**4 Лабораторна робота №4**

**СТАТИЧНИЙ семисегментнИЙ СВІТЛОДІОДНИЙ ИНДИКАТОР**

**Мета роботи**

Ознайомитися із функціональними можливостями і внутрішньою структурою відлагоджувального стенду EV8031/AVR. Вивчити внутрішню організацію рідкокристалічного індикатора, навчитися відображати інформацію на ньому.

**4.1 Короткі теоретичні відомості**

В даний час для відображення інформації широко використовуються алфавітно-цифрові РКІ-модулі. Такий модуль являє собою один або кілька рядків, кожен з яких складається із декількох знакомісць. Кожне знакомісце, в свою чергу, представляє собою масив точок (як правило, 5х7). У даній лабораторній роботі використовується РКІ-модуль, який містить 2 рядки по 16 символів (знакомісць).

Алфавітно-цифровий РКІ-модуль має наступні зовнішні контакти:

– Vdd – живлення модуля;

– Vss – спільний контакт (земля);

– V0 – керування контрастом зображення;

– RS – вибір регістру команд або даних;

– E – строб запису/читання;

– R/!W – сигнал читання/запису;

– DB7 ... DB 0 – двонаправлені лінії обміну даними (команди/символи);

– A – вивод аноду підсвітки модуля;

– K – вивод катоду підсвітки модуля.

Модуль, що використовується в лабораторній роботі, має світлодіодну підсвітку. Вивод R/!W наглухо заземлений, тому модуль доступний тільки для операцій запису команд/даних, читати з нього не можна.

Часову діаграму процедури запису команд/даних в РКІ-модуль показано на рисунку 4.1.

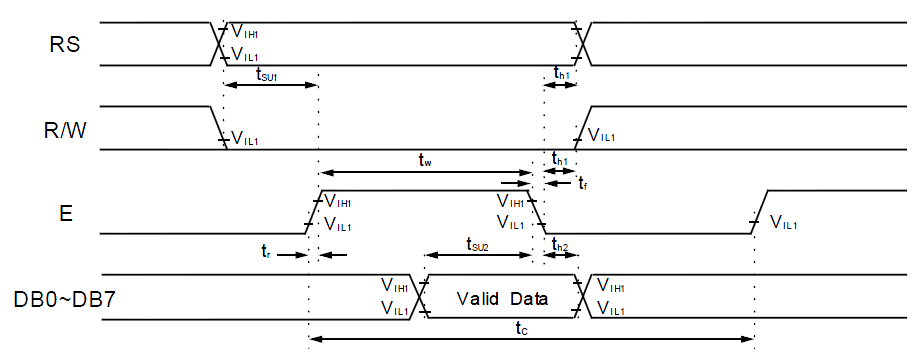


Рисунок 4.1 – Часова діаграма запису інформації в РКІ-модуль

У лабораторному стенді задача формування наведеної часової діаграми виконується системним контролером автоматично. Для підтримки інтерфейсу з РКІ-модулем в системному контролері передбачено 2 комірки із адресами 0x8004 і 0x8005. При записі байта в комірку із адресою 0x8004 він надійде в регістр команд РКІ-модуля, а з адресою 0x8005 – в регістр даних.

Спрощену внутрішню структуру РКІ-модуля показано на рисунку 4.2.

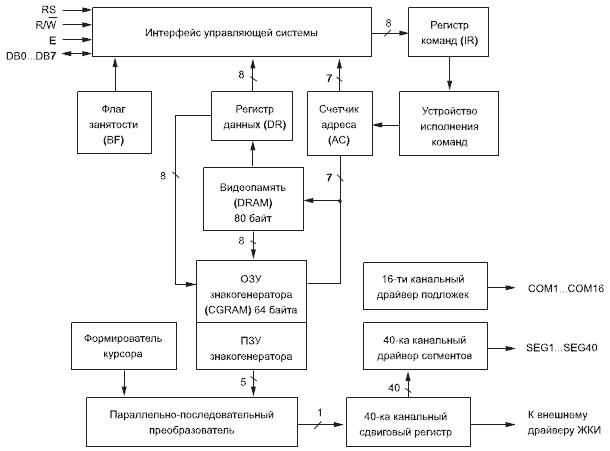


Рисунок 4.2 – Спрощена внутрішня структура РКІ-модуля

РКІ-модуль, що використовується в стенді, має 80 комірок відеопам’яті DDRAM (для зберігання символьних повідомлень). Для першого рядка модуля зарезервовано 40 комірок пам’яті DDRAM із адресами від 00h до 27h. Для другого рядка модуля також зарезервовано 40 комірок із адресами від 40h до 67h (таким чином, в адресах комірок пам’яті між першим і другим рядком є “дірка”). При послідовному запису інформації в DDRAM внутрішній контролер РКІ-модуля автоматично переходить із останньої комірки першого рядка (27h) на першу комірку другого рядка (40h), якщо ввімкнено відповідний режим автоматичного збільшення адрес. Слід пам’ятати, що на екрані РКІ-модуля відображається вміст лише 16-ти комірок для першого рядка і 16-ти комірок для другого рядка. При виконанні команди зсуву екрану вміст комірок DDRAM не змінюється!, змінюються лише вказівники початку видимих частин рядків. Наприклад, після ініціалізації модуля, ми бачимо в першому його рядку комірки DDRAM із адресами від 00h до 0Fh, а в другому – із адресами від 40h до 4Fh (рисунок 4.3).

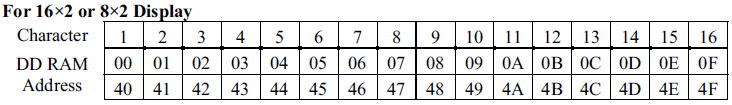


Рисунок 4.3 – Взаємна відповідність знакомісць і комірок пам’яті DDRAM для РКІ-модуля після скидання

Після виконання команди зсуву екрану вліво в першому рядку буде відображатися вміст комірок з адресами від 01h до 10h, а в другому – від 41h до 50h. Аналогічні дії виконуються і при зсуві екрану вправо.

РКІ-модуль також містить ПЗП знакогенератора (CGROM) і ОЗП знакогенератора (CGRAM). ПЗУ містить наперед встановлені виробником символи для відображення модулем. Для виведення такого наперед встановленого символу достатньо записати в відповідну комірку відеопам’яті DDRAM його код (або адресу в пам’яті CGROM). Кодову таблицю РКІ-модуля, що містить кириличні символи, показано на рисунку 4.4.

ОЗП знакогенератора (CGRAM) може використовуватися для створення символів користувача.

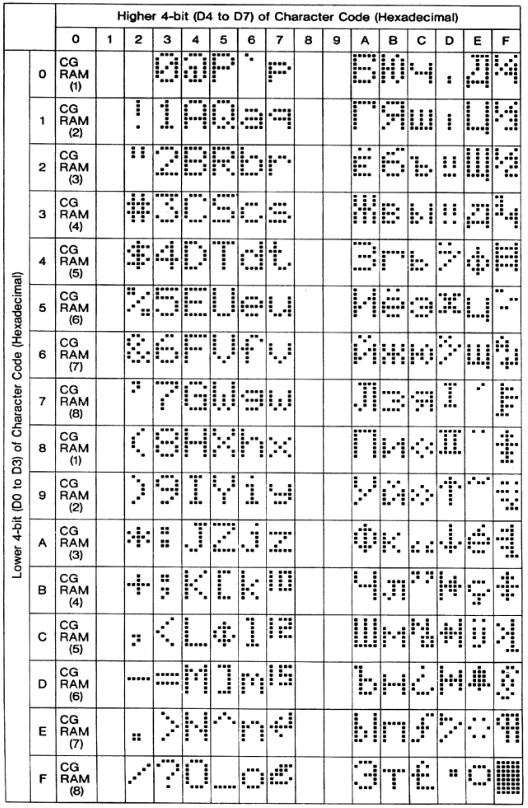


Рисунок 4.4 – Кодова таблиця РКІ-модуля

Для управління РКІ-модулем і його ініціалізації використовуються спеціальні команди. Таблицю таких команд показано на рисунку 4.5.

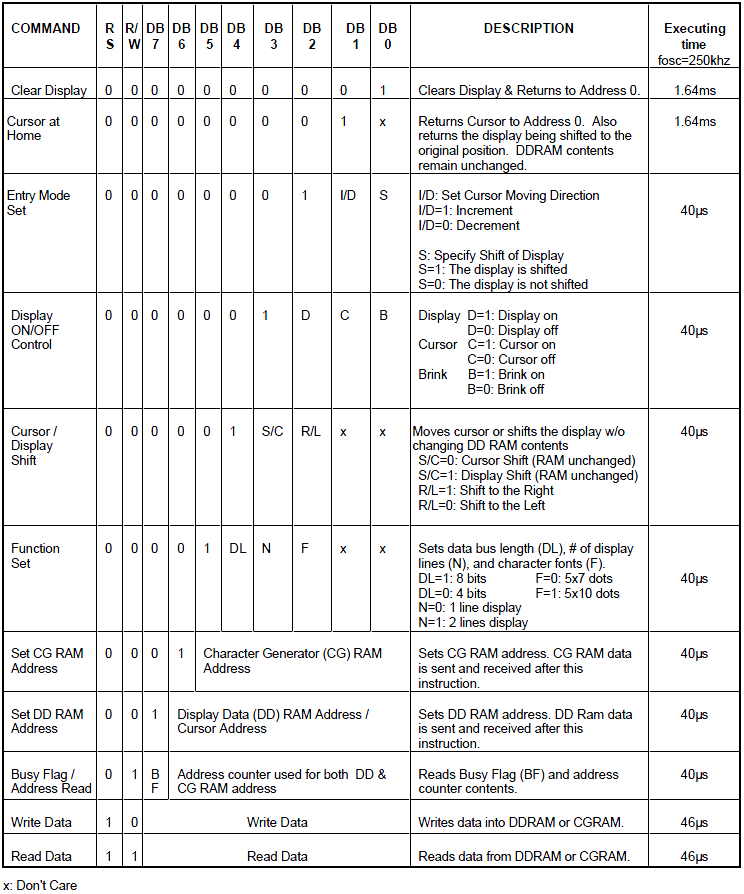


Рисунок 4.5 – Таблиця команд РКІ-модуля

Слід звернути увагу, що після подачі чергової команди або байту даних необхідно витримати паузу не менше, ніж зазначено в таблиці на рисунку 4.5. Протягом цього часу контролер РКІ-модуля обробляє попередній прийнятий байт і може некоректно відреагувати на байт, який знову надійшов. Час виконання команд в таблиці на рисунку 4.5 зазначено для випадку, коли внутрішня тактова частота роботи контролера РКІ-модуля становить 250 кГц. На практиці зустрічаються РКІ-модулі, що працюють на тактових частотах або 250 кГц, або 270 кГц. При частоті роботи в 270 кГц затримки на виконання команд/обробку даних, зазначені в таблиці на рисунку 4.5, будуть трохи меншими. На практиці, з метою забезпечення сумісності з будь-якими модулями, рекомендується використовувати затримки, які дещо перевищують ті, які наведено в таблиці на рисунку 4.5.

При ввімкненні живлення програмісту необхідно здійснити ініціалізацію РКІ-модуля. У лабораторному стенді використовується 8-бітний протокол обміну між мікроконтролером і РКІ-модулем. Ініціалізація РКІ-модуля в цьому випадку здійснюється згідно з алгоритмом, показаним на рисунку 4.6. Часові затримки на виконання алгоритму ініціалізації (на рисунку 4.6) вказано для РКІ-модуля із внутрішньою тактовою частотою у 270 кГц. При використанні більш “повільного” РКІ-модуля, наприклад, з частотою 250 кГц, всі зазначені затримки слід множити на коефіцієнт 270/250.

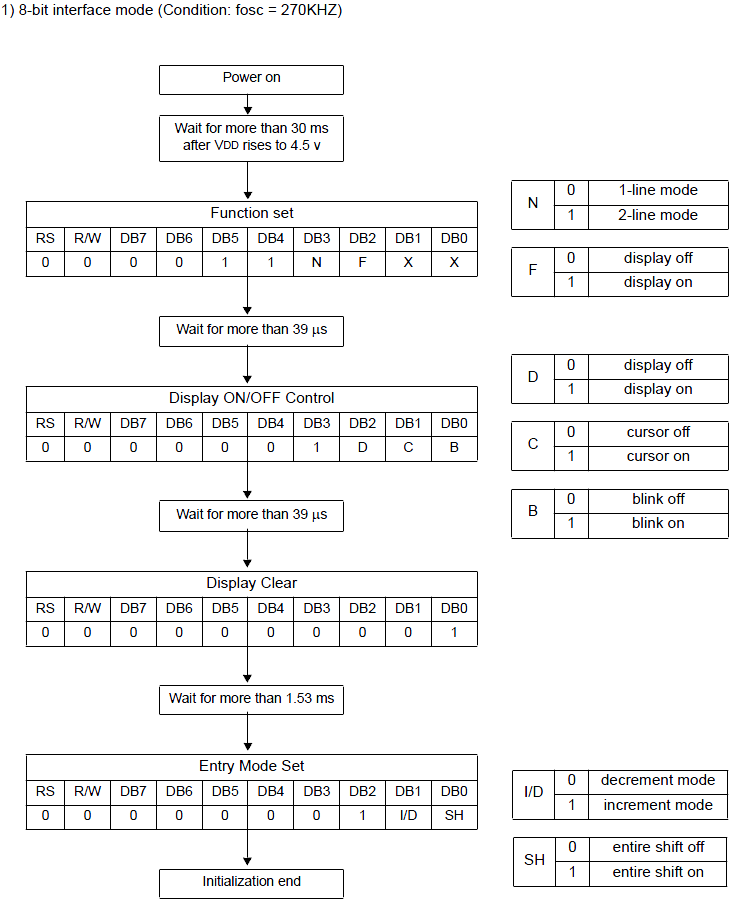


Рисунок 4.6 – Алгоритм ініціалізації РКІ-модуля

Нижче наведено приклад програми мовою асемблер, яка здійснює відображення інформаційного повідомлення на всіх знакомісцях першого і другого рядка рідкокристалічного індикатора (час виведення одного символу становить 200 мс). Після виведення повідомлення виробляється циклічний зсув вліво вмісту індикатора із періодом 1 с. Часові затримки організовуються програмним способом.

**4.3 Результати виконання завдання**

Завдання для лабораторної роботи зображено в таблиці 4.3.1

Таблиця 4.3.1 – Завдання

|  |  |
| --- | --- |
| № | Опис завдання |
| 8 | На рідкокристалічному індикаторі (в довільній позиції) реалізувати відображення секундоміра в форматі 59 → 00. Часові затримки організовуються програмно. |

Код програми наведено в лістингу 4.3.2

Лістинг 4.3.2

#define F\_CPU 7372800L //задаємо частоту кварцу (7,3728 МГц)

#include <avr/io.h>

#include <avr/iom8515.h>

#include <util/delay.h>

//задаємо константу - адресу регістра команд в РКІ

#define lcd\_comm 0x8004

//задаємо константу - адресу регістра даних в РКІ

#define lcd\_data 0x8005

int main (void) {

//початкова ініціалізація міконтролера

//дозволяємо роботу із зовнішньою пам'яттю (звернення до системного контролеру)

MCUCR = 1 << SRE;

//відключення живлення аналогового компаратора

ACSR = 1 << ACD;

//встановлюємо вказівник на адресу регістра команд РКІ

volatile unsigned char\* c = (unsigned char \*) lcd\_comm;

//встановлюємо вказівник на адресу регістра даних РКІ

volatile unsigned char\* d = (unsigned char \*) lcd\_data;

volatile unsigned char\* A = (unsigned char \*) 0x0090;

\_delay\_ms (1000); //встановлюємо затримку по ввімкненню живлення на 1 с

unsigned int counter = 0x59;

//початкова ініціалізація РКІ

\*c = 0x38; //Function Set

\_delay\_ms (10); //викликаємо затримку на 10 мс

\*c = 0x0E; //Display ON/OFF Control

\_delay\_ms (10); //викликаємо затримку на 10 мс

\*c = 0x01; //Display clear

\_delay\_ms (10); //викликаємо затримку на 10 мс

\*c = 0x06; //Entry Mode Set

\_delay\_ms (10); //викликаємо затримку на 10 мс

//Виводимо перший (верхній) рядок повідомлення

\*A = 0x30;//0

\*(A+1) = 0x31;//1

\*(A+2) = 0x32;//2

\*(A+3) = 0x33;//3

\*(A+4) = 0x34;//4

\*(A+5) = 0x35;//5

\*(A+6) = 0x36;//6

\*(A+7) = 0x37;//7

\*(A+8) = 0x38;//8

\*(A+9) = 0x39;//9

while (1) { //нескінченний цикл

\*d = \*(A+(counter / 10)); //код символу

\_delay\_ms (100); //викликаємо затримку на 200 мс

\*d = \*(A+(counter % 10)); //код символу

\_delay\_ms (100); //викликаємо затримку на 200 мс

\_delay\_ms (790); //викликаємо затримку на 790 мс

\*c = 0x01; //Display clear

\_delay\_ms (10); //викликаємо затримку на 10 мс

if (counter !=00){

counter = counter - 0x01;}

}

return 0;

}

**ВИСНОВКИ**

Ознайомилися з функціональними можливостями, внутрішньою структурою відлагоджувального стенду EV8031/AVR. Вивчили внутрішню організацію рідкокристалічного індикатора, навчилися відображати інформацію на ньому.