Int J Adv Manuf Technol
DOI 10.1007/s00170-015-6786-y

ORIGINAL ARTICLE

Integration of real-time Ethernet in LinuxCNC

Using the example of Sercos III

Elmar Wings · Marcel Müller · Marc Rochler

Received: 25 July 2014 / Accepted: 4 January 2015

Abstract In recent years, open-source software applications based on the operating system Linux have replaced previous proprietary software in many fields. However, up to now, the open-source software community did not come up with an appropriate CNC solution. Particularly, the requirement for real-time communication between system components has been a major challenge. The paper at hand presents a new approach for the integration of real-time Ethernet into LinuxCNC developed at the Institut für Maschinen- und Anlagenbau in Emden. For the first time, a CNC system entirely based on opensouce software has been realized that can compete with proprietary-embedded CNC.

Keywords LinuxCNC · EMC · Real-time Ethernet · Sercos III · CIFX 50-RE · NXIO 50-RE · Servo drive · CSB01

1 Introduction

The main task of the CNC is to control the relative motion between the tool and the workpiece. For this objective, the CNC processes drive-specific set values from G-code to control the axes of the machine tool in real-time [7, 15]. Usually, an embedded hardware-based CNC is used to process these set values and to control the motion in real time; though also CNC exist that are entirely software-based. The latter is very cost-efficient. Contrary to the embedded systems, it is possible to use standard PC hardware. Hence, the need to rely on hardware-specific operating systems does not apply. From a technical point of view, software-based CNC can outperform their counterparts in three different aspects. First and foremost, since the CNC is entirely implemented in the software, it can be modified to a great extent. Second, the use of PC hardware is accompanied with the usability of standard interfaces like Ethernet, the Universal Serial Bus (USB) and the Peripheral Component Interconnect (PCI). The integration into a company's network does not represent a problem. Finally, modern PC hardware offers an exceptional level of computing power, compared to embedded solutions. However, in order to utilize a software-based CNC to its full extent, it is necessary to establish a real-time software architecture that can interface with the system components in the most efficient way possible. The algorithm for geometry processing and trajectory generation within the CNC significantly determines the achievable quality of the set values [7].

The paper at hand focuses on the path to transmit set values from the CNC to the axes. Both the generation and transmission of the set values have to be performed in real-time. Otherwise, the axes are not able to move synchronously. The result could be harmful vibration, increased chatter, poor surface finish, and a machined workpiece that does not show precise dimensional accuracy.

E. Wings · M. Müller (✉) · M. Rochler
Institut für Maschinen- und Anlagenbau, Constantiaplatz 4, 26723, Emden, Germany
e-mail: Marcel.Mueller@hs-emden-leer.de

E. Wings
e-mail: Elmar.Wings@hs-emden-leer.de

M. Rochler
e-mail: Marc.Rochler@technik-emden.de

Published online: 18 January 2015

2 State of the art

Particularly to control servo drives, the real-time Ethernet protocol Sercos III is used to interconnect the servo drives with the control. Sercos III is deterministic with short cycle times and guarantees a high level of synchronism (low jitter) [6]. Sercos III is used to control intelligent servo drives. These kind of drives provide, e.g., a closed-loop control which enables a decentralized control. Furthermore, it relieves the CNC so that it has more performance for other tasks like the set value generation. Proprietary control systems from Bosch Rexroth AG and Beckhoff Automation use real-time Ethernet for many years.

LinuxCNC (formerly Enhanced Machine Controller or EMC2) is a software-based CNC utilizing standard PC hardware. This software is released under the terms of the GNU General Public Licence version 2 (GPLv2). This means the control software is free of charge and open-source [21].

LinuxCNC is a software for real-time control, however, it does not support real-time Ethernet; though the protocols are standardized. Exclusive interfaces like the parallel port and specific expansion boards are supported by LinuxCNC via ISA or PCI to control steppers and servo drives [18]. A driver for Modbus is integrated. Unfortunately, Modbus is not the best choice for hard real-time environments like motion control [18]. An EtherCAT drive is still under development and supports a few devices [17]. But until now, the EtherCAT driver is not a part of the LinuxCNC master branch[1]. Currently, LinuxCNC spends a lot of time with step pulse generation for stepper motors or closed-loop control for servo drives.

3 Objective

The objective of the development work carried out at the Institut für Maschinen- und Anlagenbau was to integrate a real-time capable interface into LinuxCNC so that the advantages of intelligent servo drives are suitable for LinuxCNC. This can be seen in the example of Sercos III. The solution shall enable the control of a servo drive with LinuxCNC via Sercos III in real-time. For that purpose, the focus is on the cycle time and the jitter.

4 Approach

Sercos III is using a master/slave arrangement to transmit data between nodes. For that purpose, the Sercos III master controls the timing. Usually, the CNC provides the function of the Sercos III master. Other nodes like servo drives or I/O modules are the controlled slaves [23]. Whereas, in order to obtain a high level of flexibility, the master is realized by using the PC card CIFX 50-RE from Hilscher Gesellschaft für Systemautomation mbH. The protocol stack will be executed on the PC card and can be replaced by other protocol stacks, e.g., EtherCAT, PROFINET, Ethernet/IP, by loading another protocol-specific firmware [13]. To establish communication with the CIFX 50-RE, it is required to install the userspace I/O driver uio_netx and the library libcifx [10]. The library contains the hardware-specific application programming interface CIFX-API which is described in detail in [8].

4.1 Real-time in userspace

The userspace I/O driver needs a real-time capable userspace. Therefore, the mainline Linux kernel 3.4.9 was patched to the real-time kernel 3.4.9-rt17. Following this, the patched kernel and the userspace I/O driver uio_netx are compiled and installed on a desktop PC with Ubuntu 10.04 Lucid Lynx. The RT_Preempt patch is used to patch the Linux kernel for real-time. This guarantees a real-time-capable userspace since tasks with a higher priority can preempt tasks with lower priority, regardless if the preempting task runs in userspace or kernel space [22].

This approach is not compatible with the standard repository of LinuxCNC [20] because the standard repository uses the real-time application interface (RTAI)[2] to guarantee real-time without a real-time-capable userspace. The chosen repository[4] of LinuxCNC contains EMC-RT-Preempt and the shared memory interface. EMC-RT-Preempt is a variant of LinuxCNC which is compatible with RT_Preempt[3] and runs without RTAI. The shared memory interface realizes the communication between LinuxCNC and a third party software, which is outlined in the next sections.

4.2 Interfaces

The solution is a self-developed application written in C. This application is called shared memory interface to real-time Ethernet (SHM-RTE). As shown in Fig. 1, SHM-RTE uses the shared memory interface as a connection to LinuxCNC and handles data e.g. set values and actual values. The dual-port memory is its interface to the Sercos III network. SHM-RTE uses the dual-port memory to handle the transfer of the set values to the Sercos III slaves and the transfer of the actual values generated by the Sercos III slaves to LinuxCNC. For this purpose, the CIFX-API is used. The data transfer from the dual-port memory to the Sercos III

[1] http://git.linuxcnc.org/gitweb

[2] http://www.rtai.org

[3] https://www.osadl.org/Realtime-Linux.projects-realtime-linux.0.html

slaves and vice versa is executed by the Sercos III protocol stack.

Figure 2 represents this approach in the Open systems interconnection model with its communication layers. The Sercos III protocol stack is involved from the application layer (7) to the data link layer (2). The typical transport layer protocols like the transmission control protocol (TCP) or user datagram protocol (UDP) and the typical network layer protocol internet protocol (IP) are not in use. Furthermore, the data link layer is modified to guarantee the data transfer in real-time. For this purpose, LinuxCNC and SHM-RTE work only in the upper layers. LinuxCNC generates the set values and SHM-RTE provides these. The CIFX-API enables SHM-RTE to access the dual-port memory. Figures 3, 4, and 5 show the functionality of SHM-RTE as a flow chart. The application SHM-RTE is synchronized with LinuxCNC and the Sercos III cycle. For this purpose, all applications work in real-time.

4.3 Real-time capability of SHM-RTE

As shown in Fig 3, the real-time capability of SHM-RTE has to be established. For this purpose, the scheduling policy FIFO and the highest real-time priority 99 are set for SHM-RTE according to listing 1, so that SHM-RTE is capable to run in real-time [14]. After this, the CIFX 50-RE is initialized with the chosen configuration.

```
/* shm-rte.c */
#include <sched.h>
#define RTPRIO 99
#define SCHEDULER SCHED_FIFO
struct sched_param param_me;
param_me.sched_priority=RTPRIO;
sched_setscheduler (0, SCHEDULER, &
    param_me);
```

Listing 1 Define scheduling policy and real time priority [14]

5 Transfer of set values

Today, the CNC is controlled by an NC program based on G-code. This G-code can be generated manually or with a computer-aided manufacturing (CAM) software from a drawing, created with a computer-aided design (CAD) software. The CNC processes this G-code, generates drive-specific set values and signals for other actors, and considers signals from sensors. This section describes transferring generated set values to the actors [15]. In one direction, the application SHM-RTE copies the data from the shared memory (SHM) into the dual-port memory (DPM) to provide data for the Sercos III master. In the other direction, the application SHM-RTE copies the data from the dual-port memory into the shared memory to provide the data from the slaves for LinuxCNC. This is shown in Figs. 4 to 5. Subsequent in this paper we refer to an engraving machine with cartesian coordinates (x, y, and z axis). But there is no reason to restrict the approach to this machine type.

5.1 From LinuxCNC to SHM-RTE via shared memory

LinuxCNC is modularized and built from several components. These components are pieces of software with different functions. For example, it can be a hardware driver or an external program linked to the hardware abstraction layer (HAL) e.g. motion, which accepts motion commands. The components are interconnected in the HAL and are called HAL components. In the HAL of LinuxCNC, HAL pins enable the interconnection of HAL components through HAL signals. So, in addition to the machine configuration, a HAL configuration is needed. The HAL configuration defines the components to be loaded and the interconnection between them [16]. The HAL configuration file is called SharedMemory.hal [2] and comes with the chosen repository [4]. It loads the component shm_interface [1] and connects the HAL signals Xpos, Ypos, and Zpos with this component. On the other side, these signals are connected with the HAL component motion across its HAL pins axis.0.motor-pos-cmd, axis.1.motor-pos-cmd, and axis.2.motor-pos-cmd, so that the set values for the x, y, and z axis are available at the shared memory interface. This is shown in Fig. 6. In addition to this, Fig. 6 denotes with dashed arrows that data are also transferable from SHM-RTE into LinuxCNC, e.g., to the graphical user interface (GUI).

To this point, the configuration is independent from the connected slaves. The next description is hardware specific and shows how to transfer the set values to evaluation boards. The evaluation boards are used later in the section measuring system. For each hardware module, SHM-RTE needs a hardware-specific structure with members which match to the data. For the evaluation boards, the structure is called NXIO50RE and contains the member LED_out of the data type float. In SHM-RTE, three variables are defined with the data type of the structure NXIO50RE. One variable for each module: nxio50reCL, nxio50reCN, and nxio50reCR. The members of the three variables are connected with the members from shmif_data. The variable shmif_data of the type shmif_data_t is described in [3]. Its members float_out_x, float_out_y, and float_out_z contain the set values for the x, y, and z axis, generated by LinuxCNC. Now, the set values are exported from LinuxCNC. They are available in SHM-RTE. In the next step, the data are transported to the Sercos III slaves.

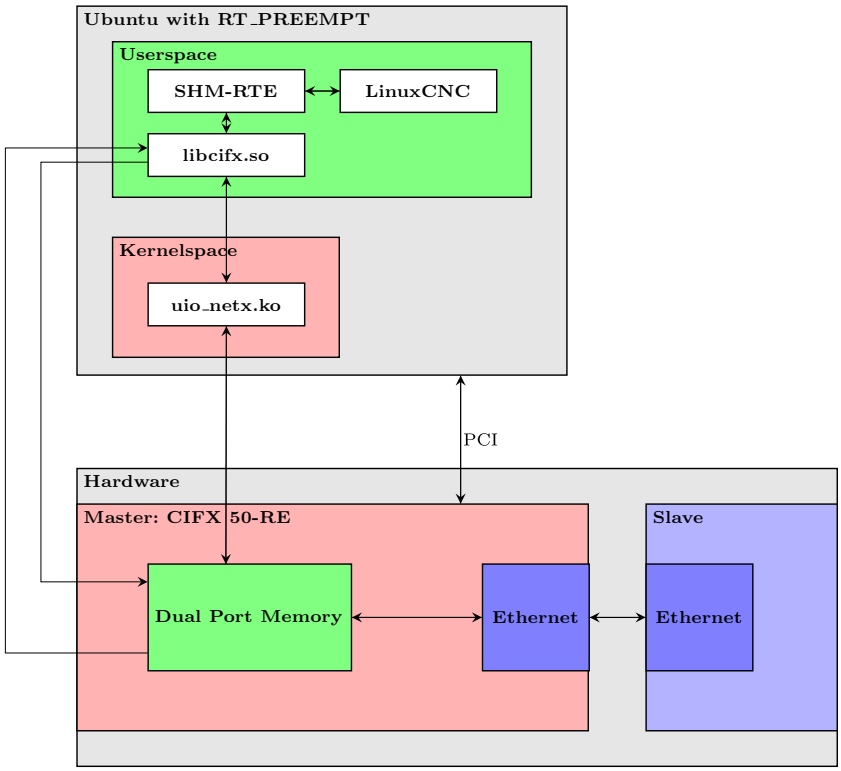

Fig. 1 Data transfer between LinuxCNC and a slave [10]

5.2 From SHM-RTE to the slave via dual-port memory

To enable the transfer of data, the driver for the CIFX 50-RE has been initialized. See detailed description in [8]. If the connection has been successfully established, the cyclic part of the application is able to start the data transfer. According to the example from [8], memory pointers are used to copy process data images. The array pabDPMMemory points to the input data image while pabDPMMemory_OUT points to the output data image. The arrays are mapped to the hardware-specific variables. For example, the content of nxio50reCL.LED_out is copied to pabDPMMemory_OUT [4]. Inside the Sercos III network, all these output data are handled as parameter S-0-1502 while the input data are handled as parameter S-0-1503 (Fig. 7).

Layer 7 Application Layer	LinuxCNC SHM-RTE Sercos III
Layer 6 Presentation Layer	
Layer 5 Session Layer	
Layer 4 Transport Layer	Sercos III
Layer 3 Network Layer	
Layer 2 Data Link Layer	Modified Ethernet
Layer 1 Physical Layer	Ethernet 802.3 PHY

Fig. 2 Open systems interconnection (OSI) model

6 Measuring system

The efficiency of this approach is verified by a small measuring system. LinuxCNC produces set values for an engraving process with three axes. SHM-RTE handles the transfer of this data to three Sercos III slaves. For this purpose, three evaluation boards of the type NXIO 50-RE are configured as Sercos III slaves according to [11].

According to [13], these boards are interconnected with the netAnalyzer PC card NANL-C500-RE and the CIFX 50-RE via cat5e patch cables in a physical line topology as shown in Fig. 8. The boards are connected to each other by Ethernet connection boards NXIO 50-RE\CA. In Fig. 8, the cat5e patch cables are drawn as arrows. The CIFX 50-RE is configured as Sercos III master with the field device tool (FDT)-based frame application SYCON.net[4] according to [9]. The timing configuration is shown in Table 1. As specified in [24], t_{Scyc} is the communication cycle time, t_1 is the

[4] http://www.hilscher.com/en/products/product-groups/software/configurator/syconnet

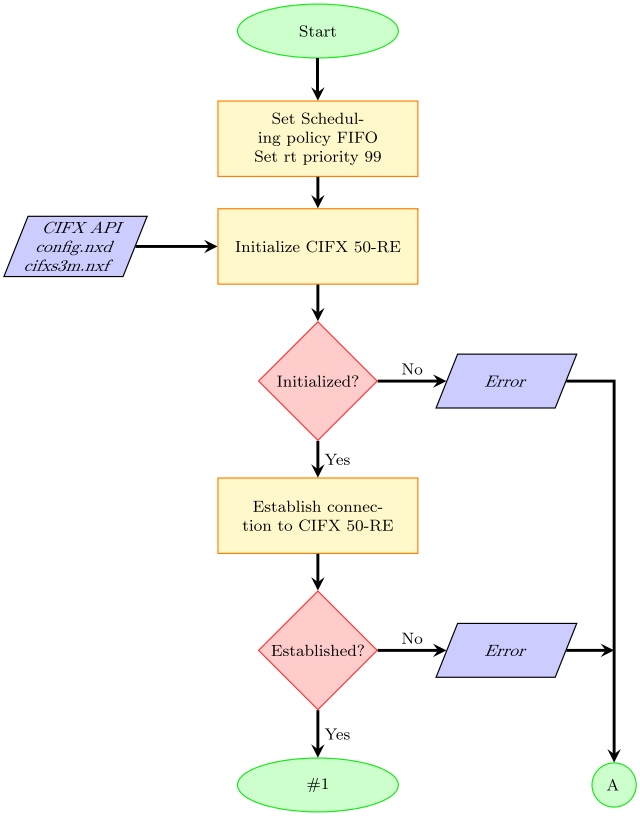

Fig. 3 Flow chart of SHM-RTE part 1

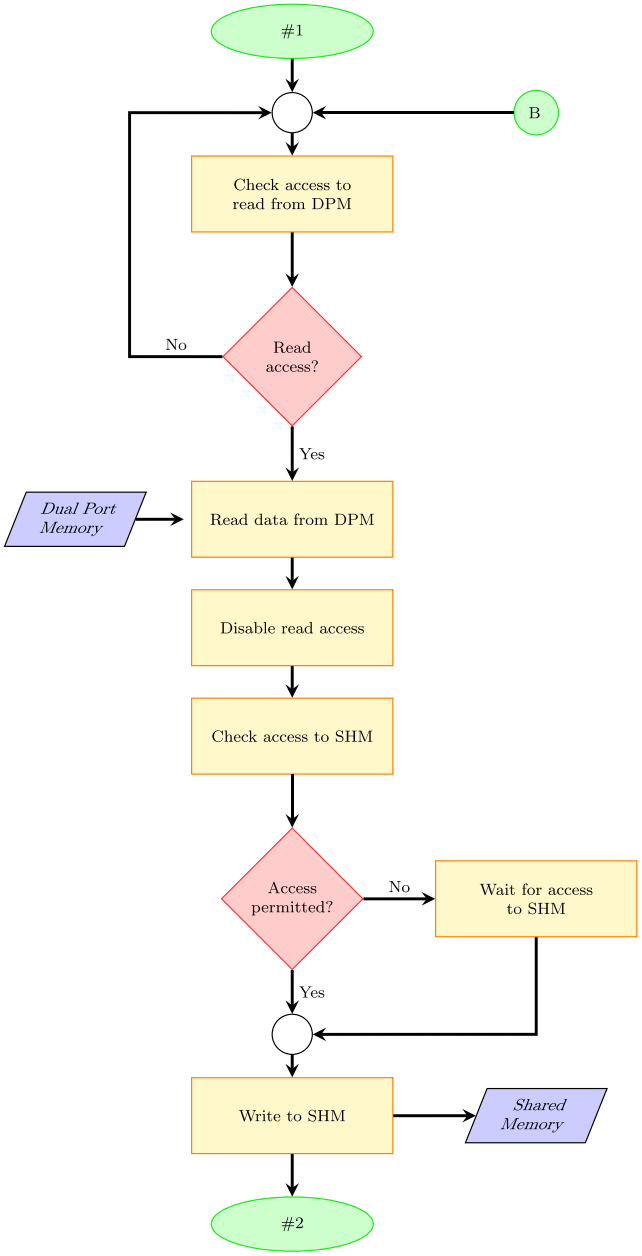

Fig. 4 Flow chart of SHM-RTE part 2

AT transmission starting time, t_4 is the feedback acquisition capture point, and t_5 is the minimum feedback processing time, whereat t_5 is determined by the slave. If $t_4 \leq t_1 - t_5$, the feedback values of the slave are transmitted in the same communication cycle. This affords a dynamic response.

All Sercos III telegrams are sent out by the master, passing through the NANL-C500-RE before they reach the slaves. The telegrams are logged and visualized by the netAnalyzer software in diagrams. For the purpose of illustration, the set values for x, y, and z axis are mapped on the 32 LEDs of each board.

7 Result

The analysis of the measurement is done according to [12]. Six hundred thousand four hundred four telegrams were analyzed. The chosen cycle time was 1 ms. Two diagrams of this analysis are shown in Figs. 9 and 10. They show logarithmically scaled master data telegrams over the cycle time. In Fig. 9, only the master's jitter is shown. The master's jitter is caused by the CIFX 50-RE. The maximum jitter is 10 ns and the standard deviation is 4 ns. In Fig. 10, both the master's and slave's jitter are shown. The slave's jitter is caused by the three slaves. Due to the full duplex characteristic of the built Ethernet network, we have a logical ring topology with a physical line topology. This is the reason for more than three jitter peaks. The average cycle time is 1,000,003 ns. The jitter is between −160 and +170 ns. The standard deviation is 26 ns or 0.003 %. According to [23], a jitter <1µs is required for motion control applications. The maximal occurred jitter is 170 ns. Therefore, this demonstrates a reliable integration of Sercos III into LinuxCNC.

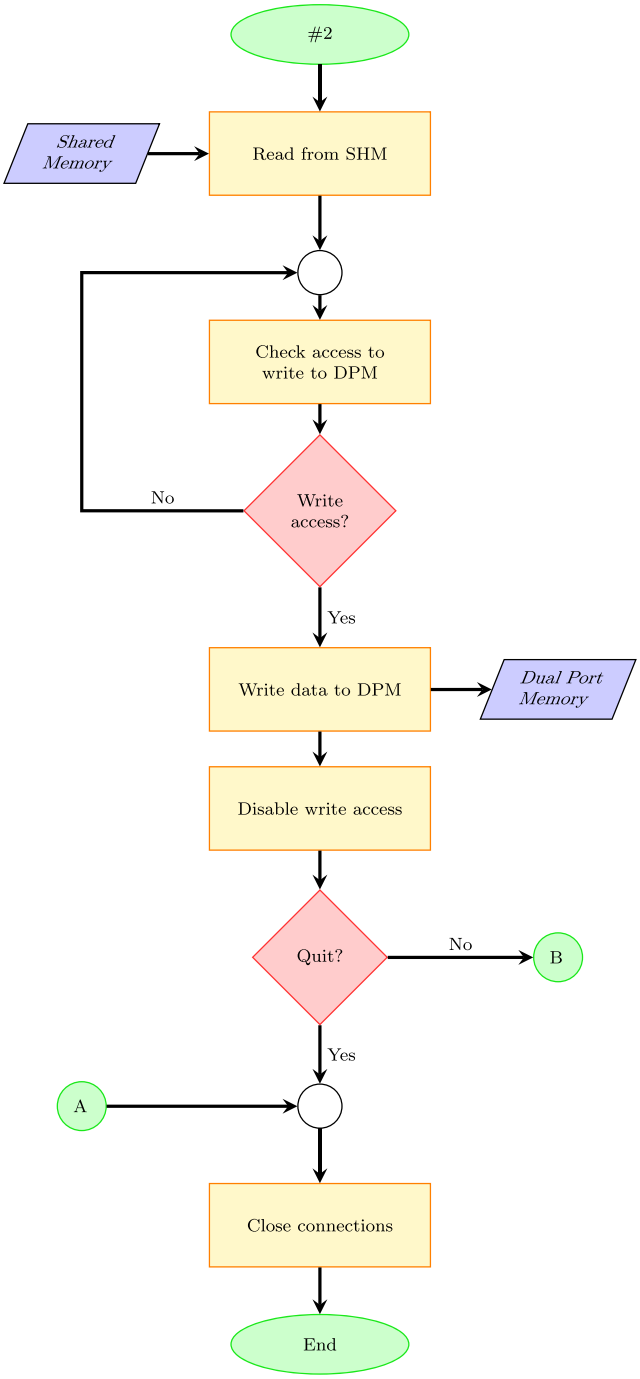

Fig. 5 Flow chart of SHM-RTE part 3

8 Integration of industrial grade hardware

The evaluation boards are not suitable for industrial environments. Therefore, we define the integration of industrial grade hardware into LinuxCNC using the example of a servo drive via Sercos III.

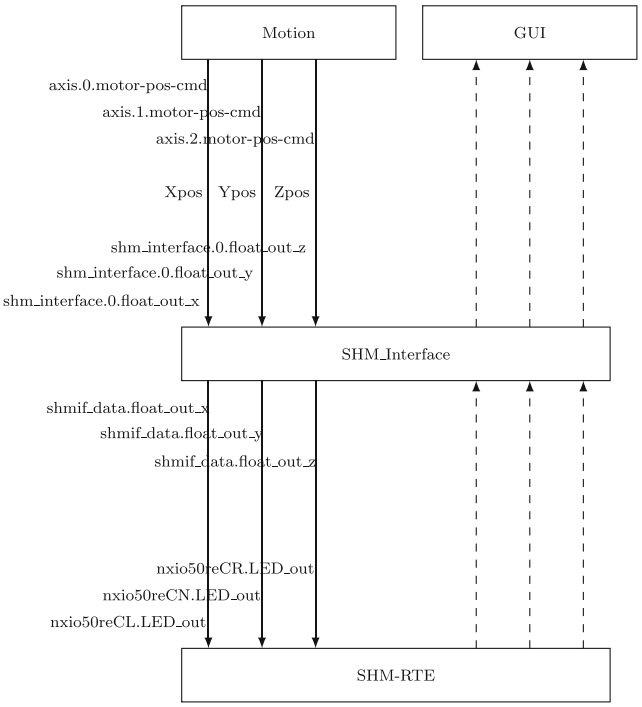

Fig. 6 Data transfer between LinuxCNC and SHM-RTE

8.1 Hardware configuration

For this purpose, the servo drive CSB01.1C-S3-ENS-NNN-NN-S-NN-FW from Bosch Rexroth AG is configured for position control. In addition, the digital I/O interface of the servo drive is implemented. The configuration of the servo drive is done according to [5] for position control with cyclic command value input. The following configurations are done for the Sercos III master:

Consumer data

- master control word (IDN S-0-0134)
- digital outputs (IDN P-0-1372)
- position command value (IDN S-0-0047)

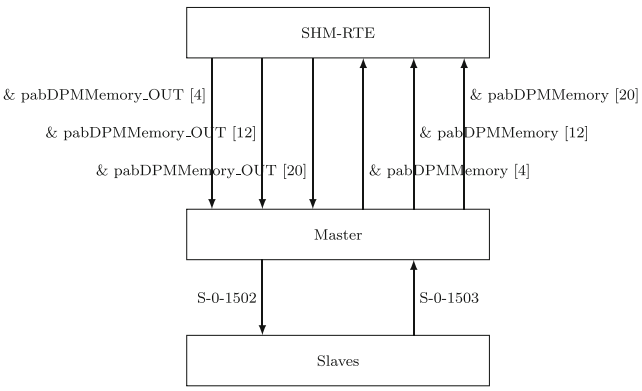

Fig. 7 Data transfer between SHM-RTE and a slave

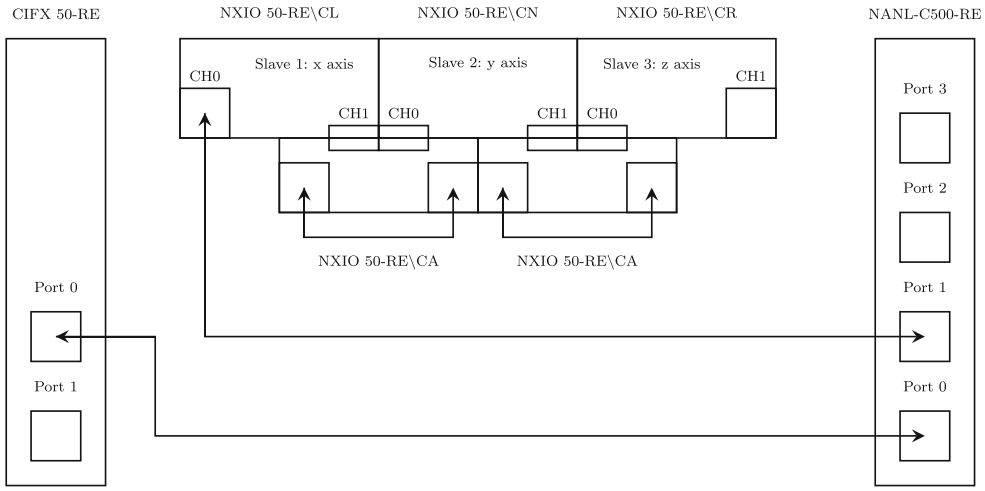

Fig. 8 System of measurement [12]

Producer data

- status word (IDN S-0-0135)
- digital inputs (IDN P-0-1371)
- position feedback value (IDN S-0-0051)

The master control word is required to enable the drive's control. For detailed information, see [5]. This servo drive provides digital outputs and digital inputs. The use is optional. The same also applies for the position feedback value because the control loop is closed by the servo drive. Therefore, the focus is on the master control word.

8.2 HAL configuration

The manual [16] gives a detailed introduction in HAL configuration and was used to connect all HAL pins. In SHM-RTE, the data for master control word, status word, digital outputs, and digital inputs are handled as integers. The position command value (also called set value) and position feedback value (also called actual value) are handled as floats. The consumer data are led out from LinuxCNC as HAL signals, while the producer data from the servo drive are transferred into LinuxCNC. For that purpose, the HAL pins are connected via HAL Signals.

These HAL pins and HAL signals can be of different types, such as bit, integer or float. Components like the decomposer comp_u32_xbit and the composer comp_xbit_u32 are loaded to convert values from type integer to several bits or to convert several bits to a value of type integer. This is shown in Fig. 11. The master control word consists of 16 bits (x_ms_0 to x_ms_15). In Fig. 11, the master control word is set bitwise in the GUI of LinuxCNC. These bits are composed into two integers and are transmitted to the Sercos III master. The Sercos III master and the Sercos III slave handle this data as IDN S-0-0134.

Table 1 Timing configuration for Sercos III master

IDN	Timing	Value in µs
S-0-1002	t_{Scyc}	1000
S-0-1006	t_1	500
S-0-1007	t_4	250
S-0-1017[0]	t_6	0
S-0-1017[1]	t_7	1

8.3 Visualization of signals

Values like the master control word can be used via G-code, manually set or visualized in LinuxCNC, if the HAL pins are well-defined and interconnected. Signals can be visualized in LinuxCNC via the Python Virtual Control Panel (PyVCP). This panel allows the definition of graphical elements in the extensible markup language (XML). A HAL signal like x_ms_0 has to be connected to a graphical element like pyvcp.x_ms_0. In this case, the graphical element pyvcp.x_ms_0 is a check button. Such a check button is defined with the tag checkbutton. For detailed information, see [18]. In this manner, the 16 bit master control word is set with 16 check buttons. Furthermore, HAL signals can be used via G-code.

```
1 ( sample_program.nc )
2 ( Turn digital output on P0=x_out_1 )
3 M62 P0
4 ( Turn digital output off P0=x_out_1 )
5 M63 P0
```

Listing 2 Using digital outputs with G-code

8.4 Using signals with G-code

Listing 2 shows an abstract from an NC program. It demonstrates how to set a HAL signal via G-code. In line 3, the command M62 P0 sets the digital output P0. In line 5, the

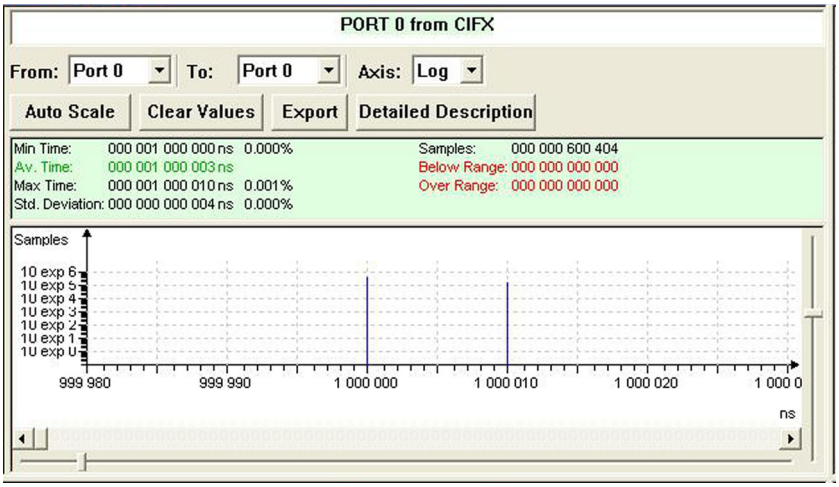

Fig. 9 Analysis of master's jitter for cycle time

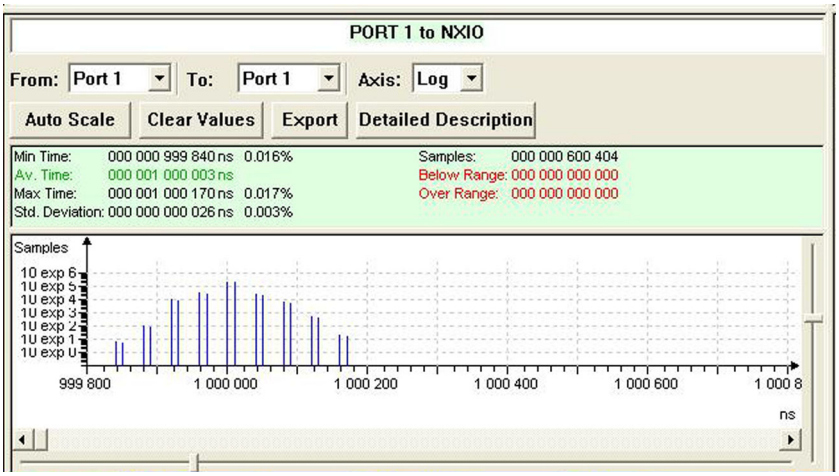

Fig. 10 Analysis of master's and slave's jitter for cycle time

command M63 P0 resets the digital output. The digital output P0 is a HAL pin of the HAL component motion and is called motion.digital-out-00. As shown in Fig. 12, P0 is connected to the drive's digital outputs (IDN P-0-1372) via HAL signal x_out_1. In addition to the digital outputs, the digital inputs of the drive are also connected to motion. The use of digital inputs and outputs is described in [19].

The integration of the servo drive CSB01.1C-S3-ENS-NNN-NN-S-NN-FW with its digital I/O was successfully realized at the Institut für Maschinen- und Anlagenbau[5]. This was verified by 24-h tests.

9 Other protocols than Sercos III

With the presented approach, we have described the integration of real-time Ethernet in LinuxCNC using the example of Sercos III. In addition to that, the CIFX 50-RE supports further protocols to integrate further hardware into LinuxCNC using other protocols than Sercos III. In this case, the CIFX 50-RE needs only the master protocol stack of the desired protocol and a new configuration with the FDT-based frame application SYCON.net. This is specified in [13]. As described before, the application SHM-RTE needs a hardware-specific structure with members which match to the data. The CIFX 50-RE supports the master protocol stack for Sercos III, EtherCAT, Ethernet/IP and PROFINET.

Subsequent tests at the Institut für Maschinen- und Anlagenbau have demonstrated that the approach is compatible with further protocols and further hardware. For example, the digital and analog I/O modules EL1014, EL2004, EL3064, and EL4004 from Beckhoff Automation were integrated into LinuxCNC using the EtherCAT coupler EK1100[6]. Similar to the set values and actual values for the servo drive, the analog outputs and analog inputs were connected with LinuxCNC. The protocols Ethernet/IP and

[5] http://mabi.hs-emden-leer.de

[6] http://www.beckhoff.de/default.asp?ethercat/ek1100.htm

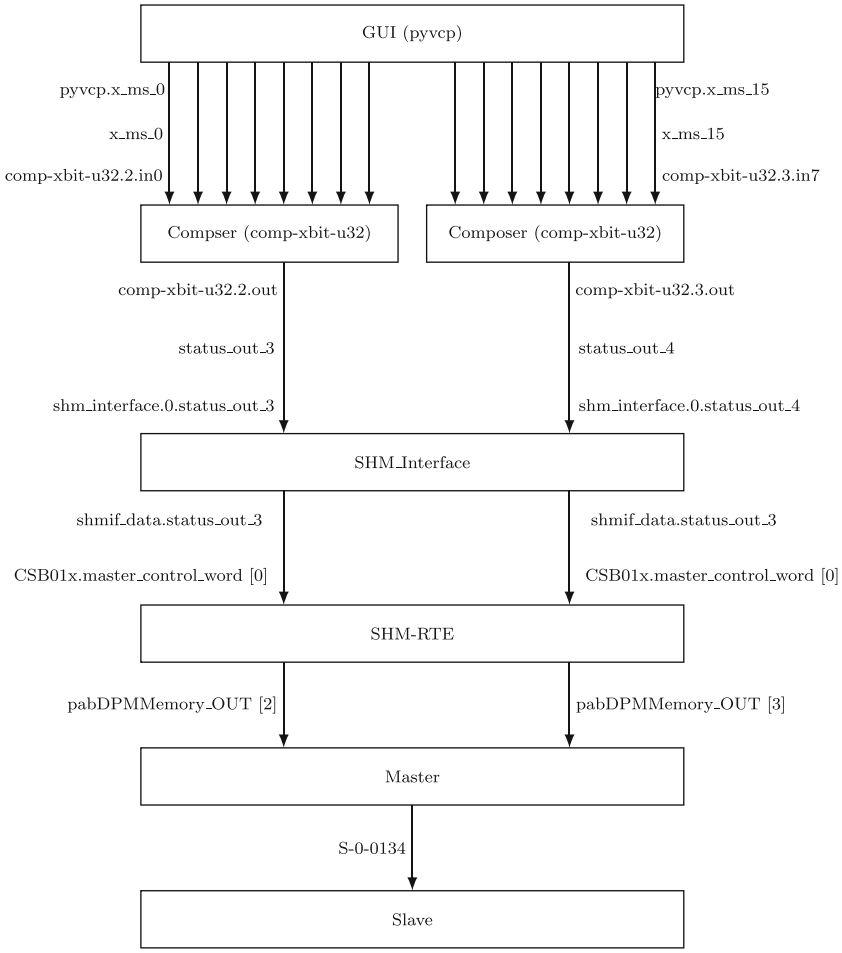

Fig. 11 Data transfer to set the master control word

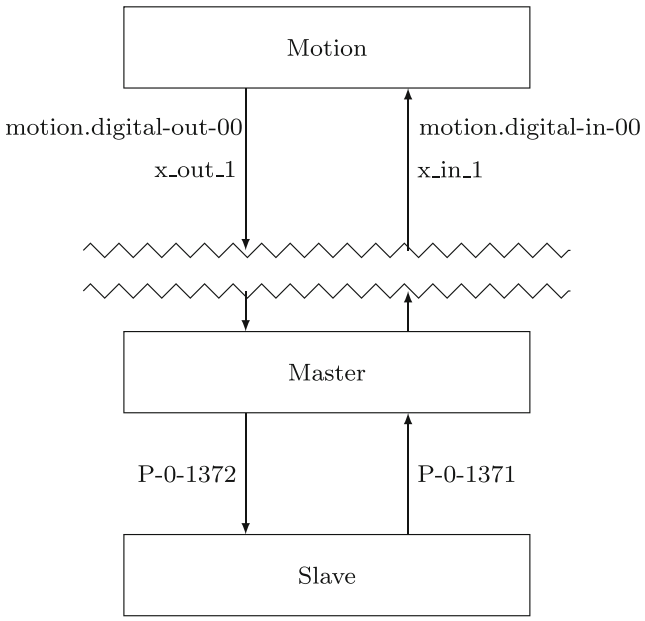

Fig. 12 Data transfer to set the master control word

PROFINET were also integrated into LinuxCNC using the described evaluation boards NXIO 50-RE. These subsequent tests are not outlined in this paper. Along first tests, the same results are expected.

10 Other controls than LinuxCNC

Although this manuscript has a focus on LinuxCNC, there is no reason to restrict the approach to LinuxCNC. Other control systems with a shared memory interface are usable if they run with a Linux-based operating system with a real-time capable userspace.

11 Conclusion

The presented approach enables the integration of the real-time Ethernet protocols Sercos III, EtherCAT, Ethernet/IP and PROFINET into LinuxCNC via the PC card CIFX 50-RE. This integration was verified with the evaluation boards NXIO 50-RE.

In addition to the evaluation boards, it was possible to integrate industrial grade hardware like a servo drive and I/O modules.

The analysis of the measurement with the NANL-C500-RE and the 24-h tests with a real servo drive show that the integration is suitable to control dynamic servo drives with LinuxCNC using real-time Ethernet.

Acknowledgments The authors would like to thank Dipl.-Ing. Thomas Peetz, M.Sc. for providing technical support that made this work possible.

References

1. Abel M (2011) Hal component: shared memory interface. https://gitorious.org/emc-rt-preempt/emc-rt-preempt/source/dbc5f99db839a1913828863828e55cb6220fc954:src/hal/drivers/shm_interface.c
2. Abel M (2011) Hal configuration for shared memory interface. https://gitorious.org/emc-rt-preempt/emc-rt-preempt/source/dbc5f99db839a1913828863828e55cb6220fc954:configs/SharedMemory/SharedMemory.hal
3. Abel M (2011) Specification of shared memory interface. https://gitorious.org/emc-rt-preempt/emc-rt-preempt/source/dbc5f99db839a1913828863828e55cb6220fc954:src/hal/drivers/shm_interface.h
4. Abel M (2012) Repository to track the adoption of emc to rt_preempt. https://gitorious.org/emc-rt-preempt/emc-rt-preempt/commits/linuxcnc_rt_shm_20120501_cleaned
5. Bosch Rexroth AG (2011) Rexroth IndraDrive firmware for drive controllers MPH,- MPB-, MPD,- MPC-08: R911332643
6. Ethernet Powerlink Standardization Group (2013) Industrial Ethernet facts: the 5 major technologies
7. Hehenberger P (2011) Computerunterstützte Fertigung. Springer-Verlag, Berlin Heidelberg
8. Hilscher Gesellschaft für Systemautomation mbH (2011) Driver Manual cifx Device Driver: Windows 2000/xp/vista/7 v1.1.x.x: DOC060701DRV21EN
9. Hilscher Gesellschaft für Systemautomation mbH (2011) Operating Instruction Manual DTM for Hilscher Sercos III Master Devices: Configuration of Hilscher Master Devices: DOC090301OI05EN
10. Hilscher Gesellschaft für Systemautomation mbH (2012) Driver Manual cifx Device Driver: Linux (Kernel 2.6.x / 3.3.x) v1.0.1.0: DOC090201DRV05EN
11. Hilscher Gesellschaft für Systemautomation mbH (2012) User Manual NXIO 50-RE-Board Hardware Description: DOC090101UM06EN
12. Hilscher Gesellschaft für Systemautomation mbH (2012) User Manual Real-Time Ethernet Kit: Analysis Examples: DOC081202UM04EN
13. Hilscher Gesellschaft für Systemautomation mbH (2012) User Manual Real-Time Ethernet Kit: Installation Operation and Configuration: DOC081105UM04EN
14. Kerrisk M (2012) The linux man-pages project: Linux man-pages online: Alphabetic list of all pages. http://man7.org/linux/man-pages/dir_all_alphabetic.html
15. Kief HB, Roschiwal HA (2013) CNC-Handbuch 2013/2014. Carl Hanser Verlag, München
16. LinuxCNC.org (2014) HAL Manual V2.5, 2014-10-29. http://linuxcnc.org/docs/2.5/pdf/LinuxCNC_HAL_Manual.pdf
17. LinuxCNC.org (2013) Ethercat realtime hal driver. http://www.wiki.linuxcnc.org/cgi-bin/wiki.pl?EtherCatDriver
18. LinuxCNC.org (2014) Integrator Manual V2.5, 2014-10-29. http://linuxcnc.org/docs/2.5/pdf/LinuxCNC_Integrator_Manual.pdf
19. LinuxCNC.org (2014) User Manual V2.5, 2014-10-29. http://linuxcnc.org/docs/2.5/pdf/LinuxCNC_User_Manual.pdf
20. LinuxCNC.org (2014) Download linuxcnc. http://linuxcnc.org/index.php/english/download
21. LinuxCNC.org (2014) Linuxcnc: Software for realtime control. http://www.linuxcnc.org/
22. OSADL eG (2014) Realtime linux road map. https://www.osadl.org/?id=99
23. Sercos International e.V. (2014) Plug and play - Sercos, the automation bus. http://sercos.com/literature/pdf/sercos3_en.pdf
24. DIN Deutsches Institut für Normung e.V. (2013) Industrial communication networks - Fieldbus specifications - Part4-19: Data-link layer protocol specification - Type 19 elements (IEC 61158-4-19:2010). Beuth Verlag, Berlin