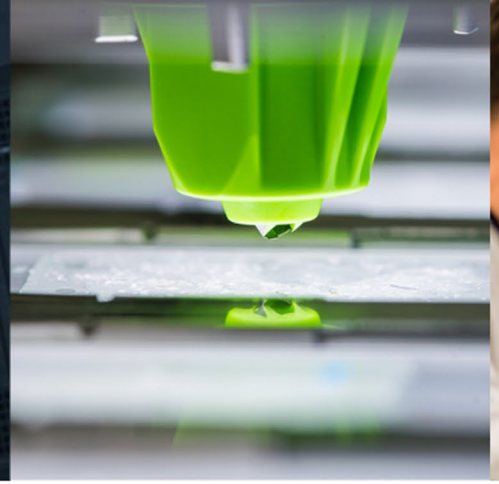
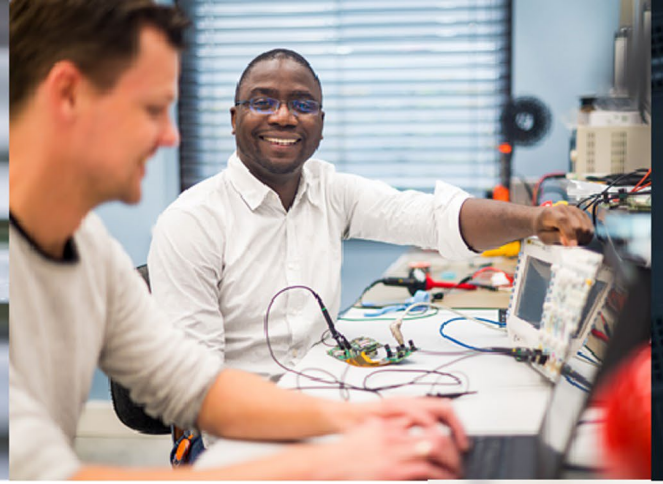
SOFTWARE



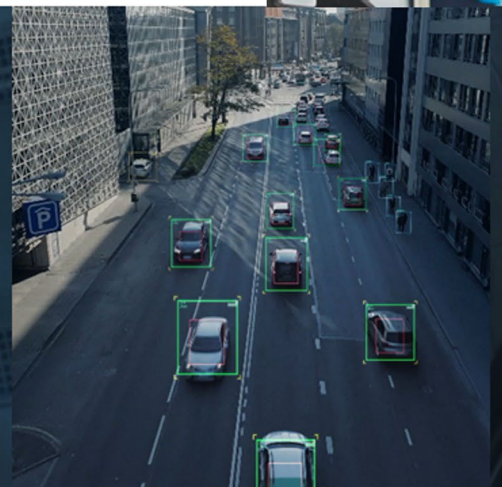


MECHATRONICS

SIOUX TECHNOLOGIES

ASSEMBLY

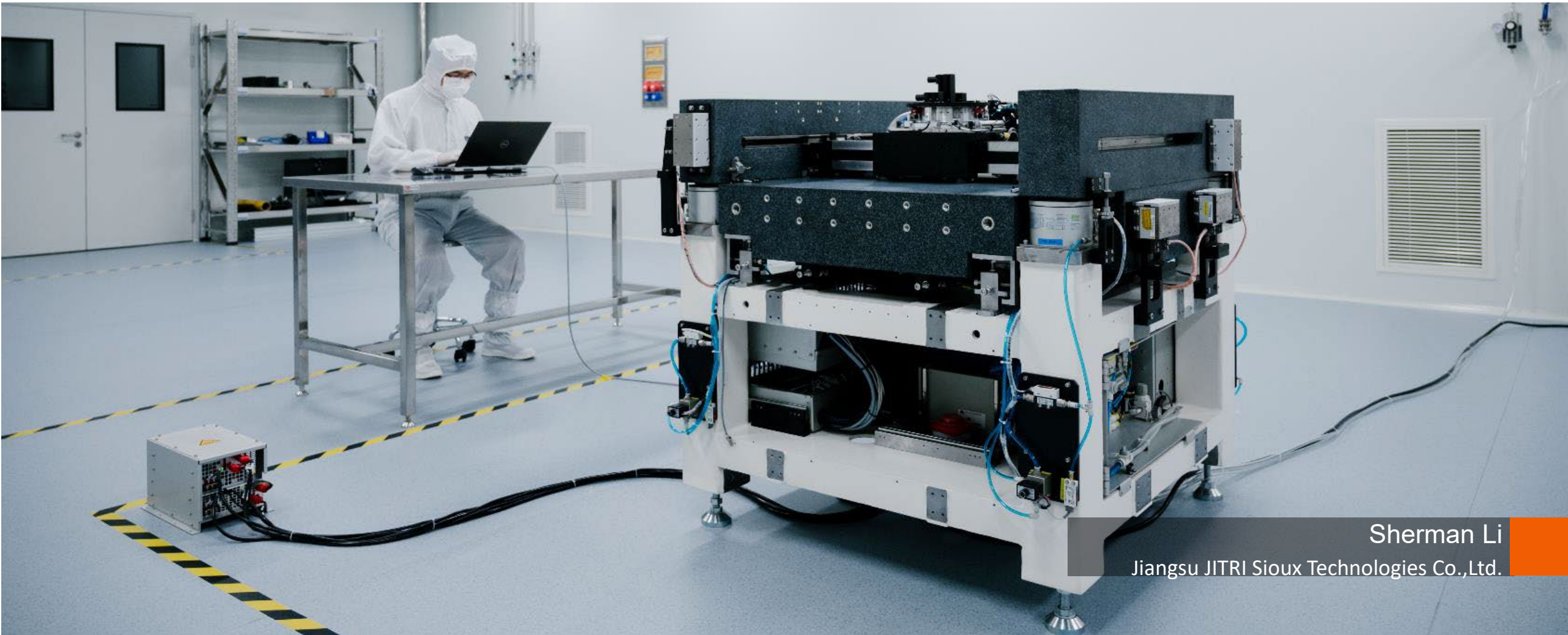
MATHWARE



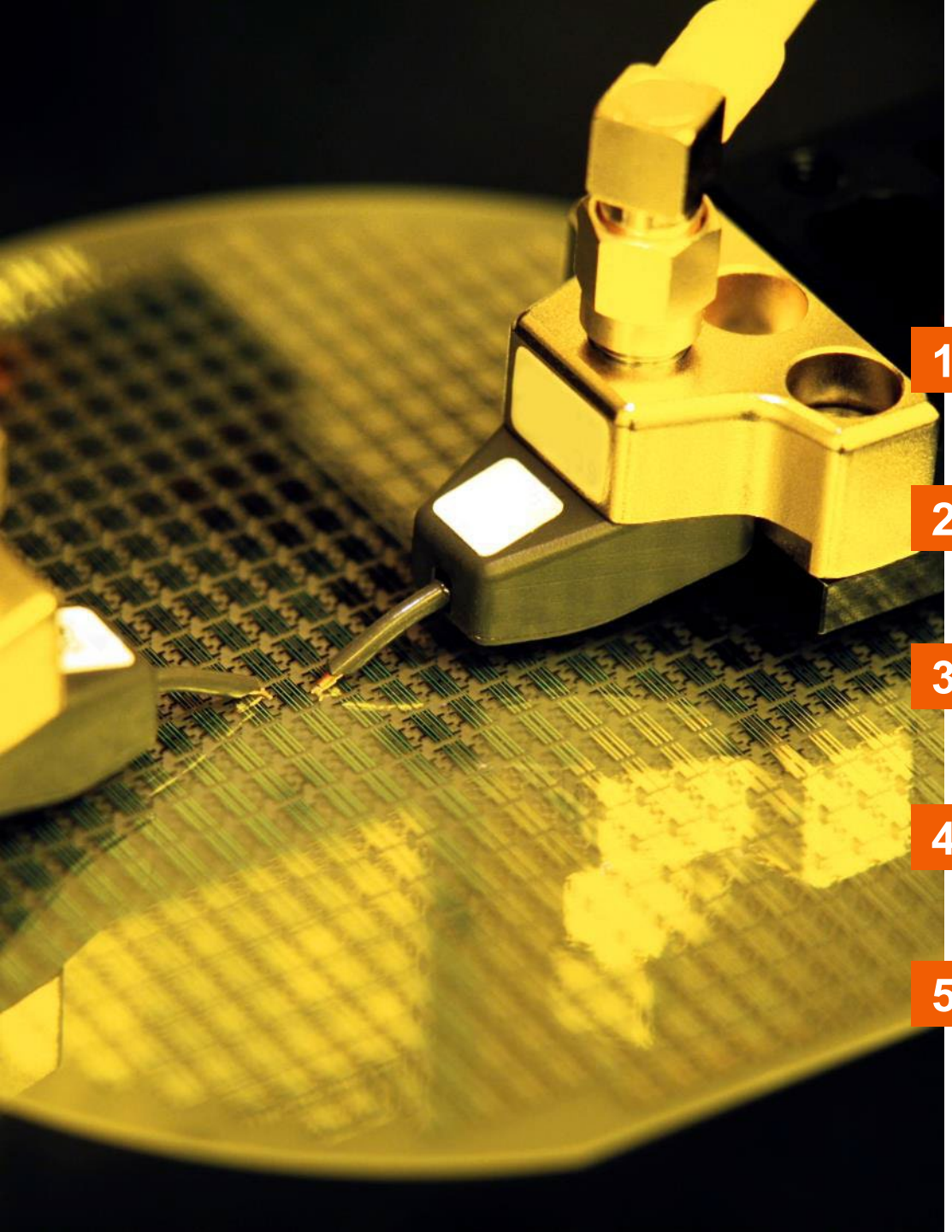
ELECTRONICS

Intelligent Motion Control Platform to Accelerate Mechatronic System R&D

SIOUX TECHNOLOGIES



Sherman Li
Jiangsu JITRI Sioux Technologies Co.,Ltd.



Agenda

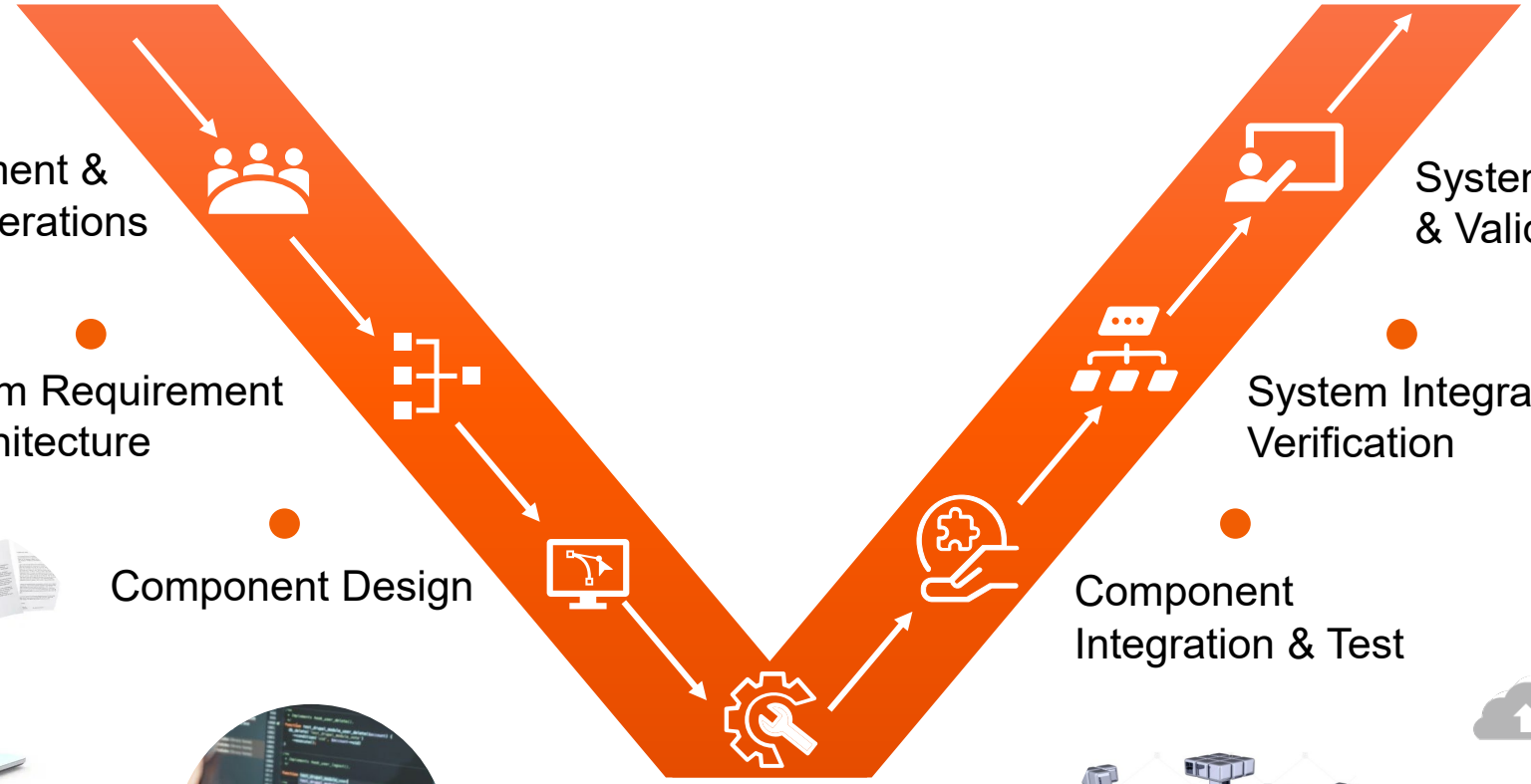
1. Background
2. Concept for Intelligent Motion Control Platform
3. Intelligent Motion Control Platform Architecture and Features
4. Application Example: UPSS Product
5. Introduce Sioux Technologies

Intelligent Motion Control Platform Background

- Created by Sioux Technologies
- Participating organizations: 31
- Number of countries: 11
- Internal name: SAXCS
- External name: I-MECH

  (www.i-mech.eu)

V Model of Mechatronic System

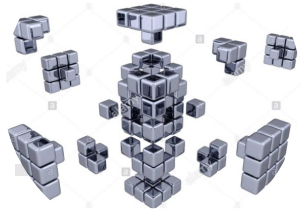


User Requirement &
Concept of Operations

System Requirement
& Architecture

Component Design

Procure, Fabricate &
Assemble Parts



Component
Integration & Test

System Integration &
Verification

System Demonstration
& Validation

Iterative research & development to make V model more agile

Enhances
Time to Market

Reduces Development
Cost

Virtual realizing the right arm of V-model

User Requirement &
Concept of Operations

System Requirement
& Architecture

Component Design

Procure, Fabricate &
Assemble Parts

Platform Architecture

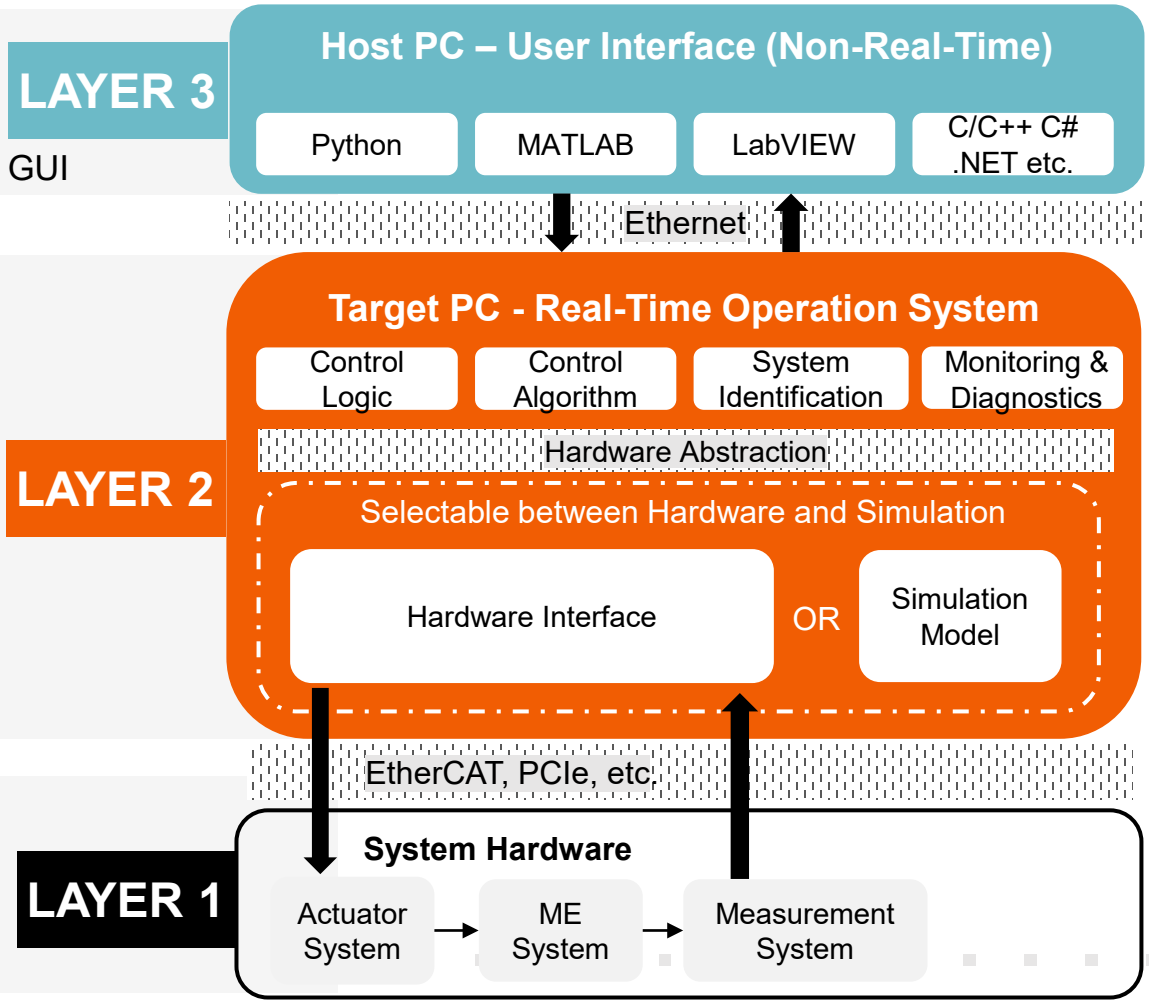
(Non-Real-Time Host PC)

- Python scripts, MATLAB, LabVIEW for quick tests GUI dev
- Use C / C++ / C# .NET to develop complicated and customized GUI

(Target PC)

- Support Real-Time Operation System
- Model-Based Design
- Auto-Generated Code
- Hardware Abstraction
- Selectable between Hardware and Simulation

- Actuator System, Mechanical System, Measurement System.
- Interface: EtherCAT, PCIe, etc.
- Compatible to commercial components and tailored I/O



Zoom in Layer 2

(Hardware Abstraction)

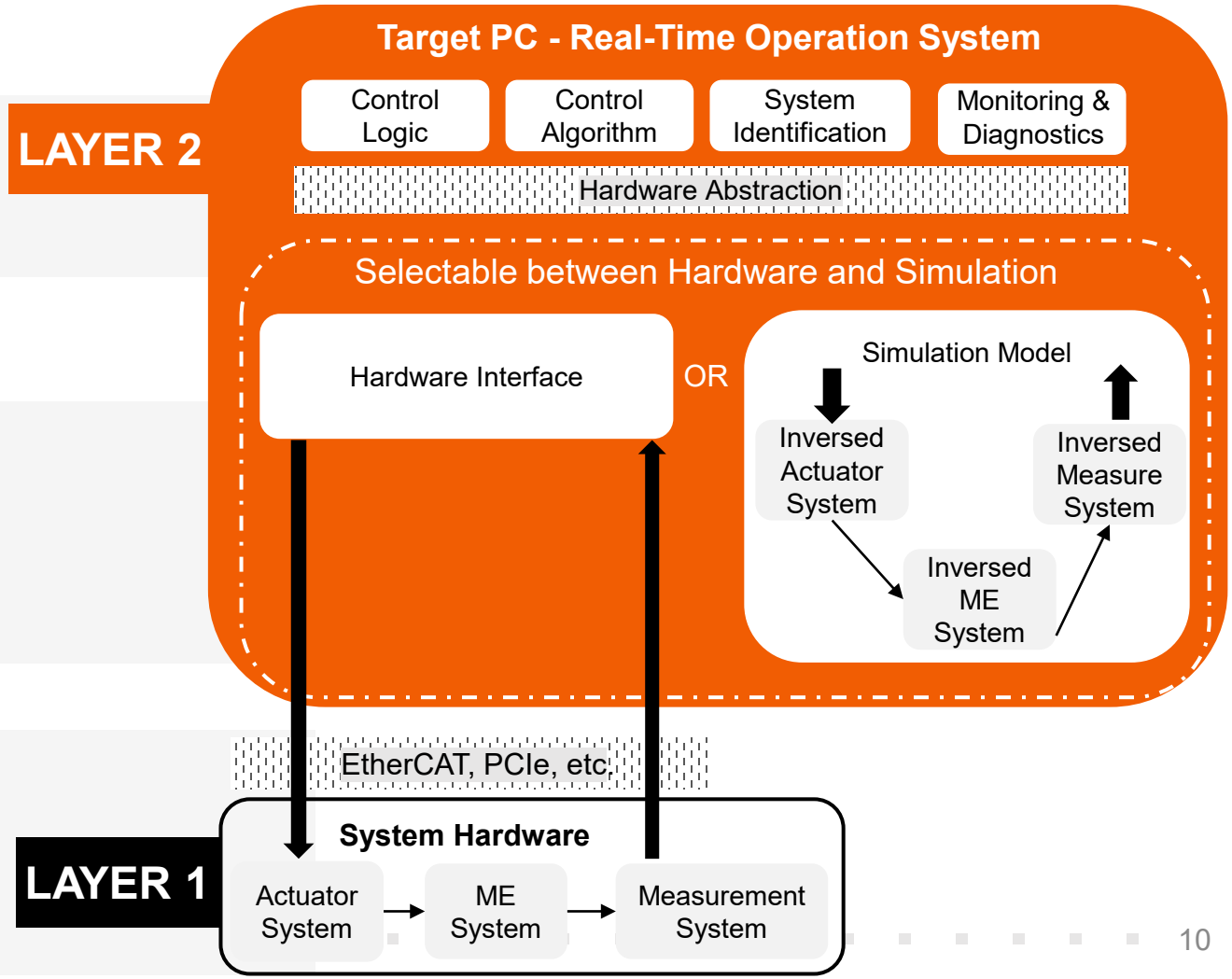
- Make the right split between embedded software designer and mechatronics designer

(Simulation Model)

- Simulated real hardware as digital twin (Inversed hardware)
- Auto-Generation RTOS codes from model

(Real Hardware)

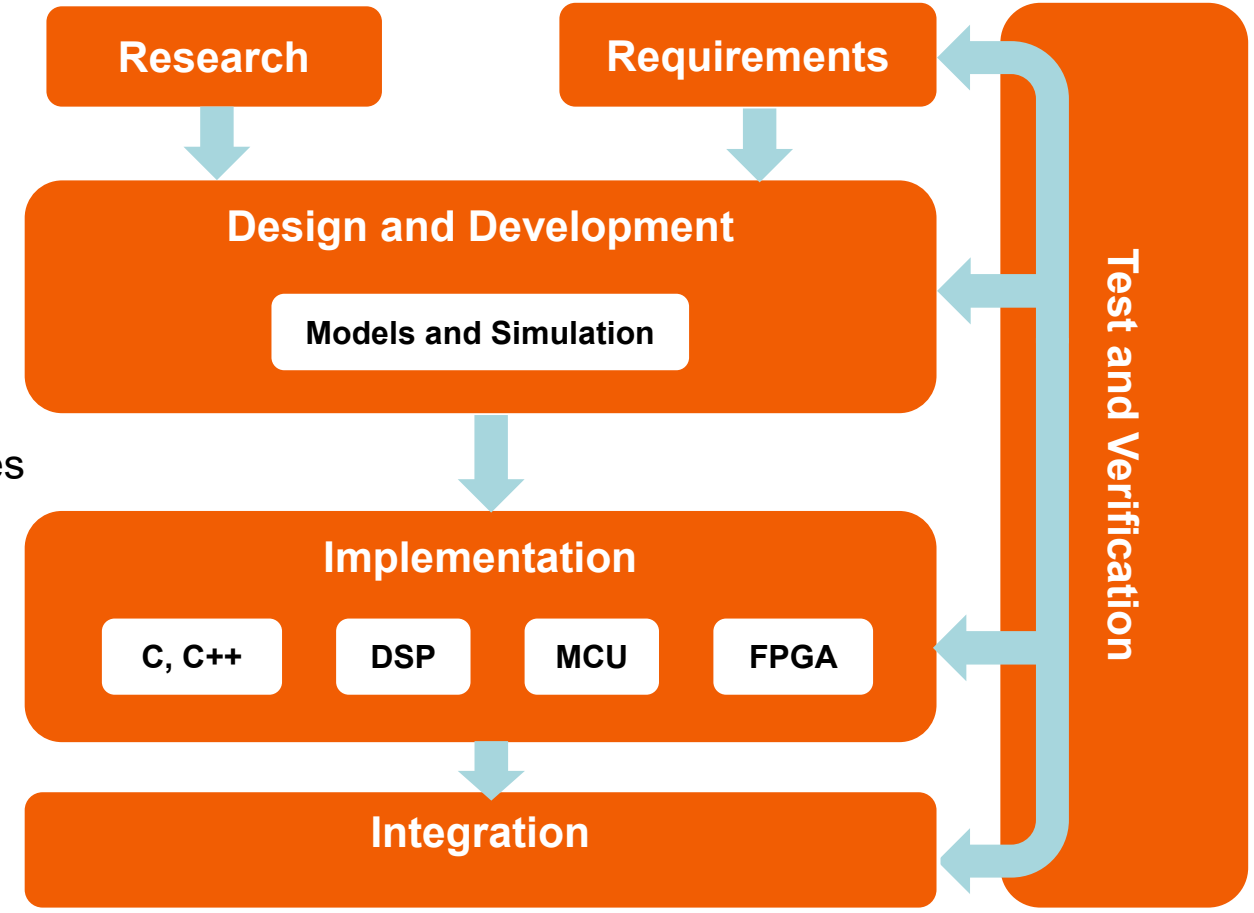
- Actuator System, Mechanical System, Measurement System
- compatible to commercial components and tailored I/O



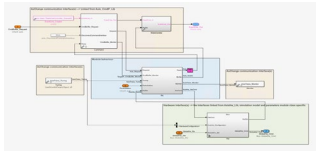
Model Based Design (MBD)

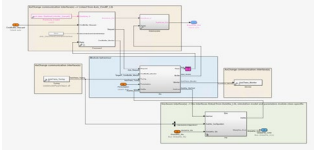
Overview

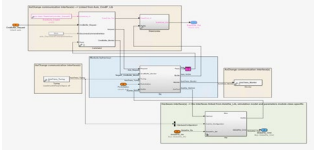
- Design and Development on Model
- Intuitive and easier to maintain
- Auto-generation from model to code
- Easy to communicate and understand across disciplines
- SIL (software-in-the-loop)
- PIL (processor-in-the-loop)
- HIL (hardware-in-the-loop)

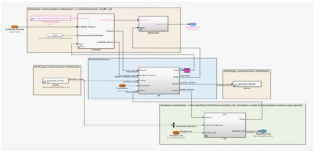


MBD: Software-in-Loop

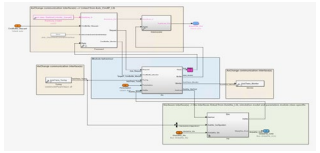




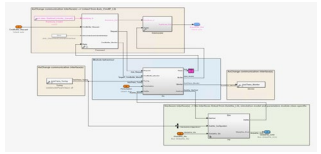


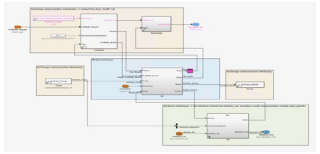


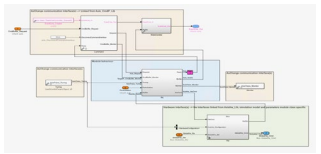
Host Computer → Motion Controller → Amplifier/ Driver → Motor → Stage → Position Sensors → Motion Controller



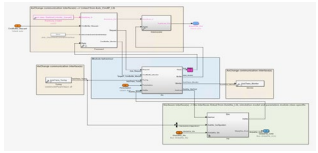
MBD: Processor-in-Loop







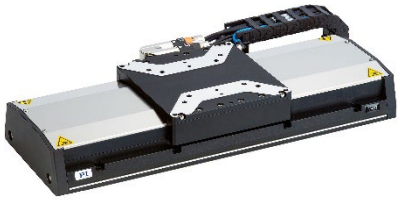
Host Computer → Motion Controller → Amplifier/ Driver → Motor → Stage → Position Sensors → Motion Controller



MBD: Hardware-in-Loop

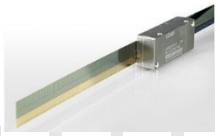






Host Computer → Motion Controller → Amplifier/ Driver → Motor → Stage

Position Sensors



Workflow

Designer

Design Side (Model-Based Design)
- Design Algorithm Model
- Design Control Strategy
- Simulation
- Optimization

Unique Data Source (XML Definition)

Auto-Generation Structure

Trigger Build

Automated Build on Server

Code Generation Toolchain

Build Finished

Cloud Sync

Change Detected & Download Binary

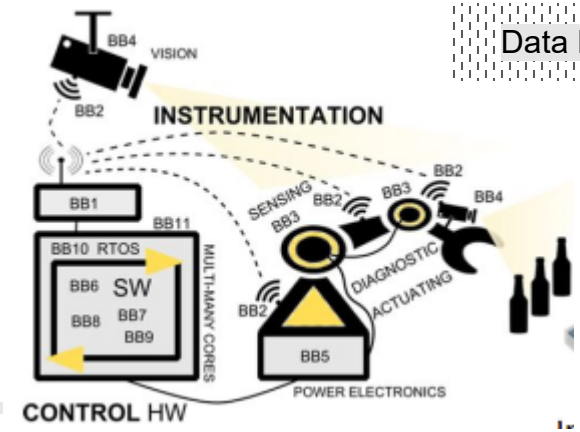
Integration Side
- Install and Configure
- Test by HIL and PIL
- Debugging
- Calibration Script

End-Users

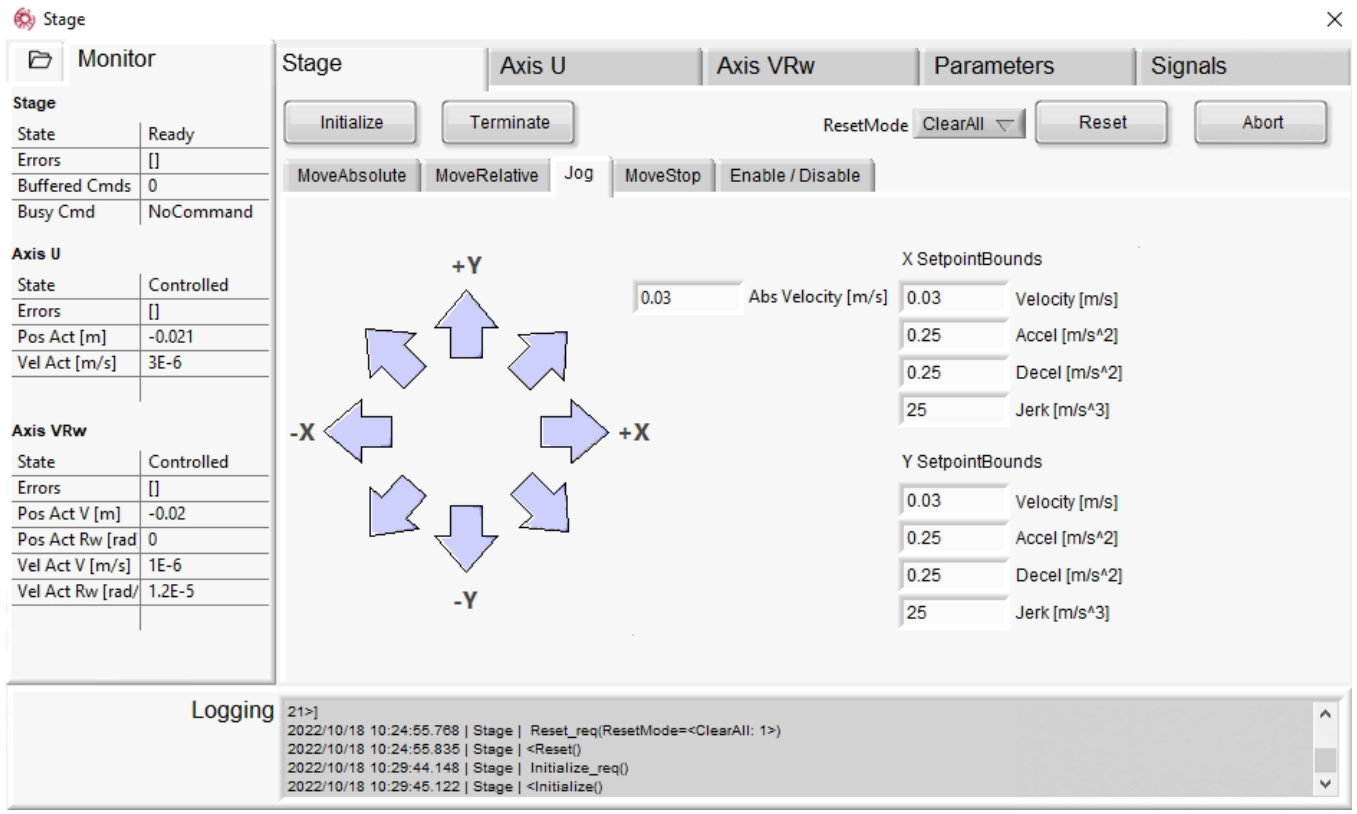
Model Changes

FRF Identification, Bode Plot, Controller Design Tool

Data Logging



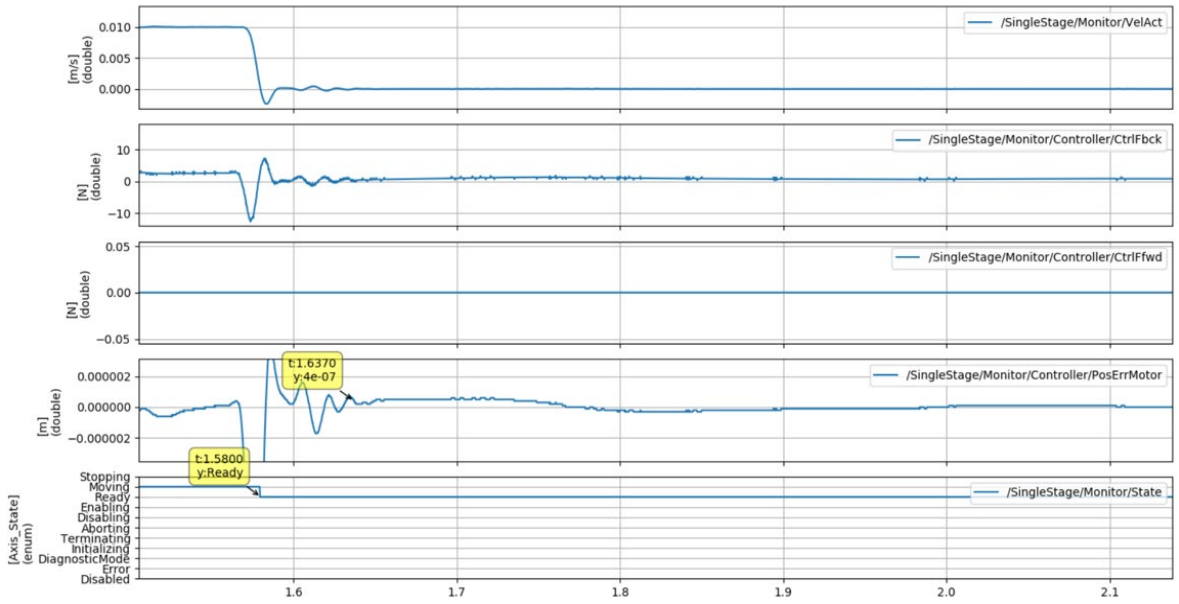
GUI and Customized Programming

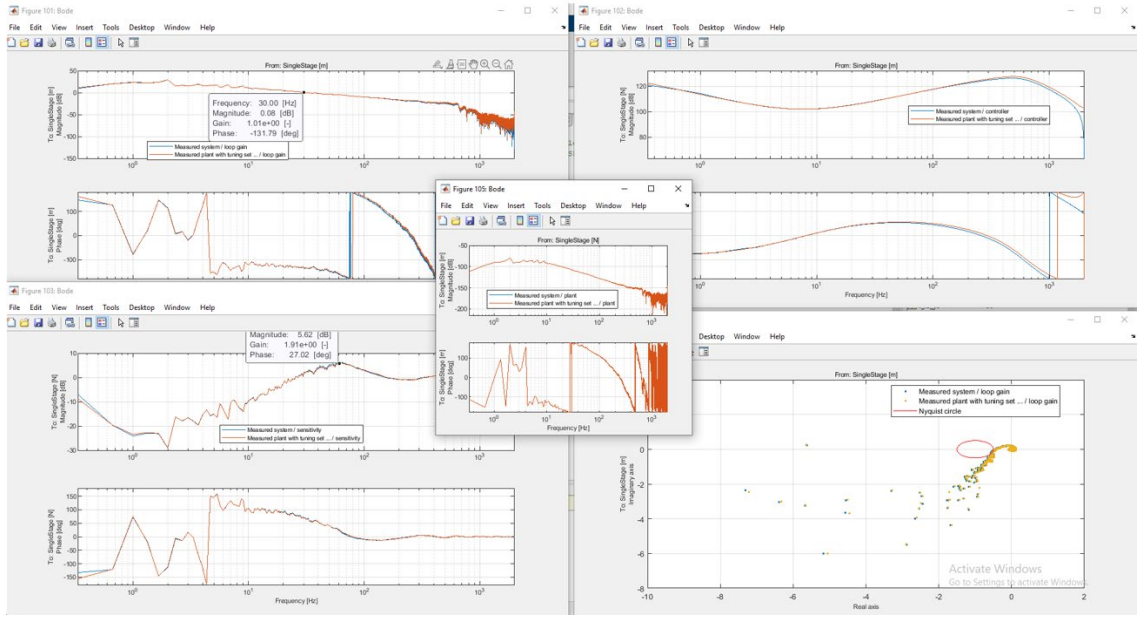



```python
#! ..\_python.bat
# first line is a so called 'shebang' and is used to executed the python script direct from explorer, refer to _python.bat for more details.
""" Tutorial command handling """
import time
from scripttools import LogConfig
from saxcscontroller.slrt.gppid import *


def main():
    LogConfig()  # refer to logging_basic tutorial
    # create the object, using the name as listed in the SystemDescription.xml
    gppid1 = GPPID('GPPID1')
    gppid2 = GPPID('GPPID2')

    gppid1.Tuning.restore()
    gppid1.Reset()
    gppid2.Tuning.restore()
    gppid2.Reset()

    gppid1.Enable()
    gppid2.Enable()

    ########################################
    # Async commands demo starts here
    ########################################

    print('first move sequential')
    gppid1.MoveRelative(Step=1, Velocity=1, Acceleration=10, Jerk=100)
    gppid2.MoveRelative(Step=1, Velocity=1, Acceleration=10, Jerk=100)
    time.sleep(1)
    print('now move in paralled, start of movements (almost) at the same time')
    # all slrt commands have a Command_req(arguments) / Command_wait(timeout) pair,
    # the _req  is used to initiate the command, without waiting for completion / exception
    # the _wait checks if the reply of the corresponding command is received, if not it waits for the reply.
    # Note: command exception and\or module error remain unnoticed ,
    #       ( i.e they remain in the communication buffer) until the _wait is called.

    gppid1.MoveRelative_req(Step=1, Velocity=1, Acceleration=10, Jerk=100)
    gppid2.MoveRelative_req(Step=1, Velocity=1, Acceleration=10, Jerk=100)
    gppid1.MoveRelative_wait()
    gppid2.MoveRelative_wait()
    # also refer to RTScheduler example for real-time synchronisation of commands
    gppid1.Disable()
    gppid2.Disable()
    print('done')
```

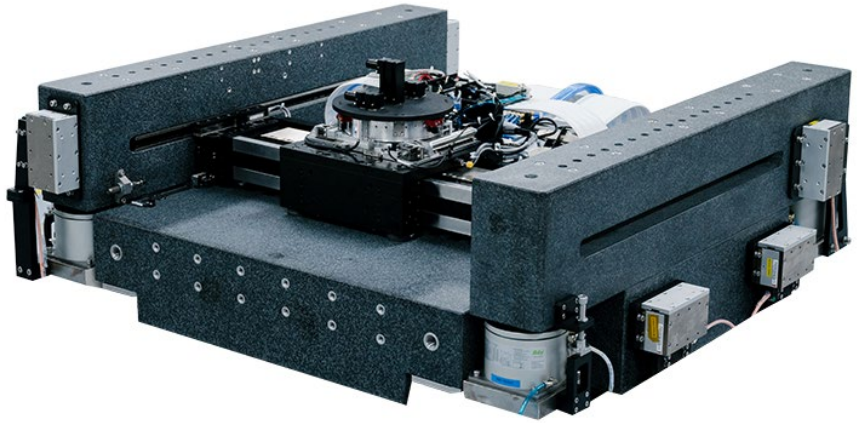
Monitoring & System Identification

- Monitor and record any signals and parameters for time domain analysis
- Analysis system frequency response function for optimizing system performance



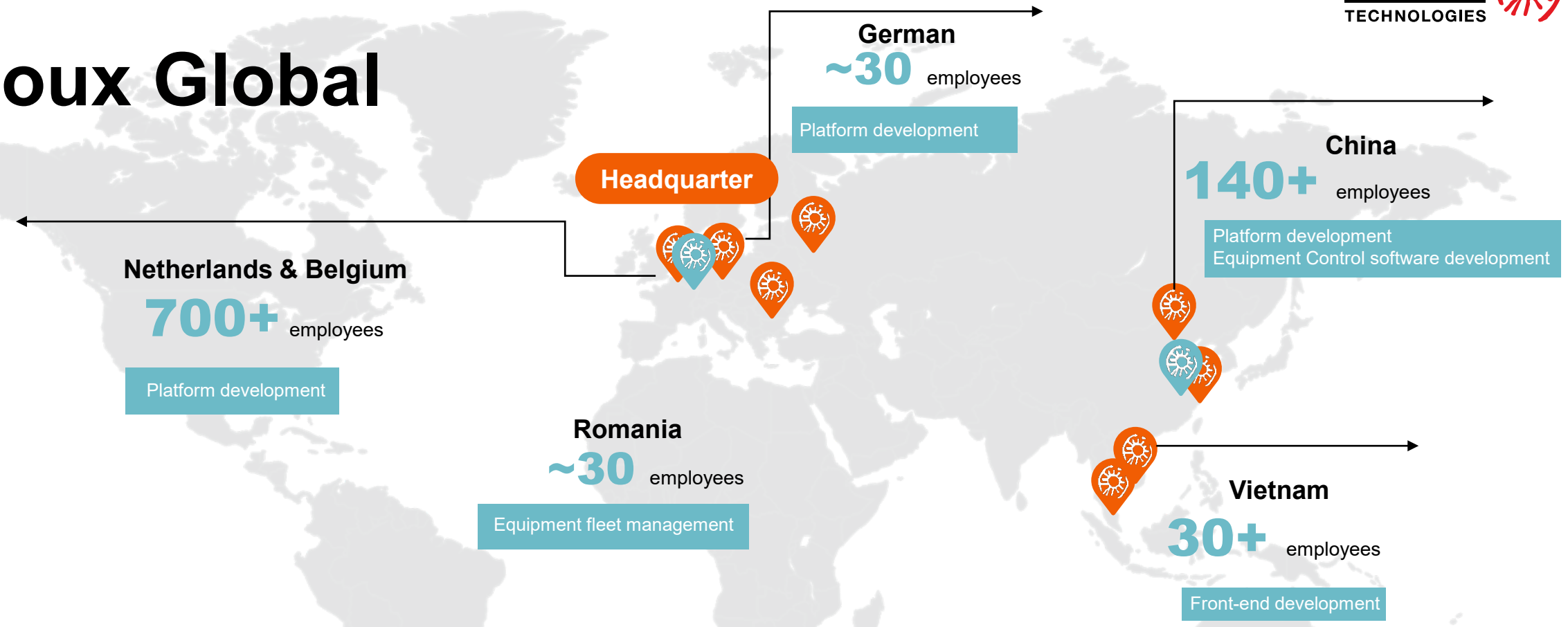


Ultra Precise Scalable System (UPSS)



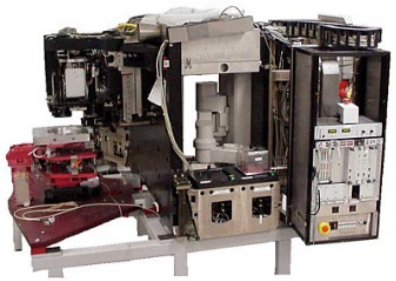
- Build with Intelligent Motion Control Platform
- XY: Up to 50nm (XY) accuracy
- Rz: 0.2urad resolution
- Customizable Active Isolator, Force Cancellation and Levelling

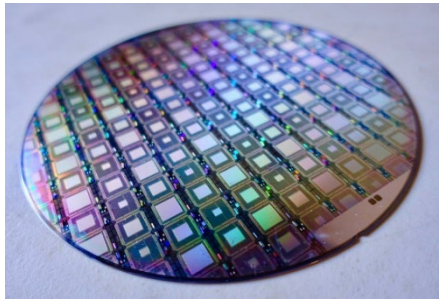
Sioux Global



Netherlands	50+ years	10	900+	>$14M
Headquarter	History	R&D centers	Engineers	Revenue

Semiconductor application





Front end

- Lithography
- Maskless imaging
- CVD,PVD
- Ion implant
- Etch
- Annealing

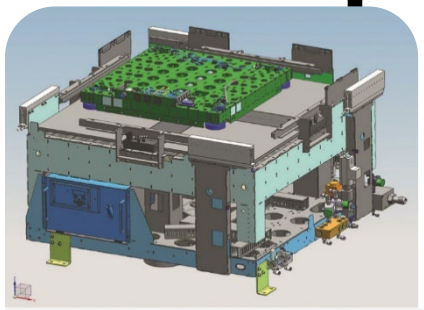
Back end

- Wafer inspection
- Laser dicing
- Wire bonding
- Advanced packaging
- SMT

Solar

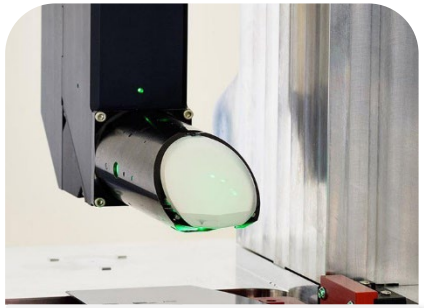
- PECVD
- ALD

Capability for Semiconductor Equipment development



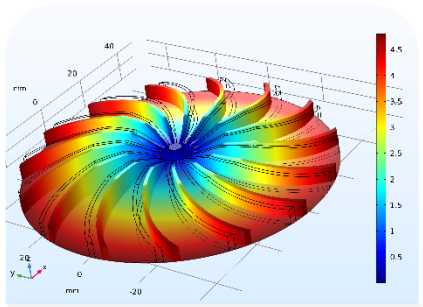
Mechatronics

- System design
- Fine mechanics design
- Dynamics modelling and analysis
- Electronics & Electrical design
- Realtime control system design



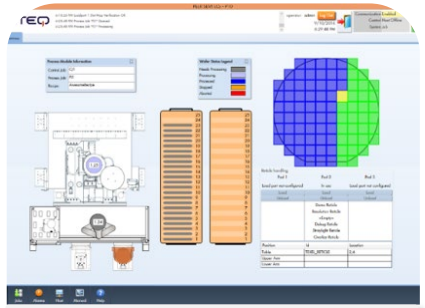
Photonics

- Precise metrology and inspection
- Machine vision
- Optical modelling
- Application of laser and sensor



Mathware

- AI and image processing
- Machine learning
- Thermo and fluids modelling
- Big data modelling and analysis



Software

- Software engineering
- Equipment control software
- Cloud platform
- Cyber security
- Localization and internationalization

UX

- Product definition
- Value proposition
- User experience
- HMI
- Data visualization
- Industry digitalization

Holistic system design capability

SIOUX TECHNOLOGIES

Sioux China

0512-62626299

info.cn@sioux.asia

Unis Building 18F, No.99 Nantiancheng Road,
Xiangcheng District, Suzhou, Jiangsu, China, 215133

We bring **high-tech**

to life