CHAITANYA BHARATHI

INSTITUE OF TECHNOLOGY

TEMPERATURE CONTROLLED DC FAN

BY-

M.SAI VIKAS,160116735169

P.SRUJAN,160116735174

M.SUMANTH NAIDU,160116735175

INDEX

CONTENTS	PAGE NO.

ABSTRACT:

Now-a-day's technology is running with time, it completely occupied the life style of human beings. Even though there is such an importance for technology in our routine life there are even people whose life styles are very far to this well known term technology. So it is our responsibility to design few reliable systems which can be even efficiently used by them. Automatic Room Temperature Controlled Fan Speed Controller is one of them. The developed system provides an environment in which no user needed to control the fan speed. Automatically control the fan speed by sensing the room temperature. These fascinating efforts to create intelligent system are to provide human being a more convenient life.

COMPONENTS USED:

- AT89S52
- ADC0804
- LM35
- 16X2 LCD
- DC fan
- 12v relay
- 1N4007 diode
- BC547
- Resistors
- Capacitors

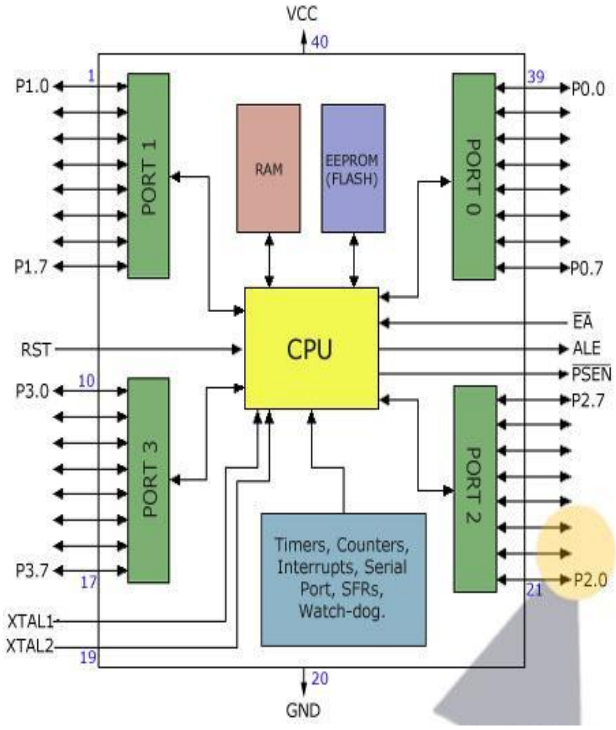
COMPONENTS DESCRIPTION:

AT89S52:

The 8051 Microcontroller was designed in 1980's by Intel. Its foundation was on Harvard Architecture and was developed principally for bringing into play in Embedded systems. At first it was created by means of NMOS technology but as NMOS technology needs more power to function therefore Intel re-intended Microcontroller 8051 employing CMOS technology and a new edition came into existence with a letter 'C' in the title name, for illustration: 80C51. These most modern Microcontrollers need fewer amount of power to function in comparison to their forerunners.

There are two buses in 8051 Microcontroller one for program and other for data. As a result, it has two storage rooms for both program and data of 64K by 8 size. The microcontroller comprise of 8 bit accumulator & 8 bit processing unit. It also consists of 8 bit B register as majorly functioning blocks and 8051 microcontroller programming is done with embedded C language using Keil software. It also has a number of other 8 bit and 16 bit registers.

The 89S52 has 4 different ports, each one having 8 Input/output lines providing a total of 32 I/O lines. Those ports can be used to output DATA and orders do other devices, or to read the state of a sensor, or a switch. Most of the ports of the 89S52 have 'dual function' meaning that they can be used for two different functions.



The first one is to perform input/output operations and the second one is used to implement special features of the microcontroller like counting external pulses, interrupting the execution of the program according to external events, performing serial data transfer or connecting the chip to a computer to update the software. Each port has 8 pins, and will be treated from the software point of view as an 8-bit variable called 'register', each bit being connected to a different Input/Output pin.

ADC 0804:

The ADC0804 is a commonly used ADC module, for projects were an external ADC is required. It is a 20-pin Single channel 8-bit ADC module. Meaning it can measure one ADC value from 0V to 5V and the precision when voltage reference (Vref –pin 9) is +5V is 19.53mV (Step size). That is for every increase of 19.53mV on input side there will be an increase of 1 bit at the output side.

This IC is very Ideal to use with Microprocessors like Raspberry Pi, Beagle bone etc.. Or even to use as a standalone ADC module. Every ADC module requires a clock to function; this IC comes with its own internal clock so you don't have to worry about it. Hence, if you are looking for a compact ADC module with a decent resolution of 8-bit then this IC is for you.

Pin Number	Pin Name	Description
1	Chip Select (CS)	Chip select is used if more than 1 ADC module is used. By default grounded
2	Read (RD)	Read pin must be grounded to read the Analog value
3	Write (WR)	Write pin should be pulsed high to start data conversion
4	CLK IN	External clock can be connected here, else RC can be used for accessing internal clock
5	Interrupt (INTR)	Goes high for interrupt request.
6	Vin (+)	Differential Analog input +. Connect to ADC input
7	Vin (-)	Differential Analog input -. Connect to ground
8	Ground	Analog Ground pin connected to ground of circuit
9	Vref/2	Reference voltage for ADC conversion.
10	Ground	Digital Ground pin connected to ground of circuit
11 to 18	Data bit 0 to bit 7	Seven output Data bit pins from which output is obtained
19	CLK R	RC timing resistor input pin for internal clock gen
20	Vcc	Powers the ADC module, use +5V

Features

- Easy to interface with all Microprocessors or works Stand alone.
- Single channel 8-bit ADC module
- On chip Clock available, no need of external Oscillator (Clock)
- Digital output various from 0 to 255
- When Vref = 5V, for every 19.53mV of analog value there will be rise of one bit on digital side (Step size)
- Available in 20-pin PDIP, SOIC packages

LM35:

The LM35 datasheet specifies that this ICs are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade)temperature.
The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centi-grade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄4˚C at room temperature and ±3⁄4˚C over a full −55 to +150˚C temperature range.

$$V_{OUT} = 10\ \text{mv/°C} \times T$$

where

- V_{OUT} is the LM35 output voltage
- T is the temperature in °C

Pin Number	Pin Name	Description
1	Vcc	Input voltage is +5V for typical applications
2	Analog Out	There will be increase in 10mV for raise of every 1°C. Can range from -1V(-55°C) to 6V(150°C)
3	Ground	Connected to ground of circuit

LM35 Features:

- Calibrated directly in ^{o} Celsius (Centigrade)
- Linear + 10.0 mV/^{o}C scale factor
- 0.5^{o}C accuracy guaranteeable (at +25^{o}C)
- Rated for full −55^{o} to +150^{o}C range

- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 µA current drain
- Low self-heating, 0.08ºC in still air
- Nonlinearity only ±1⁄4ºC typical
- Low impedance output, 0.1 Ω for 1 mA load

If the temperature is 0°C, then the output voltage will also be 0V. There will be rise of 0.01V (10mV) for every degree Celsius rise in temperature. The voltage can converted into temperature using the above formulae.

16X2 LCD DISPLAY:

LCD modules are vey commonly used in most embedded projects, the reason being its cheap price, availability and programmer friendly. Most of us would have come across these displays in our day to day life, either at PCO's or calculators. The appearance and the pinouts have already been visualized above now let us get a bit technical.

16×2 LCD is named so because; it has 16 Columns and 2 Rows. There are a lot of combinations available like, 8×1, 8×2, 10×2, 16×1, etc. but the most used one is the 16×2 LCD. So, it will have (16×2=32) 32 characters in total and each character will be made of 5×8 Pixel Dots.

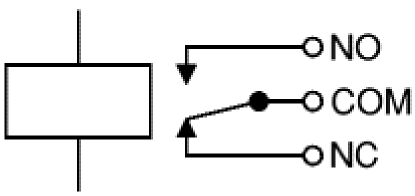
Features of 16×2 LCD module

- Operating Voltage is 4.7V to 5.3V
- Current consumption is 1mA without backlight
- Alphanumeric LCD display module, meaning can display alphabets and numbers
- Consists of two rows and each row can print 16 characters.
- Each character is build by a 5×8 pixel box
- Can work on both 8-bit and 4-bit mode
- It can also display any custom generated characters
- Available in Green and Blue Backlight

Pin No:	Pin Name:	Description
1	Vss (Ground)	Ground pin connected to system ground
2	Vdd (+5 Volt)	Powers the LCD with +5V (4.7V – 5.3V)
3	VE (Contrast V)	Decides the contrast level of display. Grounded to get maximum contrast.
4	Register Select	Connected to Microcontroller to shit between command/data register
5	Read/Write	Used to read or write data. Normally grounded to write data to LCD
6	Enable	Connected to Microcontroller Pin and toggled between 1 and 0 for data acknowledgement
7	Data Pin 0	Data pins 0 to 7 forms a 8-bit data line. They can be connected to Microcontroller to send 8-bit data. These LCD's can also operate on 4-bit mode in such case Data pin 4,5,6 and 7 will be left free.
8	Data Pin 1	
9	Data Pin 2	
10	Data Pin 3	
11	Data Pin 4	
12	Data Pin 5	
13	Data Pin 6	
14	Data Pin 7	
15	LED Positive	Backlight LED pin positive terminal
16	LED Negative	Backlight LED pin negative terminal

12V RELAY:

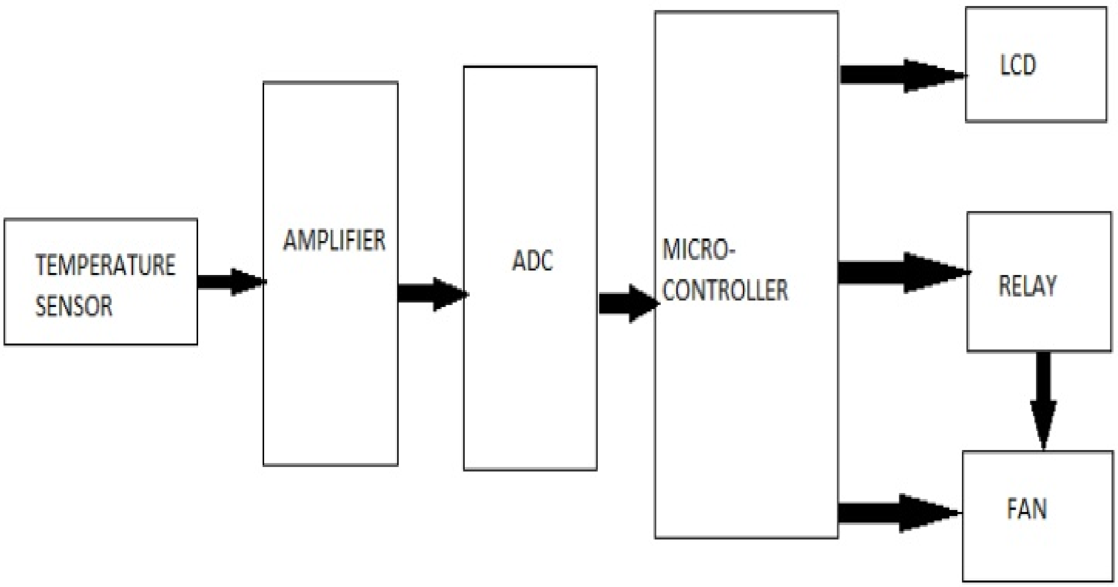
A relay is an electrically operated switch. Current flowing through the coil of the relay creates a magnetic field which attracts a lever and changes the switch contacts. The coil current can be on or off so relays have two switch positions and most have double throw(changeover) switch contacts as shown in the diagram.



Relays allow one circuit to switch a second circuit which can be completely separate from the first. For example a low voltage battery circuit can use a relay to switch a 230V AC mains circuit. There is no electrical connection inside the relay between the two circuits, the link is magnetic and mechanical.

The coil of a relay passes a relatively large current, typically 30mA for a 12V relay, but it can be as much as 100mA for relays designed to operate from lower voltages. Most ICs cannot provide this current and a transistor is usually used to amplify the small IC current to the larger value required for the relay coil. The maximum output current for the popular 555 timer IC is 200mA, enough to supply a relay coil directly.

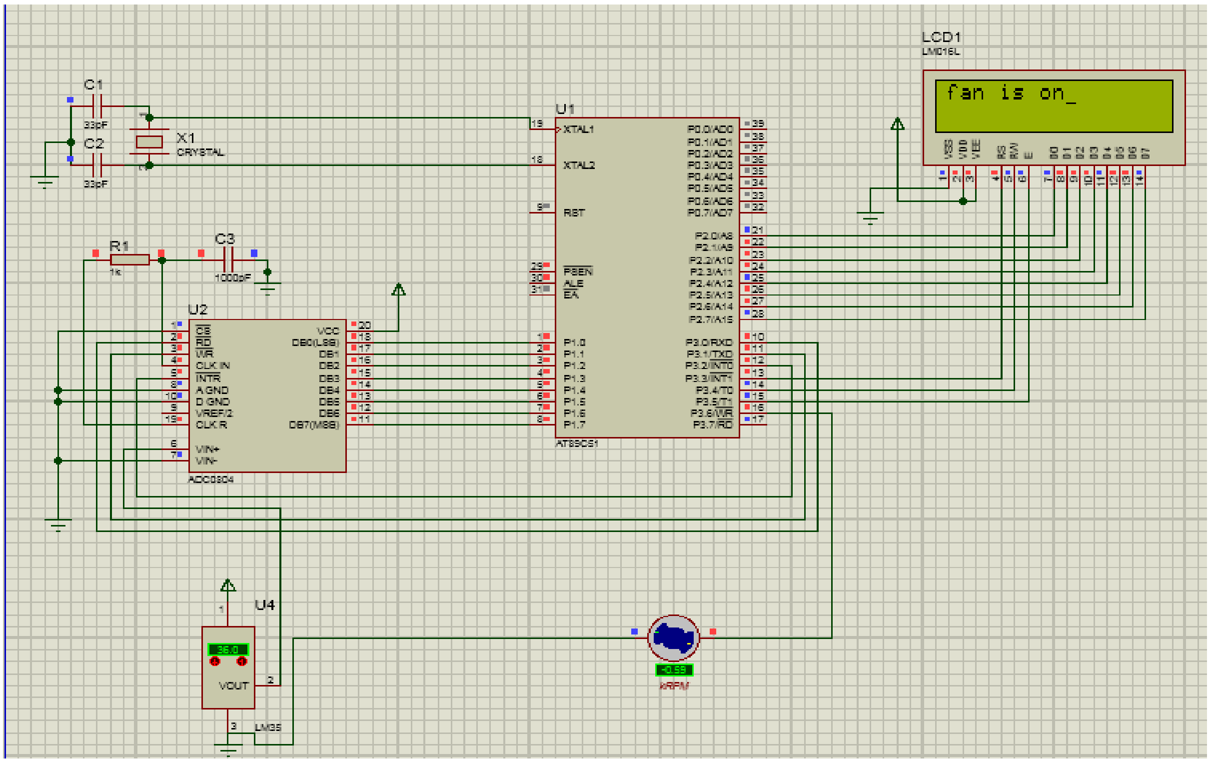
BLOCK DIAGRAM:



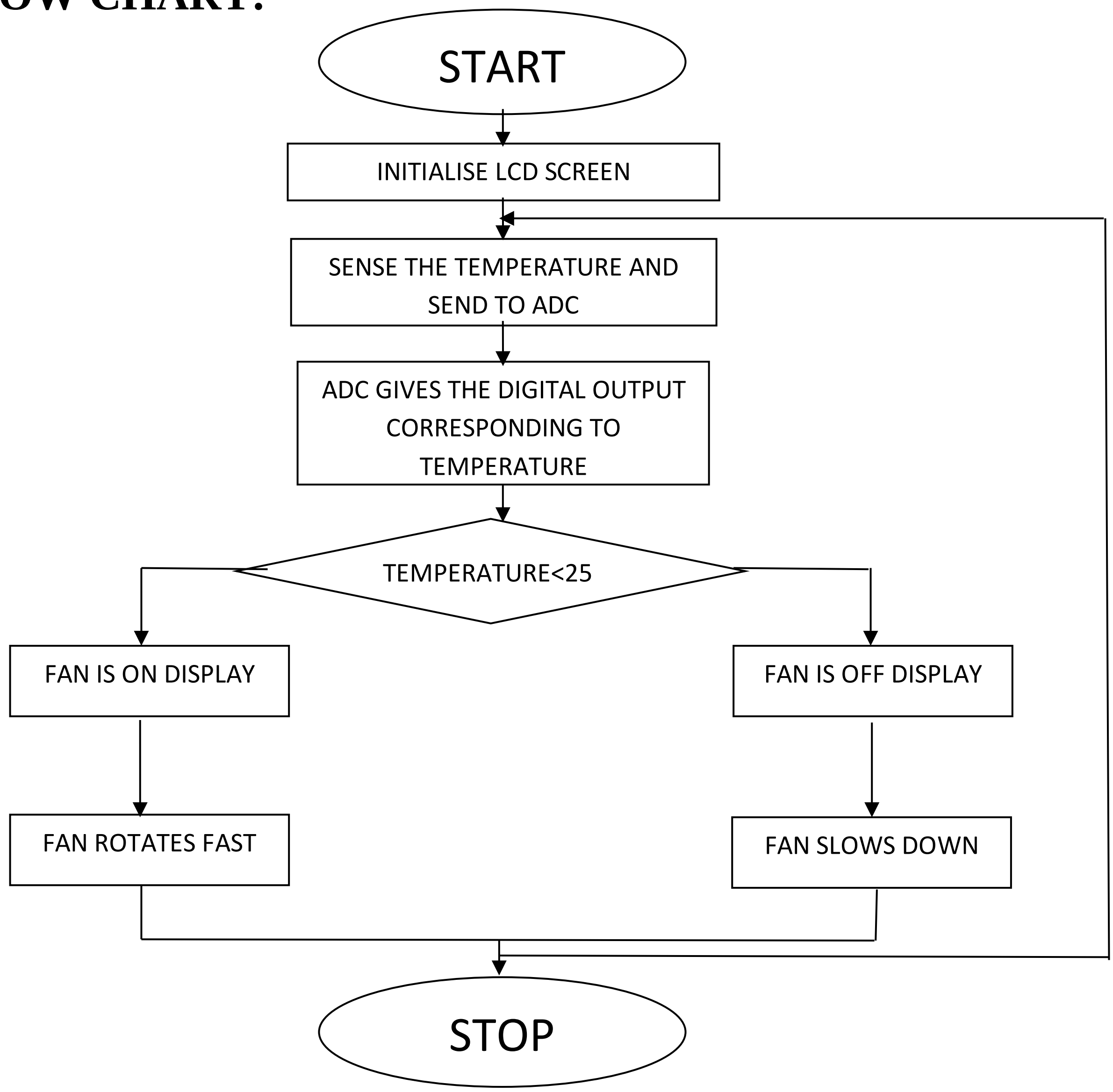
METHODOLOGY:

For the simulation of this project we used proteus software and connected the components as shown,

The temperature sensor lm35 detects the temperature and gives an analog output signal. This analog signal is given to the ADC 0804 for analog to digital conversion. this digital output is given to the micro controller and it turns on or off the fan using the relay.the fan status is sent to display on the lcd screen.


LCD1
LM016L
fan is on_
U1
XTAL1
XTAL2
RST
AT89C51
C1
33pF
C2
33pF
X1
CRYSTAL
R1
1k
C3
1000pF
U2
ADC0804
U4
VOUT
LM35

FLOW CHART:


START
INITIALISE LCD SCREEN
SENSE THE TEMPERATURE AND SEND TO ADC
ADC GIVES THE DIGITAL OUTPUT CORRESPONDING TO TEMPERATURE
TEMPERATURE<25
FAN IS ON DISPLAY
FAN IS OFF DISPLAY
FAN ROTATES FAST
FAN SLOWS DOWN
STOP

CODE IN C LANGUAGE:

```
#include<reg51.h>                          //include 8051 header files
#define dataport P2                        //lcd connected to port2
#define adc_input P1                       //adc connected to port1
sbit rd= P3^0;                             //read pin of adc connected to p3.0
sbit wr= P3^1;                             //write pin of adc connected to p3.1
sbit intr= P3^2;                           //intr pin of adc connected to p3.2
sbit rs = P3^3;                            //rs pin of lcd connected to p3.3
sbit rw = P3^4;                            //rw pin of lcd connected to p3.4
sbit e = P3^5;                             //enable pin of lcd connected to p3.5
sbit dc_fan=P3^6;                          //dc fan is connected to p3.6
void delay(unsigned int);                  //to generate delay
void lcd_cmd(unsigned char);               //lcd command write
void lcd_data(unsigned char);              //lcd data write
void lcd_data_string(unsigned char*);      //to display string on lcd
void fan(unsigned int);                    //fan control
void lcd_init();                           //lcd command initialisation
void adc_conv();                           //adc start conversion
void adc_read();                           //read adc output
unsigned char s1[]="fan is on";
unsigned char s2[]="fan is off";
void main()                                //main function starts
{
while(1)                                   //infinite loop
{
dataport=0x00;                             //lcd as output port
adc_input=0xff;                            //adc as input port
P3=0x00;                                   //port3 as output port
```

```
lcd_init();

adc_conv();

adc_read();

delay(50);

}                                   //infinte loop ends

}                                   //main ends


void lcd_init()

{

lcd_cmd(0x01);                      //clear lcd screen

delay(1);

lcd_cmd(0x06);                      //entry mode set

delay(1);

lcd_cmd(0x0e);                      //display-cursor on-off

delay(1);

lcd_cmd(0x38);                      //function set

delay(1);

lcd_cmd(0x80);                      //start display in first line

delay(1);

}


void delay(unsigned int sec )

{

int i ,j ;

for(i=0;i<sec;i++)

for(j=0; j<1275; j++);

}
```

```
void lcd_cmd(unsigned char item)
{
dataport=item;
rs= 0;                                  //select command register
rw=0;
e=1;                                    //enable high to low pulse
delay(1);
e=0;
}

void lcd_data(unsigned char item)
{
dataport = item;
rs= 1;                                  //select data register
rw=0;
e=1;                                    //enable high to low pulse
delay(1);
e=0;
}

void adc_conv()
{
wr = 0;                                 //write low to high pulse to start conversion
delay(2);
wr = 1;
while(intr);                            //wait for intr to become 0
delay(2);
intr=1;                                 //set intr for next cycle
```

```
}

void adc_read()
{
unsigned int value;
rd = 0;                         //read low to high to enable the outputs of adc
delay(2);
value=adc_input;                //copy adc_input port to value
delay(1);
rd=1;
fan(value);                     //call function to control fan
}

void fan(unsigned int i)
{
unsigned int p;
lcd_init();                     //lcd initialisation
if(i<13)                        //temperature is less than 25
{
dc_fan=0;                       //fan is off
delay(1000);
lcd_data_string("fan is off");
delay(1000);
}
else                            //temperature is more than 25
{
dc_fan=1;                       //fan is on
delay(1000);
```

```
lcd_data_string("fan is on");

delay(1000);

}

}


void lcd_data_string(char *x)

{

while(*x!='\0')

{

lcd_data(*x);

delay(2);

x++;

}

}
```

CONCLUSION:

We thus controlled a dc fan according to the temperature and obtained the outputs by simulating the above circuit and code in proteus design suite.

REFRENCES:

1. AT89S52 data sheet http://ww1.microchip.com/downloads/en/DeviceDoc/doc1919.pdf
2. ADC 0804 data sheet http://www.ti.com/lit/ds/symlink/adc0804-n.pdf
3. 16X2 LCD data sheet

https://components101.com/16x2-lcd-pinout-datasheet

4. LM35 data sheet http://www.ti.com/product/LM35

5.Proteus design suite simulation software

https://proteus.soft112.com/