# Chapter 15
# Linux Based Real-Time Control over Industrial Networks

**Peter Fodrek**
*Slovak University of Technology, Slovakia*

**Martin Foltin**
*Slovak University of Technology, Slovakia*

**Michal Blaho**
*Slovak University of Technology, Slovakia*

**Matúš Lichý**
*Invensys Systems, s.r.o., Slovakia*

**Tomáš Murgaš**
*RT Systems, s.r.o., Slovakia*

## ABSTRACT

*This chapter deals with industrial networked control systems. These systems are the most modern means of industrial control. The current approach contains several problems. Mostly discussed problem in the control theory is a network-caused variable transport delay problem. On the software point of view there is problem of the system latency, which limits time period of the discrete control. State of art research in hardware part of the system shows that latency source and transport delay variations are caused by unpredictable hardware behavior as well. Another problem is the control system security. This was the case of the Mexico gulf oil rig disaster in 2010, as well as Iran nuclear program three months later the same year, caused by worm attack at its centrifuges. Therein lies the question; how can security issues be avoided? The authors demonstrate how via using GNU/Linux, which solves timing correctness by their project of the plasma cutting machine design.*

## INTRODUCTION

In this chapter there we are to demonstrate how we use industrial networks in our main cooperation with mechanical engineering company Prvá zváračská, a.s. It uses CAN bus industrial network for periodic communication. It uses RTAI as GNU//Linux extension for hard Real-Time.

In the next part of this chapter there we are to demonstrate theoretical proof of problem of minimal sampling period of 2miliseconds found in the first part. Then we are to describe possible solution using EtherCAT Ethernet based network, too. For this purpose we are to describe test labo-

DOI: 10.4018/978-1-4666-0294-6.ch015

ratory equipment to test latencies of the industrial Ethernet network as well.

In the last part of this chapter we are to describe principles of new framework for industrial networked control using GNU//Linux. This framework was designed using knowledge form research of previous parts. This may help others to design networked control system using GNU/Linux operating systems with hard real-time executive. Before such framework there was Linux based control design too hard to design although security bug for Windows based control systems are known very well. Hardness of the Linux control systems raising initial design costs and therefore Linux based control systems are used often or almost exclusive only in mission critical control such as military and aerospace systems.

## INTRODUCTION TO PLASMA CUTTING MACHINE

Our team have been developed plasma cutting machine control system for a commercial engineering company Prvá zváračská, a.s. This part of chapter describes main principles of this project especially of the software design. There is main problem to break mechanical engineers to permanently change specifications that degrades original lasagna-ravioli code design in direction to produce only spaghetti code. Controlled machine contains mechanical portal with one motor on the X-axis but two pinion-rack mechanism with torsion shaft. Y- and Z-axis have also one motor each. Movement of each axis is provided by Permanent magnet synchronous motor with corresponding drive with CANOpen interface, There are also binary and analogue input/output modules with CAN interface for technology control. Another parts are two CANOpen operator panel with 3 line display and industrial computer with additional two port CAN interface.

## HARDWARE CONFIGURATION

In this section we are to describe used hardware and the lowest level of the software which is in lasagna code equal to hardware drivers. This is critical for hard real-time capabilities of the system. It is because when you are not successful in design of this layer then you are not able to achieve hard real-time capabilities. As Mr. Paul McKenney recognized in his 11$^{th}$ Real-Time Linux workshop in Dresden paper we are now to compensate not determinism in the hardware system especially in CPUs. He described that in 1979 then was 1 hardware problem of hard real-time per 29 software problems but in 2009 there were one software problem of hard real-time per 39 hardware problems. So there is critical to design hardware to software interaction layer. It is critical to patch hardware problems at lowest possible level not to cause too much bugs in the software.

### CANbus Interface in PC

In the prototype we used Adlink PCI-7841 because of cost and well documented Philips (now NXP) SJA1000 controller. We were mentioned that SJA1000 is capable to achieve bandwidth of 1Mb/s, but on this card it is impossible. We were used this card because of existence of previous RTLinuxFree based project and own driver derived form CanFestival project's code for hardware which is not yet included in the code. In that code there was an remark that driver does not work on some revisions of Adlink PCI-7841. Code does not worked with our card, so we made some extraction and customization of the driver's code as it is not driver but only kernel module with exported symbols which are function names. When we used only BasicCAN functions then driver worked, but when used another features of the SJA1000, system crashed. BasicCAN is mode of compatibility with Philips PCA82C200 which is pin and electrically compatible predecessor of the SJA1000. There is possibility to use SJA1000