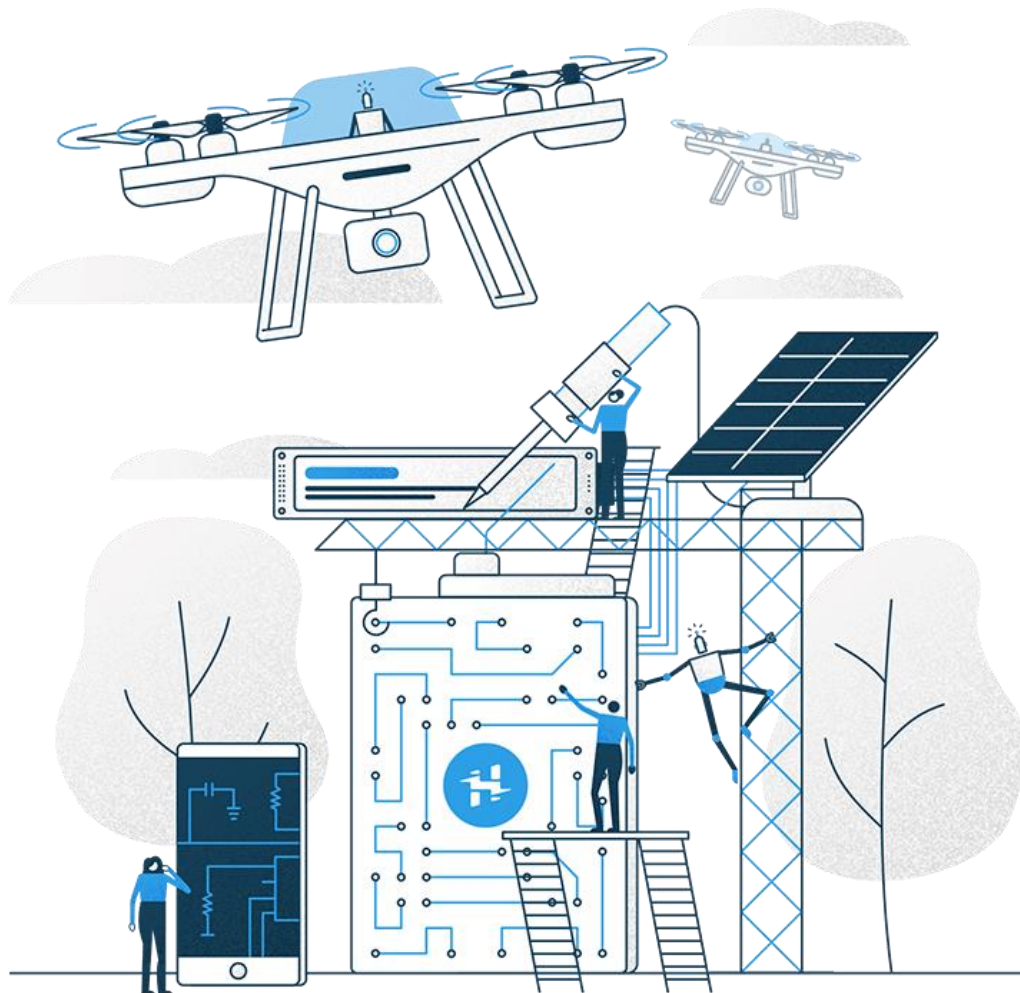


МІКРОПРОЦЕСОРНІ СИСТЕМИ УПРАВЛІННЯ



Lesson 11_3

Матрична індикація. Dot Matrix

Матричні світлодіодні індикатори (МСІ) використовуються для відображення алфавітно-цифрової інформації. Кожен з таких МСІ, виконаний у вигляді інтегральної мікросхеми, є матрицею світлодіодів розмірністю $m \times n$, де n - число стовпчиків, m - число рядків матриці. Найбільшого поширення набули МСІ з розмірністю матриці 7×5 , 9×7 , 8×8 (рис. 1).

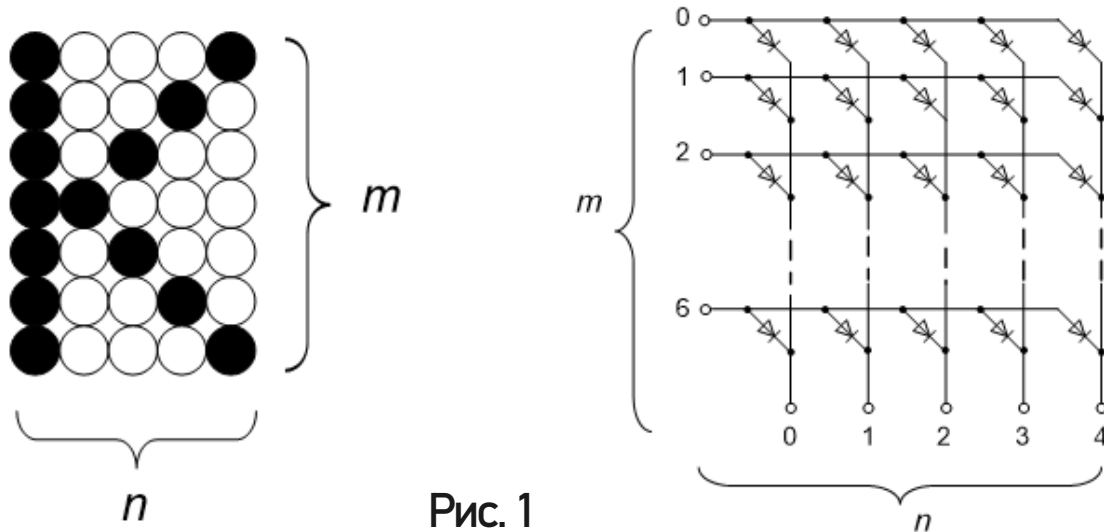
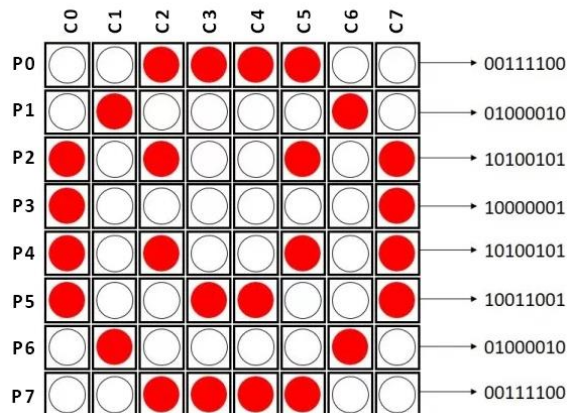
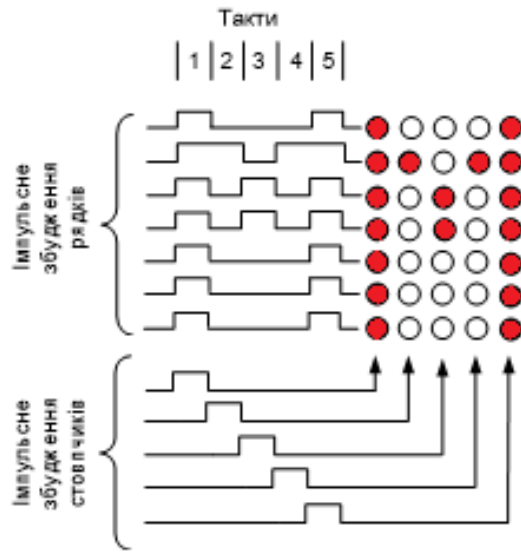


Рис. 1

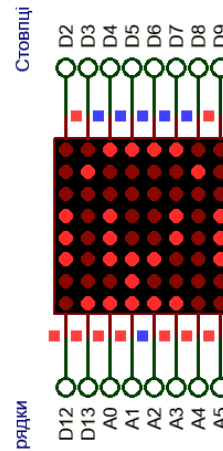
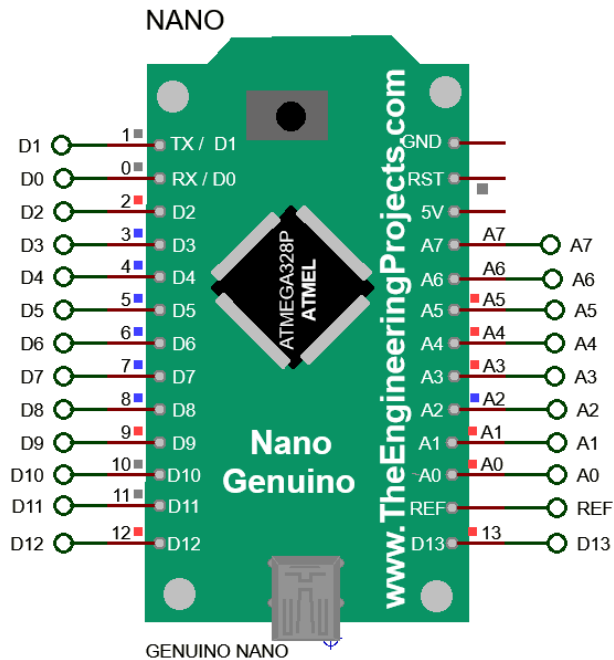
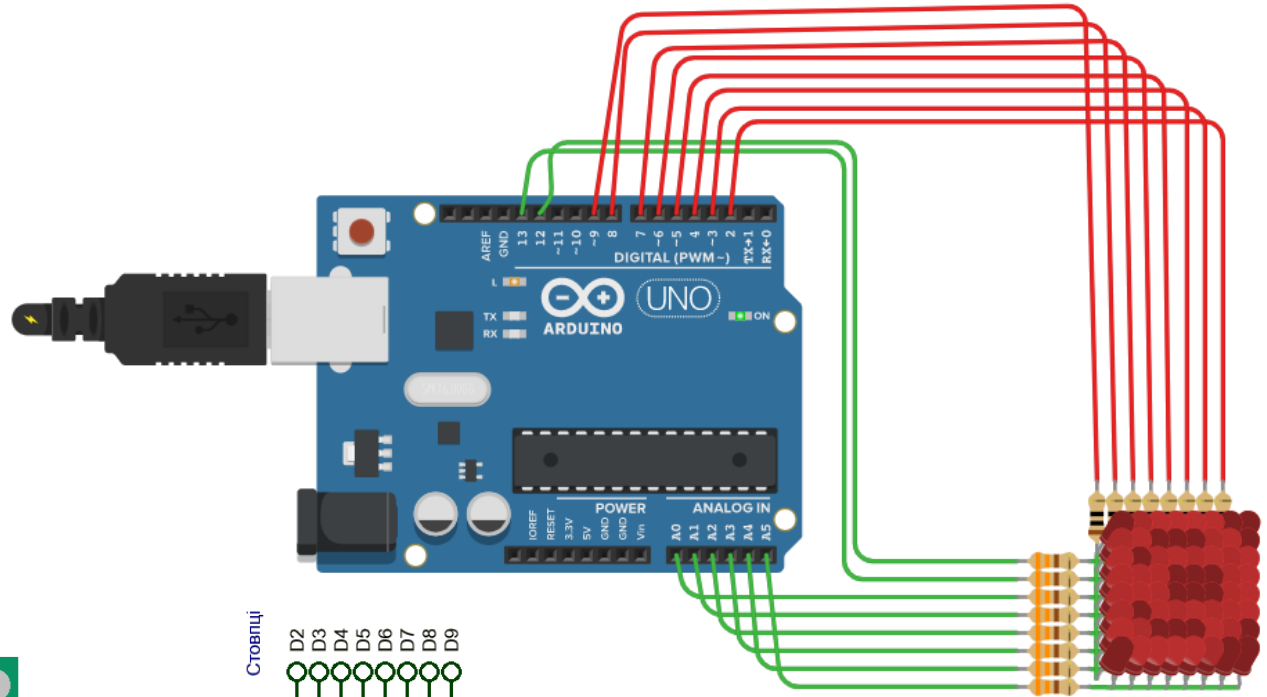
У кожному часовому такті збуджується строб імпульс відповідного стовпця. У результаті відбувається відображення інформації у всіх елементів даного стовпця. Після кожного такту відбувається зсув інформації і в наступному часовому такті збуджується строб імпульс у другому стовпчику і так далі. За п'ять тактів відбувається передача повної інформації на матричний індикатор, після чого відбувається повторення передачі, якщо по шині введення даних не поступила нова інформація. Часова діаграма формування букви М представлена на рисунку 2.

Матрична індикація. Dot Matrix



```
1 byte hf[8]= {
2   B00111100,
3   B01000010,
4   B10100101,
5   B10000001,
6   B10100101,
7   B10011001,
8   B01000010,
9   B00111100};
10
11 byte anode[8]={12, 13, 14, 15, 16 ,17, 18, 19};
12 byte catode[8]={2, 3, 4, 5, 6, 7, 8, 9};
13 void setup(){
14   for (int i=0; i<8; i++){
15     pinMode(catode[i], OUTPUT);}
16   for (int i=0; i<8; i++){
17     pinMode(anode[i], OUTPUT);}
18 }
19 void loop(){
20   for (int i=0; i<8; i++){
21     digitalWrite(anode[i], LOW);
22     for (int j=0; j<8; j++){
23       digitalWrite(catode[j],bitRead(hf[i],j));}
24     digitalWrite(anode[i], HIGH);}
25 }
```

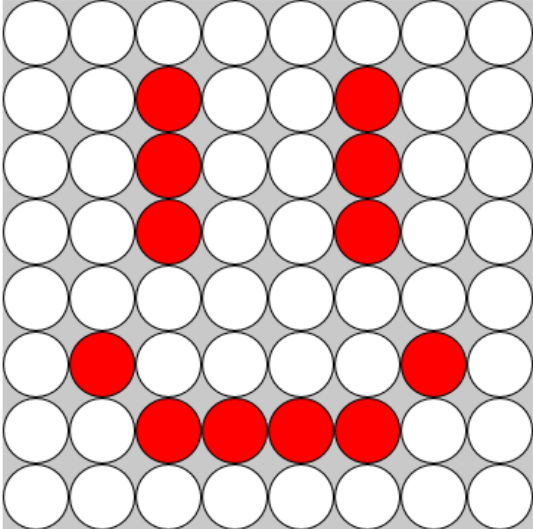
Матрична індикація. Dot Matrix



Матрична індикація. Dot Matrix

Matrix Graphics Generator

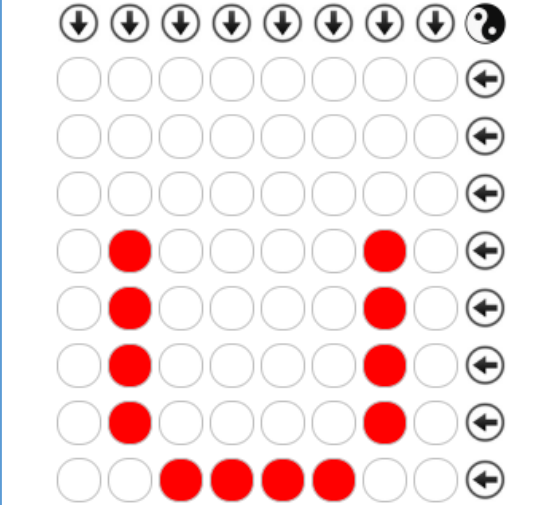
Collection Graphic



Code

0x24, 0x24, 0x24, 0x0, 0x42, 0x3c, 0x0, 0x0

Happy Face



0x0, 0x0, 0x0, 0x42, 0x42, 0x42, 0x42, 0x3c

LED Matrix Array Generator

Columns

Rows

Array name

LEDs

0	1	0	0	0	0	1	0
1	0	0	0	0	0	0	1
1	0	0	0	0	0	0	1
1	0	0	0	0	0	0	1
1	0	0	0	0	0	0	1
1	0	0	0	0	0	0	1
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0

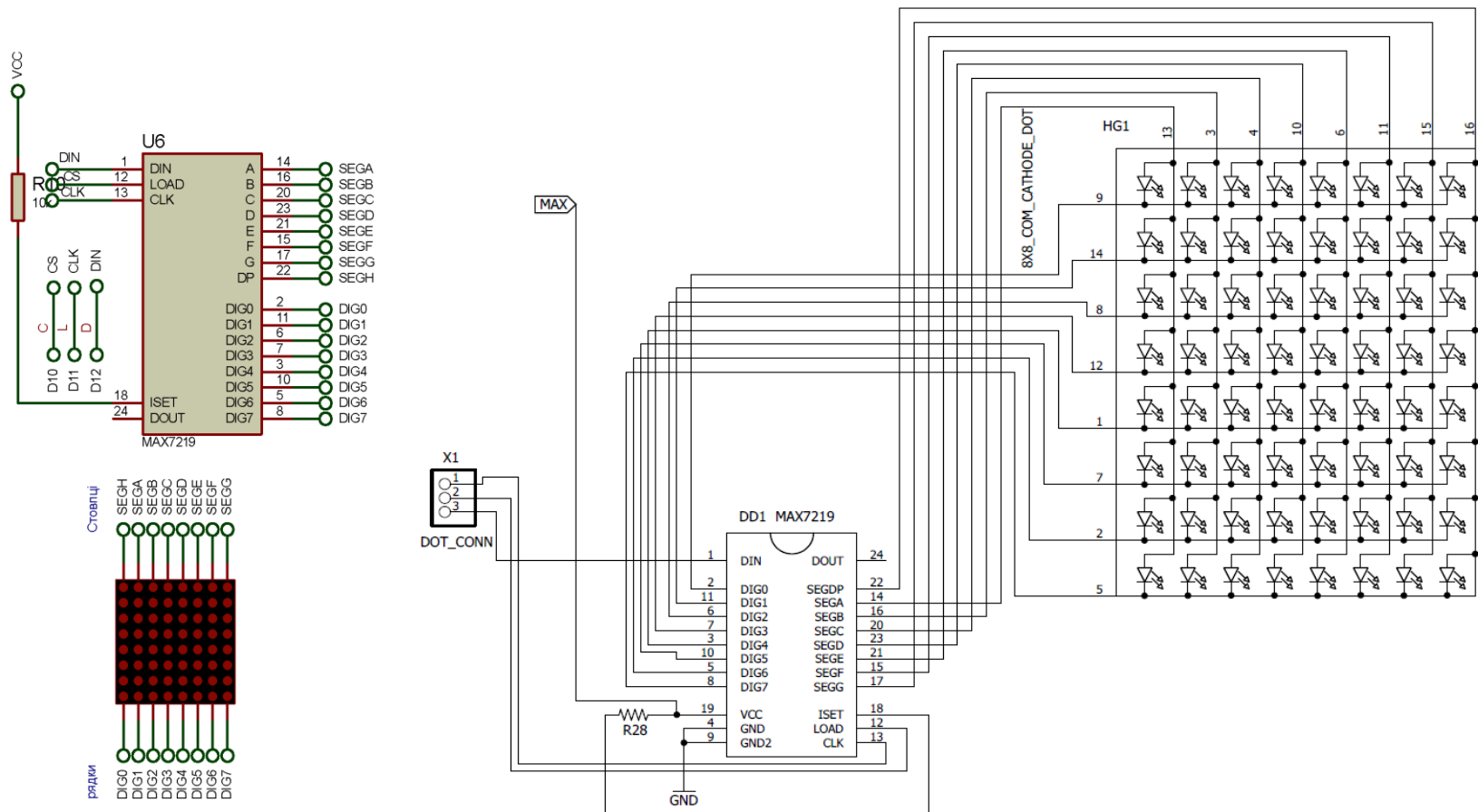
Arduino Python

Code

```
byte image[8][8] = {
  {0,1,0,0,0,0,1,0},
  {1,0,0,0,0,0,0,1},
  {1,0,0,0,0,0,0,1},
  {1,0,0,0,0,0,0,1},
  {1,0,0,0,0,0,0,1},
  {1,0,0,0,0,0,0,1},
  {0,1,0,0,0,0,1,0},
  {0,1,0,0,0,0,1,0}};
```

Матрична індикація. Dot Matrix

Спеціалізована мікросхема-драйвер MAX7219 (рис. 3) призначена для керування світлодіодною матрицею 8×8. MAX7219 — це 16-бітний регістр послідовного зсуву. Перші 8 біт задають команду, а решта 8 біт використовуються для визначення даних для команди. На 18 ніжку Iset підключається резистор «pull up», який встановлює піковий струм для сегментів. Виводи DIN (вхід даних), CLK (вхід для синхроімпульсів) і CS (вхід вибору мікросхеми) через X1 з'єднують з цифровими портами плати Arduino.



Матрична індикація. Dot Matrix

Управляти світлодіодним матрицею можна не тільки самостійно, але і за допомогою різних бібліотек. Одна з таких бібліотек **LedControl.h**.

Для роботи з бібліотекою необхідно створимо об'єкт класу **LedControl**. Типовий код ініціалізації бібліотеки буде виглядати так:

```
#include "LedControl.h"
//DIN 12, CLK 11, CS 10
LedControl LC = LedControl(12, 11, 10, 1);
```

При підключенні MAX72xx можна використовувати будь-які порти на платі Arduino, але так як є входи / виходи які використовуються для послідовного з'єднання (порти 0 та 1), а також порти які мають у своїй зв'язці світлодіодний індикатор (порт 13), то краще уникати підключення до цих портів вибравши будь-які інші доступні. Також, порти зазначені в оголошенні об'єкта класу, не потрібно формувати функцією `pinMode()` в розділі `setup()` програми, бібліотека `LedControl` при створенні об'єкта класу сама проініціалізує ці порти потрібним чином.

Четвертий параметр є кількістю пристроїв з драйвером MAX72xx підключених до однієї шини каскадом. Одному об'єкту класу `LedControl` можна адресувати до 8 пристроїв, при цьому, чим більше пристроїв на шині, тим відповідно нижче її продуктивність. Дозволені тільки значення від 1 до 8 включно, оскільки одному об'єкту класу `LedControl` можна адресувати понад 8-ми пристроїв. Якщо необхідно управляти більш вісьмома пристроями на базі драйвера MAX72xx, то можна створити ще кілька об'єктів класу `LedControl`.

Матрична індикація. Dot Matrix

Для цього потрібно використовувати іншу групу контактів підключення пристрою, відмінну від першого об'єкта класу LedControl . Наприклад ось так:

```
#include "LedControl.h"
LedControl LC1 = LedControl(12, 11, 10, 8);
LedControl LC2 = LedControl(9, 8, 7, 8);
```

При ініціалізації пристрою на базі драйвера MAX72xx (за умови підключення всього лише одного пристрою), можна використовувати таку конструкцію коду в розділі програми setup ():

```
void setup() {
    // Відключення режиму енергозбереження
    // Функція shutdown()
    LC1.shutdown(0, false);
    //Встановлення яскравості світіння світлодіодів у матриці
    LC1.setIntensity(0, 8);
    // Очистка матриці
    LC1.clearDisplay(0);
}
```


Матрична індикація. Dot Matrix

Функція `shutdown()` – це функція відключення режиму енергозбереження, світлодіоди споживають багато енергії, це може виявитися критичним, якщо пристрій працює від батарейок. Функція `shutdown()` може відключити матрицю коли потрібно буде перевести пристрій в енергозберігаючий режим, або включити. `LC1.shutdown(int address, bool set)`, де: `LC1` об'єкт класу `LedControl`, `int address` адреса пристрою на базі драйвера `MAX72xx`, `bool set` вказує режим роботи (`false` – вихід з режиму очікування та відображення даних, `true` – пристрій переходить в сплячий режим).

Функція `setIntensity()` встановлює яскравість світіння сегментів, або світлодіодів (дивлячись що підключено дисплей або матриця). Прототип виклику функції: `LC1.setIntensity(int address, int brightness)`, де: `LC1` об'єкт класу `LedControl`, `int address` адреса пристрою на базі драйвера `MAX72xx`, `int brightness` встановлює яскравість світіння сегментів, або світлодіодів. Може приймати значення від 0 до 15 (0 мінімальний рівень, 15 максимальний рівень).

Функція `clearDisplay()` очищає матрицю, за вказаною адресою. Прототип виклику функції: `LC1.clearDisplay(int address)`, де: `LC1` об'єкт класу `LedControl`, `int address` адреса пристрою на базі драйвера `MAX72xx`.

Управління LED матрицями 8×8 здійснюється трьома функціями: `setLed()`, `setRow()`, `setColumn()`.

Матрична індикація. Dot Matrix

Функція `shutdown()` – це функція відключення режиму енергозбереження, світлодіоди споживають багато енергії, це може виявитися критичним, якщо пристрій працює від батарейок. Функція `shutdown()` може відключити матрицю коли потрібно буде перевести пристрій в енергозберігаючий режим, або включити. `LC1.shutdown(int address, bool set)`, де: *LC1* об'єкт класу `LedControl`, *int address* адреса пристрою на базі драйвера `MAX72xx`, *bool set* вказує режим роботи (*false* – вихід з режиму очікування та відображення даних, *true* – пристрій переходить в сплячий режим).

Функція `setIntensity()` встановлює яскравість світіння сегментів, або світлодіодів (дивлячись що підключено дисплей або матриця). Прототип виклику функції: `LC1.setIntensity(int address, int brightness)`, де: *LC1* об'єкт класу `LedControl`, *int address* адреса пристрою на базі драйвера `MAX72xx`, *int brightness* встановлює яскравість світіння сегментів, або світлодіодів. Може приймати значення від 0 до 15 (0 мінімальний рівень, 15 максимальний рівень).

Функція `clearDisplay()` очищає матрицю, за вказаною адресою. Прототип виклику функції: `LC1.clearDisplay(int address)`, де: *LC1* об'єкт класу `LedControl`, *int address* адреса пристрою на базі драйвера `MAX72xx`.

Управління LED матрицями 8×8 здійснюється трьома функціями: `setLed()`, `setRow()`, `setColumn()`.

Матрична індикація. Dot Matrix

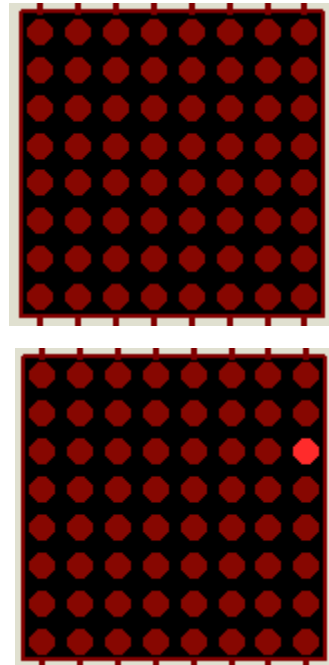
`setLed()` управляє кожним світлодіодом на матриці індивідуально. `LC1.setLed (int address, int row, int column, boolean state)`, де: LC1 об'єкт класу LedControl, int address адреса пристрою на базі драйвера MAX72xx, int row є ряд світлодіодів LED матриці 8×8, int column є стовпець світлодіодів LED матриці 8×8, boolean state це стан світлодіода (true – включений, false – виключений).

```
#include "LedControl.h"
#include "binary.h"

//DIN pin 12, CLK pin 11, CS pin 10
LedControl LC=LedControl(12,11,10,1);

void setup() {
  LC.shutdown(0, false);
  LC.setIntensity(0, 8);
  LC.clearDisplay(0);
}

void loop() {
  LC.setLed(0, 2, 7, true);
  delay(500);
  LC.setLed(0, 2, 7, false);
  delay(500);
}
```



Матрична індикація. Dot Matrix

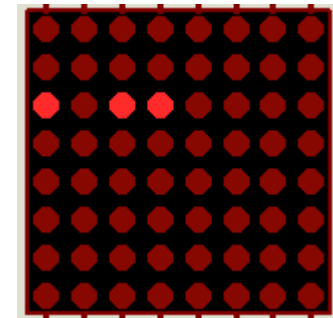
Для включення ряду світлодіодів на LED матриці 8×8 застосовується функція `setRow ()`. Прототип виклику функції: `LC.setRow (int address, int row, byte value)`, де: LC1 об'єкт класу LedControl, int address адреса пристрою на базі драйвера MAX72xx, int row є ряд світлодіодів LED матриці 8×8, byte value – змінна типу byte, значення якої буде включати певні світлодіоди в ряду LED матриці 8×8.

```
#include "LedControl.h"
#include "binary.h"

// DIN 12, CLK 11, CS 10
LedControl LC=LedControl(12,11,10,1);

void setup() {
  LC.shutdown(0, false);
  LC.setIntensity(0, 8);
  LC.clearDisplay(0);
}

void loop() {
  LC.setRow(0, 2, B10110000);
}
```



Матрична індикація. Dot Matrix

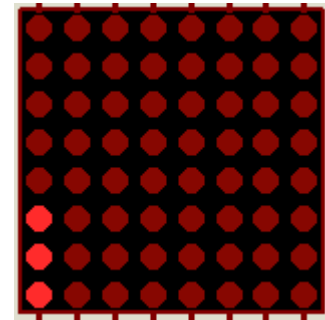
Для включення стовців світлодіодів на LED матриці 8×8 застосовується функція `setColumn ()`. Прототип виклику функції: `LC.setColumn (int address, int column, byte value)`, де: LC1 об'єкт класу `LedControl`, `int address` адреса пристрою на базі драйвера `MAX72xx`, `int column` є стовець світлодіодів LED матриці 8×8, `byte value` – змінна типу `byte`, значення якої буде включати певні світлодіоди в ствпці LED матриці 8×8

```
#include "LedControl.h"
#include "binary.h"

//DIN 12, CLK 11, CS 10
LedControl LC=LedControl(12,11,10,1);

void setup() {
  LC.shutdown(0, false);
  LC.setIntensity(0, 8);
  LC.clearDisplay(0);
}

void loop() {
  LC.setColumn (0, 0, B00000111);
}
```



Матрична індикація. Dot Matrix

Програма Matrix_One виводить зображення цифри 1 на матричному дисплеї 8×8.

```
#include "LedControl.h"
#include "binary.h"

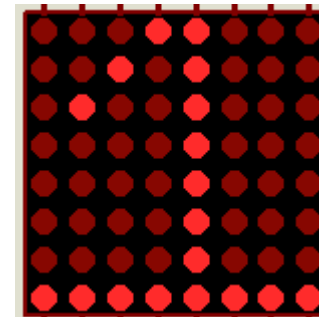
//DIN 12, CLK 11, CS 10
LedControl lc=LedControl(12,11,10,1);
// delay time between faces
unsigned long delaytime=1000;

byte one[8]= {
  B00011000,
  B00101000,
  B01001000,
  B00001000,
  B00001000,
  B00001000,
  B00001000,
  B11111111};

void setup() {
  lc.shutdown(0,false);
  // Set brightness to a medium value
  lc.setIntensity(0,8);
  lc.clearDisplay(0);
}
```

```
void drawDigital() {
  lc.setRow(0,0,one[0]);
  lc.setRow(0,1,one[1]);
  lc.setRow(0,2,one[2]);
  lc.setRow(0,3,one[3]);
  lc.setRow(0,4,one[4]);
  lc.setRow(0,5,one[5]);
  lc.setRow(0,6,one[6]);
  lc.setRow(0,7,one[7]);
  delay(delaytime);
}

void loop() {
  drawDigital();
}
```

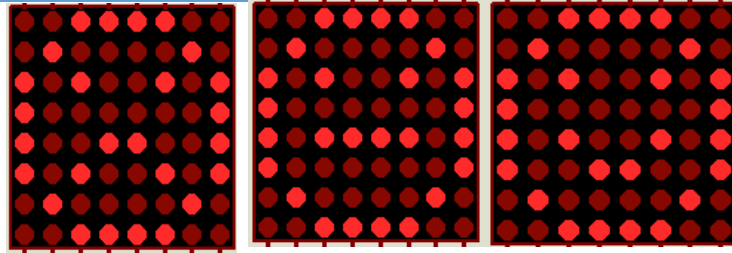


Матрична індикація. Dot Matrix

Програма Matrix_Smile на матричному дисплеї 8×8 виводиться по черзі смайлики (sad face, neutral face, happy face) у вигляді анімації.

```
#include "LedControl.h"
#include "binary.h"
LedControl lc=LedControl(12,11,10,1);
unsigned long delaytime=1000;

// happy face
byte hf[8]= {B00111100,B01000010,B10100101,B10000001,B10100101,B10011001,B01000010,B00111100};
// neutral face
byte nf[8]={B00111100, B01000010,B10100101,B10000001,B10111101,B10000001,B01000010,B00111100};
// sad face
byte sf[8]= {B00111100,B01000010,B10100101,B10000001,B10011001,B10100101,B01000010,B00111100};
```



```
void setup() {
  lc.shutdown(0, false);
  // Set brightness to a
  lc.setIntensity(0, 8);
  // Clear the display
  lc.clearDisplay(0);
}
```


```
void loop() {
  drawFaces();
}
```

```
void drawFaces() {
  // Display sad face
  lc.setRow(0, 0, sf[0]);
  lc.setRow(0, 1, sf[1]);
  lc.setRow(0, 2, sf[2]);
  lc.setRow(0, 3, sf[3]);
  lc.setRow(0, 4, sf[4]);
  lc.setRow(0, 5, sf[5]);
  lc.setRow(0, 6, sf[6]);
  lc.setRow(0, 7, sf[7]);
  delay(delaytime);
}
```

```
// Display neutral face
lc.setRow(0, 0, nf[0]);
lc.setRow(0, 1, nf[1]);
lc.setRow(0, 2, nf[2]);
lc.setRow(0, 3, nf[3]);
lc.setRow(0, 4, nf[4]);
lc.setRow(0, 5, nf[5]);
lc.setRow(0, 6, nf[6]);
lc.setRow(0, 7, nf[7]);
delay(delaytime);
```

```
// Display happy face
lc.setRow(0, 0, hf[0]);
lc.setRow(0, 1, hf[1]);
lc.setRow(0, 2, hf[2]);
lc.setRow(0, 3, hf[3]);
lc.setRow(0, 4, hf[4]);
lc.setRow(0, 5, hf[5]);
lc.setRow(0, 6, hf[6]);
lc.setRow(0, 7, hf[7]);
delay(delaytime);
}
```

Домашнє завдання

- 1) Реалізувати програму, яка передбачає використання світлодіодної матриці. Засвічуються стовпці матриці справа-наліво, потім – рядки згори-донизу, створюючи ефект роботи сканера.
- 2) Реалізувати програму, яка передбачає використання світлодіодної матриці. Засвічується крапка, яка «бігає» по контуру матриці.
- 3) Реалізувати програму, яка виводить на матрицю текст «Я  ВНАУ».