VASAVI COLLEGE OF ENGINEERING (AUTONOMOUS)
IBRAHIMBAGH, HYDERABAD
Department of Electronics and communication Engineering

Microprocessors and Microcontrollers

Activity : Simulation and case Study

Display temperature (lm 35) and moisture (pot-lin) values on lcd (16*2) (first row : temperature and second row: moisture) by interfacing 0808adc ic with 8051 microcontroller and whenever either temperature value or moisture value crosses threshold value (set as you like) indicate danger on lcd and give buzzer sound.

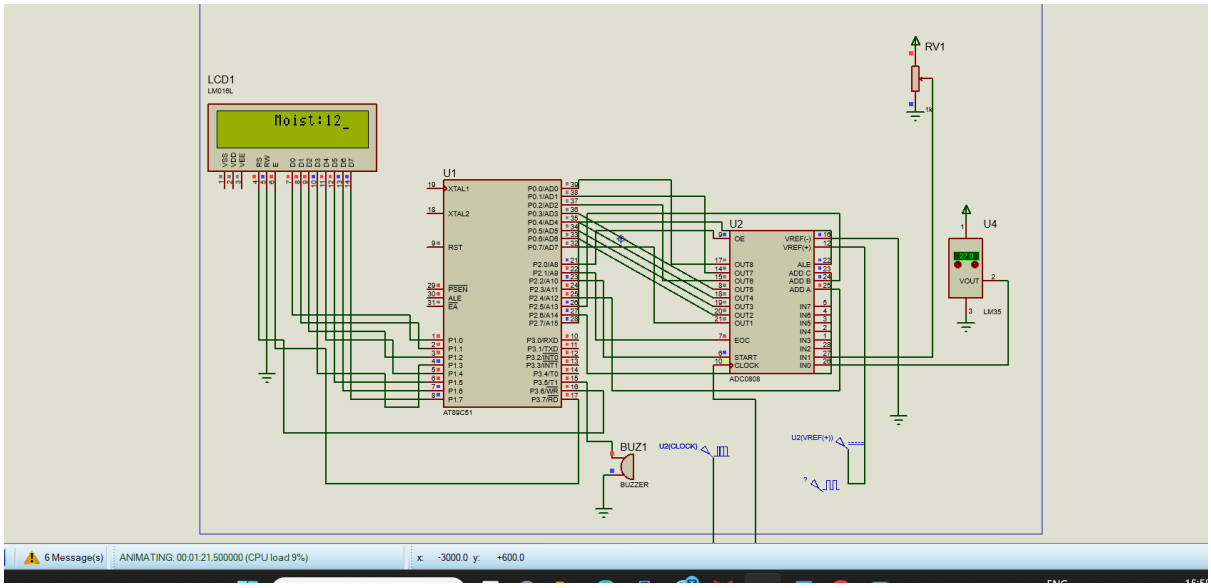
Program:

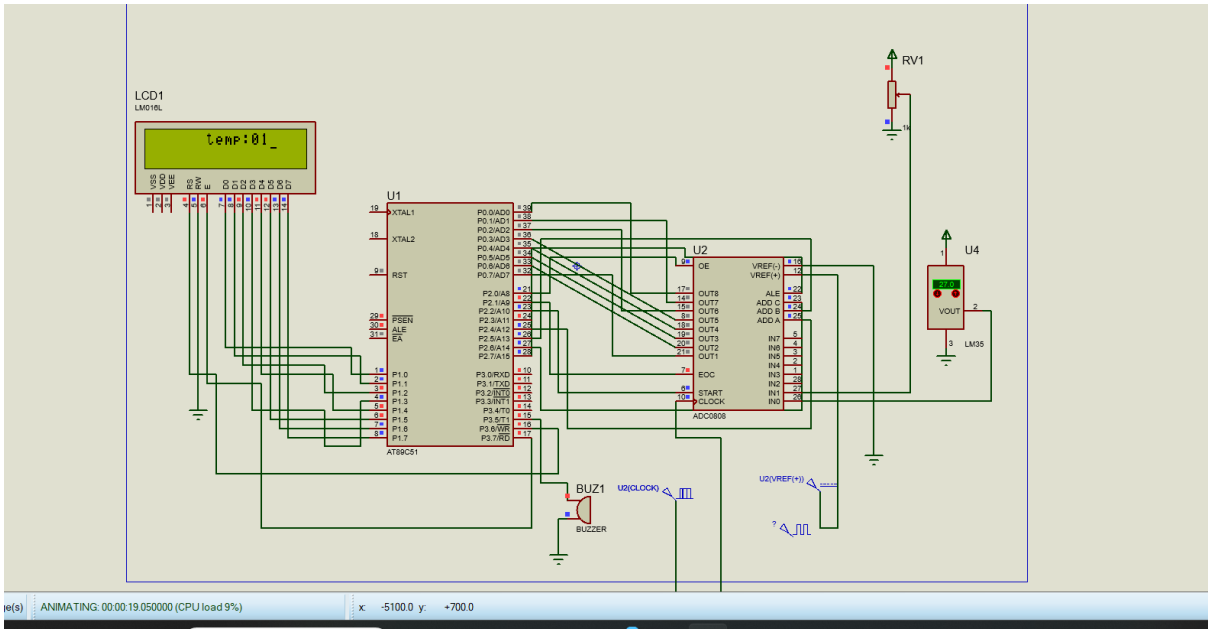
```
#include <reg51.h>
#include "lcd1.h"
#define adc_data P0
sbit OE    =P2^0;
sbit EOC   =P2^1;
sbit START  =P2^2;
sbit AA    =P2^4;
sbit BB    =P2^5;
sbit CC    =P2^6;
sbit ALE   =P2^7;
sbit buz   =P3^5;
unsigned char temp;
unsigned char moist;
unsigned char adc_val;
unsigned char adc(unsigned int ch);
void ch_sel(unsigned int sel);
void  main()
{

  lcd_init();
  stringdata("TEMP: ");
```

```
while(1) {
                          lcd_init();
   cmd(0x86);

                         stringdata("temp:");

   temp=adc(0);            //Reading Value from ADC Channel 0
   dat((temp/100)%10 +0x30);
   dat((temp/10)%10 +0x30);
   dat(temp%10 +0x30);
                         delaylcd(1);
                         lcd_init();
                         cmd(0xc0);
                           cmd(0x86);
                         stringdata("Moist:");

   moist=adc(1);           //Reading Value from ADC Channel 1
   dat((moist/100)%10 +0x30);
   dat((moist/10)%10 +0x30);
   dat(moist%10 +0x30);
                          delaylcd(1);
                         if(temp>0x36)
                         {
                                 lcd_init();
                                 cmd(0x01);
                                 buz=1;
          cmd(0xc0);
                                 stringdata("danger");
}
       }
```

```
        }
unsigned char adc(unsigned int ch)
{
   adc_data=0xff;
   ALE=START=OE=AA=BB=CC=0;
   EOC=1;
   ch_sel(ch);
   ALE=1;
   START=1;
   ALE=0;
   START=0;
   while(EOC==1);
   while(EOC==0);
   OE=1;
   adc_val=adc_data;
   OE=0;
   return adc_val;
}
void ch_sel(unsigned int sel)
{
   switch(sel) {
      case 0: CC=0;BB=0;AA=0; break;      //000
      case 1: CC=0;BB=0;AA=1; break;      //001
      case 2: CC=0;BB=1;AA=0; break;      //010
      case 3: CC=0;BB=1;AA=1; break;      //011
      case 4: CC=1;BB=0;AA=0; break;      //100
      case 5: CC=1;BB=0;AA=1; break;      //101
      case 6: CC=1;BB=1;AA=0; break;      //110
      case 7: CC=1;BB=1;AA=1; break;      //111
   }
}
```

Output:


LCD1
LM016L
Moist:12_
RV1
U1
AT89C51
U2
ADC0808
U4
LM35
BUZ1
BUZZER
6 Message(s)
ANIMATING: 00:01:21.500000 (CPU load 9%)
x: -3000.0 y: +600.0

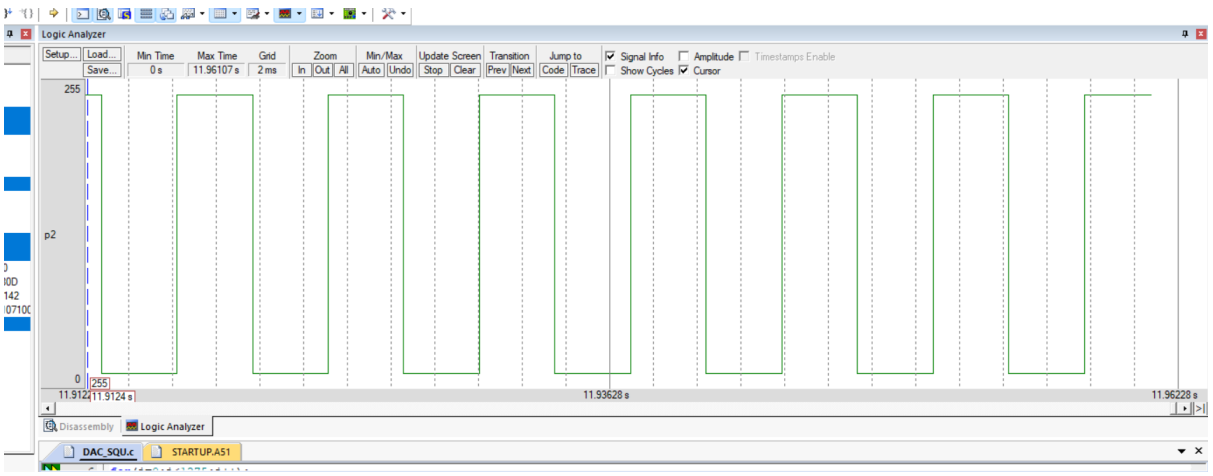

LCD1
LM016L
temp:01_
RV1
U1
AT89C51
U2
ADC0808
U4
LM35
BUZ1
BUZZER
ANIMATING: 00:00:19.050000 (CPU load 9%)
x: -5100.0 y: +700.0

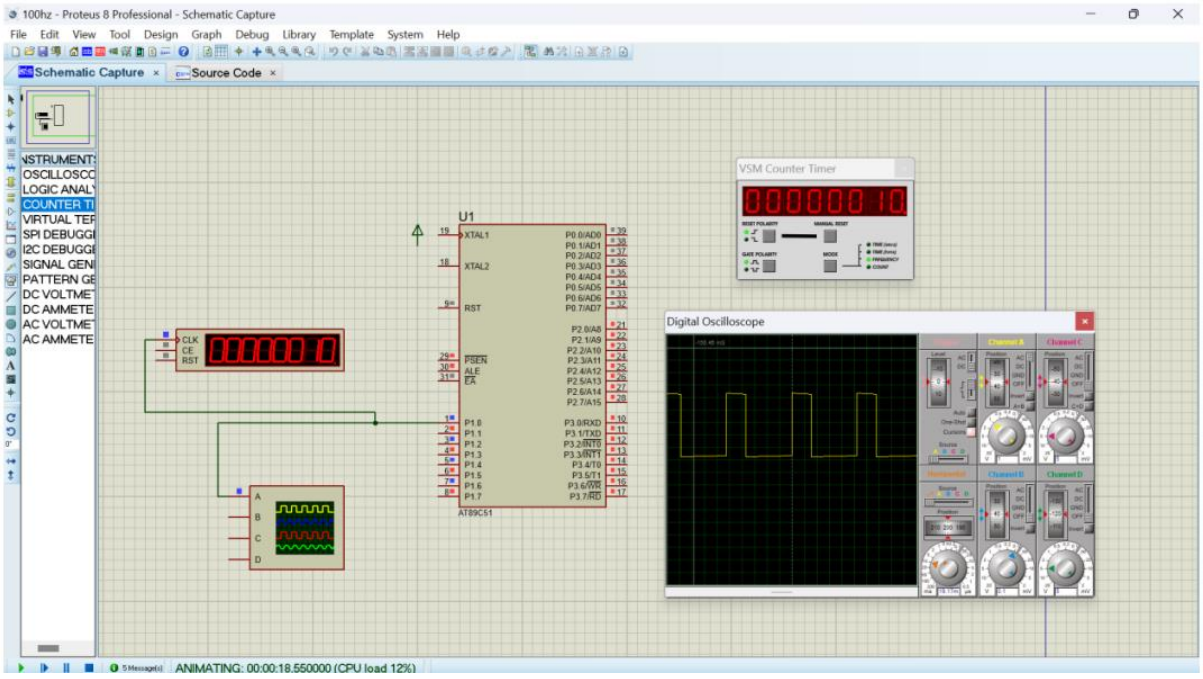
2) Write an alp to generate 100hz square wave with 50% duty cycle using timer 1 in mode-2 which is controlled by external hardware i.e hardware controlled. With all the necessary calculations.

Program:

```
#include <reg51.h>


void main(void)
{
  TMOD |= 0x20;  // Set Timer 1 to mode-2 (8-bit auto-reload)


  while (1)
  {
    TR1 = 1;  // Start Timer 1


    // Wait for external hardware control
    // You may need to implement a function or check a flag based on your hardware setup


    P1 = 0xFF;  // Set P1 to 0xFF for a high signal (50% duty cycle)


    // Wait for external hardware control
    // You may need to implement a function or check a flag based on your hardware setup


    TR1 = 0;  // Stop Timer 1
    P1 = 0x00; // Set P1 to 0x00 for a low signal (50% duty cycle)
  }
}
```

Output:


Logic Analyzer
Setup...
Load...
Save...
Min Time
0 s
Max Time
11.96107 s
Grid
2 ms
Zoom
In
Out
All
Min/Max
Auto
Undo
Update Screen
Stop
Clear
Transition
Prev
Next
Jump to
Code
Trace
Signal Info
Show Cycles
Amplitude
Timestamps Enable
Cursor
255
p2
0
11.9124
11.93628 s
11.96228 s
Disassembly
Logic Analyzer
DAC_SQU.c
STARTUP.A51


100hz - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
Source Code
COUNTER TI
U1
XTAL1
XTAL2
RST
PSEN
ALE
EA
AT89C51
CLK
CE
RST
VSM Counter Timer
Digital Oscilloscope
ANIMATING: 00:00:18.550000 (CPU load 12%)