CEAS DIGITAL LED MATRIX

Cosma Dan Cristian

Data

12.01.2023

Curs:

Masurari si Traductoare

IA Grupa: 30126

Cuprins

1. Componente folosite in acest proiect………………Pagina 3
2. Domeniul de Aplicatie……………………………………Pagina 3
3. Descrierea solutiei de rezolvare……………………Pagina 4
4. Randarile componentelor fizice ……………………Pagina 5
5. Schema tehnica a circuitului ………………………Pagina 7
6. Codul programului ……………………………………Pagina 7
7. Avantaje fata de solutiile cunoscute………………Pagina 19

Componente folosite in acest proiect:

- WS2812B-64 Only ONE Pin 24 bit color 64 RGB LEDs
- Arduino Nano 3.1 W/atmega328
- RTC DS3231 Precision Circuit
- Rezistor 330 ohmi
- Imprimanta 3D pentru piesele fizice
- Filament plastic PLA
- Cositor
- Fire dupont
- Ciocan de lipit

Domeniul de Aplicatie:

Masurarea timpului in special in folosirea sa uzuala de catre consumatori.

Descrierea solutiei de rezolvare:

Proiectul a fost implementat cu idea de a crea un ceas digital cu un aspect modern, combinand elementele fizice cu cele digitale.

Primul punct important pe care a trebuit sa il prevad in crearea acestui proiect a fost mentinerea unui buget cat mai mic pentru a ne incadra intr-un pret pe care un consumator normal l-ar considera corect din punct de vedere a achizitionarii unui astfel de obiect. Desi ne aflam intr-o criza majora a microprocesoarelor toate elementele digitale s-au putut procura la un pret mic, acest fapt oferindu-mi un buget mai mare pentru crearea si printarea partii fizice, utilizand materiale mai solide cum ar fi filamentul PLA.

Inainte de a trece la lipit si la scris codul proiectului, au fost create cateva randari ale partilor fizice care, mai apoi au fost printate cu ajutorul unei imprimante 3D comerciale, cu un blat de printare mai mare pentru a scurta timpul in care a fost formata carcasa si corpul ceasului,acesta Durand in jur de 6 ore.

In timp ce se printa partea fizica a proiectului, am inceput asamblarea componentelor electronice. Mai intai am dezizolat cateva fire pentru a putea face legatura intre placuta led si microcontrolerul Arduino Nano. Cele trei capete le vom lipi mai intai de placuta led in porturile **V+** (Putere), **IN** (linia de date) si **V-** (impamantarea), iar mai apoi vom lipi rezistenta de 330 de Ohmi liniei IN pentru a o putea proteja in caz de scurtcircuit.

In acest moment putem conecta placuta LED la Arduino nano avand grije sa lipim firele la pinii corecti:

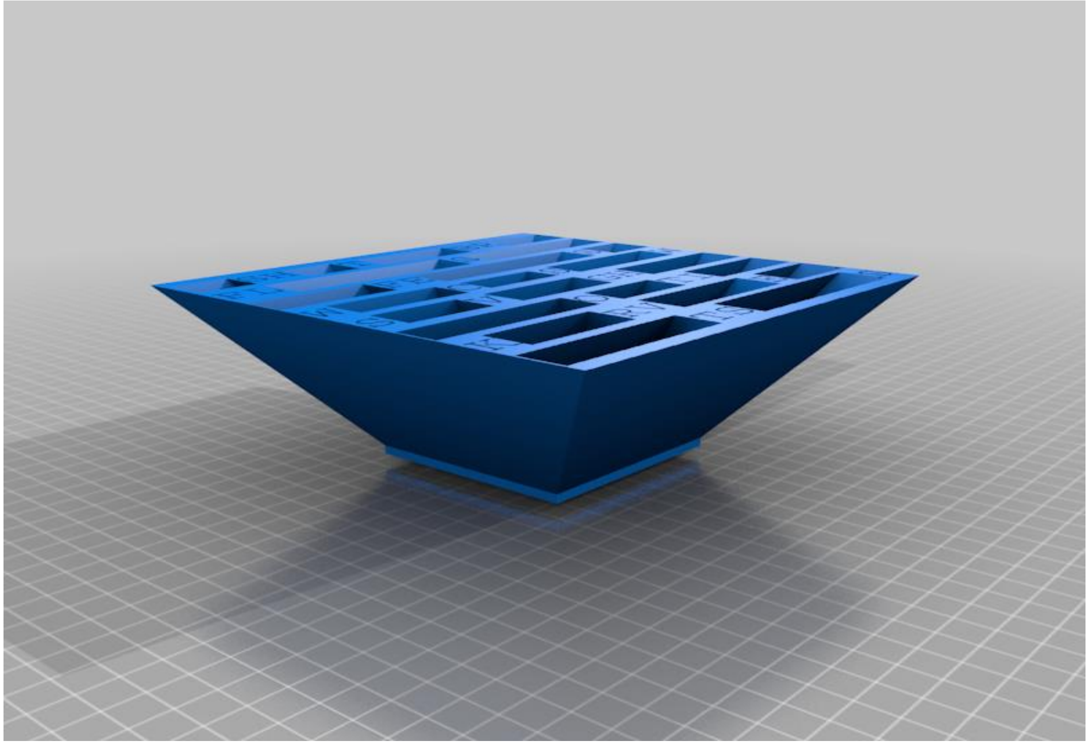
- V+→5V
- V- → GND
- IN → D6

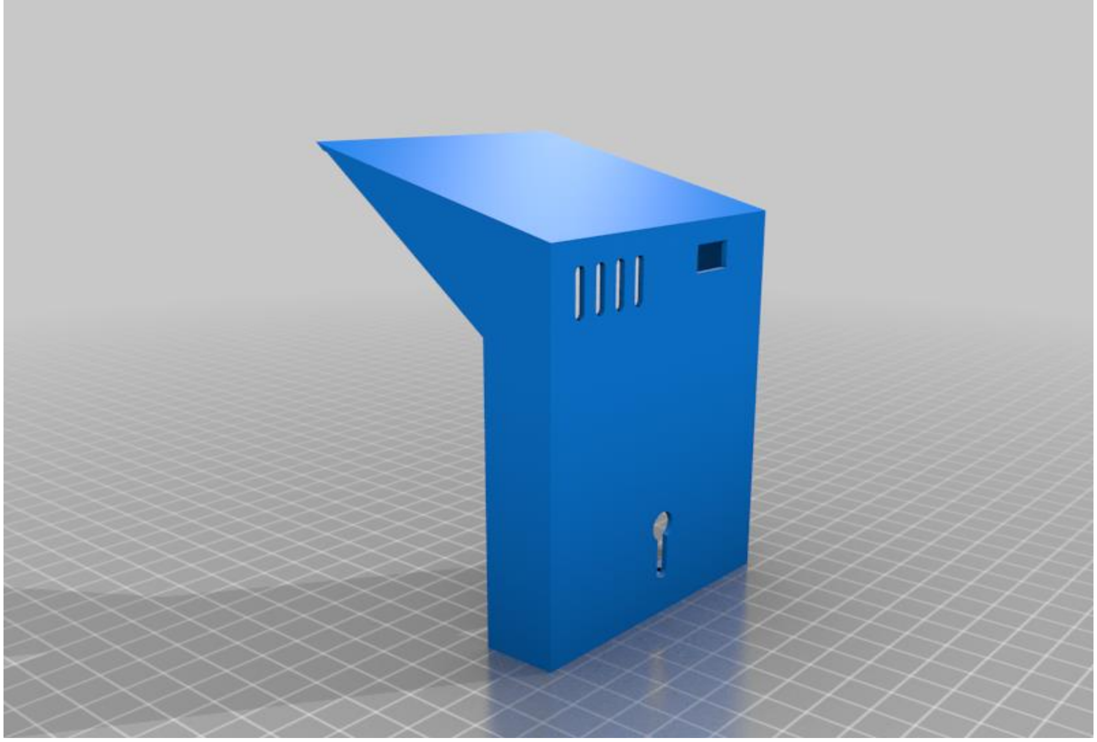
Urmatorul pas a fost conectarea firelor dupont la RTC (Real Time Clock). Aceasta este placa care permite Arduino-ului nostrum sa-si aminteasca ora chiar si atunci cand este deconectat de la current. Pinii RTC-ului utilizati si lipiti la Arduino sunt:

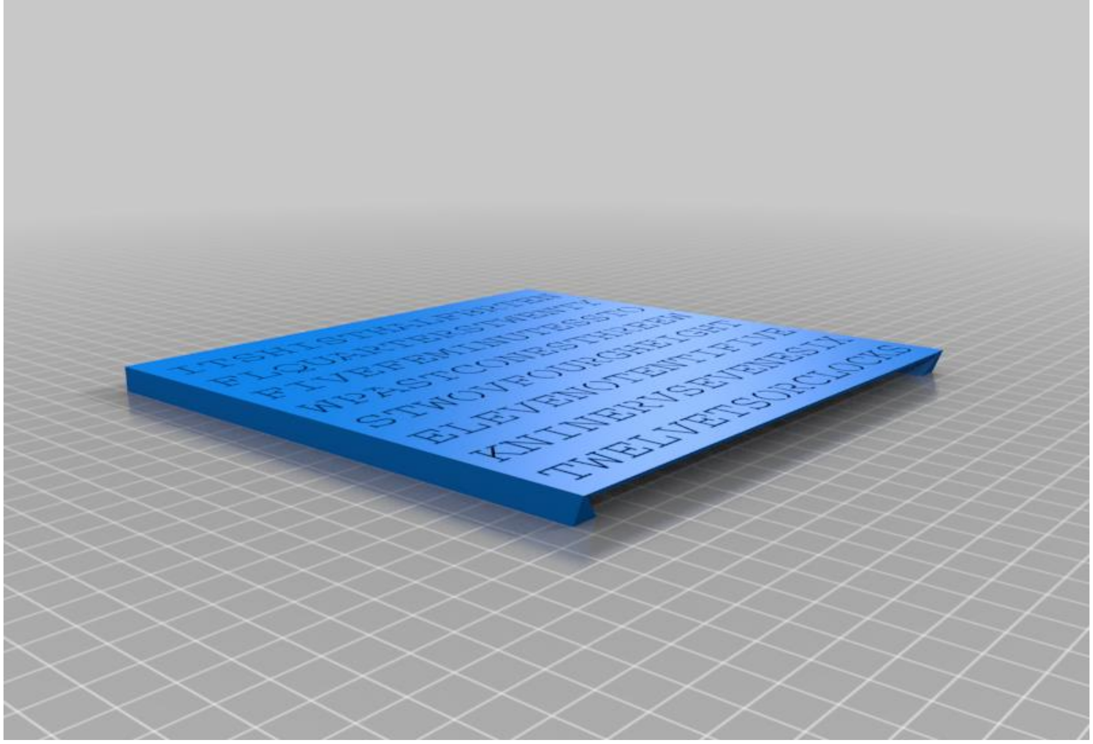
- VCC→5V
- GND→GND(Arduino)
- SDA→A4
- SCL→A5

Ca si ultim pas, introducem codul in programul Arduino IDE care va fi rescris in procesorul Atmega32 al placutei, iar mai apoi unim partea fizica si partea digitala a ceasului pentru a obtine produsul final.

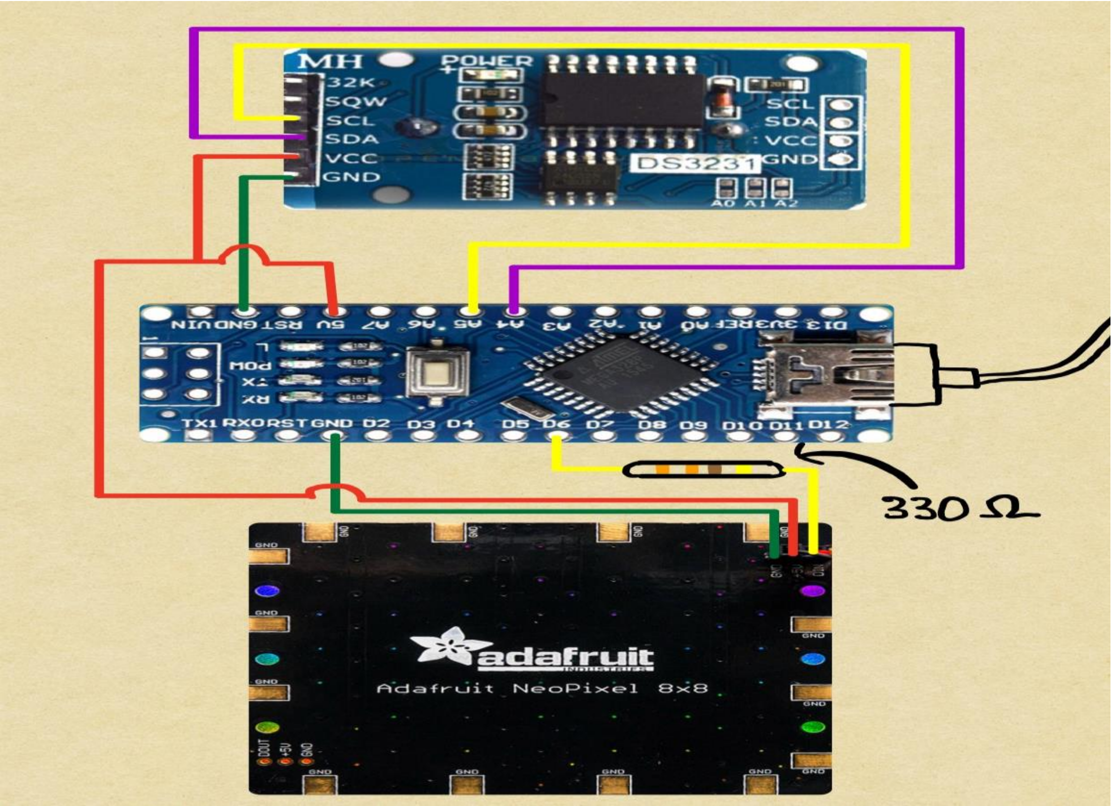
Randarile componentelor fizice:






FLQUARTERSTWENTY
FIVEFFMINUTESSTO
WPASTCONESTHREEN
STWOVFOURGHEIGHT
ELEVENOTENTIFIVE
KNINERVSEVENSIX
TWELVETSORCLOCKS

Schema tehnica a circuitului:


MH
32K
SQW
SCL
SDA
VCC
GND
POWER
DS3231
SCL
SDA
VCC
GND
A0 A1 A2
VIN GND RST 5V A7 A6 A5 A4 A3 A2 A1 A0 REF 3V3 D13
TX1 RX0 RST GND D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12
POW
TX
RX
330 Ω
adafruit
Adafruit NeoPixel 8x8
GND

Codul programului:

```
/*

  grid pattern

   I T S H I S T H A L F B R T E N
   F L Q U A R T E R S T W E N T Y
   F I V E F E M I N U T E S S T O
   W P A S T C O N E S T H R E E W
   S T W O V F O U R G H E I G H T
   E L E V E N O T E N T I F I V E
   K N I N E R V S E V E N E S I X
   T W E L V E T S O R C L O C K S

*/

//Includem librariile necesare functinarii programului:
#include <Wire.h>
```

```
    #include <RTClib.h>
    #include <DST_RTC.h>
    #include <Adafruit_GFX.h>
    #include <Adafruit_NeoMatrix.h>
    #include <Adafruit_NeoPixel.h>

    // Definim pinii de pe Arduino Nano:
    #define NEOPIN 6  // connectam la intrarea DIN pe NeoMatrix 8x8 prin
intermediul resistorului
    #define RTCGND A2 // folosim pinul ca si ground
    #define RTCPWR A3 // folosim pinul ca intrare alimentare

    RTC_DS1307 RTC; // Definim obiectul de ceas
    DST_RTC dst_rtc; // Obiectul DST
    const char rulesDST[] = "EU";
    DateTime theTime; // Tine ora curenta a ceasului


    // Configuram matricea LED 8x8
    Adafruit_NeoMatrix matrix = Adafruit_NeoMatrix(8, 8, NEOPIN,
                          NEO_MATRIX_TOP  + NEO_MATRIX_LEFT +
                          NEO_MATRIX_COLUMNS + NEO_MATRIX_PROGRESSIVE,
                          NEO_GRB         + NEO_KHZ800);


    //vector de stocare pt a verifica daca LED-urile sunt pornite sau nu
    int ledMatrix[8][8] = {
      {1, 1, 1, 0, 0, 0, 0, 0},
      {0, 0, 0, 0, 0, 0, 0, 0},
      {0, 0, 0, 0, 0, 0, 0, 0},
      {0, 0, 0, 0, 0, 0, 0, 0},
      {0, 0, 0, 0, 0, 0, 0, 0},
      {0, 0, 0, 0, 0, 0, 0, 0},
      {0, 0, 0, 0, 0, 0, 0, 0},
      {0, 0, 0, 0, 0, 0, 0, 0}
      };

    int ledCount = 64;  //

    void setup() {

    //Serial for debugging
```

```
  Serial.begin(9600);

  // setam pinmodes
  pinMode(NEOPIN, OUTPUT);


  // start clock
  Wire.begin();  // Begin I2C
  RTC.begin();   // begin clock
  RTC.adjust(DateTime(__DATE__, __TIME__));


  matrix.begin();
  matrix.setBrightness(100);
  matrix.fillScreen(0); // Initializam toti pixelii in starea de OFF
  matrix.show();


}

void loop() {

  // get the time
  theTime = dst_rtc.calculateTime(RTC.now());
  // add 2.5 minutes to get better estimates
  theTime = theTime.unixtime() + 150;
//scrie 'it is' de fiecare data:
ledMatrix[0][0] = 1;
ledMatrix[0][1] = 1;
ledMatrix[0][0] = 1;

if ((theTime.minute() > 4) && (theTime.minute() < 10)) {
    ledMatrix[2][0] = 1;
    ledMatrix[2][1] = 1;
    ledMatrix[2][2] = 1;
     Serial.print("mfive ");
  }
  if ((theTime.minute() > 9) && (theTime.minute() < 15)) {
    ledMatrix[0][6] = 1;
    ledMatrix[0][7] = 1;
     Serial.print("mten ");
  }
```

```
if ((theTime.minute() > 14) && (theTime.minute() < 20)) {
  ledMatrix[1][0] = 1;
  ledMatrix[1][1] = 1;
  ledMatrix[1][2] = 1;
  ledMatrix[1][3] = 1;
   Serial.print("mquarter ");
}
if ((theTime.minute() > 19) && (theTime.minute() < 25)) {
  ledMatrix[1][4] = 1;
  ledMatrix[1][5] = 1;
  ledMatrix[1][6] = 1;
  ledMatrix[1][7] = 1;
   Serial.print("mtwenty ");
}
if ((theTime.minute() > 24) && (theTime.minute() < 30)) {
  ledMatrix[1][4] = 1;
  ledMatrix[1][5] = 1;
  ledMatrix[1][6] = 1;
  ledMatrix[1][7] = 1;
  ledMatrix[2][0] = 1;
  ledMatrix[2][1] = 1;
  ledMatrix[2][1] = 1;
   Serial.print("mtwenty five ");
}
if ((theTime.minute() > 29) && (theTime.minute() < 35)) {
  ledMatrix[0][3] = 1;
  ledMatrix[0][4] = 1;
  ledMatrix[0][5] = 1;
   Serial.print("mhalf ");
}
if ((theTime.minute() > 34) && (theTime.minute() < 40)) {
  ledMatrix[1][4] = 1;
  ledMatrix[1][5] = 1;
  ledMatrix[1][6] = 1;
  ledMatrix[1][7] = 1;
  ledMatrix[2][0] = 1;
  ledMatrix[2][1] = 1;
  ledMatrix[2][1] = 1;
   Serial.print("mtwenty five ");
}
if ((theTime.minute() > 39) && (theTime.minute() < 45)) {
  ledMatrix[1][4] = 1;
  ledMatrix[1][5] = 1;
  ledMatrix[1][6] = 1;
  ledMatrix[1][7] = 1;
   Serial.print("mtwenty ");
}
if ((theTime.minute() > 44) && (theTime.minute() < 50)) {
  ledMatrix[1][0] = 1;
  ledMatrix[1][1] = 1;
  ledMatrix[1][2] = 1;
  ledMatrix[1][3] = 1;
   Serial.print("mquarter ");
}
if ((theTime.minute() > 49) && (theTime.minute() < 55)) {
  ledMatrix[0][6] = 1;
  ledMatrix[0][7] = 1;
   Serial.print("mten");
}
if (theTime.minute() > 54) {
  ledMatrix[2][0] = 1;
  ledMatrix[2][1] = 1;
  ledMatrix[2][2] = 1;
   Serial.print("mfive ");
}

if ((theTime.minute() < 5))
{
  switch (theTime.hour()) {
    case 1:
    case 13:
  ledMatrix[3][3] = 1;
  ledMatrix[3][4] = 1;
   Serial.print("hone ");
      break;
    case 2:
    case 14:
  ledMatrix[4][0] = 1;
  ledMatrix[4][1] = 1;
   Serial.print("htwo ");
      break;
    case 3:
    case 15:
  ledMatrix[3][7] = 1;
```

```
ledMatrix[3][6] = 1;
 Serial.print("hthree ");
    break;
  case 4:
  case 16:
ledMatrix[4][2] = 1;
ledMatrix[4][3] = 1;
ledMatrix[4][4] = 1;
 Serial.print("hfour ");
    break;
  case 5:
  case 17:
ledMatrix[5][6] = 1;
ledMatrix[5][7] = 1;
 Serial.print("hfive ");
    break;
  case 6:
  case 18:
ledMatrix[6][6] = 1;
ledMatrix[6][7] = 1;
 Serial.print("hsix ");
    break;
  case 7:
  case 19:
ledMatrix[6][3] = 1;
ledMatrix[6][4] = 1;
ledMatrix[6][5] = 1;
 Serial.print("hseven ");
    break;
  case 8:
  case 20:
ledMatrix[4][5] = 1;
ledMatrix[4][6] = 1;
ledMatrix[4][7] = 1;
 Serial.print("height ");
    break;
  case 9:
  case 21:
ledMatrix[6][0] = 1;
ledMatrix[6][1] = 1;
ledMatrix[6][2] = 1;
 Serial.print("hnine ");
```

```
      break;
    case 10:
    case 22:
  ledMatrix[5][4] = 1;
  ledMatrix[5][5] = 1;
   Serial.print("hten ");
      break;
    case 11:
    case 23:
  ledMatrix[5][0] = 1;
  ledMatrix[5][1] = 1;
  ledMatrix[5][2] = 1;
  ledMatrix[5][3] = 1;
   Serial.print("heleven ");
      break;
    case 0:
    case 12:
  ledMatrix[7][0] = 1;
  ledMatrix[7][1] = 1;
  ledMatrix[7][2] = 1;
  ledMatrix[7][3] = 1;
   Serial.print("htwelve ");
      break;
  }

}
else if ((theTime.minute() < 35) && (theTime.minute() > 4))
{
  ledMatrix[3][0] = 1;
  ledMatrix[3][1] = 1;
  ledMatrix[3][2] = 1;
   Serial.print("past");
 switch (theTime.hour()) {
    case 1:
    case 13:
  ledMatrix[3][3] = 1;
  ledMatrix[3][4] = 1;
   Serial.print("one ");
      break;
    case 2:
    case 14:
  ledMatrix[4][0] = 1;
```

```
ledMatrix[4][1] = 1;
 Serial.print("two ");
    break;
  case 3:
  case 15:
ledMatrix[3][7] = 1;
ledMatrix[3][6] = 1;
 Serial.print("three ");
    break;
  case 4:
  case 16:
ledMatrix[4][2] = 1;
ledMatrix[4][3] = 1;
ledMatrix[4][4] = 1;
 Serial.print("four ");
    break;
  case 5:
  case 17:
ledMatrix[5][6] = 1;
ledMatrix[5][7] = 1;
 Serial.print("five ");
    break;
  case 6:
  case 18:
ledMatrix[6][6] = 1;
ledMatrix[6][7] = 1;
 Serial.print("six ");
    break;
  case 7:
  case 19:
ledMatrix[6][3] = 1;
ledMatrix[6][4] = 1;
ledMatrix[6][5] = 1;
 Serial.print("seven ");
    break;
  case 8:
  case 20:
ledMatrix[4][5] = 1;
ledMatrix[4][6] = 1;
ledMatrix[4][7] = 1;
 Serial.print("eight ");
    break;
    case 9:
    case 21:
  ledMatrix[6][0] = 1;
  ledMatrix[6][1] = 1;
  ledMatrix[6][2] = 1;
   Serial.print("nine ");
      break;
    case 10:
    case 22:
  ledMatrix[5][4] = 1;
  ledMatrix[5][5] = 1;
   Serial.print("ten ");
      break;
    case 11:
    case 23:
  ledMatrix[5][0] = 1;
  ledMatrix[5][1] = 1;
  ledMatrix[5][2] = 1;
  ledMatrix[5][3] = 1;
   Serial.print("eleven ");
      break;
    case 0:
    case 12:
  ledMatrix[7][0] = 1;
  ledMatrix[7][1] = 1;
  ledMatrix[7][2] = 1;
  ledMatrix[7][3] = 1;
   Serial.print("twelve ");
      break;
  }
}
else
{

  ledMatrix[2][6] = 1;
  ledMatrix[2][7] = 1;
   Serial.print("to ");
  //Serial.print(" to ");
  switch (theTime.hour()) {
    case 1:
    case 13:
  ledMatrix[4][0] = 1;
```

```
ledMatrix[4][1] = 1;
 Serial.print("two ");
    break;
  case 14:
  case 2:
ledMatrix[3][7] = 1;
ledMatrix[3][6] = 1;
 Serial.print("three ");
    break;
  case 15:
  case 3:
ledMatrix[4][2] = 1;
ledMatrix[4][3] = 1;
ledMatrix[4][4] = 1;
 Serial.print("four ");
    break;
  case 4:
  case 16:
ledMatrix[5][6] = 1;
ledMatrix[5][7] = 1;
 Serial.print("five ");
    break;
  case 5:
  case 17:
ledMatrix[6][6] = 1;
ledMatrix[6][7] = 1;
 Serial.print("six ");
    break;
  case 6:
  case 18:
ledMatrix[6][3] = 1;
ledMatrix[6][4] = 1;
ledMatrix[6][5] = 1;
 Serial.print("seven ");
    break;
  case 7:
  case 19:
ledMatrix[4][5] = 1;
ledMatrix[4][6] = 1;
ledMatrix[4][7] = 1;
 Serial.print("eight ");
    break;
    case 8:
    case 20:
  ledMatrix[6][0] = 1;
  ledMatrix[6][1] = 1;
  ledMatrix[6][2] = 1;
   Serial.print("nine ");
      break;
    case 9:
    case 21:
  ledMatrix[5][4] = 1;
  ledMatrix[5][5] = 1;
   Serial.print("ten ");
      break;
    case 10:
    case 22:
  ledMatrix[5][0] = 1;
  ledMatrix[5][1] = 1;
  ledMatrix[5][2] = 1;
  ledMatrix[5][3] = 1;
   Serial.print("eleven ");
      break;
    case 11:
    case 23:
  ledMatrix[7][0] = 1;
  ledMatrix[7][1] = 1;
  ledMatrix[7][2] = 1;
  ledMatrix[7][3] = 1;
   Serial.print("twelve ");
      break;
    case 0:
    case 12:
  ledMatrix[3][3] = 1;
  ledMatrix[3][4] = 1;
   Serial.print("one ");
      break;
  }
 }


for(int i = 0; i < 8; i++) {
  for(int j = 0; j < 8; j++) {
    if (ledMatrix[i][j] == 1) {
        matrix.drawPixel(i, j, matrix.Color(255, 255, 0));
      }
      else {
        matrix.drawPixel(i, j, matrix.Color(0, 0, 0));
      }

    }
    Serial.println();
  }
displayTime();
checkArray();
matrix.show();

delay(5000);
resetArray();

}

void checkArray(){
  //printam vectorul pt verificare:
  for(int i = 0; i < 8; i++) {
    for(int j = 0; j < 8; j++) {
      Serial.print(ledMatrix[i][j]);
      Serial.print(",");
    }
    Serial.println();
  }
  Serial.println();
}
void resetArray(){
  //set all the array values to 0 (off):
  for(int i = 0; i < 8; i++) {
    for(int j = 0; j < 8; j++) {
      ledMatrix[i][j] = 0;
    }
  }
  Serial.print("Array reset (All LEDs switched set to off)");
  Serial.println();
  Serial.println();
}
void displayTime() {
  // luam timpul din RTC
  DateTime theTime = RTC.now();
  // printam timpul actual
  Serial.print(theTime.year(), DEC);
  Serial.print('/');
  Serial.print(theTime.month(), DEC);
  Serial.print('/');
  Serial.print(theTime.day(), DEC);
  Serial.print(' ');
  Serial.print(theTime.hour(), DEC);
  Serial.print(':');
  Serial.print(theTime.minute(), DEC);
  Serial.print(':');
  Serial.print(theTime.second(), DEC);
  Serial.println();
}
```

Avantaje fata de solutiile cunoscute:

Solutia mea ofera un mod creativ si unic de afisare a timpului, usor de utilizat de catre consummator, la un pret redus fata de alte tehnologii asemanatoare aflate pe piata in acest moment.