

به نام خدا

«آزمایشگاه ریز پردازنده و زبان اسمبلی»

آزمایش شماره 10: پروژه نهایی – رای گیری الکترونیکی

اعضای گروه: هلیاسادات هاشمی پور ۹۸۳۱۱۰۶-روژینا کاشفی ۹۸۳۱۱۱۸

هدف آزمایش

- چند برد آردوینو که هر کدام با مانیتور و کیبورد یک رای گیری را انجام میدهند و همگی به یک برد مرکزی اطلاعات جمع‌آوری شده را می‌فرستند. برد مرکزی هم در یک EEPROM نتایج رای گیری را ذخیره میکند.
- در این آزمایش یک برد آردوینو به عنوان برد مرکزی و 4 برد به عنوان بردهایی که از طریق آنها رای گیری انجام می‌شود داریم. (در مجموع پنج برد Arduino Mega2560 داریم.) 4 صفحه کلید و 4 ترمینال مجازی، 5 مانیتور، یک بازر، یک EEPROM و تعدادی LED داریم.



کد

master

کتابخانه‌های موردنظر را برای LCD و خواندن و نوشتن در حافظه اضافه و مقادیر ثابت را define کرده و سوال و جواب‌های مربوط به رای‌گیری هم در سه رشته ذخیره می‌کنیم.

```
#include <LiquidCrystal.h>
#include <Wire.h>
#define DEVICE_ADDRESS 0b1010000
#define MODE_MEMORY_ADDR 100
#define BUZZER 49

boolean voted = false;

//LCD
const int rs = 27, en = 28;
const int d4 = 29, d5 = 30, d6 = 31, d7 = 32;
//Q length Between 0 and 16
String Q = "Choose A Number: ";
//ANS1&ANS2 length Between 0 and 6
String ANS1 = "1.One";
String ANS2 = "2.Two";
String v[4] = {"", "", "", ""};
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
```

```

void readVotes(){
    uint8_t data_saved[4];
    eepromRead(MODE_MEMORY_ADDR, data_saved, 4);
    int votel = 0;
    int vote2 = 0;
    for(int i=0; i<4; i++){
        if(data_saved[i] <= '2'){
            v[i] += data_saved[i];
            vote2++;
        }
        else if(data_saved[i] == '1'){
            votel++;
            v[i] += data_saved[i];
        }
    }
    rewriteLCD("Total Votes:", 0);
    lcd.print(String(votel+vote2));
    rewriteLCD(ANS1 + ":", 1);
    lcd.print(String(votel));
    lcd.print(" "+ANS2+":");
    lcd.print(String(vote2));
}

```

این تابع رای‌هایی که در eeprom ثبت شده را می‌خواند و در یک حلقه رای مربوط به هر فرد را در عنصر آرایه مربوط به خودش ذخیره می‌کند و مجموع رای مربوط به هر گزینه را محاسبه کرده و نتیجه را در LCD مربوط به خودش (مستر) چاپ می‌کند.

```

void setVote(char temp[]){
    int index = 0;
    for(int i=0; i<question.length(); i++){
        temp[index] = question[i];
        index+=1;
    }
    temp[index]='\n';
    index+=1;

    for(int i=0; i<answer1.length(); i++){
        temp[index] = answer1[i];
        index+=1;
    }
    temp[index]='\n';
    index+=1;

    for(int i=0; i<answer2.length(); i++){
        temp[index] = answer2[i];
        index+=1;
    }
    temp[index]='\n';
    index+=1;
}

```

این تابع یک آرایه از کاراکترها دریافت می‌کند و مقادیر ثابتی که در سه رشته مربوط به سوال و دو جواب ذخیره شده‌اند را در این آرایه می‌ریزد. ابتدا در حلقه اول رشته سوال و در دو حلقه بعدی رشته‌های جواب ذخیره می‌شوند.

```

void sendQandA() {
    char temp[question.length() + answer1.length() + answer2.length() + 3];
    setVote(temp);
    char votedMsg[] = {'v','o','t','e','d','\n',' ','\n',' ','\n'};
    delay(50);
    if(v[1] == "")
        Serial1.write(temp);
    else
        Serial1.write(votedMsg);
    Serial1.flush();
    delay(10);

    if(v[2] == "")
        Serial2.write(temp);
    else
        Serial2.write(votedMsg);
    Serial2.flush();
    delay(10);

    if(v[3] == "")
        Serial3.write(temp);
    else
        Serial3.write(votedMsg);
    Serial3.flush();
    delay(10);

    if(v[0] == "")
        Serial.write(temp);
    else
        Serial.write(votedMsg);
    Serial.flush();
    delay(10);
}

```

در این تابع یک رشته به طول مجموع سوال و دو جواب ساخته شده و تابع **setVote** ورودی داده می‌شود تا مقادیر سوال و جواب در آن ریخته شوند. یک رشته هم برای نگهداری پیام **voted** ساخته می‌شود. سپس برای هر کدام از افراد چک می‌شود که اگر عنصر مربوط به رای آن‌ها خالی بود، یعنی هنوز رای نداده‌اند و پیام سوال و جواب در ترمینال مربوط به آن‌ها نوشته می‌شود. در غیر این صورت یعنی رای دریافت شده و چون هر فردی فقط یک بار مجاز به رای دادن است، پیام **voted** در ترمینال چاپ شده و سپس بافر مربوطه خالی می‌شود.

```
void rewriteLCD(String s, int row){
    lcd.setCursor(0, row);
    lcd.print("                ");
    lcd.setCursor(0, row);
    lcd.print(s);
}
```

یک رشته و یک عدد به عنوان سطر می‌گیرد و رشته ورودی را در سطر ورودی در LCD چاپ می‌کند.

```
void eepromWrite(uint8_t memory_address, uint8_t data){
    Wire.beginTransmission(DEVICE_ADDRESS);
    Wire.write(memory_address);
    Wire.write(data);
    Wire.endTransmission();
}
```

مقدار data ورودی را در memory address ورودی می‌نویسد.

```
void eepromRead(uint8_t memory_address, uint8_t *data, uint8_t _size){
    Wire.beginTransmission(DEVICE_ADDRESS);
    Wire.write(memory_address);
    Wire.endTransmission();
    Wire.requestFrom(DEVICE_ADDRESS, _size);
    for(int i=0; i<_size; i++){
        data[i] = Wire.read();
    }
}
```

مقدار data ورودی را از memory address ورودی به اندازه size می‌خواند.

```
void setup() {
    Serial.begin(9600);
    Serial1.begin(9600);
    Serial2.begin(9600);
    Serial3.begin(9600);

    Wire.begin();

    lcd.begin(16, 2);
    lcd.clear();

    pinMode(BUZZER, OUTPUT);
    readVotes();
    sendQandA();
}
```

برای هر کدام از اسلیوها یک ترمینال در نظر گرفته شده و تابع Serial.begin فراخوانی می‌شود. توابع مربوط به ارتباط با حافظه و کار با LCD فراخوانی شده و پین مربوط به بازر هم به عنوان خروجی مشخص می‌شود و سپس دو تابع readVotes و sendQandA به ترتیب صدا زده می‌شوند.

```

void loop() {
  if(Serial.available() && v[0] == ""){
    char ch = (char) Serial.read();
    v[0] += ch;
    rewriteLCD("Device4:", 0);
    rewriteLCD(v[0], 1);

    uint8_t tempData = ch;
    eepromWrite(MODE_MEMORY_ADDR, tempData);
    tone(BUZZER, 400);
    voted = true;
  }
  if(Serial1.available() && v[1] == ""){
    char ch1 = (char) Serial1.read();
    v[1] += ch1;
    rewriteLCD("Device1:", 0);
    rewriteLCD(v[1], 1);

    uint8_t tempData = ch1;
    eepromWrite(MODE_MEMORY_ADDR + 1, tempData);
    tone(BUZZER, 400);
    voted = true;
  }
  if(Serial2.available() && v[2] == ""){
    char ch2 = (char) Serial2.read();
    v[2] += ch2;
    rewriteLCD("Device2:", 0);
    rewriteLCD(v[2], 1);

    uint8_t tempData = ch2;
    eepromWrite(MODE_MEMORY_ADDR + 2, tempData);
    tone(BUZZER, 400);
    voted = true;
  }
  if(Serial3.available() && v[3] == ""){
    char ch3 = (char) Serial3.read();
    v[3] += ch3;
    rewriteLCD("Device3:", 0);
    rewriteLCD(v[3], 1);

    uint8_t tempData = ch3;
    eepromWrite(MODE_MEMORY_ADDR + 3, tempData);
    tone(BUZZER, 400);
    voted = true;
  }
  if(voted){
    delay(300);
    readVotes();
    voted = false;
  }
}

```

در هر شرط چک می‌شود که اگر در بافر ترمینال چیزی برای خواندن وجود داشت و فرد مربوطه هنوز رای را ثبت نکرده بود، یعنی عنصر آرایه رای مربوط به آن فرد خالی بود، کاراکتر مربوطه از ترمینال خوانده شود (توسط کیپد گرفته شده است) و سپس در LCD مربوط به مستر نوشته شود که دیوایس مربوط به آن فرد، رای را ثبت کرده و به چه گزینه‌ای رای داده است. سپس این کاراکتر با فراخوانی `eepromWrite` در حافظه نوشته و ذخیره می‌شود. با تابع `tone` بازر فعال شده و متغیر `sbVoted` به `true` تغییر مقدار می‌دهد که یعنی فردی رای داده است. این مراحل برای هر چهار نفر در چهار شرط تکرار شده و در شرط آخر، چک می‌شود که اگر فردی رای داده بود، با فراخوانی `readVote` مقادیر به روز رسانی شده در LCD مستر چاپ شوند.

slave

ابتدا کتابخانه‌های مورد نظر، اطلاعات مربوط به keypad، LCD، LEDها و متغیرهای مورد استفاده تعریف می‌شوند.

```
#include <LiquidCrystal.h>
#include <Keypad.h>
//Keypad
const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
char keys[ROWS][COLS] = {
  {'7','8','9','/'},
  {'4','5','6','*'},
  {'1','2','3','-'},
  {'0','0','=','+'}
};
byte rowPins[ROWS] = {6, 7, 8, 9}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {2, 3, 4, 5}; //connect to the column pinouts of the keypad
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
//LCD
const int rs = 24, en = 26, d4 = 32, d5 = 33, d6 = 34, d7 = 35;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
//LED
const uint8_t LED_PINS[] = {41, 42};
//Vars
String Question = "";
String answers1 = "";
String answers2 = "";
String temp = "";
bool isQuestionAndAnswerSet = false;
bool hasSubmitted = false;
```

```

void updateLCD(bool hasAnswer = true){
    lcd.setCursor(0, 0);
    lcd.print(Question);
    if(hasAnswer){
        lcd.setCursor(0, 1);
        String s = answers1 + answers2;
        lcd.print(s);
    }
}

```

ابتدا کرسر در ابتدای
نمایشگر تنظیم می‌شود،
گزینه‌ها به او نمایش داده
می‌شوند.

```

void setQuestionAndAnswer(){
    if(Serial.available()){
        char ch = (char) Serial.read();
        Serial3.print(ch);
        if(ch == '\n'){
            if(Question == "")
                Question = temp;
            else if(answers1 == "")
                answers1 = temp;
            else{
                answers2++++ = temp;
                isQuestionAndAnswerSet = true;
                if(Question == "voted"){
                    hasSubmitted = true;
                    Serial.end();
                    updateLCD(false);
                    return;
                }
            }
            updateLCD();
            Serial3.println();
        }
        temp = "";
    }else{
        temp += ch;
    }
}

```

در ابتدا با چک کردن
serial.available بررسی
می‌کند که آیا بایت‌هایی
برای خواندن از پورت
سریال وجود دارند که در
بافر ذخیره شده باشند یا
خیر. اگر وجود داشتند،
شروع به خواندن آن‌ها که
داده‌های ارسالی از سمت
مستر هستند میکند. در
این تابع با توجه به داده‌ی
ارسالی این مورد که
شخص رای داده است یا نه
بررسی و مشخص می‌گردد
و با پر کردن رشته‌هایی که
در ابتدای کد تعریف شده
بودند، به **updateLCD**
می‌گوید چه چیزی باید
نمایش دهد.

این تابع در واقع بررسی می‌کند که آیا کلید فشرده شده از جز اعداد هست یا خیر. تا اگر به طور مثال کاربر کلید = را فشرده بود به او اخطار داده شود.

```
bool isKeyNumber(char key){  
    int keyAsciiCode = (int)key;  
    return keyAsciiCode >= (int)'1' && keyAsciiCode <= (int)'9';  
}
```

کلید فشرده شده را می‌خواند، چک می‌کند اگر گزینه بود پس از زدن * گزینه‌ی انتخابی ثبت شود، LCD به روزرسانی شود و LED متناسب با گزینه‌ی انتخاب شده روشن گردد. معتبر بودن یا نبودن گزینه‌ی انتخابی هم با تابع isKeyNumber بررسی می‌شود.

```
void readKeypad(){  
    char key = keypad.getKey();  
    if(key){  
        Serial3.println(key);  
        static int keyNum;  
        if(isKeyNumber(key)){  
            keyNum = (int) key;  
            keyNum -= (int)'0';  
        }else if(key == '*'){  
            if(keyNum >= 1 && keyNum <= 2){  
                //turn on LED  
                digitalWrite(LED_PINS[keyNum-1], HIGH);  
                delay(500);  
                lcd.clear();  
                lcd.setCursor(0, 0);  
                lcd.print("voted");  
                saveVote(keyNum);  
                Serial3.print("Vote submitted: ");  
                Serial3.println(keyNum);  
                hasSubmitted = true;  
            }else{  
                Serial3.println("Choose between 1 and 2");  
            }  
        }  
    }  
}
```

```

void saveVote(int vote){
    char v[] = {(int)'0' + vote};
    delay(50);
    Serial.write(v);
    Serial.flush();
    delay(10);
    Serial.end();
    Serial3.println("Sent!");
}

```

گزینه‌ی انتخابی را نمایش می‌دهد
و پس از ارسال اعلام می‌کند.

```

void loop() {
    if(!isQuestionAndAnswerSet){
        setQuestionAndAnswer();
    }
    else if(!hasSubmitted){
        readKeypad();
    }
}

```

اگر باید سوال و گزینه نمایش داده
شود، این اتفاق بیفتد و اگر هنوز
شخص رای نداده است کیپد مدام
بررسی گردد.

```

void setup() {
    Serial.begin(9600);
    Serial3.begin(9600);

    lcd.begin(16, 2);
    lcd.clear();

    for(int i=0; i<2; i++){
        pinMode(LED_PINS[i], OUTPUT);
    }
}

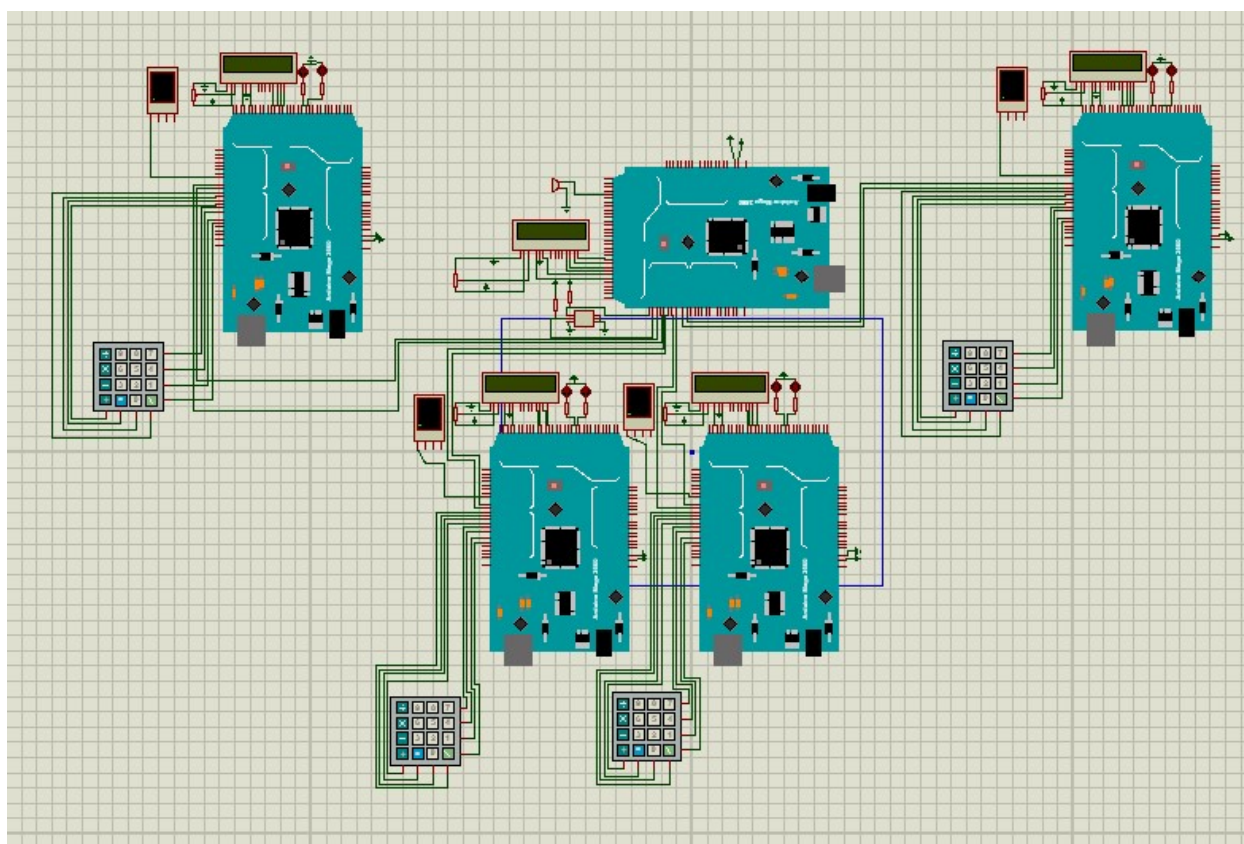
```

سریال مربوطه را تعریف
میکنیم، LCD را پاک و آماده‌ی
نمایش جدید و از آنجا که هر
برد اسلیو دو LED دارد،
پایه‌های LEDها را تعریف
می‌کنیم.

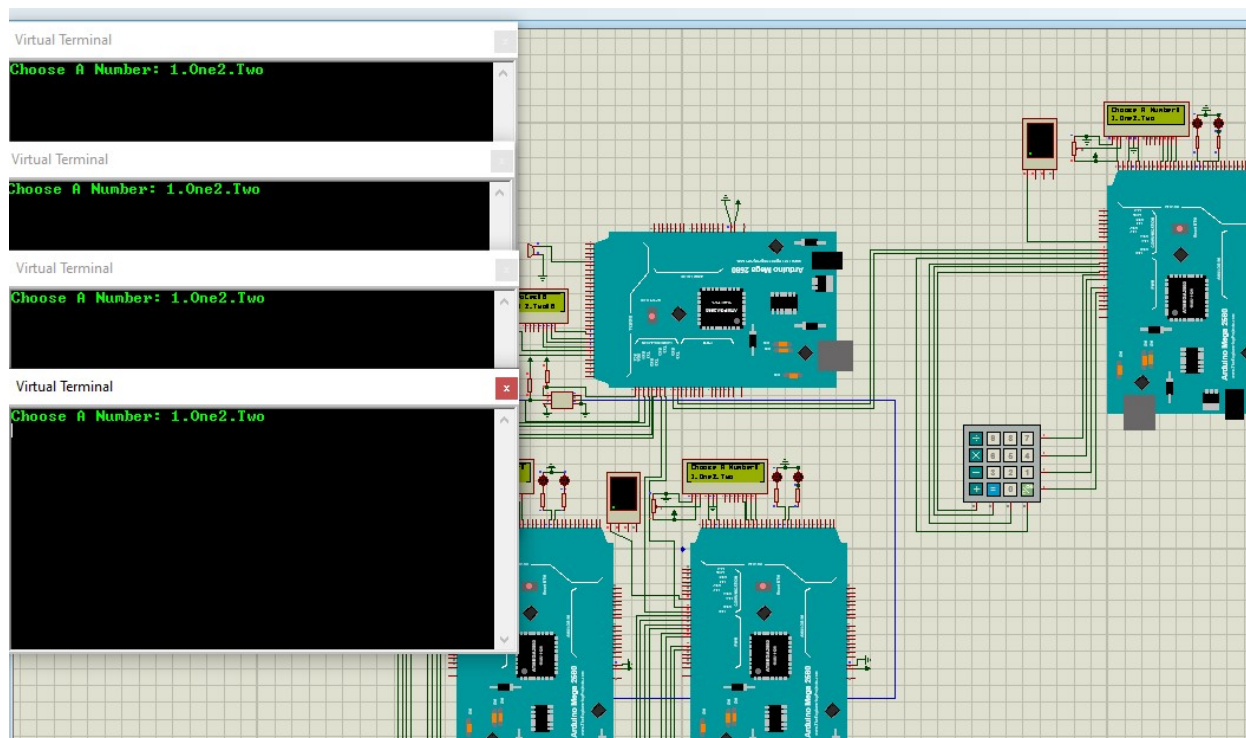
خروجی

نحوه کار به این صورت است که سوال و گزینه های موجود برای رای گیری از سمت برد مرکزی به 4 برد دیگر ارسال می شود و روی مانیتور آنها نمایش داده می شود. هر یک از این 4 برد یک کیبورد دارند که به کمک آن گزینه انتخاب شده برای ثبت رای وارد می شود. اگر ثبت رای موفقیت آمیز باشد در مانیتور متصل به برد پیام ثبت رای چاپ می شود، همچنین LED متصل به این برد ها نشان می دهد که هر برد به کدام گزینه رای داده است. هنگام ثبت رای بازر متصل به برد مرکزی به صدا در می آید و سپس در مانیتور متصل به آردوینوی مرکزی نتایج رای گیری update و چاپ میشوند. همچنین رای داده شده در EEPROM ذخیره می گردد.

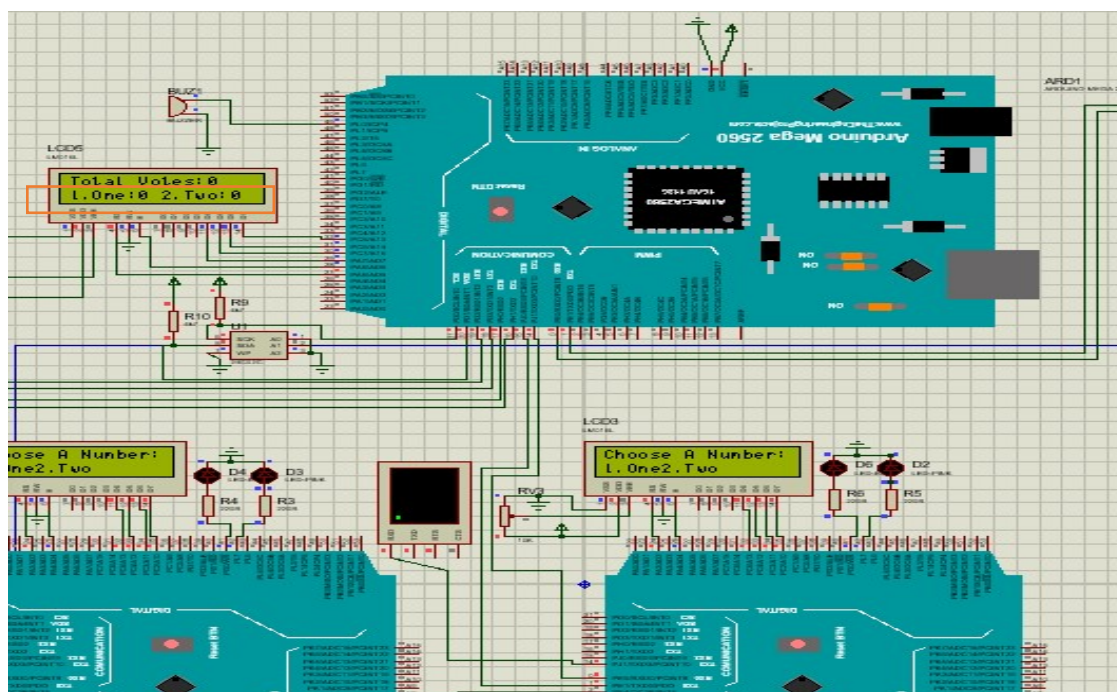
مدار طراحی شده به صورت زیر است:



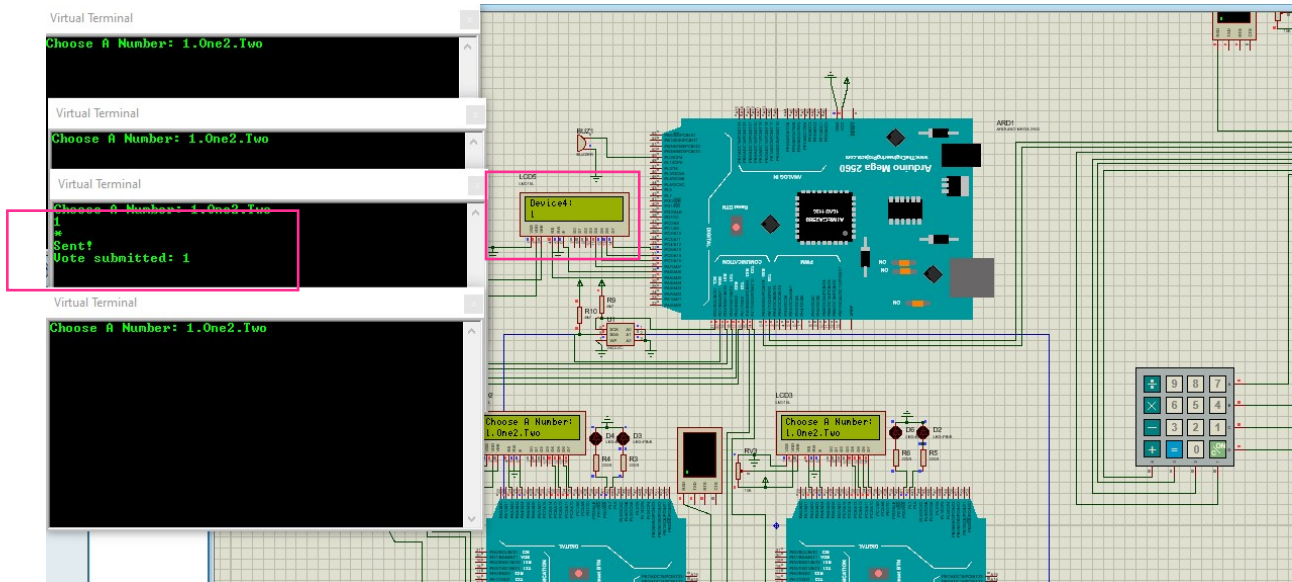
زمانی که در پروتیوس ران می گیریم شکل زیر حاصل می شود.



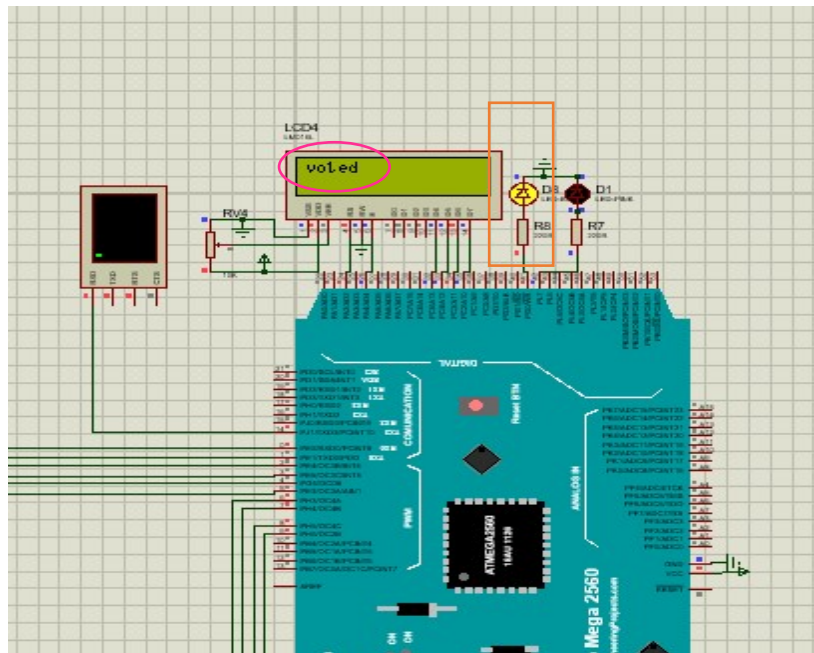
اگر توجه کنیم در همه ی ترمینال ها سوال رای گیری و گزینه های موجود برای انتخاب نوشته شده است. مانیتورها سوال و جواب چاپ شده است همانطور که می بینیم در مانیتور مرکزی تعداد رای های داده شده به جواب ها نوشته شده است.(که صفر است)



حال اگر با یکی از بردها رای بدهیم نتایج به صورت زیر است.



در اصل در برد مرکزی چاپ میشود که دستگاه شماره ۴ رای داد. سپس میبینیم که در مانیتور آن برد چاپ میشود که رای داده شده است. و LED زرد روشن میشود چون گزینه 1 را انتخاب کرده است. و در مانیتور برد مرکزی تعداد کل رای ها یک شد و تعداد رای های هرگزینه به روزرسانی شد.



برنامه را استپ کرده و دوباره ران می کنیم. میبینیم که برای دستگاه هایی، نوشته شده که رای داده است. و در برد مرکزی هم تعداد رای ها به درستی نشان داده شده اند.

