Technical Strategy to Curb Transformer Oil Theft on Distribution Networks: Case of Uganda's Power Distribution Network

Patrick Kabanda
Department of Electrical and Computer Engineering
Kampala International University
Kampala, Uganda

Samson Ssemakalu Ttondo
Department of Electrical and Computer Engineering
Makerere University
Kampala, Uganda

***Abstract*—The theft of transformer oil and windings is common on the electric power distribution network in Uganda, and the utility incurs high losses due to this vice. This paper presents a design of the proposed system that can be used to reduce on distribution transformer vandalism (aiming at stealing transformer oil). The stolen oil is used to cool and repair welding machines. The parameter of interest to be monitored by the proposed system is the transformer oil level which changes as the oil is drilled from the transformer. The system makes use of a float switch which is open when the level is normal and makes contact when the level falls due to abnormal leakage. This sends a voltage to the voltage regulator circuit which then supplies the alarm system (buzzer and light emitting diode) to scare off the vandals as well as provide an input signal to the microcontroller to condition an input to the GSM Modem. The GSM Modem then sends a text message to the technical officer to take action. This action involves opening the high voltage (HV) feeder remotely to avoid instances of supplying power to the transformer when the oil has been drained. The modeling and experimentation results are presented.**

***Keywords*— Distribution Transformer; Float Switch; GSM Modem; Microcontroller; Vandalism; Voltage Regulator**

I. INTRODUCTION

There are over 12,000 distribution transformers on the network managed by UMEME, a current local utility power company. Distribution transformers are the access points for the electric power consumers. It is therefore important to guard them against vandalism to ensure quality service delivery to the customers by minimizing on power outages as well as ensuring better revenue collection by power distribution utility company.

When a transformer is vandalized by drilling oil, it heats up, or even burns the expensive substation equipment. When these occur, the transformer trips hence long power outages imply loss of revenue to the utility company [1, 2].

According to the utility company, vandalism is ruining its business, with vital resources being diverted from upgrading the network into replacing stolen and vandalised equipment. A single 50kVA transformer provides electricity access to about 2,000 domestic clients on average, and replacing one costs the utility company US$12,000 [3]. In addition, most transformers are usually tampered with before they break down. This expenditure in most times is unbudgeted for; meaning resources meant to ensure provision of other vital services, that is, upgrading the network by the utility company are diverted to replacements. This raises the need for a system to reduce the vice by monitoring the oil level, off-setting the buzzer and sending a message to the technical officer. This system demonstrates the possibility to use sensor technology to monitor the level and flow of liquids in case of abnormal leakage.

II. DESIGN CONCEPT

The system proposed to reduce on distribution transformer vandalism gives out an audible alarm (Buzzer and a Light) as well as communication between the transformer and Technical Staff in charge of the transformer. The Technical Staff should then be able to communicate with the Control Centre to open the feeder remotely, to avoid the transformer being supplied with power when it has no oil. The alarm system consisting of the Buzzer and a Light should be able to alert the community about the state of the transformer.

This system therefore requires the following in order to be effective;

- Oil level monitoring device that can detect a fall in the level due to vandalism
- A stand-alone power supply independent of the power from the transformer
- A reliable alarm system

A reliable communication system to the Technical Staff over the existing GSM Network.

The communication between the vandalised transformer and the Technical Staff is by means of an SMS.

The model of the designed system is illustrated in Fig. 1.

III. SYSTEM IMPLEMENTATION

The system is supplied by a rechargeable stand-alone 12V DC battery. When a vandal drills the transformer casing to get the oil, a float switch previously floating on the oil, is tilted and its terminals make contact. This provides a return path for the current to the system. In the system, there is a 12V DC relay that supplies a voltage regulator circuit, which is designed to output 5V and 3V DC respectively. The 5V powers the Alarm system, while the 3V is an input signal to the microcontroller unit (MCU).

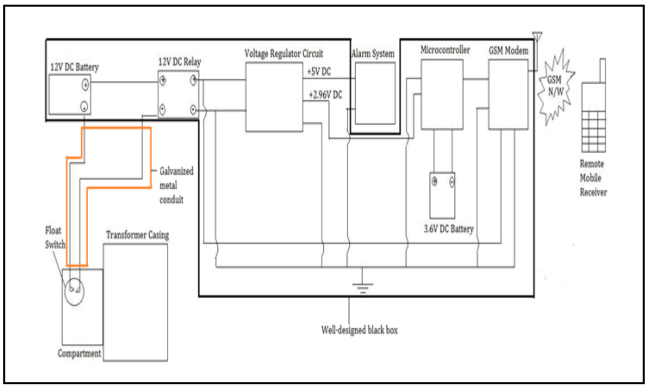

Fig. 1. Design layout of the proposed system

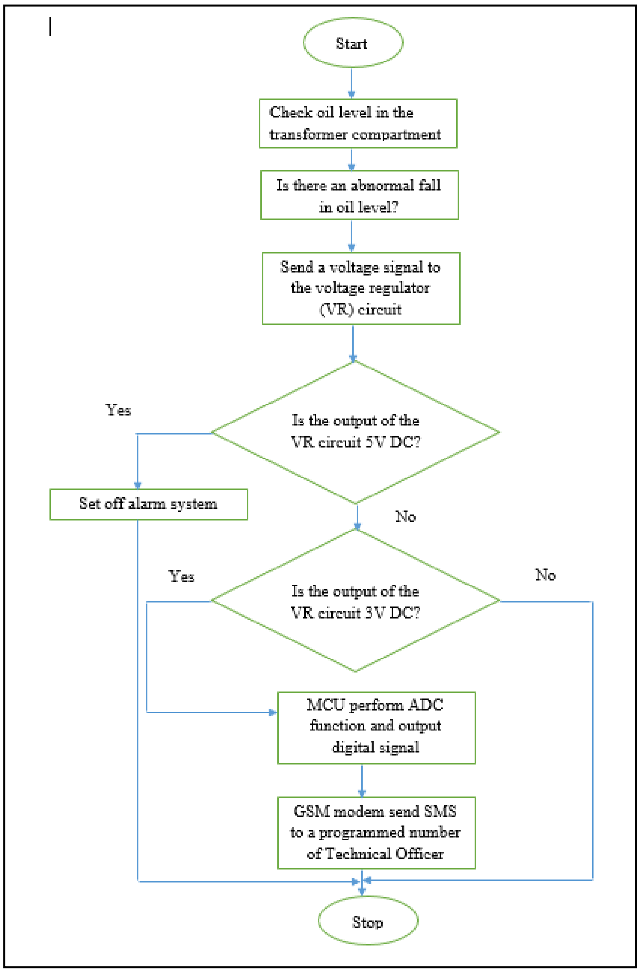

Fig. 2. Flow chart showing the sequence of operation of the system

The microcontroller performs an analogue to digital conversion, and the digital signal is sent to the GSM mobile station on the Technical Staff's phone.

The microcontroller is powered by a standalone rechargeable 4V DC battery (similar to that used in a mobile phone) while the GSM Modem is powered by the system's 12V DC battery.

The flow chart shown in Fig. 2 summarises the sequence of operation of the entire system.

A. Oil Level Monitoring

The transformer casing was fabricated to hold the oil to be used in testing the system. An extra compartment was added to the casing in order to avoid the system from getting into contact with the windings of the transformer. A hole was drilled in the casing to demonstrate a vandalism case.

The float switch level changed with the water level as it drilled out hence creating a return path for the current flowing from the 12V DC battery.

Fig. 3. Float switch inside a fabricated transformer compartment.

B. Voltage Regulation

Using MultiSIM software [4], this circuit was simulated and the battery input voltage was **12V**, output from the regulator was **5.002V** and 3.125V as the voltage supply to the microcontroller.

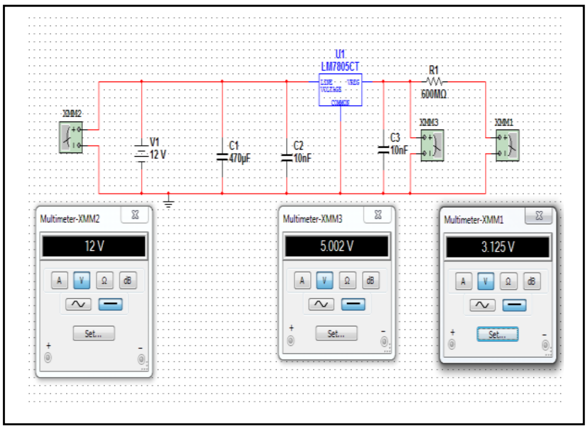

Fig. 4. Simulated voltages using MultiSIM

The 12V DC input was supplied to the LM 7805 via the relay connection [5]. The actual input was measured to be 12.29V, while the output was measured as 4.98V. The input into the microcontroller was 2.96V which was satisfactory in line with the expectations of the design – see Fig. 5.

The voltage levels at the various terminals of the voltage regulator (VR) are shown in Fig. 6.

Fig. 5. Measured battery and input voltage to the microcontroller unit (MCU).

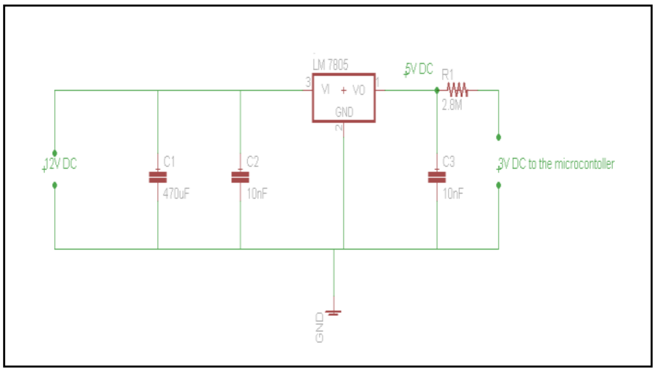

Fig. 6. Voltage levels at the various terminals of the voltage regulator

C. Alarm System

The Buzzer was connected to the 12V DC supply, while the LED to the 5V DC. In the case of normal operation (when there is no hole on the casing) the Buzzer and the LED give no output. However, when the hole on the casing was opened (implying a vandalism case) the Buzzer sounded and the LED lit, hence an alarm was set off.

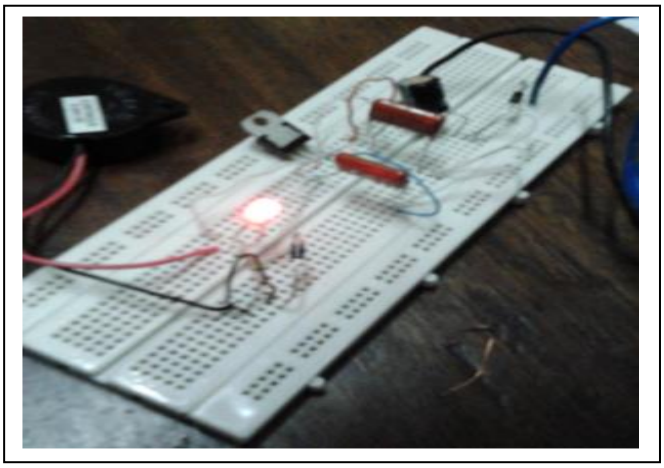

Fig. 7. Alarm system (buzzer and LED) in operation

D. Microcontroller Operation

The microcontroller deployed in this design is the MSP430G2231 of the MSP430 series of microcontrollers. This microcontroller is built around a 16-bit CPU and is designed for low cost, specifically, low power consumption embedded applications [6].

A code was written in C programming language using the CCS IDE to control the operation of the Microcontroller to achieve the required functionality [7]; details of the code are shown in Appendix A. The flow chart is shown in Fig. 8.

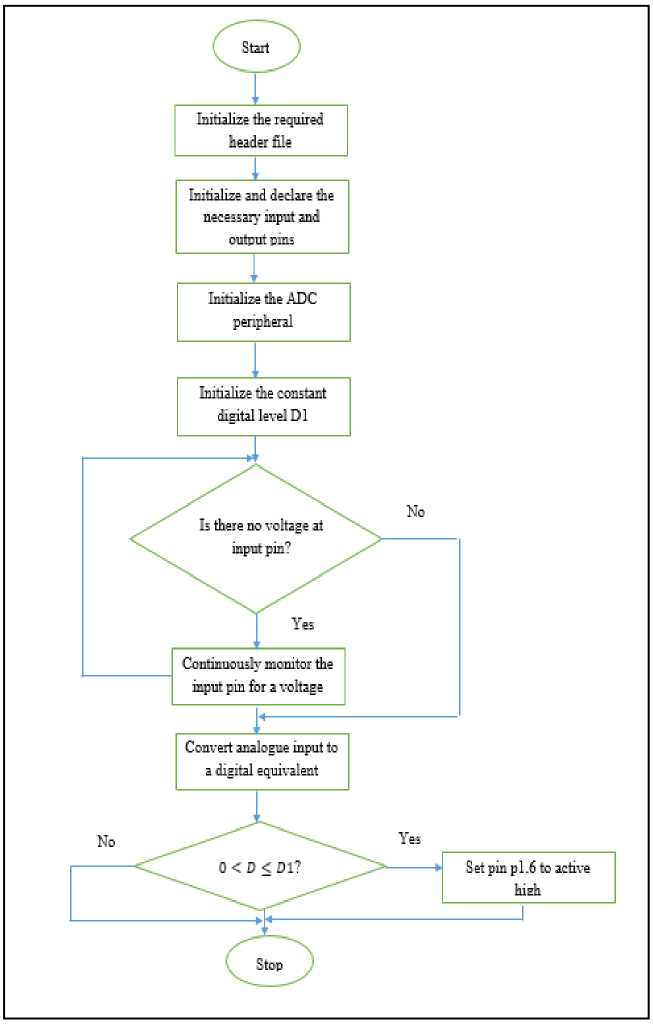

Fig. 8. Flow chart for the microcontroller operation.

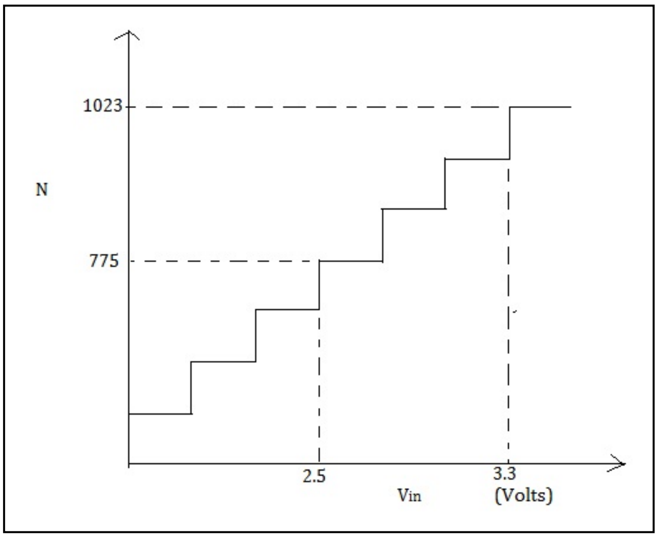

Fig. 9. Graph showing the relationship between *N* and V_{in}

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 05, May-2018

The formula for converting the input voltage to a discrete value is given by

$$N = \frac{N_{max}(V_{in})}{V_{ref}} \quad (1)$$

Where $N_{max} = 2^{10} - 1 = 1023$; since the microcontroller has a 10-bit ADC channel and $V_{ref} = 3.3V\ DC$. N is the digital equivalent that the microcontroller can read. For example, if $V_{in} = 2.5V\ DC$, then the digital equivalent N = 775. Fig. 9 illustrates this analogue-to-digital conversion (ADC).

Fig. 10. Programmer board showing LEDs due to the microcontroller operation.

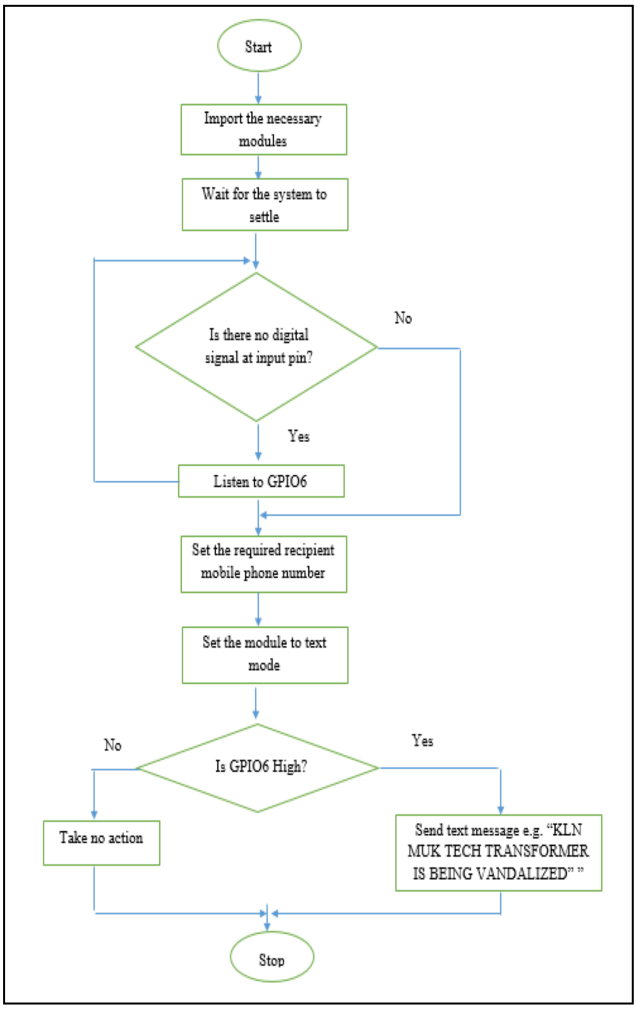

Fig. 11. Flow chart for the GSM modem operation.

E. GSM Modem Operation

The GSM modem deployed in this design is the GT863-PYof the Telit wireless solutions [8].

The GSM modem was programmed using PYTHON programming language so as to achieve the desired functionality [9]. A flow chart showing the operation of the Modem is shown in Fig. 11 and the detailed code in Appendix B. This code was then downloaded onto the GSM Module using an RS232 cable as interface with the Computer.

F. Message on the Phone

When all the system components had been connected the system was tested. The message was received as expected. It stated "KLN/MUK TECH TX IS BEING VANDALISED". This statement literally means that "the distribution transformer at Makerere University Faculty of Technology which is fed by the Kampala North feeder is being vandalized".

Fig. 12. Message received on mobile terminal (phone).

G. Entire System Layout

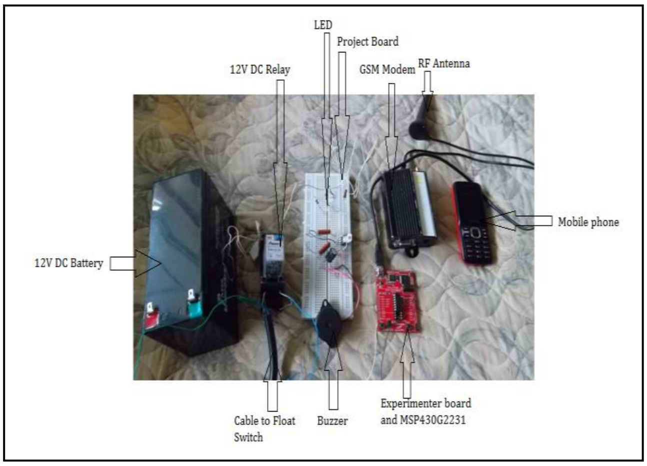

Fig. 13. Layout of the entire system

IV. CONCLUSION

This paper presented the design and prototype of a system used to reduce distribution transformer vandalism by detecting a fall in transformer oil levels. An alarm buzzer and SMS platform were successfully tested and were able to signal to the main control Centre and sound an alarm. This would ensure that customers have access to electricity without any inconveniences due to a vandalized transformer as has been the case in most areas in Uganda today. In addition, no customers would face any risks of being electrocuted in the process of vandalizing the distribution transformer.

However the limitations to this system are that there are delays in relaying the SMS to the mobile phone majorly due to inconsistencies in the GSM Network and incompatibility of local networks' service providers' SIM cards with the GSM Modem. It's recommended that utility companies use GSM Modems that are compatible with the local networks' service providers' SIM cards.

Also, the transformer mains, when available should be used to charge the batteries; this entails developing an intelligent charging system.

REFERENCES

[1] Moses, T. (2011). *UMEME Business* . Kampala: Observer)

[2] Mafabi, D. (4 May 2015). Umeme Decry Rampant Vandalism, Power Theft, Daily Monitor.

[3] Electricity Regulatory Authority (ERA) Uganda. (24 October 2011). Study on Distribution System Losses and Collection Rates by UMEME Ltd. Parsons Brinckerhoff Africa (PTY) Ltd.

[4] Luther, E., & Rodriquez, J. (2006, September 26). Introduction to Multisim Schematic Capture and SPICE Simulation Retrieved July 4, 2012, from connexions: http://cnx.org/content/col10369/1.3/.

[5] Texas Instruments. (2003). uA7800 Positive-Voltage Regulators. Dallas, Texas: Texas Instruments Incorporated.

[6] Texas Instruments. (October 2011). Getting Started with the MSP430 (pp. 26-28). Texas Instruments Incorporated.

[7] Stroustrup, B. (1997). The C++ Programming Language 3rd Edition. Murray Hill, New Jersey: Addison Wesley Longman, Inc.

[8] Telit communications: S.P.A. (2008). GT863-PY Terminal Product Description. Telit Wireless Solutions.

[9] Chun, W. J. (December 14, 2000). Core Python Programming 1st Edition. Prentice Hall.

Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 05, May-2018

APPENDIX A: MSP430 MODULE CODE

```
//this code uses an ADC to accept an input at p1.5, convert it to a digital value and then
//perform different operations depending on the result
//the first piece of the code includes the header file, makes a few definitions, declarations
//and initialisations
//Patrick Kabanda and Samson Ssemakalu Ttondo
#include <msp430g2231.h>
#define     LED1                BIT0
#define     LED2                BIT6
Unsigned int value=0;
//the next lines of code are basically configuring the ADC
void ConfigureAdc(void)
{
  // Configure ADC  Channel //
  ADC10CTL1 = INCH_5 + ADC10DIV_3;        // Channel 5, ADC10CLK/4
  ADC10CTL0 = SREF_0 + ADC10SHT_3 + ADC10ON + ADC10IE; //Vcc & Vss as reference
  ADC10AE0 = BIT5;                        //P1.5 ADC option
}
//the next lines of code are for the main function in which we stop the watch dog timer,
//set the input pin of the ADC,
//set the pin dirrecti0ns and set up the timers
void main(void)
{
  WDTCTL = WDTPW + WDTHOLD;               // Stop WDT
  BCSCTL1 = CALBC1_1MHZ;                  // Set range
  DCOCTL = CALDCO_1MHZ;
  BCSCTL2 &= ~(DIVS_3);                   // SMCLK = DCO = 1MHz
  P1DIR |= (LED1 + LED2);      //set both pins p1.0 and p1.6 as output
  P1SEL |= BIT5;
  //P1IN = P1IN & 0x20;                   // Input pin P1.5
  P1OUT &= ~(LED1 + LED2);
  ConfigureAdc();
  __enable_interrupt();                   // Enable interrupts.
//we wait some cycles to let the ADC voltage reference settle to a stable level and
//then we start the conversion by setting the ADC10CTL0 register.
//Then we enter a low-power mode to save juice:
//we'll have the result of the conversion directly in our "value" variable from
//the ADC10MEM register thanks to the ADC10 interrupt.
//At last we simply toggle the leds on the LaunchPad according to
//the number stored into "value"
//(don't forget that we have a 10-bit ADC module, so the maximum value will be 1023)
while(1)
{
    __delay_cycles(1000);                 // Wait for ADC Ref to settle
    ADC10CTL0 |= ENC + ADC10SC; // Sampling and conversion start
    __bis_SR_register(CPUOFF + GIE);      // LPM0 with interrupts enabled
    value = ADC10MEM;
//while (ADC10CTL1 & ADC10BUSY);   // Wait for ADC Conversion
//return(ADC10MEM);                       // Return ADC Value
//the formular for converting the input voltage to a discrete value N
//N=1023(vin)/(Vref) where Vref =3.3v
//if the digital equiavalent value N=775 i.e 2.5v
//the transformer oil is below the recommended levels
    if(value>775)
    {
      P1OUT &= ~(LED1 + LED2);//turn off both leds 1 and 2
      //P1OUT &= ~(LED2);//turn led 2 off
      P1OUT |= LED1;//turn led 1 red on
      }
      else //the oil levels are fine
      {
              P1OUT &= ~(LED1 + LED2);
      P1OUT |= LED2;//turn led 2 green on
      //P1OUT &= ~(LED1);//turn led 1 off
      }
}//marking the end of the while loop
}//marking the end of the main function
// ADC10 interrupt service routine
#pragma vector=ADC10_VECTOR
__interrupt void ADC10_ISR (void)
{
  __bic_SR_register_on_exit(CPUOFF);      // Return to active mode
}
```