

WHITE PAPER

# Speed Up Digital Control Design of DC-DC Converters with Simulink

DC-DC converters include a variety of topologies to convert voltage level, including buck, boost, SEPIC, and Ćuk. Digital control enables the DC-DC converter to handle varying power supplies and loads while maintaining power quality over its operational range. This white paper explains how you can use system-level simulation with Simulink® for developing digital controls for DC-DC converters.

Simulink enables you to identify design problems early and optimize the system behavior by simulating and exploring multiple design options and operational cases from a single development environment. A system model serves as the basis for all development activities, from requirements development through design, implementation, and testing.

Using a single-ended primary-inductor converter (SEPIC) example, this white paper steps through the development of power converter digital control using modeling and simulation. The digital controller will adjust the brightness of LEDs on a DC/DC LED kit made by TI (Figure 1). The SEPIC and the embedded processor are also part of the kit.

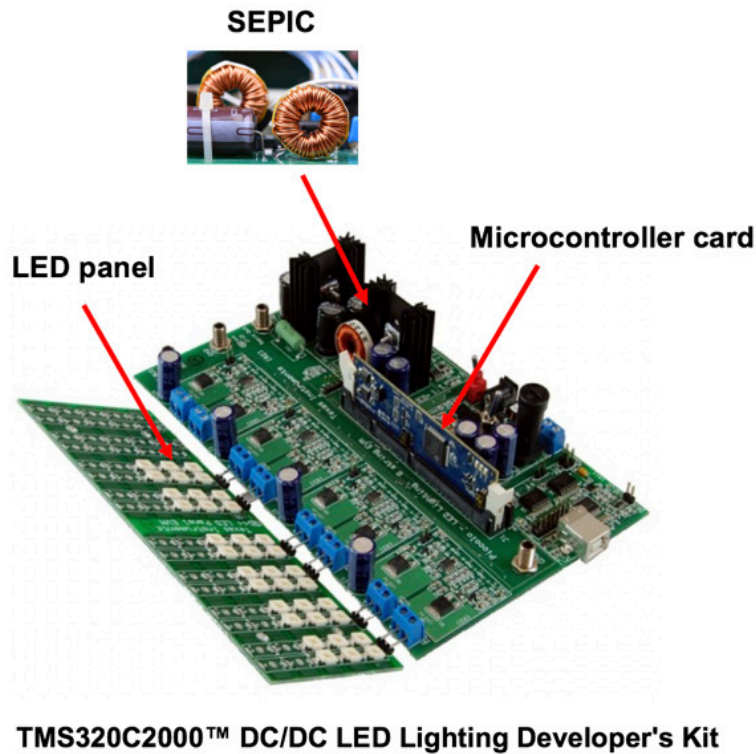


Figure 1. TI's DC/DC developer's kit, which includes the microcontroller card that will run the digital power converter controller.

You will also see how power electronics engineers can use Simulink to perform some common tasks, including:

- Modeling passive circuit elements, power semiconductors, and varying power sources and loads
- Simulating the converter in continuous and discontinuous conduction modes
- Determining power losses and simulating the thermal behavior of the converter
- Tuning the controller to meet design requirements such as rise time, overshoot, and settling time
- Generating C code from the controller model for implementation on TI's C2000™ microcontroller

*“Simulation and code generation enable us to turn changes around quickly and eliminate human errors in coding. Our algorithms typically work the first time, so we no longer waste a big part of our development cycle debugging code.”*

— Dr. Robert Turner, senior R&D engineer at ABB

#### Learn More

[Developing DC-DC Converter Control with Simulink - 4 Videos](#)

[ABB Accelerates the Delivery of Large-Scale, Grid-Connected Inverter Products with Model-Based Design - User Story](#)

## Modeling and Simulating a SEPIC

The SEPIC based on the TI hardware kit shown in Figure 1 can be modeled in Simscape™. Once the major parts of the model are in place, you can use simulation to understand the overall system behavior in continuous and discontinuous conduction modes. Additionally, the model can help you evaluate different converter configurations, explore the effects of different switching frequencies, and understand how component thermal characteristics affect the converter's efficiency.

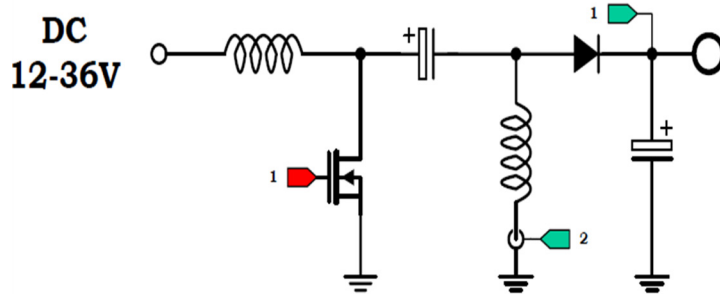
Simscape provides blocks and libraries of electrical components, enabling you to create models of electrical systems within the Simulink environment. Simscape models can contain variants of components at different levels of fidelity, letting you start with simple linear representations and work up to complex nonlinear behaviors.

The SEPIC is modeled using blocks for individual electrical components, such as a voltage source, resistor, inductor, and capacitor (Figure 2). These blocks can be parameterized, and the connections between them represent physical connections in the real system that support bidirectional flow of power. Because the hardware is already available, you can use the specifications provided with the TI kit to determine block parameters such as capacitance and inductance. Simulating the model with the TI values, you can verify whether the converter operates in continuous or discontinuous conduction mode. Note that the model is simulated in open loop with constant duty cycle.

*“Running simulations of the plant models we built with Simulink and Simscape Electrical enabled us to improve and optimize our hardware design before it was finalized. The simulations eliminated several iterations of preprototype hardware.”*

— Dr. Jakub Vonkomer, VONSCH

### SEPIC schematic



### Equivalent Simscape model

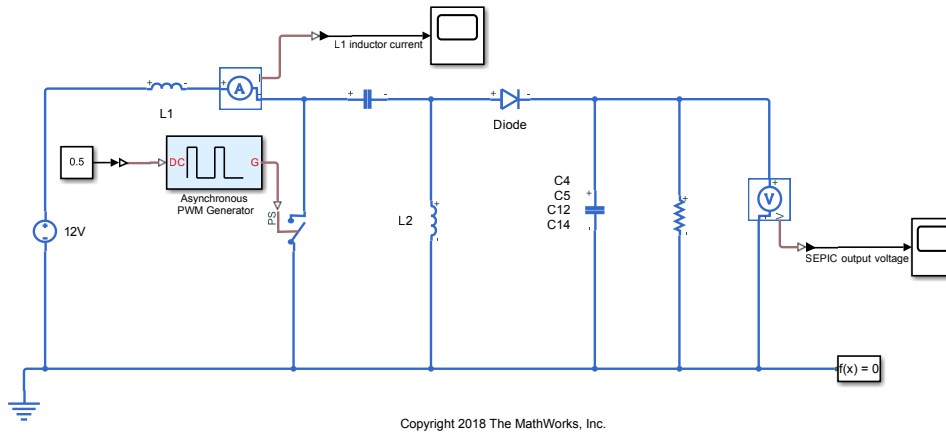


Figure 2. The schematic (top) and the equivalent Simscape model (bottom) of the SEPIC.

You may want to conduct additional analyses to further evaluate the converter design. For example, rerunning the simulation with smaller inductor values or with larger load may force the converter to operate in discontinuous conduction mode (DCM) depending on the actual values selected (Figure 3). Another observation in this model is that the switch is assumed to be ideal—that is, the nonlinear dynamics associated with the turn-on and turn-off times are not modeled in this example.

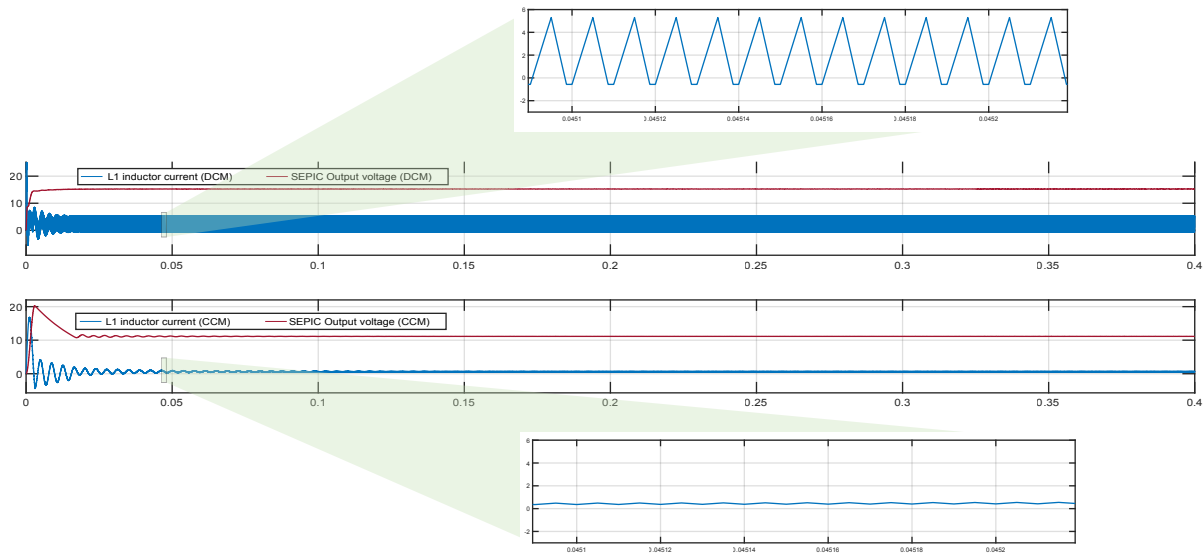


Figure 3. Simulation results of the SEPIC in discontinuous (top) and continuous (bottom) conduction mode.

**Learn More**

[Buck Boost Converter - Example](#)

[VONSCH Speeds the Development of Control Systems for Solar Inverters and Battery Chargers - User Story](#)

**Determining Power Losses and Thermal Behavior of the Converter**

As mentioned earlier, Simscape models can contain variants of components at different levels of fidelity. You can incorporate and analyze the nonlinearities in the switching behavior of the SEPIC by replacing the ideal switch with a higher-fidelity block.

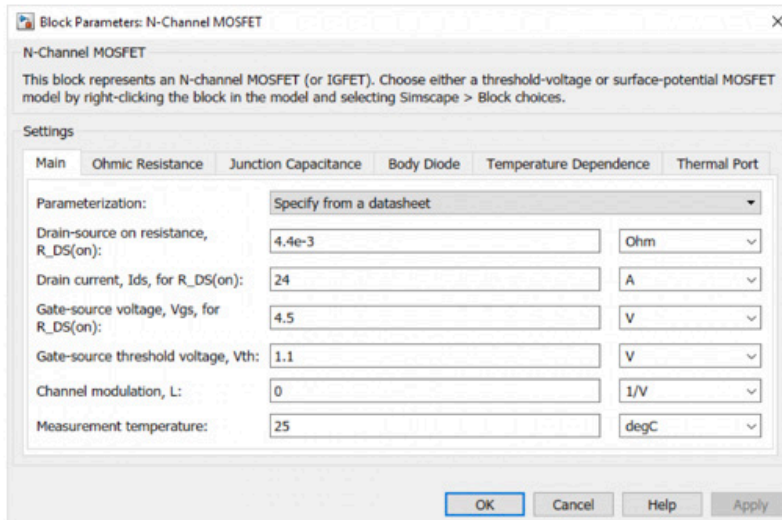
With a few small changes, the simulation model can be used for determining power losses and understanding the thermal behavior of the SEPIC. To model the nonlinear switching behavior, you can replace the ideal switch block with the N-Channel MOSFET block from Simscape

*“With Simscape Electrical we created an integrated power system model that connects electrical and thermal domains, so we get the whole picture during our mission-level simulations. If we need to model the motors that turn the solar arrays, we have the capability to integrate those mechanical components, too.”*

— Hector Hernandez, Lockheed Martin

Electrical™. To model the thermal dynamics of the converter, you can use Simscape, which lets you simulate multi-domain models in the Simulink environment.

To parameterize the MOSFET block, you can use the datasheet for the MOSFET used in the TI kit (Figure 4). Both static and dynamic characteristics from the datasheet are used to parameterize the block.



$T_A = 25^\circ\text{C}$		TYPICAL VALUE		UNIT
$V_{DS}$	Drain-to-Source Voltage	25		V
$Q_g$	Gate Charge Total (4.5 V)	6.2		nC
$Q_{gd}$	Gate Charge Gate-to-Drain	1.1		nC
$R_{DS(on)}$	Drain-to-Source On Resistance	$V_{GS} = 3\text{ V}$	5.4	m $\Omega$
		$V_{GS} = 4.5\text{ V}$	4.4	
		$V_{GS} = 8\text{ V}$	3.8	
$V_{th}$	Threshold Voltage	1.1		V

Figure 4. Parameterization of the MOSFET block using TI's datasheet.

Once the MOSFET is parameterized, you can compare the characteristic curves of the MOSFET block (steady-state, transfer, and gate charge characteristics) with the graphs provided in the datasheet to ensure a close match. As shown in the bottom row of Figure 5, the simulated and actual characteristics curves are very close, verifying that the behavior of the MOSFET model is close to that of the real hardware.

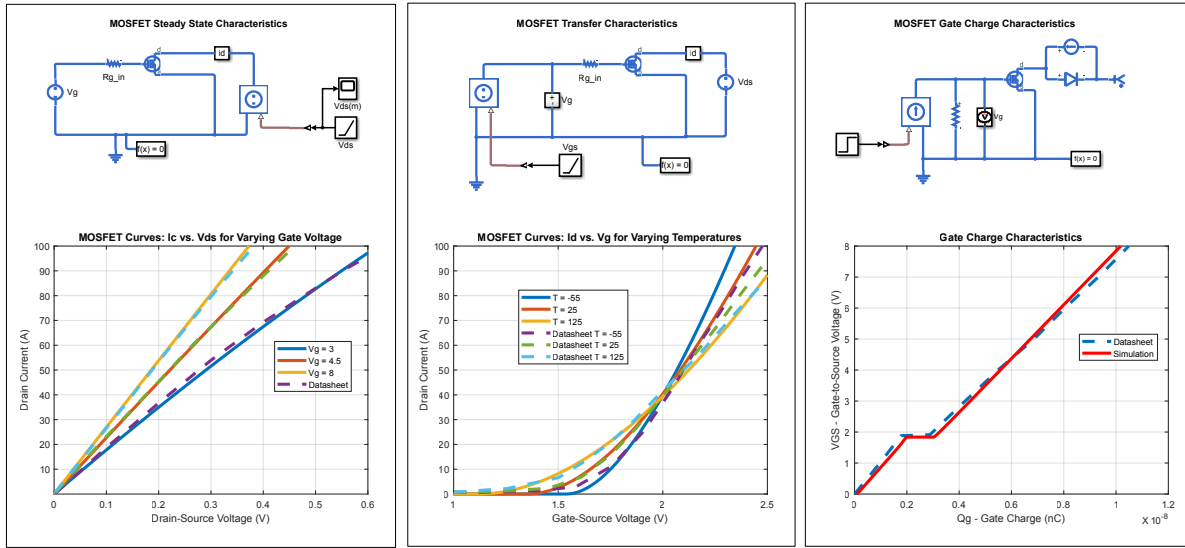


Figure 5. Comparing the characteristics of the parameterized MOSFET block in Simscape Electrical and the actual characteristics from the datasheet. The top row shows the models used to generate the plots in the second row. The responses match closely, indicating good parameterization.

The parameterized N-Channel MOSFET block can then be added to the SEPIC model, replacing the ideal switch block (Figure 6). By enabling the thermal port of the block, you can model and simulate both the electrical and thermal dynamics of the converter simultaneously. You can calculate the dissipated power losses of the MOSFET in the model, based on logged simulation data. Note that the model is still operating in open loop with a fixed duty cycle.

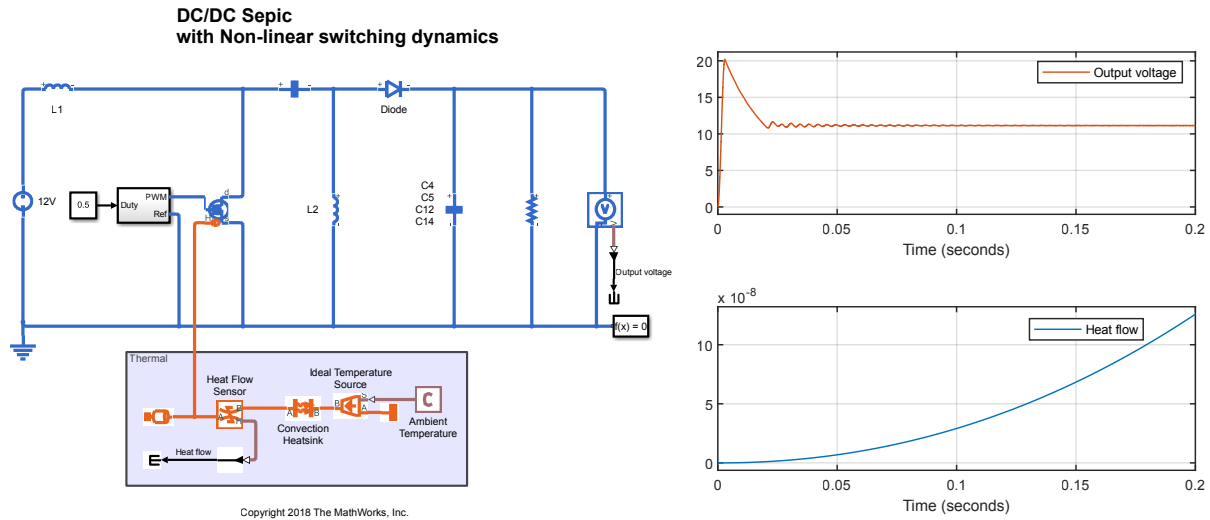


Figure 6. The SEPIC model with nonlinear switching dynamics (top), with plots of the electrical and thermal dynamics simulation results (bottom).

**Learn More**

[Lockheed Martin Simulates Orion Spacecraft Missions Using a Multidomain Power System Model - User Story](#)

[TI's N-Channel MOSFET - Datasheet](#)

[Calculating Dissipated Power Losses - Documentation](#)

## Designing a Digital Controller

To design a digital PID controller that regulates the output voltage of the converter, Simulink enables you to extract a linear model from the discontinuous system and automatically tune the controller gains.

Figure 7 shows the SEPIC model with the addition of a feedback loop implementing the digital control logic. The feedback loop measures the output voltage of the converter and uses the measurement to calculate the error from the desired reference voltage. The error is then used as input to a PID controller that controls the duty cycle of the switch in the SEPIC model.

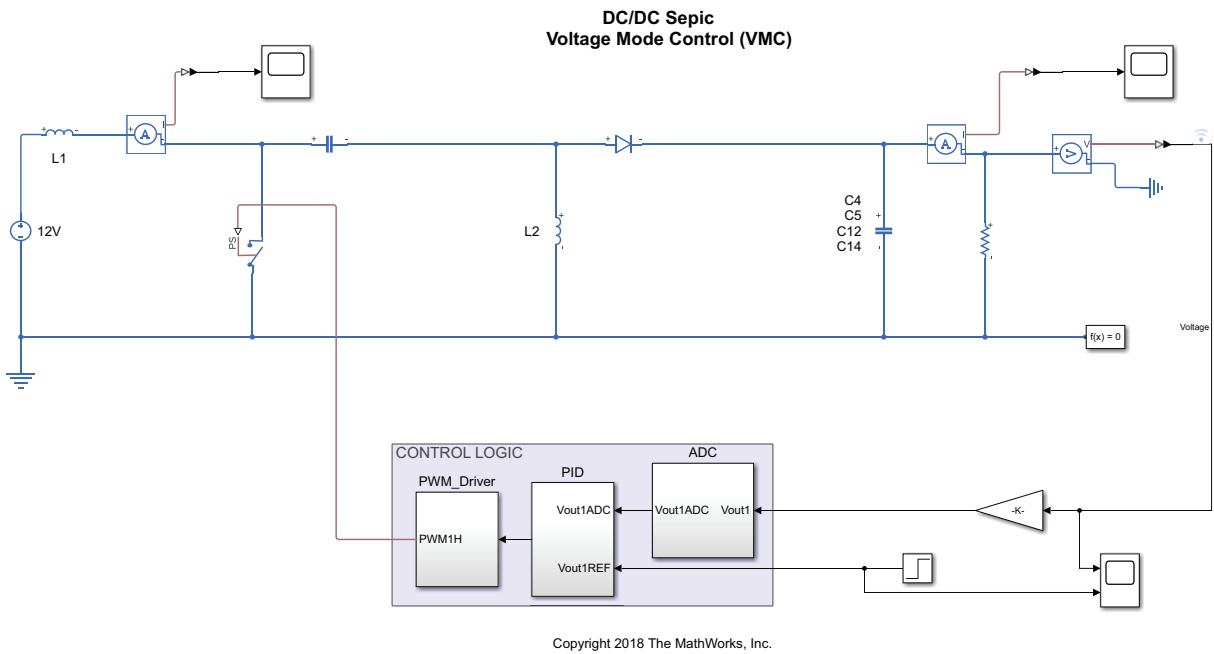


Figure 7. Closed-loop, voltage mode control.



You can simulate the model with the default PID controller gains to evaluate the behavior of the system (Figure 8). To improve performance, you can use the PID Tuner in Simulink Control Design™, which lets you automatically tune the PID controller gains. PID Tuner will attempt to automatically linearize the system to facilitate tuning of the PID gains. However, the simulation model cannot be linearized because it contains high-frequency switching. To address this challenge, you can use System Identification Toolbox™ and simulated input-output data to identify a transfer function that approximates the linearized converter dynamics.

*“Our system engineers can program, simulate, and verify the AC 800PEC controller’s regulation software very rapidly in MATLAB and Simulink. This approach markedly reduces development times and produces significantly fewer deviations from the specifications.”*

— Fritz Wittwer, ABB

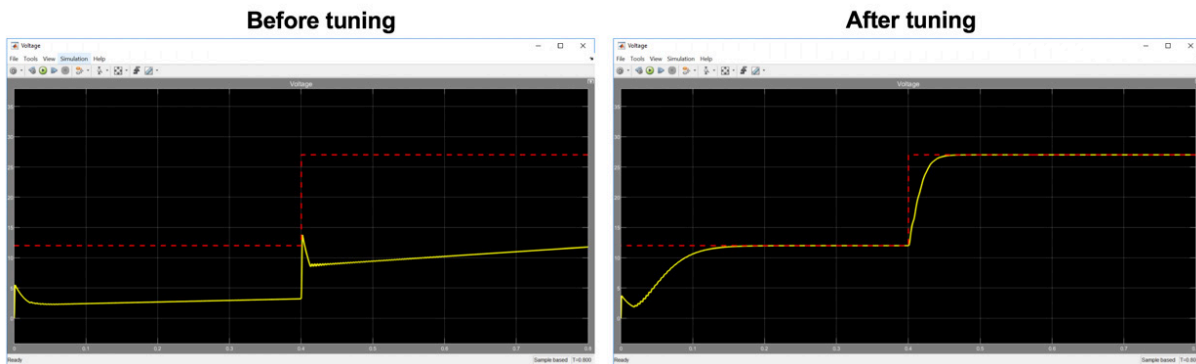


Figure 8. Performance of the PID controller before (left) and after (right) automatic tuning of the PID gains.

System Identification Toolbox lets you generate output data from PID Tuner by running the simulation for a small change in the MOSFET duty cycle. You can then identify the transfer function from simulated input-output data. You can specify the desired order for the identified model—for example, a second-order transfer function with a pair of complex poles. Once the transfer function has been obtained, PID Tuner automatically computes PID controller gains to meet user-specified bandwidth and phase margin requirements. The computed gains are used to update the gains of the PID Controller block in the Simulink model.

You can verify that the performance of the tuned controller meets the response time requirements by running the simulation with the original nonlinear system (Figure 8, right). You may also want to execute additional simulation tests to ensure that performance is acceptable in multiple scenarios. For example, you can add various load types, rerun the simulation, and examine the results.

### Learn More

[Tuning a PID Controller When a Plant Model Is Not Available](#) - Article

[ABB Accelerates Application Control Software Development for a Power Electronic Controller](#) - User Story

## Generating Code for Implementation on a Microcontroller

After verifying the controller design in simulation, you can implement the controller on hardware. Writing equivalent C/C++ code manually is error prone and could significantly delay the development time. With Simulink and Embedded Coder, you can automatically generate C/C++ code for your algorithms and export it to a production software development environment.

Using Embedded Coder®, you can generate production code from the digital controller model and deploy it to the TMS320F28035 microcontroller on the TI DC/DC LED kit. You can verify the performance of the generated code running on the microcontroller by changing the output voltage of a SEPIC on the TI kit to adjust the brightness of the LED panel.

Using Embedded Coder with a downloaded hardware support package, you can interface with the target hardware, such as TI's C2000 microcontrollers (Figure 9). The Embedded Coder Support Package for Texas Instruments C2000 Processors provides access to additional blocks, specific to the target hardware, that enable you to develop microcontroller logic and interface with your design. These blocks include drivers, PWM generation, and control of pins.

*“After riding the eBike to get a feel for how our system performed, we adjusted the control algorithm and parameters in Simulink and then re-generated code for prototype hardware. In as little as five minutes, we were ready for a test ride with the updated controller.”*

— Daniel Baumgärtner, Bosch eBike Systems

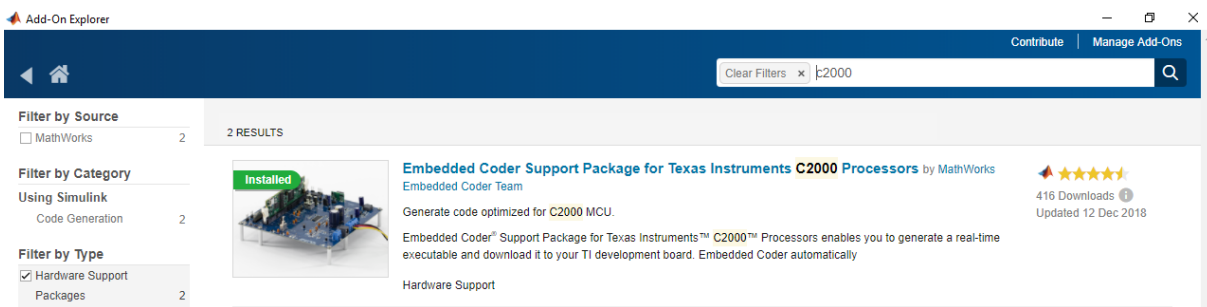


Figure 9. Interfacing with external hardware in Embedded Coder by downloading hardware support packages.

You can then update the control logic part of the model using the blocks from the support package to add analog-to-digital conversion, interrupts, and PWM generation functionalities for the TMS320F28035 microcontroller (Figure 10).

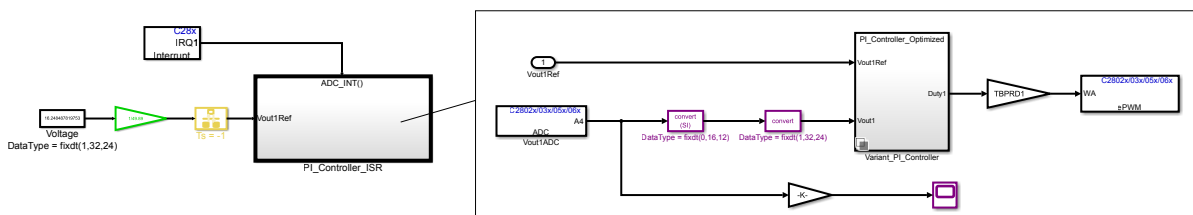


Figure 10. Adding drivers, PWM generation, and control of pins for the target hardware using the hardware support package.

With the new blocks in place, you can generate code for the microcontroller. The code generation report (Figure 11, right) provides bidirectional traceability between the Simulink model and generated code. TI's Code Composer Studio™ IDE (Figure 11, left), if downloaded, will deploy the executable to the microcontroller if the hardware is connected. Using this capability, you can generate and deploy code to the target hardware directly from Simulink.

*“The code we generated with Embedded Coder does just what it is supposed to do. Writing code with nine PI loops in it by hand and then debugging it on hardware would have added six months or more to the schedule.”*

— Brad Landseadel, Stem

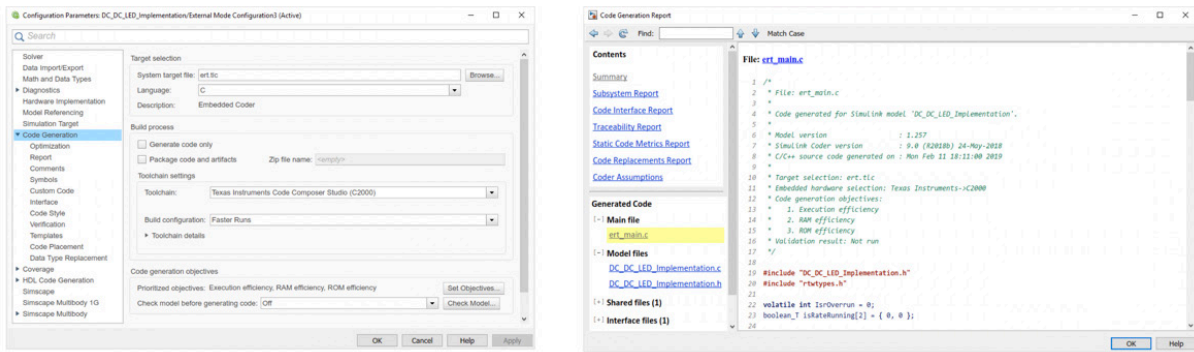


Figure 11. Generating and deploying code to the target hardware directly from Simulink. You can also access the code generation report to trace the code back to your Simulink model.

With code generated and deployed to the microcontroller, you can use the Simulink model as an interface to tune parameters in the code and to monitor signals in real time. Figure 12 shows a Simulink model in which a knob is used to change the value of reference voltage for the converter. This value is updated in the code running on the microcontroller, allowing you to change LED brightness.

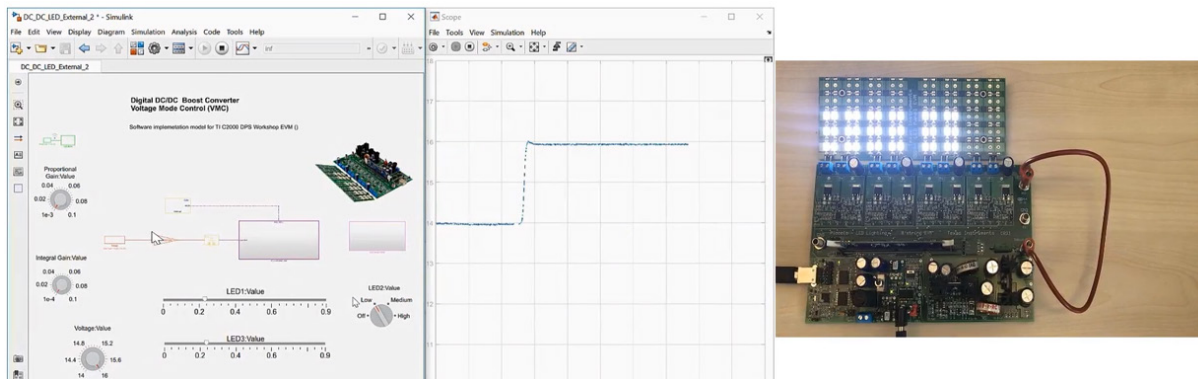


Figure 12. Using the Simulink model to tune parameters on the generated code running on the microcontroller. For example, you can use a knob to change the LED brightness command and see the LED response in real time.

## Learn More

[Tune and Monitor Models Running on Target Hardware](#) - Documentation

[Implementing a Buck Converter Controller Using Embedded Coder \(34:01\)](#) - Video

[Bosch eBike Systems Develops Electric Bike Controller with Model-Based Design](#) - User Story

[Stem Accelerates Development of Power Electronics Control Systems with Model-Based Design](#) - User Story

## Conclusion

This white paper explains how Simulink speeds up developing a digital controller for a power converter. Simulink lets you model the converter and design a PID controller using simulation to test the controller over the converter normal and abnormal operation ranges. Finally, you can generate C code from the controller model and implement it on a microcontroller, such as TI's TMS320F28035.

## Get Started

Take the next step to speed up your power converter control project:

- [Power Electronics Control Design with Simulink](#) - Overview
- [Developing DC-DC Converter Control in Simulink \(19:53\)](#) - Video
- [Trial Software for Power Electronics Control](#) - Download
- [Developing Embedded Software](#) - Consulting Services