SOME SOFTWARE ELEMENTS OF THE MICROPROCESSOR MEASURING STATION

Dragan R. MILIVOJEVIĆ, Viša TASIĆ
Department of Industrial informatics, Copper Institute Bor, Zeleni bulevar 35, Serbia 19210 Bor, phone 38130435109,
E-mail: dane@ibb-bor.co.yu

SUMMARY

Last couple of years the Department of Industrial informatics has been developing industrial control systems. As a result, a distributed control system (DCS) has been developed. The practical applications of the system in industrial plants, especially in power consumption control and copper metallurgy production processes, confirms designed system performances in real environment. The kernel of control systems is the microprocessor measuring station - MMS. In hardware and software point of view MMS is a type of PLC (Programmable Logic Controller) with some of local and remote functional possibilities. The MMS hardware configuration is adapted for actual process control requirements with adequate number of input channels and adequate response time. It is mainly dedicated as an interactive node of industrial LAN. If it is necessary, MMS can operate autonomous, doing measuring and collecting the results, as a data logger. That is a MMU (Mobile Measuring Unit) version. The paper describes the software structure of the executive system and its main functions. The details of communications software and local operation control program are out of scope of this article.

Keywords: *process control, measuring station, industrial network, dedicated software.*

1. INTRODUCTION

Microprocessor measuring station MMS is a kind of PLC and it is associated to the technological process. On one end it is connected via transmitters to physical process parameters, and on the other via communication equipment to subordinated PC computer, which is a remote interactive workstation. It is possible to say that MMS is a soft real time unit [3] regarding to functional requests and the way of their solutions. It is based on microcontroler MC68HC11 [1]. Because the measuring station is mainly designed for classical production processes, it's standard I/O module includes up to 64 analog input channels and 128 digital input and output channels, as well. Although it is designed to run in a network environment, MMS is able to operate autonomous executing all the necessary functions. There are a few important tasks in the process control method: measuring (sampling), data acquisition and data processing. All of those functions (including the unit self test, preparing, initializing and complex data transfer procedure) are provided under special residential executive software.

2. MMS SOFTWARE

There is no official model of a Real Time System Software (RTSS), but many experts agreed with the main characteristics of that software: functionality, stability, reliability and availability. The executive software system is placed as a residential module into EPROM (the operational program also). It was developed and written in Intel DOS PC in Motorola symbolic language (assembler). The special Π-assembler environment [4] was used. Program is optimized in time of execution (response time) and memory capacity. It occupies 8 KB on the high memory addresses. Analog to the similar units, MMS software has a few main functions:

- testing the correct functions of all hardware resources before start of operation,
- initialization of important parameters (registers) and preparing unit for measuring,
- management and control regularity of running,
- measuring, acquisition and logical data control,
- data processing, analyses, presentation of the actual information and storing data in appropriate form,
- generation of alarm prompts and command signals regarding to process parameter value,
- communication with the interactive PC, as well as data transfer in both direction (measuring results and remote commands).

According to this, two program parts can be recognized:

- test and control,
- running program (operational program module).

3. TEST AND CONTROL SOFTWARE

Test and control software is a complex program entity with two main functions:

- initial testing hardware resources on power on, or start,
- permanent control of properly running (on-line control), and examination of correct execution of vital function.

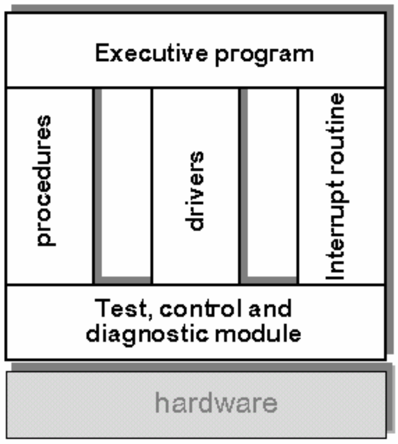

Fig. 1 MMS Software block diagram

On Power on (or Reset) sequence POST (Power On Self Test) is activated. The function of hardware resources examination is based on execution of a list of procedures containing the following modules:

- module for instruction testing (CPU Test),
- routine for timer operation examination,
- RAM test subprogram.

Test and control program starts with testing correctness of instruction execution using known operands and known result. Initial test continues with the programming timer operation including exact time intervals. The RAM test is executing in tree steps:

- the first action is to write all zeros ($00) into all RAM addresses and read them from memory,
- after that follows writing and reading all ones ($FF),
- in the third pass the second byte of address is written into every location, then read and compared.

If a mismatch occurs in any test stage, the Error code is written in Status byte and the test stops.

To protect the measured data contained in RAM, before starting the RAM test, program checks if there are any non transferred data to interactive PC. If there are, the RAM test is skipped.

On the end of initial tests, the status is placed as a result in the status byte, and on the LCD as prompts: Test CPU OK!, Test RAM OK! or Test CPU NOK!, Test RAM NOK! when test was unsuccessfull. After that MMS starts its initialization or operation mode, depending on the state of micro switches (if there was no remote initialization, default parameters are actual).

The test and control program contain also subprograms for testing some MMS functions while it is running. One of them is the module for regular A/D conversion examination. It tests every digitalization result by input sampling. The input channel adjustment means that the output of A/D conversion is between $01 and $FE. If the digitalization result is out of this interval, the measuring value is irregular (which is a kind of saturation). Program ignores the result and generates the error message. The status of wrong measuring is placed in status register, and on LC display. This program module belongs to BIST (Build In Self Test) [6] software.

4. EXECUTIVE SYSTEM

The executive software system has a task to support all of the MMS functions in operating mode. Because it has to manage and control the system resources, it may be called operating system. Such an executive system must solve three key requests: task scheduling, resource dispatching and process interconnection, or intertask communication.

Kernel (Nucleus) is the smallest part of executive software which is designed for planning and dispatching resources [2]. While MMS is in operable state there are tasks which have to be solved. The tasks start running as a consequence of some events. There are two main methods in task execution: polling (Polled Loop) and interrupt mechanism. Polling method is using for peripheral operation handling (the micro switches, LCD and functional keyboard). The high priority tasks (sending of data to PC and measuring) are also executing using polling method.

Interrupt requests (IRQ) occur in regular time intervals (periodically), aperiodically (sporadic) and as a combination of two previous cases. The IRQ sources are hardware requests: from timer for real time clock and measuring, and from serial communication interface (SCI) for data transfer. The timer requests are periodic, and the SCI are sporadic, depending on PC transfer demands. Sometimes it is possible to synchronize them, for example, when there is a Front End Processor (FEP) like a master node. Microcontroller M68HC11 does not support multilevel interrupt and it is necessary to serve each one, using interrupt vector table and Interrupt–Handling Routine, depending on it's priority. There is a possibility to change the interrupt priority, programming the HPRIO register [1].

The most often using solution for RT software is a Foreground/Background mechanism. It includes the interrupt routines and real time processes called foreground system, and a set of polling procedures for low priority tasks known as background programs [6].

Background processes are not time critical and they could be interrupted by foreground tasks [3]. One presentation of described software structure is shown on Figure 2.

MMS executive system is based on a smaller program units: procedures, drivers and interrupt handling routines, as shown in Figure 1. All of those program modules have identical formal structure, but with some particularities.

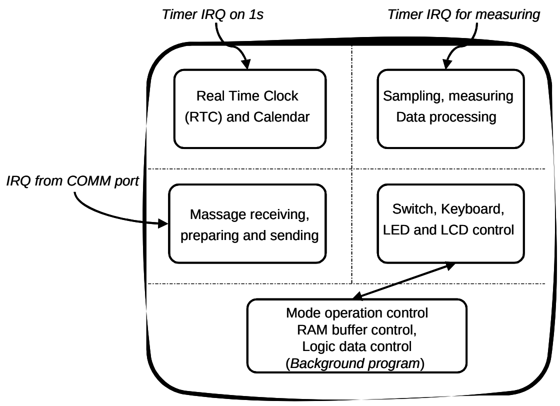

Fig. 2 Block diagram MMS SW structure (State Chart)

Procedures are created on the algorithm bases, to solve different tasks, concrete functions, or group of functions. The typical examples are: idle procedures for some delays, test procedures in selftest and BIST, data format conversion procedures, arithmetic calculations procedures etc. The procedure of timer programming and date and time generating is activated by local or remote initialization. The sampling procedure contains a special group of program modules used in measuring, logical tests of result and data processing. For message handling (preparing, receiving and sending) the communication procedure is used. The selection of MMS mode operation (initialization, active running – measuring, or local or remote control) is controlled by background control procedures. The drivers are program entities, which manage and control the peripheral MMS assemblies. They contain one or more procedures and they are executed in background. The interaction between user and MMS is carried out upon microswitches and functional keyboard. LED and LC display are used to present the MMS states, or measuring results. Changing of the state of flag registers, or semaphores, as well as manipulation with MMS using switches, or keyboard, activates the drivers. The interrupt handling routines (IHR) are the program sequences witch are immediately activated as a response to interrupt requests (IRQ). In that case there are only two hardware IRQ sources: from timer, and from SIOC. Those routines are time optimized, because there is no multilevel interrupt mechanism. That is the way to decrease overlapping possibility. If both requests ocure at the same time, request from comm port will be solved first. This is necessary because the communication and data transmission timing depend on PC, and this is quite a stochastic event on MMS side. When the communication session is finished (it can takes up to 0.1 s), the measuring and time updating starts. The start addresses of IHR are placed into interrupt vector table (IVT) to the appropriate entries, according to numbers and priority.

Executing the following instructions put the start addresses, which are calculated by compiler:

CommInterr = $ffd6; (#InterrComm);
InterrClock1 = $ffe8; (#ClockInterr1);
InterrClock2 = $ffe6; (#ClockInterr2);
InterrClock3 = $ffe4; (#ClockInterr3);
InterrClock4 = $ffe2; (#ClockInterr4);
Reset = $fffe; (#ProgramStart);

The MMS executive system starts with initialization and run in several steps (see Figure 3):

- system configuration (privileged instructions),
- disable interrupts,
- setting IVT and stack memory,
- hardware tests execution,
- register initialization, RAM buffers definition and timer preparing,
- enable interrupts.

The above actions are executing at the start of the day and are not included in operational program module.

5. OPERATIONAL PROGRAM MODULE

The MMS operational program module is realized in a form of 'endless loop', where the procedures are executed in a sequence, in background, while the interrupt routines set the flags and change the states of the semaphores. The real time clock and calendar are the background programs, which are activated at the start of operational program. If date and time initialization is not done, 01.01. 00:00:00 is used as a default.

A/D conversion preparing, sampling, logic data control and transfer to PC are the MMS key functions. The semaphore MeasureNow is examined in the main program stream, (see Figure 3). That semaphore is set by timer interrupt routine, and the value of $55 means time of measuring. Writing into ADCTL register, sampling of input channels starts, and sequence of successive approximation. After 128 machine cycles A/D conversion is completed and conversion complete flag (CCF) is set, as 27 =1 in ADCTL register. If IRQ is not automatically generated at the end of conversion, then polling mechanism is required for examination of appropriate bit in status register. The procedure described above explains the measuring method. In every measuring process calculating the average value is done in a following manner: the measuring is taken 4 times and the partial sum (for every of 8 channels) is made.

The final result for all of defined channels forms the data massage. The real date and time is added as shown on Figure 4. In standard operation mode MMS is running in a network environment and the measuring data are transferred to the subordinated PC. If some problems with communications encounter, or MMS is used as a data logger (MMU [5]) without supported PC, the data are saved in local RAM. Data storage is defined in memory map room in RAM [5]. In this case the starting address is fixed ($1040), but the end one may be $BFFF or

$DFFF, depending of version of executive and working software. Both addresses are set in initialization process after RAM test. Depending on number of input channels (message length) it is possible to write up to 3000 source messages in defined RAM zone.

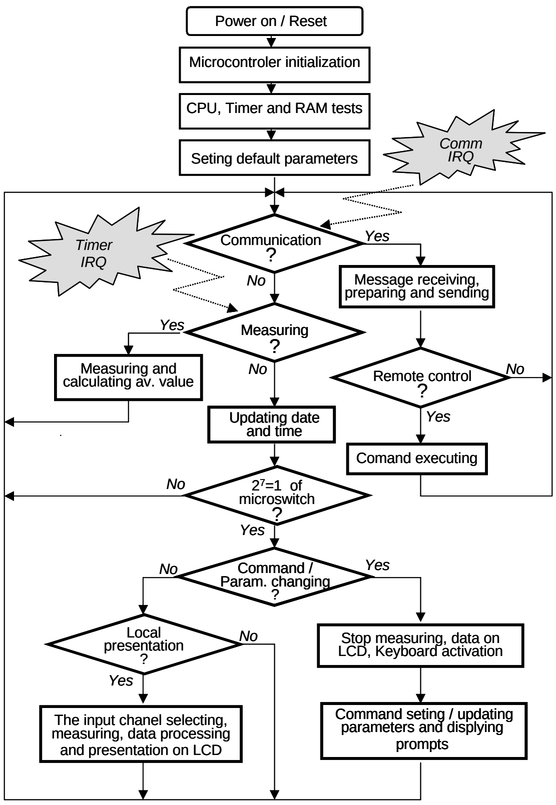

Fig. 3 MMS Working Program block diagram

Two ways of data storing are in use:

- Measuring and saving data in RAM while there is an available room, known as Mailbox Mechanism [6]. After that, the appropriate semaphore is set and measuring and memorizing is stopped,
- Continuous measuring and saving data in RAM with non fixed start and end buffer addresses. That means the new data are written over the oldest one. This is known as a Ring Buffer Principles [6] (see Fig. 5).

The first principle is used when there is a need for measuring in a fixed time period. But sometimes it is necessary to have the actual (latest) data for analyses the circumstances of an event, or accident. In such casse, the Ring Buffer Principles is more suitable. In on-line mode MMS is connected to PC and the messages are transferred immediately after appearing. When there are some historical data in MMS RAM, it is possible to transfer them to PC in following ways:

- FIFO (First In First Out) mechanism is used to download messages according to time of their appearing,
- LIFO (Last In First Out) provides the most actual data first.

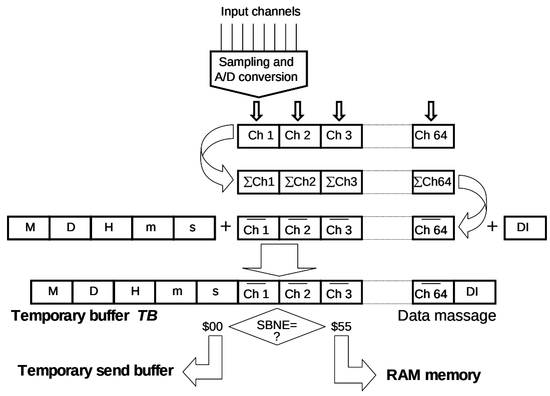

Fig. 4 The phases of message generating

In on-line mode of MMS operation, sometimes it is possible to have the actual and historical data in RAM at the same time. Both kind of messages can be transferred simultaneously, but with higher priority for actual data. That means the actual message is sent immediately after measuring, and afterwards the data from previous time period. Communication module of control software puts a control character in header [7] which determines a type of message, actual (regular) or historical.

This paper does not describe two complex executive program entities, which are responsible for local MMS control and communication in industrial networks.

All of local manipulations like a date setting, time adjusting, choice of sampling rate, number of channels etc are performed by micro switches and functional keyboard. There are a set of procedures and drivers executing in background as a program support.

Physical connection between MMS and other network nodes may be direct, or via base band modem, using the serial asynchronous communication ports. Any of them can be permanent or temporary. There is an efficient program module in MMS for receiving commands from control PC and transfer measuring data. It supports a specific protocol with defined message format [7].

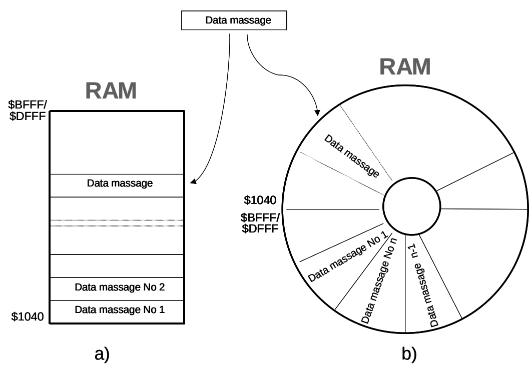

Fig. 5 Storing data in RAM: a) Mailbox Mechanism, b) Ring Buffer Principles

6. CONCLUSION

Practical MMS application has shown an optimal usage of available hardware resources. The appropriate software is responsible for optimal time scheduling and memory room management. Because of those software characteristics MMS is able to do measuring with sufficient sample rate for most of real technological processes (in copper metallurgy for instance). If there is no communication, or if the unit is used as a data logger, up to 3000 messages can be stored in local RAM and downloaded to subordinate PC. That corresponds to MMS measure autonomy of 3 to 5 days (one minute - one message). The transfer rate of 19.2 kb allows up to 20 (short) messages per second.

Although the most of the performances are well balanced, there is a place for further improvment. It should be useful to improve the drivers for keyboard to avoid the time-out principle that produces the visible delays, and makes the response time longer. Also, the communications software could have the possibility of wireless communication.

The real power of MMS is reflected in network environment, but it should be subject of an other article.

REFERENCES

[1] M68HC11 Reference Manual, Motorola Inc., 1990.

[2] Laplante P: Real-Time Systems Design And Analysis, IEEE Computer Society Press, New York, 1997.

[3] Joseph M: Real-Time Systems – Specifications, Verifications and Analysis, Prentice Hall Hertforshire, 1997.

[4] Radonjic Z: ∏-assembler user guide, Niš, Serbia, 1996. (in Serbian)

[5] Milivojevic D, Lazic B, Tasic V.: Mobile measuring station, Conference ETRAN 2000, Soko Banja, Serbia, 2000. (in Serbian)

[6] Nisanke N, Realtime Systems, University of Reading, Prentice Hall Hertforshire, 1997.

[7] Milivojevic D, Tasic V, Karabaševic D.: Communication in realized RTS systems, Conference ETRAN 2003, Herceg Novi, Montenegro, 2003. (in Serbian)

BIOGRAPHIES

Dragan R. Milivojević was born in 1949. He graduated at the Faculty of Electrical Engineering, at University of Belgrade in 1974. He defended his MSc. thesis "Microprocessor system for air quality monitoring " at the same faculty at 1984. Since 1976 he is working as hardware engineer in CIB (Copper Institute Bor) Computer center. Last 15 years, as Head of the Department of Industrial informatics and expert adviser, he is developing industrial control systems. He presented and published more then 70 articles and few books. At the moment he is a PhD. candidate at University of Belgrade, Technical faculty in Bor.

Viša Tasić was born in 1968. He graduated at the Electronic Faculty, University of Niš in 1992. He defended his MSc. thesis "Materials characterization at microwave frequencies by resonant method" at the same faculty at 2003. Since 1994 he is working as system engineer in CIB. Last 10 years, as a Head engineer for automatics at the Department of Industrial informatics, he is developing industrial control systems. He presented and published more then 40 articles. At the moment he is a PhD. candidate at Faculty of occupational safety, University of Niš.