International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 11, November-2016

OverLoad Protection using Microprocessor based OverVoltage Relay:Proteous Simulation

Paruchuri Anwitha
Electrical and Electronics Department
Chaitanya Bharathi Institute of Technology
Gandipet, Hyderabad, Telangana 500075

Sai Srujana Yalamareddy
Electrical and Electronics Department
Chaitanya Bharathi Institute of Technology
Gandipet, Hyderabad, Telangana 500075

***Abstract*— This paper presents a PROTEUS model of Micro controller based over voltage static Relay. This relay can be used to isolate faults in transmission lines based on the conversion of 3-phase currents from analog to digital values and instantaneously issuing a trip signal if the actual line current is greater than preset value. The proposed model can be implemented in transformers and generators as a Differential relay, and also it can be applied to implement over voltage protection in domestic households. The model uses various analog devices for conversion purposes and displays the current values as sensed by the micro controller.**

Keywords— ADC, static relay, Keil C Programming, I to V converter, Microcontroller.

I. INTRODUCTION

Power system protection is required to isolate a faulty section of the power system so that the system can function satisfactorily without any severe damage due to fault current. The major functional requirements of a protective relay include sensitivity, selectivity, speed and reliability. Conventional relays like electromechanical relays employed bulky sized mechanical comparators which reduced their speed and these relays often served a single function and a single characteristic. Hence domestic and industrial facilities are increasingly investing in microprocessor-based relays to greatly improve the reliability and flexibility of the electrical system and to eliminate dispensable components like auxiliary relays, wiring, switches and trip coil monitors to increase the speed. In general where tripping mechanism is employed to isolate the load under over voltage conditions, the MCB (Miniature Circuit Breaker) switch has to be manually restored to its original position after load voltage reaches to normal operating conditions. Where as in the proposed model the relay automatically restores to its original position once the system voltage reaches to normal operating conditions.

The model in this paper utilizes a current transformer to sense the AC line current and converts it into voltage using I to V converter. Then ac voltage is converted into DC using a precision rectifier. This sensed voltage is then fed to ADC0804 (Analog to Digital comparator) via sample and hold circuit. The signal so obtained is compared with reference value using AT89C51 (Atmel 89C51). Subsequently the relay signal can be issued by the micro controller if the sensed value exceeds the preset value.

II. PROPOSED MODEL [3]

In proposed model relay operation is dependent on load voltage i.e., if load voltage exceeds preset value then relay isolates the load from supply voltage. If V3 exceeds preset value then relay operates and isolates the load. If the load voltage is less than preset value then relay gets off automatically unlike MCB and hence current continues to flow through the load.

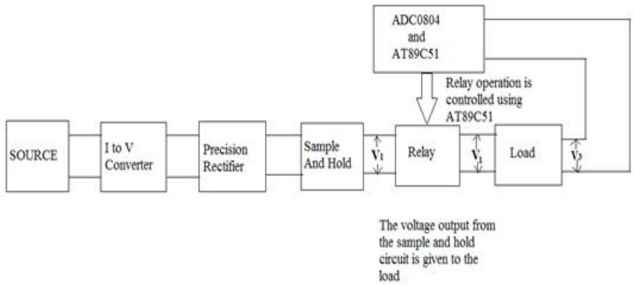

Fig 1: Proposed model

III. METHODOLOGY FOR OVERVOLTAGE PROTECTION USING MICROCONTROLLER INTERFACED RELAY

There is always a probability that an electrical power system would suffer from unexpected over voltages. The causes of these anomalous over voltages could be sudden interruption of heavy loads, lightning strikes, failure of control equipment or switching impulses. An "Over Voltage Relay" is a type of protective relay which operates when the load voltage exceeds a preset value and in a standard application it is used to identify over voltages and isolate the corresponding equipment [1]. The basic idea of this model is to compare the value of the voltage as sensed by the ADC of the microcontroller and compare it with set reference value. To perform such comparison, the program can be written in C language using Keil µ vision and can be loaded into the microcontroller. In order to sense the current, first the AC value of current is converted into AC voltage using an 'I to V' converter and then this AC voltage proportional to load current is converted into full wave rectified voltage. Then the output from precision rectifier since it is alternating it is made to hold on to its last sampled value.

The signal is now sent to analog to digital converter (ADC) to obtain signal in digital form. Microprocessor then sends a signal to ADC for start of conversion (SOC), examines whether the conversion is completed and on receipt of end of conversion (EOC) from ADC, it receives the data in digital form. If this sensed value is greater than the reference value, then the relay disconnects the apparatus from rest of the power system. The entire model was simulated in PROTEUS and satisfactory results were obtained. In order to change the value of incoming DC current, a rheostat can be connected in series to test for different values of line current. The programming is done in Keil micro vision, a popular compiler for microcontrollers.

IV. SOFTWARE UTILIZED

A. KEIL Software:

The keil C51 Compiler allows you to write 8051 microcontroller applications in C that, once compiled, have the efficiency and speed of assembly language. Language extensions in the C51 Compiler give you full access to all resources of the 8051.

The C51 Compiler translates C source files into relocatable object modules which contain full symbolic information for debugging with the µVision Debugger or an in-circuit emulator. In addition to the object file, the compiler generates a listing file which may optionally include symbol table and cross reference information.

B. PROTEUS Software:

Proteus (**pro***cessor for* **te**xt **e**asy *to* ***US****e*) is a fully functional, procedural programming language created in 1998 by Simone Zanella. Proteus incorporates many functions derived from several other languages: C, BASIC, Assembly, and Clipper/dBase; it is especially versatile in dealing with strings, having hundreds of dedicated functions; this makes it one of the richest languages for text manipulation.

V. FUNCTIONS OF DIFFERENT BLOCKS:

A. Power Supply

The power supply uses a step down transformer to step down the input mains voltage to a voltage level suitable for the electronics within the device.

B. I to V Converter

The current from the transformer is converted to voltage because the microcontroller accepts only voltage as input. Hence the current from transformer is converted voltage using I to V converter block[2].

C. Precision Rectifier

Often called as super diode, the precision rectifier is used in order to convert the alternating value of voltage into full wave rectified value. The precision rectifier is made up of Op-Amp in conjugation with diode and resistors[2].

D. Sample and Hold

As the output form the precision rectifier is alternating it cannot be given as input to ADC (Analog to Digital converter) directly. A sample and hold circuit samples an input signal and holds on to its last sampled value until the input is sampled again [2].

E. ADC0804 (Analog to Digital comparator)

As the microcontroller can compare only digital voltages with preset value and the output from Sample and Hold circuit is analog voltage, the analog value is converted to digital using ADC0804.The digital output from ADC is given as input to the microcontroller AT89C51 at port 2[2].

F. AT89C51 (ATMEL 89C51)

The digital value that is obtained from ADC0804 is given as input to AT89C51 and compared with preset value. If the preset voltage is greater than the digital voltage from the ADC0804 then the relay gets turned off. In case the magnitude of load voltage is greater than preset value then relay gets turned on and as a result load gets isolated from the supply [2].

VI. BASIC CONCEPT OF WORKING OF OVERVOLTAGE RELAY

For the competent working of all electrical and electronic devices, it is recommended to allow voltage at specified limits because voltage fluctuations in electric power supply will have adverse effects on connected loads. Over voltages are the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 11, November-2016

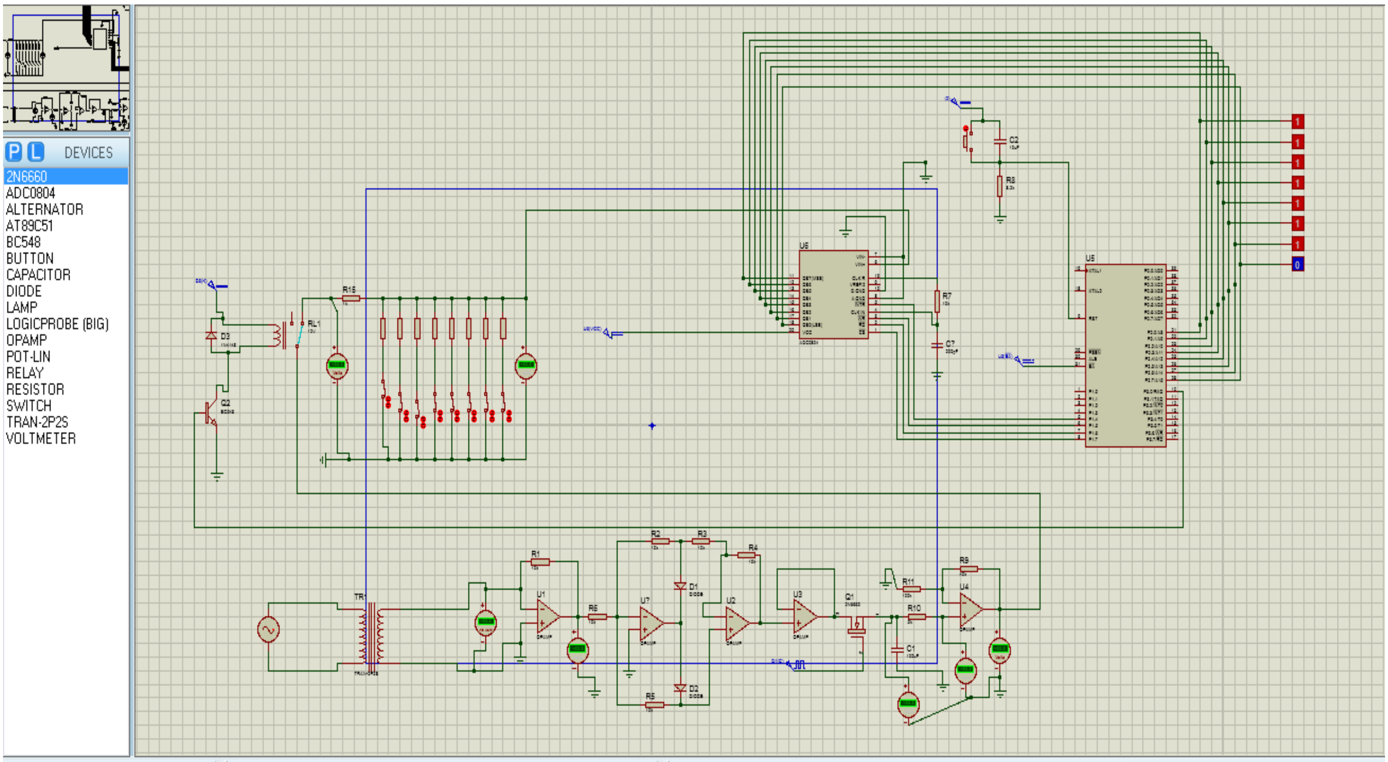

Fig: 2

voltages that exceed the normal or rated values which can lead to insulation damage in electrical appliances further provoking short circuits. They are caused by overloading, lightning strokes etc.

The principle of operation of relay is dependent on the load voltage. If load voltage exceeds beyond the preset value then relay should be switched ‘on’ to isolate load from the supply. This operation of relay is controlled using microcontroller AT89C51.The conversion of analog voltage to digital voltage using ADC0804 is controlled by AT89C51.

VII. COMPONENTS USED IN THE CIRCUIT

Program code is written in Keil micro vision software and simulation is done in Proteus software.
Components used in the circuit are listed below:

1. Voltage source – 6V, 50Hz
2. Transformer
3. Operational amplifier (Op amp)
4. Diodes – 1N4143
5. FET – 2N6660
6. BJT – BC548
7. Resistors and Capacitors of required values.
8. Relay – 12V
9. Lamps
10. ADC0804
11. AT89C51
12. Crystal Oscillator – 24MHz

VIII. DESCRIPTION OF THE CIRCUIT

1. The current from the source is converted to voltage using I to V converter
2. The Alternating voltage is converted to full wave rectified voltage using precision rectifier.
3. Sample and hold circuit samples the voltage of a continuously varying Analog voltage and holds it for a period of time as the variation in input signal can corrupt the conversion process in ADC0804.
4. ADC0804 is used to convert analog to digital value which is controlled by AT89C51.
5. The relay is operated using microcontroller by comparing the load voltage and preset voltage.
(The description is according to Fig: 2)

VIII. SIMULATION RESULTS

The circuit has been operated both for normal as well as for faulty operation.

- When the switches are opened (as shown in fig: 3) the voltage across the load falls below the preset value which is set at 5V in the program and hence the relay doesn't operate.
- When the switches are closed (as shown in fig: 4) and if the load voltage exceeds the preset value which is set at 5V in the program then the relay operates and it isolates the load.

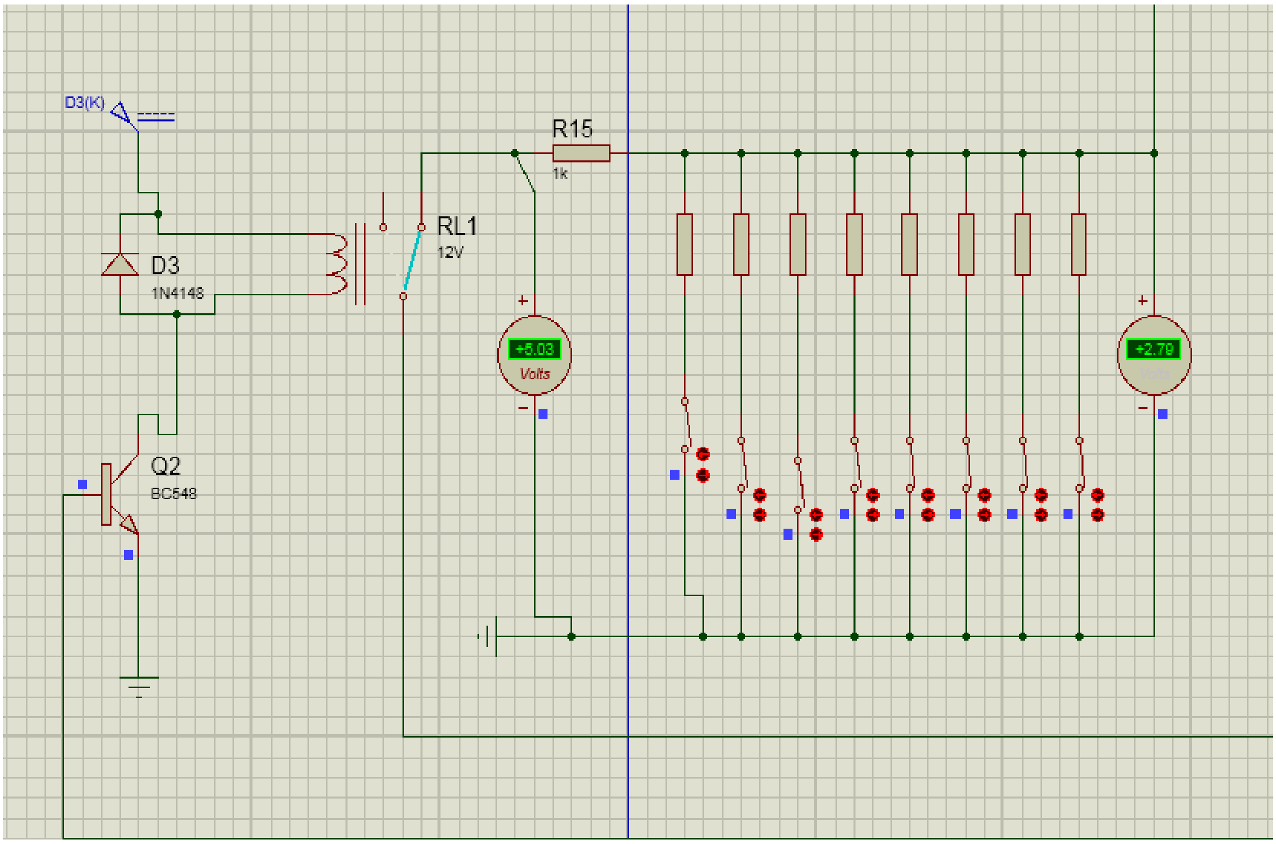

Fig: 3

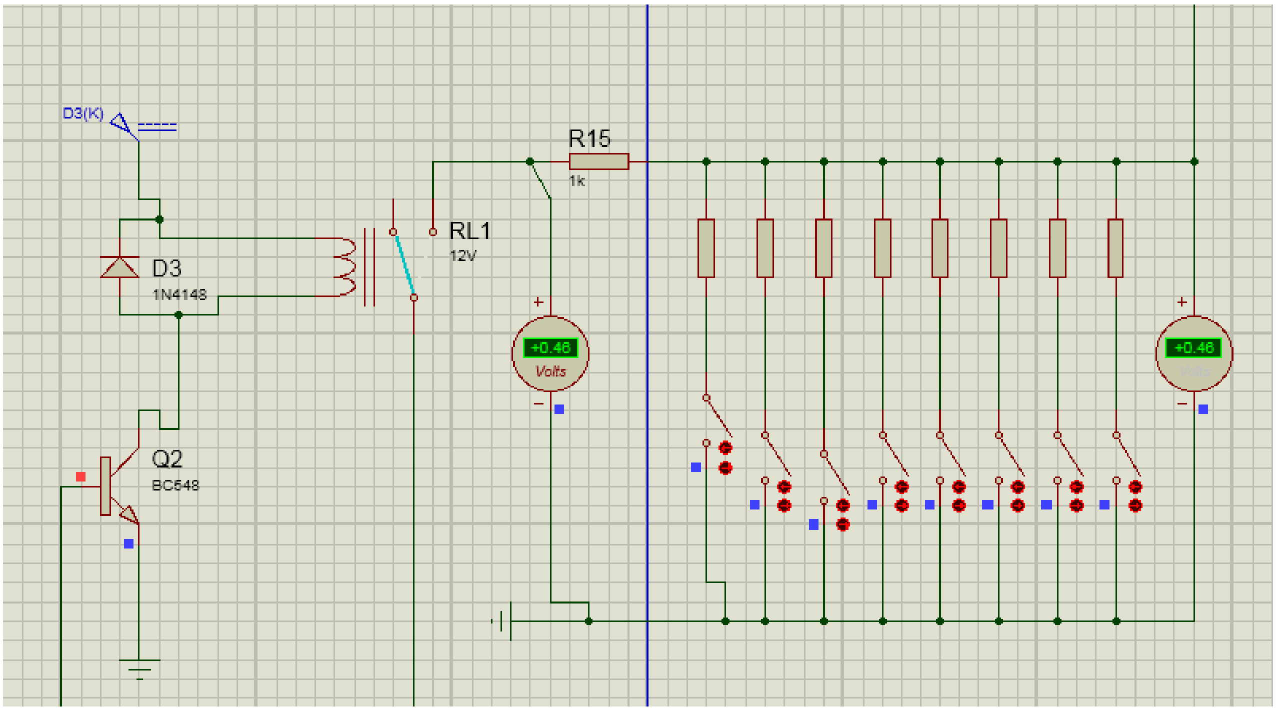

Fig: 4

IX. CONCLUSION

When load voltage is below the preset value the relay does not operate and current continues to flow through the load. Whereas when load voltage exceeds the preset value due to any faults or short circuit the relay is operated and load is isolated from the supply. As soon as the load voltage reaches normal operating conditions by rectifying the fault the relay is restored to its original position and allows the flow of current through the load.

This method of over voltage protection is preferred as it is highly reliable and not bulky in construction .The operation of relay takes place in fraction of seconds without any delay and the relay is operated even for small changes in the operating voltage and hence it is more reliable.

X. REFERENCES

[1] Bayindir R, Sefa I., Cola I., and Bektas A., (2008) "Fault detection and Load Protection Using Sensors", IEEE Transactions on Energy Conversion. (Volume: 23, Issue 3, pp.734-741).

[2] Badri Ram and D N Vishwakarma,''Power System Protection and switchgear,2nd Edition, Tata McGraw Hill education Pvt limited.

[3] TSM RAO, "Digital Numerical Relays", 1st Edition, Tata McGraw-Hill Education, 01-Jul-2005.

XI. APPENDIX

The program had been written in C language using Keil µ vision and then it has been loaded into the micro controller. The code given below has been implemented in the circuit model.

```
#include<reg51.h>
#define ADC P2
#define relayport P3
sbit R=P1^6;
sbit CS=P1^7;
sbit W=P1^5;
sbit INTR=P1^4;
sbit relay=P1^0;
void delay();
adc_read();
void delay(int a)
{
int i;
for(i=0;i<a;i++);
}
adc_read()
{
CS=0;
R=1;
W=0;
W=1;
while(INTR!=0)
{
}
CS=0;
R=0;
relayport=0x00;
if(ADC>0xFE)
while(INTR!=0)
{
relayport=0xFF;
}
}
main()
{
unsigned char i;
while(1)
{
adc_read();
i=P1;
P1=i;
delay(10000);
}
}
```