

# **SMART** ICT BY PROMIK

A Unified Solution



# STRUCTURE

1. System Overview
2. Toolbox Overview
  - Software
  - Hardware
3. Use Cases
  - Low Current Measurement
  - Test of CAN-FD Communication Interface
4. SMART ICT Functions
5. Benefits



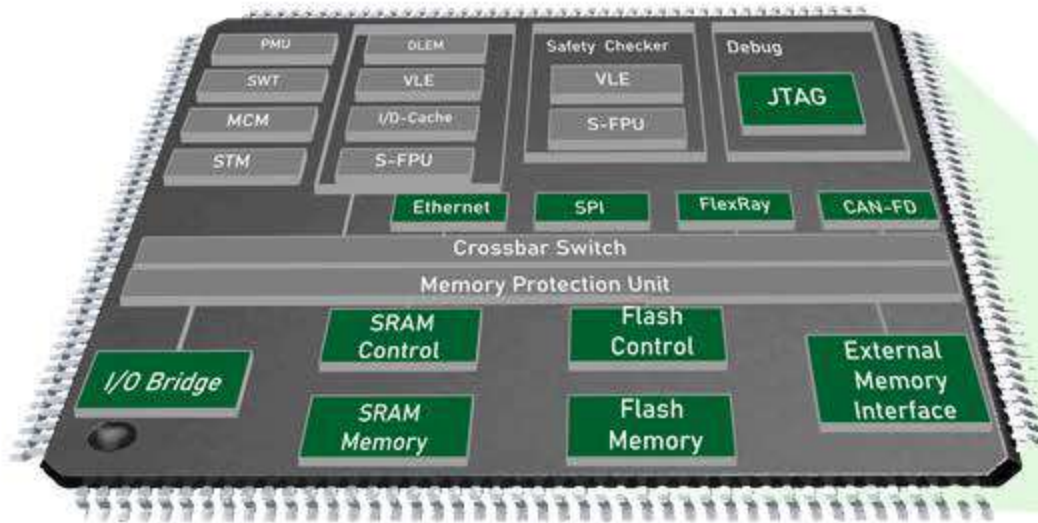


## System Overview

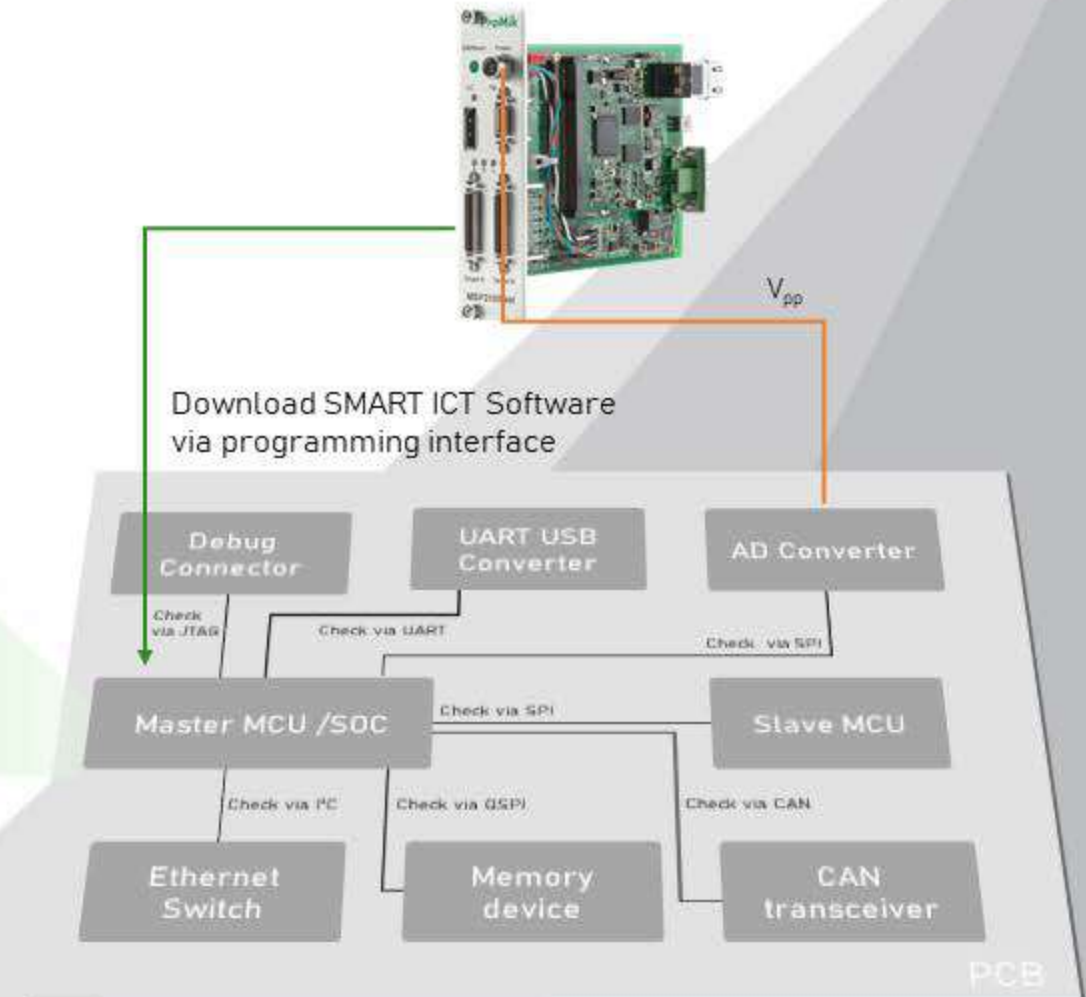


# ProMik SMART ICT System Overview

Exemplary block diagram of NXP Power PC MCU:  
(Green frames indicate testable components)



Internal Device Structure



Full application view





## Toolbox Overview



# ProMik SMART ICT Toolbox Overview

ProMik programming solutions as basis module



SMART ICT software as a Plug&Play upgrade to existing FlashTask software

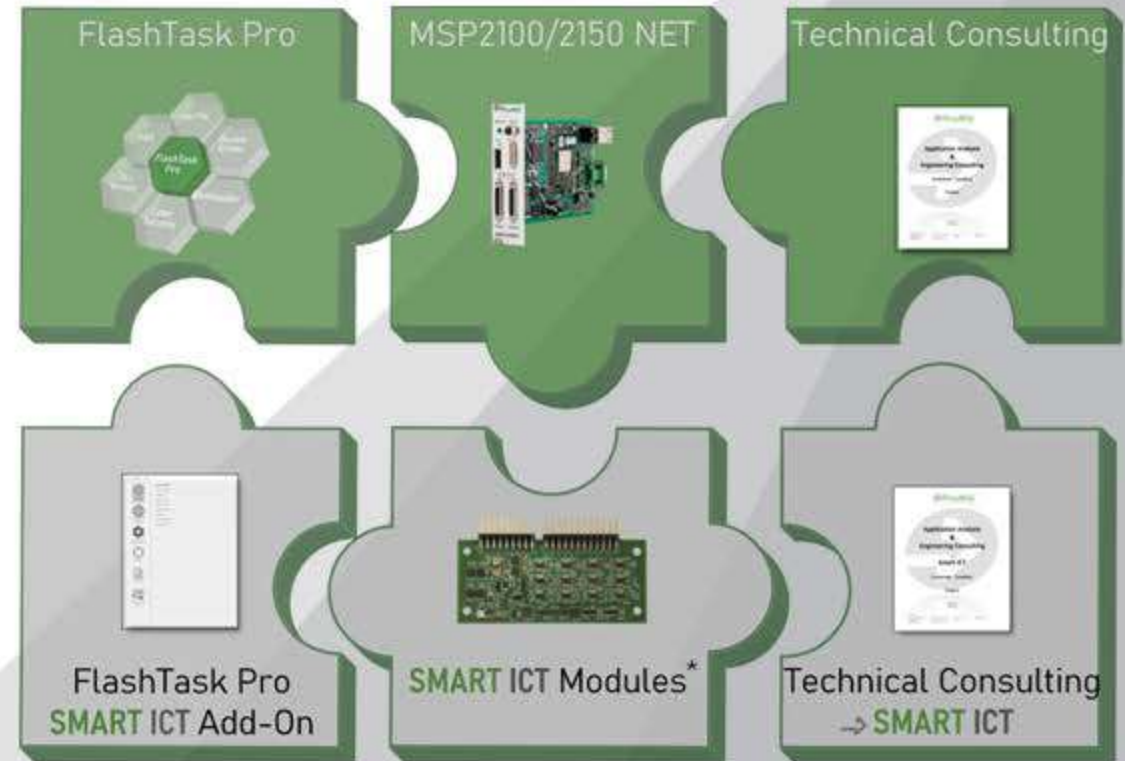
**Flashing Solution**

**SMART ICT Upgrade**

SOFTWARE

HARDWARE

INTEGRATION SUPPORT



\*optional



# ProMik SMART ICT Software: Scripting Language

Easily manageable  
scripting language

Simple integration

Intuitive parameter definition

Calling predefined  
programming sequence

Flexible SMART ICT function call

Individual SMART ICT sequence

```
1
2 msp = MSP2150net()
3 powerPc = MPC5643L()
4 canIface = CAN()
5
6 msp.log("application power on")
7 msp.power.on()
8 msp.setWatchdog(2000) # 2 kHz watchdog
9
10 canIface.testerpresent(0x380, # CAN RX ID
11                       0x381) # CAN TX ID
12
13 msp.log("application unlock")
14 if not powerPc.init(15000, # JTAG frequency
15                   4, # cable Compensation
16                   20000) # xtal Clock
17     msp.log("unlock device via CAN")
18     canIface.can_unlock(0x750, # CAN RX ID
19                       0x751) # CAN TX ID
20     powerPc.init(15000, # JTAG frequency
21               4, # cable Compensation
22               20000) # xtal Clock
23
24 msp.log("application program")
25 powerPc.erase()
26 powerPc.data.addfile("data.bin", # Data file
27                    0x80000) # Offset
28 powerPc.program()
29
30 while powerPc.busy(): # Busy wait for programming to finish
31     # Perform other tasks...
32
33 msp.power_cycle()
34
35 msp.log("application perform current measurement")
36
37 sleepCurrent = msp.measure_sleep_current_ms()
38 msp.log("measured sleep current [ms] " + sleepCurrent)
39
40 canIface.wakeup()
41 runCurrent = msp.measure_run_current_ms()
42 msp.log("measured run current [ms] " + runCurrent)
43
44 msp.power.off()
45 msp.log("application power off")
46
```





## SMART ICT Functions





# ProMik SMART ICT Function Overview



## Generic Test Libraries

- Contain device specific modules
- Intuitive function selection allows project specific Smart ICT sequence configuration



\*Additional Hardware required



# ProMik SMART ICT

## Toolbox: Example Hardware Modules

SMART ICT Module	Description
Power Control Current Sensor (PCCS)	To measure current consumption in $\mu\text{A}$ / $\text{mA}$ ranges (perfectly suited for run-/sleep current measurement)
Galvanic Isolation Module	Separates electric circuits physically in order to avoid influences on flash signals
Power Sequencer Box	Powers up the application in a certain sequence
Frequency Measurement Module	Measures signal frequency
Level Shifter	Adjusts voltages accordingly
LIN Multiplexer	Multiplies LIN signals
Watchdog Trigger	Triggers watchdog to avoid the device to shut down
PSU2048	4 independent output channels 4-50V / 4A / 50 Watts per output
Fuse Charge Pump	Sets fuses of Programmable Devices





## Use Cases



# ProMik SMART ICT

## Use Case: PCCS Module + LIN Multiplexer

### MCU Access

- Download ProMik SMART ICT Software via flash test pads.

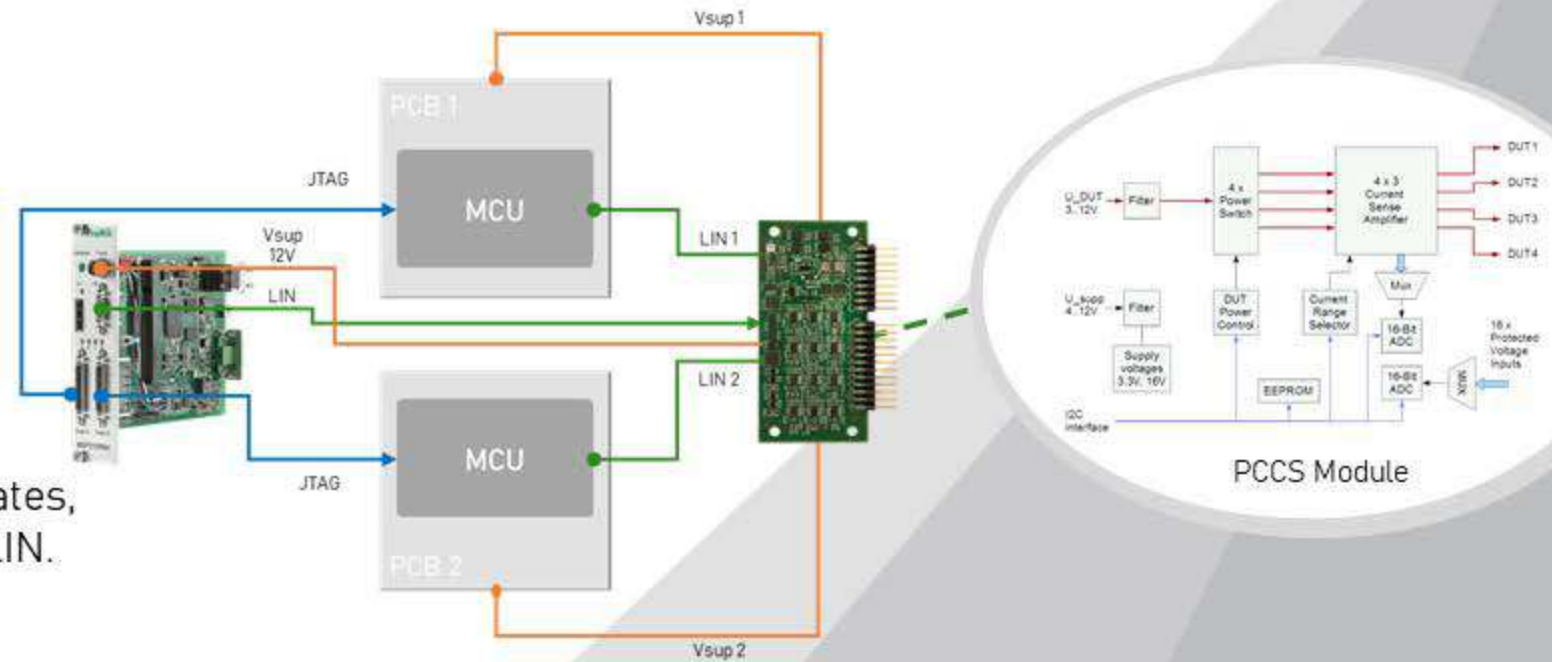
### Communication

- Activate specific application states, like Run- and Sleep mode via LIN.

### Measure Current Limits

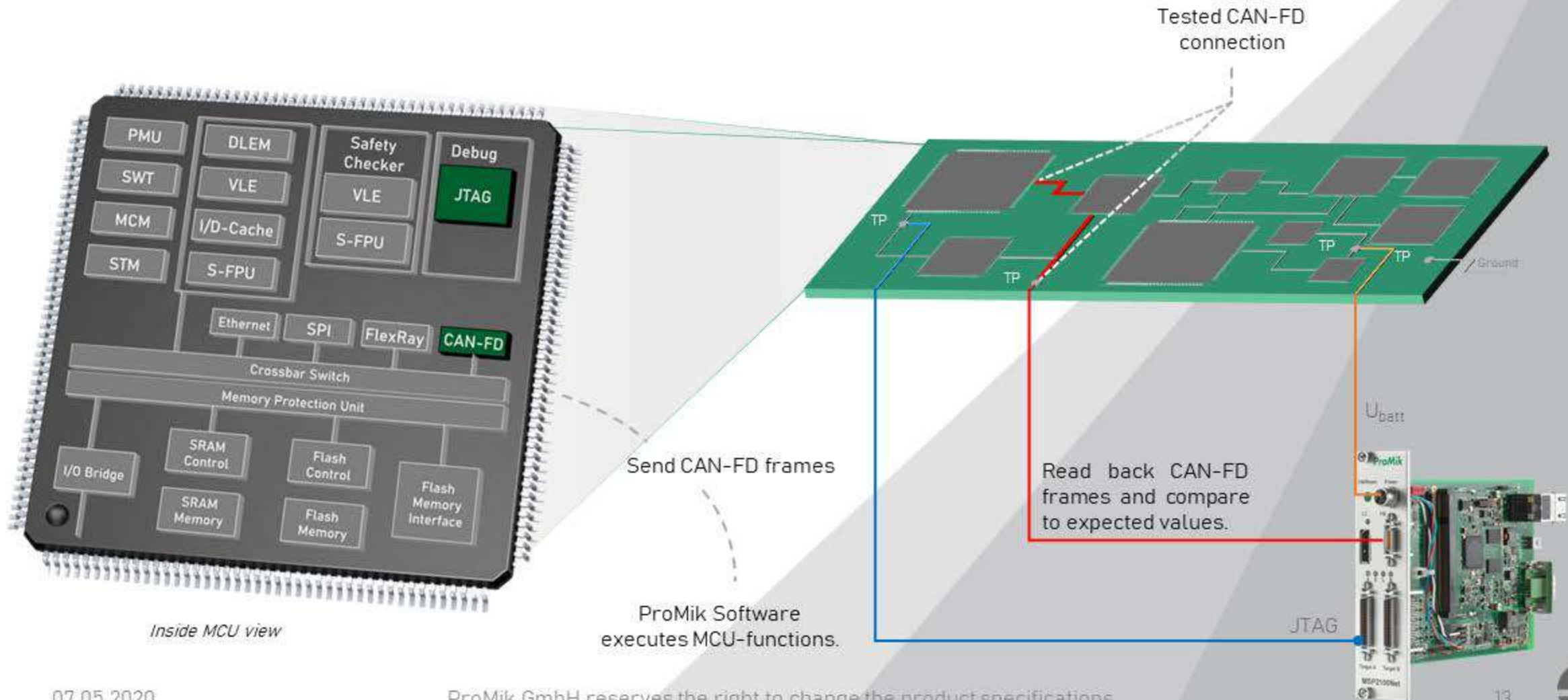
- "State 1" measurement (800  $\mu$ A)
- "State 2" measurement (< 10  $\mu$ A)
- "State 3" measurement (< 100mA)

Traceability through  
MES connection



# ProMik SMART ICT

## Use Case: Test of CAN-FD Communication Interface



Benefits



# ProMik SMART ICT

## Benefits

### **Ideally suited for small applications with less or without test pads**

- Using target device flash interface
- Parallel access on panel level
- e.g. camera applications, key applications, sensors

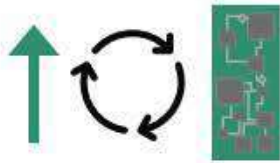
### **Dynamic control of generic library**

- Test engineer has full control over test routines, allowing dynamic test coverage
- Flexible use for various applications
- Configurable I/O lines, interface channels & routines, analogue functions, etc.



# ProMik SMART ICT Benefits

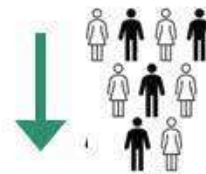
## Costs saving potential



Increased output rate



Hardware cost reduction



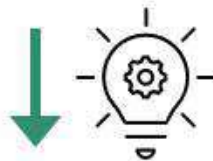
Labor cost reduction



Floor space reduction



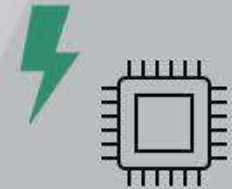
Cost-effective upgrade  
for ProMik FlashTask Pro



Lower process complexity



Reduced und balanced  
cycle times



Early identification of  
defective parts

