pascal holthuijsen

2139622

IPR -2.3

Pascal Holthuijsen

Inhoud

[Beschrijving 3](#_Toc36645087)

[Animaties: 3](#_Toc36645088)

[Letters: 3](#_Toc36645089)

[Git Link: 4](#_Toc36645090)

[Zelf geschreven code 5](#_Toc36645091)

# Beschrijving

Voor mijn opdracht heb ik een led matrix aan gestuurd die ik zelf heb gemaakt. Omdat ik deze matrix zelf heb gemaakt werkte hij niet met matrix libraries en moest ik dus alles zelf maken. Ik heb wel gebruik kunnen maken van de Arduino library: FastLED. Dit betekende dat ik het meeste om moest schrijven van led strip naar matrix. Wat ik gedaan heb is ik heb een paar animaties geschreven voor op het scherm en verder heb ik gezorgd dat er letters gedisplayed konden worden en dat er 1 letter tegelijkertijd over het scherm kan schuiven

## Animaties:

Als animaties heb ik gekozen voor een random kleuren pallet en twee rijen die in elkaar schuiven

## Letters:

Zoals eerder vermeld heb ik gezorgd dat er letters laten zien konden worden. Tot nu toe heb ik de letters A t/m L geschreven omdat ik het niet nodig vond om alles er bij te schrijven voor het testen.



# Git Link:

<https://github.com/phunderr/Individueel>

# Zelf geschreven code

De onderstaande code is de code die ik zelf heb geschreven.

#include <FastLED.h>

#define NUM\_LEDS 50

#define DATA\_PIN 14

#define CLOCK\_PIN 13

CRGB leds[NUM\_LEDS];

void setup() {

Serial.begin(57600);

Serial.println("resetting");

LEDS.addLeds<WS2812,DATA\_PIN,RGB>(leds,NUM\_LEDS);

LEDS.setBrightness(84);

}

void fadeall() { for(int i = 0; i < NUM\_LEDS; i++) { leds[i].nscale8(250); } }

uint16\_t A[] = {0b0000000111,0b0000000101, 0b0000000111, 0b0000000101, 0b0000000101 };

uint16\_t B[] = {0b0000000111,0b0000000101, 0b0000000011, 0b0000000101, 0b0000000111 };

uint16\_t C[] = {0b0000000111,0b0000000001, 0b0000000001, 0b0000000001, 0b0000000111 };

uint16\_t D[] = {0b0000000011,0b0000000101, 0b0000000101, 0b0000000101, 0b0000000011 };

uint16\_t E[] = {0b0000000111,0b0000000001, 0b0000000111, 0b0000000001, 0b0000000111 };

uint16\_t F[] = {0b0000000111,0b0000000001, 0b0000000111, 0b0000000101, 0b0000000001 };

uint16\_t G[] = {0b0000000111,0b0000000001, 0b0000000101, 0b0000000101, 0b0000000111 };

uint16\_t H[] = {0b0000000101,0b0000000101, 0b0000000111, 0b0000000101, 0b0000000101 };

uint16\_t I[] = {0b0000000111,0b0000000010, 0b0000000010, 0b0000000010, 0b0000000111 };

uint16\_t J[] = {0b0000000011,0b0000000001, 0b0000000001, 0b0000000101, 0b0000000111 };

uint16\_t K[] = {0b0000000101,0b0000000101, 0b0000000011, 0b0000000101, 0b0000000101 };

uint16\_t L[] = {0b0000000001,0b0000000001, 0b0000000001, 0b0000000001, 0b0000000111 };

void writeLetter(uint16\_t letter[] )

{

for(int i = 0; i < 5; i++)

{

for(int j = 0; j < 10; j++)

{

if(i % 2 == 0)

{

if(letter[i] & (1 << j))

{

leds[i \* 10 + j] = CHSV(0, 255, 255);

}

else

{

leds[i \* 10 + j] = CRGB::Black;

}

}

else

{

if(letter[i] & (1 << j))

{

leds[i \* 10 + (9 - j)] = CHSV(0, 255, 255);

}

else

{

leds[i \* 10 + (9 - j)] = CRGB::Black;

}

}

}

}

FastLED.show();

}

void slideLetter(uint16\_t letter[])

{

for(int k = 0; k < 10; k++)

{

for(int i = 0; i < 5; i++)

{

for(int j = 0; j < 10; j++)

{

if(i % 2 == 0)

{

if(letter[i] & (1 << j))

{

leds[i \* 10 + ((j + k) % 10)] = CHSV(0, 255, 255);

}

else

{

leds[i \* 10 + ((j + k) % 10)] = CRGB::Black;

}

}

else

{

if(letter[i] & (1 << j))

{

leds[i \* 10 + (9 - ((j + k) % 10))] = CHSV(0, 255, 255);

}

else

{

leds[i \* 10 + (9 - ((j + k) % 10))] = CRGB::Black;

}

}

}

}

FastLED.show();

delay(750);

}

}

void colourswapsnake()

{

static uint8\_t hue = 0;

Serial.print("x");

for(int i = 0; i < NUM\_LEDS; i++)

{

leds[i] = CHSV(hue++, 255, 255);

FastLED.show();

fadeall();

delay(10);

}

Serial.print("x");

for(int i = (NUM\_LEDS)-1; i >= 0; i--) {

leds[i] = CHSV(hue++, 255, 255);

FastLED.show();

fadeall();

delay(10);

}

}

void colourswap()

{

static uint8\_t hue = 0;

Serial.print("x");

for(int i = 0; i < NUM\_LEDS; i++) {

leds[i] = CHSV(hue, 255, 255);

FastLED.show();

fadeall();

}

hue += 10;//change the colour every loop to a different colour

delay(500);

}

void instantrandom()

{

static uint8\_t hue = 0;

Serial.print("x");

for(int i = 0; i < NUM\_LEDS; i++) {

hue = random(255);//set every pixel the colour to a different value

leds[i] = CHSV(hue, 255, 255);

FastLED.show();

fadeall();

}

delay(10);

}

void intertwine(uint8\_t colour)

{

for (int i = 0; i < 10; i++)

{

leds[i] = CHSV(colour, 255, 255);

leds[i + 10] = CHSV(colour, 255, 255);

leds[i + 20] = CHSV(colour, 255, 255);

leds[i + 30] = CHSV(colour, 255, 255);

leds[i + 40] = CHSV(colour, 255, 255);

FastLED.show();

fadeall();

delay(100);

}

}

void intertwineclear()

{

for (int i = 0; i < 10; i++)

{

leds[i] = CHSV(0, 0, 0);

leds[i + 10] = CHSV(0, 0, 0);

leds[i + 20] = CHSV(0, 0, 0);

leds[i + 30] = CHSV(0, 0, 0);

leds[i + 40] = CHSV(0, 0, 0);

FastLED.show();

fadeall();

delay(100);

}

}

void interTwineColourChange()

{

int hue = 0;

while(hue < 255)

{

intertwine(hue%255);

hue+=20;

intertwineclear();

}

}

void loop() {

interTwineColourChange();

delay(1000);

instantrandom();

delay(1000);

colourswap();

delay(1000);

colourswapsnake();

writeLetter(H);

delay(750);

writeLetter(A);

delay(750);

writeLetter(I);

delay(1000);

slideLetter(A);

}