

Поліщук Ю. К.
студент 33 групи фізико-математичного факультету
Науковий керівник: Вакалюк Т. А.,
кандидат педагогічних наук, доцент,
доцент кафедри прикладної математики та інформатики,
Житомирський державний університет імені Івана Франка

АЛГОРИТМ ВИВЕДЕННЯ ГРАФІЧНИХ ДАНИХ НА ДИСПЛЕЙ

У сучасному світі часто зустрічаються дисплеї, зокрема, у телевізорах, мобільних телефонах, годинниках та іншій техніці. Дисплей перетворює електричні сигнали у графічне зображення, що зрозуміле людині. Користуючись пристроями з дисплеями, користувач навіть не замислюється про принципи їх роботи.

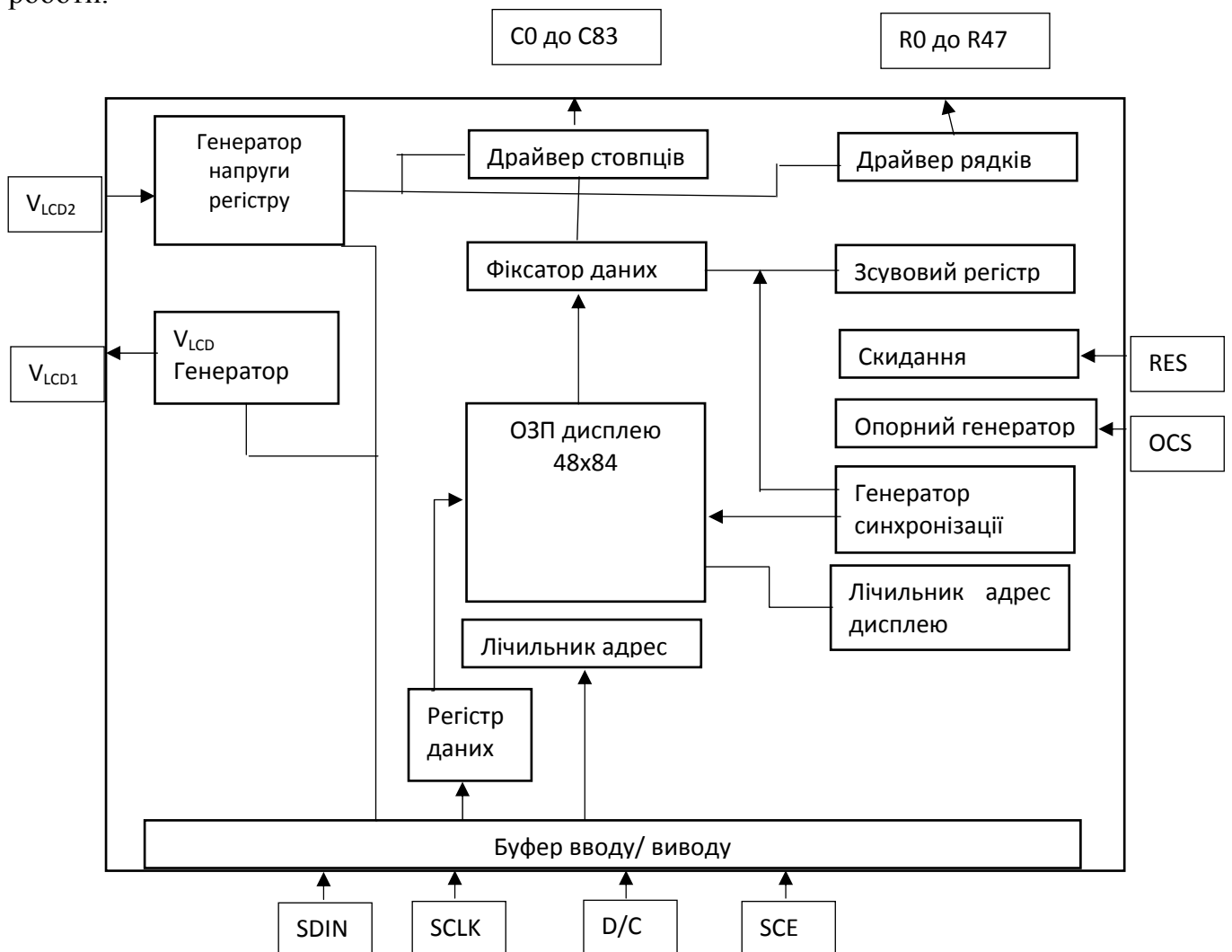


Рис. 1. Схема дисплею.

Дисплеї потрібні у всіх пристроях, де є взаємодія з користувачем. На дисплей можливо вивести меню, прогрес-бар, цифри, букви, символи, растрові зображення тощо. Перевага дисплеїв перед семисегментними індикаторами чи інших пристроїв виведення даних у тому, що є можливість виводу своїх

символів, піктограм, значків, шрифтів, та навіть арабських символів. Саме тому, метою даного дослідження є аналіз алгоритму виведення графічних даних на дисплей. У даній роботі для прикладу використано дисплей з-під мобільного телефону Nokia 3310, побудованого на мікроконтролері Philips PCD8544. Даний дисплей має розширення 48x84 пікселів.

Схема дисплею наведена на рисунку 1.

При цьому скорочення, що використані у рисунку 1, наведені у таблиці 1.

Таблиця 1

Скорочення	Значення
C0 до C83	Цифрові виходи до рядків ЖК дисплею
R0 до R47	Цифрові виходи до стовпців ЖК дисплею
V_{LCD1} , V_{LCD2}	Живлення LCD
SDIN (serial data input)	Вхід даних
SCLK (serial clock input)	Вхід такту
D/C (Data/Command)	Вибір режиму(дані/ команди)
SCE (Chip enable)	Увімкнення/ вимкнення чіпу
OCS	Вхід опорної частоти або опорного живлення
RES	Скидання

У таблиці 2 наведено байти, що потрібно передавати на дисплей для виконання певних інструкцій.

Таблиця 2

Інструкції	D/C Дані кома- нди	Байти команд								Опис
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
(H=1 або 0) базовий набір інструкцій або розширений										
NOP	0	0	0	0	0	0	0	0	0	Без операцій
Набір функцій	0	0	0	1	0	0	PD	V	H	PD- енерго-зберігаючий режим V- якщо 1 то вертикальна адресація інакше горизонтальна
Запис даних	1	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	Запис даних до ОЗП дисплею
(H=0) при базовому наборі інструкцій										
Заброньовано	0	0	0	0	0	0	1	X	X	Не використовується
Контроль дисплеєм	0	0	0	0	0	1	D	0	E	D і E 00-відобра-

										зити пустим 10-нормаль- ний режим 01-всі пік- селі вклю- чені (чорний екран) 11- Режим негативу
Заброньовано	0	0	0	0	1	X	X	X	X	Не викорис- товується
Встановити адресу ОЗП по осі Y	0	0	1	0	0	0	Y ₂	Y ₁	Y ₀	Адреса 0<=Y<=5
Встановити адресу ОЗП по осі X	0	1	X ₆	X ₅	X ₄	X ₃	X ₂	X ₁	X ₀	Адреса 0<=X<=83
(H=1) при розширеному наборі інструкцій										
Заброньовано	0	0	0	0	0	0	0	0	1	Не викорис- товується
	0	0	0	0	0	0	0	1	X	
Контроль температури	0	0	0	0	0	0	1	TC ₁	TC ₀	Встановлен- ня коефіцієн- та темпе- ратури
Заброньовано	0	0	0	0	0	1	X	X	X	Не вико рис- товується
Здвигова система	0	0	0	0	1	0	BS ₀	BS ₁	BS ₂	Встановлен- ня зсувної системи (BS _x)
Заброньовано	0	0	1	X	X	X	X	X	X	Не вико рис- товується
Встановлення V _{op}	0	1	V _{op6}	V _{op5}	V _{op4}	V _{op3}	V _{op2}	V _{op1}	V _{op0}	Встановлен- ня напруги на мікрокон- троллері

Відповідно до встановленої напруги V_{op} змінюється контраст дисплея. Температурний коефіцієнт надає змогу регулювати контрастність дисплею відповідно до температури, тому що рідкі кристали при зміні температури змінюють свої властивості.

На рисунку 2 зображена схема підключення дисплею до Arduino nano. Arduino – платформа для програмування мікроконтролерів (на основі Atmel). Дисплей підключається по інтерфейсу SPI (Serial Peripheral Interface). Програма написана в середовищі програмування Arduino IDE.

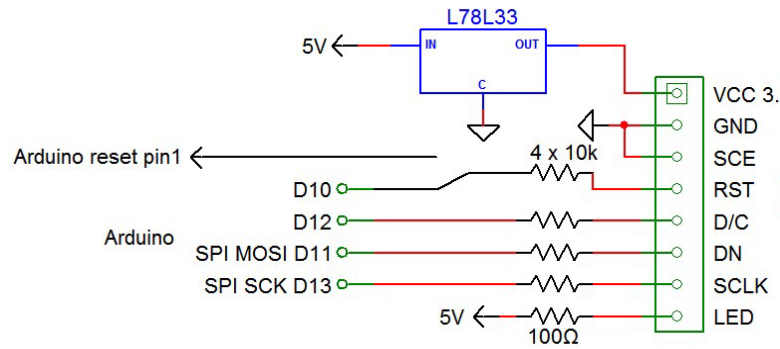


Рис. 1. Схема підключення дисплею до Arduino nano.

Приклад коду заповнення дисплею чорним кольором у даному середовищі програмування наведений нижче:

```
#define RST 12
#define CE 11
#define DC 10
#define DIN 9
#define CLK 8
void LcdWriteCmd(byte cmd)
{
digitalWrite(DC, LOW);
digitalWrite(CE, LOW);
shiftOut(DIN, CLK, MSBFIRST, cmd);
digitalWrite(CE, HIGH);
}
void LcdWriteData(byte cmd)
{
digitalWrite(CE, LOW);
shiftOut(DIN, CLK, MSBFIRST, cmd);
digitalWrite(CE, HIGH);
}
void LCDisplay(byte[] mas){
for(int i=0;i<mas.length;i++)
LcdWriteData(mas[i]);
}
void setup()
{
pinMode(RST, OUTPUT);// режим піну
pinMode(CE, OUTPUT);
pinMode(DC, OUTPUT);
pinMode(DIN, OUTPUT);
pinMode(CLK, OUTPUT);
digitalWrite(RST, LOW);//запис початкового значення
digitalWrite(RST, HIGH);
LcdWriteCmd(0x21); // Розширенні команди
LcdWriteCmd(0xB8); // Встановлення контрасту
LcdWriteCmd(0x04); // встановлення температурного коефіцієнта
LcdWriteCmd(0x14); LcdWriteCmd(0x20); // Базові команди
LcdWriteCmd(0x09); // Режим негативу
}
void loop()//головний цикл
{
}
```

Отже, у даній роботі було досліджено алгоритм передачі графічних даних між мікроконтролером і дисплеєм, а також було створено найпростішу програму для зафарбовування всіх пікселів дисплею. У подальшому планується розглянути практичну реалізацію світлодіодного дисплея на зсувних регістрах.

Список використаних джерел та літератури

1. PCD8544 48X84 pixels matrix LCD controller/driver // Philips Electronics N.V. – 1999.