LabVIEW Development Platform

Graphical Programming: Rethinking Embedded Systems

For complex projects and tight time-to-market, **developers** appreciate the Zbrain platform with graphical LabVIEW programming.

Decision-makers can scale seamlessly and efficiently from feasibility studies and MVPs to prototypes, series products and testing.

Schmid Elektronik AG | Mezikonerstr. 13
CH-9542 Münchwilen, Switzerland | schmid-elektronik.ch

▶Every new project or product development begins with the following questions:

Know-how:	Are we in control of complexity?
Dates:	Are we keeping to the time-to-market?
Costs:	Are we staying within budget?
ROI:	Will the investment pay off?

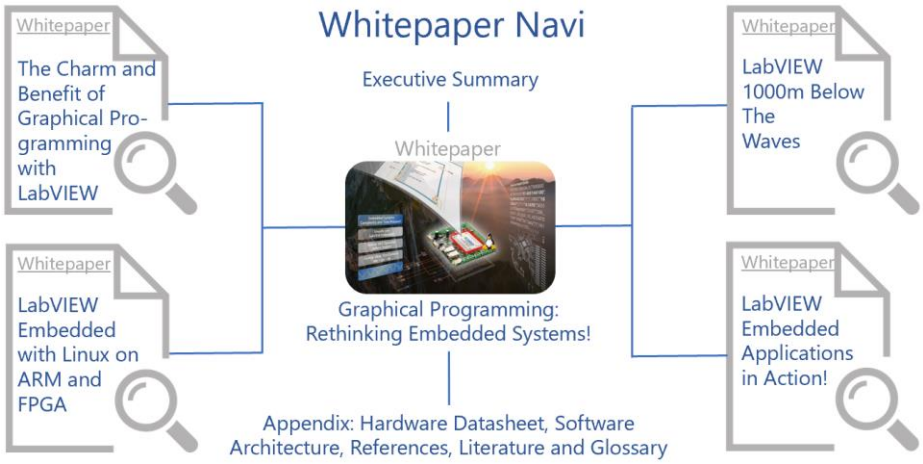
▶ The more of these questions and uncertainties are clarified in this phase, the faster the self-confidence to break new grounds. It's like a bold climber taking unknown paths and trusting in the safety rope.

▶ The Zbrain development platform with graphical NI LabVIEW and hardware and software modules is like a climbing garden. It encourages trial and error and minimizes development risks on the way to an embedded system.

Find out in this white paper...

- how "LabVIEW Embedded" takes advantage of abstraction to simplify project complexity through the ease of use of hardware and software.
- how the development of embedded systems uses the same approach over microseconds to nanoseconds.
- why a consistent programming language for proof of concepts, MVPs, prototypes, series products and tests reduces time pressure and shortens time-to-market.
- where the development platform is used:
  From deep sea measuring networks to on-track IoT things.

The white paper has a modular structure. Four linked whitepapers offer in-depth insights. If required, you can consult the appendix, the references or the glossary. Here is your "Navi":



In addition, internal document links offer shortcuts to the topics that interest you.
Note: press [ALT] + left arrow key to return to the calling link.

1 Executive Summary with Shortcuts to Main Topics

Master Complexity and Stay Ahead of the Game

Data is at the heart of modern embedded systems: obtained from physical processes, calculated in real time, available everywhere via IoT and part of dynamic control processes [1]. The required real-time behavior and process modeling inevitably mean greater complexity. The Zbrain platform with LabVIEW offers a solution for teams under time pressure and with limited resources .

Access to Embedded Technology

Graphical programming with LabVIEW increases productivity thanks to its intuitive, modular approach [2]. "LabVIEW Embedded" extends this idea to microcontrollers and FPGAs [3].
The Zbrain platform combines this with a s toolbox, opening up new application possibilities for engineers and scientists. Even system developers without knowledge of C/C++ can now become active in the embedded area. With the advent of "low-code" and "no-code", the demand for LabVIEW's graphical data flow paradigm is increasing. This link takes you directly to LabVIEW Embedded technology. Click here to move to the Zbrain platform.

From Idea to Implementation with a Single Approach

Start-ups, family SMEs and global players today rely on graphical programming of embedded systems with the Zbrain platform. They regard it as a reliable partner that accompanies them from the initial idea through proof of concept (Poc) and MVP to prototype, series production and testing. This comprehensive support saves every hour of thinking and programming work invested in all project phases. If you are particularly interested in the use of the platform in these project phases, you are welcome to read on here.

Tested 1,000m below Sea Level

The first project in the Norwegian deep sea was groundbreaking for LabVIEW Embedded [4]. High demands, away from the lab, led to success and the Zbrain platform. Concerns about the reliability of graphical programming with LabVIEW on embedded hardware are put to rest with this project. Moreover, it inspired a diverse community with unconventional applications: six-legged robot, two-liter satellite, measuring cell in the wheel hub, handheld for elevator certification, telemetry in eco-racing or the operation of a small solar power plant with 80% efficiency [5]. Are you primarily interested in the practical applications of the platform? You'll find the answers here!

H to Get Started with this Platform?

Immerse yourself in the world of graphical programming of embedded systems with the starter kit. Discover the technical details of the Zbrain platform in the WIKI. Or contact us directly - tell us your challenge and together we will find out whether the Zbrain platform is the right solution for you. Here are our contact details.

Contents

2 A Platform with LabVIEW and a Hardware/Software Toolbox

2.1. LabVIEW as a general tool for measurement and control

The name "LabVIEW" comes from "**Lab**oratory **V**irtual **I**nstrument **E**ngineering **W**orkbench". It is an integrated development environment and was created by NI (formerly National Instruments, now part of Emerson). The importance of the graphical programming language "G" and its underlying architecture is explained in the whitepaper The Charm and Benefit of Graphical Programming with LabVIEW. LabVIEW is a powerful and widely used tool that offers solutions for many measurement and control applications and is known as a development accelerator. The advantage: hardware and software integrates seamlessly and thus abstracts technical complexity. Despite its graphical concept, LabVIEW uses familiar programming principles such as data types, variables, loops, branching and object-oriented approaches. This distinguishes it from low-code and no-code approaches.

2.2. Problem areas in embedded systems require a new way of thinking

LabVIEW is often mentioned in the same breath as laboratory work or testing. The Zbrain platform, on the other hand, scales further, namely into the embedded world. There, it offers solutions for three sensitive problem areas and the development process is accelerated by a factor of two to four:

1. **Implement ideas quickly**: Experience from numerous projects proves that, thanks to graphical programming, we can efficiently utilize the creative in us and thus boost our brain power. This well-known strength and the charm of LabVIEW can be transferred to the embedded sector.
2. **Project Complexity** (lots of connections, very dynamic): Thanks to graphical abstraction, complicated and even complex systems remain manageable in the embedded area. Underlying hardware, low-level drivers, timing, operating system, multitasking, multicore and tools are conveniently abstracted so that the developer can concentrate on the essentials: application development. The apps can be loaded onto the hardware and executed there in real time at the touch of a button without any detailed knowledge.
3. **"Time wasters" during implementation**: similar to Lego®, we use a modular hardware and scalable software toolbox. This reuse speeds up implementation and reduces risks.

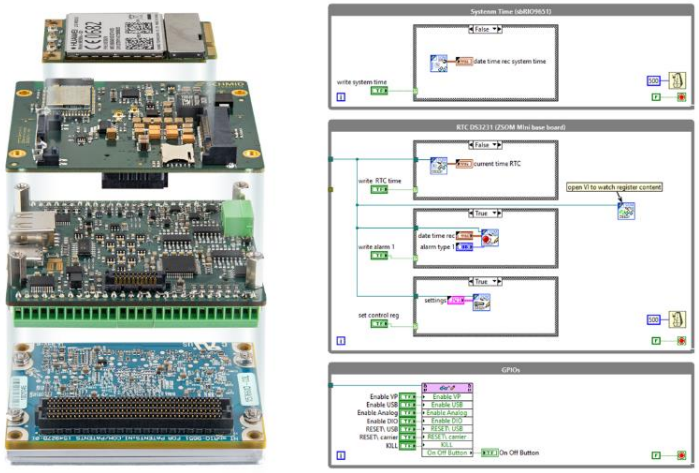


Fig.1 | On the left, an example of modular hardware. On the right, an example of graphical abstraction of multitasking, which is a lot easier and intuitive to implement than in traditional programming languages.

2.3. LabVIEW Embedded on Microcontroller and FPGA

In the early days of "LabVIEW Embedded", the traditional developer community was rather skeptical about graphical programming of embedded systems. On the microcontroller, text-based languages such as C/C++ together with microkernel, RTOS or Linux were the top dogs.

With the aforementioned pain points of modern, connected applications, the demand and desire for new ways of thinking and methods increased: more abstraction in the software without losing touch with the hardware and timing. This is exactly what LabVIEW does! After all, this graphical environment has grown up in the test environment with analog and digital signals.

We can trust its functional correctness. It has established itself as the de facto standard for testing electronic products in the industrial, transportation, semiconductor, military, aviation, space and medical sectors.

Fig.2 | Combination of two standards: LabVIEW for Microcontrollers and FPGA from NI (left) and own hardware from NI partner Schmid Elektronik (center) leads to embedded systems for series products (right)

"LabVIEW Embedded" goes a step further than testing and enables series products based on bespoke hardware outside the NI standard. The first large-scale project developed using this method in the Norwegian deep sea dispelled concerns about graphical abstraction and provided answers to the following requirements: scalable power consumption down to the milliwatt, battery operation, fast boot times, micro- and even nanosecond real-time, robust 24/7 operation and small form factors. The technology behind this approach is described in the whitepaper LabVIEW Embedded with Linux on ARM and FPGA.

It is probably for all these reasons together why "LabVIEW Embedded" has made its way into the embedded sector and established itself in certain niche applications.

2.4. The power of graphically programmed embedded systems

Typical Zbrain applications are characterized by the following features compared to "normal" LabVIEW programs:

1. The applications call for a well thought-out software architecture
2. They are real-time capable and the measured data acquisition scales up to MHz
3. They must function reliably around the clock
4. Some applications are mobile and run on battery power

In this league of applications, software engineering, hardware abstraction and scalable software architectures are just as important as in the traditional approach. When implementing the graphical LabVIEW code, it is advantageous to think in resource-saving "C". This means, for example, avoiding dynamic memory or overloaded data types, keeping an eye on memory leaks and focusing on a consistent error detection and correction scheme.

3 A single Language from Idea to Implementation

The very same programming language can be used for the entire value chain: From the proof of concept and the MVP to the prototype and the series product through to testing [10]. This shows similarities with the portability of Linux, which is why LabVIEW has already been called an "engineer's operating system". Another advantage: the software and application IP remain in-house!

3.1. LabVIEW for Proof-of-Concepts

The goal here is to create a foundation for decision-making on the technical feasibility of a project. Several options should be explored and the best one determined. LabVIEW accelerates this project phase by offering the right programming model for various problems:

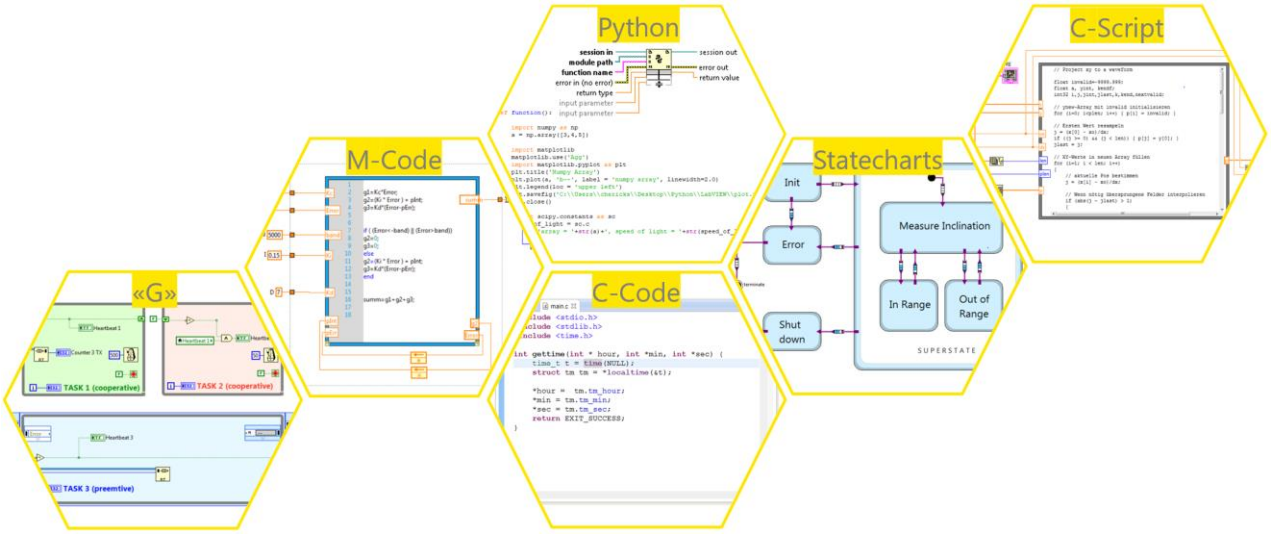


Fig.3 | The right programming model for every task,
merged in LabVIEW and executed in real time on embedded hardware.

Tricky control engineering tasks may be implemented and executed in Matlab notation. This code can be inserted into the graphical LabVIEW block diagram in the same way as software written in Python. Statecharts simplify complicated sequential logic and a C code interface enables the integration of existing algorithms or a direct access to Linux services. Then there is a variety of useful libraries, for example for mathematics, linear algebra, signal processing and toolkits such as filters, sound & vibration, machine learning and more. In addition, there is real-time synchronization of decentralized systems, display support and data communication (Ethernet, Wifi, 4G). The LabVIEW programmer can use all of this immediately without time-consuming development to prove the concept.

3.2. For Minimum Viable Products (MVPs)

At the same time, it is of advantage to put not only the technology, but also the product idea itself to the test, to enter into dialog with customers quickly and to gain feedback.

The key questions in product development are: does the market need my product and are customers ready for my new business model? Minimum Viable Products (MVPs) provide honest answers to these questions.

MVPs only consist of the most essential functions, can be implemented with minimal effort, offer a useful benefit right from the start and can be scaled iteratively, flexibly and quickly. The problem, theory and practical examples are explained in concrete and tangible terms in a separate white paper [6].

Fig.4 | Fast-moving MVPs can be iterated more quickly thanks to the development accelerator and modular hardware. IoT MVPs in no time thanks to a framework!

3.3. For prototypes

As soon as the technical feasibility has been proven and the idea has been validated on the market with an MVP, a prototype should be following immediately. Stakeholders want to literally hold the design in their hands to understand it. A prototype no longer differs significantly from a series product in terms of its form factor and the chosen materials. For this reason, todays prototypes often show the maturity of a series product, called a “proto-series”.

Fig.6 | Prototypes serve to "grasp" the design

The trick: we create a prototype by combining the LabVIEW code from the proof-of-concept and the MVP on standardized embedded hardware: e.g. on a modular and extendable ZSOM-Control or ZSOM-Mini. We use a scalable software architecture for this purpose. This porting of the graphical code leads directly to a functioning prototype without further hardware development.

3.4. For series products

The prototype is finally converted into embedded hardware, suitable for the series device. This is compact, individual, perhaps mobile, tailored to the user's needs and inexpensive. In traditional practice, this meant developing new hardware and software based on a microcontroller with RTOS and C.

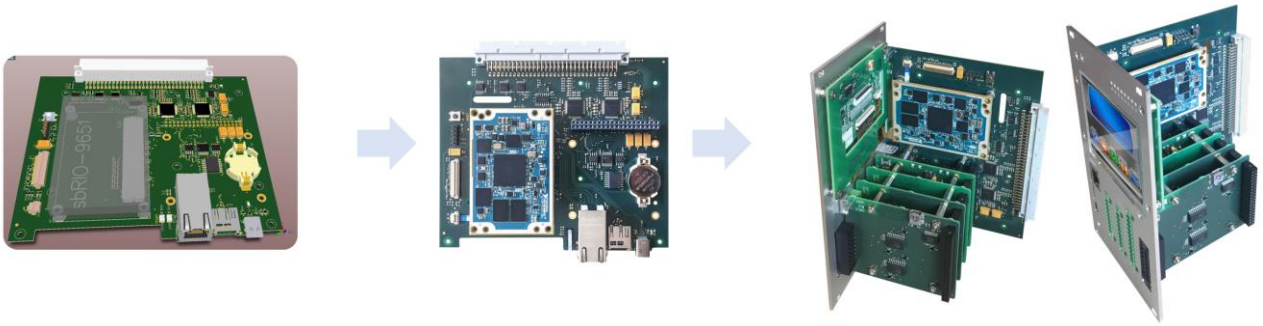


Fig.7 | Thanks to the reuse of field-tested hardware and software,
the risk to your own form factor in the series product is manageable.

How can we avoid this duplication of work with the Zbrain platform? In addition to the development tools, the platform contains a range of embedded hardware modules. These modules are based on a field-tested hardware circuit library. For each of these circuits there is a low-level driver and a LabVIEW VI (Virtual Instrument = function block). The trick now is to reuse them: A customized baseboard is created from the existing circuit library and additionally required functions. This baseboard is operated with the same Microcontroller/FPGA module as in the prototype. This approach therefore enables to transfer the previous software investments from the preliminary work to your own individual hardware. Which saves considerable development time and costs.

3.5. For the Last Mile on the test bench

Prototypes, MVPs and series products are rigorously tested before they go into the field. Failures should come to light early according to the motto "Fail Early - Fail Cheap". The NI test platform with LabVIEW has become the quasi-standard worldwide in test and measurement. There is almost no function that is not available as a standard module. And if not, Schmid Elektronik offers a service that leads to its own front ends: for example with cRIO, FlexRIO and the SLSC.

4 The Zbrain Platform: Scalable Hardware and Software

In summary, the Zbrain is a consistent design and development platform for embedded systems. Programming is done graphically with LabVIEW on modular hardware, based on an ARM9 microcontroller and an FPGA.

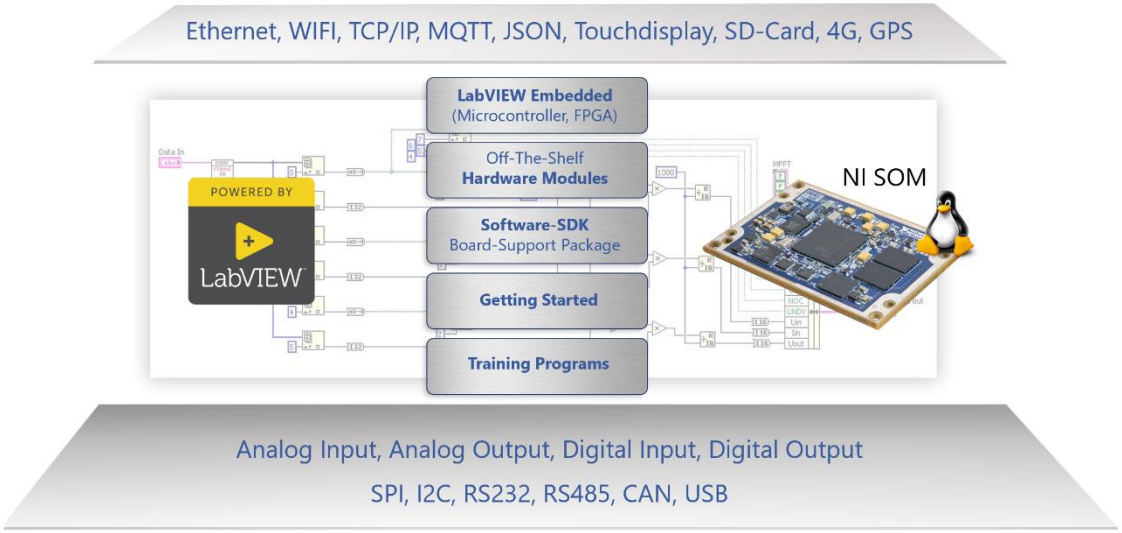


Fig.8 | The Zbrain platform essentially consists of the LabVIEW-Embedded graphical programming language, modular hardware and a scalable software development kit.

4.1. The Zbrain platform: what's in the box?

- Graphical LabVIEW development environment:
  for real-time applications on an ARM9 microcontroller and a XILINX FPGA.
  The licenses can be obtained from the NI distribution partner:
  - NI LabVIEW Realtime : Prerequisite
  - NI LabVIEW FPGA : Optional
- Standardized hardware modules:
  - ZSOM-Control, with add-ons
  - ZSOM-Mini, with add-ons
  - Customized platforms
  - All modules are available as 3D STEP models
- Zbrain SDK (Software Development Kit).
- How to get started:
  - Starter kit
  - Whitepaper, technical article
  - WIKI (online documentation)
  - Tutorials, data sheets
  - Webinars, workshops
- Training Programs:
  - Hands-on demos
  - Lectures
  - Training, also on site

4.2. The Zbrain SDK (Software Development Kit)

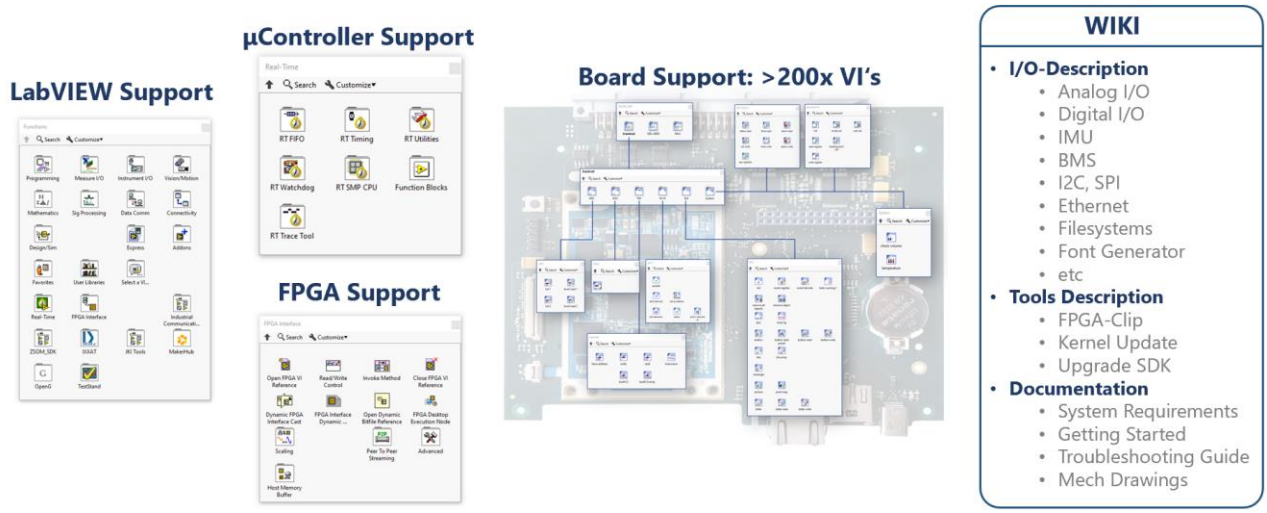
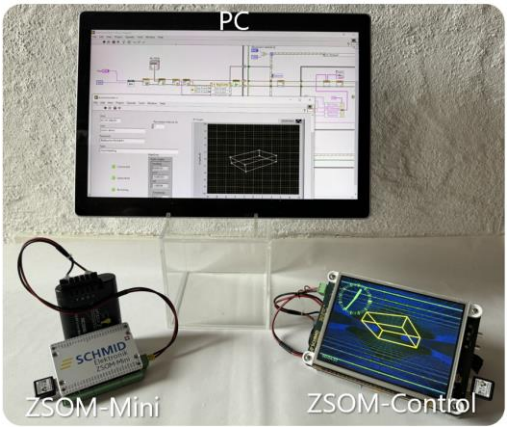


Fig.9 | The Zbrain software development kit bridges the gap between LabVIEW and standardized and individual embedded hardware.

- Basic functions of LabVIEW Embedded
  - The hardware is programmable with LabVIEW-Realtime and LabVIEW-FPGA
  - There is a system function (VI) for each low-level property
  - Fast debugging with probes and the live highlight mode - as known from LabVIEW – works also directly on the underlying embedded hardware
- Basic functions for embedded applications
  - A wide range of functions (VIs) for each hardware platform
  - A Board Support Package (BSP) with a total of over 200 VIs for microcontrollers and FPGA
  - VIs for scalable power consumption down to [mW], such as programming the RTC
  - A robust file system, display support and an IoT link, optionally GPS
  - Boot times as usual in Linux at 20-30 seconds
- Advanced functions
  - Preemptive, graphical multitasking
  - Deterministic real-time services in [µs] on ARM9
  - FPGA-VI's with real-time services in [ns]
  - Scalable power consumption and battery support
  - Robust 24/7 continuous operation, e.g. with watchdog support
  - Integration of external electronic modules via I2C/TWI, SPI, RS232 or RS485
  - Error records, either on the local file system or to an external server via TCP/IP
- Support with design & architecture
  - More than 20 program examples
  - An IoT software framework with a ZSOM-Mini, a ZSOM-Control and a desktop application, based on MQTT and JSON.
  - Team programming thanks to 100% compatibility of the supported hardware modules
  - Ready-made software architectures: start and understand in just a few minutes

4.3. A framework with IoT node, smart display and PC application

Create your own IoT application based on this LabVIEW framework! It contains a ZSOM-Mini as a stand-alone data node, a ZSOM-Control as a programmable display unit and a PC application accessing the data. In this framework, we measure the roll, pitch and yaw angle of the built-in 9DOF inertial sensor with the ZSOM-Mini.



A JSON string (JavaScript Object Notation) is generated and sent to an IoT server via WIFI using MQTT (Message Queuing Telemetry Transport). The ZSOM-Control also connects to this server, reads this data and shows the orientation of the ZSOM-Mini on the display in real time. A LabVIEW desktop application also reads the data from the IoT and logs it. The angle data can be supplemented with a time stamp and other measured data points.

Fig.10 | IoT applications from the modular system with sensor nodes (left), display (right) and evaluation software on the PC (center). All implemented with a single language - LabVIEW.

4.4. Modular Embedded Hardware - the best of both worlds

The core of the Zbrain platform is the combination of the best of two worlds - namely the standard and the individual. The standard computer is a System-On-Module (SOM) from NI in the form of a plug-in board in business card format. It has been available again in the long term since 2023. Its CPU consists of a dual ARM9 computing core, operated by a lean Linux distribution and extended by an FPGA:

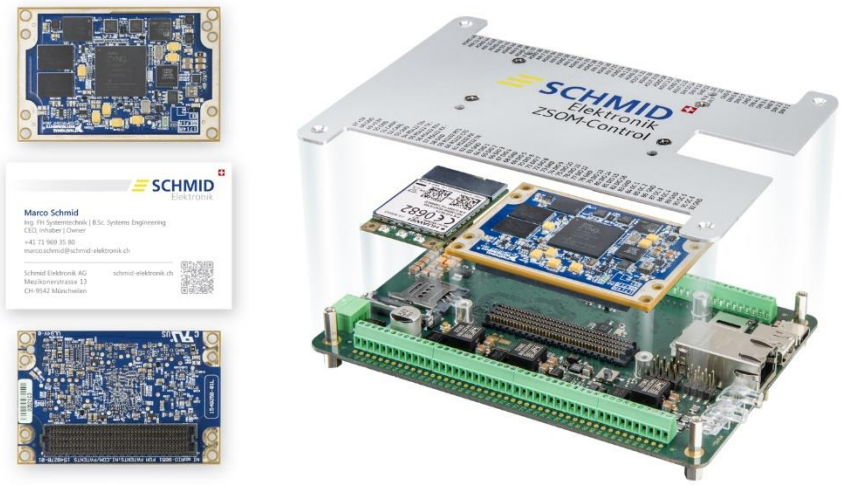


Fig.11 | The System-on-Module (SOM) from NI with microcontroller and FPGA is smaller than a business card (left) and the heart of all Zbrain hardware modules (right).

The SOM is pretty flexible when connecting customer-specific hardware modules. However, it is not functional as a single module: it is plugged into a baseboard that needs to be developed first.

Depending on the application, this baseboard integrates various functions, whether analog, digital or serial I/Os, as well as 4G, WLAN, GPS, multi-touch displays and batteries.

The advantage of this two-board approach is a lower complexity of the baseboard, as the critical high-speed circuits are already implemented on the SOM. The I/O signals usually only need to be conditioned properly.

Schmid Elektronik has developed two standard platforms on this basis: the ZSOM-Control for typical measurement and control tasks and the ZSOM-Mini for small applications.

4.5. Flexible Computer Module for measuring and control tasks: ZSOM-Control

The ZSOM-Control is a standardized, industrial single-board computer designed for harsh field use around the clock. It offers flexible analog, digital and serial process I/O and is network-compatible thanks to the optional Internet-of-Things (IoT) link. Available add-ons: Wifi, a Captouch VGA display, 4G/GPS modem and a GPS RTK module. Click here for the data sheet.

Fig.12 | The ZSOM-Control is an industrial single-board computer.
The process I/O can be connected directly to robust screw terminals.

4.6. LabVIEW compact in a cigarette pack: ZSOM-Mini

The ZSOM-Mini is a miniaturized single-board computer and also designed for harsh field use like the ZSOM-Control. It offers similar analog, digital and serial process I/O like its brother. Its strength lies primarily in the add-on modules. An IoT module, for example, gives the ZSOM-Mini WIFI, 4G and GPS. A BMS module (Battery Management System) ensures smart battery operation: charging, level indicator and manual as well as programmable startup and shutdown. This makes the ZSOM-Mini ideal for measuring handhelds or battery-powered IoT nodes. Customer-specific add-ons can be developed as required. Click here for the data sheet.

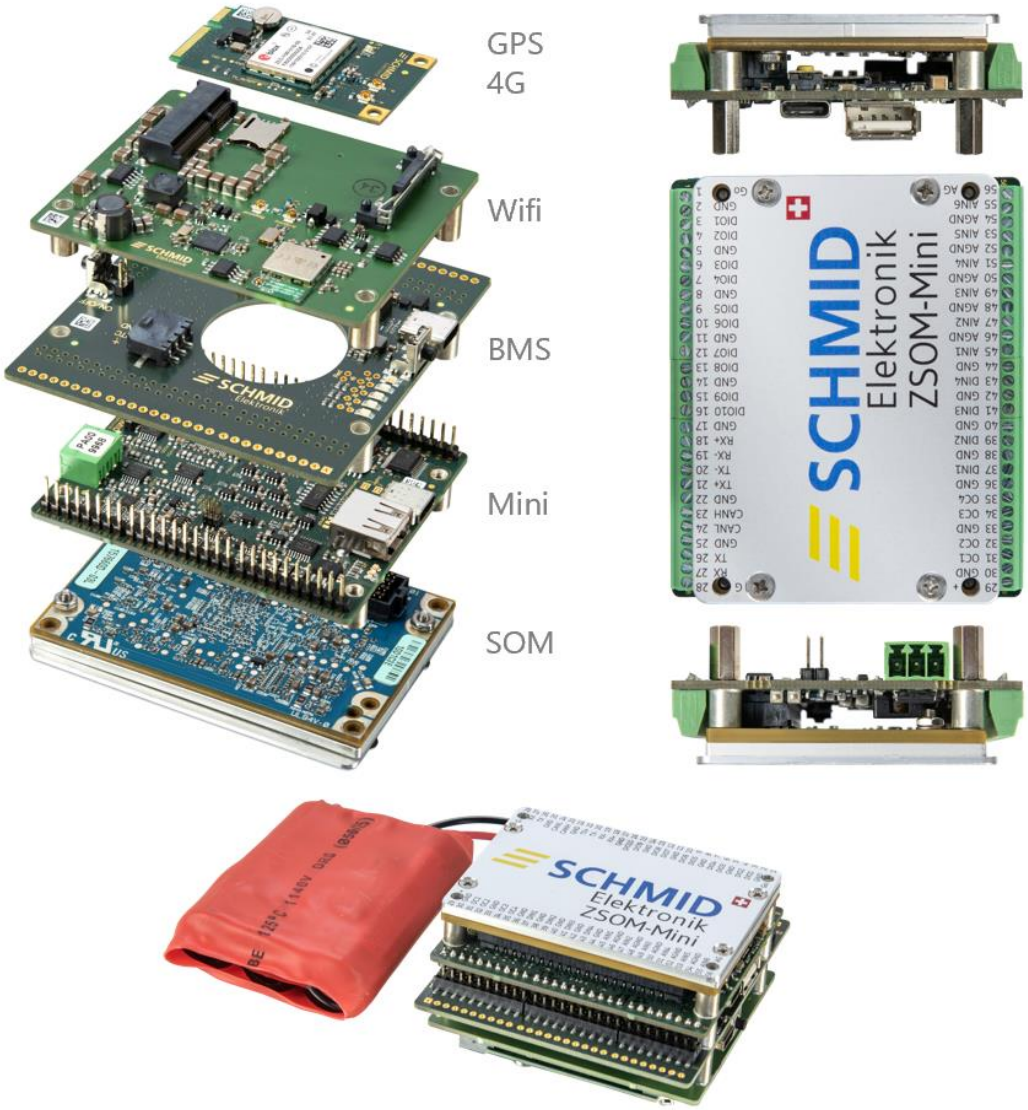


Fig.13 | The ZSOM-Mini is a robust single-board computer. Thanks to its compact form factor and modularity, it is suitable as a battery-operated, network-compatible measuring node.

4.7. Fast learning curves with the Starter Kit

The ZSOM starter kit is an ideal introduction to the Zbrain platform for anyone who wants to experience the possibilities and limits of "LabVIEW Embedded" in practice.

The motto: unpack, connect sensors and actuators, switch on and get started immediately. Set up prototypes cost-effectively in the shortest possible time, create user interfaces conveniently and connect to the Internet via WIFI or 4G. The starter kit is based on the ZSOM-Control. The LabVIEW development environment can be downloaded from ni.com.

The starter kit includes:

- **ZSOM-Control (pre-wired to match the software examples)**
- 5.7-inch VGA display
- WIFI module (USB)
- **Memory stick with SDK, application notes and software examples**
  Power supply unit

Fig.14 | Easy access to the platform with the starter kit:
unpack, connect, switch on and get started with software examples!

4.8. Customized Embedded Systems

If a LabVIEW user requires his own hardware with individual I/O for his series product, he orders a customized baseboard from Schmid Elektronik. The most recent example was a baseboard for controlling an active laser light source - with temperature compensation and low-noise supply sources in the context of fiber optic measurement technology. The whitepaper LabVIEW Embedded with Linux on ARM and FPGA shows what is possible.

5 LabVIEW Embedded Applications in Action

The possible uses of LabVIEW on embedded hardware are just as diverse as the general application possibilities of LabVIEW itself.

Fig.15 | Examples of Zbrain applications (from left to right): six-legged robot for disaster relief operations, monitoring the wear of railway infrastructure, measuring rotation in the wheel hub and controlling a compact and highly efficient solar power plant.

5.1. From stationary to mobile applications

The following examples are summarized in the whitepaper LabVIEW Embedded Applications in Action:

Field of application	Examples
Condition monitoring	• Extreme situation requires pipeline monitoring 1000m below sea level • Condition monitoring in power plants • Monitoring critical building infrastructure
Safe, Comfortable and Punctual Rail Travel	• Digital condition monitoring of railway tracks and turnouts • Predictive maintenance thanks to intelligent grinding of rail tracks
Energy Generation, Transmission and Efficiency	• Generating Generating energy with 80% efficiency from solar energy • Testing high-voltage direct current transmission (HVDC) • Increasing energy efficiency thanks to telemetry and race data
Mobility & Transportation	• Decentralized measurement technology in the wheel hub and telemetry connection • Dashboard MVP for racing drivers in motorsport • Automating racing processes thanks to intelligent vehicles • Only with the power of the sun through the Australian outback
Mobile Battery-Operated Handhelds and Portables	• Taking the elevator experience to a new level • Mobile LIDAR-Measurement Device for an Outdoor Application • A smart Laser Module that Meets the Eye • Sleep monitoring "on humans" in the medical field
Research Projects	• Autonomous spider robots for disaster relief operations • Research acceleration for two-liter satellite

5.2. 1000m below sea level –Launch of the Zbrain

The whitepaper 1000m below Sea Level - Launch of the Zbrain takes a deep dive into the first large scale project that was a tough test for applying LabVIEW in the embedded world. It was about a complex pipeline monitoring network in the Norwegian subsea that had to be developed in the record time of seven months. For a global energy corporate.

On the application side, it involved a level of complexity that Schmid Elektronik encountered again around ten years later in a telemetry system for the same energy giant.

In addition to a sense of adventure and the willingness to accept risks, the ambitious pipeline project required a new way of thinking in product development in order to master the complexity and tight schedule. The deadline could not be met using traditional development methods, such as C/C++ on microcontrollers. A method was required that abstracted the complexity of the application and the hardware: NI LabVIEW Embedded..

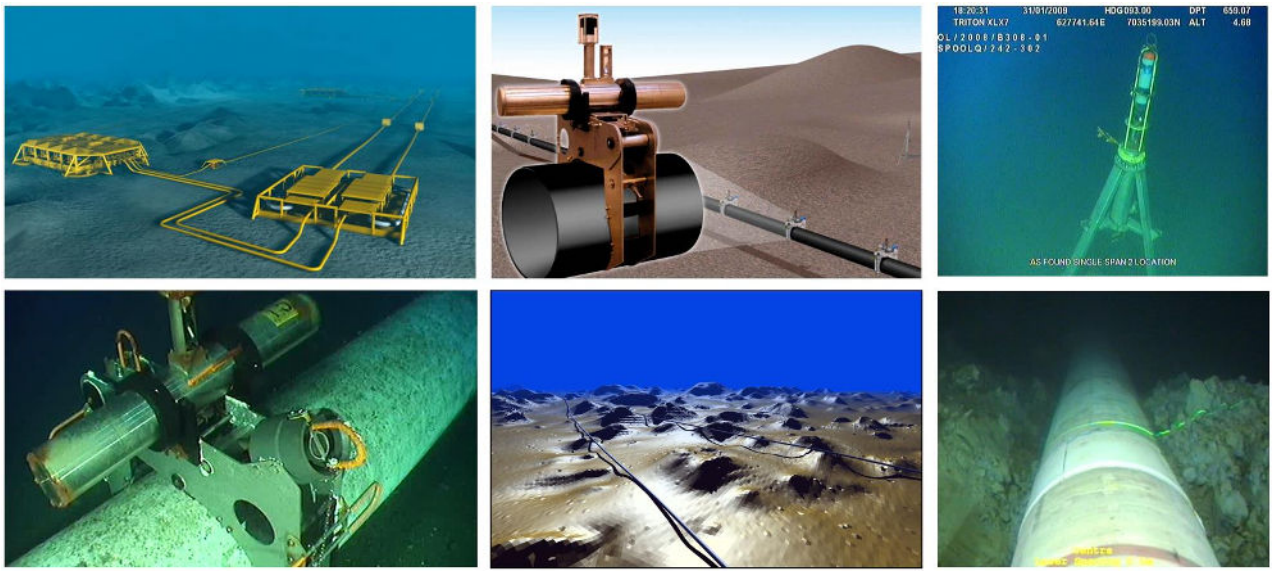


Figure 16 | Measuring network with acoustic synchronization of the nodes in milliseconds, installed in the Norwegian deep sea with 6 months of autonomous operation.

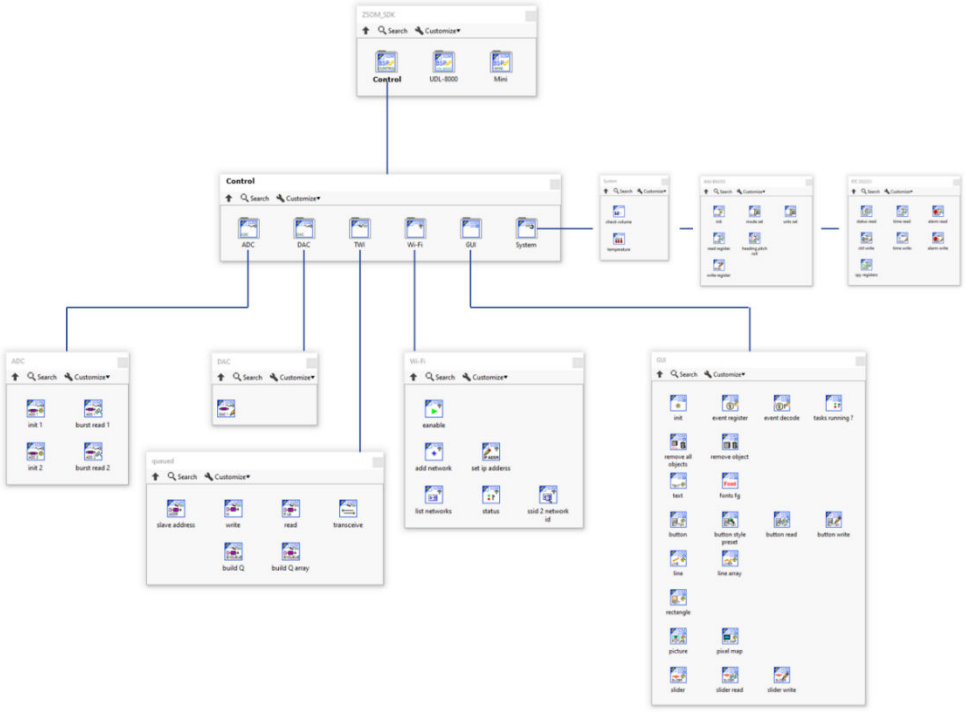
In any case, after successful completion, this led to the birth of the Zbrain platform described in this white paper, a globally marketed product from Schmid Elektronik. It brought LabVIEW into the embedded world and has enabled graphical programming of microcontrollers and FPGA ever since. This success paved the way for Schmid Elektronik's partnership with NI in 2007.

6 Appendix: Datasheet of the ZBrain Hardware Modules

	ZSOM-Control	ZSOM-Mini
CPU	ARM Cortex-A9 multicore	ARM Cortex-A9 multicore
FPGA	XILINX ZYNQ-7020	XILINX ZYNQ-7020
Internal memory	512MB DRAM, 512MB storage	512MB DRAM, 512MB storage
External memory	1x SD card	1x SD card
Operating System	Linux (Ångström)	Linux (Ångström)
Ethernet	1x GigE	Ethernet over USB
WIFI	Optional	Optional
GPS	Optional	Optional
4G	Optional	Optional
CAN	1x CAN/Open	1x CAN (Basic)
RS232	1x with handshake lines	1x
RS485	1x RS422/RS485	1x RS422/RS485s
USB	1x USB Host/Type A , 1x USB Device Type C	1x USB Host/Type A , 1x USB Device Type C
I2C	Yes	Yes
SPI	Yes	Yes
Display	5.7" multi-touch VGA/RGB (optional)	-
Analog IN	12x analog in, 16 bit, ±5V or ±10V, 450kHz simultaneous, 4th order anti aliasing with fg=200kHz	6x analog in, 16 bit, ±5V or ±10V, 450kHz simultaneous, 4th order anti aliasing with fg=200kHz.
Analog Out	4x analog out, 16 bit, ±10V, OVP, 100kHz simultaneous generation of all 4 channels.	-
Digital I/O	16x general purpose high speed I/O, configurable as digital input or output, 3.3V or 5V operation, speed in the MHz range allows to integrate high performance SPI devices.	10x general purpose high speed I/O, configurable as digital input or output, 3.3V or 5V operation, speed in the MHz range allows to integrate high performance SPI devices.
Digital IN	10x rugged digital inputs, 3-30V, OVP (DINX)	10x rugged digital inputs, 3-30V, OVP (DINX)
Digital Out	6x rugged open collector outputs, max current: 200mA	4x rugged open collector outputs, max current: 200mA.
Counter/PWM	16x (GPIO)	10x (GPIO)
Encoder	16x (GPIO)	10x (GPIO)
IMU	9x axes IMU (accelerometer, gyro, magnetometer)	9x axes IMU (accelerometer, gyro, magnetometer)
RTC	Internal (SOM), backup battery	External, precise and programmable
Watchdog	Yes	Yes
Power	wide input range 9-30V	wide input range 9-30V
battery support	Through main power supply only	BMS-Addon
Connectors	92x screw terminals	28x screw terminals
Extension	1x miniPCIe	10 pin extension board connector
Miscellaneous	1x 3.3V and 1x5V power supply for external devices, 1x reset button, 4x status LEDs	1x reset button, 4x status LEDs 1x boot on input, 1x kill signal (FPGA)
Format	b/l/h: 100 x 146 x 10 mm (with step model)	b/l/h: 64 x 76 x 22mm (with step model)

7 Appendix: Zbrain Board Support Package Architecture

Overview of a Board Support Package (BSP) architecture using the example of ZSOM Control:



8 Literature

[1] Marco Schmid : Cyber Physical Systems ganz Konkret, ELEKTRONIKPRAXIS, 2015

[2] Whitepaper : The Charm and Benefit of Graphical Programming with LabVIEW, Schmid Elektronik, 2024

[3] Whitepaper: LabVIEW Embedded with Linux on ARM and FPGA, Schmid Elektronik, 2024

[4] Whitepaper: LabVIEW 1000m below Sea Level, Schmid Elektronik, 2024

[5] Whitepaper: LabVIEW Embedded Applications in Action, Schmid Elektronik, 2024

[6] Marco Schmid : Minimum Viable Products (MVP) - an idea in a reality check, Whitepaper, 2021

[7] Schmid Elektronik : Online WIKI ZBrain Platform

[8] National Instruments : Single-Board RIO Device Reference and Procedures, NI-Website

[9] Marco Schmid : 100km with only five teaspoons of fuel, technical article, 2024, german

[10] Marco Schmid : In just eight Weeks to a Race-IoT-MVP, technical article, 2024, german

9 Glossary

Chapter 1	LabVIEW	Stands for graphical programming with the data flow concept
	Microcontroller	The microcontroller on the SOM is an ARM Cortex-A9 (ZYNQ)
	FPGA	Reconfigurable logic: **F**ield **P**rogrammable **G**ate **A**rray
	C/C++	C is the most common procedural language for microcontrollers. C++ supports object-oriented software design
	Embedded	Combination of hardware and software for embedded systems
	Low code	Visual, simple programming
	No-Code	Programming via configuration of graphical blocks
	MVPs	Minimum Viable Products: they have only the most essential functions
	WIKI	Here it is a web-based manual for the Zbrain platform
Chapter 2	Language "G"	A visual language of functional blocks that are connected with wires
	Low-level driver	Hardware-related control of electronic components, e.g. A/D converters
	Timing	Correct, predictable temporal behavior of embedded software
	Operating system	A system of programs for the operation of a microcontroller
	Multitasking	Execution of multiple, independent tasks simultaneously
	Multicore	A microcontroller with two computing cores like an ARM Cortex A9
	Real time	Specified time that certain processes may consume
	Tools	A collection of development tools for embedded systems
	Microkernel	A kind of "mini operating system" that only includes basic functions
	RTOS	A lean real-time operating system for microcontrollers with a scheduler
	Linux	Open source operating system with kernel, tools, apps and services
	Booting	Starting up an operating system until the application can start
	24/7 operation	Highly available, reliable continuous operation around the clock
	Software Engineering	Processes and technology for the development of software
	Hardware abstraction	Isolates the software from the hardware and serves as an "intermediary"
	Software Architectures	Modularization, frameworks, object orientation, design patterns
	Dynamic memory	Program fetches contiguous memory areas at runtime
	Memoryleaks	Leak in the working memor, decreasing it more and more, at some point the application runs out of memory and the software crashes.
	Error handling	Recognize runtime errors and react to them in a controlled manner: e.g. monitor memory leaks and configure watchdog.

Chapter 3	Proof of Concept	Basis for deciding whether a project can be implemented
	Prototypes	Tangible representation of a design concept, close to the series product
	IP	Intellectual property
	Matlab	Commercial software, optimized for mathematical processing
	Python	Universal, higher programming language, often runs interpreted.
	Statecharts	Graphical, simple representation of state transition diagrams
	Linear algebra	Subfield of mathematics with vector and matrix calculations
	Signal processing	Extract information from a received signal
	Filter	Signals: Blocking or passing certain frequency ranges
	Sound & Vibration	Analysis functions for audio, acoustic and vibration signals
	Ethernet	Wired data transmission within a network (LAN)
	WIFI	Wireless Fidelity: Wireless network technology
	4G	Fourth generation of mobile network technology, LTE connection
	Synchronization	Change data and processes at different locations simultaneously
	Circuits	Graphic interconnection of electronic components
	LabVIEW VI	VI = Virtual Instrument = Function block in LabVIEW
	cRIO	Compact RIO: a family of controllers with flexible I/O modules
	FlexRIO	Standard solution for high-performance I/O, e.g. for PXI platforms
	SLSC	Switch-Load-Signal-Conditioning: Signal conditioning cards for HIL
Chapter 4	XILINX	Manufacturer of programmable logic. Acquired by AMD in 2022.
	STEP model	Standard for the exchange of mechanical model data
	SDK	Collection of tools and libraries for software development
	BSP	Board Support Package: this refers to drivers for the hardware
	Debugging	Search for and rectify errors in the software
	Probes	Data lines used to examine code on microcontrollers
	Highlight Mode	Animates a LabVIEW block diagram. You can see the program execution and how the data flows.
	RTC	Real-time clock, usually supported by a button cell
	Preemptive	Used here for multitasking: the operating system has full control and allocates individual time slots to the processes
	Watchdog	Detects the failure of a system and can perform a reset
	I2C/TWI	TWI (Two-Wire Interface), originally I2C: Two-wire bus for I/O
	SPI	Serial synchronous and very fast bus for I/O modules.

	RS232	A common standard for serial, asynchronous data transmission
	RS485/422	Robust industry standard for asynchronous data transmission
	TCP/IP	A group of protocols for communication on the Internet
	MQTT	Open network protocol for IoT devices and systems
	JSON	Human- and machine-readable data format for web systems
	9DOF-IMU	Acceleration, rotation angle and magnetic field detection in a single chip. IMU: inertial measurement unit.
	IoT	Internet of Things: a network of physical objects, the "Things"
	SOM	System on Module: a computer in credit card format
	WLAN	Local radio network
	GPS	Global Positioning System for determining position
	GPS RTK	GPS with Real Time Kinematics: precise determination of position
	Captouch VGA	Capacitive display, 5.7" format and a resolution of 640 x 480 pixels
	BMS	Battery management system: monitoring, control and protection of rechargeable batteries
Chapter 5	Telemetry	Wireless transmission of measured values to another location

10 About the Autor

Marco Schmid,
marco.schmid@schmid-elektronik.ch, Tel: +41 71 969 35 90,
Engineer FH Systems Engineering, B.Sc. / Entrepreneur
The systems engineer in me has a passion for embedded systems, the graphical programming language LabVIEW, IoT things, minimum viable products (MVPs), complexity, networks, knowledge graphs & language models as well as information and data science.

As an entrepreneur, I enjoy the privilege of coaching the leadership team of a 50+ year old family business with cool DNA and 40 smart people.

At home, I like to cook Indian, Chinese, tapas and sushi. I like traveling to faraway countries and learning from other people's views. Camping in a tent connects me with nature and I feel very comfortable in the open air. I can also be found in a mountain hut from time to time. I switch off from business and the digital world and enjoy the simplicity.

11 Who is NI Partner Schmid Elektronik?

schmid-elektronik.ch: Schmid Elektronik AG is a global, independent Swiss solution provider for industrial electronics and embedded systems. Its specialty lies in three core competencies united under one roof: Engineering Services, Production Services (EMS) and customized NI LabVIEW platforms. Thanks to this one-stop shop, Schmid supports its customers from design and proof of concept through minimum viable products (MVP) and prototyping to series production and test. They are supplied from a single source with:

Engineering
- Embedded hardware design
- Embedded software design
- System integration

Production/EMS
- Prototypes
- Pilot series
- Series products

Products
- Standard LabVIEW Embedded Hardware
- Customized NI-CompactRIO modules
- Customized NI FlexRIO/SLSC cards

Industries: Clean energy, efficient mobility, measurement technology, electrical engineering, mechanical engineering, equipment manufacturing, automation, research.

Facts & figures: Founded in 1972, CHF 6-7 million turnover, 40 employees, ISO9001:2015.
The family-owned company is a system integration partner of NI (formerly National Instruments, now part of EMERSON) and technology partner of Shell Eco-marathon, a community for clean mobility and environmentally friendly energy consumption. Schmid is the owner of the "Zbrain" brand. These are graphically programmable embedded systems with LabVIEW.

Partnerships:

Schmid Elektronik has been an NI partner since 2007, specializing in LabVIEW on embedded hardware. What began with a C generator was later replaced with the NI System-On-Module (SOM), the sbRIO9651.

Schmid Elektronik is an official partner of Shell Eco-marathon and part of the Data & Technology Team. Since 2015, Schmid Elektronik has been offering development and production services as well as products and on-site services for the telemetry system. The services include: Webinars, Bootcamps, on-site Telemetry Briefings,Technical Inspection and the Telemetry Off-Track Award Jury.
The Swiss family-owned company joins the ranks of global players who are also partners of the program

The unusual rollercoaster and mountaineering story of Schmid Elektronik is told here.

Münchwilen, Switzerland, June 2024, Marco Schmid, marco.schmid@schmid-elektronik.ch