**Microcontroller Experiment**

Independently Designed Experiment - Simulating Intelligent Street Lights

Name: Han Yichen 韩一尘

Student ID: 22722051

Date: 2023/11/1

1. **Experimental Principles**

The aim of this experiment is to design a program that simulates intelligent street lights using a microcontroller STM32. This program requires the operation mode to be divided into automatic sensing mode and manual mode, using LED lights to simulate street lights and indicator lights, and receiving and sensing external light intensity and temperature through built-in photoresistors and thermistors. In automatic sensing mode, the mode indicator light comes on, and when the light intensity sensed by the photosensitive resistor is below the threshold, the light comes on. On the contrary, it goes out. In manual mode, the indicator light turns off. The manual mode is divided into two parts: one is to use buttons to operate the lights, and the other is to remotely control the lights through the remote control. The operator can control the light on/off and switch between automatic/manual modes by using buttons or remote controls. When the temperature sensing resistor receives a temperature higher than the set threshold, the system will automatically cut off all functions, extinguish all lights, and sound an alarm with a buzzer to achieve the purpose of protecting the device. In addition, we have designed an LCD display function that can display the street light mode, external light intensity, external temperature, and alarm information on it to remind the operator of the operation information of the system.

Therefore, in addition to the main STm32 driver CPU library functions (stm32f10x\_it. c, systen\_stm32f10x. c), it is also necessary to call the header files of the control LED (led. h), button (key. h), LCD display screen (LCD. h), temperature sensor (adc. h), lighting sensor (lsens. h), buzzer (beep. h), remote control (remote. h), etc. to drive the normal operation of the system.

1. **Main program code**

The main program code is shown as following:

#include "led.h"

#include "delay.h"

#include "key.h"

#include "sys.h"

#include "lcd.h"

#include "usart.h"

#include "adc.h"

#include "lsens.h"

#include "beep.h"

#include "remote.h"

#include "timer.h"

#define THRESHOLD 50 //光照阈值

#define WARN\_TEMP 5000 //温度阈值

int main(void)

{

vu8 key=0; //接受按键

u8 reKey; //接受遥控器

int autoFlag; //自动手动模式标志位

int inFlag; //遥控按键模式标志位

u8 light; //光照强度

short temp; //温度

u16 brightness;

uart\_init(115200); //初始化接口

NVIC\_PriorityGroupConfig(NVIC\_PriorityGroup\_2);

POINT\_COLOR=RED; //设置LCD显示红色

delay\_init(); //延迟函数初始化

LED\_Init(); //LED初始化

KEY\_Init(); //按键初始化

BEEP\_Init(); //蜂鸣器初始化

LCD\_Init(); //LCD初始化

Lsens\_Init(); //光敏电阻初始化

T\_Adc\_Init(); //温度传感器初始化

Remote\_Init(); //遥控器初始化

autoFlag=1; //初始模式为自动模式

inFlag=1; //初始为按键接受

LED0=0; //LED0为模式指示灯

LED1=1; //LED1为照明灯

BEEP=0; //蜂鸣器初始不响

brightness=0;

//LCD显示

LCD\_ShowString(30,30,200,16,16,"Automatic Streetlight");

LCD\_ShowString(30,50,200,16,16,"KEY0:automatic/manual mode");

LCD\_ShowString(30,70,200,16,16,"KEY1:adjust manually");

LCD\_ShowString(30,90,200,16,16,"2023/11/13 Han Yichen");

POINT\_COLOR=BLUE;

LCD\_ShowString(30,130,200,16,16,"LSENS\_VAL:");

LCD\_ShowString(30,150,200,16,16,"TEMPERATE: 00.00C");

LCD\_ShowString(30,210,200,16,16,"KEY INPUT ");

while(1)

{

//接收温度与光照

light=Lsens\_Get\_Val();

temp=Get\_Temprate();

//显示温度

if(temp<0)

{

temp=-temp;

LCD\_ShowString(30+10\*8,150,16,16,16,"-")；

}else LCD\_ShowString(30+10\*8,150,16,16,16," ")；

LCD\_ShowxNum(30+11\*8,150,temp/100,2,16,0)；

LCD\_ShowxNum(30+14\*8,150,temp%100,2,16, 0X80);

LCD\_ShowxNum(30+10\*8,130,light,3,16,0);

//模式识别

if(autoFlag==1){

LCD\_ShowString(30,190,280,16,16,"Mode: Automatic");

LED0=0;

if(light<=THRESHOLD){

LED1=0;

}

else{

LED1=1;

}

}

else if(autoFlag==0){

LCD\_ShowString(30,190,280,16,16,"Mode: Manual ");

LED0=1;

}

//温度报警

if (temp > WARN\_TEMP) {

LED1 = 0;

autoFlag = 0;

LCD\_ShowString(30, 190, 280, 16, 16, "Temperature: WARNING");

BEEP = 1;

delay\_ms(300);

BEEP = 0;

}

else{

BEEP=0;

}

if(inFlag==1){

key=KEY\_Scan(0); //扫描按键输入

if(key)

{

switch(key)

{

case KEY2\_PRES:

inFlag=0;

LCD\_ShowString(30, 210, 300, 16, 16, "REMOTE INPUT");

delay\_ms(20);

break;

case KEY1\_PRES:

if(autoFlag==0){

LED1 = !LED1;

delay\_ms(20);

break;

}

else if(autoFlag==1)

delay\_ms(20);

break;

case KEY0\_PRES:

autoFlag = !autoFlag;

LED0 = !LED0;

delay\_ms(20);

break;

}

}else delay\_ms(10);

}

else{

reKey=Remote\_Scan(); //扫描遥控器输入

if(reKey)

{

switch(reKey)

{

case 152:

if(autoFlag==0){

LED1 = !LED1;

delay\_ms(200);

break;

}

else if(autoFlag==1)

delay\_ms(200);

break;

case 104:

autoFlag = !autoFlag;

LED0 = !LED0;

delay\_ms(200);

break;

case 176:

inFlag=1;

LCD\_ShowString(30, 210, 300, 16, 16, "KEY INPUT ");

delay\_ms(200);

break;

}

}

}

}

}

As can be seen, the code is mainly divided into several parts: first, the header files corresponding to the hardware required for this experiment were imported. Secondly, at the beginning of the main function, initialization operations were performed on the hardware to be called, and some necessary parameters were declared (such as indicators for control modes such as ‘autoFlag’ and indicators for receiving sensor information such as ‘light’). Subsequently, a while (1) loop was used to continuously scan and update the hardware status using if else statements, switch statements, and other statements, along with logical statements. Finally, the predetermined functionality was achieved

1. **Experimental result**

After compiling and importing the program into the microcontroller, it was found that the microcontroller can operate as we envisioned: the system can be used normally in both automatic and manual modes, controlled by buttons or remote controls, and displayed relevant information on the LCD screen. It also responds to changes in the external environment or displays warning messages. The only regret is that perhaps due to the low battery of the laboratory remote control for many years, the operation of the remote control is not particularly sensitive, but it can basically operate normally.

1. **Improvement and perfection**

This experiment is an intelligent lighting simulation system, so the following aspects can be improved to make it more practical or intelligent: 1. You can refer to the practice of adjusting the frequency to adjust the brightness of the light through the function in timer. c in PWM experiments, allowing the lighting lamp to adjust its brightness under different lighting intensities to achieve the idea of saving electricity in daily life; 2. You can modify the output port of the LED to a different port and connect the port to a real incandescent light through a DuPont cable instead of a red and green LED light, this can make the experimental effect more realistic.