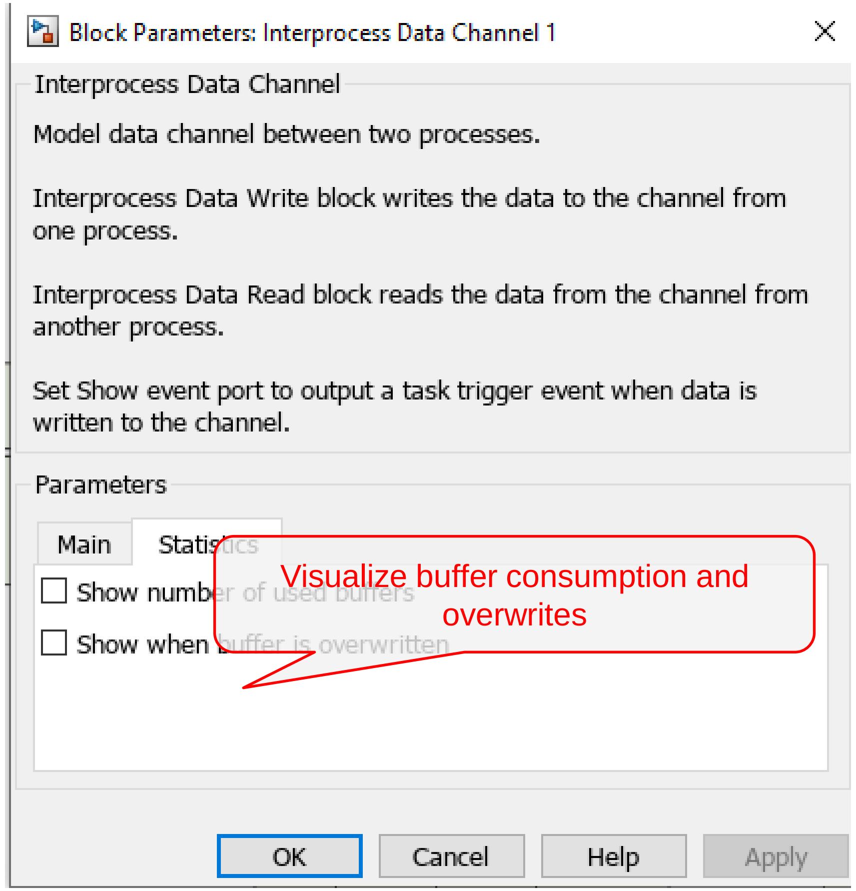
- Model/Simulate
  - Periodic/async tasks
  - Task priorities
  - Latencies
  - Duration as normal distribution

Inter-Processor Communication with **IPC Blocks**

- Model the communication buffering and delay

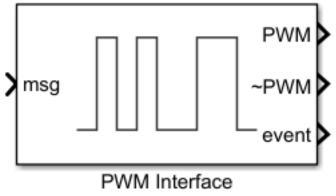


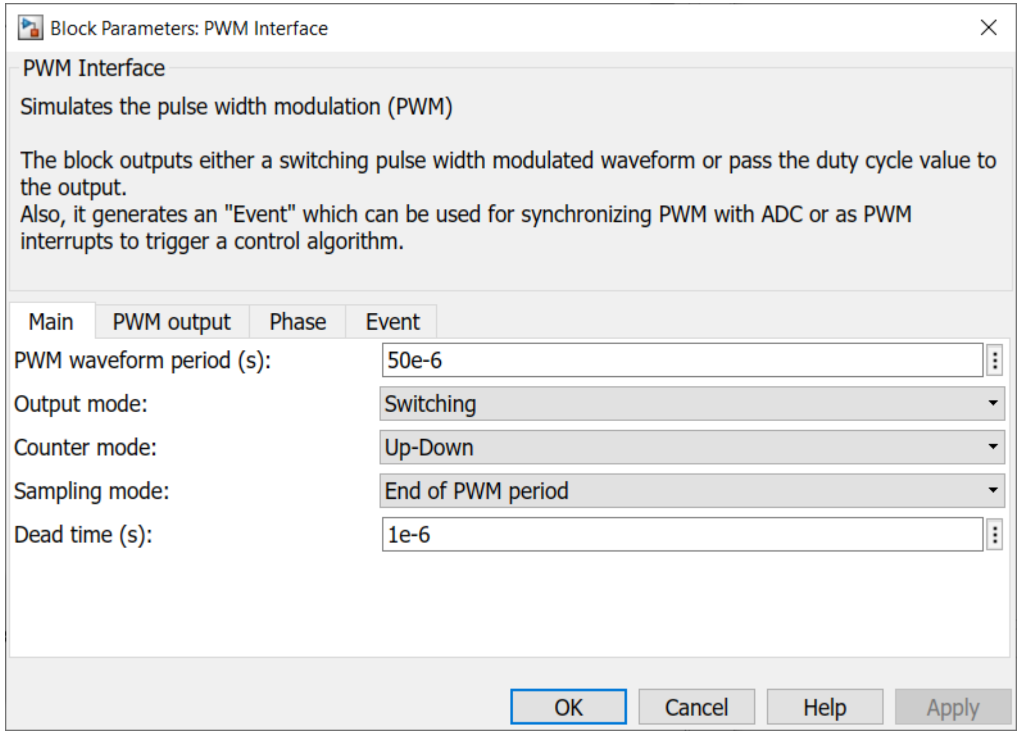


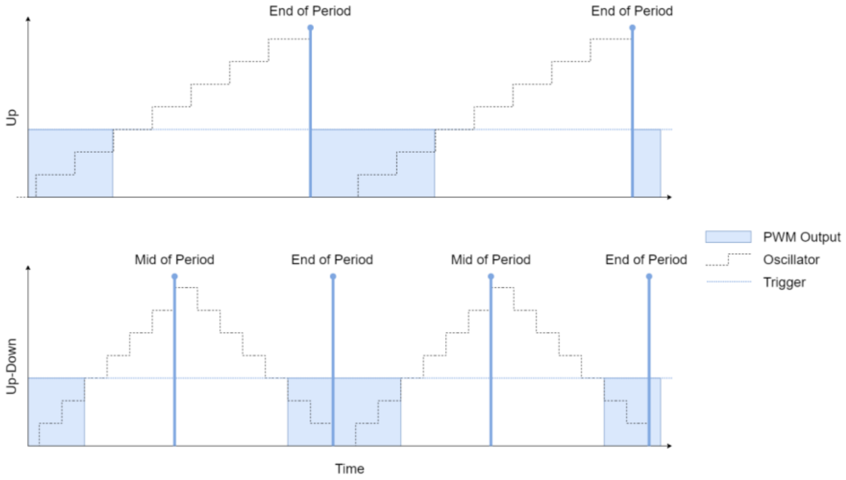


PWM Modeling

- PWM waveform simulation
- Event to synchronize with ADC or schedule a task

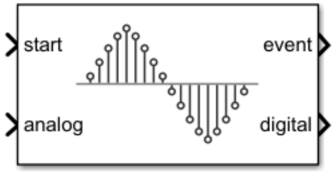




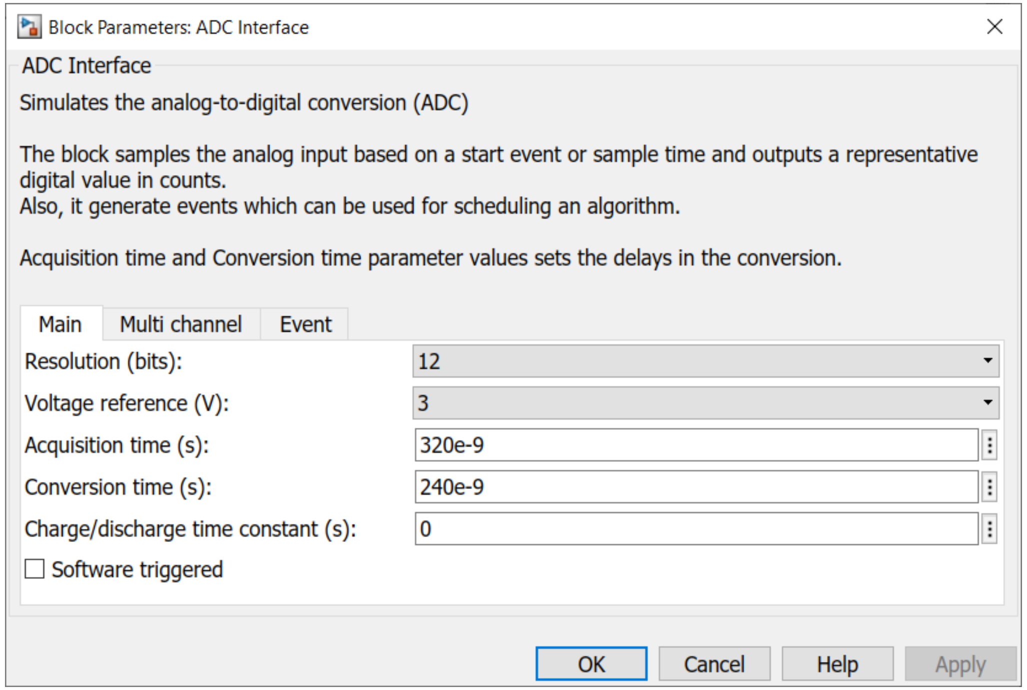


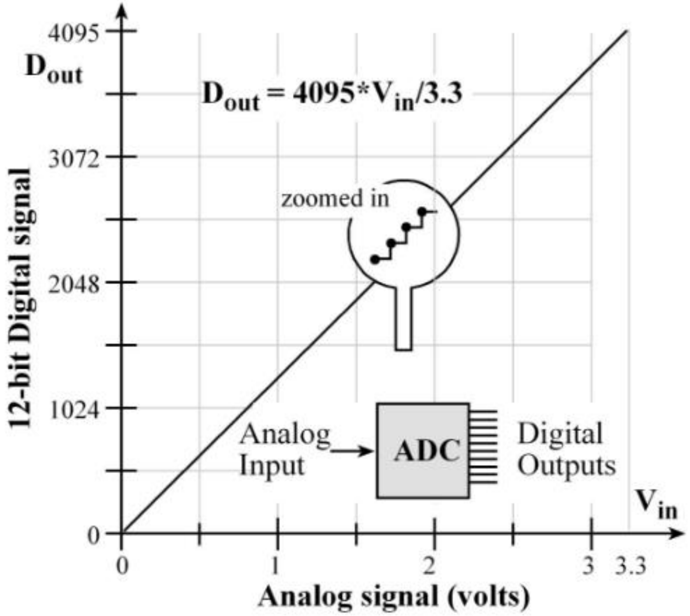
ADC Modeling

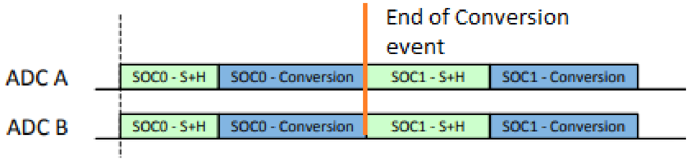
- Convert analog values to digital counts
- Model acquisition/conversion delays and trigger events



ADC Interface

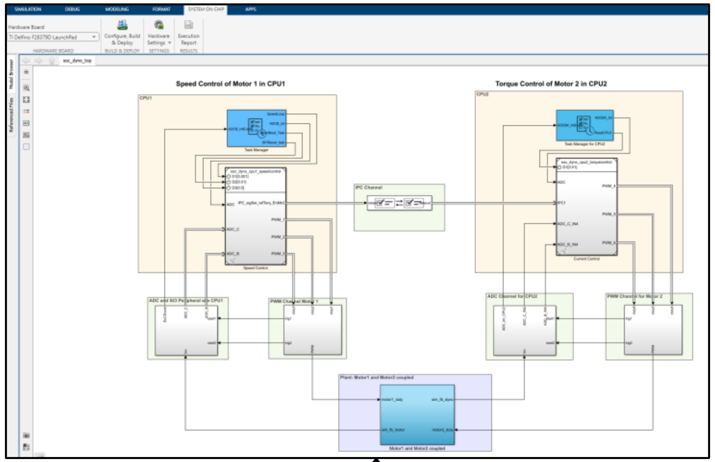


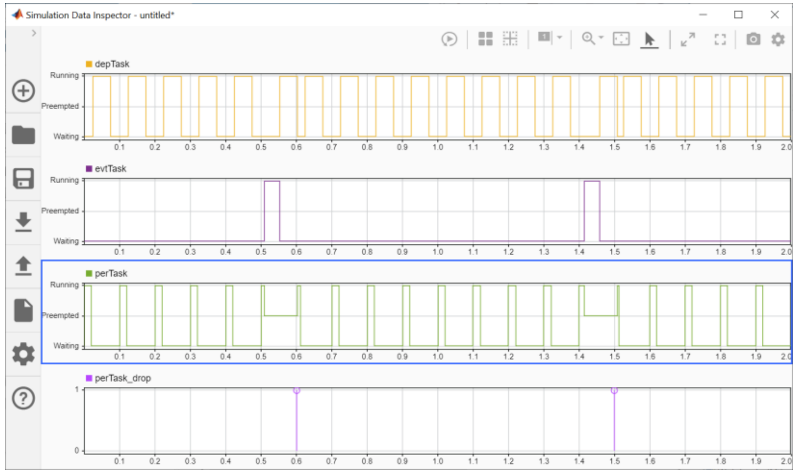


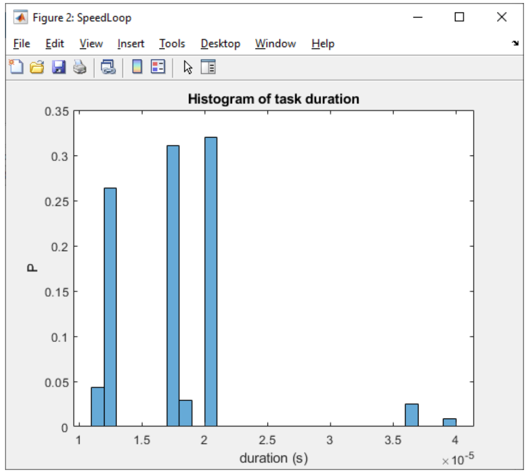


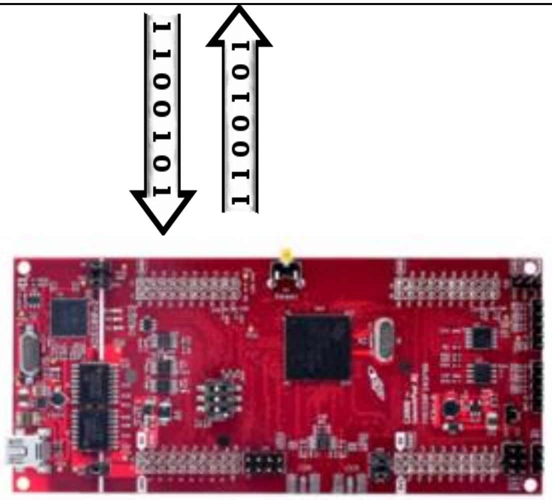
SoC Block set Key Functionalities – On-Device Profiling

- Real-time performance profiling on hardware, including
  - Task execution
  - CPU utilization
  - Communication buffering and delay
  - Real-time SDI view
  - Analysis report

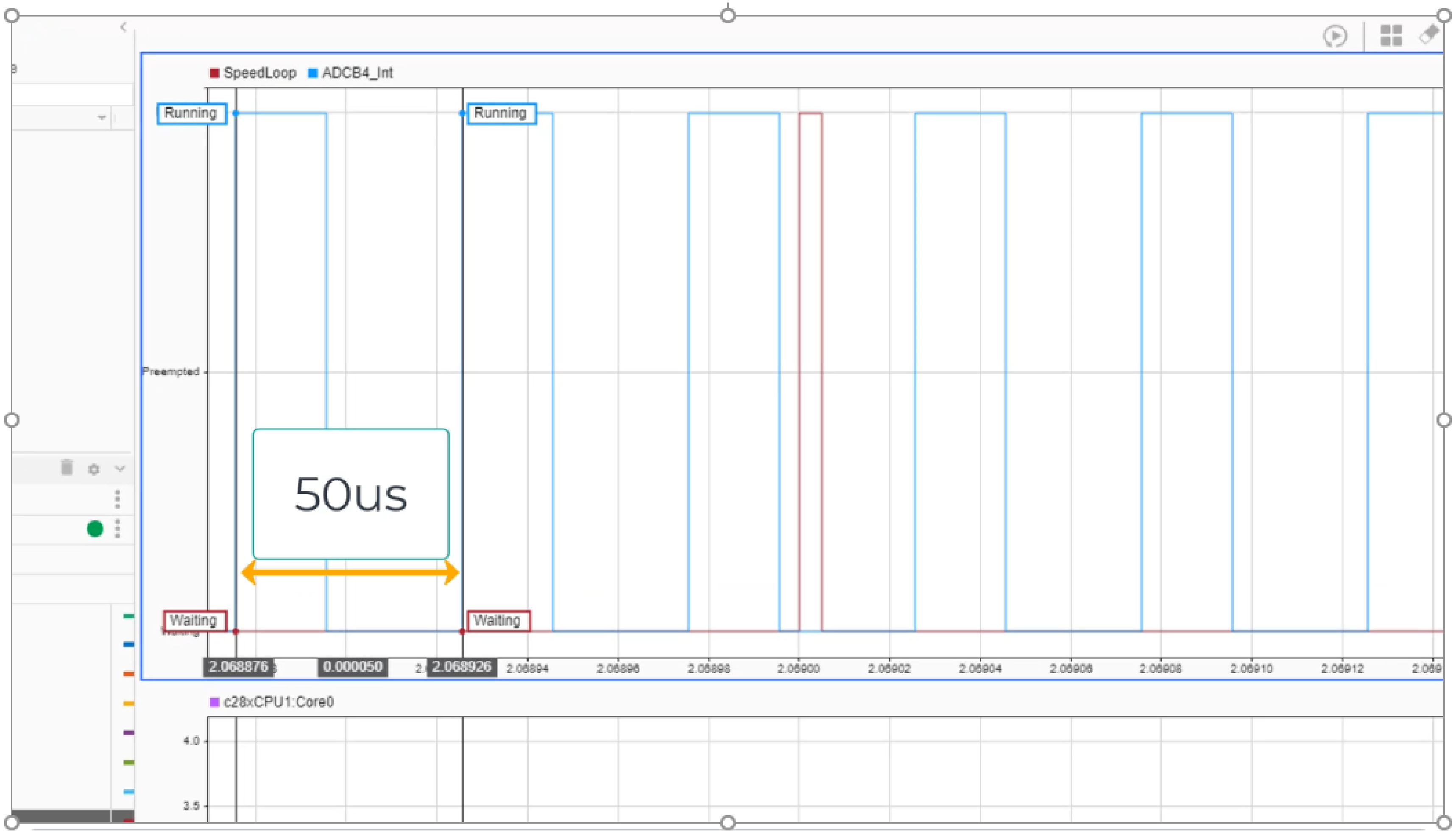








Profile Tasks in Simulation and Hardware



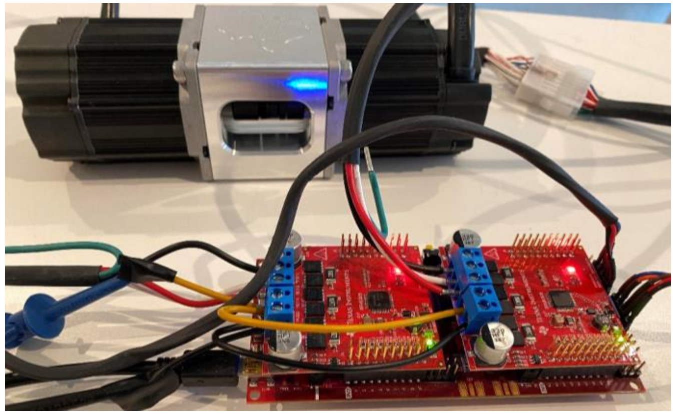
Poll Question

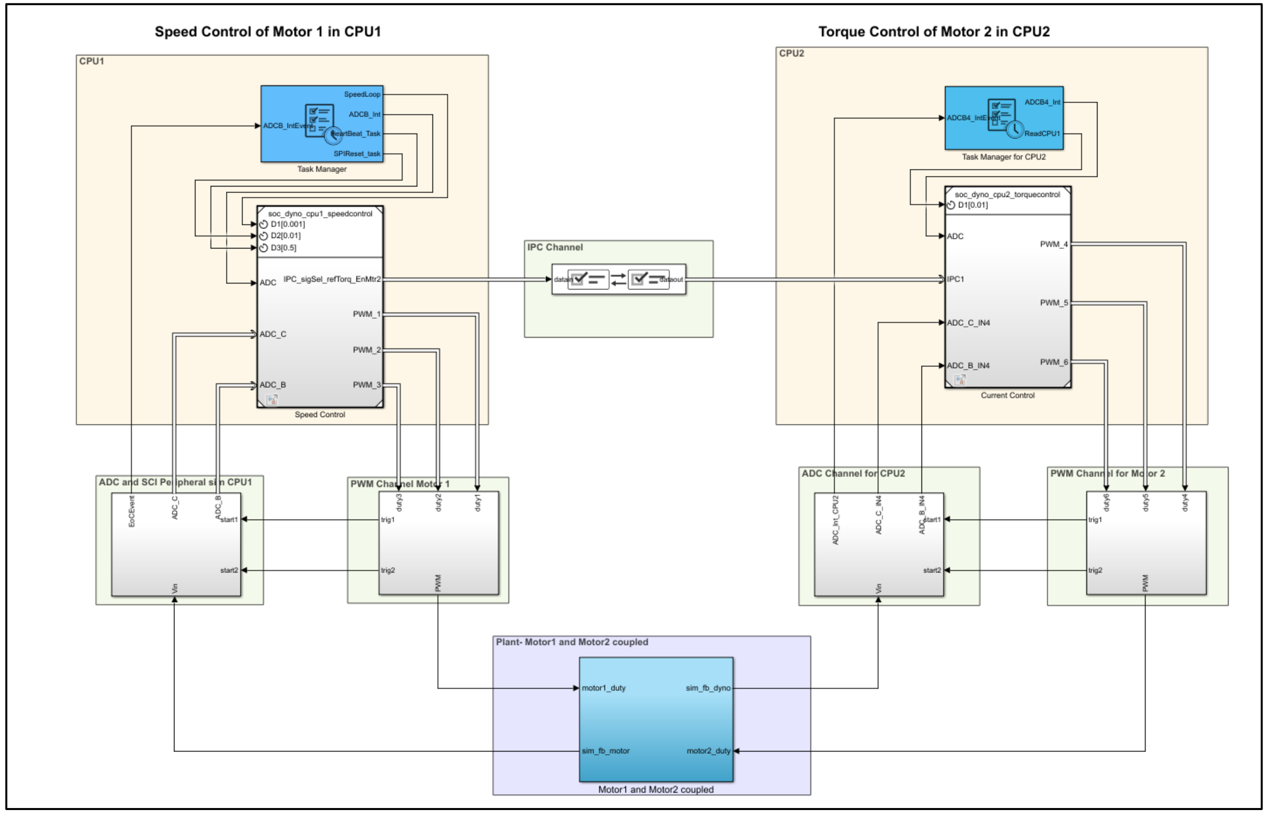
Which topics do you want to discuss more?

1. Motor control design
2. Partition Multicore Processors
3. Simulate PWM/ADC and perform device profiling

Wrap Up

- Simulate sensorless field-oriented control (FOC) on a dyno setup
- Complete Model-Based Design workflow for multicore microcontroller
- Hardware component and device driver behavior simulation
  - Enhanced on-device profiling

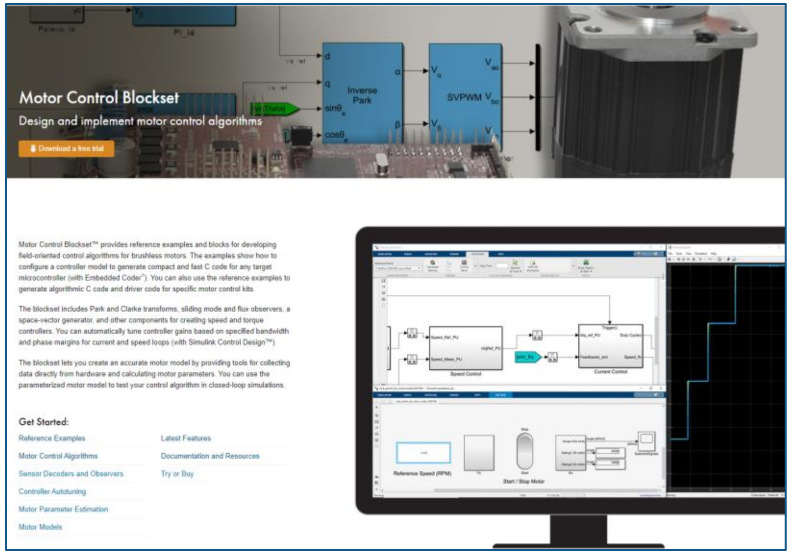




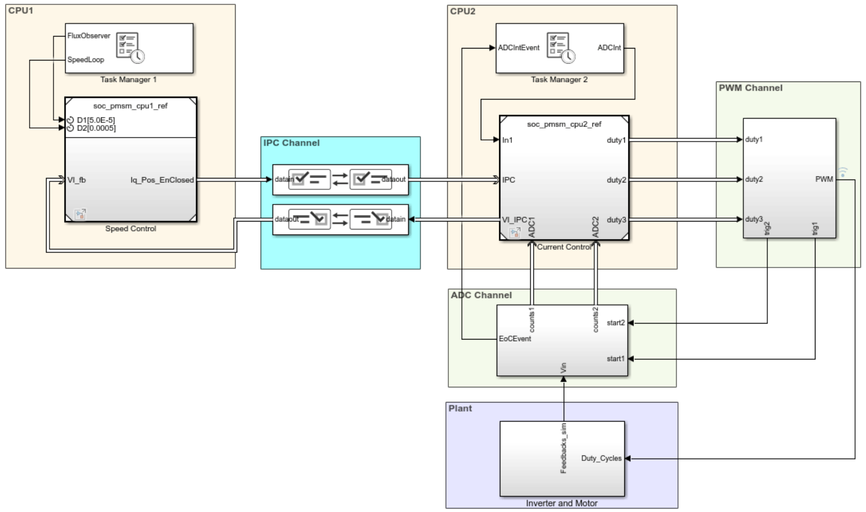
Learn More

- Recording webinars
  - Field Oriented Control Made Easy
  - Motor Control with TI Multicore MCUs Using Simulink
  - Implementing Motor and Power Electronics Control on an FPGA-Based SoC

- Shipping Demos
  - Partition Motor Control for Multiprocessor MCUs
  - Control PMSM Loaded with Dual Motor (Dyno)
  - Integrate MCU Scheduling and Peripherals in Motor Control Application



Field-Oriented Control on Dual CPU Processor



MATLAB EXPO

Thank you

MathWorks®