

Практична робота 6. Робота з матричним світлодіодним індикатором

Мета: ознайомитись з принципом роботи матричного світлодіодного індикатора 8×8 точок, драйвера керування роботою світлодіодною матрицею MAX7219; навчитись програмувати виведення інформації через інтерфейс SPI на матричний світлодіодний індикатор з використанням драйвера керування MAX7219.

Завдання: написати програму представлення символів та зображень на матричному світлодіодному індикаторі 8×8 .

Обладнання: лабораторний макет/віртуальний стенд «Arduino Learner Kit»; USB – кабель; провідники-з'єднувачі.

Загальні відомості

Матричні світлодіодні індикатори (MCI) використовуються для відображення алфавітно-цифрової інформації. Кожен з таких MCI, виконаний у вигляді інтегральної мікросхеми, є матрицею світлодіодів розмірністю $m \times n$, де n - число стовпчиків, m - число рядків матриці. Найбільшого поширення набули MCI з розмірністю матриці 7×5 , 9×7 , 8×8 (рис. 3.5).

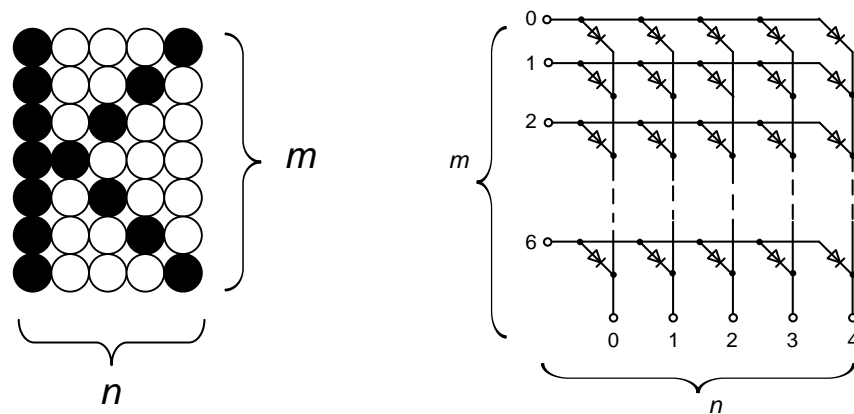


Рисунок 3.5 – Загальний вигляд та схема матричного індикатора

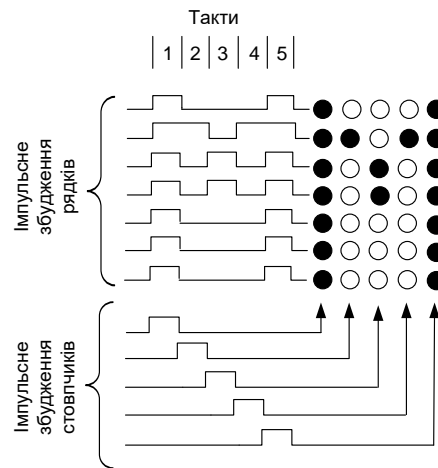


Рисунок 3.6 – Часова діаграма формування літери М

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 3.6.

У лабораторному макеті для представлення символів та зображень використовується світлодіодна матриця 8×8 точок (64 світлодіоди). Вона має 16 виводів для управління рядками та стовпцями масиву. Спеціалізована мікросхема-драйвер MAX7219 (рис. 1.7) 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. Живлення на мікросхему MAX7219 подається через перемикач SW1 (контакт 4 перевести у положення «On»). Виводи DIN (вхід даних), CLK (вхід для синхроімпульсів) і CS (вхід вибору мікросхеми) через X1 з'єднують з цифровими портами плати Arduino.

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"
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 . 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);
}

```

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: *setRow()*, *setColumn()*.

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).

Приклад скетчу:

```
#include "LedControl.h"
LedControl LC = LedControl(12, 11, 10, 5);
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);
}
```

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.

Приклад скетчу:

```
#include "LedControl.h"
LedControl LC = LedControl(12, 11, 10, 5);
void setup() {
    LC.shutdown(0, false);
    LC.setIntensity(0, 8);
    LC.clearDisplay(0);
}
void loop() {
    LC.setRow(0, 2, B10110000);
}
```

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.

Приклад скетчу:

```
#include "LedControl.h"
LedControl LC = LedControl(12, 11, 10, 5);
void setup() {
    LC.shutdown(0, false);
    LC.setIntensity(0, 8);
    LC.clearDisplay(0);
}
void loop() {
    LC.setColumn (0, 0, B00000111);
}
```

Хід виконання роботи

1. Підключити схему до комп'ютера через USB порт плати Arduino та/або запустити віртуальний стенд у середовищі Proteus 8.
2. Завантажити програму *Matrix_One* (додаток М) до лабораторного макета / віртуального стенду, попередньо виконати з'єднання MAX7219 та Arduino у відповідності до програми. Встановити бібліотеку *LedControl.h* в середовище Arduino IDE (див. п. 1.5). Дослідити роботу програми.
3. Завантажити програму *Matrix_Smile* (додаток М) до лабораторного макета / віртуального стенду. Дослідити роботу програми.

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 числа від 0 до 9, з кожним натисненням тактової кнопки.

4. Реалізувати програму, яка виводить на матрицю текст «Я ♥ ВНАУ».

5. Реалізувати програму, яка передбачає використання світлодіодної матриці. У центрі світлодіодної матриці засвічується крапка. За допомогою клавіш S0–S3 можна змінювати положення крапки вліво, вправо, вгору та вниз.

Підготувати звіт згідно ДСТУ 3008-95 (лістинг програми, висновки, перелік посилань).

Контрольні питання

1. Поясніть особливості обміну інформацією між мікросхемами інтерфейсом SPI?
2. Наведіть схему підключення 4 матричних світлодіодних індикаторів з використанням одного драйвера MAX7219. Поясніть як реалізувати програму, щоб текст рухався. Які функції бібліотеки LedControl.h будуть використовуватись?
3. Наведіть схему підключення 8 розрядного семисегментного індикатора з використанням одного драйвера MAX7219. Поясніть як реалізувати програму, щоб на індикаторі відображалась поточна дата та час. Які функції бібліотеки LedControl.h будуть використовуватись?
4. Які є візуальні редактори для створення анімаційних ефектів? Поясніть як ними користуватись.