

Khulna University Studies 2(1): 23-30

MICROPROCESSOR BASED PROTECTIVE RELAYING

Md. Shamsur Rahman Chowdhury^{a*} and Zhen Zhang Jiang^b

^a *Electronics and Communication Engineering Discipline, Khulna University, Khulna 9208, Bangladesh*

^b *Department of Electrical Engineering, South East University, Nanjing, P. R. China*

Manuscript received: October 20, 1999; Accepted: June 25, 2000

Abstract: Protective relaying using microprocessors is becoming popular mainly due to its speed and reliability. In this paper, protection against overheating of a reactor has been designed and discussed. The Intel 8088 microprocessor has been used and the program has been designed in Turbo C language. Each heater coil of the reactor is connected with a circuit breaker. When temperature in a chamber of the reactor exceeds a set value (the safety limit) the corresponding circuit breaker will trip to protect the chamber from damage due to overheating. In our particular reactor, there are ten chambers. In this paper, protection for only one chamber has been designed and discussed. Programs were written for Intel 8088 microprocessor but were executed in a computer using Intel 80386 microprocessor. It is not a problem because Intel 80 family members are upwardly compatible.

Key words: Protective relaying; Solenoid power circuit; Reactor

Introduction

Now-a-days, electromechanical relays for control and protection are being replaced by computers, since the later are more reliable than the former. There are seven main and independent tasks of an industrial control computer as follows:

1. Process instrumentation;
2. Process stabilization;
3. Acquisition of measurement data;
4. Process supervision;
5. Protection and safety;
6. On-Off control, and
7. Set-point control (governing).

* Corresponding author. Tel.: 0088-041-720171-3, Fax: 731244, E-mail: ku@bdonline.com

Here we see that fifth task is "Protection and safety" of a process. Protective relaying is unavoidable in any process industry. In the case of electromechanical relays, adjustment of timer setting is not always very accurate whereas in the case of computers, timer setting is to the point accurate. The electromechanical relays are relatively slow (switching is in the order of milliseconds) whereas computers are very fast (switching is in the order of microseconds).

Hardware System Configuration

The experimental reactor with its measurement, control and protective systems is shown in Fig. 1. Reactants within the reactor are heated by heater coils, HC0-HC9. Currents through the heater coils are controlled by variable resistors, VR0-VR9, thus controlling the temperature within the reactor chambers. Variable resistors are controlled by Servo motors, MR0-MR9. TS0-TS19 are twenty temperature sensors. Outlets of temperature sensors are connected to signal modifier systems. The motors operate through reduction gearing systems to have damping action and extended range of control (Chowdhury, 1996).

The signal modifier system consists of transducer (TD), analogue signal multiplexer (ASM), programmable gain amplifier (PGA), sample and hold (S/H) and analogue to digital converter (ADC). Temperatures are measured by temperature sensors (TS0-TS19). Thermocouples are used as temperature sensors. Temperature signals directly measured by temperature sensors are modified in the transducer (TD). Twenty measurement points from the transducer are connected to the analogue signal multiplexer (ASM), which outputs the analogue signals sequentially to the programmable gain amplifier (PGA). The output signal is amplified to a desired level by the PGA. The output of PGA is fed to the sample and hold unit (S/H). S/H unit is used to filter out the noise present in the unknown signal to be converted from analogue to digital values. From the S/H unit the analogue signal is fed to the analogue to digital converter (ADC). The analogue signal is converted to digital signal in the ADC. This digital signal from ADC is input to the I/O port of the microcomputer. The microcomputer system consists of Intel 8088 microprocessor, 2164 dynamic RAM, 8203 RAM controller, 2732 EPROM, 8254 timer, 8259A interrupt controller, 8255A parallel I/O, 8251A serial USART and 8237 DMA controller.

The output system from the microcomputer to the actuator consists of optoisolator (OI), solenoid power circuit (SPC), and solenoid relay. The opto-isolator system (OI) optically isolates the trip signal from the actuator circuit. The solenoid power circuit (SPC) is shown in Fig. 2.

The solenoid relay operates on 220V 1 PH 50 HZ AC supply. The solenoid relay is working as the actuator. Working principle of protective relaying for all the solenoid relays (SR0-SR9 and MSR) is the same.

Legend for Fig. 1.

ADC =	Analog to digital converter	POUT	Product out
ASM=	Analog signal multiplexer	PS	Pressure sensor
CB=	Circuit breaker	RC	Reactor chamber
CV=	Control valve	RG	Reduction gear system
D	Display	RR	Reactor
DAC	Digital to analog converter	RV	Relief valve
DDIS	Digital display	SB	Safety blow
FE	Fuse	S/H	Sample and hold
FIN	Feed inlet	SR	Solenoid relay
FS	Flow sensor	SPC	Solenoid power circuit
HC	Heater coil	T	Trip
ICV	Inter chamber valve	TD	Transducer
MCB	Main circuit breaker	TS	Temperature sensor
MR	Servo motor	VR	Variable resistor
MSR	Main solenoid relay	WIN	Water inlet
OI	Optoisolator	WOUT	Water outlet
PGA	Programmable gain amplifier		

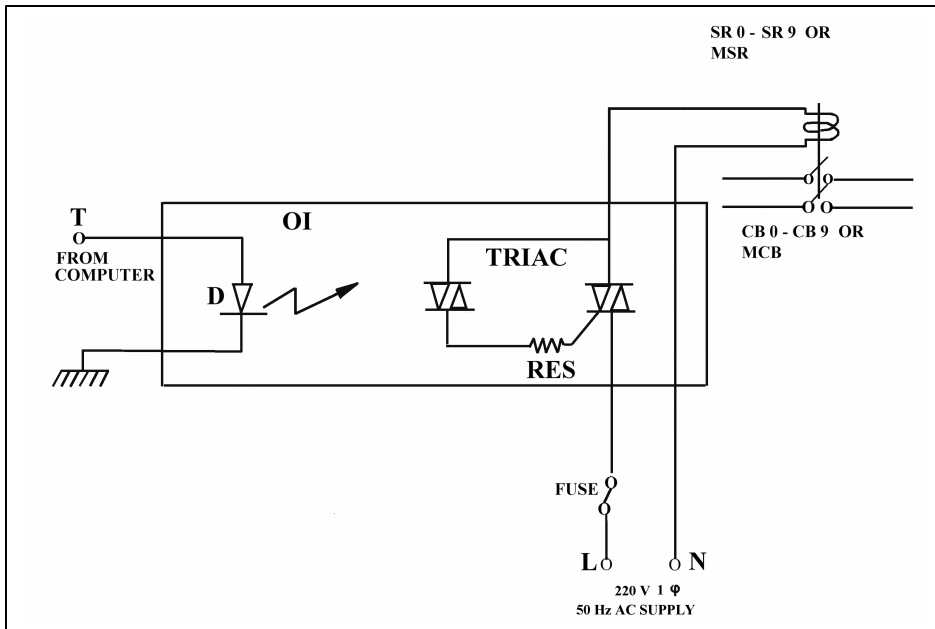


Fig. 2. The solenoid power circuit.

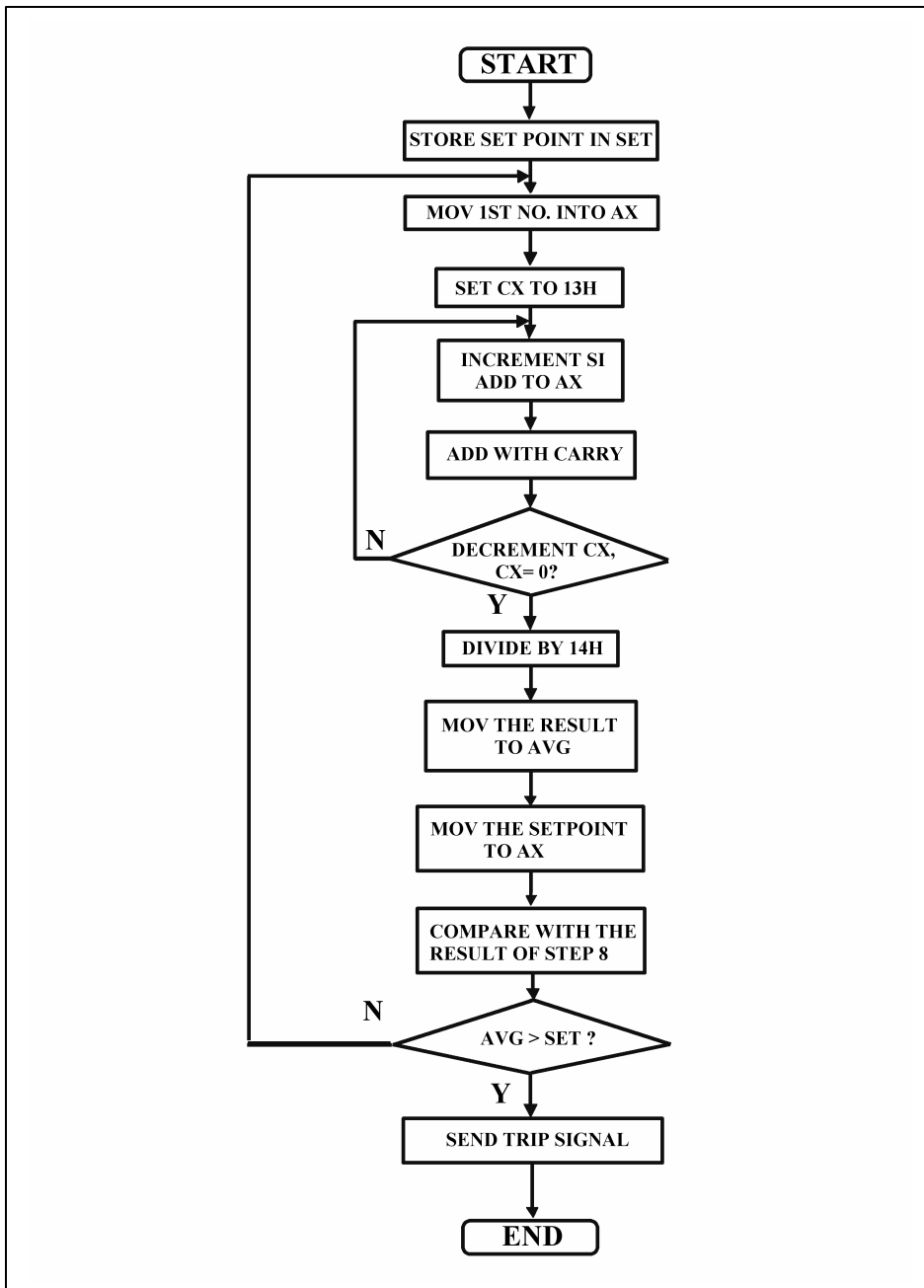


Fig. 3. Flow chart for the trip program (Chowdhury, 19961).

Development of the Program

The flowchart for trip program is shown in Fig. 3. The program in turbo C language to trip the circuit breaker is shown in Table 1. The circuit break will trip by the trip signal from T of the microcomputer.

Table 1. The program for tripping a circuit breaker (Chowdhury, 1996).

```
#include <stdio.h>
#define size 20
#define out0 0220h
main ( )

{
    int i, sum=0;
    int data[size];
    int avg;
    asm mov si,0
    for (i=0;i<size;i++)
    {
        asm push ax
        asm push ds
        asm mov ax,0010h
        asm mov ds,ax
        asm mov ax,ds:[1000]
        asm pop ds
        asm inc si
        asm inc si
        asm mov [data+si],ax
        asm pop ax
    }

    for ( i=0; i<size; i++ )
    sum=sum+data[i];
    printf ("the average is %d\n",sum/size);
    avg=sum/size;
    asm push ax
    asm mov ax,avg
    asm cmp ax,240
    asm ja trip
    asm jmp skip
trip:    asm mov ax, 0001h
        asm out out0,ax
        asm mov bx,10
back :  asm dec bx
        asm jnz back
        asm mov ax,0000h
        asm out out0,ax
skip:   asm nop
}
```

Discussion

Generally, the term "Protective relaying" is associated with the protection of power systems. Faults may occur in a power system in the form of over current, over voltage etc. When a fault occurs in a power system, the protective relay detects it, closes the trip circuit and the circuit breaker trips to disconnect the faulty system from the source. In this paper, the term "Protective relaying" is associated with the protection of a process industry. Here faults may occur in the form of over-temperature, over-pressure etc. So, although, function of protective relays is the same in both the cases, procedure of disconnecting the faulty section from the source is not always the same. Let us take the example of controlling temperature of a gas fired furnace. In the case of protection against over-temperature, the relay system detects it and shuts off the gas valve instead of circuit breaker as in the case of over current protection of a power system. The reactor described in this paper is electrically heated. There are circuit breakers in the heating system. So, protective relaying and circuit breaking in the case of a fault (say over temperature) are similar to those of power systems. The program has been designed in Turbo C language. The program could also be designed in Assembly language. Both the languages have advantages and disadvantages as follows.

Assembly language is a powerful language that gives the programmer absolute control over the computer. High-level languages are dependent on compilers, but assembly language is stand alone and independent (Murray and Pappas, 1986). When developing software from a dedicated processor, assembly language is unavoidable; because it gives the processors direct access to registers, memory and the unique world of bit-oriented instructions. Assembly language program will produce the fastest executable codes because it does not need the interpreter or compiler step.

An assembly language program takes more space than a high-level language program for the same task. For this reason, most large programs are not written in assembly language and in those cases high-level languages are preferred.

Turbo C is an exceptionally powerful language. Turbo C language allows programmers to directly access memory locations and registers within the computer (Winer, 1983). Turbo C can be used in machines of all sizes. The speed and compactness Turbo C codes are the major reasons for Turbo C's popularity. Turbo C can be said to be a high level assembly language; because Turbo C language allows a programmer to access to the bits, bytes, registers and external devices. Turbo C is a professional quality Turbo C development system that includes a rich set of libraries under MSDOS. Turbo C was originally found to be used in computer operating systems. But afterwards, it is used in writing programs ranging from data base applications to assemblers.

Turbo C is so a delicate language that a little deviation from it's set rules may cause hazards.

Conclusion

Industrial control computers are replacing now-a-days, electromechanical relays for control and protection in process industries. Reasons for this have been discussed in this paper in details. We see that simple programs can do difficult tasks of saving a very costly and very large equipment like

reactor. In future, we hope, industrial control computer will take over the charge of safeguarding a process industry, saving huge amount of money.

At first, use of an industrial control computer in the field of protective relaying has been justified. Hardware system configuration of measurement, control and protection of a reactor has been given. The protective system of the reactor against over temperature has been designed and discussed in details. Flowchart and program for protection of the reactor against over temperature have been worked out. Finally, features of two main programming languages have been discussed.

References

- Chowdhury, M.S.R., 1996. Program design for a distributed control system using 80386 and 8088 microprocessors in assembly and C languages and interfacing between the two languages. Senior Advance Course Project Report, South East University, Nanjing, P.R. China.
- Murray, III W.H. and Pappas, C.H., 1986. *80386/80286 Assembly Language Programming*. McGraw Hill, Osborne, pp. 1-6, 153-180.
- Winer, R.S., 1983. *Turbo C at Any Speed*. John Wiley and Sons Inc., USA, pp. 1-17.