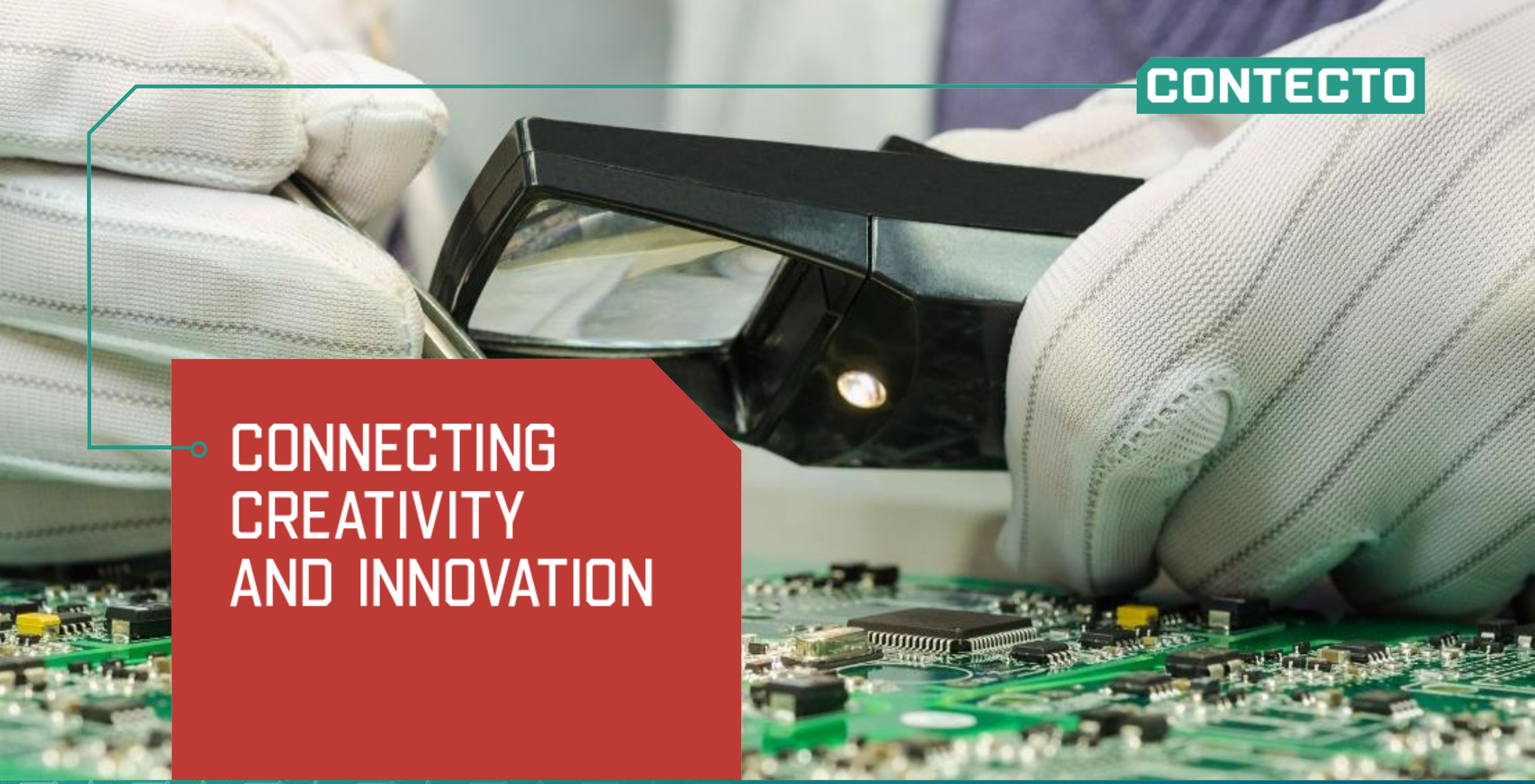
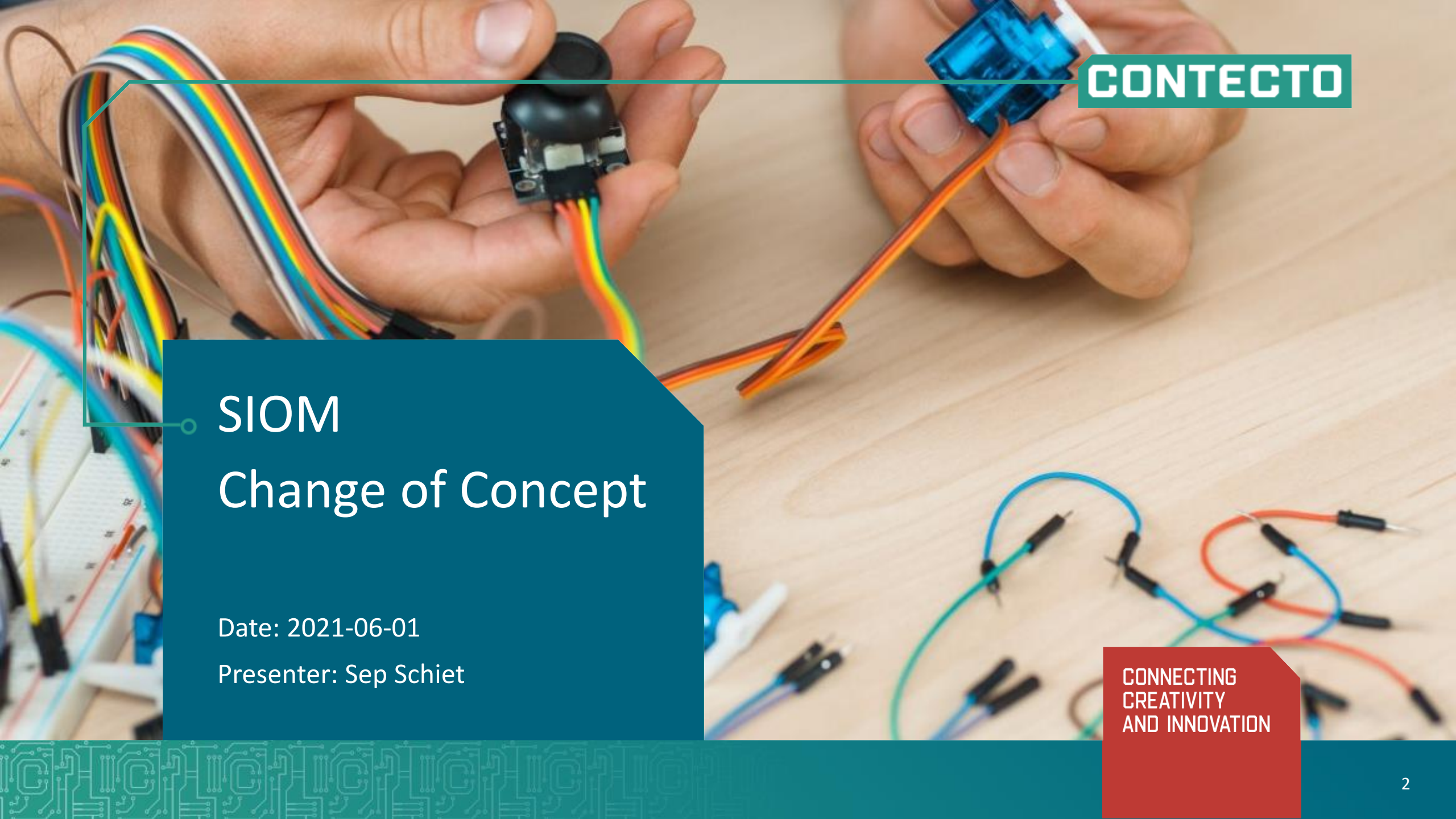


CONTECTO

CONNECTING CREATIVITY AND INNOVATION



CONTECTO

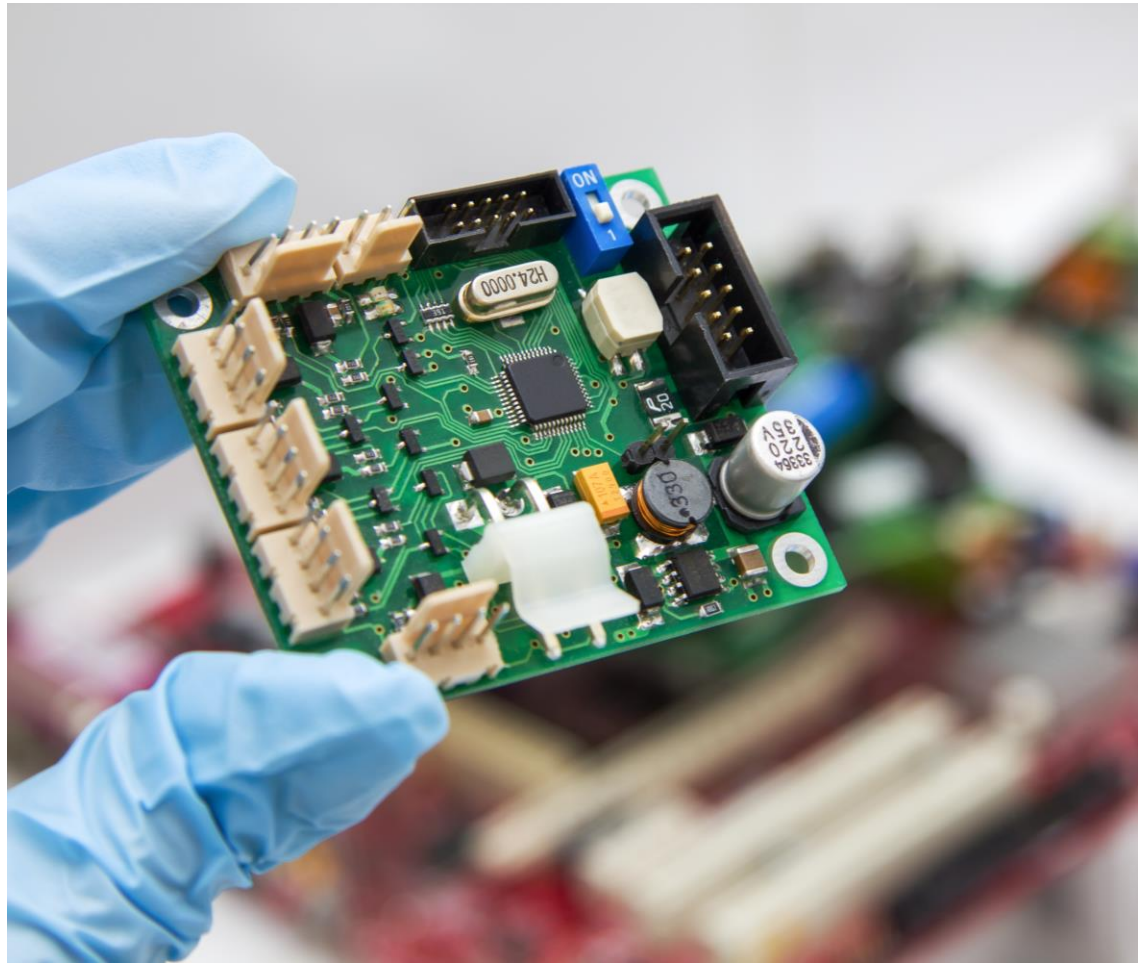
SIOM
Change of Concept

Date: 2021-06-01

Presenter: Sep Schiet

CONNECTING
CREATIVITY
AND INNOVATION

Table of contents

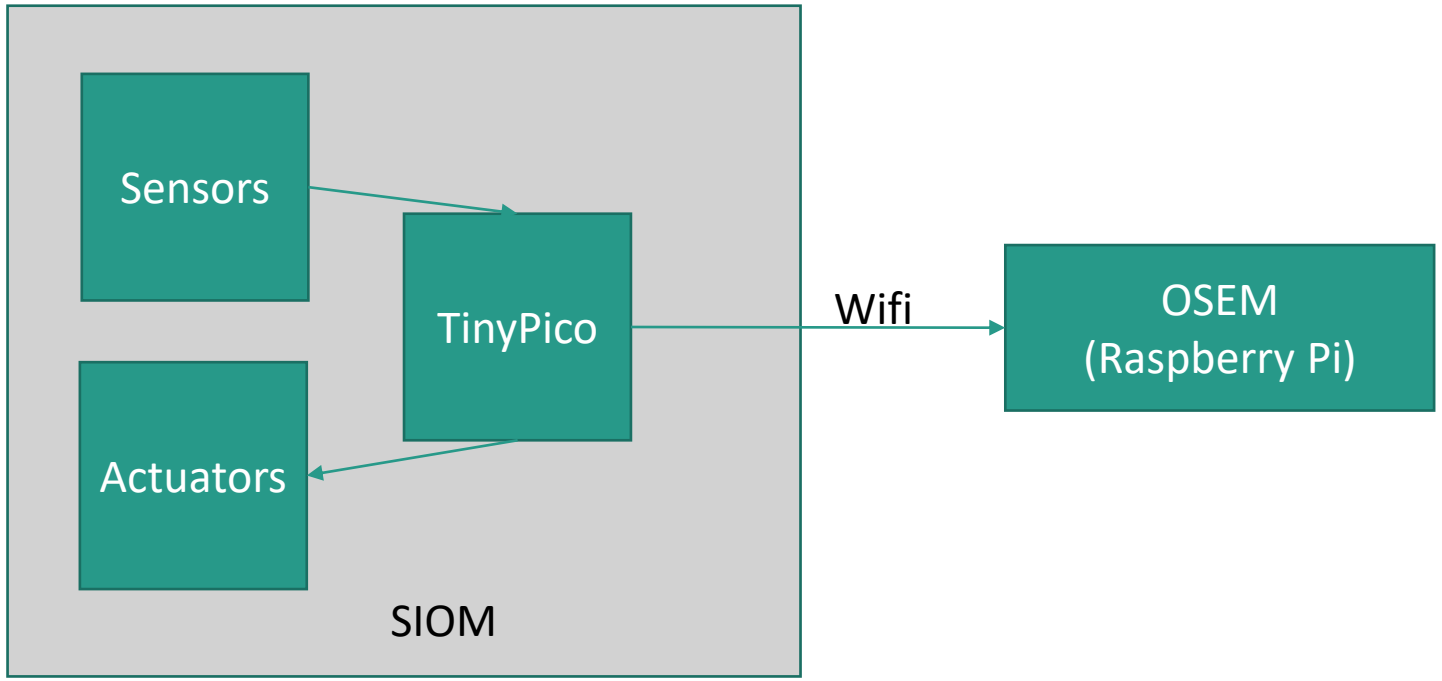


Original concept

SIOM, featuring:

- TinyPico ESP32 module with **microPython**
- Connection to input **P1 Smartmeter** port
- Connection to **8** digital **Temperature Sensors** via I2C
- Connection to **2 Flow Measurement** pulse inputs
- Connection to **2 Valve controls** via analog outputs DAC resulting 1..10V
- Connection to **spare I2C connectors** for future use
- Connection to **Heatpump** with:
  - 1x RS232/RS485 **serial interface**
  - 1x **Analog output** 1..10V or 4..20mA
- Connection to **Boiler** unit: 1x **PWM output** for electrical heater (via solid state relays)

Original concept



microPython issues

Library issues:

- Pulse input: most probably to fast without the use of interrupt sub routine (ISR). Seems no library items available.
- SPI bus for P1 port and heatpump interface unclear if available and fast enough.
- DAC for analog outputs: no library available.

Development of these is not possible within scope and financial no fit for this project. (**Major**)

IDE issues:

- The environment is not mend to program and debug as to be expected. Python has a possibility for debugging but these features are not available in microPython. The only way is way using debug-prints. For that the terminal of programming and debugging must be switched. This is really very time consuming and not useful for low level embedded topics.
- The microPython version for ESP32 (TinyPico) is limited in compare to other processors (e.g. Adafruit version).
- The IDE's are very limited or advanced although the required plugins are very limited. This results in really very time consuming development.

Conclusion

Summary:

- The IDE’s do not have a mature solution. (**Minor**)
- Some required libraries are NOT available for the TinyPico (or at all) (**Major**)

Requires to change concept!

Requirement change

- Keep open source EDA (Electronic Design Automation) [KiCad]
- Keep open source IDE (Integrated Development Environment) for Python [PyCharm Community Edition]
- Add open source IDE for C (**this is a change!**) [Eclipse/ST system workbench]
- The use of an MCU (Micro Controller Unit) as intermediary to adapt field signals to the TinyPico module
- The source of python programs will be available in GITHUB
- The source of c programs will be available in GITHUB

In summary: Everything stays the same except for adding an extra MCU programmed with the C language. The program will be provided. From a user and from a programmer point of view programming and working with the TinyPico module, not much will change. The rest of the project can continue as was designed.

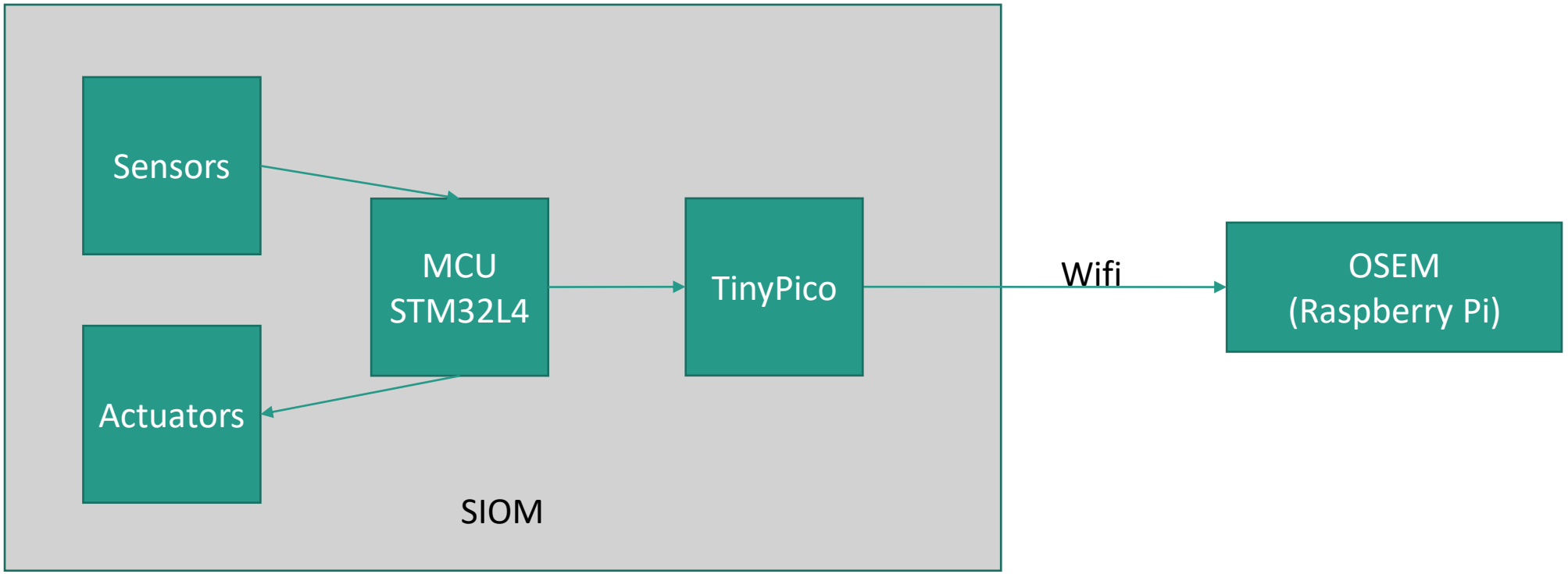
Technical side of the solution

- Add an MCU STM32L4 series processor
- Connect all field components to this processor
- Add an I2C interface to connect STM32L4 and microPython module
- The MCU will read and write the field values as simple as possible (RAW or semi-RAW data)
- A table of data will be exchanged via the mentioned I2C interface
- The tinyPico will process all data and interface via wifi and MQTT to Raspberry Pi main controller.

This will result in extra time usage to change design! (~32h)

In the meanwhile the electronic component market has “collapsed” so the availability is a very large issue. (~24h)
[e.g. MCU, power supply modules, interface chips, …]

New concept


Sensors
Actuators
MCU
STM32L4
TinyPico
Wifi
OSEM
(Raspberry Pi)
SIOM