UDC: 681.51:004.415.2.031.43
Professional paper

Software management and algorithms for real-time control systems

Nderim Zeqiri[1], Ferit Idrizi[1]

[1] *University of Tetova, Faculty of Applied Sciences, Mechatronics, n.n. Tetove,*
nderim.zeqiri@unite.edu.mk, ferit.idrizi @unite.edu.mk

Abstract

In this paper, it is mainly analyzed the aspect of finding the relevant diagrams, and the aspect of setting efficient programs for management in a given process. On the basis to control process, firstly is necessary to create opportunities for access objects through control systems that achieve a stable connection, thus fulfilling the demand for access to different resources. In the paper, above all, is asked to construct the relevant diagrams that meet the technological requirement. To fulfill this requirement, is needed a verified and respective algorithm. Also, this composition requires the deployment of an application (software) that meets these algorithmic requirements, based on the request from the practical problem. First, the computers that running external process must operate with speed that is compatible to control the process. The characteristic of the computer systems is that the sequence of execution of the program cannot be defined beforehand. External signals may interrupt or change the pre-defined program sequence, because this message is different from each other of the successive execution. The real-time system resources must be used effectively, and to be constraints in time and this criterion must be respected. Based on this requirement, is necessary to establish interconnection between requirements and the special programming methods, which increase the diapason of the time, increase the stability of the process, because the final results depend not only from the logical values that have been generated in any time, but also depend from the exactly from the time when these results are produced.

Keywords: Process, software, algorithm, real-time, control, system.

Introduction

The computers control for any physical processes has a need to process different problems. The characteristic of process computer systems is that the sequence of execution of the program cannot be defined in advance. External signals may interrupt or change the pre-defined program sequence so that this message is different for each new execution. Then, real-time computer system resources must be used effectively and time constraints should be respected. Because of this, special programming methods are needed. Unforeseen execution order makes testing much more difficult on real-time systems than conventional systems.
Therefore, from here comes the question: Can conventional programming solve real-time problems?

To answer this question, we will consider checking a plastic extruder that will serve as an example for the typical computer problem of the process. The computer should simultaneously control the temperature and sequence operations for the clip movement. If programming is by conventional methods, they will soon come to terms and dominate the code structuring problems. RT (real time) systems are systems in which the correctness of the system behavior depends on the logical results of the computations, and on the physical time when these results are produced.

Computer system for Real-time systems, heat control and impulse range

The computer system should simultaneously control the temperature and movement of the clip. The temperature is measured by measuring the continuous signal from the sensor, and the clip position is detected by the impulse counter and presented with binary signals when one of the last positions is reached. The computer does not have internal clocks and must, accordingly, present the time with an internal numerator controlled, Fig 1.

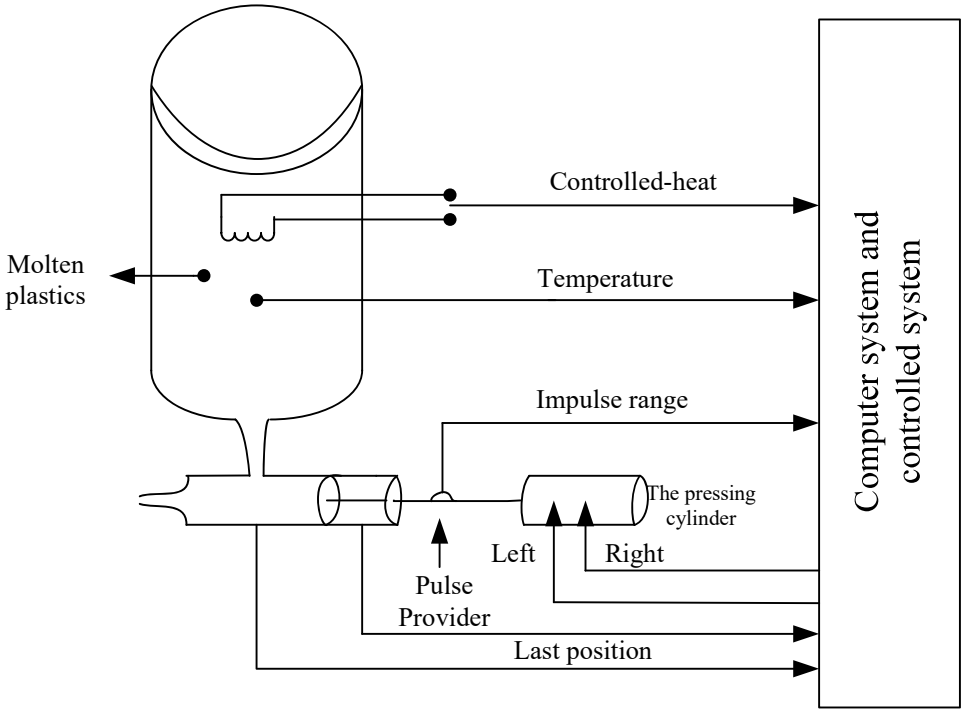

Figure 1. Plastic extruder

The container contains liquid plastic material, the temperature of which must be tolerated at a certain limit. The commanding computer constantly reads the current temperature and calculates the necessary amount of heat to maintain the desired value. The heat is provided with electrical elements (resistances) whose voltage is continuously controlled, or through relays (the relay can be mechanical or electronic-solid state) the closing time of the relay corresponds to the heat produced.

The bottom part of the extruder consists of a clip that compresses an amount of molten plastic through the small opening. When the clip is on the right side of the last position, the cylinder is filled with plastic. Then, the clip moves to the left to drop the desired amount of plastic. The

position of the clip is viewed by the impulse counter, which provides a specified number of impulses in millimeter displacement so that the number of impulses corresponds to the volume of the cast plastic material. Clip movement should be terminated when the required number of impulses arriving from a transceiver transmitter – encoder.
First, we will examine the extruder with control through conventional programming. The extruder command routine is shown through the diagram in Fig. 2.

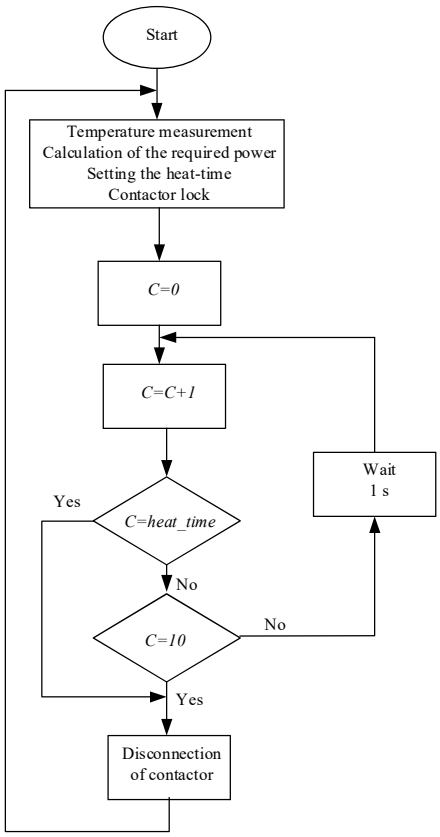

Figure 2. Temperature Diagram - as a Real Time Control Algorithm

On the basis, of above algorithm, Fig.2, we can construct the principal ideas, to calculate, the time, when the contactor is active, or inactive; to works consequently, and to control, the materials temperature, and above all to switch the contactor, to work properly in real time, because another way of this reaction, can produce, non-prediction defect in the systems, or maybe will cause any other event which be risk for all the processes,

Table 1. Temperature measurement, required power, heat time and real time, reaction

N	Temperature measurement	Required Power	Heat time (C)	Contactor	On/Off (real time)
1	50	20	0	1	On
2	60	22	0.8	1	On
3	70	24	1.6	1	On
4	80	26	2.4	1	On
5	90	28	3.2	1	On
6	100	30	4	1	On
7	110	32	4.8	1	On
8	120	34	5.6	1	On
9	130	36	6.4	1	On
10	140	38	7.2	1	On
11	150	40	8	1	On
12	160	42	8.8	1	On
13	170	44	9.6	1	On
14	180	46	10.4	0	off
15	190	48	11.2	0	off
16	200	50	12	0	off
17	210	52	12.8	0	off
18	220	54	13.6	0	off
19	230	56	14.4	0	off
20	240	58	15.2	1	off
21	250	60	16	1	off
22	260	62	16.8	1	off
23	270	64	17.6	1	off
24	280	66	18.4	1	off
	…				

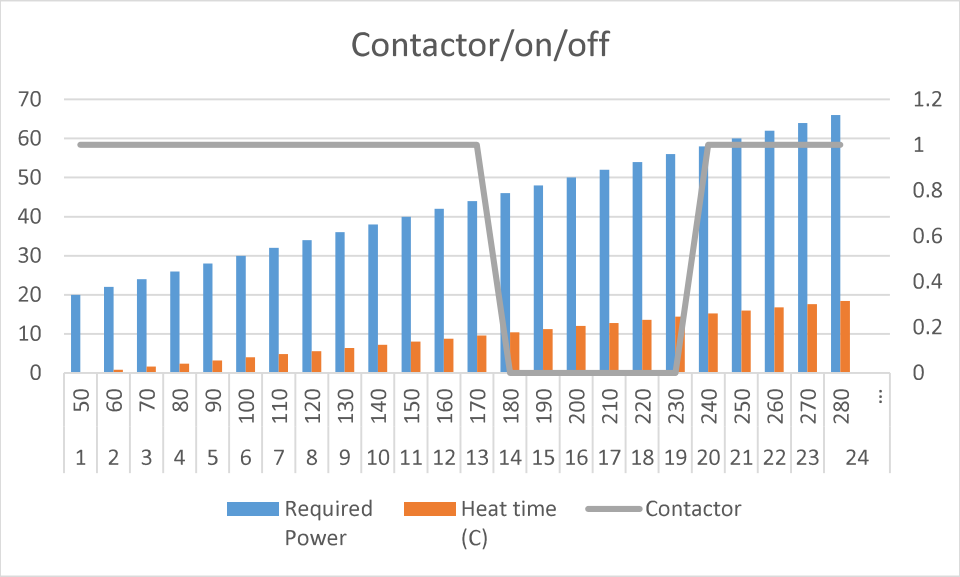

Chart 1. Graphical presentations, of the Heat-time deviation based on the above parameters

Checking the position and finding the corresponding algorithm for Real-time systems

The check sequence for the movement of the clip is shown in Fig 3. The computer begins to move to the right, looking at the sensor with the last position, stops a time until it is confirmed that the last position has been reached. As in the case of temperature control, the computer cannot do anything else while waiting in the contour (loop). Then, the clip moves to the right and the computer again tests the impulse from the encoder and counts until the puls_ref impulse number is reached. As it can be seen, in this process there is a loop of waiting at the arrival of the new impulse from the encoder. Each of the different tasks can be solved by regular sequential code. However, it is no longer trivial to combine both tasks.

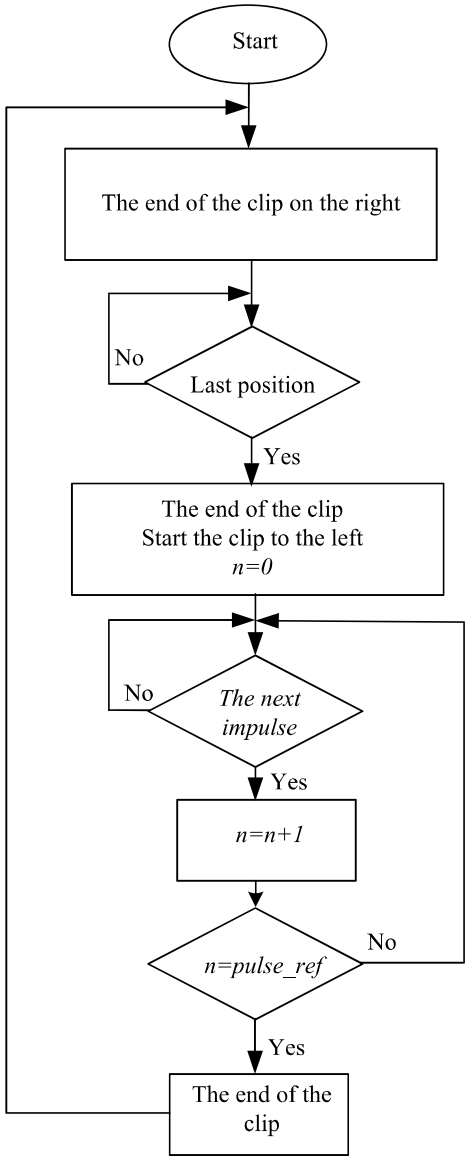

Figure 3. Position Control Diagram and Real-time System

Control system, and control with the use of interrupt

Control system, and interrupt control. The main difficulty in writing the control program for the extruder process lies in telling the computer that it is time to move from one job to another. In practice, the problem is solved by writing both tasks to be accomplished in the same machine. A duty to adjust the temperature and other tasks to control the motion of the clip. To inform the processor when it is time to accomplish one of the tasks or the other, is used, a signaling method called interrupt. With the character of the interplay, the waiting contours can be replaced with intercept reception. Which type of interrupt will be used depends on the task to be performed. The time signal is generated when the specified time is reached or when the time interval is exceeded. In this way, the central processor unit is released from the time control task. Similarly, the clip sequential control expects two interrupts, one for the last position sensor, and one for

each pulse entering the computer, Fig 4. This is accomplished by the wait_interrupt (x) command where x represents the channel where the signal for interrupt is expected. The wait_interrupt command suspends the next execution on the clip control command until an external signal is displayed for interact. Only then will the execution of the assignment be renewed again.

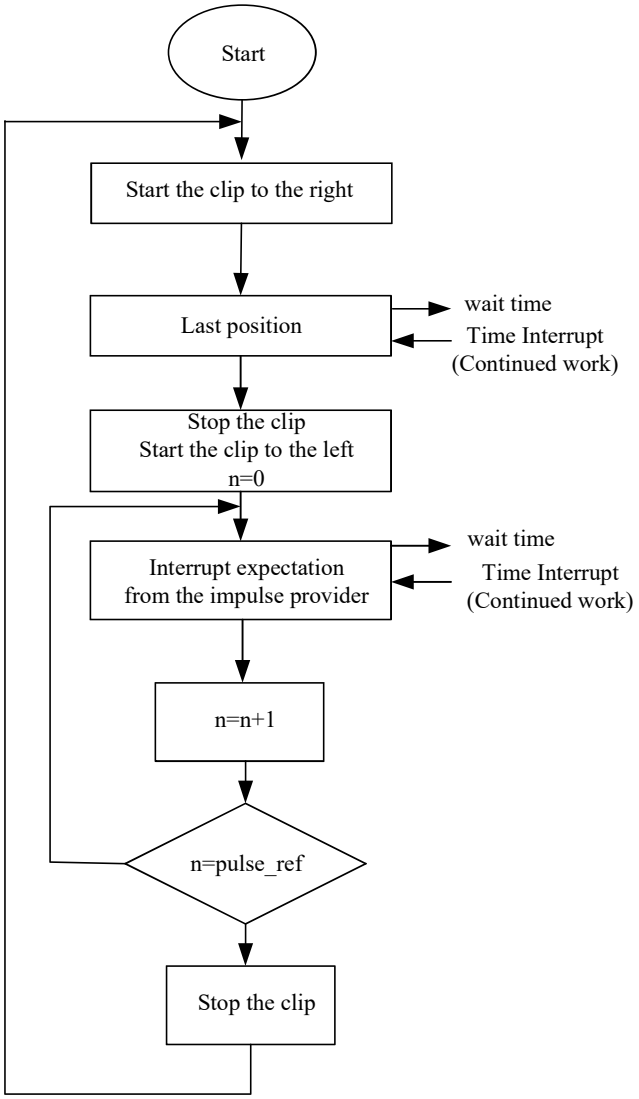

Figure 4. Use interruption for sequential position control of the clip Position-Control Diagram and Real-time System

Table 2. Sequential position control, real time, clip (left and right)

Nr	Start clip	Last position 2=left, 3=right	Stop clip (n=0)	Interrupt inspections (n=n+1)	N=pulse ref (Max=20) in the left
1	1	2	0	1	Left
2	1	2	1.5	2.5	Left
3	1	2	3	4	Left
4	1	2	4.5	5.5	Left
5	1	2	6	7	Left
6	1	2	7.5	8.5	Left
7	1	2	9	10	Left
8	1	2	10.5	11.5	Left
9	1	2	12	13	Left
10	1	2	13.5	14.5	Left
11	1	2	15	16	Left
12	1	2	16.5	17.5	Left
13	1	2	18	19	Left
14	1	3	19.5	20.5	Right
15	1	3	21	22	Right
16	1	3	22.5	23.5	Right
17	1	3	24	25	Right
18	1	3	25.5	26.5	Right
19	1	3	27	28	Right
..	..				

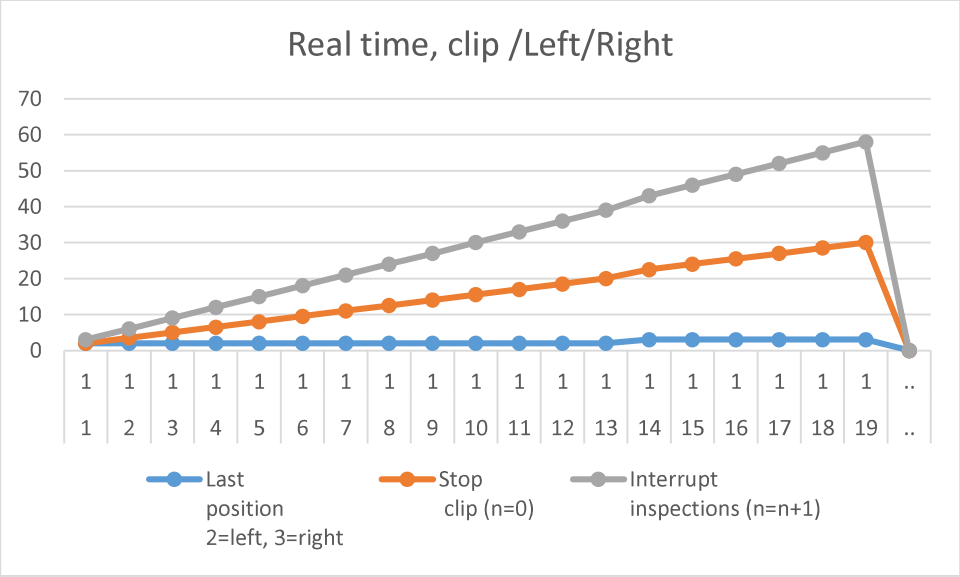

Chart 2. Real time presentations, clip (left or right) and dependence from the interrupt inspections

The process of creating software for controlled system and real time systems

The process of creating software for real-time control systems and systems. It is consisted of three phases: the choice of the algorithm, the placement of the diagram in the mainstream and writing the program with any of the programming languages. Give some information about these phases:

- The algorithm is a pool of equations and / or a range of operations that solve a problem. Since, it is an express procedure; it is independent of the computer. Algorithms that need to solve control problems in many cases are presented as statements or expressions.
- Diagram for dealing with a problem is the first step in transforming the algorithm into a computer program that enables the computer to solve the problem. The diagram shows how the algorithm will be executed systematically.
- Program. The common way of presenting such programs is the use of mnemonic languages and assemblers.

Conclusions

The process described in this paper shows some aspects that need to be considered in computer control systems. The concrete implementation of control diagrams, which make the system more stable in overall operation. The process that is commanded or controlled is only part of the problem; the rest is the commanding computer itself that controls the process. When designing the computer control system, it must always take into account its performance, which must be in accordance with the system requirements, which is controlled. In order to estimate demand capacity, many factors and parameters need to be addressed for the process computer, such as:

- Time scales (system dynamics, modeling, hardware requirements, software requirements, measurement accuracy, control actions),
- Measurement type (hardware, sensor, etc.).

Data processing software in the process control system includes sets of programs, procedures, rules, and documentation. The inclusive software band differs widely for each functional unit in particular, depending on the level of its hierarchy, its role in controlling the process, and its connection to other functional units.

References

[1]. H. Kopetz, “The temporal specification of interfaces in distributed real-time systems,” in Lecture Notes in Computer Science, Embedded Software. Heidelberg, Germany: Springer-Verlag, vol. 2211, pp. 223–236 (2001).

[2]. G. S. Virk, Digital Computer Control Systems, pp 122-135, Real-Time Computer Control Systems. Macmillan New Electronic Series. Palgrave, London. Online ISBN: 978-1-349-21490-7 (1991).

[3]. E. Closse, M. Poize, J. Pulou, J. Sifakis, P. Venier, D. Weil, and S. Yovine, “Taxys: A tool for the development and verification real-time embedded systems,” in Lecture Notes in Computer Science, Computer Aided Verification. Heidelberg, Germany: Springer-Verlag, vol. 2102, pp. 391–395 (2001).

[4]. N. Zeqiri, Managing a computer system and interface, performance and applications. Journal of Advanced Computer Science and Technology; Science Publishing Corporation, Greifswalder Platz, Bremen, Germany, pp: 23-27, ISSN: 2227-4332, (2016).

[5]. W. Yuan and K. Nahrstedt, Energy-efficient soft real-time CPU scheduling for mobile multimedia systems, In: ACM Symposium on Operating Systems Principles, pp. 149-163, (2003).

[6]. G. Chindris, M. Muresan, Deploying Simulink Models into System-On-Chip Structures. In: 29th International Spring Seminar on Electronics Technology, ISSE 2006, pp. 313–317 (2006).

[7]. J. Maki-Turja, M. Nolin, “Efficient implementation of tight response times for tasks with offsets,” Real-Time Systems, vol. 40, no. 1, pp. 77–116, (2008).

[8]. H. Kopetz and G. Grunsteidl, “TTP—A protocol for fault-tolerant real-time systems,” Computer, vol. 27, no. 1, pp. 14–23, (1994).

[9]. Cook, J.E. and Dage, J.A., Highly Reliable Upgrading of Components, Proceedings 21st International Conference on Software Engineering, Los Angeles, USA, (1999).

[10]. B. Douglass, Doing Hard Time, Developing Real-Time Systems with UML, Objects, Frameworks, and Patterns. Reading, MA: Addison-Wesley, (1999).

[11]. D. B. Stewart, R. A. Volpe, P. K. Khosla, Design of Dynamically Reconfigurable Real Time software Using Port-Based Objects, IEEE Transactions on Software Engineering, Volume 23, Nr. 12, (1997)