Build a custom user interface to operate Imperix power converters

PN175 | Posted on June 3, 2022 | Updated on May 7, 2025

Jessy ANÇAY
Sales & Project Engineer
imperix

Table of Contents

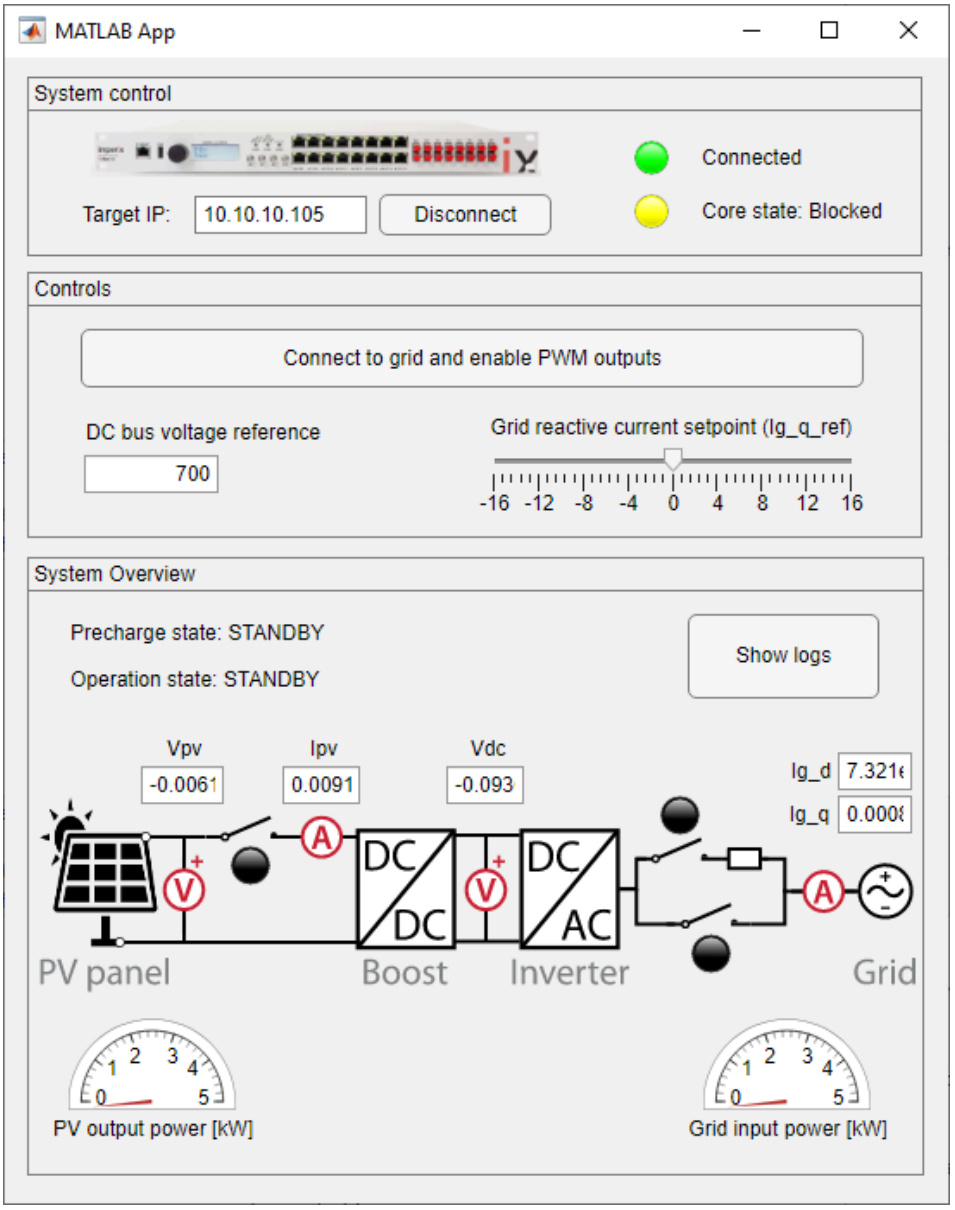
This note will address specific considerations to build a custom user interface for the operation of Imperix power converters. It will focus on the main aspects to consider when operating a grid-tied converter using the OPC UA protocol. The B-Box RCP which embeds an OPC UA server can indeed be controlled with an OPC UA GUI client. Several possibilities to implement custom interfaces are then available to the users such as for instance:

- Designing a GUI with MATLAB App Designer with the Industrial Communication Toolbox
- Using the open62541 C library and a C++ framework such as Qt or wxWidgets
- Working with Python and Freeopcua with the tkinter package

Making a GUI can be seen as a complementary solution to the imperix remote control software, which is often preferred for debug purposes. Also, the purpose of the GUI is limited to real-time interactions with the power converter controller(s) and is not related to how to program the controller(s), which can be done using dedicated SDKs.

A GUI (see below) tailored for a grid-tied PV inverter, similar to the one treated on the page: Three-phase PV inverter for grid-tied applications, made in Matlab App Designer is provided as an example. The GUI and the corresponding Simulink model can be downloaded using the following link. Please note that it requires Matlab version r2022a or newer.

This model is designed to automatically output PWM signals. It must therefore be used with the utmost precaution to avoid irreversible damage to the power converter.
Download GUI_Imperix_Converter.zip



Custom user interface example for a grid-tied PV inverter

The implementation of such an interface is typically useful in order to manage the run-time execution of a power converter, including for instance:

- Status information (running, stopped, discharge, fault, etc.)
- Real-time measurements (power flows, voltage, current, etc.)
- Fault diagnosis information (log messages, probable causes, etc.)

- Operation set points (current, power, etc.)
- Commands (start, stop, control mode change, etc.)

Accessing system and user variables from a custom user interface

Starting with a small reminder on the B-Box RCP's OPC UA variable might be useful. There are two types of variables in the B-Box RCP: the Imperix system variables and the user-created variables. The system variables provide configuration and status information about the target. For the user-created variables, when a user code is started, an OPC UA variable is created for each probe variable and tunable parameter present in the user model (Simulink, PLECS or C++). More information on the topic can be found on the page: OPC UA: the communication protocol for industrial automation applications.

Converter automation

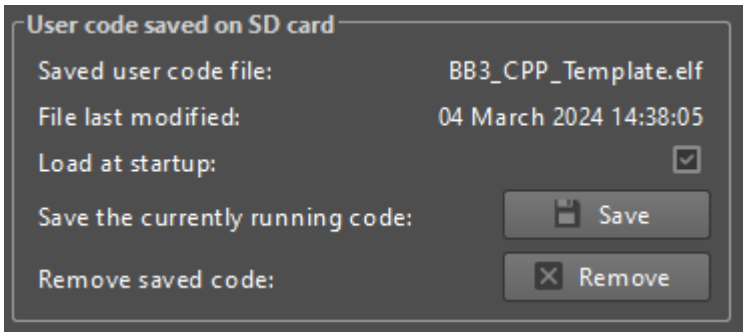
Designing a custom user interface usually translates the desire to automate and simplify the operation of a given system. The two following sections will present how to automatically launch a user code upon powering the B-Box RCP and how to implement a state machine to automate the starting procedure of the power converter.

Launch a user code on start-up

It must be noted that actions such as loading the user code file or a custom FPGA bitstream file into the target or updating the target firmware cannot be performed using OPC UA. Furthermore, a code must be running on the device for a client to be able to access user variables and operate the converter. The user code can then either be manually launched using the usual procedure or launched automatically when powering the target (see procedure below). To check that the code is indeed running on the target, one can access the `User CPU state` system variable. This `User CPU state` variable returns the CPU state which can either be *Running* or *Offline*.

To avoid starting Cockpit every time one wants to operate the converter with a custom user interface, it can be interesting to automatically start a user code when turning on the B-Box RCP. To do so, follow the procedure below:

- Launch the user code as usual by pressing Ctrl+B on Simulink or Ctrl+alt+B on PLECS. The Cockpit monitoring software should launch automatically.
- Upload the code on your target by entering its IP address.
- Once the code is running on the target, open the *Target configuration* window
- Click on *Save current code* to save the current code on the SD card inside of the B-Box RCP.
- Click on the checkbox *Load at startup* to automatically launch the code when the B-Box is turned-on.



Load user code at startup

This will save the control algorithm on the SD card inside the B-Box RCP. Then, when the B-Box is turned on, it will automatically load and start the code from the SD card.

Operational state machine implementation

The idea here is to be able to start a power converter by pressing a single button. To do so, a state machine can be implemented to automate the start-up and shut-down procedure of a power converter. This can be especially interesting in the case of grid-tied converters that require switching relays in a precise order to properly connect to the grid.

Once the converter is properly started, the PWM signals can be automatically activated using the Enable PWM outputs block. By doing so, Imperix power converter can be up and running with the press of a single button.

Enabling the PWM outputs must be done with the utmost precaution to avoid causing serious damage to the converter. Imperix recommends using the BB Control button when possible.

Safety considerations when using user interfaces

By nature, custom user interfaces are not proper monitoring tools. As a consequence, the user is less aware of what is happening inside the converter.

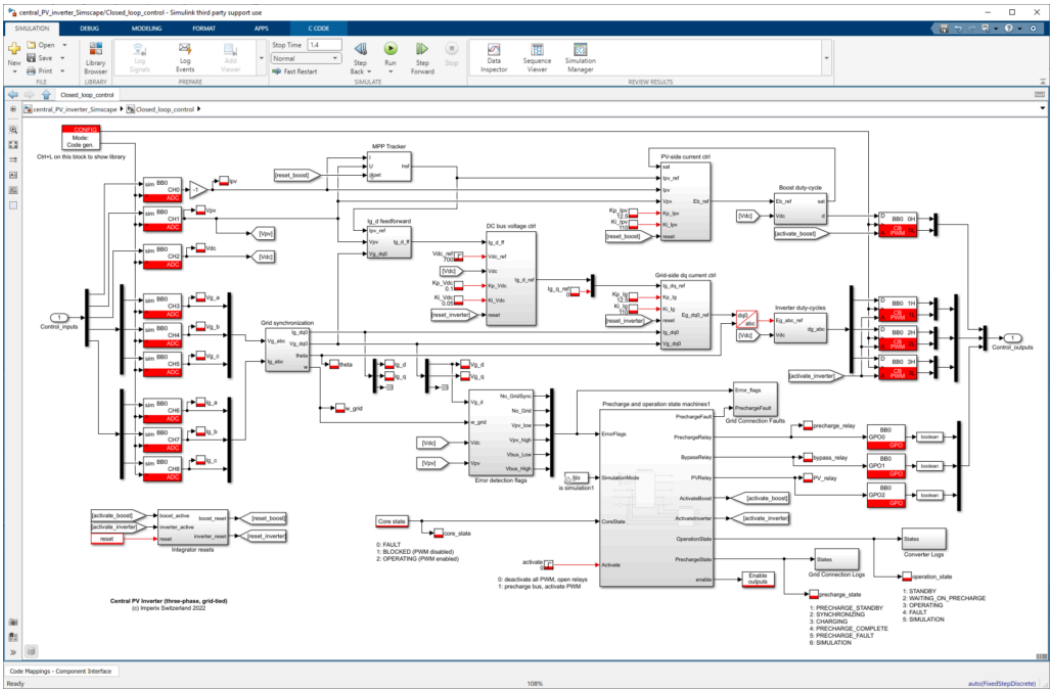
Imperix strongly recommends using the Cockpit monitoring software during the development stage of power converters or for use cases where performance is important (e.g. data logging).

Safety considerations then need to be taken even more seriously when operating Imperix converts with a custom GUI. Here is a non-exhaustive list of safety measures that can be implemented:

- Proper configuration of the B-Box analog front-end protection limits (cf. Analog front-end configuration on B-Box RCP).
- Wiring of an emergency push button to the interlock connector of the B-Box.
- Implementation of software error detection mechanism with a dedicated fault state inside the converter state machine. Logging errors with the LOG block could also be useful.

Custom user interface example

In this section, a custom GUI designed for the operation of a grid-tied three-phase PV inverter is given as an example. This GUI was developed using Matlab App Designer. The Simulink model for this example implements an operation state machine, software error detection and logging, and automatic PWM activation. Both the application and the model can be downloaded using the link at the top of the page.



Grid-tied PV inverter Simulink control model

The list of OPC UA user variables is given in the table below:

Name	Data type	Description
Vpv	Single	PV voltage [V]
Ipv	Single	PV current [A]
Vdc	Single	DC bus voltage [V]
Vdc_ref	Single	DC bus voltage reference [V]
Vg_a / Vg_b / Vg_c	Single	Grid voltage (phase A B and C) [V]
Ig_a / Ig_b / Ig_c	Single	Grid current (phase A B and C) [A]
theta	Single	Grid angle [rad]
w	Single	Grid angular frequency [rad/s]
Vg_d / Vg_q	Single	Grid voltage in dq [V]
Ig_d / Ig_q	Single	Grid current in dq [V]
Kp_Vdc / Ki_Vdc	Single	PI gains for DC bus controller
Kp_Ipv / Ki_Ipv	Single	PI gains for PV current controller
Kp_Ig / Ki_Ig	Single	PI gains for grid current controller
precharge_relay	Single	The relay used to precharge the DC bus
bypass_relay	Single	The relay used to connect to the grid
PV_relay	Single	The relay used to connect the PV panel
precharge_state	Single	State of the precharge state machine
operation_state	Single	State of the operation state machine

Table of the OPC UA user variables