How to build a buck converter

PN119 | Posted on August 18, 2021 | Updated on August 8, 2025

Jessy ANÇAY
Sales & Project Engineer
imperix • in

Table of Contents

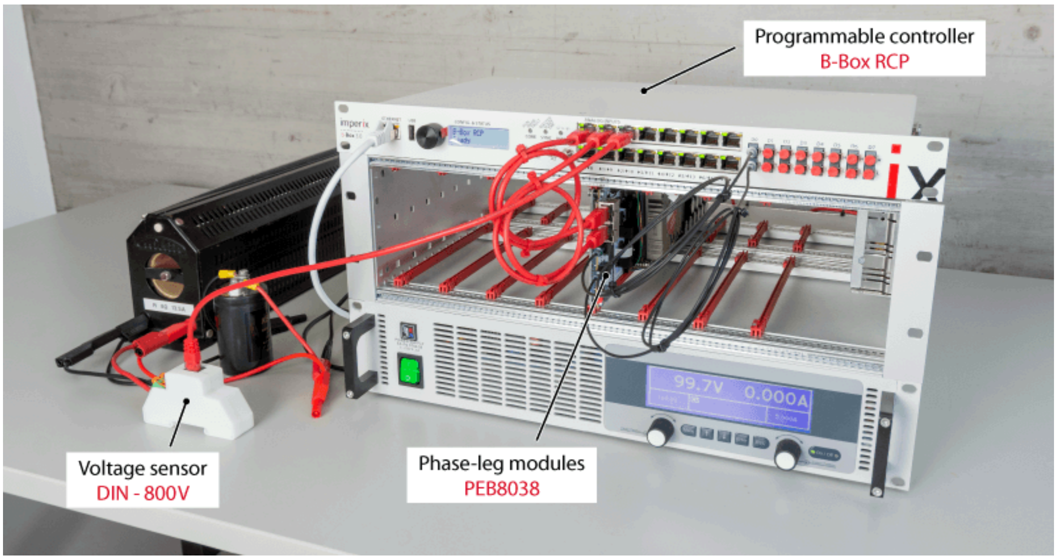
This quick-start guide describes how to build a buck converter controlled in open-loop using power modules and the B-Box RCP programmable controller using the Simulink blockset. Specifically made for users who would want to get familiar with imperix's solutions, this guide details step-by-step instructions on how to assemble and program a simple power converter. For further details on the theoretical aspects of buck converters, please refer to the page: Step-down buck converter.

Hardware requirements

The following list describes the required hardware. It comprises imperix products as well as additional components commonly available in power electronic research laboratories :

- Imperix products

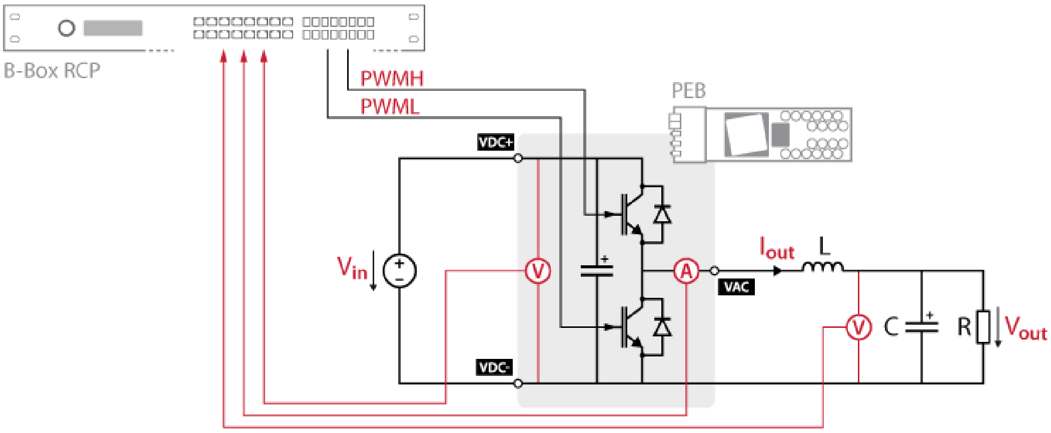
  - 1x programmable controller (B-Box RCP)
  - 1x phase-leg module (PEB8038 or PEB8024 or PEB4050)
  - 1x voltage sensor (DIN-800V)
  - Control development tools for Simulink and PLECS (ACG SDK), with a valid license
- Additional hardware
  - 1x Inductor
  - 1x Capacitor
  - 1x Resistor
  - A power supply
  - Safety laboratory cables (banana)



Complete testbench
(Power supply and passive components not sold by imperix)

Description of the implemented system

The schematic below serves as a reference for the required electrical connections. It is a buck converter made from a semiconductor switching cell, an inductor, a capacitor, and a load resistor. The converter is controlled in open-loop by a programmable controller (B-Box RCP).



Buck converter schematic

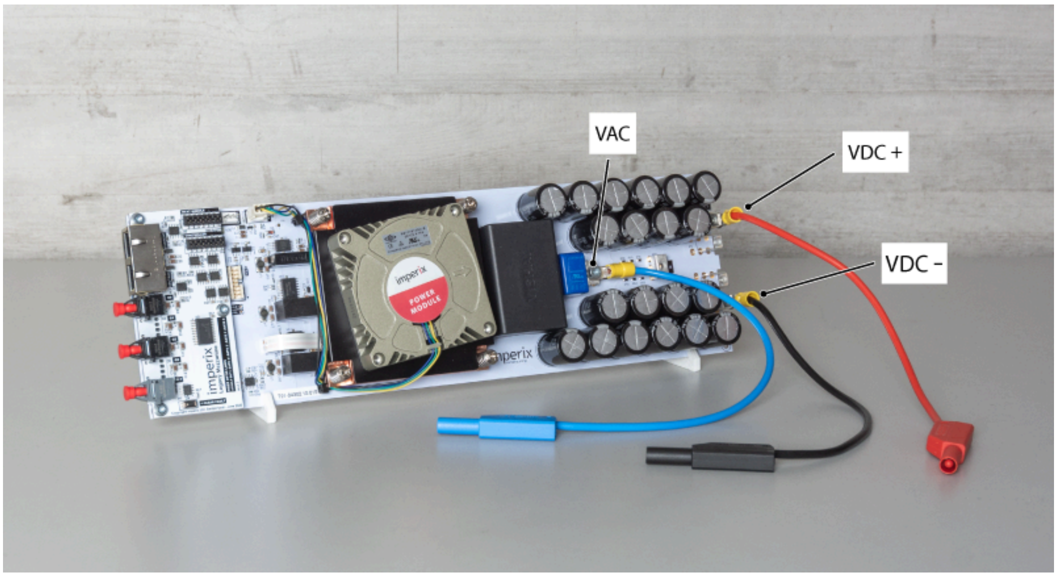
As opposed to a buck made with a diode and a transistor, this one is made with a commutation cell (two transistors). In other words, the bottom transistor's body diode acts as a flyback diode, to reduce the voltage spike across the inductor.

Note that, in the case of MOSFETs, their body diodes are known to perform poorly and will generate significant losses when the upper switch is open. To avoid this issue, a PWM signal also drives the low-side transistor so that the current goes through the channel and not the body diode.

The operating conditions for the converter were, in this specific case, defined by the load resistor current rating of 13.5 [A]. Having a resistance of 8.5 [Ω], a supply voltage of 100 [V], and therefore a maximum load current of 11.75 [A] seemed appropriate. An off-the-shelf inductor and capacitor of 2.36 [mH] and 2 [mF] respectively, guaranteed Continuous Conduction Mode (CMM) operation and acceptable output voltage ripples.

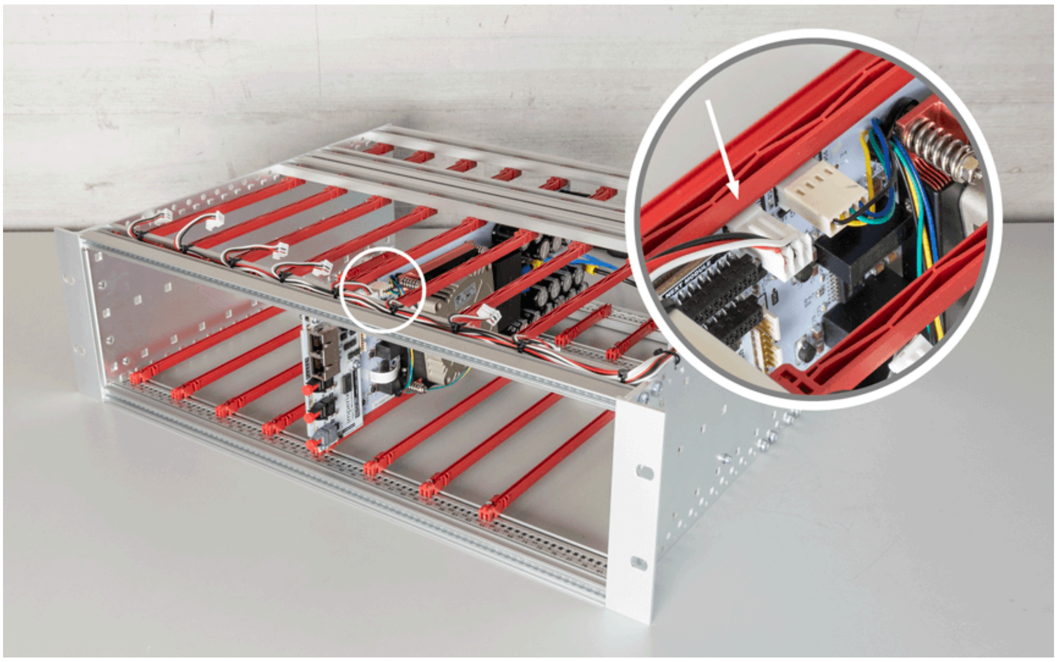
Mechanical assembly of the power equipment

The first step to build the converter is to connect a cable to the **VAC** terminal (switching midpoint) of the PEB module. It is best to do it before inserting the module into the mounting rack (the switching midpoint will be hard to reach then). As a suggestion, one could use a safety banana cable (for easier connection with other equipment) terminated by a ring terminal and a small M4 screw to secure the wire. Two similar cables could connect, in the same way, the **VDC+** and **VDC-** power terminals and the power supply.



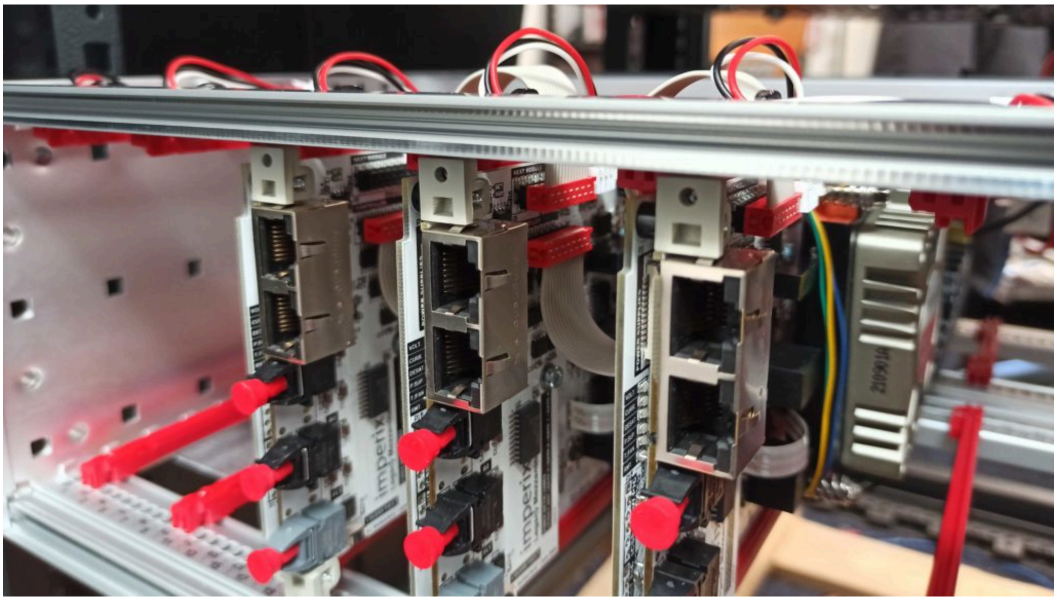
Suggested power module connection

Next, the user can insert the power module into the mounting rack as shown below. He/she also has to connect the auxiliary power supply cable from the rack to the module.



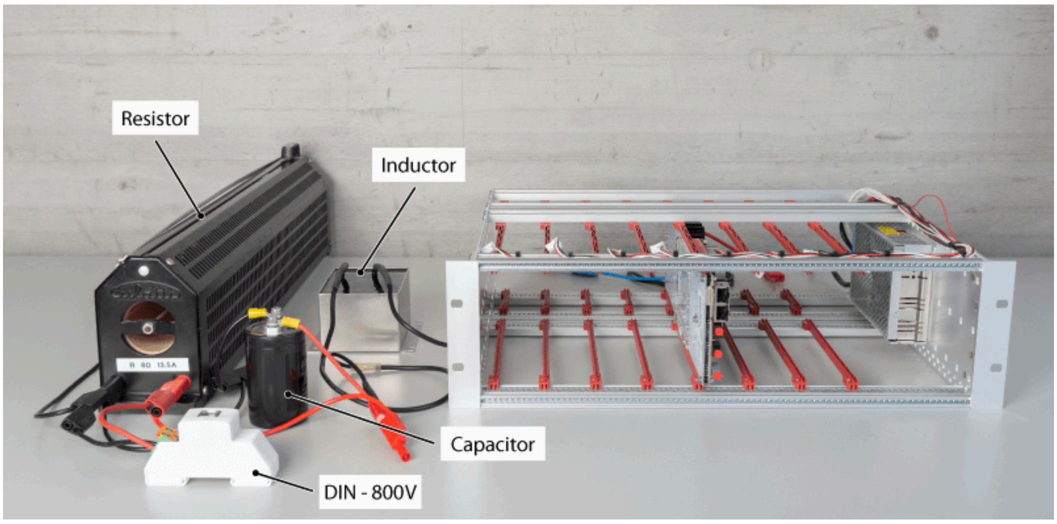
Mounting rack and power module

Note that, in this specific case, only one module is required. But for applications with several modules, a grey flat wire connects the modules for faults propagation, as shown in the picture below.



Flat cable connection for fault propagation between modules

Then, the next step is to wire the components according to the converter's schematic. The following picture shows a connection example.



Passive components connection

Connection to the controller

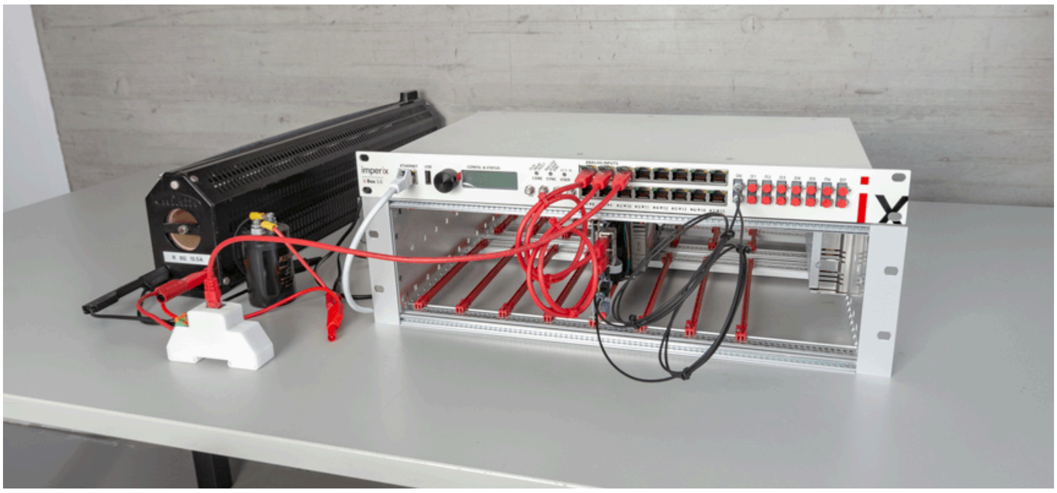
The rapid prototyping controller (B-Box RCP) connects to the power module as described in the table below.

Signal	Type	Controller side	Converter side
$PWMH$	PWM	Optical output D0H	Gate H optical receiver
$PWML$	PWM	Optical output D0L	Gate L optical receiver

Signal	Type	Controller side	Converter side
I_{out}	Measurement	Analog input 0	RJ45 port (top)
V_{in}	Measurement	Analog input 1	RJ45 port (bottom)
V_{out}	Measurement	Analog input 2	DIN-800V RJ45 port

Connection table

The following picture shows the connections once completed.



Controller connections

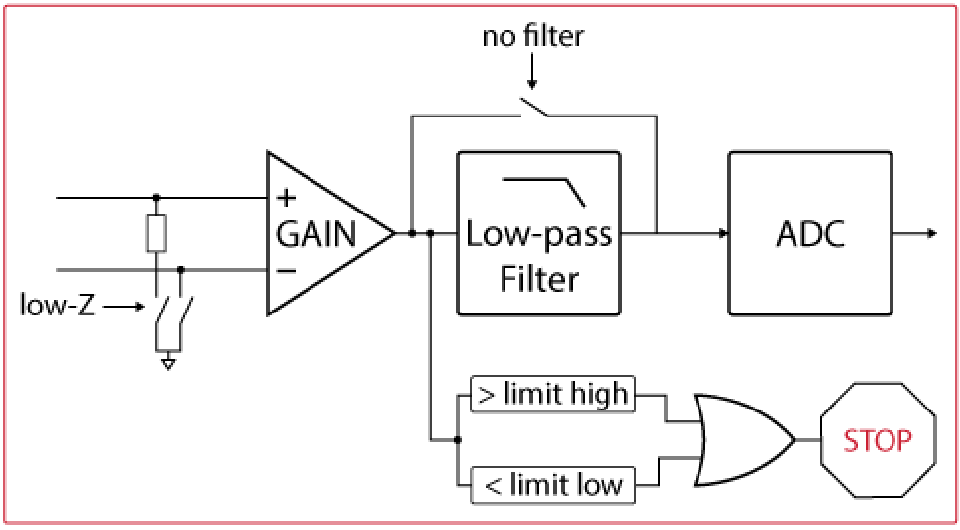
Finally, to later be able to upload the control algorithm to it, the B-Box RCP's ethernet port is connected to the computer's local network. It is also possible to have a direct connection to a computer. For more details on the procedure, please refer to the Getting started with imperix controllers (PN138).

Configuration of the analog inputs

To be able to properly retrieve the measurements, the analog input channels of the B-Box RCP need to be configured properly (more information on the analog front-end configuration of the B-Box RCP can be found in Front-end configuration on B-Box RCP (PN105). The following parameters must be configured:

- The impedance type (low or high)
- The programmable gain
- The desired filter
- The protection thresholds (limit high and low)

The following schematic illustrates the fully configurable front-end of the B-Box RCP.



Analog front-end schematic

Note that the impedance type depends on the type of sensor used and the programmable gain needs to match the one that will be later configured in the ADC block in software.

Before doing any experiments, it is essential to always properly configure the protection thresholds of the analog input for safety reasons. Their purpose is to block the PWM outputs in case of unexpected high voltages or currents.

The programmable controller's LCD screen and rotary-push button allow to read and write all configuration parameters of the analog front-end. To access the related menu:

1. Push once, select `ANALOG INPUTS` and confirm (push again).
2. Select the desired input channel and confirm.

The equation: $L = s * G * m$ is used to compute the analog input threshold, with L being the input's limit high/low, s the sensor's sensitivity, G the analog gain, and m the maximum/minimum real current or voltage value. The following table summarizes the measurement ranges (according to the previously defined operating point) and the sensor's sensitivity.

Signal	Range	Sensor	Sensitivity
I_{out}	0 – 11.75 [A]	On-board current sensor	50 [mV/A]
V_{in}	0 – 100 [V]	On-board voltage sensor	4.99 [mV/V]
V_{out}	0 – 100 [V]	DIN-800V	2.46 [mV/V]

Measurement parameters

The measuring ranges involved allow for an analog input gain of x4 on all of the controller's channels. Using the aforementioned formula the analog input limits are computed with some margin to avoid unwanted trippings. The chosen maximum output current is then set for 15 [A] meaning a front panel limit of 3 [V]. The input and output voltages are chosen to be both limited to 200 [V] which gives 4 [V] and 2 [V] respectively for the front panel limit. The configuration of the three analog input channels is summed up in the table below.

Measured signal	Input channel number	Low impedance	Gain	Filter	Limit high [V]	Limit low [V]	Disable safety	Save
I_{out}	0	no	x4	no	3	-1	no	yes
V_{in}	1	no	x4	no	4	-1	no	yes
V_{out}	2	no	x4	no	2	-1	no	yes

Analog inputs configuration

Development of the control software

Installation and setup of the computer software

Two pieces of software are required. The imperix Automated Code Generation Software Development Kit (ACG SDK) can be downloaded here. Besides, a compatible version of Matlab (2016 and newer) is required as well as the following toolboxes:

- Matlab Simulink
- Embedded coder
- Matlab coder
- Simulink coder
- The Simscape Power Systems blockset

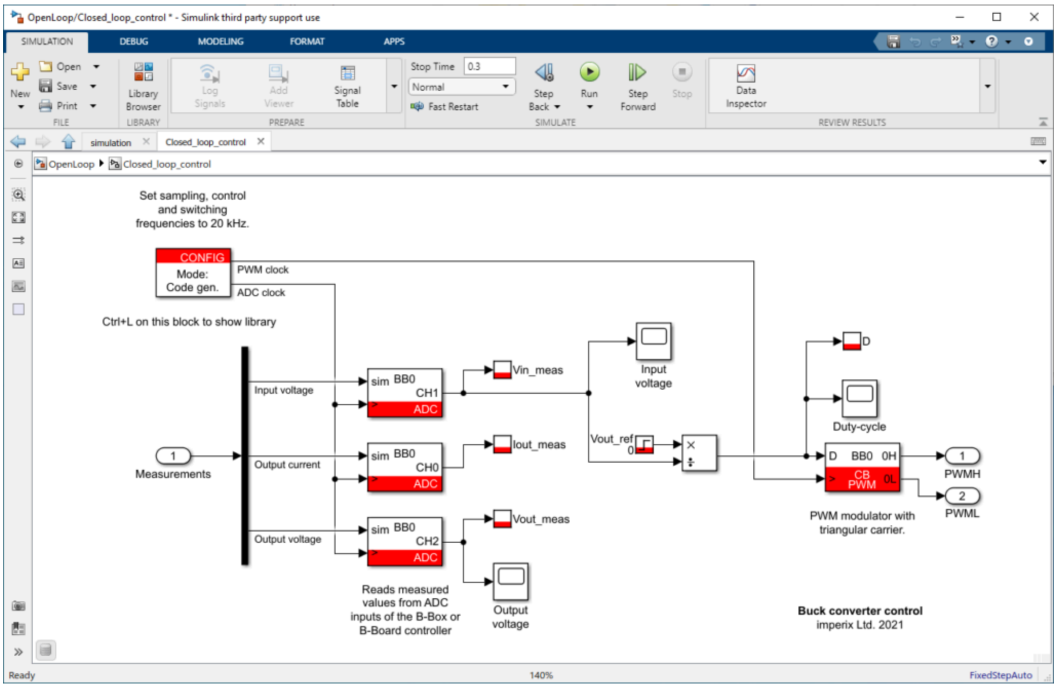
A compatible version of PLECS (4.4.2 and newer) in 64 bits can also be used instead of Matlab.

For a detailed guide on how to set up the software solution, please refer to the Installation guide for imperix ACG SDK (PN133).

Creation of the Simulink model

The control model to implement a simple feedforward control of the output voltage is given below as well as step-by-step instructions on how to create it. For details on closed-loop control please refer to TN105.

PN119_Openloop_Buck_ConverterDownload



Set sampling, control and switching frequencies to 20 kHz.

Ctrl+L on this block to show library

Reads measured values from ADC inputs of the B-Box or B-Board controller

PWM modulator with triangular carrier.

Buck converter control
imperix Ltd. 2021

Step-by-step instructions

User template

The following steps show how to build a buck converter control algorithm. The user can code, as aforementioned, using Simulink or PLECS. This guide will focus on Simulink, however, both programming options are introduced on the following two pages:

- Getting started with ACG SDK on Simulink (PN134)
- Getting started with ACG SDK on PLECS (PN136)

To start working with Simulink, one can find a project template in:

C:\imperix\BB3_ACG_SDK\simulink\user_template\imperix_template.slx

This template is preconfigured for code generation for imperix controllers. It should then be copied into the desired working directory and renamed. When opening this file, a controller and a plant block are displayed as shown below.

The plant block contains the simulation model of the circuit and will not be detailed here. Further information can be found in Simulation essentials with Simulink (PN135). The controller block is the one of interest since it is the model of the actual real-time code.

The CONFIG block

The CONFIG block, already present inside of the controller block, performs the basic configurations of the model. It is by default set to simulation. This parameter needs to be changed to code generation to program the B-Box RCP.

The ADC block

Then, the buck converter requires several blocks to work properly. First, three ADC blocks configure the ADC peripheral. The measurements are retrieved in software from raw 16 bits ADC values. The ADC block, therefore, gives the sensitivity and analog gain information required to properly compute the measured value. They need to be configured as follows:

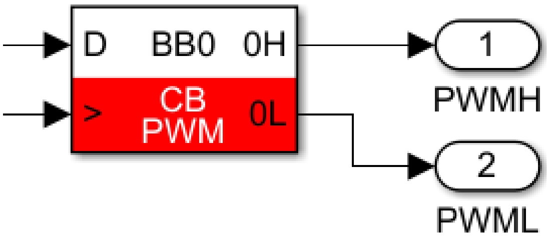
ADC 0	
Device number	0
Input channel	0
Sensor	Current PEB 8038
Gain	x4

ADC 2	
Device number	0
Input channel	2
Sensor	Voltage DIN-800V
Gain	x4

ADC 1	
Device number	0
Input channel	1
Sensor	Voltage PEB 8038
Gain	x4

The PWM block

The PWM signals are generated by the carrier-based PWM block (CB PWM) and drive the power transistors. The selected configuration for this block is shown below.



PWM block

Since both of the module's transistors are switched, the `output mode` is set to Dual. The `carrier type` is set to triangle and the `update rate` is set to single (one update instant happening at the zero of the carrier).

Duty cycle

Since the converter is designed in continuous conduction mode, the step-down ratio corresponds to the duty cycle:

$$d = \frac{V_{out}}{V_{in}}$$

The duty cycle for the PWM can therefore be computed by dividing the reference output voltage by the measured input voltage in a feedforward fashion. The user will then be able to set a voltage reference in BB Control and the algorithm automatically computes the duty cycle.

The tunable parameter and probe blocks

The last two blocks used are the tunable parameter and the probe block. The tunable parameter is used to let the user set the voltage reference while the probe is used to watch the desired signal in real-time in the BB Control utility software.

Generation and real-time execution of the controller code

At last, the user can upload the code by pressing Ctrl + B on Simulink (Ctrl + Alt + B on PLECS). This will automatically generate the C code that is then compiled, loaded, and launched on the selected target. It also launches the Cockpit software. More information on the latter can be found in the corresponding user guide (PN300).

Then, when connected, Cockpit automatically starts the code. At this point, one can add the control variables, as well as the probes to the watch list in the debugging tab. In this case, the user can add the three measured values: I_{out_meas}, V_{in_meas}, V_{out_meas} and the control variable V_{out_ref}.

A first test could be performed to check that the hardware protection limits are correctly configured on the analog frontend of the B-Box RCP. This test consists of lower the limits values and verifying that the PWM signals are indeed disabled when the limits are exceeded.

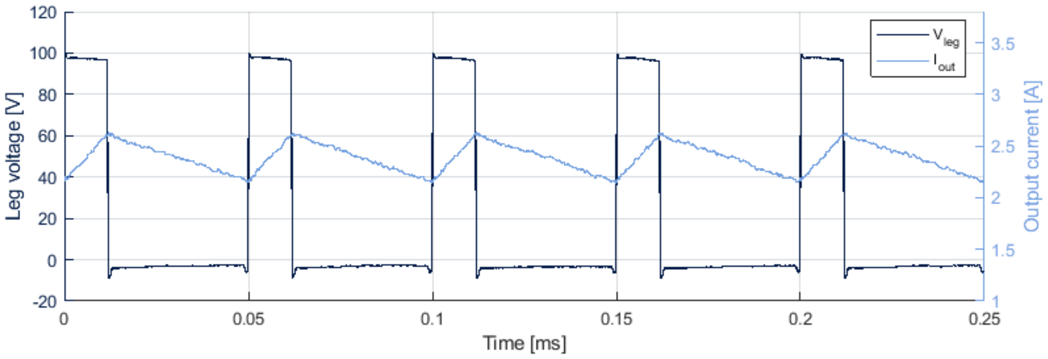
Note that the measured values might not be extremely precise. Indeed, the sensors are designed for higher current and voltage measurements (50 [A] and 800 [V]). They might therefore show a little offset when measuring smaller values.

Subsequently, one can turn on the power supply and slowly increase the voltage to 100 [V]. To make sure that the input voltage measurement is correct, the user could verify that V_{in} is indeed close to 100 [V] on BB control. The last step is then to set a value for the reference voltage, 25 [V] for instance, and press the enable output button. One should then see a measured output voltage of around 25 [V].

For step-by-step instructions on how to use the B-Box RCP or the B-Box Micro, please consult PN138.

Experimental results

The plot below shows the leg voltage as well as the output current of the buck converter built above. With a duty cycle value of 0.25, it is clear on this graph that the input voltage of 100[V] is applied to the output only during a fourth of the control period. This results then in an output voltage of 25[V].



Buck converter leg voltage and output current measurements

To go further…

The next possible step could be to add two power modules to build a three-phase voltage source inverter (VSI) like the one in TN152. A guide, similar to this one, on the assembly of a three-phase inverter is available here: How to build a 3 phase inverter. One could then combine the VSI with a boost converter to create a Three-phase PV inverter for grid-tied applications (AN006), which showcases the great potential of imperix's solution for modular power converters. Finally, the interleaving of several buck converters and the implementation of proper current control techniques is discussed in Interleaved buck converter current control.