

## Driving Virtual Prototypes through Python Asynchronously

### Description:

The increasing complexity of embedded devices, such as modern MEMS sensor systems, makes an early availability of a model-based executable specification nearly indispensable. Those high level models can support the design space exploration and uncover negative design decisions or specification inaccuracies at an early stage of system development.

However, in order to the execution of the virtual prototype and its inputs and outputs, a compiled test bench is too inflexible. On the other hand, scripting languages provide a comfortable and flexible programming environment. In particular, Python is popular among researchers and developers with many implementations and extensions (SciPy, Numpy, Matplotlib, IPython, Anaconda, etc.). Furthermore, bindings to MatLab, GNU Octave, Qt, C, C++, etc. exist for Python.

In this project, a Python interface to interact with SystemC-based Virtual System Prototypes (VSPs) should be created. Thereby, an execution in separate thread is mandatory in order to avoid blocking the Python interpreter. The interface should support, data input & output to & from the VSP as well as accessing & modifying its internal states. Furthermore, the execution of the VSP should be controlled from within the Python interface (e.g., stepping).

### The following tasks have to be conducted:

- Literature research about SystemC / Python C/C++
- Explore & present possible approaches
- Python interface specification and elaboration
- Prototype implementation for a given VSP
- Discussion and documentation of the results

### Supervisor:

- [M.Sc. Jens Rudolf](#)
- [Prof. Dr.-Ing. Christian Haubelt](#)

