ALARM / INTERLOCK SYSTEM

L.Kotchenda, P.Kravtsov, V.Trofimov

УДК 539.107.4   УДК 621.317.799:681.325 ??

АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ ДЛЯ ЭКСПЕРИМЕНТА PHENIX

Л. Коченда, П. Кравцов, В. Трофимов

Рассматривается микропроцессорная автоматическая система управления и блокировки для эксперимента PHENIX, построенная на базе однокристальной ЭВМ Intel 87C51. Система обеспечивает надежное управление установками газообеспечения детекторов DC/PC и TEC/TRD и защиту детекторов в случае аварийных ситуаций.

Гибкая настройка алгоритмов реакции системы позволяет использовать ее в широком круге экспериментальных установок.

ABSTRACT

The Alarm / Interlock System is based on Intel 87C51 microcontroller. It provides reliable control of the gas systems for PHENIX DC/PC and TEC/TRD detectors (PHENIX Project, Brookhaven, USA) and prevents the detectors from operating under unsafe conditions.

Flexible alarm action configuration gives one a possibility to use it not only for Gas System, but also in any project that requires simple and fast interlock system.

Alarm System Description

The objective behind the Alarm System was to create a reliable control system that will operate gas system and take some actions in case of certain alarm conditions. The Alarm System (Fig.1) is based on Intel 8051 microcontroller running at a 11.0592 MHZ clock frequency [1]. It has 8Kb non-volatile RAM to keep alarms configuration and status information there. Analog signals from sensors go through two 16-channel multiplexers and instrumentation amplifier to 12-bit ADC (Burr-Brown ADS7806 [2,3]), which can be replaced by 16-bit one. CPU takes care of handling these signals, comparing them to alarm thresholds, and sets corresponding digital output signals via four 3-wire registers. In order to block some alarms a keypad is used. All triggered and blocked alarms are indicated on two lines LED indicator. There is a supervisor watch-dog LTC1232 on this board, which attends to 8051 program faults and resets CPU as fast as in 0.8 s in this case. Alarm system can be connected to PC via standard RS-232 port.

To change blocking of any particular alarm one should enter the number of this alarm (1÷32) and press [*] button on keypad, while [#] button unblocks current alarm. One can block or unblock all alarms together by entering "79" as alarm number and pushing [*] or [#] button.
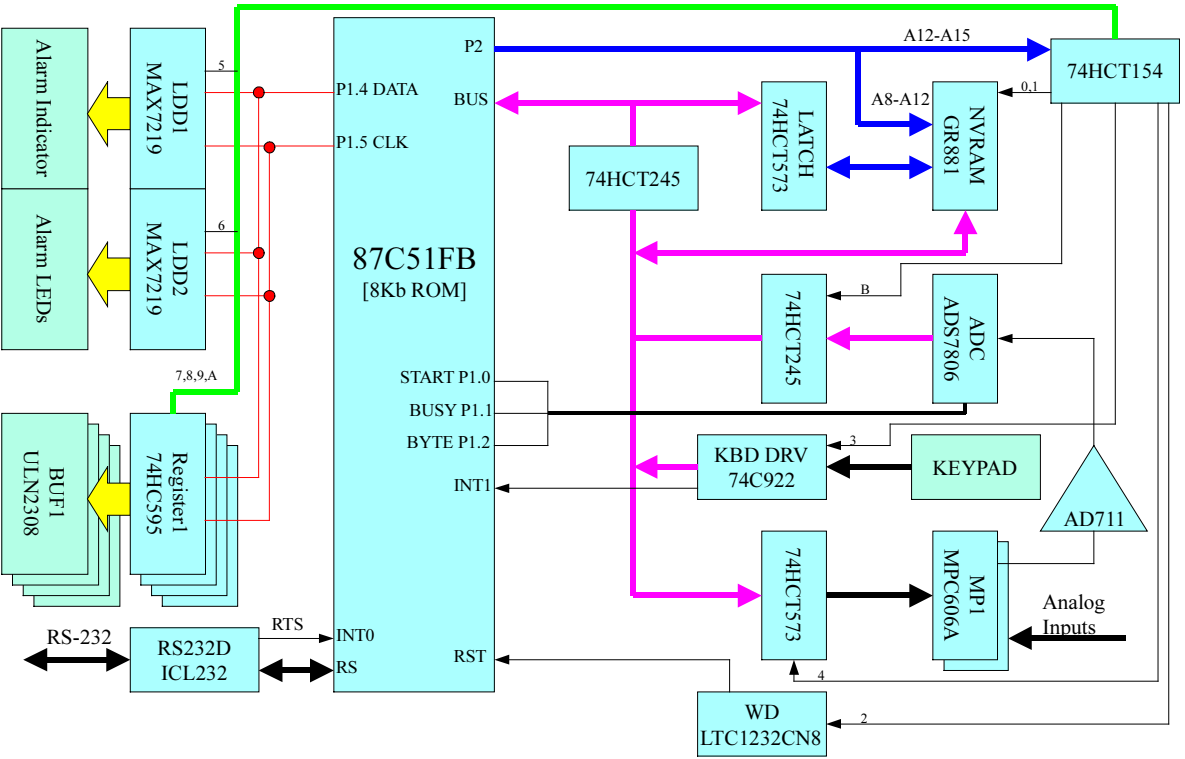
Fig. 1. Alarm System function diagram.

Alarm Indicator

Alarm LEDs

BUF1 ULN2308

LDD1 MAX7219

LDD2 MAX7219

Register1 74HC595

5

6

7,8,9,A

P1.4 DATA

P1.5 CLK

87C51FB
[8Kb ROM]

P2

BUS

START P1.0

BUSY P1.1

BYTE P1.2

INT1

INT0

RS

RST

A12-A15

A8-A12

74HCT154

0,1

LATCH 74HCT573

NVRAM GR881

74HCT245

74HCT245

ADC ADS7806

B

KBD DRV 74C922

KEYPAD

3

74HCT573

MP1 MPC606A

AD711

Analog Inputs

RS-232

RS232D ICL232

RTS

WD LTC1232CN8

4

2

Alarms handling

Alarms handling algorithm is shown in Fig.2. Each analog signal is read by the CPU, averaged by certain samples, and compared with the threshold with an appropriate sign of comparison ("<" or ">"). If an alarm occurred at this signal (analog value exceeds the threshold), the program lights correct LED. Then it takes into account the blocking status of the current alarm. If alarm is not blocked, software takes appropriate action by changing the state of the digital outputs. In case of the blocked alarm, program just lights both LED for this alarm (ALARM LED and BLOCK LED) and tries to return digital outputs to non-alarm state, if it's allowed by other channels.

The idea was as follows. If alarm is not blocked and there is an alarm situation in some channel (some parameter exceeds the threshold) the code changes the state of the digital outputs only in the memory, and stores in the memory which
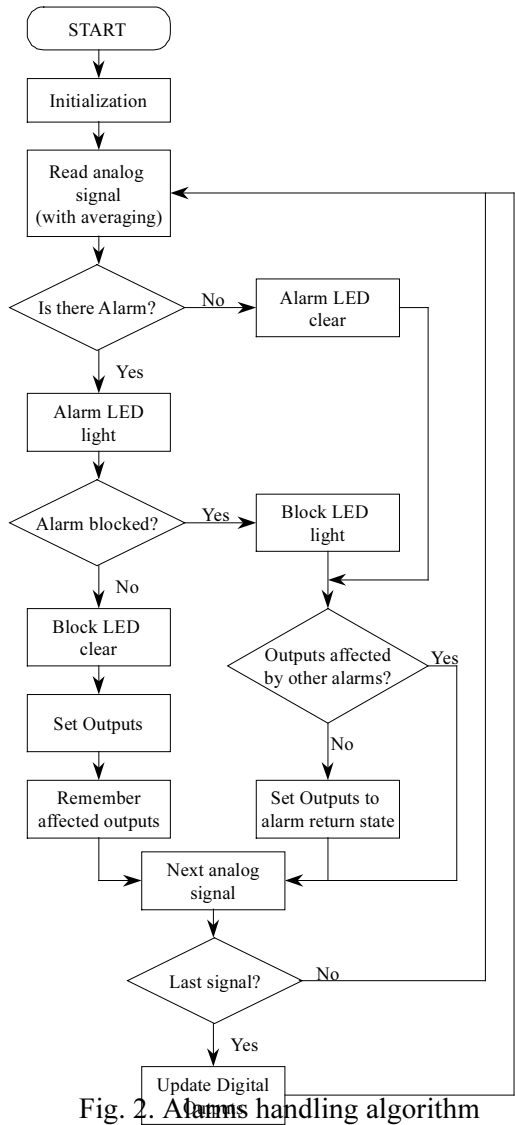

Fig. 2. Alarms handling algorithm

outputs it changed. If there is no alarm, the code changes only those outputs, which were not alarmed by the previous channels, making use of the stored list of the changed outputs. So the outputs which changed their states because of the alarm are not affected by the alarm recovery procedure, if we consider a single processing (handling) cycle. Only after processing of all the channels the result of their influence upon the digital outputs really comes out of the memory. This eliminates wrong digital output states and "flashing" of these signals.

PC software and alarms configuration

Alarm System provides reliable RS-232 data exchange in order to change alarms configuration and reading system status. The software for PC is developed for Windows platform. The window of this program has three tabs: one for reading Alarm System status (Fig. 3), second for configuring alarms (Fig. 4), and third for Analog Inputs test (Fig. 5).

The "System parameters" tab includes small communication setup. User must select COM port connected to Alarm System. The speed of data exchange is always set to 19200b/s. Program will not communicate with Alarm System without opening appropriate COM port. If program sees Alarm System connected to specified port, it displays internal software version of this system. Port will be automatically disconnected in 10 seconds if no Alarm System observed on this port. The "Read system parameters" button reads Alarm System status information. It updates the following parameters:

- **Up time** – the time since last power-up,
- **WD resets** – watch-dog counter, number of WD resets since last power-up,
- **Averaging by** – number of analog signal samples to average by,
- **Valves state** – current state of all devices connected to Alarm System
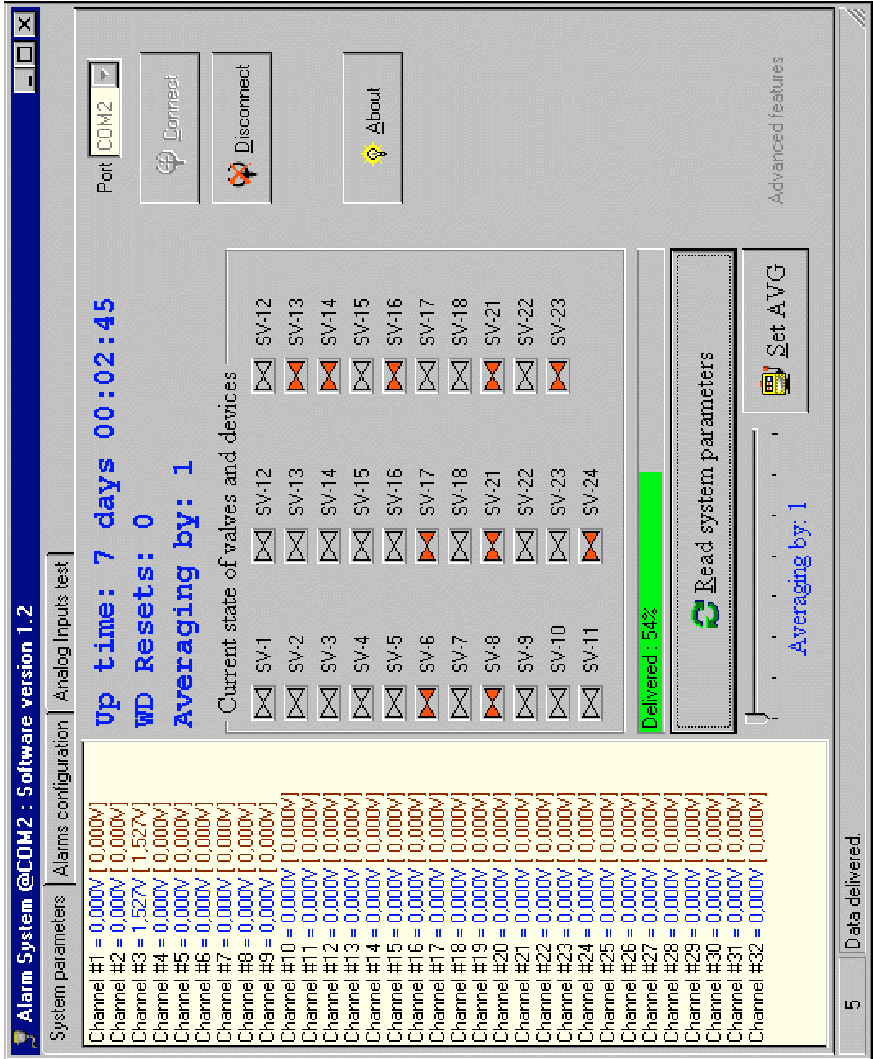- **Physical values and voltages** of analog signals.

Fig. 3. "System parameters" tab.

Besides, user can change averaging number using trackbar at bottom of the window and "Set AVG" button. This value will be taken into account in the next alarm handling cycle. Time required for one 32-channel cycle as function of averaging number is given in table 1.

Table 1. Cycle time for different averaging numbers

Average by	Time, ms
1	22
2	30
4	42
8	68
16	120
32	220
64	420
128	800

The "Alarms configuration" tab (Fig.4) is used to configure alarms and actions to be taken in case of any particular alarm. For every alarm user is able to change:

- **Alarm event name** – this string is used only in the PC program for better understanding what this alarm is used for, instead of numbers in Alarm System
- **Level** – alarm threshold. Exceeding this threshold analog signal value causes an alarm
- **Unit** – only in PC program, units of physical value of this analog signal
- **KA** – A coefficient for calculating physical value from voltage. PC only
- **KB** – B coefficient for calculating physical value from voltage. PC only
- **Devices configuration** – alarm action, see further description
- **Alarm threshold sign** – alarm comparison sign

Alarm System @COM2 : Software version 1.2

System parameters | Alarms configuration | Analog Inputs test

Alarm event name	Level	Unit	KA	KB
Alarm 0	3.000	%	0.000	1.000
Alarm 1	2.500	V	0.000	1.000
Alarm 2	3.000	V	0.000	1.000
Alarm 3	3.000	V	0.000	1.000
Alarm 4	3.000	V	0.000	1.000
Alarm 5	3.000	V	0.000	1.000
Alarm 6	3.000	V	0.000	1.000
Alarm 7	3.000	V	0.000	1.000
Alarm 8	3.000	V	0.000	1.000
Alarm 9	3.000	V	0.000	1.000
Alarm 10	3.000	V	0.000	1.000
Alarm 11	3.000	V	0.000	1.000
Alarm 12	3.000	V	0.000	1.000
Alarm 13	3.000	V	0.000	1.000
Alarm 14	3.000	V	0.000	1.000
Alarm 15	3.000	V	0.000	1.000
Alarm 16	3.000	V	0.000	1.000
Alarm 17	3.000	V	0.000	1.000
Alarm 18	3.000	V	0.000	1.000
Alarm 19	3.000	V	0.000	1.000
Alarm 20	3.000	V	0.000	1.000

Alarm 0

Solenoid Valves and devices configuration

SV-1 SV-12
SV-2 SV-13
SV-3 SV-14
SV-4 SV-15
SV-5 SV-16
SV-6 SV-17
SV-7 SV-18
SV-8 SV-21
SV-9 SV-22
SV-10 SV-23
SV-11 SV-24

SV-12
SV-13
SV-14
SV-15
SV-16
SV-17
SV-18
SV-21
SV-22
SV-23

Open ALL   Close ALL   Don't touch ALL

Alarm Threshold sign
○ Sensor data > Level causes alarm
○ Sensor data < Level causes alarm

Read   Write

Delivered : 0%

Registry   TXT   Binary   Binary

0   Data delivered.

Fig. 4. "Alarms configuration" tab.

Alarm names, physical value units, coefficients are intended to appear only in PC program. Voltage is calculated from physical value by the formula:

$$V = \frac{S - A}{B},$$

where S is physical value, V - voltage, A and B – coefficients.

Solenoid valves and devices configuration allows one to customize the alarm action. If an alarm occurred, the *red* valves will be closed and *red* devices will be turned off. *Green* valves will be open, *green* devices will be turned on. *Gray* valves and devices *will not be affected* in case of this particular alarm. For example, in the valves configuration showed on Fig. 4, SV-1 and SV-2 will be closed, SV-4 will be open while all other valves will not be affected, i.e. their state will be intact.

"Read" button downloads current alarm configuration from Alarm System, "Write" button uploads it to Alarm System. *Note that COM port should be open before these operations*. User can save current configuration to Registry (it is done automatically when window closes or by "Registry" button), save and load it to/from binary file. Besides, "TXT" button exports alarm configuration to text file as table with alarms in rows and valves state in columns, which can be extremely useful for analyzing in MS Excel to see the whole configuration of all the alarms on one page.

Analog Inputs test

There is special feature implemented in the software to check noise protection of analog input channels (Fig. 5). When this mode is activated, Alarm System CPU is sending all 32 digitized and averaged analog signals once per alarm handling cycle. The number of such samples is given by PC software and may not exceed 255 (restricted to 10÷100 by PC software). This allows one to investigate dependence of the statistical error on averaging constant.
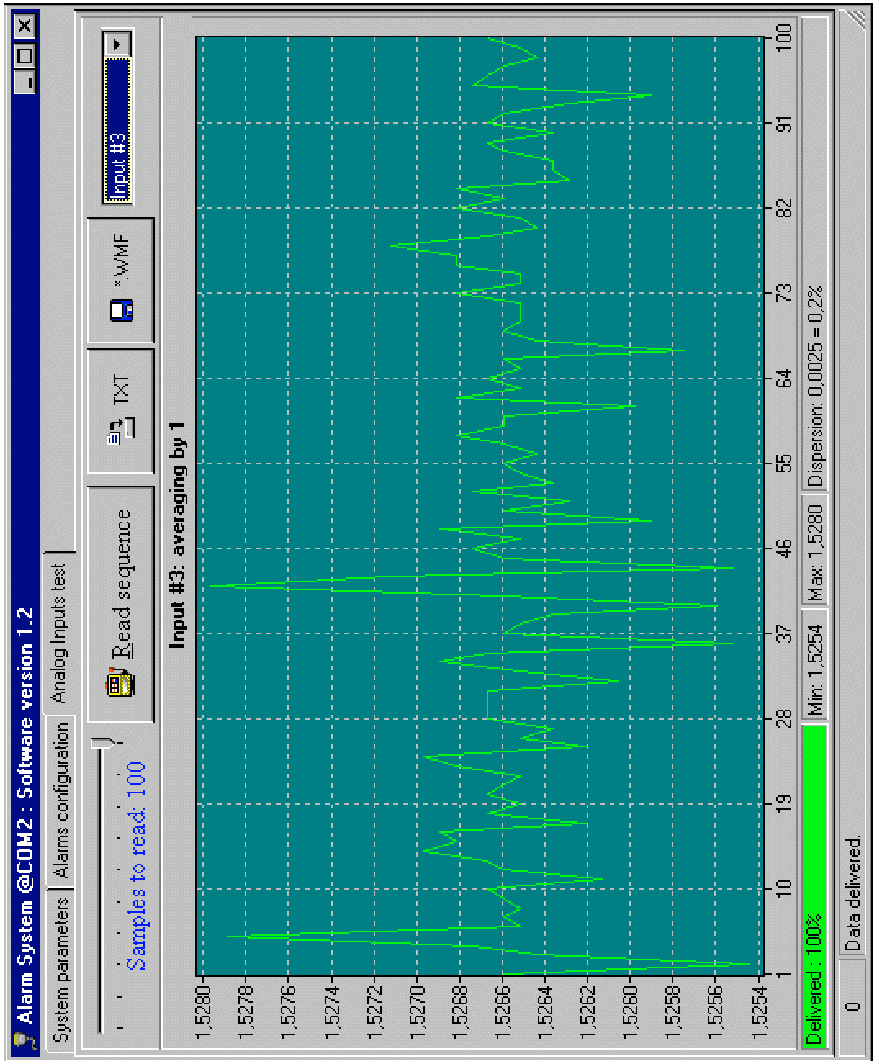
Fig. 5. "Analog Inputs test" tab.

The results of the averaged sample measurements are shown on Fig. 6 and Table 2. It is obvious that dispersion decreases with increasing number of samples which signal is averaged by. There is no necessity to increase it over 32 for signals with low noise, because the dispersion is almost constant there. Besides, measured value goes higher with increasing averaging number, which is regular and reflects a calculation error during averaging. That is not important for the Alarm System itself, because the shift of the values is negligible (1.5 mV) and comparable to measurement error 1 mV.

Table 2. Dispersion for different averaging numbers

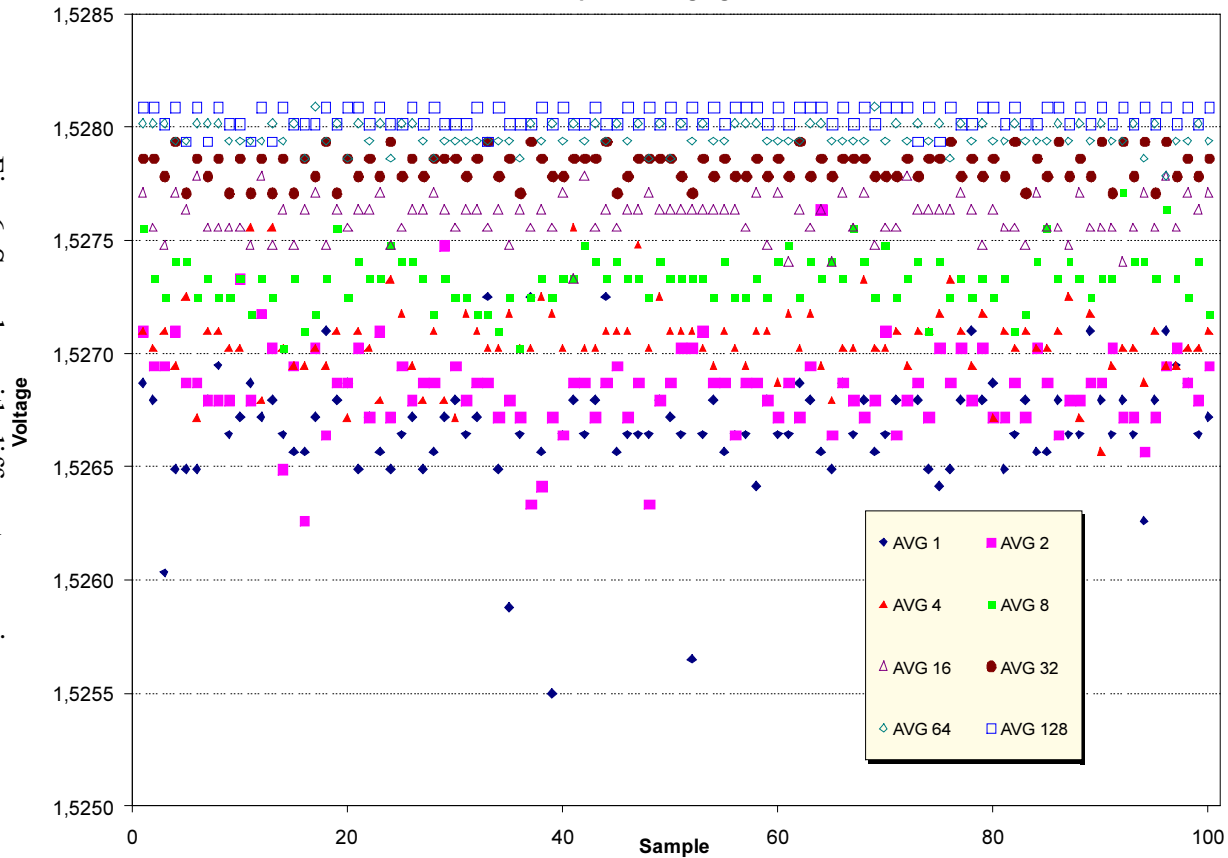| Average by	Dispersion, ADC units	Dispersion, mV
1	23	1.75
2	18	1.37
4	13	0.99
8	9	0.69
16	6	0.46
32	3	0.23
64	4	0.31
128	2	0.15

Fig. 6. Samples with different averaging.

Conclusion

During tests and measurements Alarm System appears to be very reliable and stable device. For instance, system had been worked over 7 days (Fig. 3) with no watchdog resets, i.e. no 8051 CPU software faults. Flexible alarm action configuration gives one a possibility to use it not only for Gas System, but also in any project that requires simple and fast interlock system. The main features of the Alarm system are:

- 32 analog input channels with 16-bit resolution
- 32 digital output channels (transistor switches)
- RS-232 interface at 19200 baud
- Separate and total alarm blocking from keypad on front panel
- 22 ms minimal response time (without averaging)
- Flexible alarm actions configuration for each particular alarm
- System information available via RS-232, including watchdog resets counter, up time, analog values and digital output status
- Analog inputs test mode for choosing convenient averaging number
- Power consumption - 15 mA for ±12 V and 250 mA for 5 V with all LEDs lighted

ACKNOWLEDGEMENTS

REFERENCES

1. *MCS-51 Intel Family of Single-Chip Microcomputers*. User's manual. Intel Corp., 1981.
2. *Tips for using the ads78xx family of a/d converters*, Burr-Brown Application Note AB-95.
3. *ADS7807 Low-Power 16-Bit Sampling CMOS analog-to-digital converter*, Burr-Brown product information.
4. Avocet C51 Reference Manual.