实验三 IIC总线EEPROM读写

实验目的：

1. 掌握IIC总线的使用方式
2. 掌握EEPROM芯片的工作原理
3. 掌握IIC总线通信方式实现MCU与24C02的数据读写

实验内容：

学习IIC总线传输机制，通过单片机MCU的I/O实现IIC总线在EEPROM 24C02上读写数据并显示显示在数码管上。

参考资料：芯片手册文档，

实验步骤：

1. 编写EEPROM写入程序，将数据（学号后面8位）保存在芯片，下载到开发板进行调试，实现显示写入完毕功能。
2. 编写EEPROM读出程序，将数据（学号后面8位）显示在数码管。

实验要求：

编写实验报告，主要包括关键步骤的实现和效果截屏，并分析实验过程中出现的问题和分析解决方法。

1. 实现代码

#include "reg51.h"

#include "intrins.h"

#define uchar unsigned char

#define uint unsigned int

#define out P2

#define SMG\_PORT P0

sbit scl = P2^1;

sbit sda = P2^0;

sbit key1 = P3^2;

sbit key2 = P3^3;

sbit LSA = P2^2;

sbit LSB = P2^3;

sbit LSC = P2^4;

uchar data mem[4]\_at\_ 0x55;

uchar mem[4] = {21, 12, 10, 90};

uchar data rec\_mem[4] \_at\_ 0x60;

uchar rec\_mem[4] = {0x00, 0x00, 0x00, 0x00};

uchar gsmg\_code[17]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,

0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71};

void start(void);

void stop(void);

void sack(void);

bit rack(void);

void ackn(void);

void send\_byte(uchar);

uchar rec\_byte();

void write(void);

void read(void);

void delay4us(void);

void show(void);

void smg\_display(uchar dat[], uchar pos);

void main(void)

{

EA=1;EX0=1;EX1=1;

**while**(1)

{

show();

}

}

void ext0()interrupt 0

{

write();

}

void ext1()interrupt 2

{

read();

show();

}

void read(void)

{

uchar i;

bit f;

start();

send\_byte(0xa0);

f = rack();

**if**(!f)

{

start();

send\_byte(0xa0);

f = rack();

send\_byte(0x00);

f = rack();

}

**if**(!f)

{

start();

send\_byte(0xa1);

f = rack();

}

**if**(!f)

{

**for**(i=0; i<3; i++)

{

rec\_mem[i] = rec\_byte();

sack();

}

rec\_mem[3] = rec\_byte();

ackn();

}

stop();

out = rec\_mem[3];

**while**(!key2);

}

void write(void)

{

uchar i;

bit f;

start();

send\_byte(0xa0);

f = rack();

**if**(!f)

{

send\_byte(0x00);

f = rack();

**if**(!f)

{

**for**(i=0; i<4; i++)

{

send\_byte(mem[i]);

f = rack();

**if**(f) **break**;

}

}

}

stop();

out = 0xc3;

**while**(!key1);

}

void start(void)

{

scl = 1;

sda = 1;

delay4us();

sda = 0;

delay4us();

scl = 0;

}

void stop(void)

{

scl = 0;

sda = 0;

delay4us();

scl = 1;

delay4us();

sda = 1;

delay4us();

sda = 0;

}

bit rack(void)

{

bit flag;

scl = 1;

delay4us();

flag = sda;

scl = 0;

**return**(flag);

}

void sack(void)

{

sda = 0;

delay4us();

scl = 1;

delay4us();

scl = 0;

delay4us();

sda = 1;

delay4us();

}

void ackn(void)

{

sda = 1;

delay4us();

scl = 1;

delay4us();

scl = 0;

delay4us();

sda = 0;

}

uchar rec\_byte(void)

{

uchar i, temp;

**for**(i=0; i<8; i++)

{

temp <<= 1;

scl = 1;

delay4us();

temp |= sda;

scl = 0;

delay4us();

}

**return**(temp);

}

void send\_byte(uchar temp)

{

uchar i;

scl = 0;

**for**(i=0; i<8; i++)

{

sda = (bit)(temp&0x80);

scl = 1;

delay4us();

scl = 0;

temp <<= 1;

}

sda = 1;

}

void show(void)

{

uchar i=0;

uchar dat[8];*//= {2,1, 1,2, 1,0, 9,0};*

**for**(i=0; i<4; i++)

{

dat[i\*2+1] = rec\_mem[i]%10;

dat[i\*2] = rec\_mem[i]/10;

}

smg\_display(dat, 1);

}

void smg\_display(uchar dat[], uchar pos)

{

uchar i=0;

uchar pos\_temp=pos-1;

uint k=0;

*//out = 0xc3;*

**for**(i=pos\_temp;i<8;i++)

{

**switch**(i)*//浣?閫?*

{

**case** 0: LSC=1;LSB=1;LSA=1;**break**;

**case** 1: LSC=1;LSB=1;LSA=0;**break**;

**case** 2: LSC=1;LSB=0;LSA=1;**break**;

**case** 3: LSC=1;LSB=0;LSA=0;**break**;

**case** 4: LSC=0;LSB=1;LSA=1;**break**;

**case** 5: LSC=0;LSB=1;LSA=0;**break**;

**case** 6: LSC=0;LSB=0;LSA=1;**break**;

**case** 7: LSC=0;LSB=0;LSA=0;**break**;

}

SMG\_PORT=gsmg\_code[dat[i-pos\_temp]];*//浼?閫?娈?閫?鏁?鎹?*

**for**(k=0; k<250; k++)

{

*//delsy4us();*

}

SMG\_PORT=0x00;*//娑?闊?*

}

}

void delay4us(void)

{

\_nop\_();\_nop\_();\_nop\_();\_nop\_();

}

1. 演示效果

图片包含 桌子, 游戏机, 电路, 电脑

描述已自动生成

桌子上放了游戏机

低可信度描述已自动生成