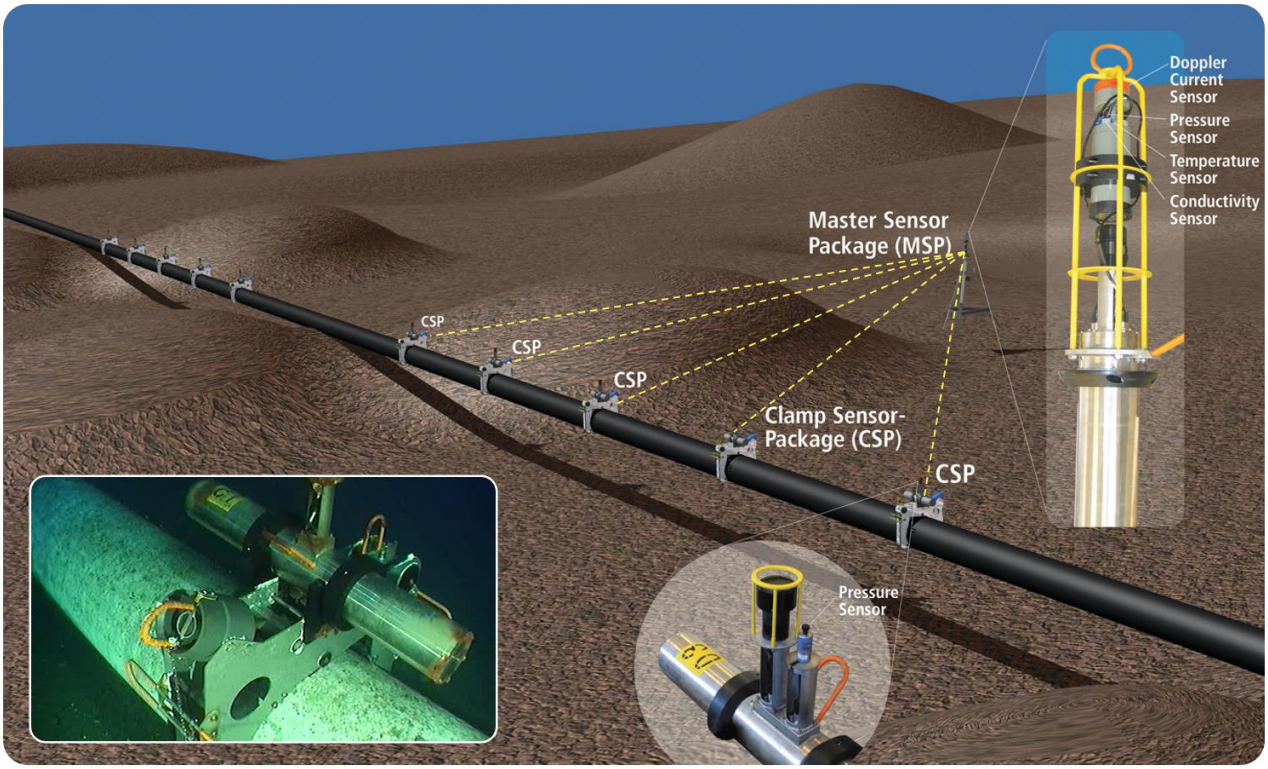
Whitepaper

LabVIEW 1000m Below the Waves



A battery-powered measurement network, consisting of a master package (MSP) and several sensor clamps (CSP), autonomously monitors gas pipeline vibrations in the Norwegian deep sea for months at a time (Image: NAXYS).

Gas production in the Ormen Lange in Norway's largest industrial project to date required permanent condition monitoring due to extreme deep-sea conditions. The solution presented here - an autonomous pipeline monitoring system - consists of several synchronized nodes that analyze vibrations and water parameters and communicate wirelessly via acoustic modems. It has been realised with LabVIEW Embedded.

Content

The measurement network can operate under harsh conditions for long periods of time and demands the highest levels of hardware and software reliability, smart error handling and efficient power management. LabVIEW Embedded, together with standard microcontroller-based hardware from Schmid Elektronik, provided the stability, versatility, performance and battery life needed to meet development time and quality requirements.

This technical solution of high-precision, synchro-nized data acquisition in the deep sea is a world first. This also applies to the use of graphical programming for an application of this size, which is distributed on a low-power target system.

1 Extreme Situation requires Condition Monitoring

It is the year 2006 and we are in Ormen Lange on the Scandinavian west coast. This is where the world's largest natural gas reservoir is currently located. In Norway's most expensive industrial project to that date, a total of 1200 km of pipelines are to be laid in the sea between Nyhamna and England that will eventually cover up to 20% of the British gas consumption for the next 40 years (Fig. 1). The future operator is the same energy company that will be asking for a telemetry application 10 years later for Shell Eco-marathon...

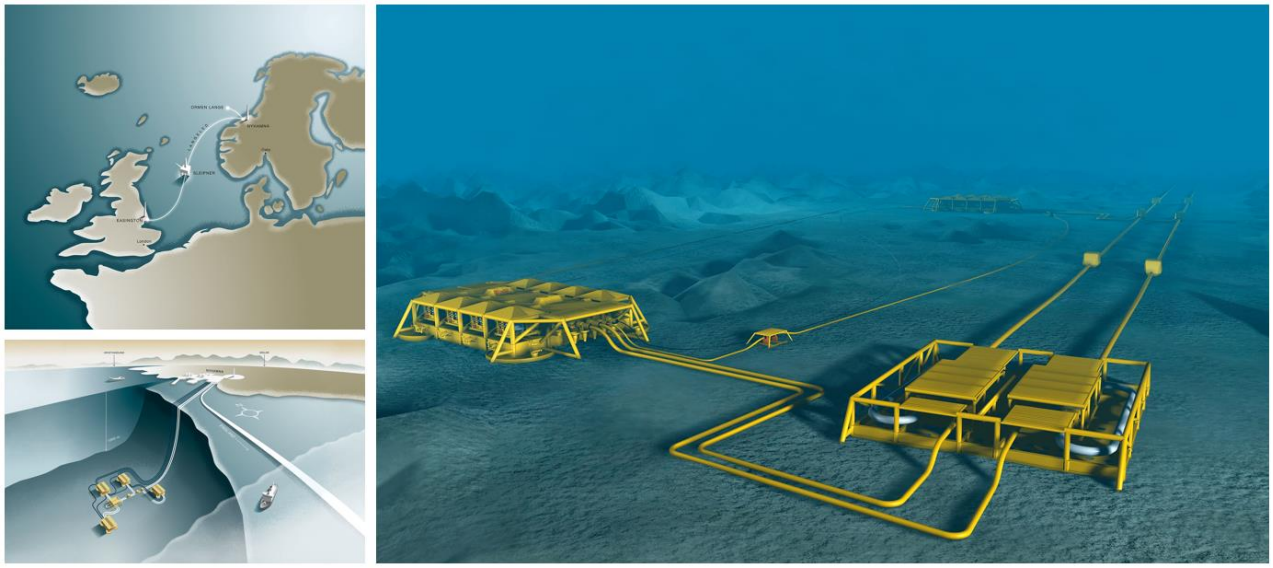


Fig. 1 | The development of the Ormen Lange gas field is the largest industrial project in Norway's history to date. The field is located off the west coast of Norway in the deep sea and consists of 24 subsea wells spread over four subsea platforms.

In the critical section between Ormen Lange and Nyhamna, the pipelines will run over the "Storegga", an 800 km long, uneven terrain (Fig.2). In some places, the pipes are not even in contact with the seabed. Vibrations are expected in these free-floating areas as a result of strong ocean currents, which could lead to damage.

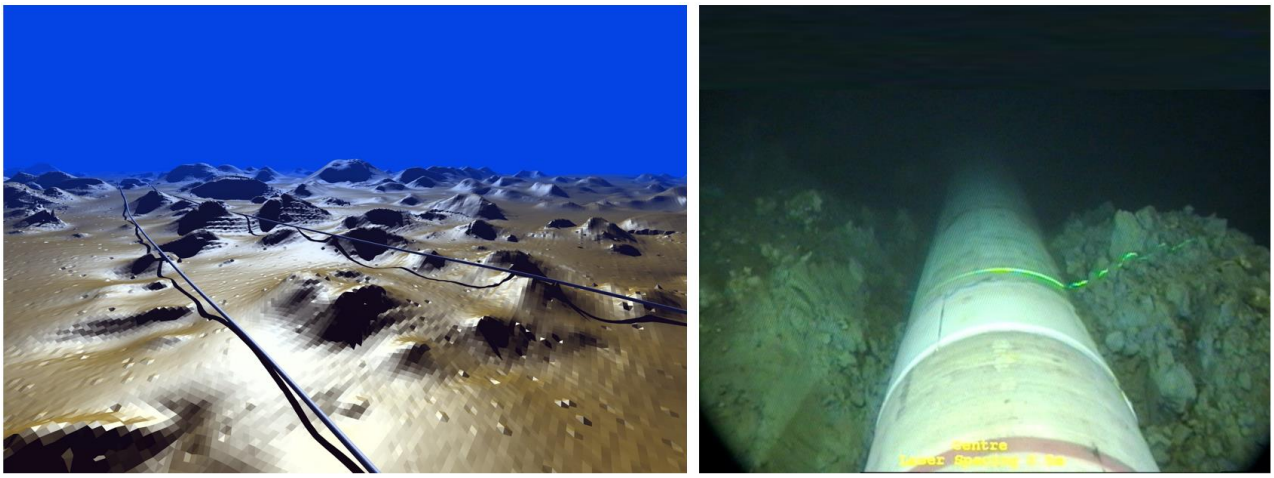


Fig. 2 | Laying the pipelines themselves is one of the most challenging projects ever undertaken in the field of pipeline laying; this is mainly due to the difficult difficult terrain and the strong ocean currents.

The company Naxys Technologies from Bergen, Norway, specializes in deep-sea technology and is entrusted with a measuring network for permanent monitoring. The delivery date is fixed and must be met under all circumstances. Naxys contacts the Swiss family-owned SME Schmid Elektronik. As an early adopter, Schmid is currently testing the feasibility of NI's very first "LabVIEW Embedded" version for its laser-based monitoring systems for railway tracks and turnouts with the aim of preventive maintenance.

Fig. 3 | A Remote Operating Vehicle (ROV, left) anchors the CSP (Clamp Sensor Package, centre) on the pipeline and the MSP (Master Sensor Package, right) on the seabed.

The requirements for the pipeline monitoring system are extremely high. It starts with a full-blown application logic: 10x parallel processes, 5x channels for inter-process communication, 200x subroutines, some of which are considerable, and complex mathematical evaluation algorithms that have to be executed in real time. In addition, there is precise, time-synchronized 3D vibration measurement, a 6-month battery life and a hundert percent autonomy in the subsea.

In addition to a willingness to take risks and an engineering spirit, the ambitious project requires a new way of thinking in product development: abstracting complexity with LabVIEW! This is the only way the project manager sees a chance of coping with the high level of complexity and the tight schedule.

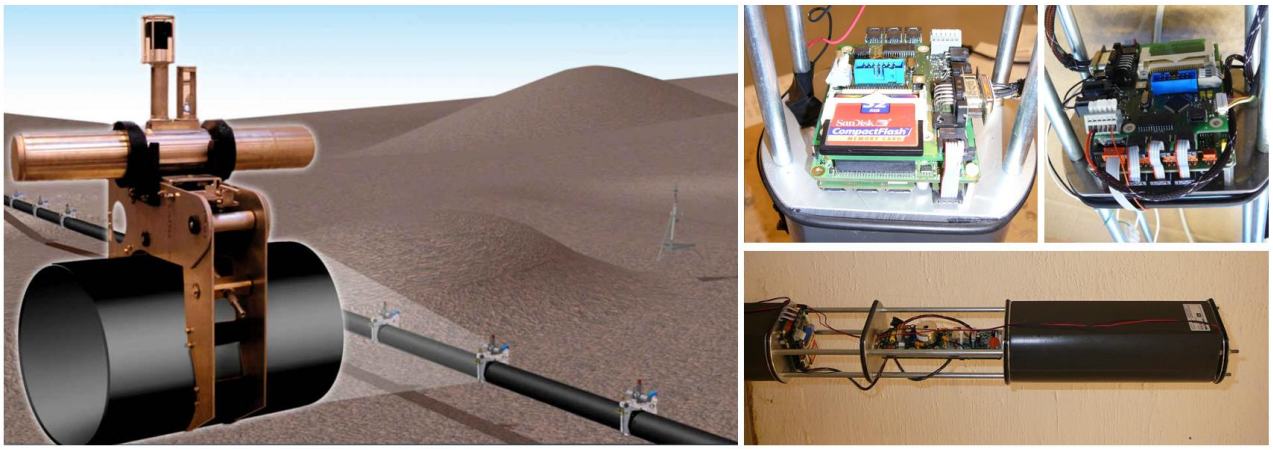


Fig. 4 | A leak-proof, stainless steel housing contains the hardware, the battery packs, the modem and the antennas.

2 Time-Synchronous Vibration Measurement

The monitoring system consists of a master (MSP: Master Sensor Package) anchored on the seabed near the pipeline and several nodes (CSP: Clamp Sensor Package) in the form of measuring clamps mounted directly on the pipe (Fig. 1). The master synchronizes the measurement of the nodes by communicating with them via acoustic modems through sound waves.

This wireless connection has impressive advantages over cabling: Installation by an underwater robot is simpler, collisions between the robot and the cables can be ruled out and, last but not least, cable breaks due to the expected pipeline vibrations are avoided.

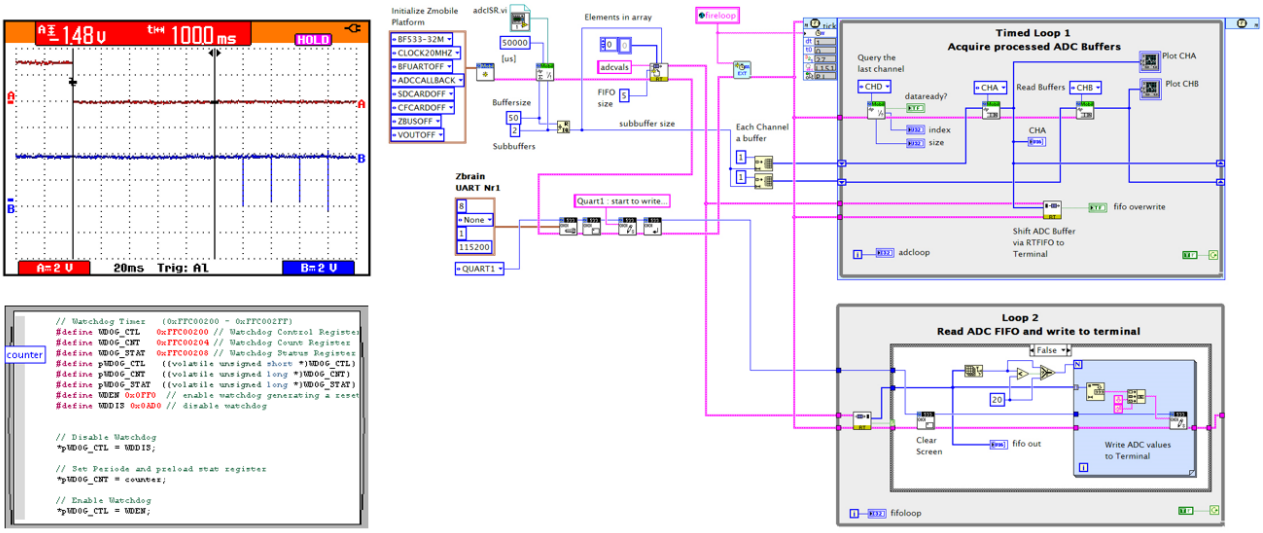


Fig. 5 | Example of event-controlled, buffered multiple analogue data acquisition with the required real-time accuracy in the microsecond range and 24/7-Operation.

Precise measurement of movements and vibration modes requires that the logging process within all nodes starts at exactly the same time. With a distance of 100m between the nodes, the total deviation must not exceed 2.5 ms, which requires timing in the deterministic two-digit microsecond range within the node. This main requirement is met with the following solution approach: the master initiates the measurement process by acoustically measuring the distance to each measurement node. Environmental parameters such as temperature, flow and salinity distort the sound propagation, which is why they are measured and compensated for.

A second acoustic signal triggers the synchronized measurement. In addition to the trigger, the acoustic data packet contains the respective node addresses and the global logging start time. Each node now measures the internal software runtime that elapses between the trigger interrupt and the start of the measurement function with microsecond accuracy (Fig. 5). Thanks to this information, the node knows exactly when data logging must start and configures the start time of the analog measurement with a precise timer. Registering the trigger and setting the timer "tunnels" LabVIEW in "C" directly at the interrupt level.

3 Half a year on the battery

A second tough criterion is the duration of the measurement campaigns of six months. This calls for a system with a power consumption in the double-digit milliwatt range, because the available storage space in the pressure housing severely limits the battery capacity.

The trick is to combine the powerful main CPU with an ultra-low-power microcontroller, an MSP430. This economical "little one" monitors the vibration sensors with smart triggers and wakes up the powerful "big one" according to a programmable schedule or when required (Fig. 6).

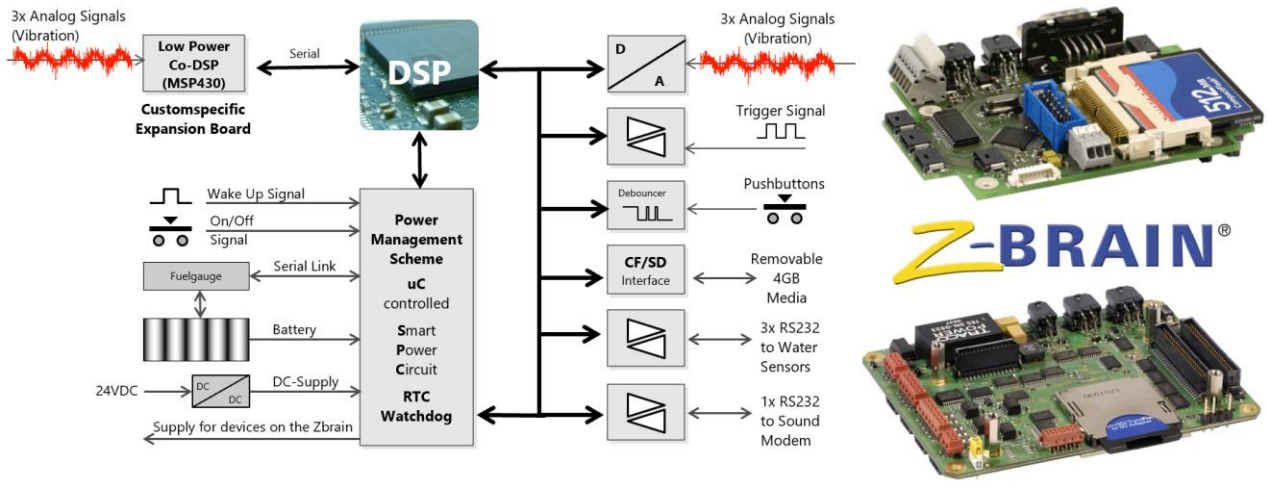


Fig. 6 | A customised low-power embedded system with a main DSP and an MSP430 as a low-power coprocessor was developed for this application. Following the successful completion of the project, this resulted in the Zbrain platform, which can be programmed graphically using LabVIEW Embedded, with a modular system consisting of hardware and software.

The LabVIEW application continuously measures the power consumption and the remaining battery capacity on the main digital signal processor (DSP). By switching off electronic components that are temporarily not required and scaling the CPU clock in a targeted manner, power consumption is reduced to a minimum during runtime.

As soon as the CPU is required again, e.g. when calculating a FFT, the clock is switched up again. For this reason, graphical abstraction was partially dispensed with for some algorithms and the much faster implementation in fixed-point arithmetic or even embedded "C" was preferred with a low, energy-saving processor clock (Fig. 7).

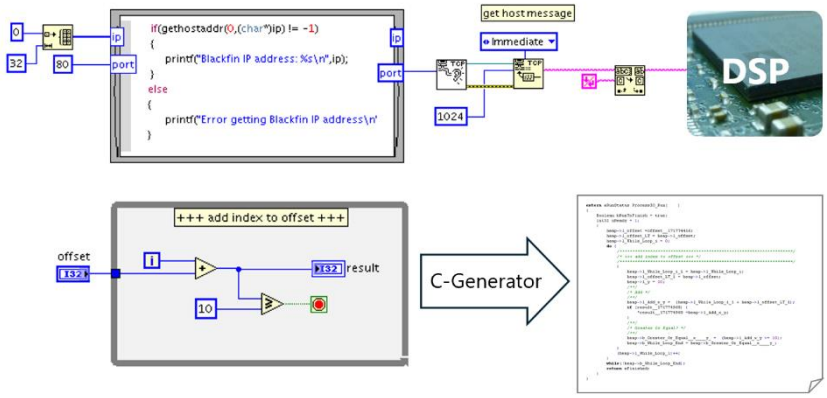


Fig. 7 | In LabVIEW Embedded, the abstract graphical data flow complements the powerful "C" programming language. These two were not only a successful pair back then with the C generator, but are still a successful pair today under Linux.

To minimize power consumption, the hardware switches off completely and can be "woken up" again at intervals of between 30 minutes and 3 hours via a precision RTC. In this operating mode, the small microcontroller measures the pipeline oscillations and wakes up the main CPU when the limit values are exceeded, which is quickly available thanks to fast boot times. Thanks to this energy scheme, all measuring nodes survived the six months.

4 Six-Months Continuous Operation

The third important design factor is complete autonomy during the 6-month measurement campaign. At 1000 meters below sea level, there is no possibility of somehow rescuing the system "from the outside". Consistent error handling therefore ensures continuous industrial 24/7 operation. All possible errors are isolated and registered at several levels, from low-level drivers to problems in the application logic (Fig. 8).

Error correction consists of several escalation stages, which means that each node has the means to regenerate itself. All events and errors are continuously communicated in the network and processed globally. If the worst-case scenario occurs and the master node fails, any node can take over the master function if necessary. The key to this is two different configuration files on the internal memory.

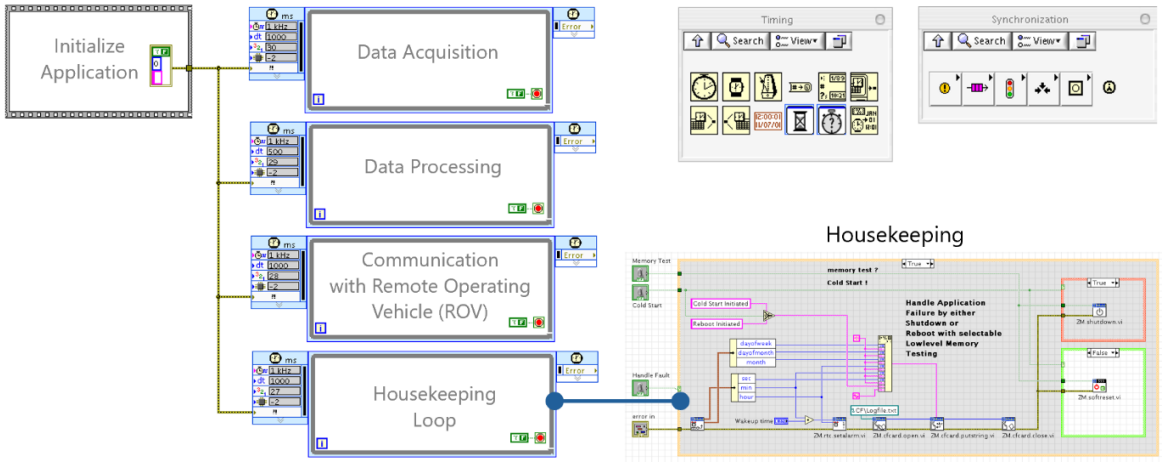


Fig. 8 | Robust multitasking operation scheme of up to ten parallel loops with data acquisition, external communication as well as error detection and correction for 24/7 availability.

5 A Question of Quality and Reliability

The considerable complexity and quality that had to be implemented within the tight schedule of just seven months is one of the main reasons why the LabVIEW graphical programming environment was chosen for this task.

It creates the prerequisites for a consistent and team-oriented development environment with the required range of functions, abstraction of low-level complexity, integral type checking, continuous range checking, integrated error handling, multitasking, real-time operation and, finally, support for a mix of languages.

All of this relieves Naxys developers of time-consuming details so that they can concentrate on functionality right from the start. The challenges of functional safety and reliability are met with the following seven best practices:

1. ISO9001-certified systematic development process of all companies involved in the project as a prerequisite for quality assurance, document control and traceability.
2. A scalable, transparent and object-based software architecture that also enables code reviews with independent bodies.
3. A resource-saving programming style that is easy on the CPU and memory.
4. Use of independent memory areas for firmware, configuration parameters and measurement data.
5. Low-level error handling, CPU self-tests, ROM test and RAM test and localization of memory leaks.
6. Error correction through targeted resets and warm starts, and
7. Interception of extreme situations through cold starts via a watchdog.

In dynamic long-term tests, worst-case scenarios are run through for days, first in the private bathtub, then in “official” tanks and finally on the coast. I/O, processor load, measurement deterministics, execution times, jitter, timing, multitasking reliability, memory management and race conditions are put through their paces. Only then do we dive into the sea...

6 Proven over Several Years

The complexity, the high development costs and, above all, the enormous deadline pressure posed new challenges for everyone involved in the project. The technology mix with LabVIEW Embedded and microcontroller hardware proved to be a tightrope walk in retrospect, e.g. when the abstraction comfort popular in LabVIEW had to be partially sacrificed in favor of increased performance. Nevertheless, graphical application programming at system level was the right choice, as the traditional, text-based programming approach would have extended application development by a factor of four and the project would have failed just because of that.

This meant that installation and gas production was able to start on schedule at full load in 2007 with the confidence that a monitoring system would report critical pipeline conditions immediately..

7 Glossary

Intro Chapter 1	Modem	A module that exchanges information (data) between the device and the outside world.
	Early Adopter	A pioneer with an early willingness to embrace a technological innovation.
	Interprocess-Communication	When several software processes exchange data with each other, for example via a queue.
Chapter 2	MSP	Master Sensor Package
	CSP	Clamp Sensor Package
	MSP430	Low-Power 16-bit Microcontroller.
	Deterministic	The events are clearly predetermined.
	Software runtime	Here: the time that elapses between the occurrence of a trigger and the beginning of a measurement data acquisition.
	Trigger	Automatically do something when a certain event occurs.
	Interrupt	React immediately and asynchronously to internal or external events. There is a separate software level in the microcontroller for this purpose.
	Timer	Here: an electronic timer in the CPU.
Chapter 3	DSP	Digital Signal Prozessor
	CPU-Clock	The clock frequency (number of cycles per second) of a microcontroller.
	FFT	Fast Fourier Transformation: converts signals from the time domain to the frequency domain.
	C	A general programming language.
	Fixed-Point	Fixed-point number: consists of a fixed number of digits before and after the decimal point. The position of the decimal point is fixed. Advantage: can be calculated much faster than floating point.

	RTC	**R**eal-**T**ime-**C**lock
	Booting	The start-up of an operating system (bootloader) until the application can start.
Chapter 4	24/7	Highly available, reliable continuous operation around the clock.
	Low-Level-Driver	Hardware-related control of electronic components, e.g. AD converters.
	Application logic	The process of a software application.
	Errorhandling	Intercept errors in a targeted manner and react to them in a controlled manner.
Chapter 5	Type checking	Check the data type used in the software.
	Range checking	Check whether a variable is in the valid range.
	Multitasking	Execution of multiple, independent tasks simultaneously.
	Real-time	Specified time that certain processes may consume.
	Mix of languages	If several programming languages can be combined in the same code.
	Best Practices	Follow a practical approach that is based on experience.
	ISO9001	A globally popular management system.
	Architecture	Modularisation, frameworks, object orientation, design patterns.
	Code-Reviews	A means of improving the quality of the code.
	Firmware	Software that issues machine commands to the hardware components.
	ROM-Test	Testing the program memory.
	RAM-Test	Testing the data memory.
	Memoryleaks	Leak in the working memory. This decreases more and more, at some point the application runs out of memory and the software crashes.
	Warm start, reboot	Restarting an embedded system when it is already switched on and "warm". For example, a planned reset.
	Cold start	Restart of an embedded system in which the entire system, including the operating system, is completely reset.
	Watchdog	Function that recognises a failure and can react accordingly. For example, if the software hangs.
	Jitter	Differences in the runtime of individual data or signals. This has a massive impact on real-time operation and must be taken into account.
	Race Condition	When two processes access a resource at the same time and an unstable or undefined status occurs.
Chapter 6	C-Generator	In this context, C code is generated from the graphical LabVIEW code, which is then converted into firmware using standard tools. EOL since 2017.

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