**Руководство Администратора для Устройства Мониторинга игры «Собери последовательность» на Arduino Mega 2560**

**Оглавление**

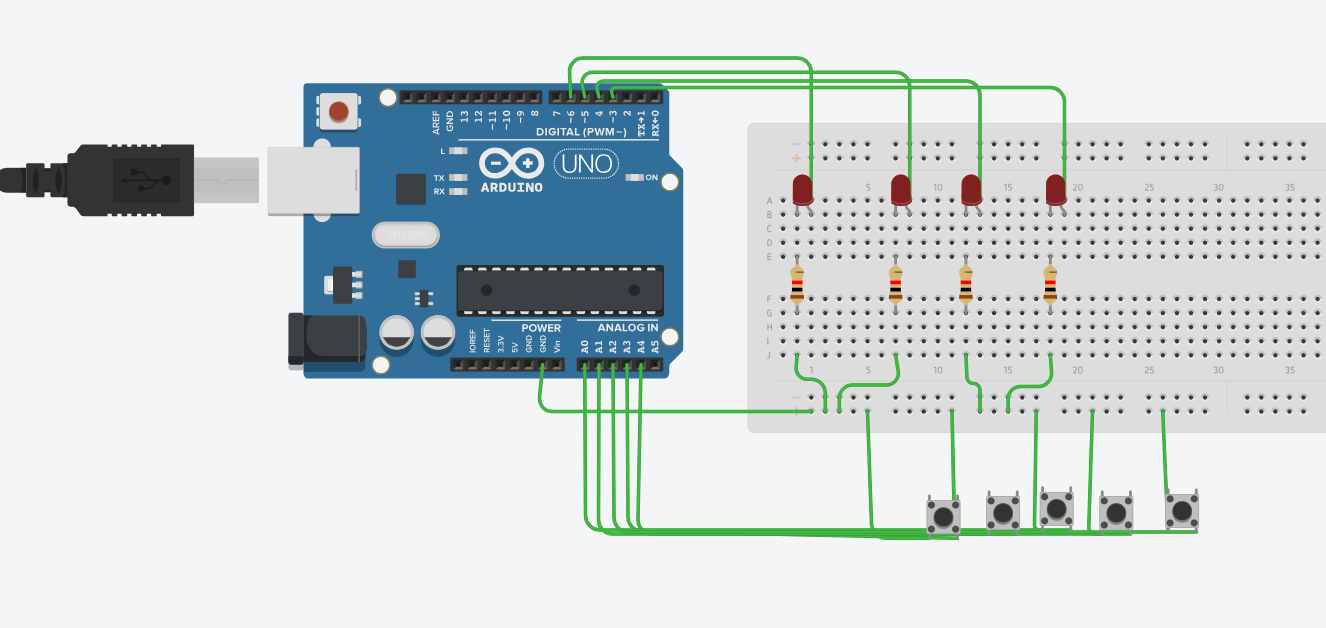
**Введение**

1. **Установка и Подготовка**
   1. **Установка Arduino IDE**
   2. **Подключение устройства**
   3. **Загрузка Программы на Arduino Mega 2560**
2. **Обновление Программного Обеспечения.**
   1. **Загрузка Новой Версии Программы**
3. **Настройки Дополнительных Функций**
   1. **Изменение Периода Мониторинга**
   2. **Добавление Дополнительных светодиодов и кнопок.**
4. **Обслуживание и Решение Проблем**
   1. **Обновление Firmware Микроконтроллера**
   2. **Решение Ошибок и Проблем**
5. **Установка и Подготовка**
   1. **Установка Arduino IDE**

Для управления программным обеспечением устройства, убедитесь, что на вашем компьютере установлено Arduino IDE. Вы можете скачать его с официального сайта Arduino и следовать инструкции по установке.

* 1. **Подключение устройства**

1. Подключите Arduino Mea 2560 к компьютеру с помощью USB-кабеля.

2. Убедитесь, что все светодиоды подключены к соответствующим портам 

(Рис 1)

* 1. **Загрузка Программы на Arduino Mega 2560**

1. Откройте Arduino IDE.

2. Откройте программу мониторинга, представленную в приложении документации.

3. Выберите правильный порт и плату в меню «Инструменты»

4. Загрузите программу на Arduino Mega 2560, нажав «Загрузить».

1. **Обновление Программного Обеспечения.**
   1. **Загрузка Новой Версии Программы**

1. Получите новую версию программы мониторинга из официального источника.

2. Откройте Arduino IDE и программу мониторинга.

3. Загрузите новую версию программы на Arduino Mega 2560.

1. **Настройки Дополнительных Функций**
   1. **Изменение Периода Мониторинга**

1. Откройте программу мониторинга в Arduino IDE

2. Найдите соответствующую часть кода, отвечающую за период мониторинга.

3. Изменение периода по необходимости.

4. Загрузите изменённую программу на Arduino Mega 2560.

* 1. **Добавление Дополнительных светодиодов и кнопок.**

1. Подключите кнопки и светодиоды к свободному порту Arduino Mega 2560.

2. В программе добавьте необходимые данные в код.

3. Проверьте и адаптируйте программу к новым кнопкам и светодиодам.

4. Загрузите изменённую программу на Arduino Mega 2560.

1. **Обслуживание и Решение Проблем**
   1. **Обновление Firmware Микроконтроллера**

1. Периодически проверяйте наличие обновлений прошивки микроконтроллера Arduino Mega 2560.

2. Посетите официальный сайт Arduino для получения инструкций по обновлению прошивки.

* 1. **Решение Ошибок и Проблем**

1. При обнаружении ошибок в программе, обратитесь к логам компиляции в Arduino IDE для получения дополнительной информации.

2. Проверьте подключение светодиодов, и убедитесь, что все компоненты работают правильно.

3. В случае сложных проблем обратитесь к документации проекта или обращайтесь за поддержкой в сообществе Arduino.

Эти рекомендации помогут вам эффективно управлять устройством игры **«Собери последовательность».** Следуйте инструкциям и периодически обновляйте программное и аппаратное обеспечение для поддержания стабильной работы системы.

**Приложение:**

// пин для пьезодинамика

#define buzzer 11

// пины для светодиодов

int led[] = {6,5,4,3};

// пины для кнопок

int but[] = {A4,A0,A1,A2,A3};

// варианты комбинаций

char\* kod[] = {"0132", "0321", "2301", "2130", "2310", "2031", "2013", "0123", "1320", "1203", "1230", "3021", "3012"};

int del = 1000;

int w = -1;

byte level = 1; // Начальный уровень

byte b1, b2, b3, b4, b5;

String KOD, OTV, A;

// Расчёт элементов в массивах

byte num\_led = sizeof(led) / sizeof(int\*); // sizeof используется для получения размера каждого из элементов массива

byte num\_but = sizeof(but) / sizeof(int\*);

byte num\_cod = sizeof(kod) / sizeof(char\*);

byte len = strlen(kod[0]);

// Проверяем, находятся ли кнопки в выключенном состоянии

boolean button1WasUp = true;

boolean button2WasUp = true;

boolean button3WasUp = true;

boolean button4WasUp = true;

boolean button5WasUp = true;

void setup() {

pinMode(buzzer, OUTPUT); // устанавливает режим работы для порта с выходом сигнала.

for (byte i = 0; i <= num\_led; i++) { // цикл for, который итерируется от 0 до количества LED.

pinMode(led[i], OUTPUT); // устанавливает режим работы вывода, который используется для управления светодиодом.

}

//Инициализация пинов, связанных с кнопками

for (byte i = 0; i <= num\_but; i++) {

pinMode(but[i], INPUT\_PULLUP);

}

Serial.begin(9600); //устанавливает соединения с между компьютером и микроконтроллером

Serial.println("LEVEL - " + String(level)); // выводит текущий уровень

Serial.println("");

}

void loop() {

// выбираем комбинацию

while (w == -1) {

int x = random(0, num\_cod - 1); //генерируем случайное число в диапазоне 0, num cod - 1

KOD = {kod[x]}; // меняем значение переменной на, полученное из массива по индексу x

b1 = 0; b2 = 0; b3 = 0; b4 = 0; b5 = 0;

w = 0;

}

// поочередно включаем светодиоды

while (w <= len - 1) {

A = KOD[w];

int a = A.toInt();

digitalWrite(led[a], HIGH);

tone(buzzer, 4 \* led[a]);

delay(10);

noTone(buzzer);

delay(del);

digitalWrite(led[a], LOW);

w++;

}

// ждем нажатия кнопок

while (w >= len - 1 and w <= len + len) {

boolean button1IsUp = digitalRead(but[0]);

boolean button2IsUp = digitalRead(but[1]);

boolean button3IsUp = digitalRead(but[2]);

boolean button4IsUp = digitalRead(but[3]);

boolean button5IsUp = digitalRead(but[4]);

if (b5 == 0 and button5WasUp and !button5IsUp) { // Кнопка сброса уровня

Serial.println("");

Serial.println("Сброс!"); // Печатает слово «Сброс»

Serial.println("");

tone(buzzer, 100);

delay(1000);

noTone(buzzer);

Serial.println("LEVEL - " + String(level)); //Отображает текущий уровень

for (byte i = 0; i <= num\_led; i++) {

digitalWrite(led[i], LOW);

}

delay(2000);

KOD = "";

OTV = "";

w = -1;

}

if (b1 == 0 and button1WasUp and !button1IsUp) { //Нажатие первой кнопки

delay(10);

button1IsUp = digitalRead(but[0]);

if (!button1IsUp) {

tone(buzzer, 4 \* led[0]);

delay(10);

noTone(buzzer);

OTV = OTV + "0";

digitalWrite(led[0], HIGH);

b1 = 1;

w++;

}

}

button1WasUp = button1IsUp;

if (b2 == 0 and button2WasUp and !button2IsUp) { //Нажатие второй кнопки

delay(10);

button2IsUp = digitalRead(but[1]);

if (!button2IsUp) {

tone(buzzer, 4 \* led[1]);

delay(10);

noTone(buzzer);

OTV = OTV + "1";

digitalWrite(led[1], HIGH);

b2 = 1;

w++;

}

}

button2WasUp = button2IsUp;

if (b3 == 0 and button3WasUp and !button3IsUp) { //Нажатие третьей кнопки

delay(10);

button3IsUp = digitalRead(but[2]);

if (!button3IsUp) {

tone(buzzer, 4 \* led[2]);

delay(10);

noTone(buzzer);

OTV = OTV + "2";

digitalWrite(led[2], HIGH);

b3 = 1;

w++;

}

}

button3WasUp = button3IsUp;

if (b4 == 0 and button4WasUp and !button4IsUp) { //Нажатие четвёртой кнопки

delay(10);

button4IsUp = digitalRead(but[3]);

if (!button4IsUp) {

tone(buzzer, 4 \* led[3]);

delay(10);

noTone(buzzer);

OTV = OTV + "3";

digitalWrite(led[3], HIGH);

b4 = 1;

w++;

}

}

button4WasUp = button4IsUp;

// если правильно - начинаем новый уровень

if (w == len + len and OTV == KOD) {

Serial.println("");

Serial.println("DA!"); // Печатает слово «DA!»

Serial.println("");

level++; //Повышает уровень на 1

Serial.println("LEVEL - " + String(level)); //Отображает текущий уровень

del = del - 100;

if (del < 10) { del = 100; }

delay(1000);

for (byte i = 0; i <= num\_led; i++) {

digitalWrite(led[i], LOW);

}

delay(2000);

KOD = "";

OTV = "";

w = -1;

}

// если неправильно - задаем новую комбинацию

if (w == len + len and OTV != KOD) {

Serial.println("");

Serial.println("HET!"); // Печатает слово «НЕТ!»

Serial.println("");

tone(buzzer, 100);

delay(1000);

noTone(buzzer);

Serial.println("LEVEL - " + String(level)); //Отображает текущий уровень

for (byte i = 0; i <= num\_led; i++) {

digitalWrite(led[i], LOW);

}

delay(2000);

KOD = "";

OTV = "";

w = -1;

}

}

}