

به نام خدا

گزارش کار آزمایش هشتم ریزپردازنده

گروه 7

آریان بوکانی 9731012

وسایل استفاده شده در این آزمایش:

- Arduino Mega 2560
- Green LED
- Keypad
- EEPROM
- LCD 16*2

توضیحاتی در مورد قطعه‌ی جدید استفاده شده (EEPROM):

EEPROM مخفف عبارت Electrically Erasable Programmable Read-Only Memory یا حافظه‌ی فقط خواندنی قابل برنامه‌ریزی و پاکسازی به صورت الکتریکی است. در این آزمایش از AT24C512 استفاده شده است که دارای دو پایه برای مشخص کردن آدرس، دو پایه‌ی SCL و SDA برای ارتباط و یک پایه بنام WP است که به معنای Write Protect است. این پایه High Active است و اگر به زمین وصل شود، دستگاه EEPROM مربوطه قابلیت Read و Write خواهد داشت.

توضیحاتی در مورد کتابخانه‌ی Wire:

از این کتابخانه برای ارتباط میکروکنترلر با دستگاه‌های مختلف بر اساس پروتوکل I2C استفاده می‌شود.

برای ایجاد ارتباط باید پایه‌های SCL و SDA دستگاه بیرونی را به ترتیب به پایه‌های 21 و 20 میکروکنترلر متصل کرد. به این منظور باید تابع Wire.begin() را در قسمت setup صدا زد تا ایجاد ارتباط با این پروتوکل میسر شود.

: Wire.beginTransmission(byte address)

برای اتصال با گیرنده و شروع ارتباط با دستگاه متناظر با address استفاده می‌شود.

: Wire.endTransmission()

برای اتمام ارسال داده استفاده می‌شود.

: Wire.write(byte data)

برای نوشتن روی سیگنال باس I2C استفاده می‌شود.

: Wire.available()

تعداد بیت‌های در دسترس برای خواندن توسط تابع read را برمی‌گرداند.

: Wire.read()

برای خواندن از روی سیگنال باس I2C استفاده می‌شود.

: Wire.requestFrom(byte address, int quantity)

این تابع توسط master برای درخواست داده از slave است. با فراخوانی این تابع master از slave درخواست چند بایت دیتا به اندازه‌ی quantity را انجام می‌دهد. خروجی آن تعداد بایت‌هایی است که از دستگاه slave خوانده می‌شود.

پی‌نوشت:

در این آزمایش برای راحت‌تر شدن کار کل عملیات مربوط به read و write در دو تابع readI2C و writeI2C نوشته شده‌اند.

پاسخ به پرسش‌های دستورکار:

1. در چه کاربرد‌هایی EEPROM به کار برده می‌شود؟ چرا در اینجا حافظه Flash یا RAM را به کار نمی‌بریم؟

تفاوت حافظه RAM با EEPROM در چی می‌باشد؟

از این حافظه برای نگهداری و خواندن داده‌ها با حجم مناسب استفاده می‌شود و می‌توان بارهای بار اطلاعات موجود در آن را با استفاده از ولتاژ تغییر داد. هنگام استفاده از Flash برای نوشتن دوباره باید تعداد زیادی از داده‌ها را نیز پاک کرد. برای مثال اگر بخواهیم یک بایت را بازنویسی کنیم باید حدود 4096 بایت را پاک کرد. همچنین حافظه‌ی Flash حدوداً تا ده هزار بار توانایی پاک شدن و نوشته شدن را دارد درحالی که برای EEPROM این مقدار حدود صد هزار بار است. همچنین نمی‌توان از حافظه‌ی RAM استفاده کرد چون این حافظه موقتی است. حافظه‌ی RAM هم برای خواندن است و هم برای نوشتن. درحالی که حافظه‌ی EEPROM فقط خواندنی است.

2. اگر بخواهیم برای نگهداری مدهای کاری حافظه Flash را به کار ببریم، فرآیند نوشتن باید چگونه انجام شود

که داده‌های دیگری که بر روی همان بلاک می‌باشند از دست نروند؟

برای حفاظت از داده‌های هر Page که معمولاً 4KB هستند، باید همه‌ی این داده‌ها را در جای دیگری ذخیره کرد تا داده‌های مهم از بین نروند و عملکرد سیستم مختل نشود.

3. اگر یک حافظه EEPROM بیرونی دارای 4 KB حافظه و 2 پایه آدرس باشد در این صورت حداکثر چند KB حافظه EEPROM بیرونی می توان بر روی یک باس مشترک داشت؟

بدلیل وجود دو پایه برای آدرس حداکثر چهار حافظه می توان داشت که در نهایت می توان 16 KB داده ذخیره کرد.

4. فرکانس کلاک در کدام دستگاه پیکربندی می شود؟ کلاک را کدام دستگاه فراهم میکند؟ با توجه به زمان مورد نیاز برای انجام عملیات نوشتن، با فرض اینکه کلاک را 10 KHz تنظیم کرده باشیم در این صورت حداکثر با چه نرخ می تواند عملیات نوشتن را انجام داد؟

فرکانس کلاک و کلاک در قسمت PMC یا Power Management Controller تامین می شود. برای خواندن یا نوشتن هر بایت در این پروتوکل به هفت بیت برای آدرس دهی، یک بیت ACK، یک بیت مشخص کردن read یا write و هشت بیت دیتا و دو بیت شروع و پایان نیاز است. بنابراین با فرض کلاک 10 KHz دارای نرخ تبادل 526 بایت بر ثانیه خواهیم بود.

کد آزمایش :

```
#include <LiquidCrystal.h>
```

```
#include <Keypad.h>
```

```
#include <Wire.h>
```

```
# define ADDRESS 0b1010000
```

```
# define RS_PIN 13
```

```
# define EN_PIN 12
```

```
# define D4_PIN 11
```

```
# define D5_PIN 10
```

```
# define D6_PIN 9
```

```
# define D7_PIN 8
```

```
# define led1 7
```

```
# define led2 6
```

```
# define led3 5
```

```
# define led4 4
```

```
int mode = 1;
```

```
int mode_time1 = 2;
```

```
int mode_time2 = 2;
```

```
int mode_time3 = 2;
```

```
int mode_time4 = 2;
```

```
int startingTime;
```

```
// 1 means pre washing
```

```
// 2 means washing
```

```
// 3 means post washing
```

```
// 4 means drying
```

```
char keys[4][3] = {
```

```
    {'1', '2', '3'},
```

```
    {'4', '5', '6'},
```

```
    {'7', '8', '9'},
```

```
    {'*', '0', '#'}  
};
```

```
byte pin_rows[4] = {25, 26, 27, 28}; //connect to the row pinouts of the keypad
```

```
byte pin_columns[3] = {24, 23, 22}; //connect to the column pinouts of the keypad
```

```
Keypad keypad = Keypad(makeKeymap(keys), pin_rows, pin_columns, 4, 3);  
LiquidCrystal lcd(RS_PIN, EN_PIN, D4_PIN, D5_PIN, D6_PIN, D7_PIN);
```

```
byte readI2C(byte deviceAddress, byte dataAddress){  
    Wire.beginTransmission(ADDRESS);  
    Wire.write(deviceAddress);  
    Wire.write(dataAddress);  
    Wire.endTransmission();  
    Wire.requestFrom(ADDRESS, 1);  
    if(Wire.available())  
        return Wire.read();  
    else  
        return 0xFF;  
}
```

```
void writeI2C(byte deviceAddress, byte dataAddress, byte data){  
    Wire.beginTransmission(ADDRESS);  
    Wire.write(deviceAddress);  
    Wire.write(dataAddress);  
    Wire.write(data);  
    Wire.endTransmission();  
    delay(10);  
}
```

```
void setup() {
```

```
pinMode(led1, OUTPUT);  
pinMode(led2, OUTPUT);  
pinMode(led3, OUTPUT);  
pinMode(led4, OUTPUT);  
  
lcd.begin(16, 2);  
Wire.begin();  
writeI2C(0x00, 0x00, mode);  
}
```

```
void loop(){  
    mode = readI2C(0x00, 0x00);  
    lcd.clear();  
    lcd.home();  
    lcd.print("Default mode:");  
    lcd.print(mode);  
    delay(500);  
    lcd.clear();  
    lcd.print("Press * to start");  
    lcd.setCursor(0, 1);  
    lcd.print("Press # for time");  
    char inputKey = keypad.waitForKey();  
    if (inputKey == '#'){  
        // user wants to change the timer  
        lcd.clear();
```

```

lcd.home();

lcd.print("enter new timer:");

lcd.setCursor(0, 1);

String newTime = "";

char input = keypad.waitForKey();

while(input != '*' && input != '#'){

    newTime += input;

    lcd.clear();

    lcd.home();

    lcd.print("enter new timer:");

    lcd.setCursor(0, 1);

    lcd.print(newTime);

    input = keypad.waitForKey();

}

int mode_time = newTime.toInt();

delay(500);

lcd.clear();

lcd.home();

lcd.print("which mode?");

input = keypad.waitForKey();

if (input == '1')

    mode_time1 = mode_time;

else if (input == '2')

    mode_time2 = mode_time;

else if (input == '3')

```



```

    mode_time3 = mode_time;
else if (input == '4')
    mode_time4 = mode_time;
lcd.clear();
lcd.print("Mode " + String(input) + " time");
lcd.setCursor(0, 1);
lcd.print("changed.");
delay(1000);
}
else if(inputKey == '*'){
    //user wants to run the washing machine
    bool exitFlag = false;
    lcd.clear();
    lcd.home();
    lcd.print("* : continue");
    lcd.setCursor(0, 1);
    lcd.print("# : change def");
    char nextKey = keypad.waitForKey();
    if (nextKey == '*'){
        lcd.clear();
        lcd.home();
        lcd.print("Current mode:" + String(mode));
        delay(200);
        while(true){
            if (mode == 1){

```

```

digitalWrite(led1, HIGH);
startingTime = millis() / 1000;
while(true){
    lcd.clear();
    lcd.home();
    lcd.print("mode 1 running");
    char exitKey = keypad.getKey();
    if (exitKey == '*'){
        lcd.clear();
        lcd.home();
        lcd.print("Exiting");
        exitFlag = true;
        delay(200);
        digitalWrite(led1, LOW);
        break;
    }
    lcd.setCursor(0, 1);
    int toPrint = mode_time1 - (millis()/1000 - startingTime);
    if (toPrint >= 0)
        lcd.print("Remaining:" + String(toPrint));
    else{
        mode++;
        digitalWrite(led1, LOW);
        break;
    }
}

```

```

        delay(200);
    }
}

else if (mode == 2){
    digitalWrite(led2, HIGH);
    startingTime = millis() / 1000;
    while(true){
        lcd.clear();
        lcd.home();
        lcd.print("mode 2 running");
        char exitKey = keypad.getKey();
        if (exitKey == '*'){
            lcd.clear();
            lcd.home();
            lcd.print("Exiting");
            exitFlag = true;
            delay(200);
            digitalWrite(led2, LOW);
            break;
        }
        lcd.setCursor(0, 1);
        int toPrint = mode_time2 - (millis()/1000 - startingTime);
        if (toPrint >= 0)
            lcd.print("Remaining:" + String(toPrint));
        else{

```

```

        mode++;
        digitalWrite(led2, LOW);
        break;
    }
    delay(200);
}
}
else if (mode == 3){
    digitalWrite(led3, HIGH);
    startingTime = millis() / 1000;
    while(true){
        lcd.clear();
        lcd.home();
        lcd.print("mode 3 running");
        char exitKey = keypad.getKey();
        if (exitKey == '*'){
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("Exiting");
            exitFlag = true;
            delay(200);
            digitalWrite(led3, LOW);
            break;
        }
        lcd.setCursor(0, 1);

```

```

int toPrint = mode_time3 - (millis()/1000 - startingTime);
if (toPrint >= 0)
    lcd.print("Remaining:" + String(toPrint));
else{
    mode++;
    digitalWrite(led3, LOW);
    break;
}
delay(200);
}
}

else if (mode == 4){
    digitalWrite(led4, HIGH);
    startingTime = millis() / 1000;
    while(true){
        lcd.clear();
        lcd.home();
        lcd.print("mode 4 running");
        char exitKey = keypad.getKey();
        if (exitKey == '*'){
            lcd.clear();
            lcd.home();
            lcd.print("Exiting");
            delay(200);
            exitFlag = true;

```

```

    digitalWrite(led4, LOW);
    break;
}
lcd.setCursor(0, 1);
int toPrint = mode_time4 - (millis()/1000 - startingTime);
if (toPrint >= 0)
    lcd.print("Remaining:" + String(toPrint));
else{
    mode++;
    digitalWrite(led1, HIGH);
    digitalWrite(led2, HIGH);
    digitalWrite(led3, HIGH);
    digitalWrite(led4, HIGH);
    lcd.clear();
    lcd.home();
    lcd.print("Finished washing");
    delay(500);
    digitalWrite(led1, LOW);
    digitalWrite(led2, LOW);
    digitalWrite(led3, LOW);
    digitalWrite(led4, LOW);
    break;
}
delay(200);
}

```

```

    }
    if (exitFlag || mode > 4)
        break;
    }
}

else if (nextKey == '#'){
    //user wants to change default mode
    lcd.clear();
    lcd.home();
    lcd.print("Enter def mode:");
    char newMode = keypad.waitForKey();
    lcd.setCursor(0, 1);
    lcd.print(newMode - '0');

    //writing
    writeI2C(0x00, 0x00, newMode - '0');
    lcd.clear();
    lcd.home();
    lcd.print("Savaing done!");
    delay(2000);
}

else{
    lcd.clear();
    lcd.home();
    lcd.print("Invalid input");
}

```

```

delay(2000);

}

}

else{

  lcd.clear();

  lcd.home();

  lcd.print("Invalid input");

  delay(2000);

}

}

```

چند نمونه از اجرای میکرو:



