

ПАСПОРТ
Аппаратно-программный комплекс
«Интерактивный Кубик Рубика Cube King»

Ответственный исполнитель

Д.С. Марущенко

К.А. Емельянов

Комсомольск-на-Амуре 2018

Содержание

1	Общие положения	3
1.1	Наименование проект.....	3
1.4	Сведения об использованных при проектировании нормативно-технических документах	3
2	Назначение и принцип действия	5
2.1	Назначение проекта.....	5
2.2	Области использования проекта.....	5
2.3	Принцип действия	5
3	Состав проекта и комплектность.....	6
4	Устройство и описание работы проекта	7
4.1	Устройство проекта.....	7
4.2	Описание работы проекта	8
5	Условия эксплуатации	9
5.1	Правила и особенности размещения проекта	9
5.2	Меры безопасности	9
5.3	Правила хранения и транспортирования	10
	ПРИЛОЖЕНИЕ А	11

						Лист
						2
Изм.	Лист	№ докум.	Подпись	Дата		

1 Общие положения

Настоящий паспорт является документом, предназначенным для ознакомления с основными техническими характеристиками, устройством, правилами установки и эксплуатации устройства «Интерактивный Кубик Рубика Cube King» (далее «проект»).

Паспорт входит в комплект поставки проекта. Прежде, чем пользоваться проектом, внимательно изучите правила обращения и порядок работы с ним. В связи с постоянной работой по усовершенствованию проекта, повышающей его надежность и улучшающей условия эксплуатации, в конструкцию могут быть внесены изменения, не отраженные в данном издании.

1.1 Наименование проекта

Полное наименование системы – аппаратно-программный комплекс «Интерактивный Кубик Рубика Cube King».

1.2 Сведения об использованных при проектировании нормативно-технических документах

При проектировании использованы следующие нормативно-технические документы:

ГОСТ 2.001-2013. Единая система конструкторской документации. Общие положения.

ГОСТ 2.102-2013. Единая система конструкторской документации. Виды и комплектность конструкторских документов.

ГОСТ 2.105-95. Единая система конструкторской документации. Общие требования к текстовым документам.

ГОСТ 2.610-2006. Единая система конструкторской документации. Правила выполнения эксплуатационных документов.

									Лист
									3
Изм.	Лист	№ докум.	Подпись	Дата					

ГОСТ 2.004-88. Единая система конструкторской документации. Общие требования к выполнению конструкторских технологических документов на печатающих и графических устройствах вывода ЭВМ.

ГОСТ 2.051-2006. Единая система конструкторской документации. Электронные документы. Общие положения.

ГОСТ 2.052-2006. Единая система конструкторской документации. Электронная модель проекта. Общие положения.

ГОСТ 2.601-2013. Единая система конструкторской документации. Эксплуатационные документы.

										Лист
										4
Изм.	Лист	№ докум.	Подпись	Дата						

2 Назначение и принцип действия

2.1 Назначение проекта

Проект предназначен для развлекательно – соревновательных целей, а конкретно для замены по удобству и скорости уже существующий Кубик Рубика.

2.2 Области использования проекта

Данный проект может применяться как для домашнего использования, так и для соревновательных целей

2.3 Принцип действия

Принцип действия такой же как и у обычного Кубика Рубика, но только вся логика граней, и их переключения легла на плечи микроконтроллера и индикаторов.

Основные функции:

- 1) Индикация граней кубика
- 2) Кулачковое переключение сторон (с помощью небольшого поворота грани на 0.5 – 1 см, для переключения цвета).
- 3) Случайная генерация цвета граней, согласованная с логикой кубика (для этого необходимо нажать на любые две грани, и удерживать в таком положении 5 секунд для его “запутывания”).

									Лист
									5
Изм.	Лист	№ докум.	Подпись	Дата					

3 Состав проекта и комплектность

В комплект поставки входит:

- Кубик Рубика “CubeKing”
- Зарядное устройство
- Паспорт.

Использованные материалы и компоненты

Наименование	Количество, шт
Arduino pro nano	1
Кнопка контактная	12
RGB светодиод	54
Аккумулятор Li-ion 3.7v 780мАч	1
Преобразователь DC-DC 3.7В – 5В	1
Монтажный провод, фторопласт 0.2 мм ²	4 метра
Монтажная плата 5x5 см	1

						Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

4 Устройство и описание работы проекта

4.1 Устройство проекта

Структурная схема проекта представлена на рисунке 1.

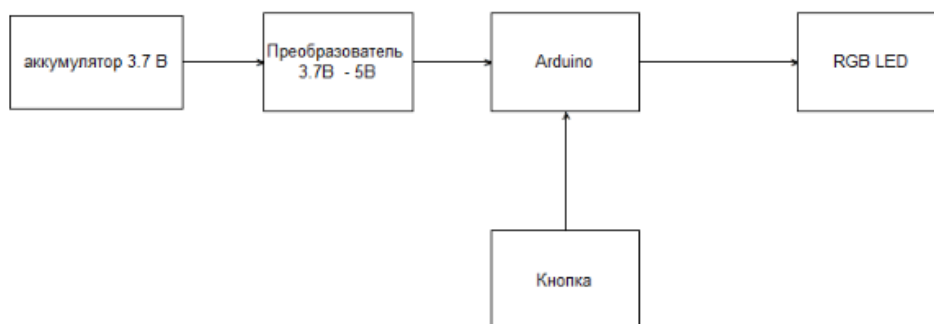


Рисунок 1 – Структурная схема проекта

Устройство запитывается от преобразователя напряжения 3.7 В – 5 В, на который подается номинальное напряжение аккумулятора 3.7 В. Led лента запитывается от входа микроконтроллера.

Управление всеми модулями выполняется контроллером Arduino pro mini через последовательные интерфейсы.

Принципиальная электрическая схема приведена в приложении А.

4.2 Описание работы проекта

Перед началом использования, устройство необходимо зарядить, а также удостовериться не контактировало ли оно с влагой.

Принцип работы основных функций:

1) Устройство передает в COM порт информацию о текущем расположении сторон Кубика, при нажатии на кнопку движения грани, в COM порт выводится одна команда из 12 возможных, которая меняет цветовое расположение граней на ту, которую задал пользователь. При различном расположении попарно расположенных граней в течении 5 секунд, кубик из пула команд случайным образом, в количестве, 50 раз выбирает одну команду из 12, таким образом, запутывая кубик для дальнейшего времяпрепровождения с ним. При перезапуске устройства предыдущие параметры сохраняются.

						Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

5 Условия эксплуатации

Проект выпускается в климатическом исполнении УХЛ 4.2 по ГОСТ 15150-69 и предназначен для использования в стационарных условиях в закрытых помещениях при соответствующих климатических условиях:

- интервал температур от +10 до +35 °С;
- относительная влажность воздуха до 80 % при температуре +25 °С;
- высота над уровнем моря не более 2000 м;
- атмосферное давление от 86,6 до 106 кПа (от 650 до 800 мм рт. ст.).

В помещении, где используется проект не должно возникать условий для конденсации влаги (выпадения росы).

Для обеспечения безотказной работы, сохранения точности и его сбережения необходимо соблюдать следующие правила:

- изучить паспорт, прежде чем приступить к работе с проектом;
- предохранять проект от ударов и повреждений;
- не допускать самостоятельную разборку проекта.

5.1 Правила и особенности размещения проекта

ВНИМАНИЕ! При эксплуатации проекта запрещается проводить самостоятельно какие-то либо работы по извлечению и установке внутренних компонентов проекта.

5.2 Меры безопасности

Необходимо соблюдать требования техники безопасности и следующие меры предосторожности:

- не оставлять проект включенным без наблюдения;
- после транспортировки в холодное время года проект необходимо выдержать при комнатной температуре не менее двух часов;

									Лист
									9
Изм.	Лист	№ докум.	Подпись	Дата					

- внутренние осмотры и ремонт проекта должны производиться только квалифицированными специалистами;
- не устанавливайте проект на неустойчивой подставке, стойке или ненадежном кронштейне.

5.3 Правила хранения и транспортирования

Транспортирование проекта в упакованном виде может производиться железнодорожным, автомобильным (в закрытых транспортных средствах), воздушным, речным и морским видами транспорта в соответствии с правилами перевозок грузов, действующих на транспорт данного вида. Условия транспортирования проекта по части воздействия климатических факторов должны соответствовать группе 5 по ГОСТ 15150.

После транспортирования проект должно быть выдержано не менее 2 часов в транспортной таре при температуре $20\pm 5^{\circ}\text{C}$ и относительной влажности воздуха не более 80%.

Распакованное проект должно храниться в отапливаемом и вентилируемом чистом помещении при температуре от $+5$ до $+40^{\circ}\text{C}$ и относительной влажности воздуха не более 60 %. При температуре ниже 25°C допускается увеличение относительной влажности до 80 %. Воздух в помещении не должен содержать примесей, вызывающих коррозию металлов, налеты на поверхностях оптических деталей.

						Лист
						10
Изм.	Лист	№ докум.	Подпись	Дата		

ПРИЛОЖЕНИЕ А

(обязательное)

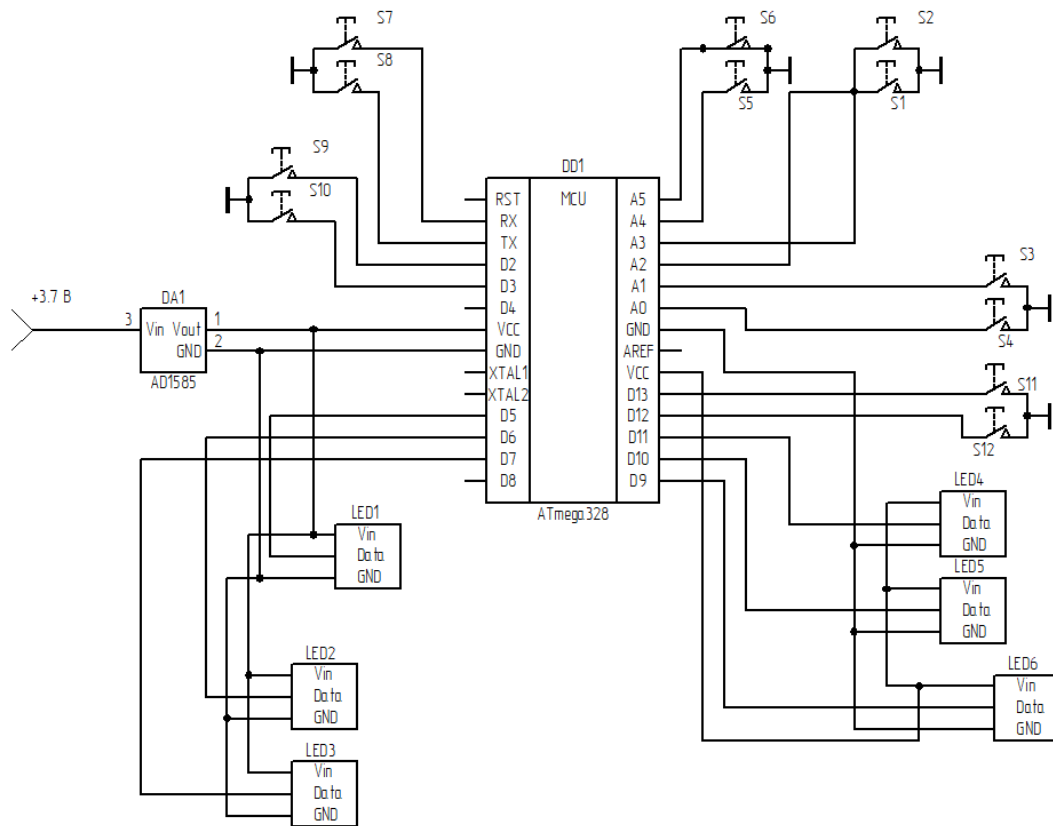


Рисунок А1 – Принципиальная электрическая схема проекта

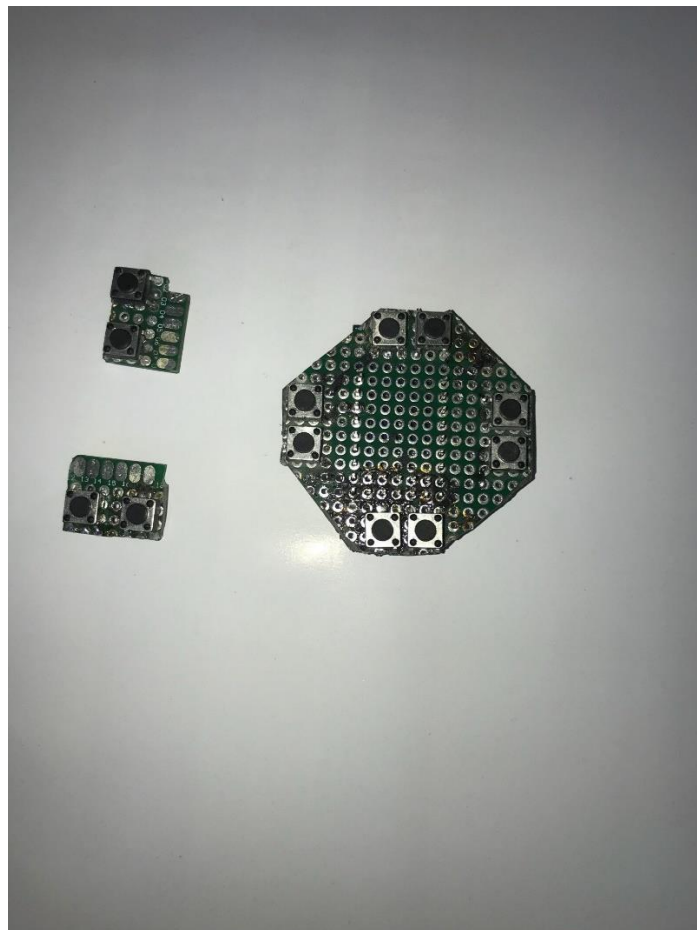


Рисунок А2 – Расстановка элементов на макетную плату

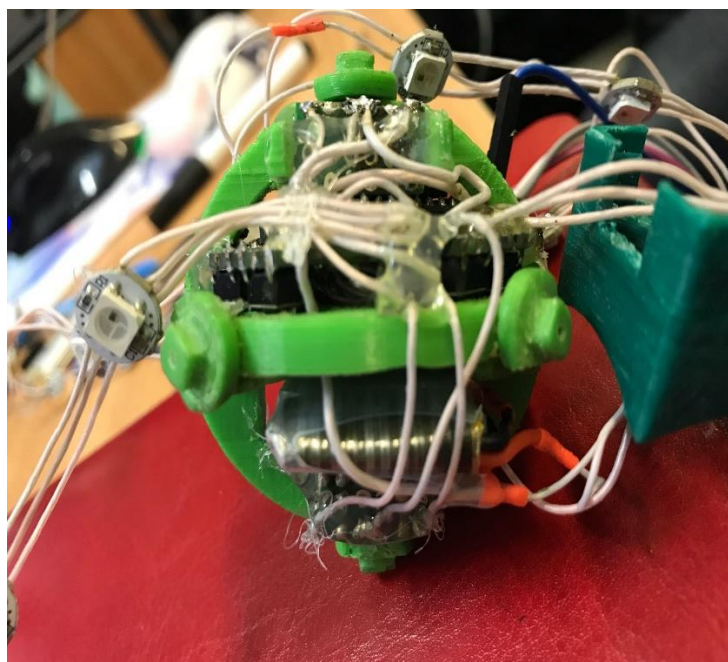


Рисунок А3 – Ось “Cube King’a”



Рисунок А4 – Внешний вид “Cube King’a”

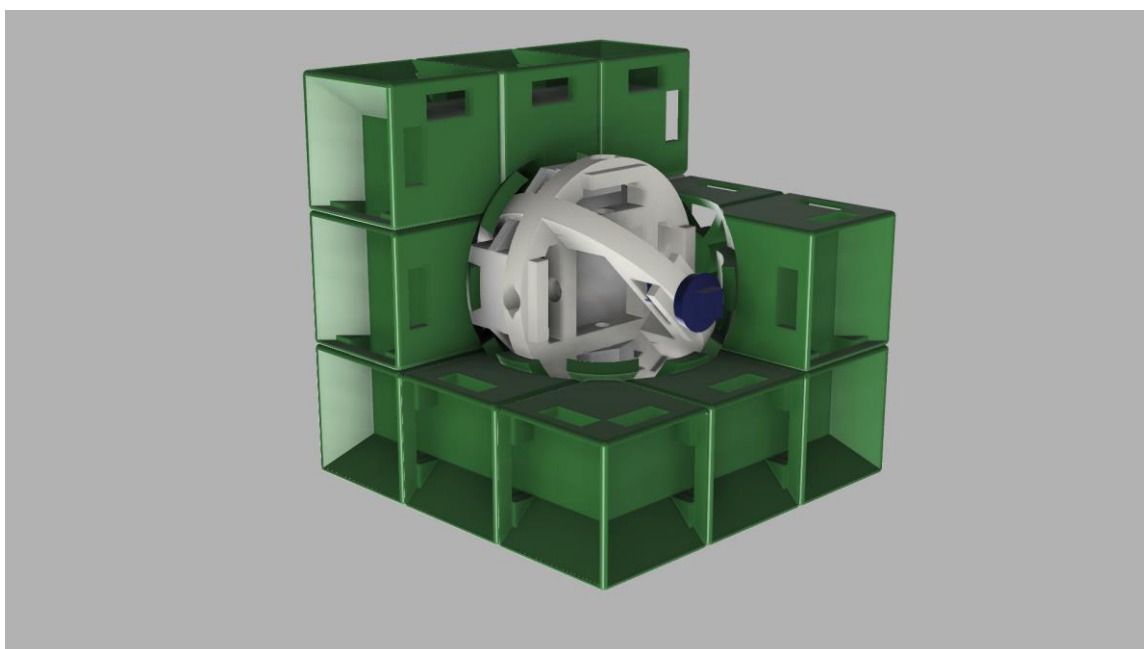


Рисунок А5 – 3D модель Cube King’a

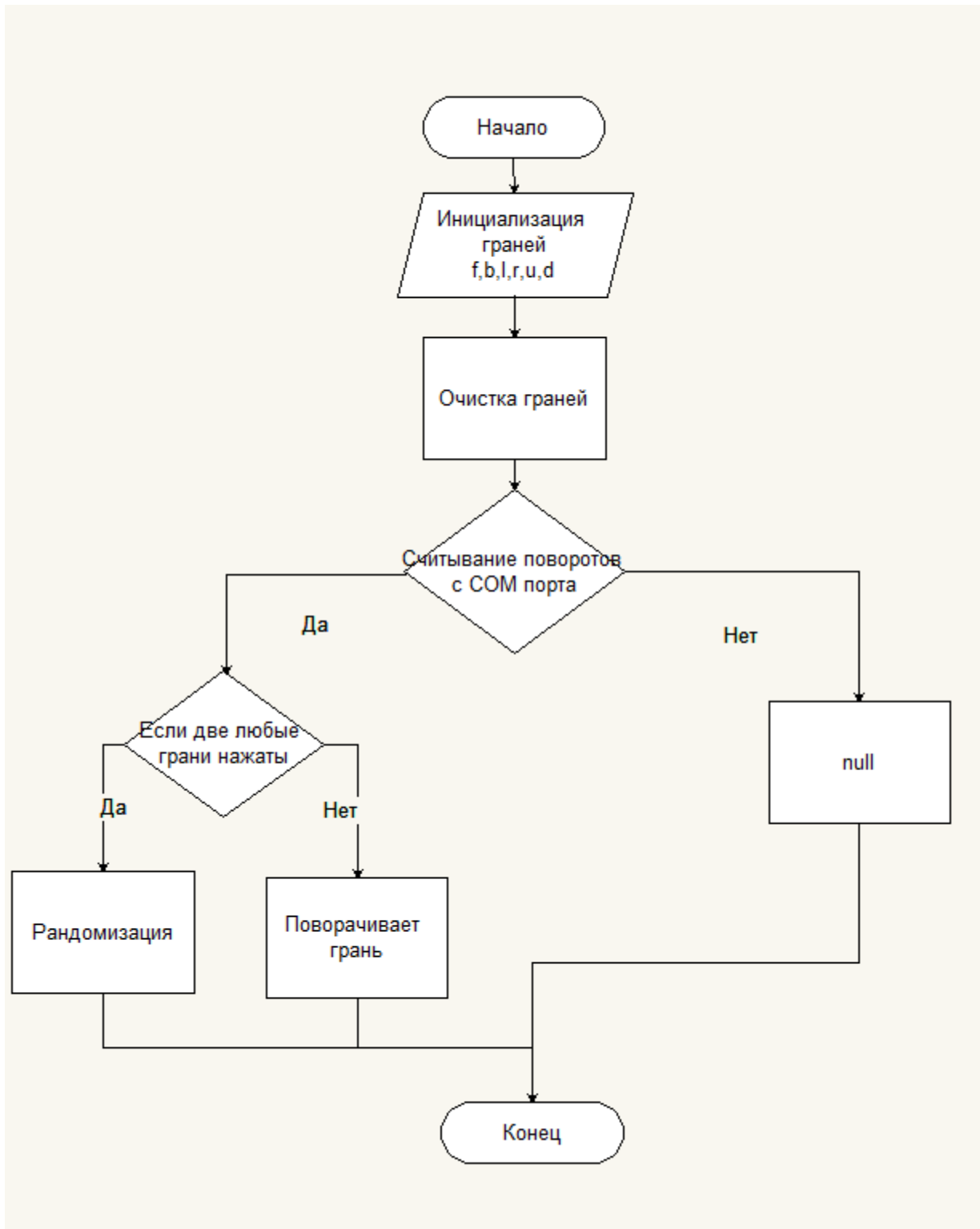


Рисунок А6 – Блок схема программы

Листинг управляющей программы проекта

```
#include <Adafruit_NeoPixel.h> //библиотека RGB светодиодов
#include <RBD_Timer.h> //Прерывания
#include <RBD_Button.h> //Кнопки

#define admin_mode_on
int randomizetime = 3000;
int temp;
int values[6][3][3];
int values_saved[6][3][3];
int x[9]={1,2,2,1,0,0,0,1,2};
int y[9]={1,1,2,2,2,1,0,0,0};
double br = 0.2;

Adafruit_NeoPixel f = Adafruit_NeoPixel(9, 2, NEO_RGB +
NEO_KHZ800); //Обозначение стрипов и их пинов
Adafruit_NeoPixel b = Adafruit_NeoPixel(9, 3, NEO_RGB + NEO_KHZ800);
Adafruit_NeoPixel l = Adafruit_NeoPixel(9, 4, NEO_RGB + NEO_KHZ800);
Adafruit_NeoPixel r = Adafruit_NeoPixel(9, 5, NEO_RGB + NEO_KHZ800);
Adafruit_NeoPixel u = Adafruit_NeoPixel(9, 6, NEO_RGB + NEO_KHZ800);
Adafruit_NeoPixel d = Adafruit_NeoPixel(9, 7, NEO_RGB + NEO_KHZ800);

uint32_t color[6]={f.Color(255*br, 255*br, 255*br), //Создание массива цветов
b.Color(255*br, 0*br, 0*br),
l.Color(0*br, 255*br, 0*br),
r.Color(0*br, 0*br, 255*br),
u.Color(255*br, 0*br, 255*br),
d.Color(255*br, 255*br, 0*br)};

RBD::Button keypad[] = {RBD::Button(8), RBD::Button(9), //Создание массива
кнопок
RBD::Button(10), RBD::Button(11),
RBD::Button(12), RBD::Button(15),
RBD::Button(16), RBD::Button(17),
RBD::Button(18), RBD::Button(19),
RBD::Button(20), RBD::Button(21)};

Adafruit_NeoPixel edges[6]={f,b,l,r,u,d}; //Расположение цветовых граней

// КУБ
```

```

void Clear()
{
for (int i = 0; i < 3; i++)
for (int j = 0; j < 3; j++)
for (int k = 0; k < 6; k++)
{
values[k][i][j] = k;
values_saved[k][i][j] = k;
}
}

```

```

void Swap(int k1, int i1, int j1, int k2, int i2, int j2)
{
temp = values[k1][i1][j1];
values[k1][i1][j1] = values[k2][i2][j2];
values[k2][i2][j2] = temp;
}

```

```

void Swap(int k1, int i1, int j1, int k2, int i2, int j2, int k3, int i3, int j3, int k4, int i4,
int j4)
{
temp = values[k1][i1][j1];
values[k1][i1][j1] = values[k2][i2][j2];
values[k2][i2][j2] = values[k3][i3][j3];
values[k3][i3][j3] = values[k4][i4][j4];
values[k4][i4][j4] = temp;
}

```

```

void RotateFace(int e, int count_spins)//Логика вращения кубика
{
if (e < 0 && e >= 6) return;
switch (count_spins % 4)
{
case 1:
Swap(e, 0, 1, e, 1, 2, e, 2, 1, e, 1, 0);
Swap(e, 0, 0, e, 0, 2, e, 2, 2, e, 2, 0);
break;

case 2:
Swap(e, 0, 0, e, 2, 2);
Swap(e, 1, 0, e, 1, 2);
Swap(e, 0, 1, e, 2, 1);
Swap(e, 0, 2, e, 2, 0);
break;
}
}

```



```

case 3:
Swap(e, 1, 0, e, 2, 1, e, 1, 2, e, 0, 1);
Swap(e, 0, 0, e, 2, 0, e, 2, 2, e, 0, 2);
break;
}
}

void RotateAroundAxisF(int count_spins)//Логика вращения кубика
{
RotateFace(0, count_spins);
RotateFace(5, count_spins);

switch (count_spins % 4)
{
case 1:
for (int i = 0; i < 3; i++)
for (int j = 0; j < 3; j++)
Swap(1, i, j, 2, i, j, 3, i, j, 4, i, j);
break;

case 2:
for (int i = 0; i < 3; i++)
for (int j = 0; j < 3; j++)
{
Swap(1, i, j, 3, i, j);
Swap(2, i, j, 4, i, j);
}
break;

case 3:
for (int i = 0; i < 3; i++)
for (int j = 0; j < 3; j++)
Swap(4, i, j, 3, i, j, 2, i, j, 1, i, j);
break;
}
}

void MoveLeftToCenter()//Логика вращения кубика
{
RotateFace(1, 1);
RotateFace(3, 3);

// TOP -> LEFT -> БОТТОМ -> RIGHT -> TOP
Swap(0, 0, 2, 4, 2, 2, 5, 2, 2, 2, 0, 0);
Swap(0, 0, 1, 4, 1, 2, 5, 2, 1, 2, 1, 0);

```

```
Swap(0, 0, 0, 4, 0, 2, 5, 2, 0, 2, 2, 0);
```

```
Swap(0, 1, 2, 4, 2, 1, 5, 1, 2, 2, 0, 1);
```

```
Swap(0, 1, 1, 4, 1, 1, 5, 1, 1, 2, 1, 1);
```

```
Swap(0, 1, 0, 4, 0, 1, 5, 1, 0, 2, 2, 1);
```

```
Swap(0, 2, 2, 4, 2, 0, 5, 0, 2, 2, 0, 2);
```

```
Swap(0, 2, 1, 4, 1, 0, 5, 0, 1, 2, 1, 2);
```

```
Swap(0, 2, 0, 4, 0, 0, 5, 0, 0, 2, 2, 2);
```

```
}
```

```
void MoveRightToCenter()//Логика вращения кубика
```

```
{
```

```
RotateFace(1, 3);
```

```
RotateFace(3, 1);
```

```
// TOP -> RIGHT -> BOTTOM -> LEFT -> TOP
```

```
Swap(0, 0, 2, 2, 0, 0, 5, 2, 2, 4, 2, 2);
```

```
Swap(0, 0, 1, 2, 1, 0, 5, 2, 1, 4, 1, 2);
```

```
Swap(0, 0, 0, 2, 2, 0, 5, 2, 0, 4, 0, 2);
```

```
Swap(0, 1, 2, 2, 0, 1, 5, 1, 2, 4, 2, 1);
```

```
Swap(0, 1, 1, 2, 1, 1, 5, 1, 1, 4, 1, 1);
```

```
Swap(0, 1, 0, 2, 2, 1, 5, 1, 0, 4, 0, 1);
```

```
Swap(0, 2, 2, 2, 0, 2, 5, 0, 2, 4, 2, 0);
```

```
Swap(0, 2, 1, 2, 1, 2, 5, 0, 1, 4, 1, 0);
```

```
Swap(0, 2, 0, 2, 2, 2, 5, 0, 0, 4, 0, 0);
```

```
}
```

```
void MoveUpToCenter()
```

```
{
```

```
RotateAroundAxisF(1);
```

```
MoveRightToCenter();
```

```
RotateAroundAxisF(3);
```

```
}
```

```
void MoveDownToCenter()
```

```
{
```

```
RotateAroundAxisF(1);
```

```
MoveLeftToCenter();
```

```
RotateAroundAxisF(3);
```

```
}
```

```
void Front(int spin_count)
```

										Лист
Изм.	Лист	№ докум.	Подпись	Дата						18

```

{
#ifdef admin_mode_on
Serial.print("FRONT ");
Serial.println(spin_count);
#endif

// Создание частиц
RotateFace(0, spin_count);
switch (spin_count % 4)
{
case 1:
for (int k = 0; k < 3; k++)
Swap(1, k, 0, 2, k, 0, 3, k, 0, 4, k, 0);
break;

case 2:
for (int k = 0; k < 3; k++)
{
Swap(1, k, 0, 3, k, 0);
Swap(2, k, 0, 4, k, 0);
}
break;

case 3:
for (int k = 0; k < 3; k++)
Swap(4, k, 0, 3, k, 0, 2, k, 0, 1, k, 0);
break;
}
}

void U(int spin_count)
{
#ifdef admin_mode_on
Serial.print("UP "); //Вывод в COM порт текущей команды поворота
Serial.println(spin_count);
#endif

RotateAroundAxisF(1);
MoveRightToCenter();
Front(spin_count);
MoveLeftToCenter();
RotateAroundAxisF(3);
}

void Down(int spin_count)

```

```

{
#ifdef admin_mode_on
Serial.print("DOWN ");//Вывод в COM порт текущей команды поворота
Serial.println(spin_count);
#endif

RotateAroundAxisF(3);
MoveRightToCenter();
Front(spin_count);
MoveLeftToCenter();
RotateAroundAxisF(1);
}

void L(int spin_count)
{
#ifdef admin_mode_on
Serial.print("LEFT ");//Вывод в COM порт текущей команды поворота
Serial.println(spin_count);
#endif

MoveLeftToCenter();
Front(spin_count);
MoveRightToCenter();
}

void R(int spin_count)
{
#ifdef admin_mode_on
Serial.print("RIGHT ");//Вывод в COM порт текущей команды поворота
Serial.println(spin_count);
#endif

MoveRightToCenter();
Front(spin_count);
MoveLeftToCenter();
}

void Back(int spin_count)
{
#ifdef admin_mode_on
Serial.print("BACK ");//Вывод в COM порт текущей команды поворота
Serial.println(spin_count);
#endif

MoveRightToCenter();

```

						Лист
						20
Изм.	Лист	№ докум.	Подпись	Дата		

```

MoveRightToCenter();
Front(spin_count);
MoveRightToCenter();
MoveRightToCenter();
}

void SaveValues()
{
for (int i = 0; i < 3; i++)
for (int j = 0; j < 3; j++)
for (int k = 0; k < 6; k++)
values_saved[k][i][j] = values[k][i][j];
}

void LoadValues()
{
for (int i = 0; i < 3; i++)
for (int j = 0; j < 3; j++)
for (int k = 0; k < 6; k++)
values[k][i][j] = values_saved[k][i][j];
}

// HE КУБ
void Draw()
{
// 1
/* for (int k = 0; k < 6; k++)
{
for (int i = 0; i < 3; i++)
{
for (int j = 0; j < 3; j++)
{
Serial.print(values[k][i][j]);
Serial.print(" ");
}
Serial.println();
}
Serial.println();
}
*/

// 2
// k == 0
for (int i = 0; i < 3; i++)

```

```

{
Serial.print(" ");
for (int j = 0; j < 3; j++)
{
Serial.print(values[0][j][i]);
Serial.print(" ");
}
Serial.println();
}
Serial.println();

```

```

// 1 - 4
for (int i = 0; i < 3; i++)
{
Serial.print(" ");
for (int k = 1; k < 5; k++)
{
for (int j = 0; j < 3; j++)
{
Serial.print(values[k][j][i]);
Serial.print(" ");
}
Serial.print(" ");
}
Serial.println();
}
Serial.println();

```

```

// k == 5
for (int i = 0; i < 3; i++)
{
Serial.print(" ");
for (int j = 0; j < 3; j++)
{
Serial.print(values[5][j][i]);
Serial.print(" ");
}
Serial.println();
}
Serial.println("\n\n\n");
}

```

```

void refresh() { // обновление цветового кода на COM порте

```

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		22

```

for (int k = 0; k < 6; k++)
{
for (int i = 0; i < 9; i++)
{
edges[k].setPixelColor(i, color[values[k][y[i]][x[i]]]);
}
edges[k].show();
}
}

```

```

void randomize() { // функция попарного рандома граней
for (int i = 0; i < 50; i++)
{
switch (random(6))
{
case 0: Front(random(1,4)); break;
case 1: Back(random(1,4)); break;
case 2: L(random(1,4)); break;
case 3: R(random(1,4)); break;
case 4: U(random(1,4)); break;
case 5: Down(random(1,4)); break;

}
}
refresh();
}

```

```

void setup() {
for (int i = 0; i < 6; i++)
{
edges[i].begin();
edges[i].show();
}
randomSeed(analogRead(A0));
Serial.begin(9600);
Clear();
refresh();
}

```

```

void loop() { //Функция вывода в ком порт команды в зависимости от нажатой
кнопки

```

```

// put your main code here, to run repeatedly:
#ifdef admin_mode_on

```

										Лист
										23
Изм.	Лист	№ докум.	Подпись	Дата						

```

Draw();
#endif
while (Serial.available() > 0)
{
char ch = Serial.read();
switch (ch)
{
case 'F': Front(1); break;
case 'B': Back(1); break;
case 'L': L(1); break;
case 'R': R(1); break;
case 'U': U(1); break;
case 'D': Down(1); break;

case 'f': Front(3); break;
case 'b': Back(3); break;
case 'l': L(3); break;
case 'r': R(3); break;
case 'u': U(3); break;
case 'd': Down(3); break;
case 'z': randomize(); break;

case 'C': case 'c': Clear(); break;
refresh();
Draw();
}
}
if (keypad[0].onPressed()) {Front(1); refresh();}
if (keypad[1].onPressed()) {Front(3); refresh();}
if (keypad[4].onPressed()) {Back(1); refresh();}
if (keypad[5].onPressed()) {Back(3); refresh();}
if (keypad[6].onPressed()) {L(1); refresh();}
if (keypad[7].onPressed()) {L(3); refresh();}
if (keypad[2].onPressed()) {R(1); refresh();}
if (keypad[3].onPressed()) {R(3); refresh();}
// if (keypad[10].onPressed()) {U(1); refresh();}
// if (keypad[11].onPressed()) {U(3); refresh();}
if (keypad[8].onPressed()) {Down(1); refresh();}
if (keypad[9].onPressed()) {Down(3); refresh();}

// if
((timer[0].clicked()&&timer[1].clicked())||(timer[2].clicked()&&timer[3].clicked())||
(timer[4].clicked()&&timer[5].clicked())) {

```



```
//  
if(((timer[0].clickTime())>=randomizetime)&&(timer[1].clickTime())>=randomizeti  
me))||((timer[2].clickTime())>=randomizetime)&&(timer[3].clickTime())>=randomiz  
etime))||((timer[4].clickTime())>=randomizetime)&&(timer[5].clickTime())>=rando  
mizetime)))  
// { randomize(); refresh(); }
```

									Лист
									25
Изм.	Лист	№ докум.	Подпись	Дата					