

# Professional Python for Embedded

Online Live

Power Workshop

Applicable Technologies	Requirements	Contact
Python, PYNQ	Basic knowledge on OOP	Michael Schwarz P. +49 7664 91313-15 E. info@plc2.de
Fee (net per person)	Inclusive	Duration
<b>OL</b> € 2,800	Training material	5 days
<b>PW</b> € 3,200	Plus beverages during breaks Lunch	5 days

## Workshop

Within the embedded computing sphere, the programming language C has long been considered the standard. However, more complex applications and faster time-to-market requirements call for alternatives.

Traditionally used for web and desktop applications, Python offers such an alternative, thanks to its support for C/C++ libraries and the provision of frameworks for processing complex algorithms. Those associated frameworks enable the development of complex algorithms used across data analytics, Machine Learning (ML), and Artificial Intelligence (AI) applications. Of course, these applications are hot topics within embedded computing and are driving the adoption of Python, especially within the Industrial Internet of Things (IIoT) at the edge.

Xilinx® offers an open source framework (PYNQ) that allows Python to interact with the Zynq® portfolio.

This is realized by hybrid libraries, which are a new form of libraries. Such a hybrid library includes both a bit-stream and the associated hardware-related C code, but also an Application Programming Interface (API) for Python. In this course, you will learn the basics of programming with Python, which you will use in a workshop. You will then be able to develop prototypes based on Python on embedded systems in a short time. As an attendee of the extended workshop, you will also receive a profound training in the use of libraries to process large data, which will form the basis for application development in the field of AI, ML, and Big Data.

Due to accompanying exercises, the course offers in-depth and practice-oriented training. Attendees of the online live course will do the practical exercises in the afternoon on their own.

## Agenda

- |   |   |
|---|---|
| <p><b>01. Introduction</b><br/>Environment setup<br/>Reasons to use Python<br/>The use of Jupyter</p> <p><b>02. Development with Python</b><br/>Variables<br/>Types, I/O, and import<br/>Operators and namespace<br/>Datatypes</p> <p><b>03. Flow control</b><br/>Statements<br/>Loops</p> <p><b>04. Functions</b><br/>Function and argument<br/>Recursion<br/>Modules and packages</p> <p><b>05. Objects and classes</b><br/>OOP<br/>Inheritance</p> | <p><b>06. Advanced</b><br/>Iterators and generators<br/>Decorators and closures</p> <p><b>07. Data analysis with Python</b><br/>Structures and sorting<br/>Operators and algorithms<br/>Statistics</p> <p><b>08. Embedded hardware</b><br/>PYNQ architecture<br/>Board setup</p> <p><b>09. Libraries</b><br/>CTypes</p> <p><b>10. Workshop - applications</b><br/>GPIO app<br/>Face and eye detection<br/>HDMI streaming and processing</p> <p><b>11. FPGA</b><br/>Creation of a custom overlay (Vivado®)</p> |
|---|---|