#include <TroykaRTC.h>

#include <Wire.h>

#include <FastLED.h>

#define NUM\_LEDS 256

#define DATA\_PIN 4

#define LEN\_TIME 12

#define LEN\_DATE 12

#define LEN\_DOW 12

RTC clock;

char timeF[LEN\_TIME];

char date[LEN\_DATE];

char weekDay[LEN\_DOW];

int hours\_1;

int minuts\_1;

int hours\_2;

int minuts\_2;

CRGB leds[NUM\_LEDS];

unsigned long time;

int coord(int a,int b){

 if(b%2==0){

 int num=(b-1)\*16+a-1;

 return num;

 }

 else {

 int num=b\*16-a;

 return num;

 }

}

void num(int x, int y, int z, float time) {

 leds[coord(x,y)] = CRGB::Black;

 leds[coord(x+1,y)] = CRGB::Black;

 leds[coord(x+2,y)] = CRGB::Black;

 leds[coord(x,y+1)] = CRGB::Black;

 leds[coord(x+1,y+1)] = CRGB::Black;

 leds[coord(x+2,y+1)] = CRGB::Black;

 leds[coord(x,y+2)] = CRGB::Black;

 leds[coord(x+1,y+2)] = CRGB::Black;

 leds[coord(x+2,y+2)] = CRGB::Black;

 leds[coord(x,y+3)] = CRGB::Black;

 leds[coord(x+1,y+3)] = CRGB::Black;

 leds[coord(x+2,y+3)] = CRGB::Black;

 leds[coord(x,y+4)] = CRGB::Black;

 leds[coord(x+1,y+4)] = CRGB::Black;

 leds[coord(x+2,y+4)] = CRGB::Black;

 if (z==0) {

 leds[coord(x,y)] = CHSV(time, 187, 75);

 leds[coord(x,y+1)] = CHSV(time, 187, 75);

 leds[coord(x,y+2)] = CHSV(time, 187, 75);

 leds[coord(x,y+3)] = CHSV(time, 187, 75);

 leds[coord(x,y+4)] = CHSV(time, 187, 75);

 leds[coord(x+2,y)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+1)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+2)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+3)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+4)] = CHSV(time, 187, 75);

 leds[coord(x+1,y)] = CHSV(time, 187, 75);

 leds[coord(x+1,y+4)] = CHSV(time, 187, 75);

 }

 if (z==1) {

 leds[coord(x+1,y)] = CHSV(time, 187, 75);

 leds[coord(x+1,y+1)] = CHSV(time, 187, 75);

 leds[coord(x+1,y+2)] = CHSV(time, 187, 75);

 leds[coord(x+1,y+3)] = CHSV(time, 187, 75);

 leds[coord(x+1,y+4)] = CHSV(time, 187, 75);

 leds[coord(x,y+1)] = CHSV(time, 187, 75);

 leds[coord(x,y+4)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+4)] = CHSV(time, 187, 75);

 }

 if (z==2) {

 leds[coord(x,y)] = CHSV(time, 187, 75);

 leds[coord(x+1,y)] = CHSV(time, 187, 75);

 leds[coord(x+2,y)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+1)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+2)] = CHSV(time, 187, 75);

 leds[coord(x+1,y+2)] = CHSV(time, 187, 75);

 leds[coord(x,y+2)] = CHSV(time, 187, 75);

 leds[coord(x,y+3)] = CHSV(time, 187, 75);

 leds[coord(x,y+4)] = CHSV(time, 187, 75);

 leds[coord(x+1,y+4)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+4)] = CHSV(time, 187, 75);

 }

 if (z==3) {

 leds[coord(x,y)] = CHSV(time, 187, 75);

 leds[coord(x+1,y)] = CHSV(time, 187, 75);

 leds[coord(x+2,y)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+1)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+2)] = CHSV(time, 187, 75);

 leds[coord(x+1,y+2)] = CHSV(time, 187, 75);

 leds[coord(x,y+2)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+3)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+4)] = CHSV(time, 187, 75);

 leds[coord(x+1,y+4)] = CHSV(time, 187, 75);

 leds[coord(x,y+4)] = CHSV(time, 187, 75);

 }

 if (z==4) {

 leds[coord(x,y)] = CHSV(time, 187, 75);

 leds[coord(x,y+1)] = CHSV(time, 187, 75);

 leds[coord(x,y+2)] = CHSV(time, 187, 75);

 leds[coord(x+1,y+2)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+2)] = CHSV(time, 187, 75);

 leds[coord(x+2,y)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+1)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+3)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+4)] = CHSV(time, 187, 75);

 }

 if (z==5){

 leds[coord(x,y)] = CHSV(time, 187, 75);

 leds[coord(x+1,y)] = CHSV(time, 187, 75);

 leds[coord(x+2,y)] = CHSV(time, 187, 75);

 leds[coord(x,y+1)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+2)] = CHSV(time, 187, 75);

 leds[coord(x+1,y+2)] = CHSV(time, 187, 75);

 leds[coord(x,y+2)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+3)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+4)] = CHSV(time, 187, 75);

 leds[coord(x+1,y+4)] = CHSV(time, 187, 75);

 leds[coord(x,y+4)] = CHSV(time, 187, 75);

 }

 if (z==6) {

 leds[coord(x,y)] = CHSV(time, 187, 75);

 leds[coord(x+1,y)] = CHSV(time, 187, 75);

 leds[coord(x+2,y)] = CHSV(time, 187, 75);

 leds[coord(x,y+1)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+2)] = CHSV(time, 187, 75);

 leds[coord(x+1,y+2)] = CHSV(time, 187, 75);

 leds[coord(x,y+2)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+3)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+4)] = CHSV(time, 187, 75);

 leds[coord(x+1,y+4)] = CHSV(time, 187, 75);

 leds[coord(x,y+4)] = CHSV(time, 187, 75);

 leds[coord(x,y+3)] = CHSV(time, 187, 75);

 }

 if (z==7) {

 leds[coord(x,y)] = CHSV(time, 187, 75);

 leds[coord(x+1,y)] = CHSV(time, 187, 75);

 leds[coord(x+2,y)] = CHSV(time, 187, 75);

 leds[coord(x+1,y+2)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+1)] = CHSV(time, 187, 75);

 leds[coord(x,y+3)] = CHSV(time, 187, 75);

 leds[coord(x,y+4)] = CHSV(time, 187, 75);

 }

 if (z==8) {

 leds[coord(x,y)] = CHSV(time, 187, 75);

 leds[coord(x,y+1)] = CHSV(time, 187, 75);

 leds[coord(x,y+2)] = CHSV(time, 187, 75);

 leds[coord(x,y+3)] = CHSV(time, 187, 75);

 leds[coord(x,y+4)] = CHSV(time, 187, 75);

 leds[coord(x+2,y)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+1)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+2)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+3)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+4)] = CHSV(time, 187, 75);

 leds[coord(x+1,y)] = CHSV(time, 187, 75);

 leds[coord(x+1,y+4)] = CHSV(time, 187, 75);

 leds[coord(x+1,y+2)] = CHSV(time, 187, 75);

 }

 if (z==9) {

 leds[coord(x,y)] = CHSV(time, 187, 75);

 leds[coord(x,y+1)] = CHSV(time, 187, 75);

 leds[coord(x,y+2)] = CHSV(time, 187, 75);

 leds[coord(x,y+4)] = CHSV(time, 187, 75);

 leds[coord(x+2,y)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+1)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+2)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+3)] = CHSV(time, 187, 75);

 leds[coord(x+2,y+4)] = CHSV(time, 187, 75);

 leds[coord(x+1,y)] = CHSV(time, 187, 75);

 leds[coord(x+1,y+4)] = CHSV(time, 187, 75);

 leds[coord(x+1,y+2)] = CHSV(time, 187, 75);

 }

}

void setup() {

 Serial.begin(9600);

 clock.begin();

 clock.set(\_\_TIMESTAMP\_\_);

 FastLED.addLeds <NEOPIXEL, DATA\_PIN> (leds, NUM\_LEDS);

}

void loop() {

 clock.read();

 clock.getTimeStamp(timeF, date, weekDay);

 hours\_1 = timeF[0]-'0';

 minuts\_1 = timeF[3]-'0';

 hours\_2 = timeF[1]-'0';

 minuts\_2 = timeF[4]-'0';

 Serial.print(hours\_1);

 Serial.print("\t");

 Serial.print(hours\_2);

 Serial.print("\t");

 Serial.print(minuts\_1);

 Serial.print("\t");

 Serial.print(minuts\_2);

 Serial.print("\t");

 float T = ((hours\_1\*10+hours\_2)\*60+(minuts\_1\*10+minuts\_2))/(1440/255);

 num(2, 2, hours\_1, T);

 num(6, 2, hours\_2, T);

 num(9, 11, minuts\_1, T);

 num(13, 11, minuts\_2, T);

 for (int a = 0; a < 17; a ++) {

 leds[coord(16-a,1+a)]= CHSV(T, 187, 75);

 }

 FastLED.show();

}