REAL-TIME CRANE CONTROL VIA PC

Jakub Arm
Doctoral Degree Programme (1), FEEC BUT
E-mail: xarmja00@stud.feec.vutbr.cz

Supervised by: Zdeněk Bradáč
E-mail: bradac@feec.vutbr.cz

Abstract: In this paper, creation of SCADA system, which controls and monitors model of gantry crane in real time, is described. This crane will be controlled via process station communicating with PC via Modbus TCP protocol. Moreover, HTTP server will be created to monitor crane state over intranet. On this work, the difference between real-time and fast execution will be described and it will be shown on the built system that the fast execution system can behave like a real-time system under specific conditions.

Keywords: Real-time, Modbus TCP, PC control, deadline, HTTP

1 INTRODUCTION

The world of control automation can be divided into process control and machine control. In machine control, real-time executing of controllers is more important in sense of lesser time which the controller has to react on events in. One of the standard machine controllers is PLC (Programmable Logic Controller) which can react via digital and analog outputs on incoming events, e.g. digital and analog inputs. PLC can share information among other PLCs and HMI (Human Machine Interface) to control and monitor state of machine via specific or standard communication bus, e.g. Profibus DP or Modbus. It can be included into SCADA (Supervisory Control And Data Acquisition) system via OPC (OLE for Process Control) drivers. PLC executes its program, written in language according to IEC 61131-3, cyclic real-time. Another way to control some machine is using some computer and some process station. In this case, programmer is not limited by the computing power of the PLC but of computing power of PC which is mostly grater. However, programmer has to make some prevention to meet real-time capabilities of such system. In this contribution, the system on regular PC will be created to show that fast execution of program can lead in specific circumstances to achieve real-time capabilities but real-time capabilities are not guaranteed by underlying operating system.

2 APPLICATION CREATION

In this chapter, solution of implementation of exemplary SCADA system into PC is presented. This PC is connected via ethernet cable with process station (see Figure 1) which communicates using standard Modbus TCP protocol. Inputs and outputs of the process station Modicon Momentum TSX is wired to sensors and actuators of gantry crane model, i.e. linear motor for gantry moving, absolute sensor of gantry position, grabbing magnet solenoid, etc.

2.1 MODBUS PROTOCOL

Modbus protocol is standard industry protocol. It has several forms, i.e. RTU/ASCII via RS232 or RS485 physical line and TCP via ethernet. Modbus can be located in Application layer in ISO/OSI reference communication model. It uses underlying layer to exchange messages and provides inter-

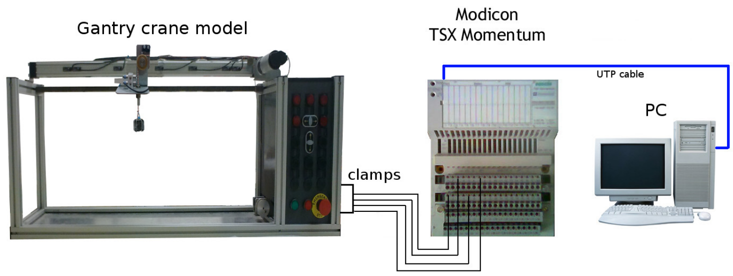

Figure 1: Schema of the system

face to send commands in Modbus way, i.e. read coil, write coil, read input register or write holding register.

2.2 Application concept

To communicate with process station, it has to be created driver in form of library which implements Modbus client protocol via ethernet TCP. This library will be created in C language to achieve the fastest execution and linked into DLL (Dynamic-Link Library).

The user application, which is part of the SCADA system, will be created in C# language. This language has been chosen because of good performance together with good graphic programming capabilities via GDI (Graphics Device Interface) and capabilities of binding DLL via PInvoke mechanism.

Application (see Figure 2) consists of two threads. First takes care of rendering graphic as a state of the model. Second has a concept of endless loop. At the beginning, some initialization has been done, then the thread is periodically executing program for get and set the state model IO (see Figure 3). So after every loop, the application has nearly current state of the model and can react on incoming events. This loop is periodically executed with period of 10 ms. This period has been chosen as a compromise between small period, which consumes to much computing power, and big period, which discards nearly real-time capabilities.

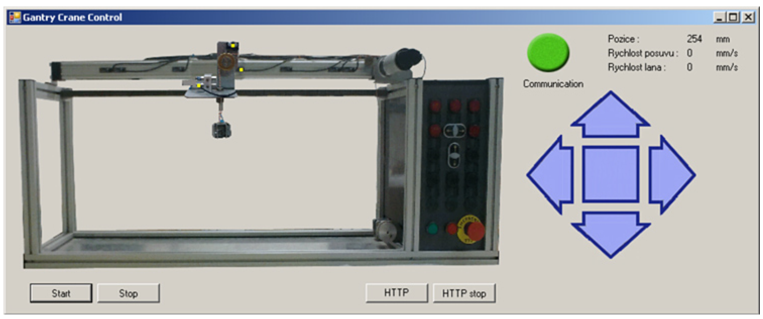

Figure 2: Control application [1]

The solenoid, which grabs things on the surface of the model, is moved by spinning reel in vertical direction and also by linear motor placed on top in horizontal direction alongside in vertical direction. To achieve movement only in horizontal direction, spinning reel has to rotate with speed which depends linear on speed of the linear motor. Conversion rate may be calculated when all raw values

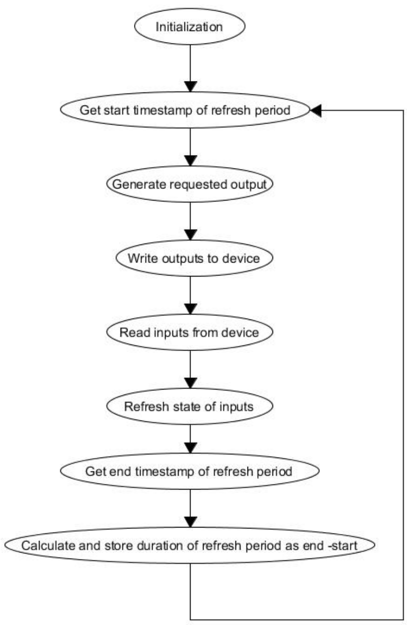

Figure 3: Diagram of control thread

of motor speed are standardized from word data type values into standard physical values of float or double data type according to formula (1) where v is standardized value and x raw analog value of 2 bytes size.

$$v[mm/s] = \frac{x - x_{min}}{x_{max} - x_{min}}(v_{max} - v_{min}) + v_{min} \tag{1}$$

Parallel to application, as another thread, there is running HTTP (HyperText Transfer Protocol) service which provides state of the model on intranet. This service is sharing resources with application, reading state of model. When some HTTP request comes, the service will parse the incoming request and its parameters via. According to the HTTP method and parameters, it will send response in form of HTTP web page. So reading of the state of model and parsing arguments is processing on server side. Server can simultaneously serve more than one client in separate threads.

3 REAL-TIME CAPABILITIES

There are some definitions of real-time system. Firstly, *the system has to meet deadlines*. Secondly, *every information has to be in the right time on the right place* [4]. Thirdly, *a real-time system is one whose logical correctness is based on both the correctness of the outputs and their timeliness* [3]. Hard real-time system are system which every information after deadline in utility function is potentially harmful. In soft real-time systems, this information can be used. However, its utility value is falling.

3.1 SOURCES OF NON-DETERMINISM

Determinism in real-time concept is ability of function or system to do defined operations in defined time. In this system, the main sources non-determinism are :

- communication via ethernet using CSMA/CD (Carrier Sense Multiple Access with Collision Detection) algorithm is considered to be non-deterministic

- non-determinism kernel of underlying operating system and its scheduler

This severity of these impacts is decreased using these arrangements :

- Only two devices are connected via ethernet and type of communication is defined as master/slave. There is still ethernet switch which causes some additional latency but can cause more deterministic behaviour of the communication system cause of switching among connected segments. This additional latency grows with network complexity. But, in small network, using switch can lead to higher communication speed beacuse ethernet switch eliminates impact of collisions during CSMA/CD algorithm.
- The priority of control thread is increased, all unused applications and services are closed to get the most computing power. But the non-deterministic kernel is still used. So the control over scheduler and kernel functions is still insufficient.

To achieve full real-time capabilities it is necessary to use deterministic kernel which means deterministic scheduler and preemptible kernel functions. It is also necessary to eliminate non-deterministic page fault by locking memory or stack overflow error. It has to be also eliminated impact of priority inversion effect, deadlocks and livelocks [4].

3.2 Seeming of real-time

The duration of refresh thread execution has been measured for specific count of cycles as $(12,34 \pm 1,13)$ *ms*. It has been measured on Intel Celeron 1,6 GHz with 1 GB RAM and integrated graphic card. There is some negligible non-deterministic delay till control thread is being executed again. But overall period, which this system is able to react on events after, does not exceed in this particular observation the value of 20,00 ms. It can be seen by human that application controls and monitors the model real-time. Even in this particular measurement can be said that this system is real-time with deadline value of 20,00 ms.

Refresh rate of web page model state is 1 s which is lesser considered as real-time system by human. This is one problem of conception of term real-time that real-time capabilities are compared with human perception. But, if it is declared that deadline is e.g. 5 s and if it is guaranteed that every refresh time does not exceed e.g. 2 s, it will have to be said according to definitions that this is real-time system.

3.3 Real-time vs fast execution

So the application seems to be able to consider as a real-time operating application. But, nor underlying system neither the application can not fully guaranteed that the application thread will be always running in same amount of time even that the thread will be executed at all. When these impacts are not eliminated, the system can be only called fast executing not real-time. Even, when the system meets real-time deadlines. Because it does not fulfil determinism condition.

3.4 Soft real-time

If the refresh time is not guaranteed, the system might be considered as soft real-time where deadline overruns are tolerable, but not desired. Because the system reacts in finite time in case of non-failure executing. Thus majority of systems, which reacts in finite time, may be considered as soft real-time. It only depends on shape of the utilization function.

Another comparison of soft to hard real-time systems can also be *how much harm they cause in case of later reaction than deadline*. This might be better conception for human perception. With this conception is connected to term QoS (Quality-of-Service).

4 CONCLUSION

The application, which controls and monitors gantry crane model via process station, has been created. It has been shown that PC can control machines via interfaces directly. It has been pointed out the difference between fast executing and real-time executing of control device on the created system. It has been observed that non-real-time fast executing system can behave like a real-time system under specific conditions. It has been described what can cause non-deterministic behaviour and how deterministic capabilities can be achieved.

ACKNOWLEDGEMENT

This paper was made possible by grant No. FEKT-S-14-2429 - "The research of new control methods, measurement procedures and intelligent instruments in automation", and the related financial assistance was provided from the internal science fund of Brno University of Technology.

REFERENCES

[1] Arm, J.: Gantry crane control via Modbus TCP. Brno, 2011, FEEC BUT.

[2] Petters, S.: Real-Time Systems. Australia, NICTA 2007/2008, UNSW.

[3] Laplante, P., Ovaska, S.: Real-Time Systems Design and Analysis, 4th Edition. Wiley 2011, ISBN 978-0-470-76864-8.

[4] Kucera, P.: Real-Time Operating Systems IX. 2012, FEEC BUT.

[5] Ganssle, J.: Real Time Programming. July, 1998.

[6] Burns, A.: Scheduling Hard Real-Time Systems: a review. May, 1991, Software Engineering Journal.

[7] Podila, P.: HTTP: The Protocol Every Web Developer Must Know. April, 2013.

[8] Penumuchu, Ch.: Simple Real-time Operating System. Trafford Publishing, 2007, ISBN 978-1-4251-1782-5.