Integration of Digimatic Measuring Tool into LinuxCNC Controlled Milling Machine by Using MODBUS

Lukas Pusman, Kamil Kosturik

***Abstract* — This paper describes a custom-designed measuring system based on Digimatic indicator and its integration into LinuxCNC control software. The system is composed of a digital indicator equipped with Digimatic output and a MODBUS slave module which is connected to the CNC machine's PC via MODBUS. Although, the described measuring system was primarily intended for the evaluation of mechanical parameters of a custom-designed CNC milling machine, it can be used more widely.**

***Keywords* — Classic Ladder, Computer Numeric Control (CNC), Digimatic, Digital absolute indicator, Graphical User Interface (GUI), Hardware Abstraction Layer (HAL), Linux CNC, Milling machine, MODBUS, Python Virtual Control Panels (pyVCP).**

I. Introduction

In the beginning of 2014, a project which was concerned with design and production of a custom CNC milling machine was finished and there arose a need to evaluate somehow the mechanical parameters of the machine, especially the angles among the axes.

The machine is a three-axis, relatively small CNC mill, which is able to machine various plastic and aluminum-alloy parts in sizes up to 300 x 200 x 80 mm. All the three axes are using linear shafts and ball-bearing slide units for their movement. Each shaft is supported on both sides with an aluminum block, which is attached to the base with screws. These blocks can be shimmed a little bit up by using thin steal plates and this way, one axis can be set precisely perpendicular to the other one. To be able to measure and calculate which block needs to be shimmed and how much, a special measuring system based on a digital indicator was designed. This measuring system, its construction and software integration are subjects of this paper and will be described in the following text.

This research was supported by Ministry of Education, Youth and Sports of the Czech Republic under Research Project SGS-2012-019 FEL UWB as well as by grant CZ.1.05./2.1.00/03.0094 Regional Innovation Centre for Electrical Engineering (RICE) of Research and Development for Innovations.

Corresponding Author Lukas Pusman is with the Faculty of Electrical Engineering, University of West Bohemia, Univerzitni 26, 306 14 Plzen, Czech Republic (phone: +420 377634227; e-mail: pusman@kae.zcu.cz).

Kamil Kosturik is with the Faculty of Electrical Engineering, University of West Bohemia, Univerzitni 26, 306 14 Plzen, Czech Republic (phone: +420 377634227; e-mail: kosturik@kae.zcu.cz).

II. Measuring System Overview

The whole measuring system is shown in the block diagram in Fig 1.

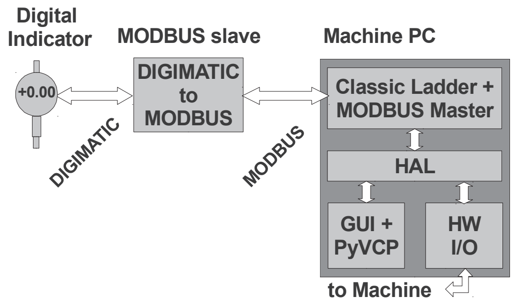

Fig. 1. Block diagram of the measuring system.

The digital absolute indicator is a product of Mitutoyo Corporation and the full model name is ABSOLUTE Digimatic Indicator ID-S Series 543. It has resolution 0.01 mm and is equipped with Digimatic output. By using this output the indicator is connected to Digimatic side of the MODBUS slave module.

The MODBUS slave module is a custom-designed circuit board and will be more described later. The module continuously translates requests from MODBUS to Digimatic and sends back data from the measuring tool. Since MODBUS is using the serial port as the physical layer[1], the serial to USB converter is build-in and the module is connected to the PC via USB cable.

The MODBUS master is implemented in the same PC which controls the milling machine. This solution brings a possibility to use the measured value directly for machine control and its automation. For example, after a button is pressed the spindle with attached indicator can precisely touch the surface under the indicator's tip and/or the measured value can be shown in the GUI.

III. MODBUS Slave Module

The MODBUS slave module is made on a single circuit board and is based on Freescale HCS08 MCU, MC9S08DZ16 in 32-pin package (more information in [1]). As the converter from the serial port to USB, FTDI FT232RL IC (more information in [2]) is installed. The whole functionality of the module is realized at the

[1] MODBUS is positioned at level 7 of ISO/OSI model therefore it can be implemented on various physical layers in general [3].

software level. There is implemented the MODBUS protocol according to the MODBUS standard [3] and also one module which maintains the communication via Digimatic.

A. MODBUS Implementation

MODBUS is the well-known request/reply type protocol widely used in industrial applications and devices. It allows one master and up to 254 slaves connected to one bus. The communication is always initiated by the master which uses the ID of each slave to address him a message. There are used function codes to specify which service is requested or replied in MODBUS [3].

Since MODBUS uses many function codes for many special purposes which are implemented device by device, it is ineffective to implement all of them. The indicator measuring tool returns only one value which can be stored in one 16-bit read-only register and therefore function *"Read Input Registers"* is enough to implement for accessing this value. Next functions which are suitable to implement are some diagnostic ones, e.g. *"Read Exception Status"* or *"Report Slave ID"*. All the implemented functions are listed in Table 1.

TABLE 1: LIST OF IMPLEMENTED MODBUS FUNCTIONS.

	Code	*Function name*	*Data*
Req.	0x04	*Read Input Registers*	Address [2B]: 0x0001, Count [2B]: 0x0001
Rep.	0x04		Count [1B]: 0x02, Value [2B]: 0x0000 to 0xFFFF
Req.	0x07	*Read Exception Status*	None
Rep.	0x07		Value [1B]: 0x00 to 0xFF
Req.	0x11	*Report Slave ID*	None
Rep.	0x11		Count [1B]: 0x02, ID [1B]: 0x01, Running indicator [1B]: 0x00 or 0xFF

As it was mentioned above, the MODBUS slave module is connected to the master (PC) by using the serial port. For this case it is available specification [4] which describes how to implement MODBUS over the serial line. There are two modes available per the specification, RTU (Remote Terminal Unit) and ASCII. According to the selected mode the specific setup of the serial port has to be done. In case of RTU, the port setup has to be 8 data bits plus one even-parity bit plus one stop bit (11-bit frame). Unlikely, Classic Ladder software which provides the MODBUS master on the PC side supports only no-parity setup (10-bit frame). This fact requires changing the serial port setup regardless what specification [4] says.

B. Digimatic Implementation

SPC (Digimatic) Input/Output is a Mitutoyo proprietary communication interface which is used for data gathering from various measurement tools like calipers or indicators [5]. Digimatic uses a special connector on the device side and a 2.54 mm pitch compatible one on the side of a data processor or hub. The connector on the non-device side has 10 pins but only 4 of them are used in this case. Pins DATA and CK (clock) are open-collector outputs from the device. Pin REQ (request) is internally pulled up and is used as the input for the transmission initiation. The last remaining pin is GND, the ground. Figure 2 shows how the signal timing looks like.

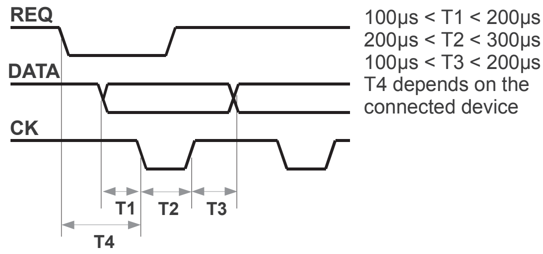

Fig. 2. Digimatic timing chart [5].

The output frame consists of 13 digits and each digit consists of 4 bits. Digits are transmitted from D1 to D13 and bits in each of them are transmitted from LSB (2^0) to MSB (2^3) [5]. The data format is fully explained in Figure 3.

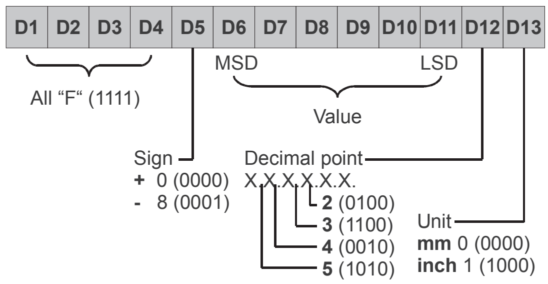

Fig. 3. Digimatic data format [5].

IV. LINUXCNC AND SOFTWARE INTEGRATION

The above described milling machine is controlled from the PC by using LinuxCNC. This software is installed together with modified LTS (Long Term Support) version of Linux operation system Ubuntu with modified real-time Kernel. LinuxCNC is a descendent of NIST Enhanced Machine Controller (EMC) and it is designated for real-time control applications. It is distributed under GNU GPLv2 (General Public License version 2). LinuxCNC contains a lot of tools and utilities like user interfaces (GUI), G-code programming utilities, Python Virtual Control Panels (PyVCP), Classic Ladder Programmable Logic Controller (PLC), Hardware Abstraction Layer (HAL), hardware drivers etc. [6].

In the described application GUI called Axis is used for controlling the machine. This pre-programmed GUI is suitable for the three or four axes milling machines and allows all the basic functions like homing, emergency stop, G-code loading etc. By using PyVCP the GUI can be extended for some additional control panels for different purposes e.g. automated lubrication, automated tool change and others. In case of the described application, PyVCP additional panel is added to show the measured value from the connected indicator.

A. Classic Ladder and MODBUS Master

Classic Ladder is a kind of a graphical programming language (originally used for industrial PLCs programming) which allows drawing diagrams composed of rungs leading from the left (input) to the right (output) side which are performing some logical function. These multiple rungs are placed horizontally between the left and the right rail and therefore wording “ladder” is used [7].

In addition, Classic Ladder tool allows creating the MODBUS master which is able to poll periodically one or more slaves. The value of indicator is periodically obtained from MODBUS and after it is processed by Classic Ladder logic it is passed to HAL as the output variables. The Classic Ladder program is shown in Figure 4 and the three rungs are visible. The most bottom one performs only an assignment from one variable called *“level”* to other one called *“indicator”*. The remaining two rungs perform comparison if *“level”* is bigger than 100 (actually 1.00 mm) or if *“level”* is equal to zero. The output of these comparisons is finally passed to the HAL as variables *“%Q0”* and *“%Q1”* as well as variable “*indicator”*.

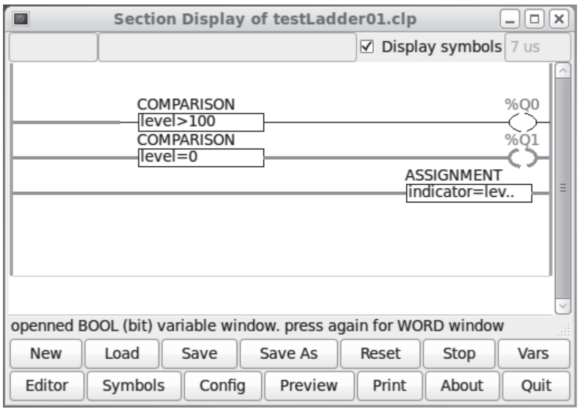

Fig. 4. Classic Ladder program.

B. Hardware Abstraction Layer (HAL) and PyVCP

At the highest level, HAL can be described as a tool which allows connecting some parts together without the exact knowledge how these parts work inside. Therefore there are two main terms used in HAL, “components” and “interconnections”. The further description of HAL is over the scope of this paper. More information is available in [8].

The HAL setup was already done by using a graphical wizard during the CNC milling machine project, therefore an additional manually written HAL configurations have to be added to allow the connection of the described measuring system. The following extract of the HAL description file shows how the connections between Classic Ladder and HAL core are programmed.

```
# Load components - Classic Ladder (CL)
loadrt classicladder_rt
# Add functions - CL refresh at servo thread
addf classicladder.0.refresh servo-thread
# Connect CL variables to HAL signals
net indicator_press <= classicladder.0.out-00
net indicator_free <= classicladder.0.out-01
net indicator_value <= classicladder.0.s32out-00
# Load user components - CL configuration file
loadusr classicladder --modmaster testLadder01.clp
```

The second code extract shows how the signals from HAL are connected to the PyVCP custom panel which finally shows the indicator value.

```
# Connect HAL signals to PyVCP custom panel
net indicator_value => pyvcp.indicator_level_num
net indicator_press => pyvcp.indicator_press_py
net indicator_free => pyvcp.indicator_free_py
# Load user components - PyVCP xml description
loadusr testPanel01.xml
```

In the last line of the above code example, the xml description of PyVCP custom panel is loaded. It is shown below how the xml might look like. There is the LED component defined, connected to *“indicator_press_py”* signal from HAL.

```
<led>
   <halpin>"indicator_press_py"</halpin>
   <size>20</size>
   <on_color>"red"</on_color>
   <off_color>"green"</off_color>
</led>
```

V. Conclusion

The main aim of the project described in this paper was the composition of a custom measuring system, which is able to measure the spindle position and pass this value directly to LinuxCNC software which is controlling the CNC milling machine. This aim was fulfilled.

The measuring system, respectively the way how to pass some external inputs to LinuxCNC and HAL has been verified and it brings new possibilities and applications.

The next approach will be focused on the further automation of the CNC machine. The next aim is the implementation of automated surface touching by using the described measuring system.

References

[1] Freescale Semiconductor (2008) *MC9S08DZ16 Data Sheet*, Denver, Colorado: Freescale Semiconductor Literature Distribution Center. Available at: http://www.freescale.com.

[2] Future Technology Devices International Ltd. (2006) FT232R USB UART I.C., Glasgow, United Kingdom: Future Technology Devices International Ltd.. Available at: http://www.ftdichip.com.

[3] Modbus Organisation (2012) *MODBUS Application Protocol Specification*, : Modbus.org. Available at: http://www.modbus.org.

[4] Modbus Organisation (2006) *MODBUS over Serial Line Specification and Implementation Guide*, : Modbus.org. Available at: http://www.modbus.org.

[5] Mitutoyo Corporation (2002) *ID-H0530/0560: Digimatic Indicator User's Manual*, : Mitutoyo. Available at: http://www.mitutoyo.co.jp.

[6] *LinuxCNC.org*. 2014. LinuxCNC.org. [ONLINE] Available at: http://www.linuxcnc.org. [Accessed 28 September 2014]

[7] *Classicladder Introduction*. 2014. Classicladder Introduction. [ONLINE] Available at: http://www.linuxcnc.org/docs/html/ladder/ladder_intro.html. [Accessed 28 September 2014]

[8] *HAL Introduction*. 2014. HAL Introduction. [ONLINE] Available at: http://www.linuxcnc.org/docs/html/hal/intro.html. [Accessed 28 September 2014]