ПМГ „Академик Боян Петканчин“ – Хасково

НП „Обучение за ИТ кариера“

Документация

Модул 8: Въведение в операционни системи и вградени системи

Тема: Geometric Shapes

Изготвила:

Ферай Фахри

<https://github.com/feray03>

**Хасково 2021г.**

Съдържание

[Какво представлява проекта? 3](#_Toc69251697)

[Блок схема 3](#_Toc69251698)

[Електрическа схема 3](#_Toc69251699)

[Списък от съставни части 4](#_Toc69251700)

[Сорс код 4](#_Toc69251701)

[Заключение 11](#_Toc69251702)

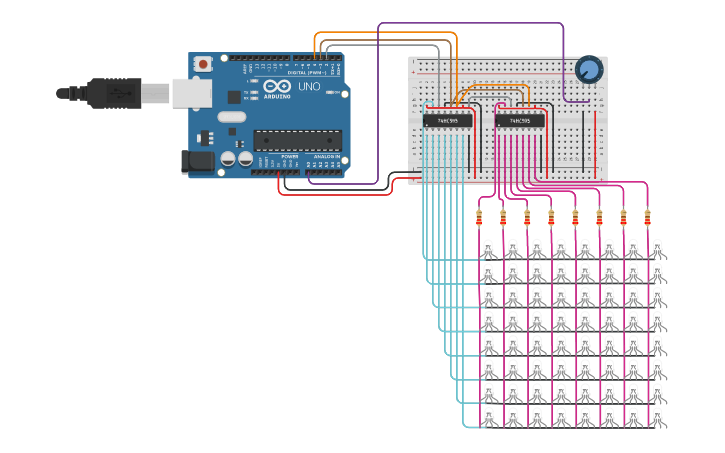


# ftestickers square frame abstract Sticker by PennyannКакво представлява проекта?

Проекта представлява матрица от RGB LED лампички, в която се очертават различни фигурки.

# Блок схема

# Електрическа схема



# Списък от съставни части

1. Arduino Uno R3 - 1
2. LED RGB – 64
3. 250 kΩ Potentiometer – 1
4. 8-Bit Shift Register - 2
5. 220 Ω Resistor – 8

# Сорс код

const int data = 2;

const int store = 3;

const int shift = 4;

int potiValue;

// column counter

int j = 0;

// showDuration counter

int k;

int row[8] = {127, 191, 223, 239, 247, 251, 253, 254};

// Heart

int columnH[8] = {102, 153, 129, 129, 129, 66, 36, 24};

// Heart1

int columnH1[8] = {102, 255, 219, 195, 195, 102, 60, 24};

// Heart2

int columnH2[8] = {102, 255, 255, 255, 255, 126, 60, 24};

// Triangle

int columnT[8] = {0, 0, 24, 36, 66, 255, 0, 0};

// Square

int columnS[8] = {0, 126, 66, 66, 66, 66, 126, 0};

// Circle

int columnC[8] = {60, 66, 129, 129, 129, 129, 66, 60};

// Rectangle

int columnR[8] = {126, 66, 66, 66, 66, 66, 66, 126};

// Moon

int columnM[8] = {28, 56, 112, 112, 112, 112, 56, 28};

void setup()

{

Serial.begin(9600);

// 74HC595

pinMode(data, OUTPUT); // data

pinMode(store, OUTPUT); // store

pinMode(shift, OUTPUT); // shift

}

void poti()

{

potiValue = analogRead(0);

}

void heart()

{

for(k = 0; k < 15; k++)

{

for(int i = 0; i < 8; i++)

{

digitalWrite(store, LOW);

shiftOut(data, shift, LSBFIRST, columnH[j]);

shiftOut(data, shift, LSBFIRST, row[i]);

digitalWrite(store, HIGH);

j++;

poti();

delay(potiValue);

}

j = 0;

}

}

void heart1()

{

for(k = 0; k < 15; k++)

{

for(int i = 0; i < 8; i++)

{

digitalWrite(store, LOW);

shiftOut(data, shift, LSBFIRST, columnH1[j]);

shiftOut(data, shift, LSBFIRST, row[i]);

digitalWrite(store, HIGH);

j++;

poti();

delay(potiValue);

}

j = 0;

}

}

void heart2()

{

for(k = 0; k < 70; k++)

{

for(int i = 0; i < 8; i++)

{

digitalWrite(store, LOW);

shiftOut(data, shift, LSBFIRST, columnH2[j]);

shiftOut(data, shift, LSBFIRST, row[i]);

digitalWrite(store, HIGH);

j++;

poti();

delay(potiValue);

}

j = 0;

}

}

void triangle()

{

for(k = 0; k < 70; k++)

{

for(int i = 0; i < 8; i++)

{

digitalWrite(store, LOW);

shiftOut(data, shift, LSBFIRST, columnT[j]);

shiftOut(data, shift, LSBFIRST, row[i]);

digitalWrite(store, HIGH);

j++;

poti();

delay(potiValue);

}

j = 0;

}

}

void square()

{

for(k = 0; k < 70; k++)

{

for(int i = 0; i < 8; i++)

{

digitalWrite(store, LOW);

shiftOut(data, shift, LSBFIRST, columnS[j]);

shiftOut(data, shift, LSBFIRST, row[i]);

digitalWrite(store, HIGH);

j++;

poti();

delay(potiValue);

}

j = 0;

}

}

void circle()

{

for(k = 0; k < 70; k++)

{

for(int i = 0; i < 8; i++)

{

digitalWrite(store, LOW);

shiftOut(data, shift, LSBFIRST, columnC[j]);

shiftOut(data, shift, LSBFIRST, row[i]);

digitalWrite(store, HIGH);

j++;

poti();

delay(potiValue);

}

j = 0;

}

}

void rectangle()

{

for(k = 0; k < 70; k++)

{

for(int i = 0; i < 8; i++)

{

digitalWrite(store, LOW);

shiftOut(data, shift, LSBFIRST, columnR[j]);

shiftOut(data, shift, LSBFIRST, row[i]);

digitalWrite(store, HIGH);

j++;

poti();

delay(potiValue);

}

j = 0;

}

}

void moon()

{

for(k = 0; k < 70; k++)

{

for(int i = 0; i < 8; i++)

{

digitalWrite(store, LOW);

shiftOut(data, shift, LSBFIRST, columnM[j]);

shiftOut(data, shift, LSBFIRST, row[i]);

digitalWrite(store, HIGH);

j++;

poti();

delay(potiValue);

}

j = 0;

}

}

void loop()

{

heart();

heart1();

heart2();

triangle();

square();

circle();

rectangle();

moon();

heart();

}

# Заключение