

Whitepaper

LabVIEW Embedded with Linux on ARM and FPGA



For those interested in learning more about the use of LabVIEW in embedded applications and the underlying technology, this whitepaper provides a deep dive. It offers additional insights and knowledge that some might not expect from LabVIEW. In the image gallery, we are accompanied by a real project based on the NI System-on-Module (SOM). It is the electronic heart and brain of the "sunflower". This is a compact solar power plant with an 80% efficiency of energy utilization.

Contents

1	What is LabVIEW Embedded?.....	2
2	The Flexibility of Linux.....	2
3	My own LabVIEW Hardware, please!	3
4	Production in SMT Machinery in Batch Size 1.....	5
5	Tailored LabVIEW Hardware from the One-Stop-Shop	5
6	Developing Low-Level Device Drivers	6
7	Program an FPGA with the Mouse	7
8	The Balance between Microcontroller and FPGA.....	7
9	Possibilities and Limits	8
10	Appendix: Cyber-Security Low-Hanging Fruit.....	9
11	Glossary	10

1 What is LabVIEW Embedded?

LabVIEW is often mentioned in the same breath as laboratory work or testing. LabVIEW Embedded, on the other hand, scales further, namely into the area of embedded systems. The advantage is obvious: use LabVIEW on your own hardware for graphically programmable series products! The catch: many convenient features that we appreciate from NI's off-the-shelf hardware have to be developed ourselves. This includes the integration of custom specific analog converters and their driver development. In addition, special requirements are placed on application development, such as a scalable architecture and consistent error handling.

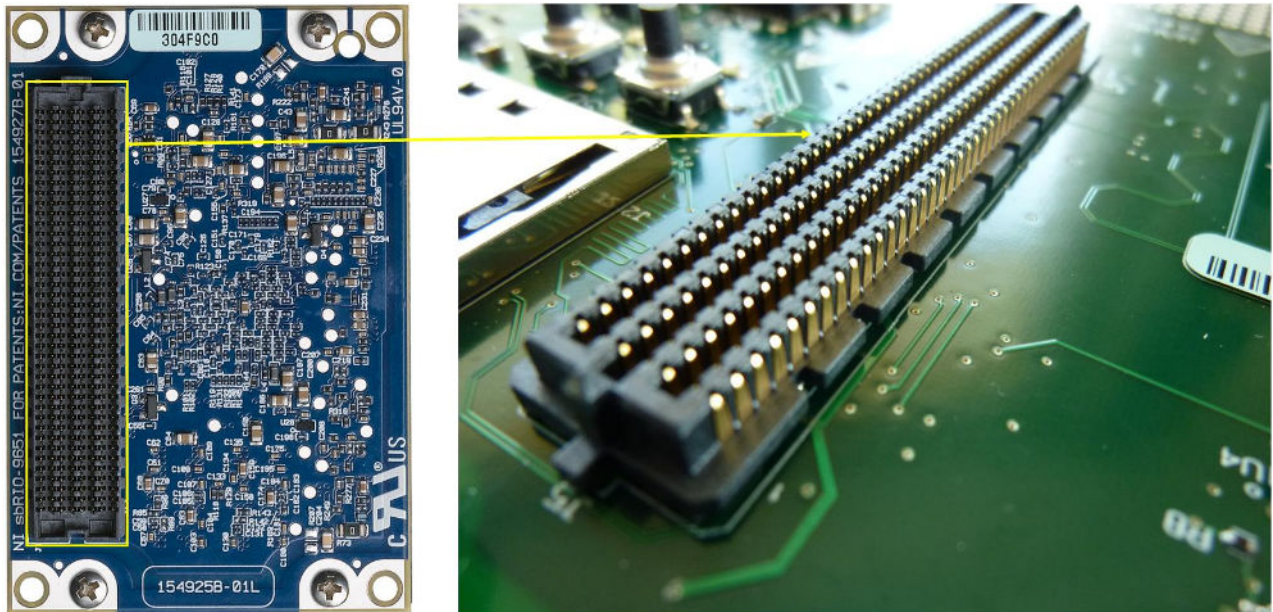


Fig. 1 | The 320-pin SEARAY connector between the SOM and the baseboard is a BGA (Ball Grid Array).

This whitepaper provides an overview of how the NI standard can be combined with customer-specific requirements using the case study of a real project. This standard is synonymous here with the NI System-on-Module, sbRIO9651 or SOM for short. This module with the dual-core ARM Cortex A9 and FPGA is again available for the long term. The two-board approach is described with this universal, flexible SOM, plugged into a baseboard with I/O of your choice (Fig. 1, right).

2 The Flexibility of Linux

LabVIEW Embedded runs on Open Source Linux. Thanks to this widespread operating system and the associated ecosystem, new possibilities are opening up for the developer community. Installed on the SOM is the SUMO distribution with the 4.14 kernel optimized for the embedded sector with a repository on NI's servers. The LabVIEW code is mapped to the operating system in

accordance with the Linux standard. There, the Linux ecosystem can be leveraged in LabVIEW with the usual functions. For example, the graphical environment has direct access to the "command-line" using a VI, allowing system commands to be executed directly. It is furthermore possible to access operating system libraries via the C-API (Application Programming Interface). In addition, experienced programmers even have the option of configuring the kernel individually. The system can be easily managed via one of the shells using an external terminal such as PuTTY (Fig. 6). It is even easier with the web-based, graphical configuration tool WebDAV.

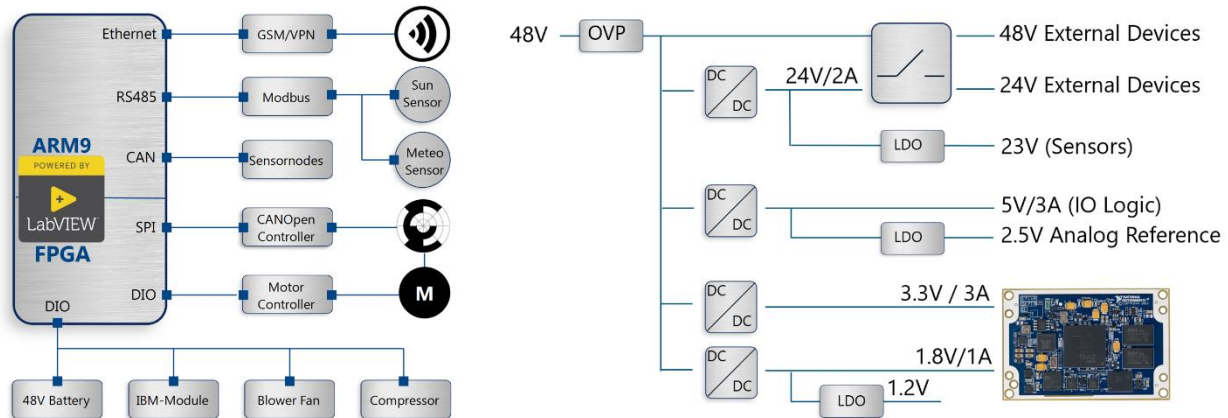


Fig. 2 | The first step is to design the baseboard.
On the left are the functions, on the right the power concept.

3 My own LabVIEW Hardware, please!

Almost every electronic I/O module available on the market can be connected to the SOM and controlled with LabVIEW (Fig.2, left). The possibilities are manifold: analog and digital I/O (memory mapped), synchronous (SPI, I²C) and asynchronous (UART) serial interfaces or parallel high-speed bus systems. Typical examples of such components are analog to digital converters (ADC).

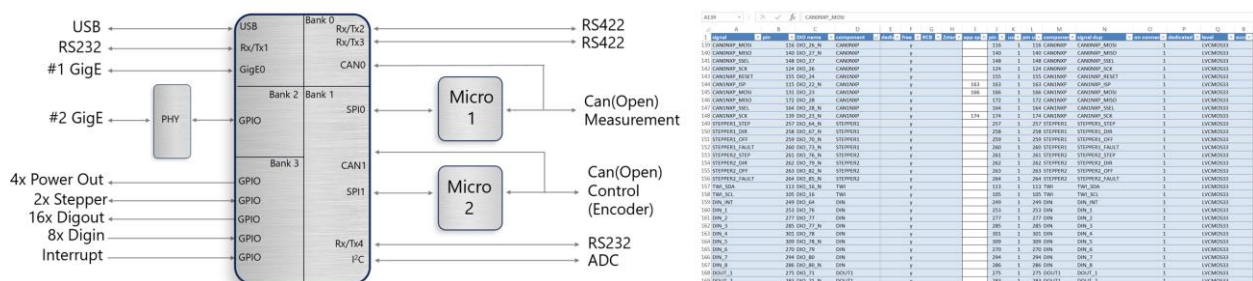


Fig. 3 | The next step is to assign the pins of the SEARAY connector (Fig. 1) to the I/O.
Due to the multiple assignment, we recommend defining a standard (right).

The advantage: the form and function of the hardware can be adapted to any task. This flexibility comes at a price, however, since this is where the two-board approach that is popular in the embedded sector comes into play. Hardware in the form of a baseboard with a proper power supply must always be developed, as the SOM alone does not function. The risk is manageable, as the time-critical high-speed circuits around the microcontroller and memory are already routed on the plugged-in SOM.

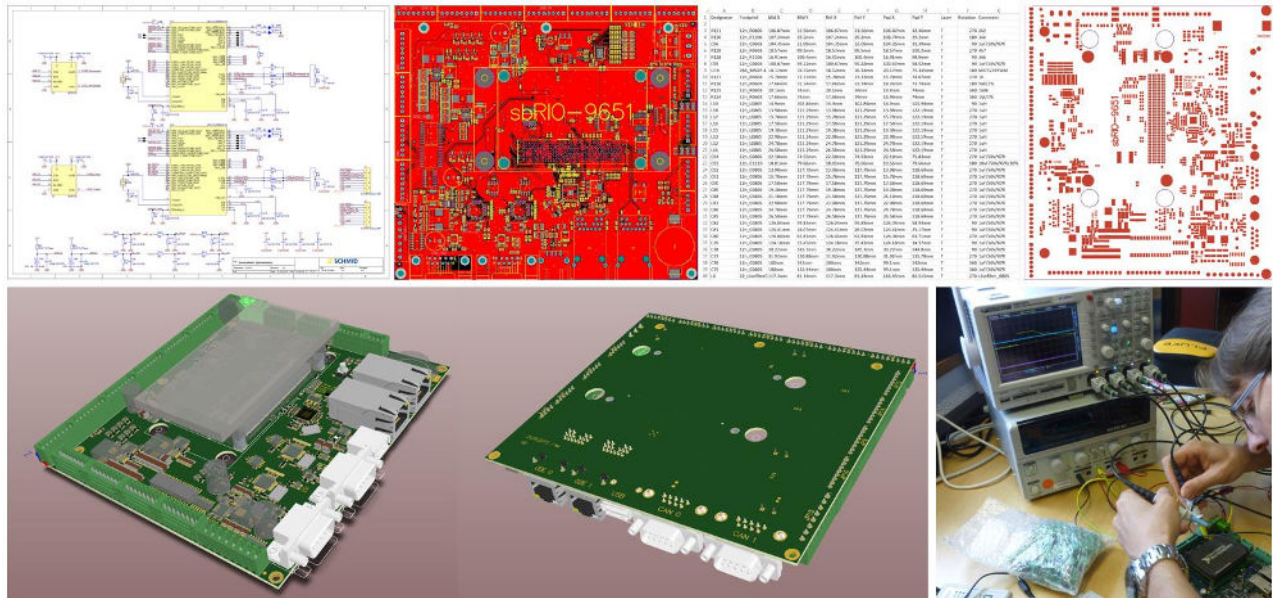


Fig. 4 | The development steps for embedded systems: create schematic design, create layout, generate production data (pick & place, template data), putting into operation after production

Due to the densely packed BGA connector, advanced developer knowledge is nevertheless required for the baseboard design. A hardware developer usually goes through the following steps from the initial idea to the final baseboard:

1. Analyze the requirements and evaluate the I/O components, e.g. A/D converters
2. Design of the baseboard (Fig. 2) and assignment of the SOM pins (Fig. 3)
3. The function, operating and limit values as well as timings can be found in the data sheets.
4. Create the schematics and incorporate knowledge from the data sheet and the application notes
5. Generate the bills of materials (BOM) and parts list for purchasing and production
6. Define the physical footprints according to production specifications and IPC standards
7. Create the board geometry and locate connectors and mounting holes
8. Map the mass and supply concept to the multilayer structure
9. Place electronic components and unbundle the connections
10. Route the layout, fill ground/power areas, check signal integrity
11. Create production data for PCB manufacturer and SMT line

4 Production in SMT Machinery in Batch Size 1

The 320-pin Searay connector between the SOM and the baseboard is a BGA (Ball Grid Array, Fig. 1). The soldering points are therefore not accessible from the outside. That's why it is advantageous for production to be carried out by an EMS (Electronic Manufacturing Service provider) with professionally equipped SMT machinery (Fig. 5). At the start of development, the quantities are often small, so it is beneficial to choose a manufacturer that can handle prototyping, NPI (New Product Introduction) and the batch size 1. Technically, on schedule and also economically.

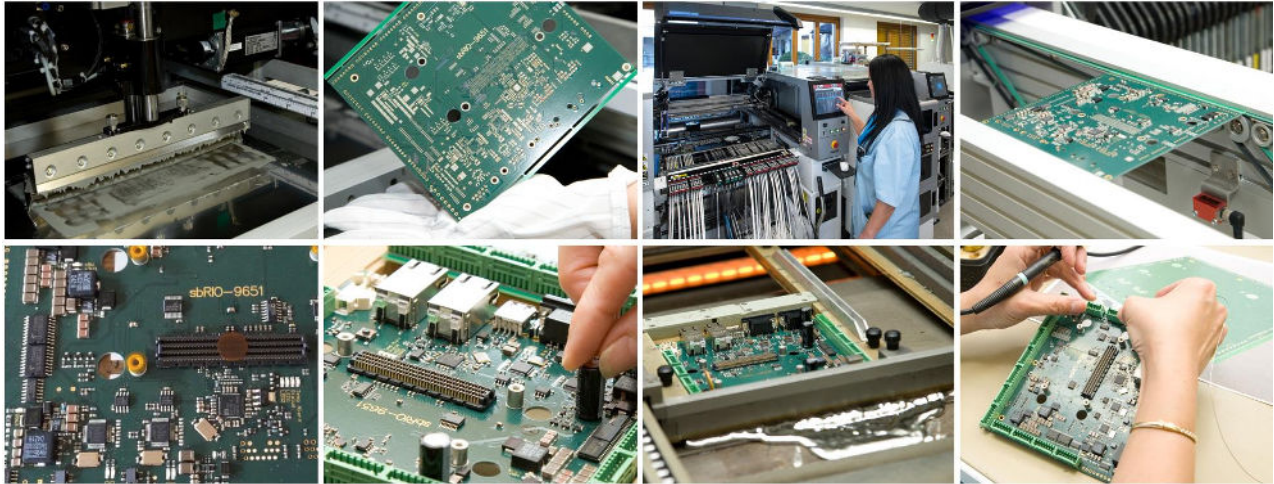


Fig. 5 | How an embedded system is manufactured: Solder paste printing, automatic assembly, reflow soldering, manual assembly, wave soldering and repairs

5 Tailored LabVIEW Hardware from the One-Stop-Shop

If the EDA infrastructure (Electronic Design Automation), the required hardware, software and operating system expertise or the SMT machinery are missing, NI partner Schmid Elektronik offers a one-stop-shop for tailored LabVIEW hardware. This includes a development and production service for individual baseboards with customer-specific industrial electronics. A production team optimized for low-volume/high-mix and a high-tech machinery optimized for prototyping and series production are also available, supplemented by a test and EMC laboratory. On the software side, this includes modification of the Linux kernel as well as the development of individual device drivers and their integration into the LabVIEW environment. All in all, a convenient service for anyone who wants to run LabVIEW on their own hardware. A customer once put it this way:

"We value the mutual trust and professional cooperation with our partner Schmid Elektronik, which offers absolutely reliable quality in the development of hardware and software as well as in production and testing. The agile and open-minded way of working enables flexible, lean processes, especially for our prototypes and small series. Schmid Elektronik is one of our pillars in adapting the NI platform to our special needs in the field of high-voltage DC transmission."

(Julian Lange, Siemens Energy)

6 Developing Low-Level Device Drivers

As soon as the hardware is available, the low-level drivers are developed. Drivers access external I/O modules and they are abstracted with a high-level VI. If the components are connected to the FPGA pins, the drivers are written graphically on the FPGA using LabVIEW. If they are connected directly to the microcontroller, three options are offered under Linux:

1. If the driver is available as a binary, it can be executed directly from LabVIEW by the command line VI. In this case, LabVIEW is logged on to Linux as "lvuser". The achievable response times are in the double-digit millisecond range due to access via the operating system (Fig. 6, top left).
2. If the driver is available as a C source, the executable is built using the Eclipse IDE.
3. Finally, a library (shared object) can be created with Eclipse and accessed from LabVIEW via the C-API-VI, similar to a DLL under Windows. Thanks to direct access to the C library, response times in the two-digit microsecond range are possible (Fig. 6, bottom left).

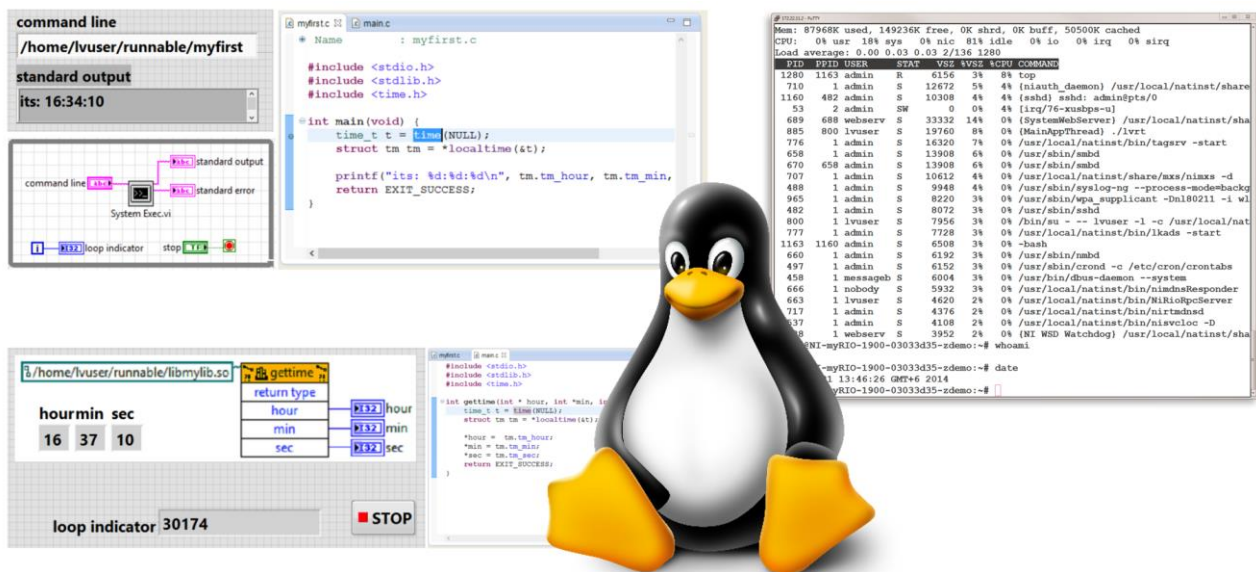


Fig. 6 | Interfaces between LabVIEW and Linux: top left, option 1) and bottom left, option 3. On the right, the PuTTY external terminal commonly used with Linux.

7 Program an FPGA with the Mouse

The Artix 7 FPGA in the SOM is software-configurable, parallel hardware. Time-critical tasks and I/Os are delegated to it in order to relieve the microcontroller. Decisive key data are the functions that can be realized and the timing. Until now, specialists were needed to program FPGAs. With LabVIEW, even engineers without this experience gain access to the powerful technology of reconfigurable logic (Fig. 7). Using intuitive function blocks, a wide range of analog, digital and serial process signals can be integrated, linked and pre-processed in parallel before they are forwarded to the microcontroller.

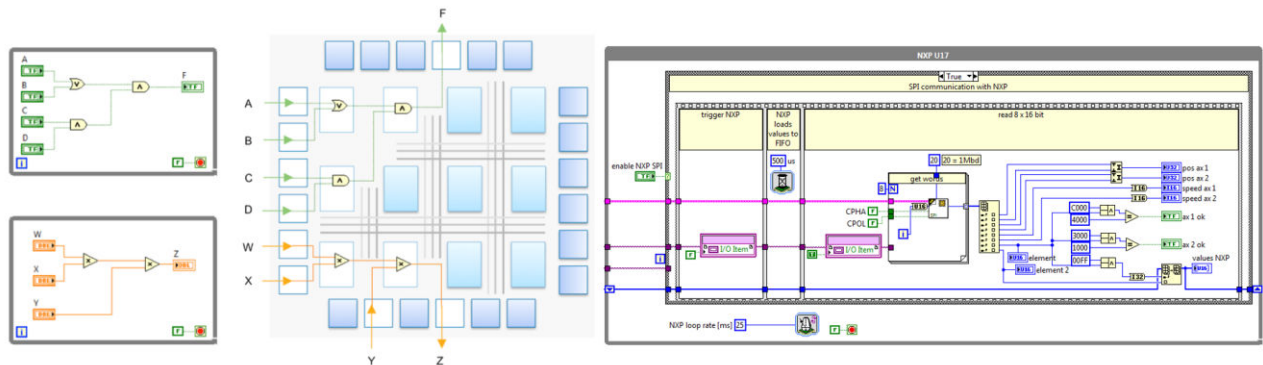


Fig. 7 | Developing low-level drivers with LabVIEW FPGA. Example: communication between the FPGA and an external microcontroller, which can integrate an encoder via CANOpen

The functionality of the application is directly limited by the number of gates available. Known characteristic metrics, which operation (addition, filter, FFT) requires how many gates, provide a valuable basis for deciding on the maximum possible range of functions. When it comes to timing, even the graphically-minded application programmer thinks in "ticks", the smallest FPGA time increment in the order of nanoseconds. This defines how much time is required for logical and mathematical operations and I/O accesses. The result: guide values for the overall system timing.

8 The Balance between Microcontroller and FPGA

When designing the software, the embedded application should be carefully balanced between the microcontroller and the FPGA. The former is responsible for the main high-level functions. Low-level details such as device drivers, time-critical code, digital filters, combinatorial and sequential logic, scaling, fixed-point and integer arithmetic are better outsourced to the FPGA. The LabVIEW FPGA diagram is finally translated into VHDL code with the FPGA tools, compiled into a bitfile (firmware) and loaded into the FPGA via LabVIEW Realtime.

9 Possibilities and Limits

Due to the usually high project requirements of our customers, we have often reached technical limits over the last 15 years. Robust device drivers in multitasking mode required quite some know-how and were often real time wasters. Every now and then, the graphical overhead put a spoke in our (performance) wheel and we had to optimize code in a targeted manner. Consistent error handling turned out to be the key. Apart from the limitations in Table 1, Schmid Elektronik's Zbrain offers a flexible platform with possibilities that can cover the entire area of value creation with just a single programming paradigm: proof-of-concept, minimum viable products (MVPs), prototypes, series products and testing.

Possibilities	Limits (No-Mans-Land)
• Complex application logic, multitasking • Robust 24/7 real-time operation • ADC block sampling standard 450kHz, scalable up to 40MHz • Flexible data communication • Customized I/O modules • Floating point arithmetic • Safety functions (in the FPGA)	• Continuous lowest-power applications in mW. Low-power for LabVIEW can be solved with a combination of the SOM with an RTC or a low-power coprocessor. • LabVIEW on your own microcontroller. This was previously possible with the C-Generator. This entered the end-of-life cycle (EOL) in 2017. • This puts 8-bit small targets out of reach • Price-sensitive quantities

Table 1 | Possibilities and limits of LabVIEW Embedded

10 Appendix: Cyber-Security Low-Hanging Fruit

Security of the SOM against external manipulation and attacks

It depends on how strongly you want to secure the SOM against such attacks. As shown below in the links to the security best practices, NI offers various levels of security: from "Open" (i.e. not secure at all, but convenient to work with), to "Recommended" and "Extreme" (completely sealed off). The topic of "network" is also addressed. SELinux (Security Enhanced Linux) can be used to restrict access. This limits the access of processes to the minimum necessary resources. The higher the security level, the more complex the configuration and administration. Documentation on the NI website:

- [SELinux - Addressing Access Control Security in LabVIEW RIO Devices - NI](#)
- [NI Linux Real-Time Security User Guide](#)
- [Overview of Best Practices for Security on RIO Systems - NI](#)
 - [Best Practices for Security on RIO Systems: Part 1 Recommended](#)
 - [Best Practices for Security on RIO Systems: Part 2 Optional](#)
 - [Best Practices for Security on RIO Systems: Part 3 Extreme](#)

Information on the kernel and the NI RT system image

- Single-board RIOs based on the ARM architecture, including the NI System-on-Module, are delivered with the Linux kernel 4.14 and the Sumo distribution.
- The RTOS team announces important changes and corrections via the open source NILRT repository on [GitHub](#).
- NI Linux RT images have a backwards compatibility of 4 years. For example, the 2022Q4 image supports LabVIEW 2022, 2021, 2020 and 2019.
- An NI Linux RT image can be downloaded [here](#).
- NI hardware and software operating system compatibility is described [here](#).
- Regular updates on critical security features are available [here](#).
- Companies such as neosoft.ca offer further security components, e.g. a firewall
- See also [NI Linux RT Knowledge Base](#) from Hampel Software Engineering

NI-Hardware- und Software-Betriebssystemkompatibilität

Wählen Sie aus der Dropdown-Liste ein NI-Produkt aus. Bitte beachten Sie, dass einige Hardwaremodelle von mehreren Betriebssystemen unterstützt werden und für jeden Treiber, der die Unterstützung von Betriebssystemen.

Hinweis: Informationen zur Unterstützung älterer Betriebssysteme finden Sie in der Datei „NI Hardware and Software Operating System Compatibility“ im Abschnitt [Downloads](#) am Ende dieser Seite.

6890-3851 (CompactRIO)

Inden folgenden Tabellen sehen Sie die jeweils aktuelle Version des Treibers, für die Unterstützung verfügbar ist.*

Installiert auf	NI CompactRIO/NI-900
Windows	Windows 11 (64 Bit) Windows 10 (64 Bit) Windows Server (2022, 2019, 2016, 2012 R2) Windows Server (2012 R2, 2016, 2019, 2022) HMI/Touchscreen
Linux Desktop	Ubuntu 22.04 Ubuntu 20.04 Ubuntu 18.04 Ubuntu 16.04 Ubuntu 14.04 Ubuntu 12.04 Ubuntu 10.04 Ubuntu 8.04 Ubuntu 6.04 Ubuntu 4.04 Ubuntu 2.04 Ubuntu 0.04
Linux Real-Time	Linux Real-Time Linux Real-Time

* Green: Issues Resolved; Yellow: In Progress; Red: Not Supported

NI Linux RT System Image

Um das Betriebssystem NI Linux Real-Time auf den meisten Real-Time-Controllern von NI auszuführen, ist das NI Linux RT System Image erforderlich.

[Mehr lesen](#)

DOWNLOADS

Unterstütztes OS: Windows

Version: 2024 Q2

Enthaltene Editionen: 2024 Q2, 2024 Q1, 2023 Q4, 2023 Q3, 2023 Q2, 2023 Q1, 2022 Q4, 2022 Q3, 2021 Q4, 2021 Q3, 2021 Q2, 2021 Q1, 2020 Q4, 2020 Q3, 2020 Q2, 2020 Q1, 2019 Q4, 2019 Q3, 2019 Q2, 2019 Q1, 2018 Q4, 2018 Q3, 2018 Q2, 2018 Q1, 2017 Q4, 2017 Q3, 2017 Q2, 2017 Q1, 2016 Q4, 2016 Q3, 2016 Q2, 2016 Q1, 2015 Q4, 2015 Q3, 2015 Q2, 2015 Q1, 2014 Q4, 2014 Q3, 2014 Q2, 2014 Q1, 2013 Q4, 2013 Q3, 2013 Q2, 2013 Q1, 2012 Q4, 2012 Q3, 2012 Q2, 2012 Q1, 2011 Q4, 2011 Q3, 2011 Q2, 2011 Q1, 2010 Q4, 2010 Q3, 2010 Q2, 2010 Q1, 2009 Q4, 2009 Q3, 2009 Q2, 2009 Q1, 2008 Q4, 2008 Q3, 2008 Q2, 2008 Q1, 2007 Q4, 2007 Q3, 2007 Q2, 2007 Q1, 2006 Q4, 2006 Q3, 2006 Q2, 2006 Q1, 2005 Q4, 2005 Q3, 2005 Q2, 2005 Q1, 2004 Q4, 2004 Q3, 2004 Q2, 2004 Q1, 2003 Q4, 2003 Q3, 2003 Q2, 2003 Q1, 2002 Q4, 2002 Q3, 2002 Q2, 2002 Q1, 2001 Q4, 2001 Q3, 2001 Q2, 2001 Q1, 2000 Q4, 2000 Q3, 2000 Q2, 2000 Q1, 1999 Q4, 1999 Q3, 1999 Q2, 1999 Q1, 1998 Q4, 1998 Q3, 1998 Q2, 1998 Q1, 1997 Q4, 1997 Q3, 1997 Q2, 1997 Q1, 1996 Q4, 1996 Q3, 1996 Q2, 1996 Q1, 1995 Q4, 1995 Q3, 1995 Q2, 1995 Q1, 1994 Q4, 1994 Q3, 1994 Q2, 1994 Q1, 1993 Q4, 1993 Q3, 1993 Q2, 1993 Q1, 1992 Q4, 1992 Q3, 1992 Q2, 1992 Q1, 1991 Q4, 1991 Q3, 1991 Q2, 1991 Q1, 1990 Q4, 1990 Q3, 1990 Q2, 1990 Q1, 1989 Q4, 1989 Q3, 1989 Q2, 1989 Q1, 1988 Q4, 1988 Q3, 1988 Q2, 1988 Q1, 1987 Q4, 1987 Q3, 1987 Q2, 1987 Q1, 1986 Q4, 1986 Q3, 1986 Q2, 1986 Q1, 1985 Q4, 1985 Q3, 1985 Q2, 1985 Q1, 1984 Q4, 1984 Q3, 1984 Q2, 1984 Q1, 1983 Q4, 1983 Q3, 1983 Q2, 1983 Q1, 1982 Q4, 1982 Q3, 1982 Q2, 1982 Q1, 1981 Q4, 1981 Q3, 1981 Q2, 1981 Q1, 1980 Q4, 1980 Q3, 1980 Q2, 1980 Q1, 1979 Q4, 1979 Q3, 1979 Q2, 1979 Q1, 1978 Q4, 1978 Q3, 1978 Q2, 1978 Q1, 1977 Q4, 1977 Q3, 1977 Q2, 1977 Q1, 1976 Q4, 1976 Q3, 1976 Q2, 1976 Q1, 1975 Q4, 1975 Q3, 1975 Q2, 1975 Q1, 1974 Q4, 1974 Q3, 1974 Q2, 1974 Q1, 1973 Q4, 1973 Q3, 1973 Q2, 1973 Q1, 1972 Q4, 1972 Q3, 1972 Q2, 1972 Q1, 1971 Q4, 1971 Q3, 1971 Q2, 1971 Q1, 1970 Q4, 1970 Q3, 1970 Q2, 1970 Q1, 1969 Q4, 1969 Q3, 1969 Q2, 1969 Q1, 1968 Q4, 1968 Q3, 1968 Q2, 1968 Q1, 1967 Q4, 1967 Q3, 1967 Q2, 1967 Q1, 1966 Q4, 1966 Q3, 1966 Q2, 1966 Q1, 1965 Q4, 1965 Q3, 1965 Q2, 1965 Q1, 1964 Q4, 1964 Q3, 1964 Q2, 1964 Q1, 1963 Q4, 1963 Q3, 1963 Q2, 1963 Q1, 1962 Q4, 1962 Q3, 1962 Q2, 1962 Q1, 1961 Q4, 1961 Q3, 1961 Q2, 1961 Q1, 1960 Q4, 1960 Q3, 1960 Q2, 1960 Q1, 1959 Q4, 1959 Q3, 1959 Q2, 1959 Q1, 1958 Q4, 1958 Q3, 1958 Q2, 1958 Q1, 1957 Q4, 1957 Q3, 1957 Q2, 1957 Q1, 1956 Q4, 1956 Q3, 1956 Q2, 1956 Q1, 1955 Q4, 1955 Q3, 1955 Q2, 1955 Q1, 1954 Q4, 1954 Q3, 1954 Q2, 1954 Q1, 1953 Q4, 1953 Q3, 1953 Q2, 1953 Q1, 1952 Q4, 1952 Q3, 1952 Q2, 1952 Q1, 1951 Q4, 1951 Q3, 1951 Q2, 1951 Q1, 1950 Q4, 1950 Q3, 1950 Q2, 1950 Q1, 1949 Q4, 1949 Q3, 1949 Q2, 1949 Q1, 1948 Q4, 1948 Q3, 1948 Q2, 1948 Q1, 1947 Q4, 1947 Q3, 1947 Q2, 1947 Q1, 1946 Q4, 1946 Q3, 1946 Q2, 1946 Q1, 1945 Q4, 1945 Q3, 1945 Q2, 1945 Q1, 1944 Q4, 1944 Q3, 1944 Q2, 1944 Q1, 1943 Q4, 1943 Q3, 1943 Q2, 1943 Q1, 1942 Q4, 1942 Q3, 1942 Q2, 1942 Q1, 1941 Q4, 1941 Q3, 1941 Q2, 1941 Q1, 1940 Q4, 1940 Q3, 1940 Q2, 1940 Q1, 1939 Q4, 1939 Q3, 1939 Q2, 1939 Q1, 1938 Q4, 1938 Q3, 1938 Q2, 1938 Q1, 1937 Q4, 1937 Q3, 1937 Q2, 1937 Q1, 1936 Q4, 1936 Q3, 1936 Q2, 1936 Q1, 1935 Q4, 1935 Q3, 1935 Q2, 1935 Q1, 1934 Q4, 1934 Q3, 1934 Q2, 1934 Q1, 1933 Q4, 1933 Q3, 1933 Q2, 1933 Q1, 1932 Q4, 1932 Q3, 1932 Q2, 1932 Q1, 1931 Q4, 1931 Q3, 1931 Q2, 1931 Q1, 1930 Q4, 1930 Q3, 1930 Q2, 1930 Q1, 1929 Q4, 1929 Q3, 1929 Q2, 1929 Q1, 1928 Q4, 1928 Q3, 1928 Q2, 1928 Q1, 1927 Q4, 1927 Q3, 1927 Q2, 1927 Q1, 1926 Q4, 1926 Q3, 1926 Q2, 1926 Q1, 1925 Q4, 1925 Q3, 1925 Q2, 1925 Q1, 1924 Q4, 1924 Q3, 1924 Q2, 1924 Q1, 1923 Q4, 1923 Q3, 1923 Q2, 1923 Q1, 1922 Q4, 1922 Q3, 1922 Q2, 1922 Q1, 1921 Q4, 1921 Q3, 1921 Q2, 1921 Q1, 1920 Q4, 1920 Q3, 1920 Q2, 1920 Q1, 1919 Q4, 1919 Q3, 1919 Q2, 1919 Q1, 1918 Q4, 1918 Q3, 1918 Q2, 1918 Q1, 1917 Q4, 1917 Q3, 1917 Q2, 1917 Q1, 1916 Q4, 1916 Q3, 1916 Q2, 1916 Q1, 1915 Q4, 1915 Q3, 1915 Q2, 1915 Q1, 1914 Q4, 1914 Q3, 1914 Q2, 1914 Q1, 1913 Q4, 1913 Q3, 1913 Q2, 1913 Q1, 1912 Q4, 1912 Q3, 1912 Q2, 1912 Q1, 1911 Q4, 1911 Q3, 1911 Q2, 1911 Q1, 1910 Q4, 1910 Q3, 1910 Q2, 1910 Q1, 1909 Q4, 1909 Q3, 1909 Q2, 1909 Q1, 1908 Q4, 1908 Q3, 1908 Q2, 1908 Q1, 1907 Q4, 1907 Q3, 1907 Q2, 1907 Q1, 1906 Q4, 1906 Q3, 1906 Q2, 1906 Q1, 1905 Q4, 1905 Q3, 1905 Q2, 1905 Q1, 1904 Q4, 1904 Q3, 1904 Q2, 1904 Q1, 1903 Q4, 1903 Q3, 1903 Q2, 1903 Q1, 1902 Q4, 1902 Q3, 1902 Q2, 1902 Q1, 1901 Q4, 1901 Q3, 1901 Q2, 1901 Q1, 1900 Q4, 1900 Q3, 1900 Q2, 1900 Q1, 1899 Q4, 1899 Q3, 1899 Q2, 1899 Q1, 1898 Q4, 1898 Q3, 1898 Q2, 1898 Q1, 1897 Q4, 1897 Q3, 1897 Q2, 1897 Q1, 1896 Q4, 1896 Q3, 1896 Q2, 1896 Q1, 1895 Q4, 1895 Q3, 1895 Q2, 1895 Q1, 1894 Q4, 1894 Q3, 1894 Q2, 1894 Q1, 1893 Q4, 1893 Q3, 1893 Q2, 1893 Q1, 1892 Q4, 1892 Q3, 1892 Q2, 1892 Q1, 1891 Q4, 1891 Q3, 1891 Q2, 1891 Q1, 1890 Q4, 1890 Q3, 1890 Q2, 1890 Q1, 1889 Q4, 1889 Q3, 1889 Q2, 1889 Q1, 1888 Q4, 1888 Q3, 1888 Q2, 1888 Q1, 1887 Q4, 1887 Q3, 1887 Q2, 1887 Q1, 1886 Q4, 1886 Q3, 1886 Q2, 1886 Q1, 1885 Q4, 1885 Q3, 1885 Q2, 1885 Q1, 1884 Q4, 1884 Q3, 1884 Q2, 1884 Q1, 1883 Q4, 1883 Q3, 1883 Q2, 1883 Q1, 1882 Q4, 1882 Q3, 1882 Q2, 1882 Q1, 1881 Q4, 1881 Q3, 1881 Q2, 1881 Q1, 1880 Q4, 1880 Q3, 1880 Q2, 1880 Q1, 1879 Q4, 1879 Q3, 1879 Q2, 1879 Q1, 1878 Q4, 1878 Q3, 1878 Q2, 1878 Q1, 1877 Q4, 1877 Q3, 1877 Q2, 1877 Q1, 1876 Q4, 1876 Q3, 1876 Q2, 1876 Q1, 1875 Q4, 1875 Q3, 1875 Q2, 1875 Q1, 1874 Q4, 1874 Q3, 1874 Q2, 1874 Q1, 1873 Q4, 1873 Q3, 1873 Q2, 1873 Q1, 1872 Q4, 1872 Q3, 1872 Q2, 1872 Q1, 1871 Q4, 1871 Q3, 1871 Q2, 1871 Q1, 1870 Q4, 1870 Q3, 1870 Q2, 1870 Q1, 1869 Q4, 1869 Q3, 1869 Q2, 1869 Q1, 1868 Q4, 1868 Q3, 1868 Q2, 1868 Q1, 1867 Q4, 1867 Q3, 1867 Q2, 1867 Q1, 1866 Q4, 1866 Q3, 1866 Q2, 1866 Q1, 1865 Q4, 1865 Q3, 1865 Q2, 1865 Q1, 1864 Q4, 1864 Q3, 1864 Q2, 1864 Q1, 1863 Q4, 1863 Q3, 1863 Q2, 1863 Q1, 1862 Q4, 1862 Q3, 1862 Q2, 1862 Q1, 1861 Q4, 1861 Q3, 1861 Q2, 1861 Q1, 1860 Q4, 1860 Q3, 1860 Q2, 1860 Q1, 1859 Q4, 1859 Q3, 1859 Q2, 1859 Q1, 1858 Q4, 1858 Q3, 1858 Q2, 1858 Q1, 1857 Q4, 1857 Q3, 1857 Q2, 1857 Q1, 1856 Q4, 1856 Q3, 1856 Q2, 1856 Q1, 1855 Q4, 1855 Q3, 1855 Q2, 1855 Q1, 1854 Q4, 1854 Q3, 1854 Q2, 1854 Q1, 1853 Q4, 1853 Q3, 1853 Q2, 1853 Q1, 1852 Q4, 1852 Q3, 1852 Q2, 1852 Q1, 1851 Q4, 1851 Q3, 1851 Q2, 1851 Q1, 1850 Q4, 1850 Q3, 1850 Q2, 1850 Q1, 1849 Q4, 1849 Q3, 1849 Q2, 1849 Q1, 1848 Q4, 1848 Q3, 1848 Q2, 1848 Q1, 1847 Q4, 1847 Q3, 1847 Q2, 1847 Q1, 1846 Q4, 1846 Q3, 1846 Q2, 1846 Q1, 1845 Q4, 1845 Q3, 1845 Q2, 1845 Q1, 1844 Q4, 1844 Q3, 1844 Q2, 1844 Q1, 1843 Q4, 1843 Q3, 1843 Q2, 1843 Q1, 1842 Q4, 1842 Q3, 1842 Q2, 1842 Q1, 1841 Q4, 1841 Q3, 1841 Q2, 1841 Q1, 1840 Q4, 1840 Q3, 1840 Q2, 1840 Q1, 1839 Q4, 1839 Q3, 1839 Q2, 1839 Q1, 1838 Q4, 1838 Q3, 1838 Q2, 1838 Q1, 1837 Q4, 1837 Q3, 1837 Q2, 1837 Q1, 1836 Q4, 1836 Q3, 1836 Q2, 1836 Q1, 1835 Q4, 1835 Q3, 1835 Q2, 1835 Q1, 1834 Q4, 1834 Q3, 1834 Q2, 1834 Q1, 1833 Q4, 1833 Q3, 1833 Q2, 1833 Q1, 1832 Q4, 1832 Q3, 1832 Q2, 1832 Q1, 1831 Q4, 1831 Q3, 1831 Q2, 1831 Q1, 1830 Q4, 1830 Q3, 1830 Q2, 1830 Q1, 1829 Q4, 1829 Q3, 1829 Q2, 1829 Q1, 1828 Q4, 1828 Q3, 1828 Q2, 1828 Q1, 1827 Q4, 1827 Q3, 1827 Q2, 1827 Q1, 1826 Q4, 1826 Q3, 1826 Q2, 1826 Q1, 1825 Q4, 1825 Q3, 1825 Q2, 1825 Q1, 1824 Q4, 1824 Q3, 1824 Q2, 1824 Q1, 1823 Q4, 1823 Q3, 1823 Q2, 1823 Q1, 1822 Q4, 1822 Q3, 1822 Q2, 1822 Q1, 1821 Q4, 1821 Q3, 1821 Q2, 1821 Q1, 1820 Q4, 1820 Q3, 1820 Q2, 1820 Q1, 1819 Q4, 1819 Q3, 1819 Q2, 1819 Q1, 1818 Q4, 1818 Q3, 1818 Q2, 1818 Q1, 1817 Q4, 1817 Q3, 1817 Q2, 1817 Q1, 1816 Q4, 1816 Q3, 1816 Q2, 1816 Q1, 1815 Q4, 1815 Q3, 1815 Q2, 1815 Q1, 1814 Q4, 1814 Q3, 1814 Q2, 1814 Q1, 1813 Q4, 1813 Q3, 1813 Q2, 1813 Q1, 1812 Q4, 1812 Q3, 1812 Q2, 1812 Q1, 1811 Q4, 1811 Q3, 1811 Q2, 1811 Q1, 1810 Q4, 1810 Q3, 1810 Q2, 1810 Q1, 1809 Q4, 1809 Q3, 1809 Q2, 1809 Q1, 1808 Q4, 1808 Q3, 1808 Q2, 1808 Q1, 1807 Q4, 1807 Q3, 1807 Q2, 1807 Q1, 1806 Q4, 1806 Q3, 1806 Q2, 1806 Q1, 1805 Q4, 1805 Q3, 1805 Q2, 1805 Q1, 1804 Q4, 1804 Q3, 1804 Q2, 1804 Q1, 1803 Q4, 1803 Q3, 1803 Q2, 1803 Q1, 1802 Q4, 1802 Q3, 1802 Q2, 1802 Q1, 1801 Q4, 1801 Q3, 1801 Q2, 1801 Q1, 1800 Q4, 1800 Q3, 1800 Q2, 1800 Q1, 1799 Q4, 1799 Q3, 1799 Q2, 1799 Q1, 1798 Q4, 1798 Q3, 1798 Q2, 1798 Q1, 1797 Q4, 1797 Q3, 1797 Q2, 1797 Q1, 1796 Q4, 1796 Q3, 1796 Q2, 1796 Q1, 1795 Q4, 1795 Q3, 1795 Q2, 1795 Q1, 1794 Q4, 1794 Q3, 1794 Q2, 1794 Q1, 1793 Q4, 1793 Q3, 1793 Q2, 1793 Q1, 1792 Q4, 1792 Q3, 1792 Q2, 1792 Q1, 1791 Q4, 1791 Q3, 1791 Q2, 1791 Q1, 1790 Q4, 1790 Q3, 1790 Q2, 1790 Q1, 1789 Q4, 1789 Q3, 1789 Q2, 1789 Q1, 1788 Q4, 1788 Q3, 1788 Q2, 1788 Q1, 1787 Q4, 1787 Q3, 1787 Q2, 1787 Q1, 1786 Q4, 1786 Q3, 1786 Q2, 1786 Q1, 1785 Q4, 1785 Q3, 1785 Q2, 1785 Q1, 1784 Q4, 1784 Q3, 1784 Q2, 1784 Q1, 1783 Q4, 1783 Q3, 1783 Q2, 1783 Q1, 1782 Q4, 1782 Q3, 1782 Q2, 1782 Q1, 1781 Q4, 1781 Q3, 1781 Q2, 1781 Q1, 1780 Q4, 1780 Q3, 1780 Q2, 1780 Q1, 1779 Q4, 1779 Q3, 1779 Q2, 1779 Q1, 1778 Q4, 1778 Q3, 1778 Q2, 1778 Q1, 1777 Q4, 1777 Q3, 1777 Q2, 1777 Q1, 1776 Q4, 1776 Q3, 1776 Q2, 1776 Q1, 1775 Q4, 1775 Q3, 1775 Q2, 1775 Q1, 1774 Q4, 1774 Q3, 1774 Q2, 1774 Q1, 1773 Q4, 1773 Q3, 1773 Q2, 1773 Q1, 1772 Q4, 1772 Q3, 1772 Q2, 1772 Q1, 1771 Q4, 1771 Q3, 1771 Q2, 1771 Q1, 1770 Q4, 1770 Q3, 1770 Q2, 1770 Q1, 1769 Q4, 1769 Q3, 1769 Q2, 1769 Q1, 1768 Q4, 1768 Q3, 1768 Q2, 1768 Q1, 1767 Q4, 1767 Q3, 1767 Q2, 1767 Q1, 1766 Q4, 1766 Q3, 1766 Q2, 1766 Q1, 1765 Q4, 1765 Q3, 1765 Q2, 1765 Q1, 1764 Q4, 1764 Q3, 1764 Q2, 1764 Q1, 1763 Q4, 1763 Q3, 1763 Q2, 1763 Q1, 1762 Q4, 1762 Q3, 1762 Q2, 1762 Q1, 1761 Q4, 1761 Q3, 1761 Q2, 1761 Q1, 1760 Q4, 1760 Q3, 1760 Q2, 1760 Q1, 1759 Q4, 1759 Q3, 1759 Q2, 1759 Q1, 1758 Q4, 1758 Q3, 1758 Q2, 1758 Q1, 1757 Q4, 1757 Q3, 1757 Q2, 1757 Q1, 1756 Q4, 1756 Q3, 1756 Q2, 1756 Q1, 1755 Q4, 1755 Q3, 1755 Q2, 1755 Q1, 1754 Q4, 1754 Q3, 1754 Q2, 1754 Q1, 1753 Q4, 1753 Q3, 1753 Q2, 1753 Q1, 1752 Q4, 1752 Q3, 1752 Q2, 1752 Q1, 1751 Q4, 1751 Q3, 1751 Q2, 1751 Q1, 1750 Q4, 1750 Q3, 1750 Q2, 1750 Q1, 1749 Q4, 1749 Q3, 1749 Q2, 1749 Q1, 1748 Q4, 1748 Q3, 1748 Q2, 1748 Q1, 1747 Q4, 1747 Q3, 1747 Q2, 1747 Q1, 1746 Q4, 1746 Q3, 1746 Q2, 1746 Q1, 1745 Q4, 1745 Q3, 1745 Q2, 1745 Q1, 1744 Q4, 1744 Q3, 1744 Q2, 1744 Q1, 1743 Q4, 1743 Q3, 1743 Q2, 1743 Q1, 1742 Q4, 1742 Q3, 1742 Q2, 1742 Q1, 1741 Q4, 1741 Q3, 1741 Q2, 1741 Q1, 1740 Q4, 1740 Q3, 1740 Q2, 1740 Q1, 1739 Q4, 1739 Q3, 1739 Q2, 1739 Q1, 1738 Q4, 1738 Q3, 1738 Q2, 1738 Q1, 1737 Q4, 1737 Q3, 1737 Q2, 1737 Q1, 1736 Q4, 1736 Q3, 1736 Q2, 1736 Q1, 1735 Q4, 1735 Q3, 1735 Q2, 1735 Q1, 1734 Q4, 1734 Q3, 1734 Q2, 1734 Q1, 1733 Q4, 1733 Q3, 1733 Q2, 1733 Q1, 1732 Q4, 1732 Q3, 1732 Q2, 1732 Q1, 1731 Q4, 1731 Q3, 1731 Q2, 1731 Q1, 1730 Q4, 1730 Q3, 1730 Q2, 1730 Q1, 1729 Q4, 1729 Q3, 1729 Q2, 1729 Q1, 1728 Q4, 1728 Q3, 1728 Q2, 1728 Q1, 1727 Q4, 1727 Q3, 1727 Q2, 1727 Q1, 1726 Q4, 1726 Q3, 1726 Q2, 1726 Q1, 1725 Q4, 1725 Q3, 1725 Q2, 1725 Q1, 1724 Q4, 1724 Q3, 1724 Q2, 1724 Q1, 1723 Q4, 1723 Q3, 1723 Q2, 1723 Q1, 1722 Q4, 1722 Q3, 1722 Q2, 1722 Q1, 1721 Q4, 1721 Q3, 1721 Q2, 1721 Q1, 1720 Q4, 1720 Q3, 1720 Q2, 1720 Q1, 1719 Q4, 1719 Q3, 1719 Q2, 1719 Q1, 1718 Q4, 1718 Q3, 1718 Q2, 1718 Q1, 1717 Q4, 1717 Q3, 1717 Q2, 1717 Q1, 1716 Q4, 1716 Q3, 1716 Q2, 1716 Q1, 1715 Q4, 1715 Q3, 1715 Q2, 1715 Q1, 1714 Q4, 1714 Q3, 1714 Q2, 1714 Q1, 1713 Q4, 1713 Q3, 1713 Q2, 1713 Q1, 1712 Q4, 1712 Q3, 1712 Q2, 1712 Q1, 1711 Q4, 1711 Q3, 1711 Q2, 1711 Q1, 1710 Q4, 1710 Q3, 1710 Q2, 1710 Q1, 1709 Q4, 1709 Q3, 1709 Q2, 1709 Q1, 1708 Q4, 1708 Q3, 1708 Q2, 1708 Q1, 1707 Q4, 1707 Q3, 1707 Q2, 1707 Q1, 1706 Q4, 1706 Q3, 1706 Q2, 1706 Q1, 1705 Q4, 1705 Q3, 1705 Q2, 1705 Q1, 1704 Q4, 1704 Q3, 1704 Q2, 1704 Q1, 1703 Q4, 1703 Q3, 1703 Q2, 1703 Q1, 1702 Q4, 1702 Q3, 1702 Q2, 1702 Q1, 1701 Q4, 1701 Q3, 1701 Q2, 1701 Q1, 1700 Q4, 1700 Q3, 1700 Q2, 1700 Q1, 1699 Q4, 1699 Q3, 1699 Q2, 1699 Q1, 1698 Q4, 1698 Q3, 1698 Q2, 1698 Q1, 1697 Q4, 1697 Q3, 1697 Q2, 1697 Q1, 1696 Q4, 1696 Q3, 1696 Q2, 1696 Q1, 1695 Q4, 1695 Q3, 1695 Q2, 1695 Q1, 1694 Q4, 1694 Q3, 1694 Q2, 1694 Q1, 1693 Q4, 1693 Q3, 1693 Q2, 1693 Q1, 1692 Q4, 1692 Q3, 1692 Q2, 1692 Q1, 1691 Q4, 1691 Q3, 1691 Q2, 1691 Q1, 1690 Q4, 1690 Q3, 1690 Q2, 1690 Q1, 1689 Q4, 1689 Q3, 1689 Q2, 1689 Q1, 1688 Q4, 1688 Q3, 1688 Q2, 1688 Q1, 1687 Q4, 1687 Q3, 1687 Q2, 1687 Q1, 1686 Q4, 1686 Q3, 1686 Q2, 1686 Q1, 1685 Q4, 1685 Q3, 1685 Q2, 1685 Q1, 1684 Q4, 1684 Q3, 1684 Q2, 1684 Q1, 1683 Q4, 1683 Q3, 1683 Q2, 1683 Q1, 1682 Q4, 1682 Q3, 1682 Q2, 1682 Q1, 1681 Q4, 1681 Q3, 1681 Q2, 1681 Q1, 1680 Q4, 1680 Q3, 1680 Q2, 1680 Q1, 1679 Q4, 1679 Q3, 167

11 Glossary

Chapter 1	Embedded Systems	Combination of hardware and software that performs a sub-function within a larger system in which they are “embedded”.
	SOM, sbRIO9651	The NI S ystem- O n- M odule in the size of a business card.
	Dualcore	A microcontroller with two computing cores.
	ARM-Cortex A9	A dual-core 32-bit microcontroller with an ARM v7 instruction set.
	FPGA	Reconfigurable logic: F ield P rogrammable G ate A rray.
	Two-Board Approach	Here: a business card sized module contains only the microcontroller and its chipset and is generally backwards and forwards compatible in form, fit and function. It is plugged into a customised baseboard. This is a popular approach in the embedded industry, as microcontrollers and memory are usually fast-moving, unlike application I/O.
	Baseboard	Contains all application-specific components and accepts the microcontroller module via a high-pin connector, in this case a 320-pin SEARAY.
Chapter 2	Linux	Widely used open source operating system.
	Kernel	The main component of an operating system, in this case Linux.
	Linux Distribution	A selection of harmonised software, tools, apps and services around a Linux kernel.
	Repository, Repo	A central repository that developers use to make changes to the source code of an application or an image.
	Command-Line	Here it is a Linux interface that accepts text lines.
	C-API	A pplication P rogramming I nterface. In this context, it is a special LabVIEW VI that can execute external C code. This enables hardware access, for example.
	Shell	A command line interface to the Linux operating system, e.g. PuTTY.
	PuTTY	Is a S ecure S hell (SSH) to establish a connection to Linux.
	WebDAV	W eb-based D istributed A uthoring and V ersioning: Data exchange with a Linux system.
Chapter 3	I/O-Component	Here: components that integrates sensor I nput and actuator O utput.
	Memory Mapped	Fast, parallel access from the microcontroller to an I/O module via data bus.
	Synchronous	Here: Timed serial communication, e.g. SPI.
	Asynchronous	Here: Time-delayed communication, synchronisation with start/stop bit.
	SPI	Serial synchronous and very fast onboard bus for I/O modules.
	I ² C	TWI (Two-Wire Interface), originally I ² C: two-wire onboard bus for I/O.
	UART	U niversal, A synchronous R eceiver & T ransmitter for asynchronous data transmission. Very common for RS232, for example.
	A/D-Converter	Analogue converter, more precisely : Analogue to digital converter.
	Microcontroller	The NI SOM here is an ARM Cortex-A9 microcontroller.

	BGA-Connector	Ball Grid Array: a type of housing in which the connections are located at the bottom of the component. This is in the form of small solder beads that are arranged next to each other in a grid of columns and rows. In this way, a large number of connections are possible in a small space. However, such components can only be soldered by machine.
	Schematics	The graphical representation of an electronic circuit with components.
	Layout	Geometric arrangement of components/parts on an electronic circuit board and connecting them according to the diagram.
	Datasheet	This is a product description with properties, functions, specifications and materials of an electronic component.
	Application Notes	Detailed description of how an electronic component described in the data sheet is used in a specific application or use case.
	Bill of Materials (BOM)	A list of all electronic components contained in an assembly. It is the interface between development and production.
	IPC-Standards, Footprint	IPC creates standards for the component pads in the layout so that they can be easily produced by different electronics manufacturers.
	Multilayer	A printed circuit board consisting of several layers (4,6,8,12, etc).
	Signal Integrity	When the signals on the circuit board along the conductor tracks behave exactly as desired in terms of voltage and current.
	SMT and SMD	SMT: Surface Mounted Technology. SMD: Surface Mounted Devices.
Chapter 4	SEARAY	This is a 320-pin connector between two circuit boards that enables fast signals. The pitch of the BGA (see above) is 1.27mm.
	EMS	E lectronic M anufacturing S ervices.
	NPI	N ew P roduct I ntroduction. A product innovation is optimally transferred to series production.
	Batch Size 1	Production of individual items in one-piece flow. The background to this is a growing customer need to customise products. Changeover costs are reduced to a minimum thanks to lean processes.
	Solder Paste Printing	This is the very first step in SMT assembly. The solder paste - a mixture of tiny solder balls and flux - is applied precisely to the PCB to be assembled using a metal stencil.
	SMD Assembly	A pick & place robot picks the components from a reel with a gripper, measures them with an optical camera and places them on the board. That's why it's called pick & place. The components remain attached to the sticky solder paste until they are soldered.
	Reflow Soldering	The solder paste applied by the solder paste printer is re-melted/re-flowed in an oven via a controlled temperature profile and connects the solder connections of the components to the pads/landing surfaces of the PCB.
	THT-Assembly	This is called " T hrough- H ole T echnology" and is used when wired components are inserted into the circuit board and soldered over the wave.
	Wave Soldering	A previously hand-assembled circuit board passes through a solder bath on a conveyor belt. The entire underside of the circuit board is wetted by a literal wave of solder tin.

Chapter 5	EDA	E lectronic D esign A utomation: Software for the development of electronic hardware with schematic and layout. This is a sub-area of CAD.
	One-Stop-Shop	A customer receives several services from a single source. In this case, they can have customised LabVIEW hardware developed and produced and also receive the low-level drivers. This allows them to concentrate on the LabVIEW application from the very first hour.
	Low-Volume/ High-Mix	Here: electronics production of small batches (e.g. 50 units) of different products with several changeover operations during one working day.
	EMC Laboratory	A laboratory in which devices are tested for their immunity to interference and emitted interference. EMC : E lectro- M agnetic- C ompatibilty.
	Device Driver	Hardware-related control of electronic components, e.g. AD converters.
Chapter 6	High-Level-VI	A function block in LabVIEW that calls subroutines.
	FPGA	Reconfigurable logic: F ield P rogrammable G ate A rray.
	C-Source	Source code written in the "C" programming language.
	Eclipse-IDE	An integrated development environment for many programming languages, e.g. C
	Shared-Object	A code library under Linux (*.so).
	DLL	D ynamic L inked L ibrary: a code library under Windows (*.dll).
Chapter 7	Artix-7-FPGA	Alongside the microcontroller, a key component of the ZYNQ-7000 family.
	Gates	Logic circuits/cells in an FPGA: about 85,000 in the SOM.
	FFT	F ast F ourier T ransformation: converts signals from the time domain to the frequency domain.
	Ticks	A clock cycle in the FPGA in the order of nanoseconds.
	Timing	Here it is a time synchronisation of different software parts.
Chapter 8	Digital Filters	Amplify or attenuate certain frequencies in a signal.
	Combinatorial Logic	The output value depends directly on the input value.
	Sequential Logic	Output values also depend on stored states.
	Fixed-Point	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.
	Floating-Point	A number that is split into an exponent for the order of magnitude and a mantissa. Used for the practical and dynamic notation of very large and very small numbers.
	VHDL-Code	V ery H igh S peed I ntegrated C ircuits H ardware D escription L anguage: Hardware description language for FPGAs.
	Bitfile	Contains all the information of the LabVIEW block diagram, to configure the logic of the FPGA.
	Firmware	Software that issues machine commands to the hardware components.

	LabVIEW Realtime	A LabVIEW variant that is executed in real time down to microseconds on a microcontroller.
	LabVIEW FPGA	A LabVIEW variant that is executed in real time down to [ns] on an FPGA.
Chapter 9	Multitasking	Execution of multiple, independent tasks simultaneously.
	Overhead	Data that is not part of the user data but is required as additional information, e.g. for the abstraction of graphical code.
	Error handling	Intercept errors in a targeted manner and react to them in a controlled manner.
	Zbrain	Platform for graphical programming with LabVIEW Embedded, combined with a toolbox of modular hardware and scalable software as well as an SDK (Software Development Kit) from NI partner Schmid Elektronik.
	MVPs	M inimum V iable P roducts contain only the most necessary functions and are often needed to test an idea.
	24/7	Continuous industrial operation: the system runs around the clock.
	ADC-Blocksampling	Analogue values are recorded in blocks. High sampling rates are possible within these blocks. These blocks are then processed before the next block is recorded.
	Low-Power mW	Scalable, energy-saving operation of embedded systems.
	RTC	R ea L - T ime- C lock
	Coprocessor	A second microcontroller in addition to the main microcontroller. In this context, a low-power microcontroller controls the much more powerful main microcontroller. The system can then switch back and forth between low power and high power as required.
	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.
	8Bit-Small Targets	Miniature microprocessors that process 8 bits during one clock cycle.
Appendix	Cyber Security	Measures to protect computers against malicious attacks.
	SELinux	A Linux kernel extension that controls access to certain resources.
	Best Practices	Best practice based on experience.
	ARM-Architecture	A dvanced R ISC M achine: these are microcontrollers based on the RISC architecture (RISC: R educed I nstruction S et C omputer) and dominate in embedded systems. They take a different approach to hardware design than the better-known x86 architecture.
	Linux Kernel 4.14	A kernel from 2017 with long-term support until 2024.
	SUMO Distribution	The NI Linux RT Distribution for the System on Module.
	RTOS	R ea L - T ime O perating S ystem.