

Ļoti ļoti daudz mirdzdiožu!

Ivars Driķis

2019. gada 4. septembrī

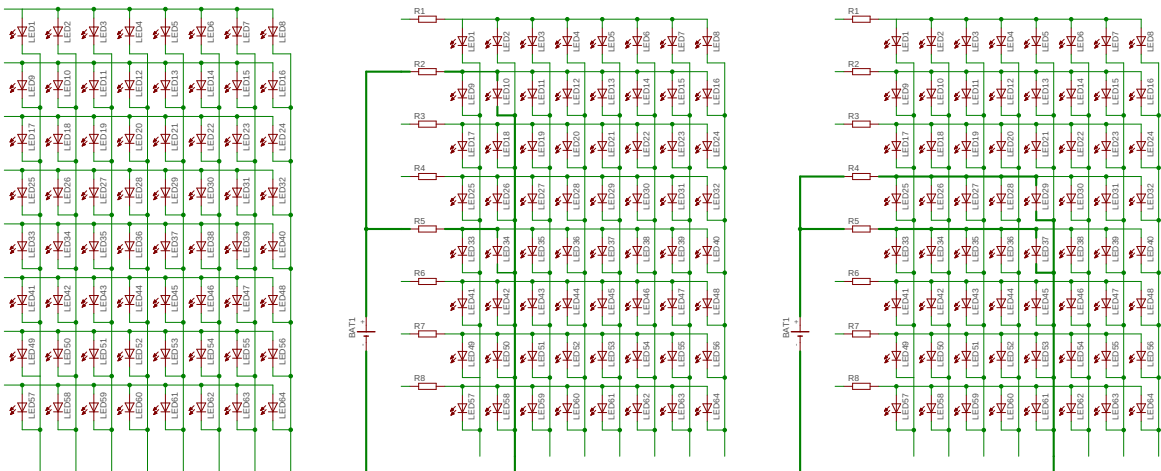
1 Kā pieslēgt ļoti daudz mirdzdiožu?

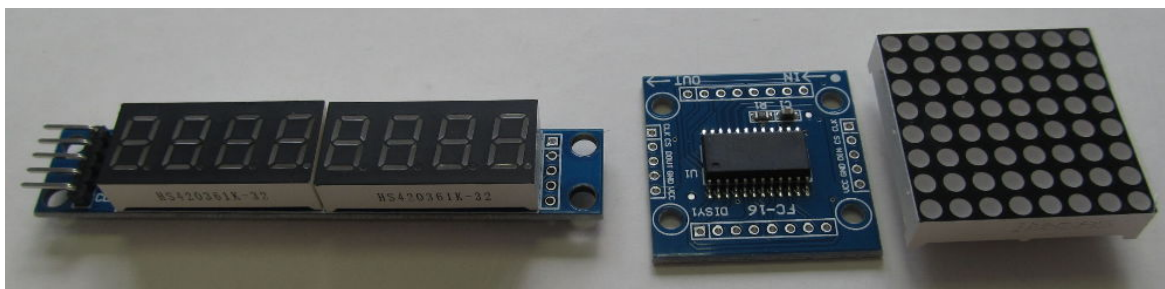
Ja nepieciešams vadīt ar Arduino vairākus 7 segmentu indikatorus vai arī vairākus desmitus mirdzdiožu¹, tad viegli izrēķināt, ka visu Arduino izeju nepietiek lai pieslēgtu nepieciešamo skaitu mirdzdiožu. Ne mazāk nozīmīga problēma ir arī milzīgais savienotājvadu skaits, kas šādai konstrukcijai nepieciešams. Risinājums ir asprātīgs un vienkāršs, pie kam atrisina abas problēmas (sk. 1 attēlā pa kreisi): ir “horizontālie” vadi un “vertikālie” vadi un to krustpunktos ievietotas mirdzdiodes kā tas redzams. Tas strādā, redzams blakus attēlos. Ja pie barošanas avota pieslēdz 2 “vertikālo” un 2 un 5 “horizontālo” līnijas, tad spīdēs tikai 10 un 34 mirdzdiodes. Savukārt, ja pie barošanas avota pieslēdz 5 “vertikālo” un 4 un 5 “horizontālo” līnijas, tad spīdēs tikai 29 un 37 mirdzdiodes. Tātad, lai vadītu 64 mirdzdiodes, pietiek ar 16 Arduino izejām kā arī ar tikai 16 vadiem! Atliek tikai pietiekoši ātri pārslēgt “vertikālās” līnijas, lai cilvēka acs nejustu pārtraukumus mirdzdiodes spīdēšanā.

Izskatīsim iespēju šādu matricu pieslēgt tieši Arduino izejām. Pieredze rāda, ka mirdzdiode pietiekami spīgti spīd patērējot 2 mA strāvu. Tā kā strāva diodei tiek padota tikai 1/8 no tās darbības laika, tad mirdzdiodei ir jāpadod 16 mA. Diodes ir konstruētas, lai šādu režīmu izturētu. Tā kā vienai “vertikālajai” līnijai var nākties strādāt 8 diodēm, kas tai pieslēgtas, tad maksimālā strāva, kura jāiztur Arduino līnijai ir 128 mA. Dokumentācijā atrodam, ka maksimālā strāva, kura nedrīkst pārsniegt Arduino līnijā pieslēdzot to GND (sink current) ir 40 mA, tātad stipri par maz un nepieciešami papildus tranzistori, kas šo strāvu var izturēt. Bet lietderīgāk ir izmantot specializētu mikroshēmu MAX7219 [1], kas ne tikai nodrošina nepieciešamo strāvu “vertikālajās” un “horizontālajās” līnijas, bet arī patstāvīgi

¹Tādi bija mans pirmais un otrais Arduino projekti

Att. 1: 8x8 mirdzdiožu matrica un tās darbība.





Att. 2: Rūpnieciski ražoti MAX7219 paneli 7 segmentu indikatora un 8x8 diožu matricas vadīšanai

veic nepieciešamās līniju pārslēgšanas, lai ieslēgtu tikai tās mirdzdiodes, kuras norādītas šās mikroshēmas iekšējā atmiņā. Savukārt informāciju par mirdzdiodes uz dotās mikroshēmas atmiņu nogādā izmantojot tikai 3 datu līnijas. Rūpnieciski ražotu panelu piemēri redzami 2 attēlā. Pa kreisi panelis, kas vada astoņus 7-segmentu indikatorus, pa labi viena no versijām 8x8 diožu matricas vadīšanai.

Ļoti iespaidīgi izskatās skrejošo uzrakstu lentu, kas izveidota no 8x8 diožu matricām. Tāpēc tika nolemts 2 attēlā redzamo 8x8 diožu matricas paneli papildināt ar papildus platīti, kas ļauj ērti un ātri izveidot diožu matricu lentu. Kas no tā sanāca, redzams 3 attēlā.

2 Vienkāršs 8x8 mirdzdiožu ekrāns

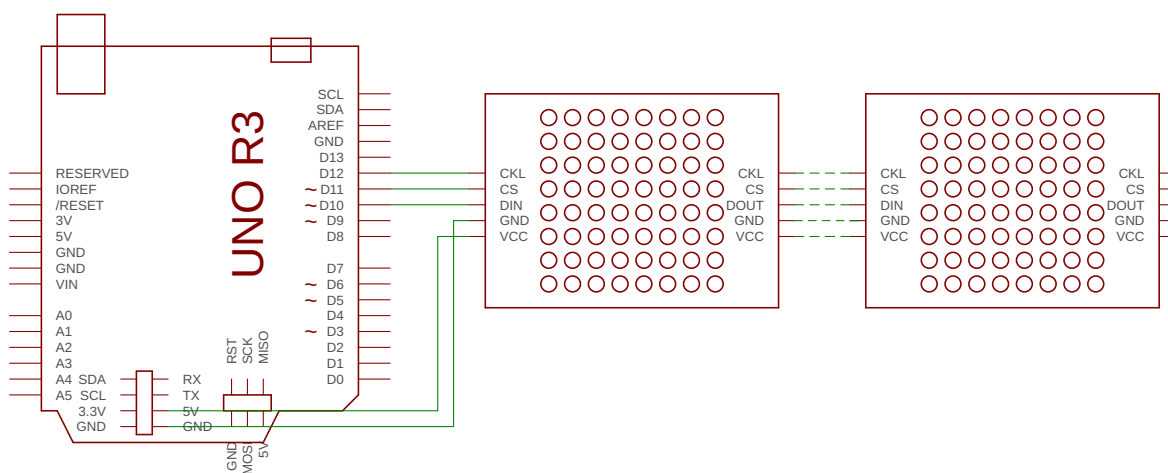
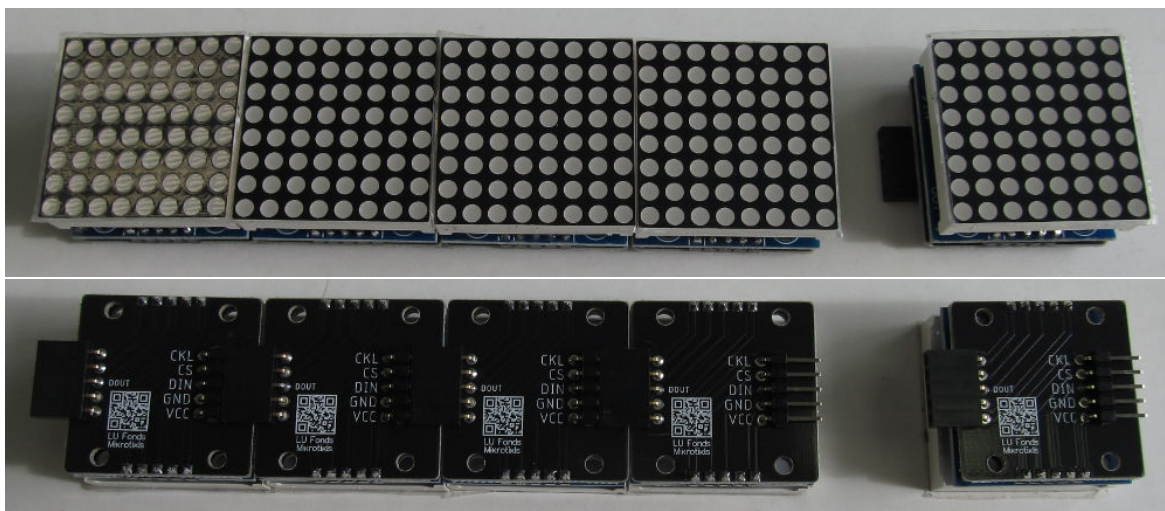
Lai vadītu 8x8 mirdzdiožu ekrānu izmantot MAX7219 mikroshēmu lietderīgi izmantot LedControl bibliotēku [2]. Kā to darīt, skatamies piemēros kā arī ļoti labi izveidotā šeit [3]. Kā pievieno bibliotēkas, var lasīt arī manā iepriekšējā blogā [4]. Izaicinājums būtu izrakstīt Conway's Game of Life [5] priekš vairākām šādām matricām. Tālāk dotais piemērs ņemts no LedControl bibliotēkas piemēriem. Šeit soli pa solim tiek ieslēgtas katra no matricas mirdzdiodes, pēc tam tādā pašā secībā tās tiek izslēgtas.

Listing 1: led8x8.ino

```

1 #include <LedControl.h>
2
3 const int pinDIN = 10;
4 const int pinCLK = 12;
5 const int pinCS = 11;
6
7 LedControl lc=LedControl(pinDIN, pinCLK, pinCS, 1);
8 int displ = 0;
9
10 void setup()
11 {
12   lc.shutdown(displ, false);
13   lc.setIntensity(displ, 8);
14   lc.clearDisplay( displ );
15 }
16
17 void loop()
18 {
19   for (int row=0; row<8; row++) {
20     for (int col=0; col<8; col++) {
21       lc.setLed( displ, col, row, true );
22       delay(25);
23     }

```



Att. 3: Uzlabotie MAX7219 paneļi 7 segmentu indikatora un 8x8 vadīšanai kā arī tā pieslēgšana pie Arduino

```

24 }
25
26 for (int row=0; row<8; row++) {
27     for (int col=0; col<8; col++) {
28         lc.setLed(displ, col, row, false);
29         delay(25);
30     }
31 }
32 }

```

3 Skrejošais teksts

Lai izveidotu skrejošo tekstu, iesaku izmantot MaxMatrix bibliotēku [6]. Lai to izmantotu, jārikojas šādi:

- Dodamies uz GitHub sadaļu [6], spiežam Clone or Download un izvēlamies Download ZIP. Iegūstam failu max7219-master.zip.

- Atarhivējam šo failu un tur iekša esošo katalogu MaxMatrix pārkopējam un katalogu Arduino/libraries/ pie pārējām bibliotēka.

Nedaudz pārvietoju tur esošo programmu, izlaboju kļūdiņu un viss strādā līdz pat 9 indikatoriem.

Listing 2: led8x8.ino

```

1  /*
2     Based on the following library:
3     GitHub | riyas - org/max7219  https://github.com/riyas - org/max7219
4  */
5
6  #include <MaxMatrix.h>
7  #include <avr/pgmspace.h>
8
9  PROGMEM const unsigned char CH[] = {
10  3, 8, B00000000, B00000000, B00000000, B00000000, B00000000, // space
11  1, 8, B01011111, B00000000, B00000000, B00000000, B00000000, // !
12  3, 8, B00000011, B00000000, B00000011, B00000000, B00000000, // "
13  5, 8, B00010100, B00111110, B00010100, B00111110, B00010100, // #
14  4, 8, B00100100, B01101010, B00101011, B00010010, B00000000, // $
15  5, 8, B01100011, B00010011, B00001000, B01100100, B01100011, // %
16  5, 8, B00110110, B01001001, B01010110, B00100000, B01010000, // &
17  1, 8, B00000011, B00000000, B00000000, B00000000, B00000000, // '
18  3, 8, B00011100, B00100010, B01000001, B00000000, B00000000, // (
19  3, 8, B01000001, B00100010, B00011100, B00000000, B00000000, // )
20  5, 8, B00101000, B00011000, B00001110, B00011000, B00101000, // *
21  5, 8, B00001000, B00001000, B00111110, B00001000, B00001000, // +
22  2, 8, B10110000, B01110000, B00000000, B00000000, B00000000, // ,
23  4, 8, B00001000, B00001000, B00001000, B00001000, B00000000, // -
24  2, 8, B01100000, B01100000, B00000000, B00000000, B00000000, // .
25  4, 8, B01100000, B00011000, B00000110, B00000001, B00000000, // /
26  4, 8, B00111110, B01000001, B01000001, B00111110, B00000000, // 0
27  3, 8, B01000010, B01111111, B01000000, B00000000, B00000000, // 1
28  4, 8, B01100010, B01010001, B01001001, B01000110, B00000000, // 2
29  4, 8, B00100010, B01000001, B01001001, B00110110, B00000000, // 3
30  4, 8, B00011000, B00010100, B00010010, B01111111, B00000000, // 4
31  4, 8, B00100111, B01000101, B01000101, B00111001, B00000000, // 5
32  4, 8, B00111110, B01001001, B01001001, B00110000, B00000000, // 6
33  4, 8, B01100001, B00010001, B00001001, B00000111, B00000000, // 7
34  4, 8, B00110110, B01001001, B01001001, B00110110, B00000000, // 8
35  4, 8, B00000110, B01001001, B01001001, B00111110, B00000000, // 9
36  2, 8, B01010000, B00000000, B00000000, B00000000, B00000000, // :
37  2, 8, B10000000, B01010000, B00000000, B00000000, B00000000, // ;
38  3, 8, B00010000, B00101000, B01000100, B00000000, B00000000, // <
39  3, 8, B00010100, B00010100, B00010100, B00000000, B00000000, // =
40  3, 8, B01000100, B00101000, B00010000, B00000000, B00000000, // >
41  4, 8, B00000010, B01011001, B00001001, B00000110, B00000000, // ?
42  5, 8, B00111110, B01001001, B01010101, B01011101, B00001110, // @
43  4, 8, B01111110, B00010001, B00010001, B01111110, B00000000, // A
44  4, 8, B01111111, B01001001, B01001001, B00110110, B00000000, // B
45  4, 8, B00111110, B01000001, B01000001, B00100010, B00000000, // C
46  4, 8, B01111111, B01000001, B01000001, B00111110, B00000000, // D
47  4, 8, B01111111, B01001001, B01001001, B01000001, B00000000, // E
48  4, 8, B01111111, B00001001, B00001001, B00000001, B00000000, // F
49  4, 8, B00111110, B01000001, B01001001, B01111010, B00000000, // G
50  4, 8, B01111111, B00001000, B00001000, B01111111, B00000000, // H
51  3, 8, B01000001, B01111111, B01000001, B00000000, B00000000, // I
52  4, 8, B00110000, B01000000, B01000001, B00111111, B00000000, // J
53  4, 8, B01111111, B00001000, B00010100, B01100011, B00000000, // K
54  4, 8, B01111111, B01000000, B01000000, B01000000, B00000000, // L

```

```

55 5, 8, B01111111, B00000010, B00001100, B00000010, B01111111, // M
56 5, 8, B01111111, B00000100, B00001000, B00010000, B01111111, // N
57 4, 8, B00111110, B01000001, B01000001, B00111110, B00000000, // O
58 4, 8, B01111111, B00001001, B00001001, B00000110, B00000000, // P
59 4, 8, B00111110, B01000001, B01000001, B10111110, B00000000, // Q
60 4, 8, B01111111, B00001001, B00001001, B01110110, B00000000, // R
61 4, 8, B01000110, B01001001, B01001001, B00110010, B00000000, // S
62 5, 8, B00000001, B00000001, B01111111, B00000001, B00000001, // T
63 4, 8, B00111111, B01000000, B01000000, B00111111, B00000000, // U
64 5, 8, B00001111, B00110000, B01000000, B00110000, B00001111, // V
65 5, 8, B00111111, B01000000, B00111000, B01000000, B00111111, // W
66 5, 8, B01100011, B00010100, B00001000, B00010100, B01100011, // X
67 5, 8, B00000111, B00001000, B01110000, B00001000, B00000111, // Y
68 4, 8, B01100001, B01010001, B01001001, B01000111, B00000000, // Z
69 2, 8, B01111111, B01000001, B00000000, B00000000, B00000000, // [
70 4, 8, B00000001, B00000110, B00011000, B01100000, B00000000, // \ backslash
71 2, 8, B01000001, B01111111, B00000000, B00000000, B00000000, // ]
72 3, 8, B00000010, B00000001, B00000010, B00000000, B00000000, // ^ hat
73 4, 8, B01000000, B01000000, B01000000, B01000000, B00000000, // _
74 2, 8, B00000001, B00000010, B00000000, B00000000, B00000000, // `
75 4, 8, B00100000, B01010100, B01010100, B01111000, B00000000, // a
76 4, 8, B01111111, B01000100, B01000100, B00111000, B00000000, // b
77 4, 8, B00111000, B01000100, B01000100, B00101000, B00000000, // c
78 4, 8, B00111000, B01000100, B01000100, B01111111, B00000000, // d
79 4, 8, B00111000, B01010100, B01010100, B00011000, B00000000, // e
80 3, 8, B00000100, B01111110, B00000101, B00000000, B00000000, // f
81 4, 8, B10011000, B10100100, B10100100, B01111000, B00000000, // g
82 4, 8, B01111111, B00000100, B00000100, B01111000, B00000000, // h
83 3, 8, B01000100, B01111101, B01000000, B00000000, B00000000, // i
84 4, 8, B01000000, B10000000, B10000100, B01111101, B00000000, // j
85 4, 8, B01111111, B00010000, B00101000, B01000100, B00000000, // k
86 3, 8, B01000001, B01111111, B01000000, B00000000, B00000000, // l
87 5, 8, B01111100, B00000100, B01111100, B00000100, B01111000, // m
88 4, 8, B01111100, B00000100, B00000100, B01111000, B00000000, // n
89 4, 8, B00111000, B01000100, B01000100, B00111000, B00000000, // o
90 4, 8, B11111100, B00100100, B00100100, B00011000, B00000000, // p
91 4, 8, B00011000, B00100100, B00100100, B11111100, B00000000, // q
92 4, 8, B01111100, B00001000, B00000100, B00000100, B00000000, // r
93 4, 8, B01001000, B01010100, B01010100, B00100100, B00000000, // s
94 3, 8, B00000100, B00111111, B01000100, B00000000, B00000000, // t
95 4, 8, B00111100, B01000000, B01000000, B01111100, B00000000, // u
96 5, 8, B00011100, B00100000, B01000000, B00100000, B00011100, // v
97 5, 8, B00111100, B01000000, B00111100, B01000000, B00111100, // w
98 5, 8, B01000100, B00101000, B00010000, B00101000, B01000100, // x
99 4, 8, B10011100, B10100000, B10100000, B01111100, B00000000, // y
100 3, 8, B01100100, B01010100, B01001100, B00000000, B00000000, // z
101 3, 8, B00001000, B00110110, B01000001, B00000000, B00000000, // {
102 1, 8, B01111111, B00000000, B00000000, B00000000, B00000000, // |
103 3, 8, B01000001, B00110110, B00001000, B00000000, B00000000, // }
104 4, 8, B00001000, B00000100, B00001000, B00000100, B00000000, // ~
105 };
106
107 int DIN = 10; // DIN pin of MAX7219 module
108 int CLK = 12; // CLK pin of MAX7219 module
109 int CS = 11; // CS pin of MAX7219 module
110 int maxInUse = 9; // 1 ... 9
111
112 MaxMatrix m(DIN, CS, CLK, maxInUse);
113 byte buffer[10];
114
115 void setup()
116 {

```

```
117 m.init();
118 m.setIntensity(15);
119 }
120
121 void loop()
122 {
123   printStringWithShift("LU Fonds & Mikrotikls ", 10);
124 }
125
126 void printCharWithShift(char c, int shift_speed)
127 {
128   long start, duration;
129
130   if (c < 32) return;
131
132   memcpy_P(buffer, CH + 7 * (c-32), 7);
133   m.writeSprite(8*maxInUse, 0, buffer);
134   m.setColumn(8*maxInUse + buffer[0], 0);
135
136   for (int i = 0; i < buffer[0] + 1; i++) {
137     start = millis();
138     m.shiftLeft(false, false);
139     duration = shift_speed - (millis() - start);
140     if (duration > 0) delay(duration);
141   }
142 }
143
144 void printStringWithShift(char* s, int shift_speed)
145 {
146   while(*s != 0) {
147     printCharWithShift(*s, shift_speed);
148     s++;
149   }
150 }
```

Pateicības

Platīte izgatavota Latvijas Universitātes Fonda projekta “Programmējam ar prieku” ietvaros un to finansē AS Mikrotīkls.

Literatūras saraksts

- [1] Serially Interfaced, 8-Digit LED Display Drivers MAX7219/MAX7221. <https://datasheets.maximintegrated.com/en/ds/MAX7219-MAX7221.pdf>. [Online; accessed 2017.02.10].
- [2] Arduino Reference. LedControl library. <http://playground.arduino.cc/Main/LedControl>. [Online; accessed 2017.02.10].
- [3] Tutorial – Arduino and the MAX7219 LED Display Driver IC. <http://tronixstuff.com/2013/10/11/tutorial-arduino-max7219-led-display-driver-ic/>. [Online; accessed 2017.02.10].
- [4] Ivars Driķis. Attāluma mērīšana un bibliotēkas. <http://blogi.lu.lv/drikis/2018/04/24/attaluma-merisana-un-bibliotekas/>. [Online; accessed 2018.09.24].

- [5] Conway's Game of Life. https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life. [Online; accessed 2017.02.10].
- [6] Led matrix library. <https://github.com/riyas-org/max7219/>. [Online; accessed 2018.09.24].