International Journal of Engineering Innovation & Research
Volume 3, Issue 4, ISSN: 2277 – 5668

A Novel Microcontroller Based Transmission Line Differential Protection

Majid Pakdel
Electrical Engineering Department,
University of Zanjan, Zanjan, Iran
Email: pakdel@mail.ru

Kazem Mazlumi
Electrical Engineering Department,
University of Zanjan, Zanjan, Iran
Email: kmazlumi@znu.ac.ir

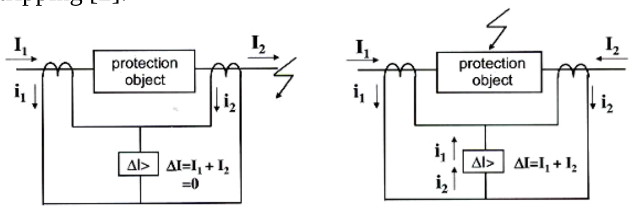
***Abstract* – As power systems grow both in size and complexity; it becomes common to use long and heavily two terminal lines as well as multi terminal and tapped lines. This in turn has created difficult problems for their protection such as distance relay under-reach and over-reach, effect of load current, high resistance faults and pilot wire limitations. Recent developments in digital differential protection have been aimed at overcoming these limitations. In this paper a novel PIC 16F88 microcontroller based transmission line differential protection is proposed. The operation of newly microcontroller based differential protection scheme is verified through simulation studies using Proteus software.**

***Keywords* – Microcontroller, Differential Protection, Transmission Line, Proteus Software.**

I. INTRODUCTION

The pilot relaying schemes are used for the protection of transmission line sections. In these schemes some electrical quantities at the two ends of the transmission line are compared and hence they require some sort of interconnecting channel over which information can be transmitted from one end to the other. Such an interconnecting channel is called a pilot. Three different types of such channels are presently in use, namely wire pilot, carrier-current pilot and microwave pilot. A wire pilot maybe buried private cables or alternatively rented telephone lines. A carrier-current pilot is one in which a low voltage high frequency signal (50kHz-700kHz) is used to transmit information from one end of the line to the other. In this scheme, the pilot signal is coupled directly to the same high voltage line which is to be protected. This type of pilot is also called a power line carrier. A microwave pilot is a radio channel of very high frequency, 450 to 10,000 MHz [1]. Wire pilot schemes are usually economical for distances up to 30 km. Carrier-current schemes are more economical for longer distances. When the number of services requiring pilot channels exceeds the technical or economical capabilities of carrier-current pilot, the microwave pilot is employed. A distance range up to 150 km is possible in a flat country. The microwave link may operate up to up to 40 to 60 km without repeater station. The system is applicable only where there is a clear line of sight between stations. The power requirement for signal transmission is less than a watt because highly directive antennas are employed [1].

The simplest and most frequently applied form of differential protection is shown in Fig. 1.The current transformers (CTs) at the extremities of the differential protection zone are connected in series on the secondary side so that the currents circulate through the CTs during an external fault (Fig. 1(a)) and no current flows through the differential measuring branch where the differential relay is situated. In the event of an internal fault (Fig. 1(b)) the fault currents flow towards the fault location so that the secondary currents add up and flow via the differential branch. The differential relay picks up and initiates tripping [2].



(a) (b)

Fig.1. (a) External fault, (b) Internal fault [2]

For line differential protection, the CTs at the two terminals of the protected line are far apart. In this case the connection circuit according to Fig. 2 is used (three core pilot differential protection). Three pilot wire cores are required for the connection between the two stations which typically are provided as a twisted triplet via a communication cable. Current differential relays are connected at both terminals in the differential core which in the event of an internal fault trip the circuit breakers in their respective stations [2].

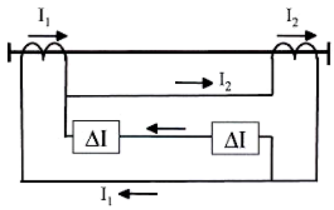


Fig.2. Line differential protection [2]

To further reduce the number of pilot wire cores required, the interposing current transformers (CTs) are also summation transformers, whereby the phase currents are combined to a single summated current as shown in Fig.3 [2]. Wire pilot protection includes the following schemes [1]:

a) Circulating current scheme
b) Balanced voltage scheme
c) Transley scheme (AEI)
d) Transley S protection
e) Half-wave comparison scheme

In general pilot wire based protection schemes are limited for the length below typically 30 km and they cannot be applied satisfactorily in many transmission circuits. The most widely used scheme for protection of EHV and UHV power lines is the carrier current protection which includes the following schemes [1]:

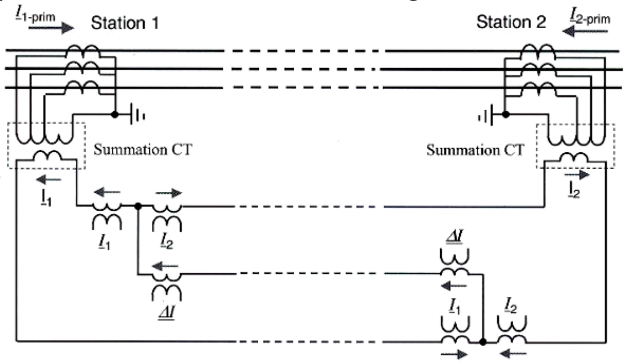


Fig.3. Three pilot wire core differential protection [2]

a) Phase comparison carrier current protection
b) Carrier aided distance protection
(i) Carrier transfer scheme
(1) Under-reaching scheme
(2) Permissive under-reach scheme
(3) Permissive over-reach scheme
(ii) Carrier acceleration scheme
(iii) Carrier blocking scheme

Optical fiber channels are used widely in recent years. Optical fibers are fine strands of glass. They behave as waveguides for light. As they are capable of transmitting light over considerable distance, they can be used to provide an optical communication link with enormous information carrying capacity. Optical fibers are embedded within the cable conductors, either earth or phase conductor. Optical fiber communication plays an important role in protection signaling nowadays [3-6]. Recent development in digital differential protection schemes have been aimed at overcoming the traditional scheme limitations. In this paper a novel PIC 16F88 microcontroller based transmission line differential protection is proposed. The operation of newly microcontroller based differential protection scheme is verified through simulation studies using Proteus software.

II. Case Study

As power systems grow both in size and complexity, it becomes common to use long and heavily two terminal lines as well as multi terminal and tapped lines. This in turn has created difficult problems for their protection such problems include:
(a) Distance relay under-reach
(b) Distance relay over-reach
(c) Effect of load current
(d) High resistance faults
(e) Pilot wire limitations

Recent development in digital differential protection schemes have been aimed at overcoming the traditional scheme limitations. The scheme can be effectively used in this manner is the current based digital differential protection scheme. The basic principle of line current differential protection used for two and multi lines is essentially the same as the percentage differential protection [7]. Consider the three-terminal line shown in Fig. 4. An instantaneous differential current signal D(t) is typically formed using the instantaneous currents at three ends such that:

$$D(t) = i_x(t) + i_y(t) + i_z(t) \qquad (1)$$

Where $i_x(t)$, $i_y(t)$ and $i_z(t)$ are the instantaneous currents measured at ends X, Y and Z, respectively. Under healthy conditions, ideally, the magnitude of the differential quantity D(t) should be zero. In practice, it has a small value both under normal operating conditions and external faults, due to line charging current and other errors such as a mismatch of current transformers. However, once an internal fault occurs, the differential quantity D(t) approximates to the fault current $i_f(t)$, i.e. the current flowing out of the fault point. Therefore we have:

$$\begin{cases} D(t) = 0 \text{ (Under normal and external fault conditions)} \\ D(t) = i_f(t) \text{ (Under internal fault conditions)} \end{cases} \qquad (2)$$

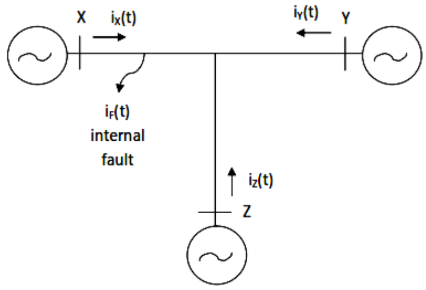


Fig.4. Three terminal line configuration

The relay configuration to protect a three terminal feeder is shown in Fig.5. In this case, two devices must be applied at each terminal, whereby the outside core of each relay pilot wire triplet is looped through the corresponding second relay at each terminal. Thereby the current from each of the three line terminals is available in each device for numerical computation of the stabilizing and tripping current and we have:

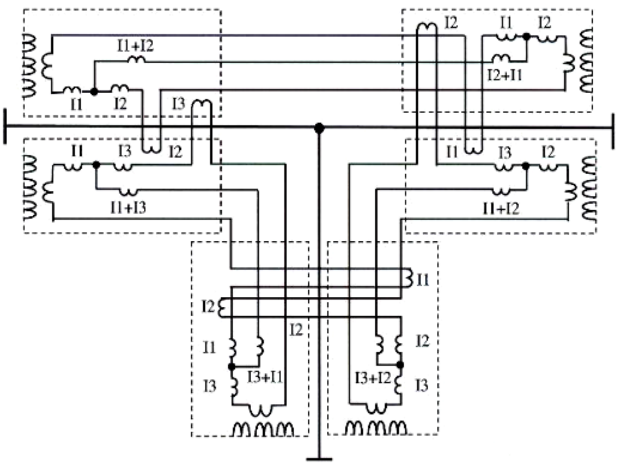


Fig.5. Three terminal protection with three pilot wire core differential protection scheme [2]

International Journal of Engineering Innovation & Research
Volume 3, Issue 4, ISSN: 2277 – 5668

$$I_{Res} = |I_1| + |I_2| + |I_3| \qquad (3)$$

$$I_{OP} = |I_1 + I_2 + I_3|$$

The tripping signals from the two devices at each terminal are connected in parallel [2]. The use of optical fiber cables in the three ended system is shown in Fig. 6, where the relays at each line end are digital/numerical relays interconnected by optical fiber links can be dedicated to the protection system or multiplexed in which case multiplexing equipment will be used to terminate the fibers. If I_A, I_B and I_C are the current vector signals at line ends A, B and C then for a healthy circuit we have:

$$I_A + I_B + I_C = 0 \qquad (4)$$

The basic principles of operation of the system are that each relay measures its local three phase currents and sends its values to the other relays. Each relay then calculates, for each phase, a resultant differential current and also a bias current, which is used to restrain the relay in the manner conventional for biased differential unit protection. The bias feature is necessary in this scheme because it is designed to operate from conventional current transformers (CTs) that are subject to transient transformation errors.

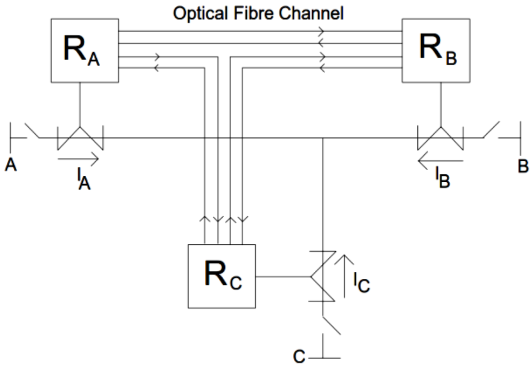


Fig.6. Current differential protection for three-terminal using optical fiber channel signaling

III. Proposed Microcontroller Based Differential Protection

The block diagram of the interface for realization of the proposed PIC 16F88 microcontroller based differential relay in a three terminal system is shown in Fig. 7. The current signals are fed through summation current transformers (CTs), then they are converted to fiber optic signals using analog to fiber optic converter and transmitted through fiber optic channels. At the receiver ends the fiber optic signals are converted to analog signals using fiber optic to coaxial converter. The received analog signals are added with together using an operational amplifier signals adding circuit. The output signal of adder is rectified and fed to the A/D converter pin (AN0) of PIC 16F88 microcontroller. The microcontroller reads the instantaneous magnitude of the received signal via the A/D converter pin (AN0) then this value is stored in the memory. The stored value which is equal to M is compared with the predetermined value of M_1 which remains stored in the memory. M_1 is a small value both under normal operating conditions and external faults, due to line charging current and other errors such as a mismatch of current transformers. If M is greater than M_1 the tripping signal is sent instantaneously through RB7 pin of the microcontroller. If M is less than M_1, the microcontroller goes back to the starting point, starts reading the input signal via the A/D converter pin (AN0) and again proceeds according the program. The output digital signal of RB7 pin is converted to optical fiber signal using a digital fiber optic link and the trip signals are transmitted via optical fiber channel to the three terminal system relays. The received signals at relay locations are converted to digital signals using digital fiber optical to digital coaxial converter then the output digital signals are fed to drivers to excite the trip circuit of respective circuit breakers. The program flowchart of the microcontroller is shown in Fig. 8.

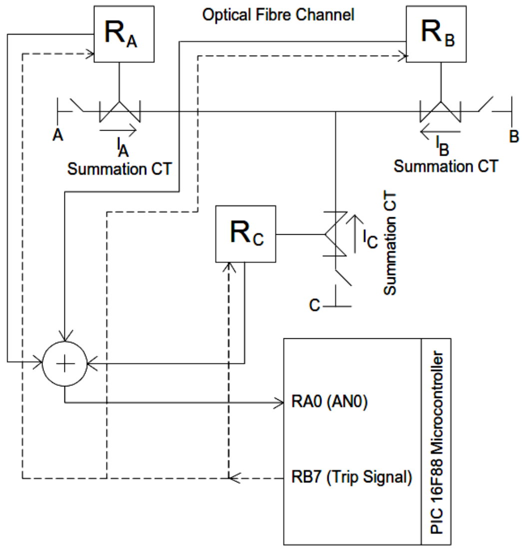


Fig.7. Block diagram of interface for the proposed differential relay

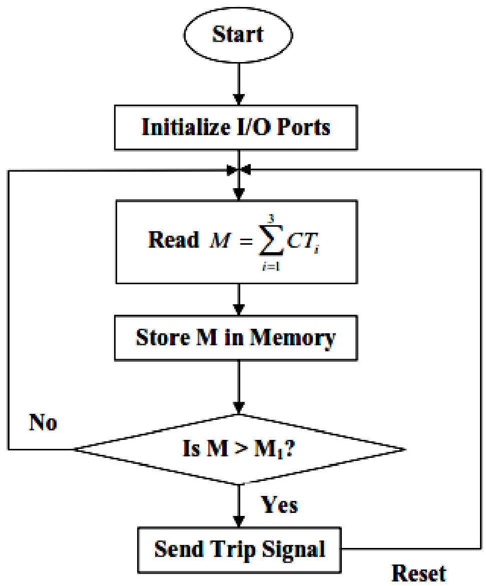


Fig.8. Program flowchart for proposed differential relay

International Journal of Engineering Innovation & Research
Volume 3, Issue 4, ISSN: 2277 – 5668

IV. Simulation Results

The implementation for the proposed mho relay in Proteus software which consists of a typical three terminal transmission line, three current transformers (CTs), an operational amplifier adding circuit and the PIC 16F88 microcontroller is depicted in Fig. 9. The levels of current signals are stepped down to the electronic levels by using auxiliary current transformers (CTs). The hardware and software described in section III for the implementation of the proposed differential relay can easily be executed on PIC 16F88 microcontroller. The written C code and the resulted hex code of the program algorithm are given in appendix A and B respectively. The source voltage amplitude is considered 325 V in simulation studies and its frequency is 50 Hz. Simulation results have been done in a single phase system and it can easily be extended for three phase systems. The levels of current signals are stepped down to the electronic levels by using three CTs. The CT ratios are selected 10A/1A and its burden is 0.5 Ω. A slave relay whose contacts are connected in series with the trip circuit of the circuit breaker is used to actuate the trip circuit after receiving the trip signal output from driver (ULN2003A) which is excited by microcontroller through an I/O port line (RB7) on occurrence of an internal fault.

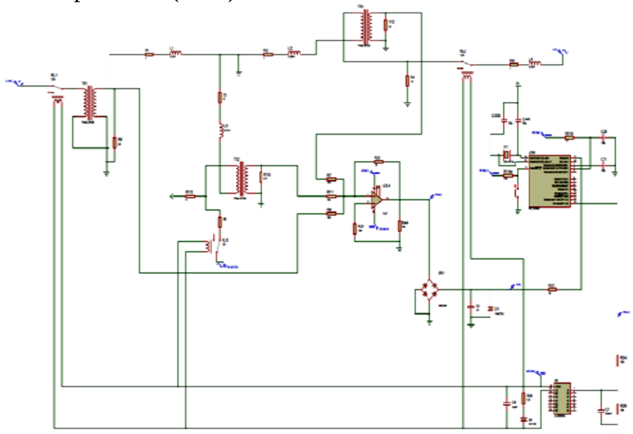


Fig.9. Implementation for the proposed differential relay in Proteus software

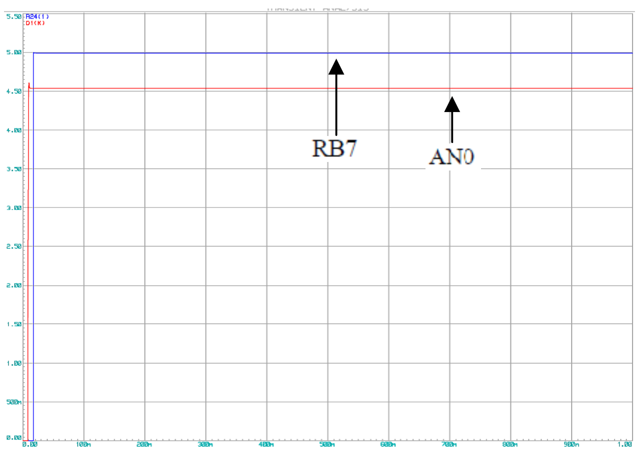


Fig.10. Waveforms of AN0 and RB7 for an internal fault [trip signal (RB7) is on at 20 ms]

The predetermined value for M_1 is equal to 1. The operation of the proposed differential relay for an internal and an external fault (under normal conditions) are shown in Fig. 10 and Fig. 11 respectively. As shown in Fig. 10, for an internal fault, the tripping signal is sent instantaneously. For an external fault and normal conditions as depicted in Fig. 11, the tripping signal is not sent and its magnitude is equal to zero.

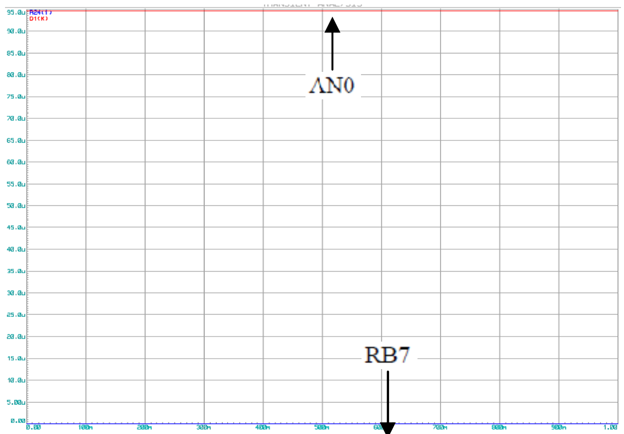


Fig.11. Waveforms of AN0 and RB7 for an external fault (normal conditions) [trip signal (RB7) is off]

V. Conclusion

In this paper the design and implementation for the protection of a typical three terminal transmission line using a novel microcontroller based differential relay was presented. Microprocessor based protective relays provide flexibility and are superior to conventional electromagnetic and static relays. As power systems grow both in size and complexity, it becomes common to use long and heavily two terminal lines as well as multi terminal and tapped lines. This in turn has created difficult problems for their protection such as distance relay under-reach and over-reach, effect of load current, high resistance faults and pilot wire limitations. Recent developments in digital differential protection have been aimed at overcoming these limitations. A novel PIC 16F88 microcontroller based transmission line differential protection was proposed in this paper. The operation of newly microcontroller based differential protection scheme was verified through simulation studies using Proteus software.

Appendix

A. *Microcontroller C code*

```
#define MX_PIC
//Defines for microcontroller
#define P16F88
#define MX_EE
#define MX_EE_TYPE2
#define MX_EE_SIZE 256
#define MX_SPI
#define MX_SPI_B
```

International Journal of Engineering Innovation & Research
Volume 3, Issue 4, ISSN: 2277 – 5668

```
#define MX_SPI_SDI 1
#define MX_SPI_SDO 2
#define MX_SPI_SCK 4
#define MX_UART
#define MX_UART_B
#define MX_UART_TX 5
#define MX_UART_RX 2
#define MX_I2C
#define MX_I2C_B
#define MX_I2C_SDA 1
#define MX_I2C_SCL 4
#define MX_PWM
#define MX_PWM_CNT 1
#define MX_PWM_TRIS1 trisb
#define MX_PWM_1 0
#define MX_PWM_TRIS1a trisb
#define MX_PWM_1a 3

//Functions
#define MX_CLK_SPEED 4000000
#ifdef _BOOSTC
#include <system.h>
#endif
#ifdef HI_TECH_C
#include <pic.h>
#endif

//Configuration data
#ifdef _BOOSTC
#pragma DATA 0x2007, 0x3f2a
#endif
#ifdef HI_TECH_C
__CONFIG(0x3f2a);
#endif
#ifdef _BOOSTC
#pragma DATA 0x2008, 0x3ffc
#endif
#ifdef HI_TECH_C
__CONFIG(0x3ffc);
#endif

//Internal functions
#include "E:\Program Files\Matrix Multimedia\Flowcode V4\FCD\internals.h"

//Macro function declarations

//Variable declarations
float FCV_IN1;

//Defines:

/**** Macro Substitutions ****
0 = Which ADC Channel
40 = Acquisition time
3 = Conversion Speed
0 = VRef+ Option
500 = VRef Voltage x 0.01V
******************************/

//ADC0: //Macro function declarations

void FCD_ADC0_SampleADC();
char FCD_ADC0_ReadAsByte();
short FCD_ADC0_ReadAsInt();
float FCD_ADC0_ReadAsVoltage();
void FCD_ADC0_ReadAsString(char* FCR_RETVAL, char FCR_RETVAL_SIZE);

//ADC0: //Macro implementations

void FCD_ADC0_SampleADC()
{
	/*******Supported Devices******************************************************************
	// 16F88
	*************************************************************************************/

	#define MX_ADC_CHANNEL		0
	#define MX_ADC_SAMP_TIME	40
	#define MX_ADC_CONV_SP		3
	#define MX_ADC_VREF_OPT		0

	//set up ADC conversion
	char old_tris, cnt;
	adcon1 = 0x00;

	//assign VREF functionality
	#if (MX_ADC_VREF_OPT == 1)
		st_bit(adcon1, VCFG1);
	#endif

	//find appropriate bit
	#if (MX_ADC_CHANNEL == 0)
		#define MX_ADC_TRIS_REG  trisa
		#define MX_ADC_TRIS_MSK  0x01
		ansel = 0x01;
	#endif
	#if (MX_ADC_CHANNEL == 1)
		#define MX_ADC_TRIS_REG  trisa
		#define MX_ADC_TRIS_MSK  0x02
		ansel = 0x02;
	#endif
	#if (MX_ADC_CHANNEL == 2)
		#define MX_ADC_TRIS_REG  trisa
		#define MX_ADC_TRIS_MSK  0x04
		ansel = 0x04;
	#endif
	#if (MX_ADC_CHANNEL == 3)
		#define MX_ADC_TRIS_REG  trisa
		#define MX_ADC_TRIS_MSK  0x08
		ansel = 0x08;
		#if (MX_ADC_VREF_OPT != 0)
			#pragma error "Target device is currently using AN3 for VREF+"
		#endif
	#endif
	#if (MX_ADC_CHANNEL == 4)
		#define MX_ADC_TRIS_REG  trisa
		#define MX_ADC_TRIS_MSK  0x10
		ansel = 0x10;
	#endif
	#if (MX_ADC_CHANNEL == 5)
		#define MX_ADC_TRIS_REG  trisb
		#define MX_ADC_TRIS_MSK  0x20
		ansel = 0x20;
	#endif
	#if (MX_ADC_CHANNEL == 6)
		#define MX_ADC_TRIS_REG  trisb
		#define MX_ADC_TRIS_MSK  0x40
		ansel = 0x40;
	#endif

	//sanity check
	#ifndef MX_ADC_TRIS_REG
		#pragma error "ADC Type 2 conversion code error - please contact technical support"
	#endif

	//store old tris value, and set the i/o pin as an input
	old_tris = MX_ADC_TRIS_REG;
	MX_ADC_TRIS_REG = MX_ADC_TRIS_REG | MX_ADC_TRIS_MSK;

	//assign conversion speed
```

```
#if (MX_ADC_CONV_SP > 3)
        st_bit(adcon1, ADCS2);
#endif

//turn ADC on
adcon0 = (0x01 | (MX_ADC_CONV_SP << 6)) |
(MX_ADC_CHANNEL << 3);

//wait the acquisition time
cnt = 0;
while (cnt < MX_ADC_SAMP_TIME) cnt++;

//begin conversion and wait until it has finished
adcon0 = adcon0 | 0x04;
while (adcon0 & 0x04);

//restore old tris value, and reset adc registers
MX_ADC_TRIS_REG = old_tris;
ansel = 0x00;
adcon0 = 0x00;

#undef MX_ADC_TRIS_REG
#undef MX_ADC_TRIS_MSK
#undef MX_ADC_SAMP_TIME
#undef MX_ADC_CONV_SP
#undef MX_ADC_VREF_OPT
#undef MX_ADC_CHANNEL

}

char FCD_ADC0_ReadAsByte()
{

        FCD_ADC0_SampleADC();

        return adresh;

}

short FCD_ADC0_ReadAsInt()
{

        short iRetVal;

        FCD_ADC0_SampleADC();

        iRetVal = (adresh << 2);
        iRetVal = iRetVal | (adresl >> 6);

        return (iRetVal);

}
float FCD_ADC0_ReadAsVoltage()
{

        int iSample;
        float fSample, fVoltage, fVperDiv;

        #define MX_ADC_VREF_V          500

        iSample = FCD_ADC0_ReadAsInt();

                //Read as 10-bit Integer

        #ifdef _BOOSTC
                fVoltage =
float32_from_int32(MX_ADC_VREF_V);
        //Convert reference voltage count to floating point
(0 - 500 x 10mV)
                fVoltage = float32_mul(fVoltage,
0.01);
    //Convert reference voltage count to actual voltage (0 - 5)
                fVperDiv = float32_mul(fVoltage,
0.000976);
    //Convert actual voltage to voltage per division (VRef / 1024)
                fSample =
float32_from_int32(iSample);
                //Convert to floating point variable
                fVoltage = float32_mul(fSample,
fVperDiv);
    //Calculate floating point voltage
        #endif

        #ifdef HI_TECH_C
                fVoltage = MX_ADC_VREF_V;

                //Convert reference voltage count to
floating point (0 - 500 x 10mV)
                fVoltage = fVoltage * 0.01;

                //Convert reference voltage count to
actual voltage (0 - 5)
                fVperDiv = fVoltage * 0.000976;

        //Convert actual voltage to voltage per division
(VRef / 1024)
                fSample = iSample;

                        //Convert to floating point
variable
                fVoltage = fSample * fVperDiv;

        //Calculate floating point voltage
        #endif

        #undef MX_ADC_VREF_V
        return (fVoltage);

}

void FCD_ADC0_ReadAsString(char* FCR_RETVAL, char
FCR_RETVAL_SIZE)
{

        float fVoltage;

        fVoltage = FCD_ADC0_ReadAsVoltage();
        FCI_FLOAT_TO_STRING(fVoltage, 2,
FCR_RETVAL, FCR_RETVAL_SIZE);    //Convert to String

}

//Macro implementations

void main()
{

    //Initialisation
    ansel = 0;
    cmcon = 0x07;


    //Interrupt initialisation code
    option_reg = 0xC0;


    //Loop
    //Loop: While 1
    while (1)
    {
        //Call Component Macro
        //Call Component Macro:
in1=ADC(0)::ReadAsVoltage
        FCV_IN1 = FCD_ADC0_ReadAsVoltage();
```

```
			//Decision
			//Decision: in1 > 1.0?
			if (FCV_IN1 > 1.0)
			{
				//Output
				//Output: 1 -> B7
				trisb = trisb & 0x7f;
				if (1)
					portb = (portb & 0x7f) |
0x80;
				else
					portb = portb & 0x7f;

			}

		}

		mainendloop: goto mainendloop;
}

void MX_INTERRUPT_MACRO(void)
{
}
```

B. *Microcontroller HEX code*

```
:02000000262CAC
:08000800FF00030E8312031335
:10001000B8000A0EB900040EBA008A110A124E2C5A
:1000200083120313E2014030E300BE08031D1A28C7
:100030000130E200E208031D30283E08033A031DA8
:100040002328E30130287F30E300C40803192D285A
:100050003E08023A0319E30130283E030319E30185
:100060002C08EA002D08EB002E08EC002F08ED000C
:100070006308EE00EF01F001F101E31F4228EF03F6
:10008000F003F1036E08EA076F0803186F0FEB0720
:1000900070080318700FEC0771080318710FED0753
:1000A0007F302C05E40000303C02031D5928FD3050
:1000B0003B02031CE6283B08FD3C3C080318031DDB
:1000C000BC1B63286B28FD303B0603193C08031D4D
:1000D0008228ED1F82282830BF04E308031D7A28F8
:1000E0007F30AE00FF30AC00AD00AF01FE30BB0092
:1000F000BC010800FF30BB00BC01AC01AD01AE018A
:10010000AF010800BC1FE628EF013D030319EF0A09
:10011000EE013C08803AF3007F307302031D922801
:10012000FF303B02031CEE0AF201EE080319EF0850
:10013000031DF20A80306D05F000F101F10AF008AC
:10014000031D7110E501F1080319F208031DE50A0A
:10015000BB080319BC080319DA283B09EE003C0967
:10016000EF00EE0A0319EF0A20306E026F08031841
:10017000BB280319BD28EF1FCE2870103B09F100E2
:10018000F10A2C1870140310AF0CAE0CAD0CAC0CB3
:10019000F103031DC12870182C14DA282C082D0433
:1001A0002E042F04FF390319DA280130AC00AD0109
:1001B000AE01AF01BB01BC017F302C05E400E508B6
:1001C0000319E628E4080319E6283F16E408031D8E
:1001D000BF166308EE00EF01F001F101E31FF32801
:1001E000EF03F003F1036E08AC076F0803186F0FFD
:1001F000AD0770080318700FAE0771080318710F70
:10020000AF070730EE00EE0803190D290310AF0CFD
:10021000AE0CAD0CAC0CEE03032940306406031D9C
:100220001429E208031D2C102C082D042E042F0481
:10023000FF39031D0800BB01BC0108008312031332
:100240002F1C26290230BB070318BC0AAE1F2C291D
:10025000BB0A031D2C29BC0A3B08AF000310AF0CDE
:10026000AE133B18AE17C408031DAF17080040308B
:1002700083120313E200EA01EB01EC01ED01AC0192
:10028000AD01AE01AF016208E300E203E308031928
:1002900084290310AC0DAD0DAE0DAF0DED1B2C146C
:1002A0000310EA0DEB0DEC0DED0DAB1F7C292408BE
:1002B000EA0725080318250FEB0726080318260F61
:1002C000EC0727080318270FED0727086D02031D09
:1002D000732926086C02031D732925086B02031D70
:1002E000732924086A0203187C29AC0A0319AD0A91
:1002F0000319AE0A0319AF0AAF1B84290310A80D16
:10030000A90DAA0DAB0D4329BC0120306202BB0030
:1003100031CBC03E21BBC036A086B046C046D0481
:10032000FF39031D2C140800831203132C082D041D
:100330002E042F04FF390319A9292F1B08000310CD
:10034000AC0DAD0DAE0DAF0DBB080319BC03BB0367
:100350009D29BB01BC010800831603139F010130D6
:100360009B0005088312D600013083160504850022
:10037000C13083129F00D701283057020318C229C9
:10038000D70ABC2904301F049F001F19C52956082D
:10039000831685009B0183129F0108005408A40066
:1003A0005508A5005608A600A613A7015708C000C7
:1003B000C0130310C00DD61B40145808A8005908DC
:1003C000A9005A08AA00AA13AB015B08C100C11317
:1003D0000310C10DDA1B4114C201D71FF1290130EE
:1003E000C200C301DB1FF6290130C30043084206E7
:1003F000C4005C10400A03195C14DC10410A0319A4
:10040000DC145C112408250426042704FF39031991
:100410005C15DC11280829042A042B04FF39031970
:10042000DC155C18152ADC1C5B2A5C1C1D2A5C1977
:100430001D2A561B1D2A3F144D2ADC1C252ADC19B7
:10044000252A5A1B252A3F144D2A5C1C292A5C1D8B
:100450004D2ADC1C2D2ADC1D4D2ADC1D342AC10846
:1004600031D342A3F144D2A5C1D3B2AC008031D7E
:100470003B2A3F144D2AE201E3017F30E500803042
:10048000E400C408031DE5176508E9006408E800F6
:100490006308E7006208E6000800E201E301C030FB
:1004A000E400FF30E5006508E9006408E80000307A
:1004B000E700E60008007F30DD00C008031D7C2A4D
:1004C0005C1D722AE201E301E501E401C408031D99
:1004D000E5176508E9006408E8006308E7006208BA
:1004E000E6000800DD03A61B7D2A0310A40DA50D60
:1004F000A60DA70DDD0A732AA617C108031D9C2AA5
:10050000DC1D922AE201E301E501E401C408031DB8
:10051000E5176508E9006408E8006308E700620879
:10052000E6000800DD03AA1B9D2A0310A80DA90DF3
:10053000AA0DAB0DDD0A932AAA170730E200E208E4
:100540000319A92A0310A40DA50DA60DA70DE203FA
:100550009F2A0830E200E2080319B52A0310A80D0B
:10056000A90DAA0DAB0DE203AB2A3721BB08031975
:10057000BC080319E82A3B09E2003C09E300E20A4F
:100580000319E30A2030620263080318C92A031919
:10059000CB2AE31FDC2A5C123B09E400E40A2C1896
:1005A0005C160310AF0CAE0CAD0CAC0CE403031DD9
:1005B000CF2A5C1A2C14E82A2C082D042E042F04B0
:1005C000FF390319E82A0130AC00AD01AE01AF01DB
:1005D0004008BB00BC014108BB070318BC0A5D080A
:1005E000BB02031CBC032F1BFE2A0310AC0DAD0D78
:1005F000AE0DAF0DBB080319BC03BB0310201E21B9
:100600002C08DE002D08DF002E08E0002F08E10096
:100610006108E9006008E8005F08E7005E08E6009E
:1006200008005408AC005508AD005608AE00570845
:10063000AF002C08003C031D2F2B2D08003C031D90
:100640002F2B2E08003C031D2F2B2F08803C031D51
:100650002F2B0130C4009E30BB00BC01542BC401C1
:10066000AF1F4F2B0130C4002C08003CDC000030D1
:10067000DD00DE00DF002D08031C2D0FDD022E083B
:1006800031C2E0FDE022F08031C2F0FDF025C0855
:10069000AC005D08AD005E08AE005F08AF009C30A6
:1006A000BB00BC01942110201E212C08D8002D086D
:1006B000D9002E08DA002F08DB005B08DF005A089B
:1006C000DE005908DD005808DC000800AC211E08D7
:1006D000D400D501D40DD50DD40DD50DFC30D405E5
:1006E00083161E0C8312D600D60CD60E0330560588
:1006F000D404D5085408D6005508D700080066234E
:100700005608C6005708C700F430D4000130D500A1
:10071000D601D70111235C08CC005D08CD005E082E
:10072000CE005F08CF004C08D4004D08D5004E081D
:10073000D6004F08D7000A30D800D730D900233070
:10074000DA003C30DB00CE216608CC006708CD0023
:100750006808CE006908CF004C08D4004D08D500C9
:100760004E08D6004F08D7004030D800DA30D90004
:100770007F30DA003A30DB00CE216608D00067080F
:10078000D1006808D2006908D3004608D4004708A1
```

:10079000D500D601D701C71FCF2BD603D70311230E
:1007A0005C08C8005D08C9005E08CA005F08CB008D
:1007B0004808D4004908D5004A08D6004B08D7009D
:1007C0005008D8005108D9005208DA005308DB005D
:1007D000CE216608CC006708CD006808CE00690805
:1007E000CF004C08D4004D08D5004E08D6004F0865
:1007F000D7000800831603139B0107309C00C0300C
:1008000081007F235408B4005508B5005608B6008F
:100810005708B70037083F3C031D182C3608803CAA
:10082000031D182C3508003C031D182C3408003C0F
:100830000318012C7F308316060586007F30831253
:100840000605C600803046048600012CD530831290
:100850000313A000C430A100BB30A200DC30A30011
:10086000A401A501A601A701A801A901AA01AB0144
:10087000AC01AD01AE01AF01C001C101BB01BC01C2
:10088000C201C301C401C501B001B101B201B3018C
:10089000BE01BF01BD018A110A12FA2B8312031394
:1008A0003A0E8400390E8A00380E8300FF0E7F0E48
:0208B00009003D
:02400E002A3F47
:02401000FC3F73
:00000001FF

REFERENCES

[1] B. Ram, D. N. Vishwakarma, Power System Protection and Switchgear, McGraw-Hill Publishing Co., 12th reprint, 2004.

[2] G. Ziegler, Numerical Differential Protection, Principles and Applications, Publicis Corporate Publishing, Erlangen, 2005.

[3] Law, C.T. and Mueller, A.D., "Novel fibre-optics-based current sensor for power system protection,", Power and Energy Society General Meeting, 2012 IEEE, pp. 1-5.

[4] Gartia, A. ; Gartia, A. ; Gulati, A. ; Kumar, C., "Microcontroller based line differential protection using fiber optic communication,", Innovative Smart Grid Technologies - Asia (ISGT Asia), 2013 IEEE, pp. 1-4.

[5] Orr, P, Niewczas, P. ; Dysko, A. ; Booth, C., "FBG-based fibre-optic current sensors for power systems protection: Laboratory evaluation,", Universities Power Engineering Conference (UPEC), 2009 Proceedings of the 44th International, pp. 1-5.

[6] Tuan-jie Cao, Chang-zhen Dai ; Jian-yu Chen ; Zhuo-fei Yu, "A new method for channel monitoring for fibre optic line differential protection,", Electricity Distribution, 2008. CICED 2008. China International Conference on, pp. 1-4.

[7] A. T. Johns and S. K. Salman, Digital Protection for Power Systems, Peter Peregrinus Ltd., 1995.

AUTHOR'S PROFILE

Majid Pakdel was born in Mianeh, Iran, 1981. He received his B.Sc. and M.Sc. degrees from Amirkabir University of Technology and Isfahan University of Technology, Iran, in 2003 and 2006, respectively all in electrical engineering. He is currently a PhD student in department of electrical engineering, University of Zanjan, Zanjan, Iran. His research interests include power electronics, power quality, power system protection and renewable energies.

Kazem Mazlumi was born in Tehran, Iran, in 1976. He received the B.Sc. degree in Electrical Engineering from Amirkabir University of Technology, Tehran, Iran, in 2000, the M.Sc. degree from Sharif University of Technology, Tehran, Iran, in 2003, and the Ph.D. degree from Amirkabir University of Technology, in 2009. He is currently an Assistant Professor with University of Zanjan, Zanjan, Iran. His research interests include power system protection, power system analysis, power system reliability, and renewable energies.