بسمه تعالی

آزمایشگاه ریزپردازنده و زبان های اسمبلی

استاد مربوطه:

مهندس معصوم زاده

گزارش کار آزمایش چهارم

ازمایش کتابخوان

فروغ افخمی9831703

نیم سال دوم 1402-1401

کد نوشتن و خواندن روی eeprom:

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

#define ADDR_Ax 0b000 //A2, A1, A0
#define ADDR (0b1010 << 3) + ADDR_Ax

// each line is defined as 10 bytes
int curr_line = 0;
char buffer[17] = {0};
char text[] = "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim
veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse
cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non
proident, sunt in culpa qui officia deserunt mollit anim id est laborum.";

void eeprom_write(uint16_t memory_address, uint8_t* data, int _size);
void eeprom_read(uint16_t memory_address, uint8_t* data, int _size);

void readLineE2P()
{
  eeprom_read(curr_line * 16, buffer, 16);
  Serial.println(buffer);
}

void setup() {
  Wire.begin();        // join I2C bus (address optional for master)
  delay(50);
  //for(int i = i; i < 44; i++)
  //  eeprom_write(10*i, &text[10*i], 10);
  Serial.begin(9600);  // start serial for output
  //Serial.println(strlen(text));
  readLineE2P();
  pinMode(2,INPUT);
  pinMode(3,INPUT);
}

void loop() {
  if(digitalRead(2)) //up button
  {
    curr_line++;
    readLineE2P();
    delay(1000);
```

```
  }
  if(digitalRead(3)) //down button
  {
    if(curr_line > 0)
      curr_line--;
    readLineE2P();
    delay(1000);
  }
}

void eeprom_write(uint16_t memory_address, uint8_t* data, int _size) {
  Wire.beginTransmission(ADDR);
  //Wire.write((uint8_t)((memory_address & 0xFF00) >> 8));
  Wire.write((uint8_t)((memory_address & 0x00FF) >> 0));
  for (int i = 0; i < _size; i++) {
    Wire.write(data[i]);
    //Serial.print("write: ");
    //Serial.println(data[i]);
  }
  Wire.endTransmission();
  delay(100);
}

void eeprom_read(uint16_t memory_address, uint8_t* data, int _size) {
  Wire.beginTransmission(ADDR);
  //Wire.write((uint8_t)((memory_address & 0xFF00) >> 8));
  Wire.write((uint8_t)((memory_address & 0x00FF) >> 0));
  Wire.endTransmission();

  Wire.requestFrom(ADDR, _size);
  for (int i = 0; i < _size; i++) {
    data[i] = Wire.read();
    //Serial.print("read: ");
    //Serial.println((byte)data[i]);
  }
}
```

توضیح کد:

ابتدا ما باید دو تابع eeprom_read و eeprom_write را تعریف می‌کنیم. در تابع eeprom_write، ابتدا با کمک تابع transmission انتقال به دستگاه جانبی I2C را با آدرس داده شده آغاز می کنیم و داده ها را در ادرس مشخص شده write می کنیم. سپس transmission را به پایان می‌رسانیم. از شروع تا پایان انتقال هرچه را write کنیم وارد صف می‌شود تا از

طریق پروتکل منتقل شود.در اینجا ادرس های مموری ما 16 بیتی هستند و با (uint8_t)((memory_address & 0x00FF)

8 بیت بالای مموری ادرس را میفرستیم.سپس با حلقه for داده ها را میفرستیم. در نهایت انتقال پایان می‌یابد.

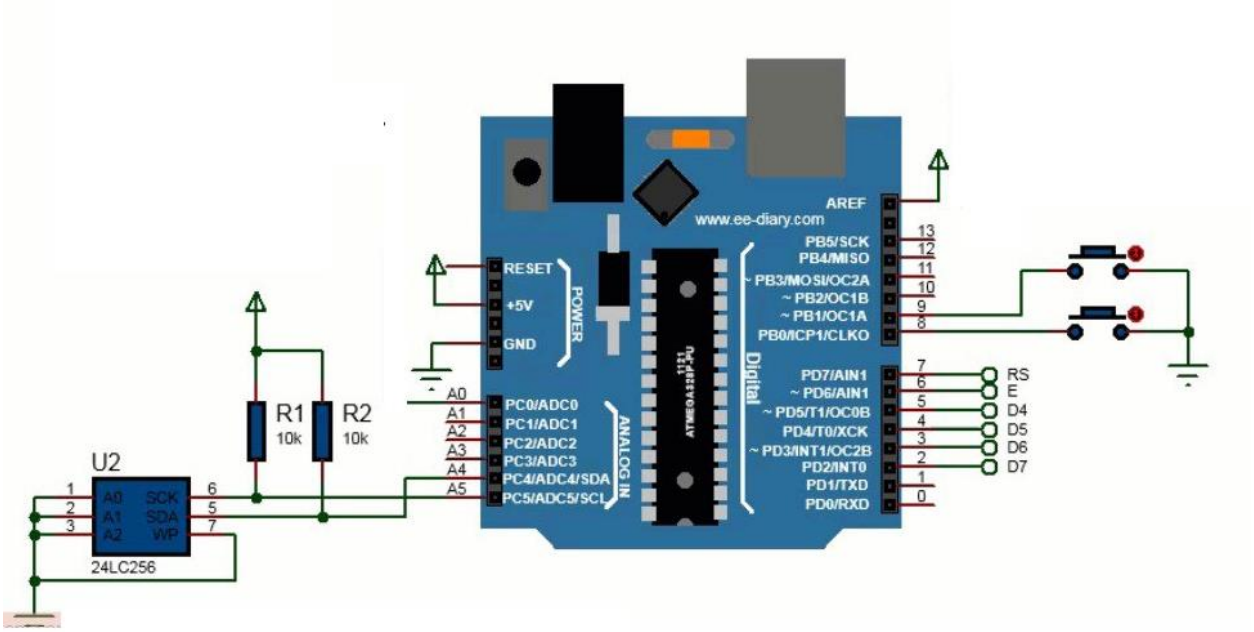
در تابع eeprom_read، ابتدا با کمک تابع transmission انتقال به دستگاه جانبی I2C را با آدرس داده شده آغاز می کنیم و داده ها را در ادرس مشخص شده read می کنیم. سپس transmission را به پایان می‌رسانیم. از شروع تا پایان انتقال هرچه را read کنیم وارد صف می‌شود تا از طریق پروتکل منتقل شود.در اینجا ادرس های مموری ما 16 بیتی هستند و با(uint8_t)((memory_address & 0x00FF)8 بیت بالای مموری ادرس را میفرستیم.سپس درخواست میفرستیم که ادرس مشخص شده به اندازه مشخص شده بخواند و در ارایه data ذخیره کند.

تابع readLineE2P از بافر به اندازه 16 بیت در ادرس مشخص شده میخواند و 16 بیت 16 بیت جلو می‌رود و مقدار خوانده شده از buffer را serial پرینت میکند.

در بخش setup، کتابخانه Wire را راه اندازی کنید و به عنوان Master به گذرگاه I2C بپیوندید.سپس Serial.begin ارتباط سریال را فعال می‌کند.حال readLineE2P را فعال می‌کنیم و پین های 2 و 3 را به عنوان ورودی set می‌کنیم.

حال در بخش Loop، با توجه به اینکه کدام کلید فشار داده شده است خط کنونی را یکی جلو یا عقب می‌بریم و یک داده 16 بیتی را از انجا می‌خوانیم.

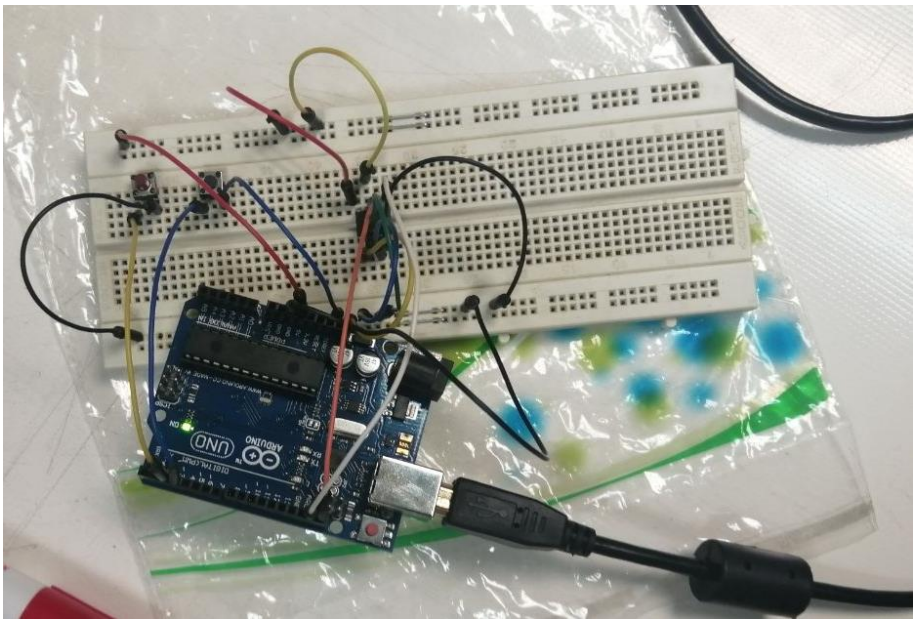
شماتیک مدار بسته شده:



باید دقت شود که در این شماتیک از 24LC256 استفاده شده است ولی ما ازAT24C16 استفاده کرده ایم. در نتیجه باید به اتصال پایه ها باکمک دیتا شیت دقت کرد.

پایه ی 1 تا 4 و7 AT24C16 را به زمین وصل می‌کنیم.پایه ی 5 یاSDA را به پایه ی پایین اردوینو AREF وصل می‌کنیم.پایه ی 6 یا SCL را به پایه ی AREF وصل می‌کنیم. همچنین پایه های 8 و 7 را به VCC وصل می‌کنیم.

مدار بسته شده به صورت عملی:



نتیجه:

خروجی ازمایش خواندن از ممووری

خروجی کد ما به این شکل شد که احتمالا به علت عدم شناسایی فونت نوشته شده در ممووری می‌باشد.