

Елементи індикації. Кнопки та датчики. Оптичні датчики

[1 Елементи індикації](#)

[2 Кнопки та датчики](#)

[3 Оптичні датчики](#)

1 Елементи індикації

1.1 Пристрої індикації

Практично кожен мікропроцесорний пристрій (МПП) містить елементи індикації. Як індикатори в даний час найчастіше застосовуються світлодіоди. На ринку є величезний вибір світлодіодів, найрізноманітніших видів і розмірів.

У МПП світлові індикатори можуть служити для відображення різних режимів роботи: попередження про критичні ситуації, відображення ходу прийому керуючих сигналів тощо. Підключити одиночний світлодіодний індикатор до МК дуже просто. На рисунку 1.1 приведена схема підключення світлодіода безпосередньо до виводу порту МК.

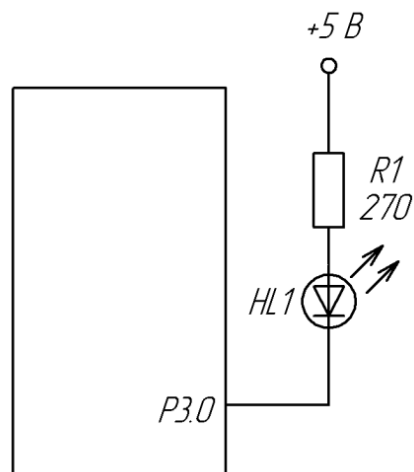


Рисунок 1.1– Схема підключення світлодіодного індикатора

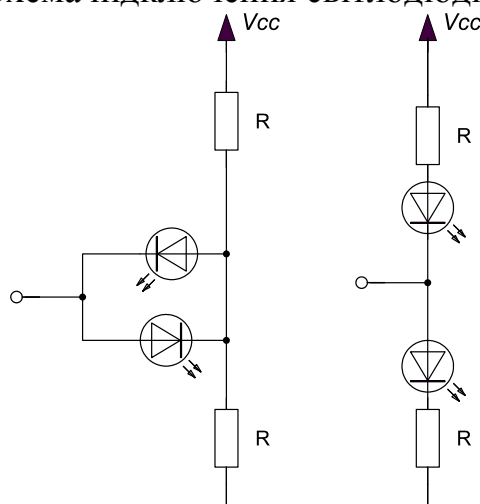


Рисунок 1.2 – Управління двома світлодіодами одним виходом МК

Усі вихідні каскади МК мають достатню навантажувальну здатність для того, щоб витримати підключення одного світлодіодного індикатора із споживаним струмом у робочому режимі не більше 20 мА.

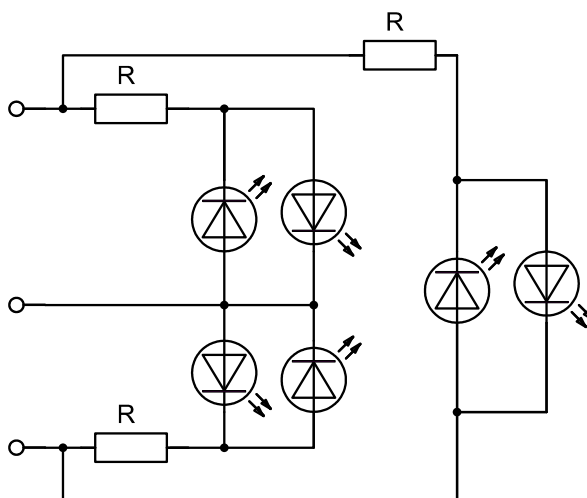


Рисунок 1.3 – Включення трьох пар зустрічно-паралельних світлодіодів

Для управління двома світлодіодами одним виходом у МК передбачено активні вихідні каскади, і для перемикання режиму роботи (введення або виведення) служить спеціальний регістр. Таким чином, сигнал кожного виходу будь-якого порту може мати 3 значення - «0», «1» і високоімпедансний («Z») стан. Це дозволяє керувати двома світлодіодами за допомогою одного виводу (рисунок 1.2).

При роботі порту в режимі виходу, залежно від стану «0» або «1» горить відповідно верхній або нижній світлодіод. При перемиканні в Z-стан, і при відповідному виборі резисторів струм через світлодіоди дуже малий і їх світіння майже непомітно.

Цікавим є включення трьох пар зустрічно-паралельних світлодіодів за схемою «зірка» (рисунок 1.3).

Відповідними сигналами МК можна домогтись світіння будь-якого з шести світлодіодів, а також їх комбінацій. Використовуючи динамічну індикацію, можна отримати будь-які комбінації.

1.2 Організація динамічної індикації

Дуже часто МК використовується не тільки для управління роботою конструкції, але і для того, щоб повідомити що-небудь користувачеві і/або отримати від нього які-небудь вказівки про подальшу роботу. Наприклад, електронний годинник, крім власне відліків часу, повинен його ще відображати, а також дозволяти змінювати покази (встановлювати точний час). Якщо вся "інформація" зводиться до мигання парою світлодіодів, яких-небудь спеціальних зусиль по відображенню інформації з боку розробника конструкції не вимагається, але якщо таких світлодіодів виявляється два-три

десятки, тут вже потрібне застосування додаткових засобів - як апаратних, так і програмних. Як правило, в цьому випадку відображення інформації виконують у режимі динамічної індикації - це найбільш економічний за кількістю використаних ліній спосіб.

Для організації динамічної індикації застосовується матриця, що складається з ліній рядків і ліній стовпчиків (рисунок 1.4). На перетині стовпчика і рядка матриці розташований індикаторний елемент - світлодіод. Для того, щоб запалити той або інший елемент, необхідно подати на матрицю не один, як в звичайних індикаторах, а два сигнали: логічна 1 на відповідному рядку і логічний 0 на відповідному стовпчику матриці. Через односторонню провідність світлодіода кожна комбінація сигналів на входах рядків і стовпчиків однозначно включає рівно один індикаторний елемент.

Головна перевага динамічної індикації - невелике число ліній, що управляють: для матриці світлодіодів розміром $N \times N$ елементів потрібно всього $2N$ сигналів, що управляють. За таку економію, втім, доводиться платити - справа в тому, що при почерговому виведенні інформації на кожен світлодіод матриці його яскравість світіння, буде в N^2 разів нижче, ніж при безпосередньому виведенні інформації на один світлодіод, що "окремо стоїть". Тому в пристроях, що використовують динамічну індикацію, виведення інформації здійснюється не на кожен світлодіод окремо, а на один рядок або на один стовпчик цілком - у цьому випадку яскравість світіння світлодіодів падає тільки в N разів.

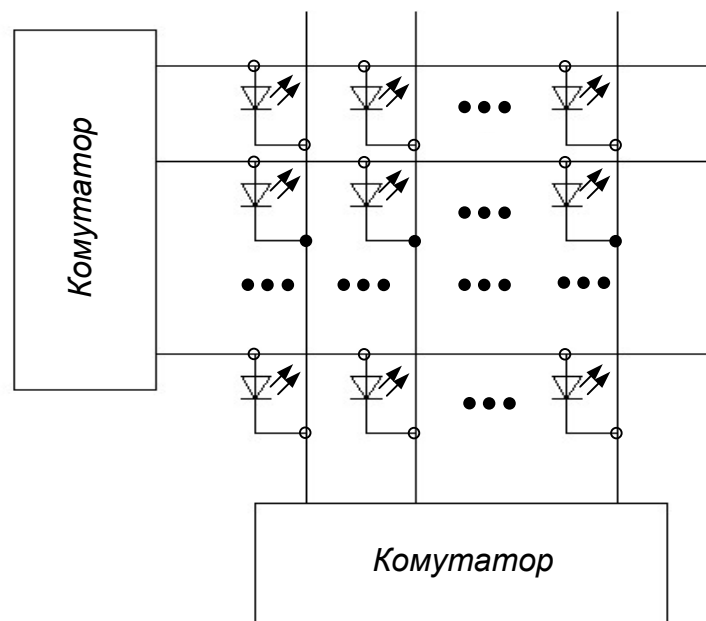


Рисунок 1.4 – Організація динамічної індикації

Розглянемо декілька варіантів реалізації динамічної індикації із застосуванням МК. Як індикаторний елемент передбачається застосування семисегментних індикаторів, але кожен такий індикатор з легкістю можна замінити і групою світлодіодів.

Схема реалізації динамічної індикація без додаткових елементів приведена на рисунку 1.5. До порту В МК підключені катооди всіх світлодіодів матриці, а до порту А – аноди кожного індикатора, що створюють матрицю. На лініях порту А організовується одиниця, що "біжить". На лінії порту В при кожному положенні одиниці, що біжить, виводиться семисегментний код того символу, який повинен горіти в даному знакомісті. Для індикаторів із загальним катодом замість одиниці, що біжить, використовується нуль, що біжить. Перевага такого способу індикації - у відсутності яких-небудь додаткових компонентів (окрім самих світлодіодних індикаторів), головний недолік – значна перевитрата ліній портів. Таке рішення для МК може забезпечити роботу не більше 5 семисегментних індикаторів одночасно, і при цьому МК стає "глухий і сліпий", оскільки жодної вільної лінії у нього не залишається. При використанні інших МК з великою кількістю ніжок – наприклад, АТТiny28, вказана проблема знімається.

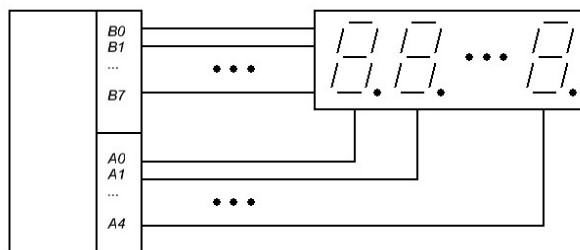


Рисунок 1.5 – Схема реалізації динамічної індикація без додаткових елементів

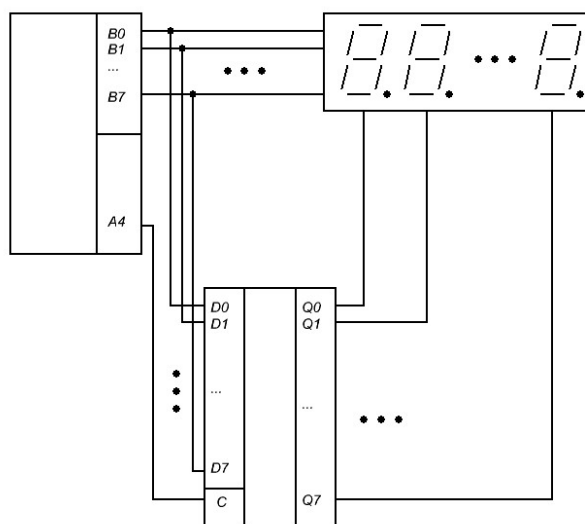


Рисунок 1.6 – Схема реалізації динамічної індикації з одним додатковим елементом

Схема реалізації динамічної індикації з одним додатковим елементом приведена на рисунку 6. До порту В МК підключені катооди всіх світлодіодів матриці, а також - входи буферного регістра. До виходів же буферного

регiстра підключені аноди кожного з семисегментних індикаторів. На порт В спочатку виводиться одиниця, що біжить, яка записується в буферному регістрі сигналом С, що утворюється однією з ліній порту А. Потім на лінії порту В видається семисегментний код символу, який повинен горіти в знакомісці, визначуваному сигналом на виході буферного регістра. В даному випадку лінії порту В використовуються в режимі часового мультиплексування, тобто по ним по черзі передається і код символу, і номер знакомісця. Перевага такого способу індикації - менша витрата ліній портів МК (окрім порту В - всього одна додаткова лінія) і можливість роботи до 8 індикаторів одночасно.

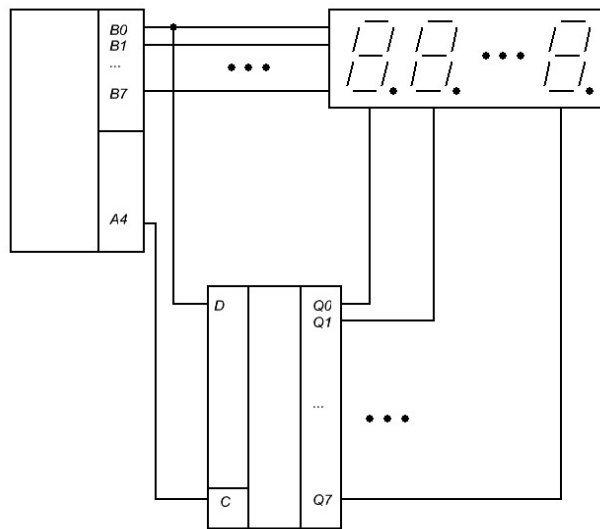


Рисунок 1.7 – Схема реалізації динамічної індикації з регістром зсуву.

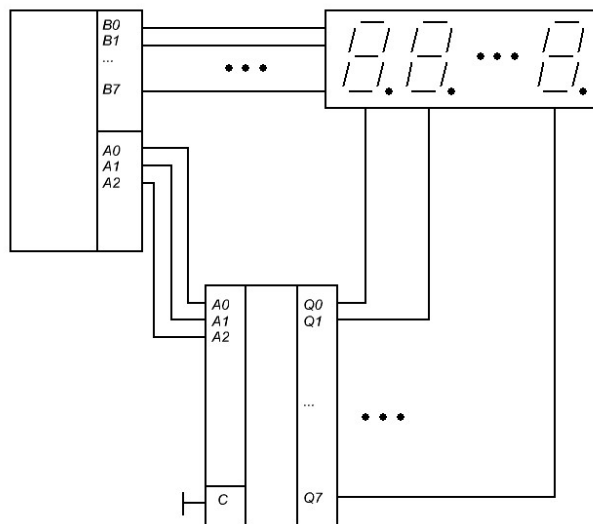


Рисунок 1.8 – Схема реалізації динамічної індикації з дешифратором

Ще один варіант індикації з одним додатковим елементом наведений на рисунку 1.7. У цьому варіанті одиниця, що біжить, реалізується за допомогою регістра зсуву. Порт В у цій схемі також використовується в режимі часового мультиплексування - як для видачі символу, так і для занесення чергового біта до регістра зсуву. Схема також вимагає всього одну додаткову лінію (крім порту В), але за габаритами виходить трохи менше

попереднього варіанту. Ще один варіант реалізації схеми з одним додатковим елементом наведений на рисунку 1.8. Така схема придатна тільки для індикаторів із загальним катодом, оскільки для організації нуля, що біжить, в ній використовується дешифратор. Схема вимагає три додаткові лінії (крім порту В), проте у багатьох випадках вона може виявитися цілком корисною.

Схема реалізації динамічної індикації з двома додатковими елементами наведена на рисунку 1.9. У схемі використовуються два послідовні регістри зсуву: один - для розгортки зображення по стовпчиках (замінює порт А), інший - для розгортки зображення по рядках (замінює порт В). Перевага такого рішення – всього три лінії порту МК. Крім того, при такому варіанті реалізації блок динамічної індикації легко оформити і у вигляді окремої плати. Недолік такого рішення - два додаткових компонента.

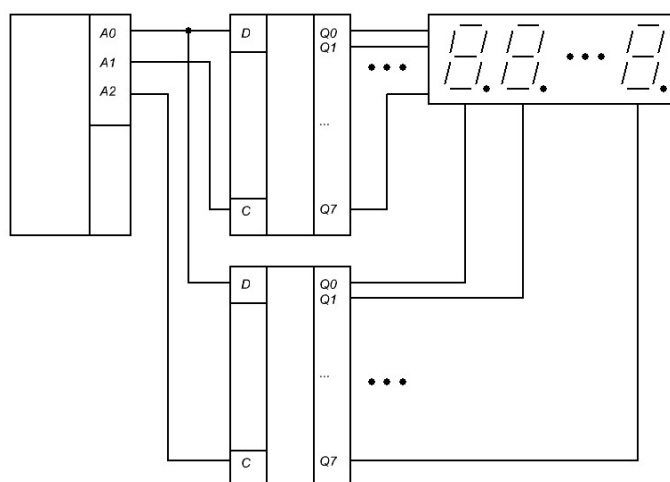


Рисунок 1.9 – Схема реалізації динамічної індикації з двома додатковими елементами

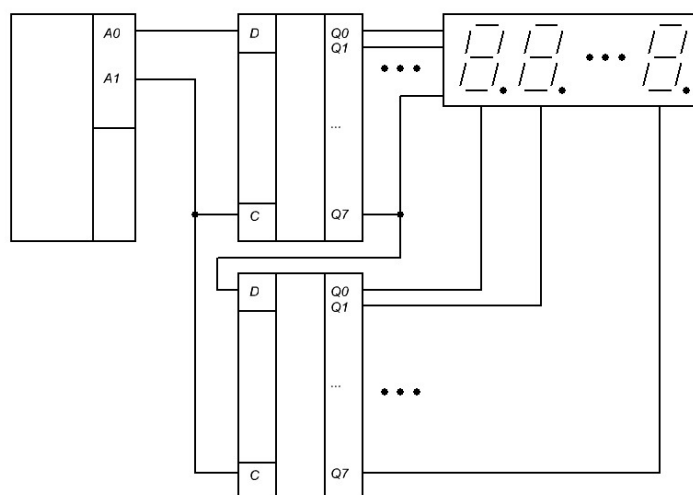


Рисунок 1.10 – Схема динамічної індикації з двома лініями керування

Ще одна схема реалізації наведена на рисунку 1.10. Перевага такої схеми – всього дві лінії порту. Недолік – складніша програма управління і велика тривалість формування вихідних сигналів, що викликає деяке паразитне підсвічування індикаторів (помітно тільки в темряві).

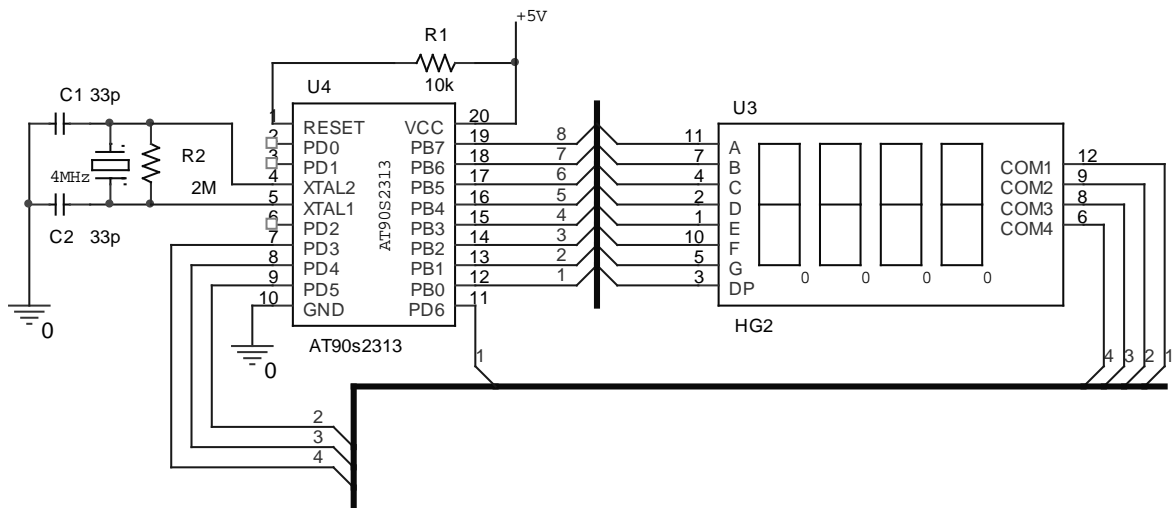


Рисунок 1.10 – Підключення динамічного індикатора до МК AT90S2313.

Практична схема підключення динамічного індикатора на МК AT90S2313 за схемою, що наведена на рисунку 1.5, зображена на рисунку 1.10. Символ, що буде світитись на індикаторі визначається сигналом логічної одиниці на вивід COM1, 2, 3 або 4 індикатора. На шині даних (A, B, C ...) активним рівнем являється рівень логічного 0.

1.3 Відображення інформації на LCD індикаторі

Для відображення поточних значень температури (температури жала та температури, що задана блоком керування) необхідно застосувати LCD індикатор з контролером типу HD44780 фірми Hitachi [4], який є промисловим стандартом де-факто на ринку цифро-буквених індикаторів. LCD на базі HD44780 підключається до AVR мікроконтролера безпосередньо до портів (рисунок 1.11). Є два способи підключення - на 8 біт і на 4 біти. У восьми бітному режимі простіше передавати байти – не потрібно зсувати байт, але в чотирьох бітному режимі використовується на чотири ніжки контролера менше. Для підключення індикатора до мікроконтролера буде задіяний порт D мікроконтролера ATmega8 у чотирьох бітному режимі (рисунок 1.12).

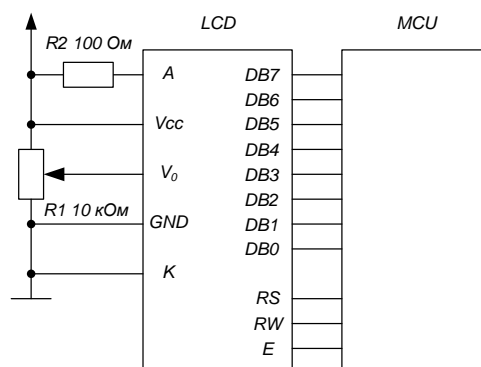


Рисунок 1.11– Підключення LCD індикатора на базі HD44780 до AVR мікроконтролерів

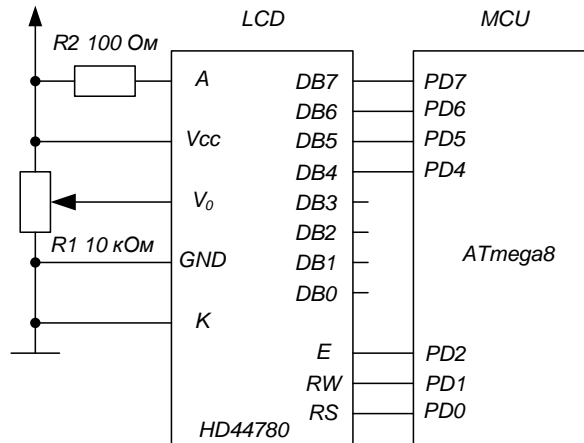


Рисунок 1.12 – Підключення LCD індикатора до мікроконтролера ATmega8

Виводи DB7...DB0 це шина даних/адреси.

E – вхід стробування. Імпульс напруги на цій лінії повідомляє дисплею, що потрібно забирати/віддавати дані з/на шину даних.

RW – визначає в якому напрямку рухаються дані. Якщо 1, то читання з дисплея, якщо 0 то запис у дисплей.

RS – визначає, що передається команда (RS=0) або дані (RS=1). Дані будуть записані на по поточній адресі, а команда виконана контролером.

GND – мінус, він же загальний.

Vcc – плюс живлення, 5В.

V0 – вхід контрастності.

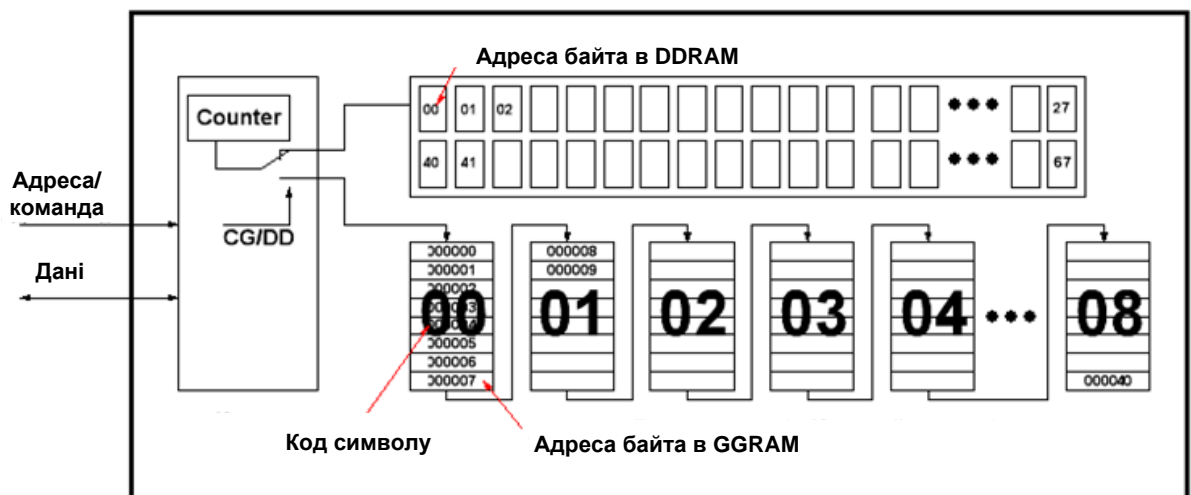


Рисунок 1.13 – Структура адресації контролера HD44780

Контролер HD44780 має свій блок керування, що обробляє команди й пам'ять. Пам'ять ділиться на три види: DDRAM, CGROM, CGRAM.

DDRAM - пам'ять дисплея. Все що запишеться в DDRAM буде виведено на екран. Тобто, наприклад, записали код 0x31 – на екрані відображається

символ «1», так як 0x31 це ASCII код цифри 1. Але є одна особливість – DDRAM пам'ять набагато більша ніж видима область екрана. Як правило, DDRAM містить 80 осередків – 40 у першому рядку й 40 у другому; на дисплеї може рухатися по лінійці осередків (рисунок 1.14). Наприклад, можна записати в DDRAM відразу п'ять пунктів меню, а потім просто показувати по одному пункті. Для переміщення дисплея є спец команда. Також є поняття курсору – це місце в яке буде записаний наступний символ, тобто поточне значення лічильника адреси. Курсор не обов'язково може бути на екрані, він може розташовуватися й за екраном або бути відключений зовсім.

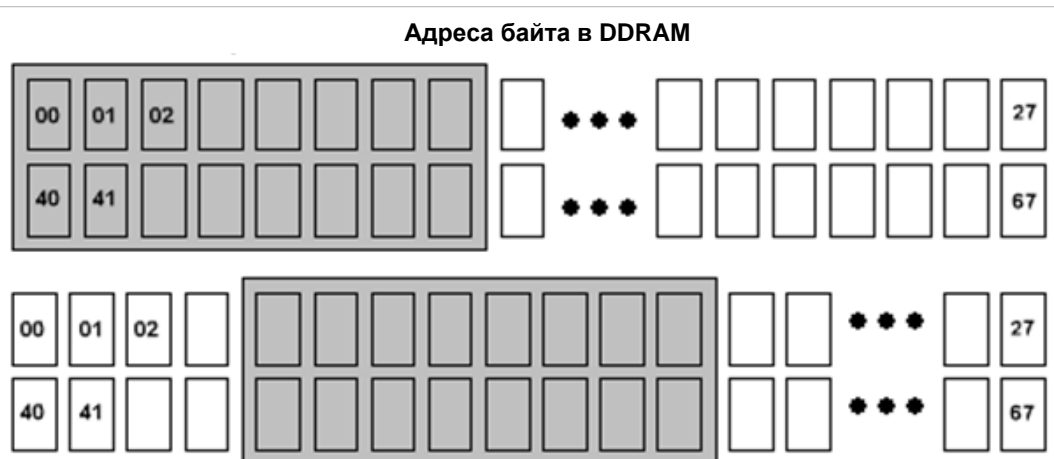


Рисунок 1.14 – Видима й прихована область екранної пам'яті

CGROM – таблиця символів. Коли записуємо в осередок DDRAM байт, то з таблиці береться символ і рисується на екрані. CGROM не можна змінити, тому важливо, щоб вона мала англійські або російські букви.

CGRAM – теж таблиця символів, але її можна міняти, створюючи свої символи. Адресується вона лінійно, тобто спочатку йде 8 байт одного символу, як наведено на рисунку 1.15.

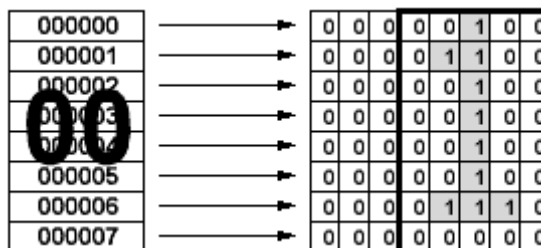


Рисунок 15 – Формування символу в комірці CGRAM

Усього в CGRAM може бути 8 символів, відповідно CGRAM має 64 байта пам'яті. Ці програмувальні символи мають коди від 0x00 до 0x07. Так що, записавши, наприклад, у перші 8 байт CGRAM (перший символ з кодом

00) будь який символ, і записавши в DDRAM нуль (код першого символу в CGRAM) побачимо на екрані цей символ.

Командою вибирається в яку саме пам'ять і починаючи з якої адреси будуть записуватись дані. А потім пересилаємо байти. Система команд контролера індикатора дуже проста. Про те, що передається команда контролеру дисплея повідомить RS=0. Сама команда складається зі старшого біта, що визначає за що відповідає дана команда й біт параметрів, що вказують контролеру HD44780 як далі працювати.

Таблиця 1 – Таблиця команд HD44780

DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Значення
0	0	0	0	0	0	0	1	Очищення екрана. Лічильник адреси на 0 позицію DDRAM
0	0	0	0	0	0	1	-	Адресація на DDRAM скидання зсуву. Лічильник адреси на 0
0	0	0	0	0	1	I/D	S	Настроювання зсуву екрана й курсору
0	0	0	0	1	D	C	B	Настроювання режиму відображення
0	0	0	1	S/C	R/L	-	-	Зсув курсору або екрана, залежно від біт
0	0	1	DL	N	F	-	-	Вибір числа ліній, ширини шини й розміру символу
0	1	AG	AG	AG	AG	AG	AG	Перемкнути адресацію на SGRAM і задати адресу BSGRAM
1	AD	AD	AD	AD	AD	AD	AD	Перемкнути адресацію на DDRAM і задати адресу в DDRAM

I/D – інкремент або декремент лічильника адреси. По замовчуванні встановлений 0 (декремент).

S – зсув екрана, якщо поставити 1, то з кожним новим символом буде пересуватись вікно екрана, поки не досягне кінця DDRAM.

D – включити дисплей. Якщо поставити 0 то зображення зникне, а в цей час можемо у відеопам'ять записати нову інформацію. Для того, щоб вона з'явилася в цю позицію треба записати 1.

C – включити курсор у вигляді прочерку.

B – зробити курсор у вигляді чорного квадрата.

S/C – зсув курсору або екрана. Якщо встановлено 0, то зсувається курсор. Якщо 1, то екран. По одному разу за команду.

R/L – визначає напрямок зсуву курсору й екрану (0 – ліворуч, 1 – праворуч).

D/L – біт, що визначає ширину шини даних (1 – 8 біт, 0 – 4 біти).

N – число рядків (0 – один рядок, 1 – два рядки).

F – розмір символу (0 – 5x8, 1 – 5x10).

AG – адреса в пам'яті CGRAM.

AD – адреса в пам'яті DDRAM.

Завдання:

1. Включити дисплей.
2. Очистити вміст.
3. Зрушити курсор на одну позицію.
4. І записати туди "1".

Рішення (послідовність команд) :

Насамперед *Ініціалізація* дисплея без якої велика частина дисплеїв на HD44780 просто відмовиться працювати (шина 8 біт, курсор в 0)

00111000 – Шина 8 біт, 2 рядки

00000001 – Очищення екрану

00000110 – Інкремент адреси. Екран не рухається

1. **00001100** – Включили дисплей (D=1)
2. **00000001** – Очистили дисплей. Показчик встав на DDRAM
3. **00010100** – Зсув курсору (S/C=0) праворуч (R/L=1)
4. **00110001** – запис даних (ніжка RS=1) код "1" 0x31

Жирним шрифтом виділений ідентифікатор команди.

Завдання: створити свій символ з кодом 01 і вивести його на екран.
Ініціалізація дисплея виконана і він готовий до прийому даних.

Рішення:

1. **01001000** Вибираємо в CGRAM адресу 0x08 - якраз початок другого символу (на один символ йде 8 байт)
2. **00000001** Це пішли 8 байт даних. (RS=1)
3. **00000010** Малюємо значок блискавки
4. **00000100**
5. **00001000**
6. **00011111** Старші три біта не діють
7. **00000010** Туди можна писати що завгодно, на
8. **00000100** результат впливати не буде.
9. **00001000** Останній байт даних
10. **10000000** Це команда - перемикання адреси на **DDRAM** і показчик на адресу **00000000** - перший символ в першому рядку.
11. **00000001** Дані (RS=1), код 01 - саме в нього засунули блискавку.

Алгоритм читання/запису в LCD контролер HD44780

Напрям, а також команда/дані визначаються ніжками, а читання і запис здійснюється по переходу строба (вивід E) з 1 в 0

Ініціалізація портів мікроконтролера

RS, RW, E - в режим виходу.

DB7– DB0 в режим входу.

Очікування готовності, читання прапора зайнятості.

1. Порт даних на вхід з підтяжкою (DDR=0, PORT=1)
2. RS=0 (команда)
3. RW=1 (читання)
4. E=1 (У цей момент дані з LCD виводяться на шину)
5. Пауза (14 тактів процесора на 8МГц)
6. E=0
7. Читаємо з порту. Якщо біт 7 (Busy flag) встановлений, то повторюємо усе наново, поки не скинеться.

Запис команди

1. Очікування готовності
2. RS=0 (команда)
3. RW=0 (запис)
4. E=1 (У цей момент дані з LCD виводяться на шину)
5. Порт на вихід
6. Вивести в порт код команди
7. Пауза
8. E=0
9. Порт на вхід

Запис Даних

1. Очікування готовності
2. RS=1 (Дані)
3. RW=0 (запис)
4. E=1 (У цей момент дані з LCD виводяться на шину)
5. Порт на вихід
6. Вивести в порт код команди
7. Пауза
8. E=0
9. Порт на вхід

Читання команди

1. Очікування готовності
2. Порт даних на вхід з підтяжкою (DDR=0, PORT=1)
3. RS=0 (команда)
4. RW=1 (читання)
5. E = 1 (У цей момент дані з LCD виводяться на шину)
6. Пауза
7. Читаємо дані з порту
8. E=0

Читання даних

1. Очікування готовності
2. Порт даних на вхід з підтяжкою (DDR=0, PORT=1)
3. RS=1 (Дані)
4. RW=1 (читання)
5. E = 1 (Підготуватись .У цей момент дані з LCD виводяться на шину)
6. Пауза
7. Прочитуємо дані з порту
8. E=0

З чотирьох розрядною шиною усе точно також, тільки там кожна операція читання/запису робиться за два строба.

Запис:

1. E=1
2. Пауза
3. Виставили в порт старшу тетраду
4. E=0
5. Пауза
6. E=1
7. Пауза
8. Виставили в порт молодшу тетраду
9. E=0

Читання

1. E=1
2. Пауза
3. Читаємо з порту старшу тетраду
4. E=0
5. Пауза
6. E=1
7. Пауза
8. Читаємо з порту молодшу тетраду
9. E=0

Бібліотека для LCD на базі HD44780 на ASM

<http://easyelectronics.ru/avr-uchebnyj-kurs-biblioteka-dlya-lcd-na-baze-hd44780.html>

Бібліотека для LCD на базі HD44780 на C

<http://radiodod.ru/content/view/112/62/>

2 Кнопки та датчики

Практично жоден МПП не обходиться без кнопок і простих датчиків. За допомогою цих периферійних елементів в МПП поступає різна інформація, яка використовується для зміни алгоритму роботи програми. Схема підключення контактного датчика до МК наведена на рисунку 2.1. У наведеному прикладі датчик підключений до лінії PD0 порту D МК. Через цей вхід МК проводить зчитування стану датчика. Датчик можна підключити і до будь-якої іншої лінії будь-якого з портів МК.

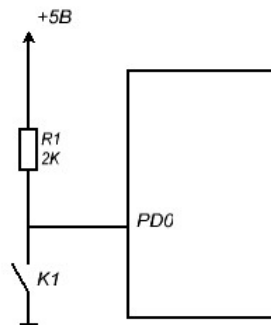


Рисунок 2.1 – Підключення контактного датчика до МК

У початковому стані контакти датчика розімкнені. На вхід МК через резистор R1 прикладається напруга від джерела живлення + 5 В. МК сприймає цю напругу як сигнал логічної одиниці. При спрацьовуванні датчика контакти замикаються і з'єднують вивід МК із загальним дротом. Тепер мікросхема сприймає вхідний рівень сигналу як логічний нуль. Резистор R1 при цьому служить струмообмежувальним елементом, запобігаючи короткому замиканню між шиною живлення і загальним дротом. Деякі МК мають свої внутрішні резистори навантаження, які можуть замінити зовнішній резистор. Схема підключення декількох датчиків або кнопок до МК зображена на рисунку 2.2.

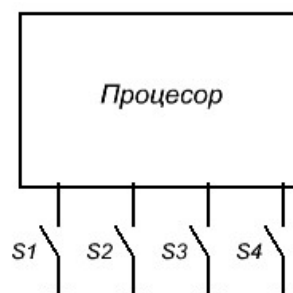


Рисунок 2.2 – Підключення кнопок або простих датчиків до МК

У схемі, що зображена на рисунку 3, при натисненні однієї з клавіш змінюється постійна напруга на відповідному вході процесора, яка розпізнається процесором і дешифрується в певну команду. Ця напруга максимальна (приблизно 5 В), коли кнопки не натиснуті, і мінімальна (0 В) при натиснутій клавіші S1.

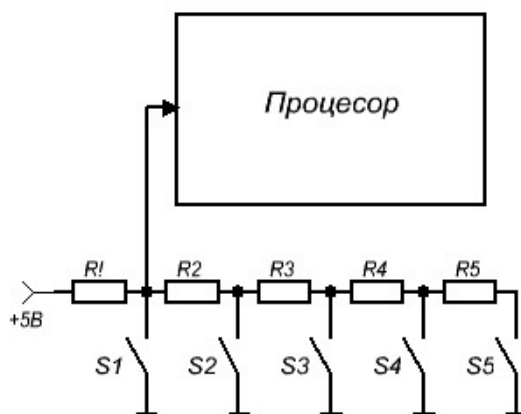


Рисунок 2.3– Підключення кнопок зміною напруги на аналоговому вході МК

Існує два види клавіатур, що підключаються до МК: з скануванням і з кодуванням.

Блок-схема 12-клавішної клавіатури з скануванням показана на рисунку 2.4. Клавіші розташовані у вузлах матриці, у якої чотири лінії рядків і три лінії стовпчиків. На лінії стовпчиків по черзі подається негативний імпульс (логічний «0»). У цей момент перевіряється стан чотирьох ліній рядків. Якщо натиснутих клавіш немає, всі лінії рядків мають високий рівень (вони підключені до напруги +5V через резистори). Якщо ж клавіша натискається, і на лінії стовпчика, відповідного натиснутій клавіші, все ще нуль, то адекватна лінія рядка також стає рівною нулю. Знаючи номери стовпчика і рядка, можна отримати позицію натиснутої клавіші.

У клавіатурі з кодуванням застосовують спеціалізовані мікросхеми, які виявляють натиснення клавіші і передають її код. Приклад такого пристрою є мікросхема ММ74С922 (National Semiconductors) .

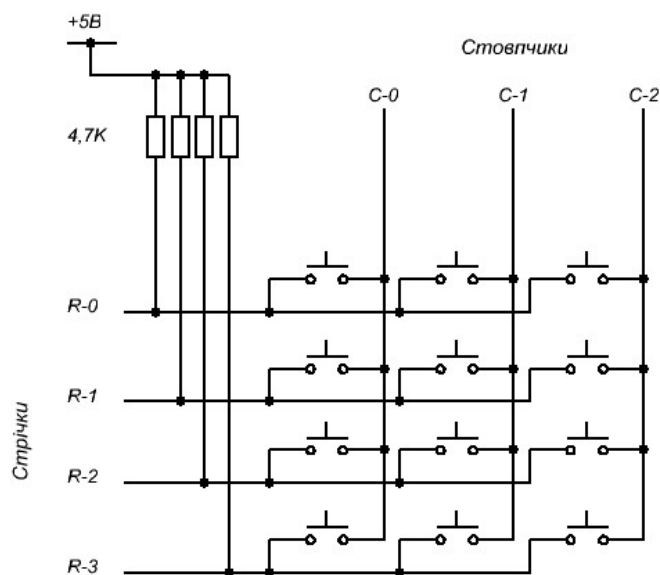


Рисунок 2.4 – Матрична клавіатура 4x3

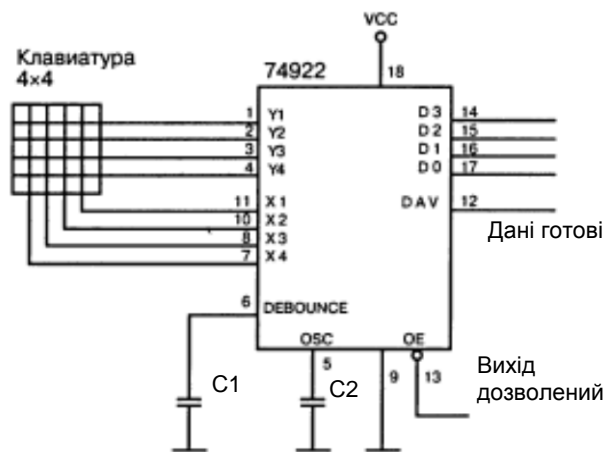


Рисунок 2.5 – Клавіатура 4x4 на ІМС 74922

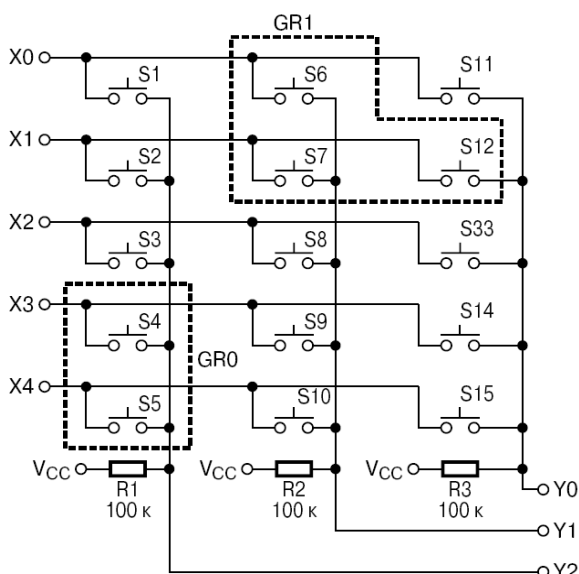


Рисунок 2.6 – Структура матричної клавіатури

Схема 74922 являє собою генератор синхроімпульсів, лічильник і дешифратор, які сканують клавіатуру, визначають натиснуті клавіші й перетворюють цю подію у двійковий код. Коли відбувається натискання клавіші, мікросхема переводить лінію DAV у високий стан, показуючи, що надійшли дані.

Розглянемо організацію клавіатури 5x3, що зображена на рисунку 2.6. Горизонтальні лінії X0...X4 (ряди або рядки) є вихідними, а вертикальні Y0...Y2 (колонки) є вхідними. У неактивному перебуванні на виходах X0...X4 присутній рівень логічної 1, а входи Y0...Y2 зазвичай підтягнуті до живлення резисторами. Для цієї мети, як правило, використовуються вбудовані резистори (pullup), які є у деяких портів МК. При скануванні на горизонтальних лініях матриці формується сигнал нуля, що "біжить". На решті виходів зберігається рівень логічної 1. Для кожної фази нуля, що "біжить", проводиться зчитування і аналіз коду Y0...Y2. При замиканні якої-небудь з клавіш матриці, програмно встановлюється код даної клавіші. Окрім розпізнавання положення натиснутої клавіші слід програмно захиститися від

"тремтіння" контактів, тобто від впливу перехідних процесів, а також ситуацій, пов'язаних з одночасним натисненням декількох клавіш.

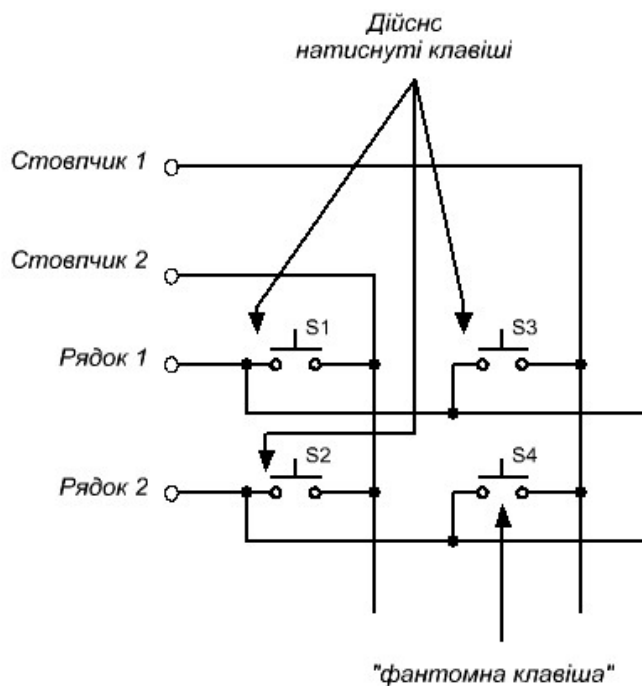


Рисунок 2.7 – Помилкова клавіша при скануванні

Для того, щоб запобігти протіканню небезпечних струмів при одночасному замиканні декількох клавіш у одному стовпчику, в лініях X0...X4 зазвичай встановлюють послідовно розв'язуючі резистори або діоди. З цією ж метою можна використовувати і інший метод сканування, при якому всі неактивні горизонтальні шини, окрім шини нуля, що "біжить", програмно призначаються входами. Вплив перехідних процесів можна виключити, якщо повторно зчитувати стан входів матриці сканування через певну часову затримку.

У невеликих клавіатурних матрицях комбінації декількох натиснутих клавіш не використовуються. Але якщо це все-таки буде потрібно, то для сканування краще використовувати метод з призначенням входами неактивних рядків. У цьому випадку можна упевнено визначати дві одночасно натиснуті клавіші в одному рядку і навіть визначати позиції трьох одночасно натиснутих клавіш. Але це справедливо тільки якщо топологія трьох натиснутих кнопок не утворює прямого кута, як показано на рисунку 7.

Для раціонального використання ресурсу портів їх виводи, які використовуються для сканування клавіатури, можна використовувати і для інших функцій, наприклад, для підтримки динамічної світлодіодної індикації. На рисунку 8 показаний приклад такого мультиплексування портів МК.

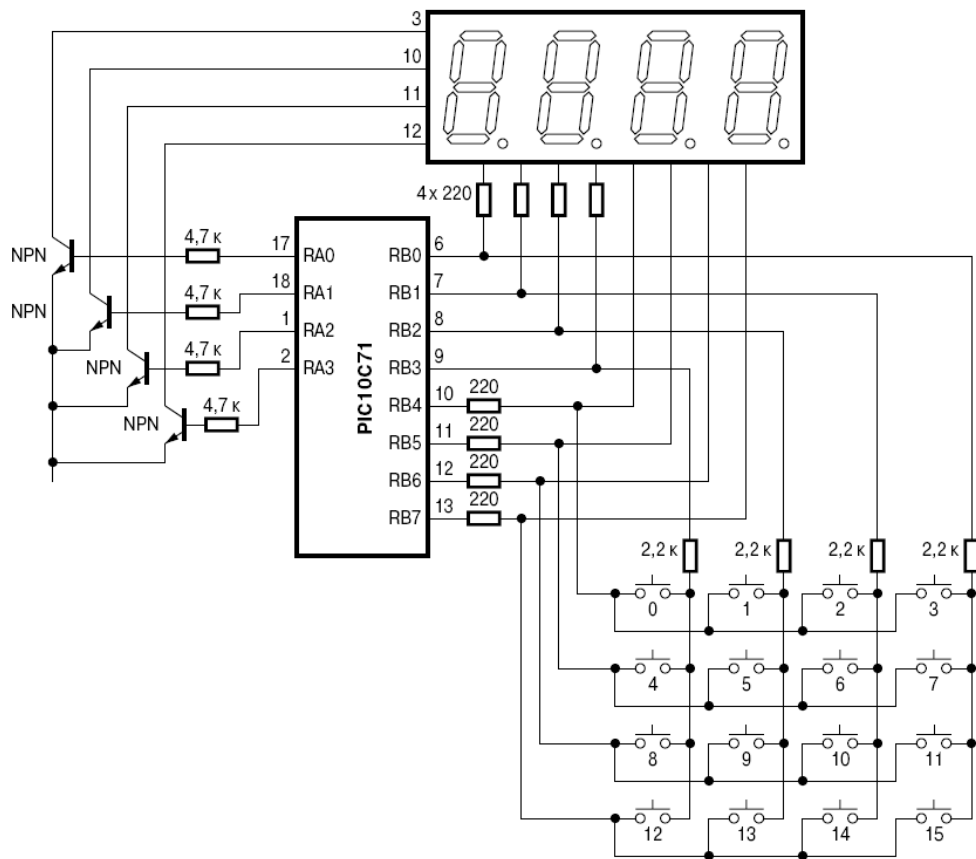


Рисунок 2.8 – Мультиплексорне управління матричною клавіатурою і семисегментним світлодіодним табло

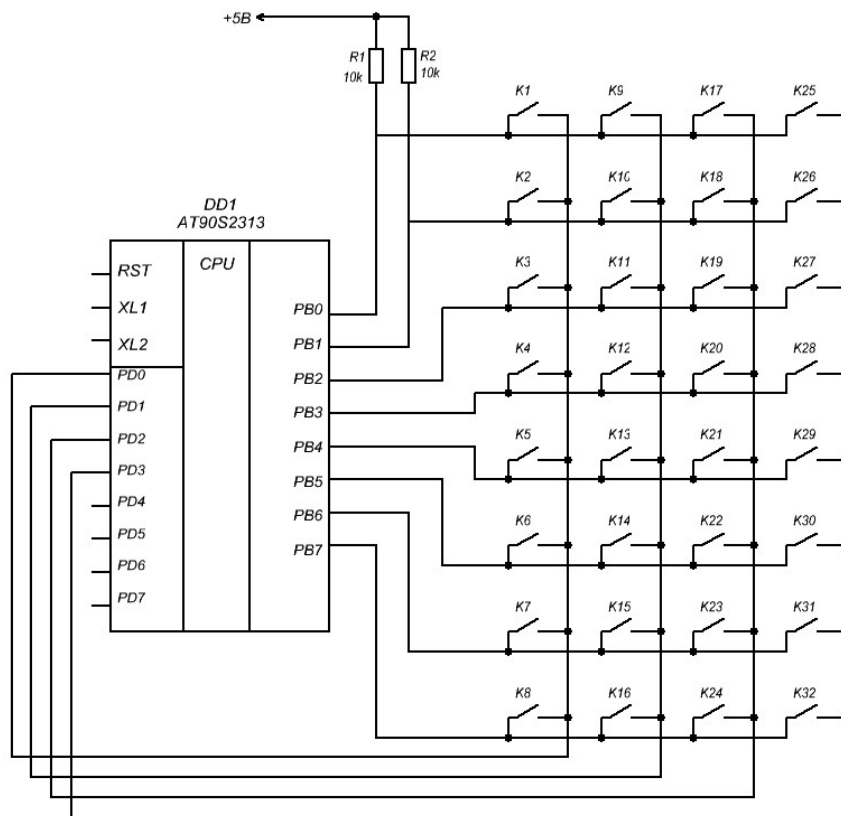


Рисунок 2.9 – Схема підключення клавіатури у вигляді матриці клавіш

Схема матричної клавіатури на 32 клавіші наведена на рисунку 2.9. Додавши один дешифратор, можна заощадити дві лінії порту PD. Схема клавіатури з дешифратором наведена на рисунку 10. У цій схемі для вибору одного з чотирьох стовпчиків клавіатури використовується дешифратор DD2 типу К555ИД4. У такій схемі для сканування стовпчиків МК повинен подавати на виходи PD0 і PD1 двійковий код номера стовпчика. Код поступає на входи А0 і А1 дешифратора. В результаті один з його виходів (той, номер якого відповідає коду, що поступив) прийме нульове значення. На решті виходів буде одиниця. Так, при коді 00В на вході дешифратора вихід Q0 (вивід 9) приймає нульове значення. Для коду 01В - нуль буде на виході Q1 і так далі. Таким чином, МК може перебирати всі чотири стовпчики, використовуючи всього два розряди.

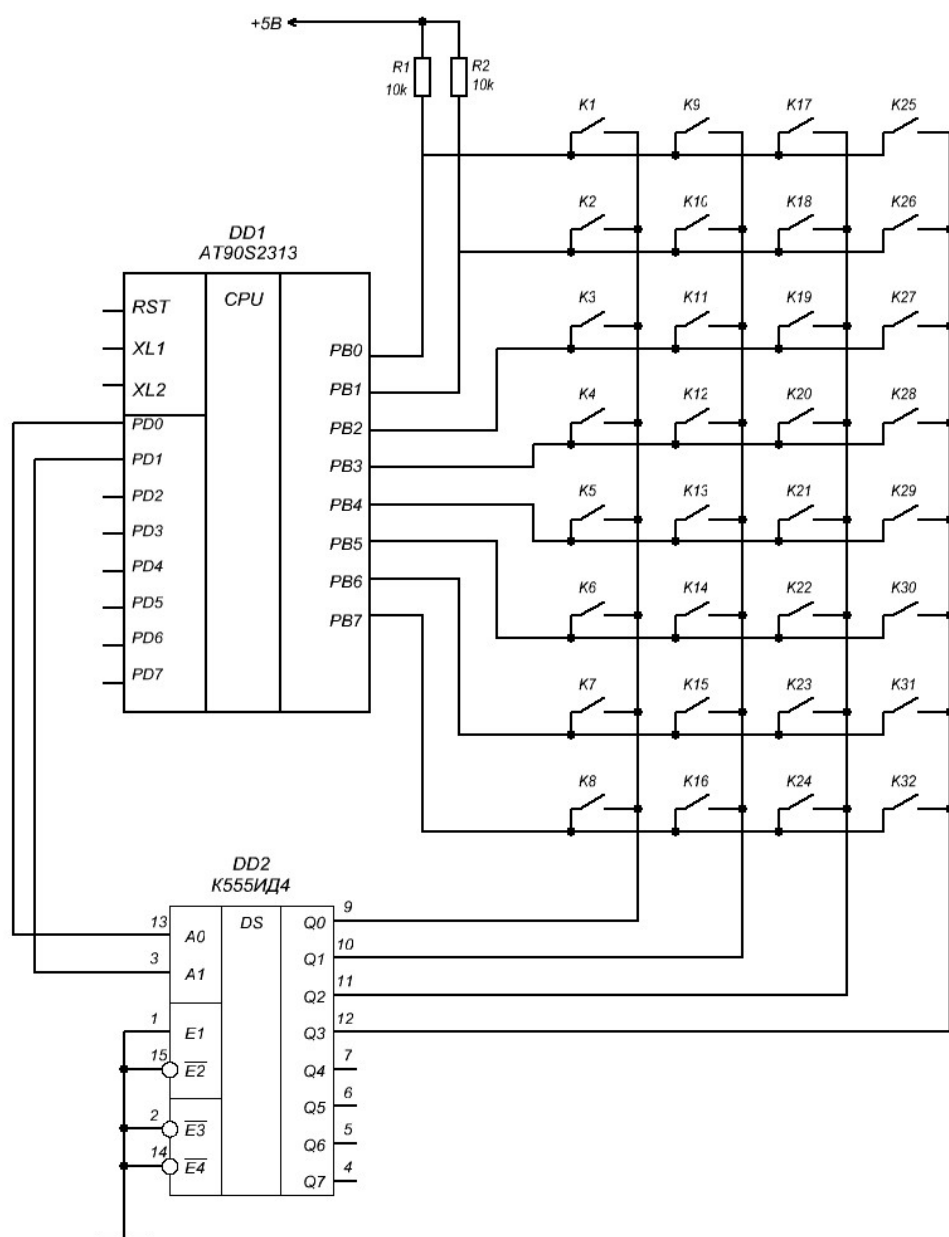


Рисунок 2.10 – Підключення клавіатури з використанням дешифратора

Існують і інші способи зменшення числа виводів МК, призначених для сканування клавіатури. Реалізація одного з таких способів наведена на рисунку 2.11. У звичайних матричних клавіатурах при замиканні кожного контакту утворюється електричне з'єднання між вихідним і вхідним ланками схеми сканування. Але якщо змінити топологію розташування клавiш так, щоб при замиканні контакту забезпечувалася зміна потенціалів групи шин, то число кодованих клавiш можна значно збільшити. Така конструкція клавіатури забезпечує кодування більшого числа клавiш, ніж звичайна матрична, при однаковому числі шин, що використовуються для кодування. Збільшення досягнуте за рахунок того, що додаткові кодові комбінації утворюються парами шин кодування.

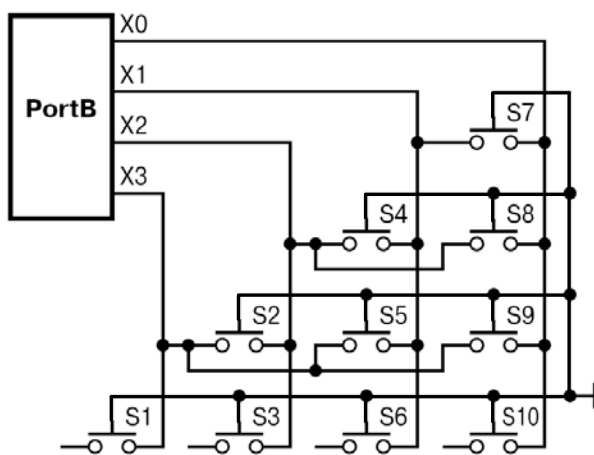


Рисунок 2.11 – Топологія клавіатури з одним контактором

Для подібного варіанту топології сканування не потрібне. Всі шини є входами з підтяжкою до напруги живлення резисторами. Контактор з'єднаний із загальною шиною. При замиканні контакту потенціал контактора передається на одну або дві кодові шини, на перетині яких він знаходиться. Якщо ввести другий замикаючий контактор з потенціалом живлення, то можна удвічі збільшити число сканованих клавiш. Для цього варіанту необхідно проводити сканування, наприклад, з періодом 20-30 мс. При скануванні використовується ефект пам'яті на паразитній ємності шин. У першій фазі сканування всі шини визначаються виходами, і на них подається потенціал логічного 0. Потім виводи порту перевизначаються як входи, і проводиться зчитування їх стану. Факт замикання фіксується по зміні потенціалу, після чого проводиться друга фаза сканування. Цього разу на шини подається потенціал живлення, проводиться аналіз можливих станів шин і визначаються позиції замкнених контактів. Наприклад, при використанні чотирьох шин із звичайною матричною топологією можна сканувати всього 4 кнопки, а при використанні вищеописаного методу вже 20 (рисунок 2.12).

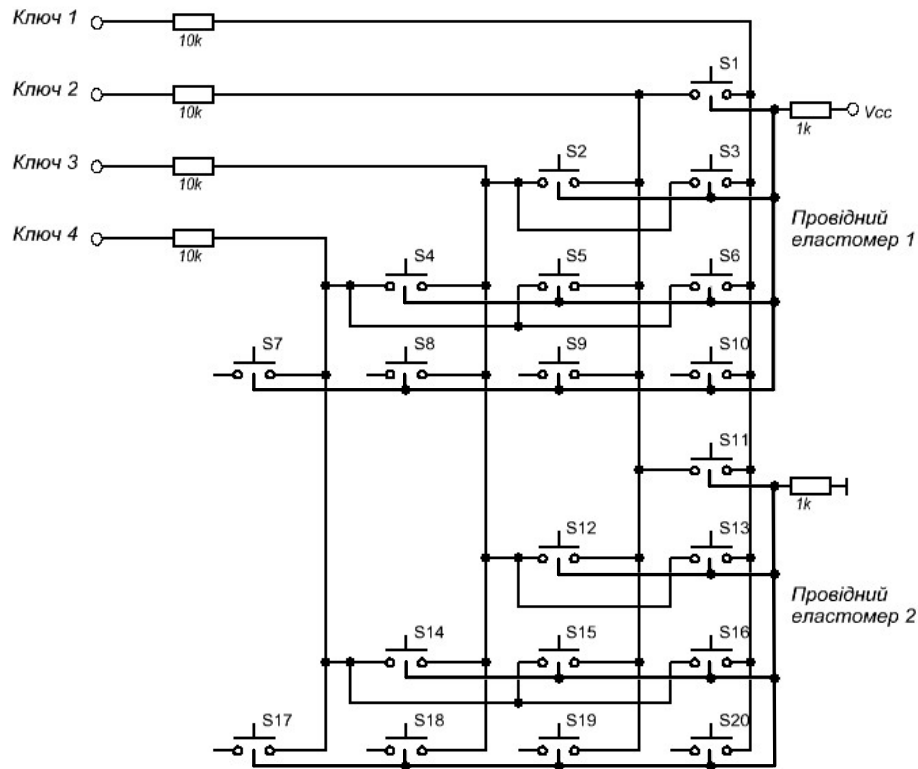


Рисунок 2.12 – Топологія клавіатури на 20 клавіш з 4 лініями сканування

;Підпрограма опитування клавіатури 3*5

;Опитування першого стовпчика

Клава:

ldi temp, 0b11110111

out PortB, temp

sbis PinB,0

ldi sklav,1

sbis PinB,1

ldi sklav,6

sbis PinB,2

ldi sklav,11

;код стовпчика 1

;

;Перевіряємо клавішу 1

;Якщо натиснута зберігаємо код
клавіші в регістрі стану клавіатури

;Перевіряємо клавішу 6

;Якщо натиснута зберігаємо код
клавіші в регістрі стану клавіатури

;Перевіряємо клавішу 11

;Якщо натиснута зберігаємо код
клавіші в регістрі стану клавіатури

; Опитування другого стовпчика

ldi temp, 0b11101111

out PortB, temp

sbis PinB,0

ldi sklav,2

sbis PinB,1

ldi sklav,7

sbis PinB,2

ldi sklav,12

;код стовпчика 2

;

;Перевіряємо клавішу 2

;Якщо натиснута зберігаємо код
клавіші в регістрі стану клавіатури

;Перевіряємо клавішу 7

;Якщо натиснута зберігаємо код
клавіші в регістрі стану клавіатури

;Перевіряємо клавішу 12

;Якщо натиснута зберігаємо код
клавіші в регістрі стану клавіатури

; Опитування третього стовпчика

```
ldi temp, 0b11011111      ;код стовпчика 3
out PortB, temp           ;
sbis PinB,0               ;Перевіряємо клавiшу 3
ldi sklav,3               ;Якщо натиснута зберігаємо код
                           ;клавiші в реєстрі стану клавiатури
sbis PinB,1               ;Перевіряємо клавiшу 8
ldi sklav,8               ;Якщо натиснута зберігаємо код
                           ;клавiші в реєстрі стану клавiатури
sbis PinB,2               ;Перевіряємо клавiшу 13
ldi sklav,13              ;Якщо натиснута зберігаємо код
                           ;клавiші в реєстрі стану клавiатури
```

; Опитування четвертого стовпчика

```
ldi temp, 0b10111111     ;код стовпчика 4
out PortB, temp           ;
sbis PinB,0               ;Перевіряємо клавiшу 4
ldi sklav,4               ;Якщо натиснута зберігаємо код
                           ;клавiші в реєстрі стану клавiатури
sbis PinB,1               ;Перевіряємо клавiшу 9
ldi sklav,9               ;Якщо натиснута зберігаємо код
                           ;клавiші в реєстрі стану клавiатури
sbis PinB,2               ;Перевіряємо клавiшу 14
ldi sklav,14              ;Якщо натиснута зберігаємо код в
                           ;клавiші в реєстрі стану клавiатури
```

; Опитування п'ятого стовпчика

```
ldi temp, 0b01111111    ;код стовпчика 5
out PortB, temp           ;
sbis PinB,0               ;Перевіряємо клавiшу 5
ldi sklav,5               ;Якщо натиснута зберігаємо код
                           ;клавiші в реєстрі стану клавiатури
sbis PinB,1               ;Перевіряємо клавiшу 10
ldi sklav,10              ;Якщо натиснута зберігаємо код
                           ;клавiші в реєстрі стану клавiатури
sbis PinB,2               ;Перевіряємо клавiшу 15
ldi sklav,15              ;Якщо натиснута зберігаємо код
                           ;клавiші в реєстрі стану клавiатури
rcall drebezg             ;Виклик підпрограми захисту від тремтіння
                           ;контактів
```

; Достатньо 4 NOP. Потрібно для того, щоб перед перевіркою дати ніжці час на те,
; щоб зайняти потрібний рівень. Річ у тому, що реальна схема має деяке значення
; C і L, яке робить неможливою миттєву зміну рівня, невелика затримка все ж є. А
; на швидкості в 8МГц і вище процесор клацає команди з такою швидкістю, що
; напруга на виводі ще не спала, а ми вже перевіряємо стан виводу.

```
ret                       ;Вихід з підпрограми
```

Алгоритм з використанням переривання

Ініціалізація:

1. Піни лінії, наприклад, X робимо виходами;
2. Піни лінії, наприклад, Y робимо входами;
3. На лінії X виводиться рівень 1;
4. На лініях Y включаємо підтягуючі резистори;
5. Ініціалізували один з таймерів, на ньому обробляємо події клавіатури;
6. Дозволяємо переривання по переповнюванню таймера.

Робота:

1. По перериванню читаємо стан порту Y;
2. Записуємо поточне значення в тимчасову змінну;
- 3а. Порівнюємо набутого значення і попереднє, якщо значення не співпадають: обнуляємо лічильник повторів і записуємо нове значення в змінну що зберігає попереднє значення;
- 3б. Порівнюємо набутого значення і попереднє, якщо значення співпадають - інкремент значення лічильника повторень. Якщо лічильник повторень співпадає з довірчим значення, виставляємо відповідний Flag і обнуляємо лічильник;
4. Встановлюємо логічну 1 на поточній нозі лінії X і логічний 0 на наступній;
5. Виходимо з переривання.

Опис програм на ASM для опитування клавіатури

<http://easyelectronics.ru/avr-uchebnyj-kurs-procedura-skanirovaniya-klaviatory.html>

http://radiokot.ru/start/mcu_fpga/avr/16/ – опитування клавіатури в циклі

<http://atmel.com/dyn...nts/doc1232.pdf> – від розробників

Часто виникає необхідність використання великої кількості кнопок для різних цілей. Існують різні варіанти реалізації цього завдання. Коли необхідно підключити клавіатуру до МК з обмеженою кількістю вільних ніжок, тоді використовують підключення клавіатури по трьох сигнальних дротах. Додаткові елементи: регістр зсуву SN74198N і декілька резисторів (рис. 2.13).

Кнопки підключені одним виводом до корпусу, другим до регістра зсуву. Резистори R1 - R8 створюють високий логічний рівень на вході регістра зсуву в той час, як відповідна йому кнопка розімкнена. Як тільки кнопка замикається, на вході регістра утворюється логічний нуль, оскільки він виявляється безпосередньо підключений до загального дроту. Значення резистора в 10 кОм не дає протікати занадто великому струму, поки кнопка замкнута, і досить добре створює високий логічний рівень, поки кнопка розімкнена.

Регістр зсуву має вхідний порт (D0 – D7), вихідний порт (Q0 – Q7) і сигнали управління (SR, SL, CLK, S0, S1, MR).

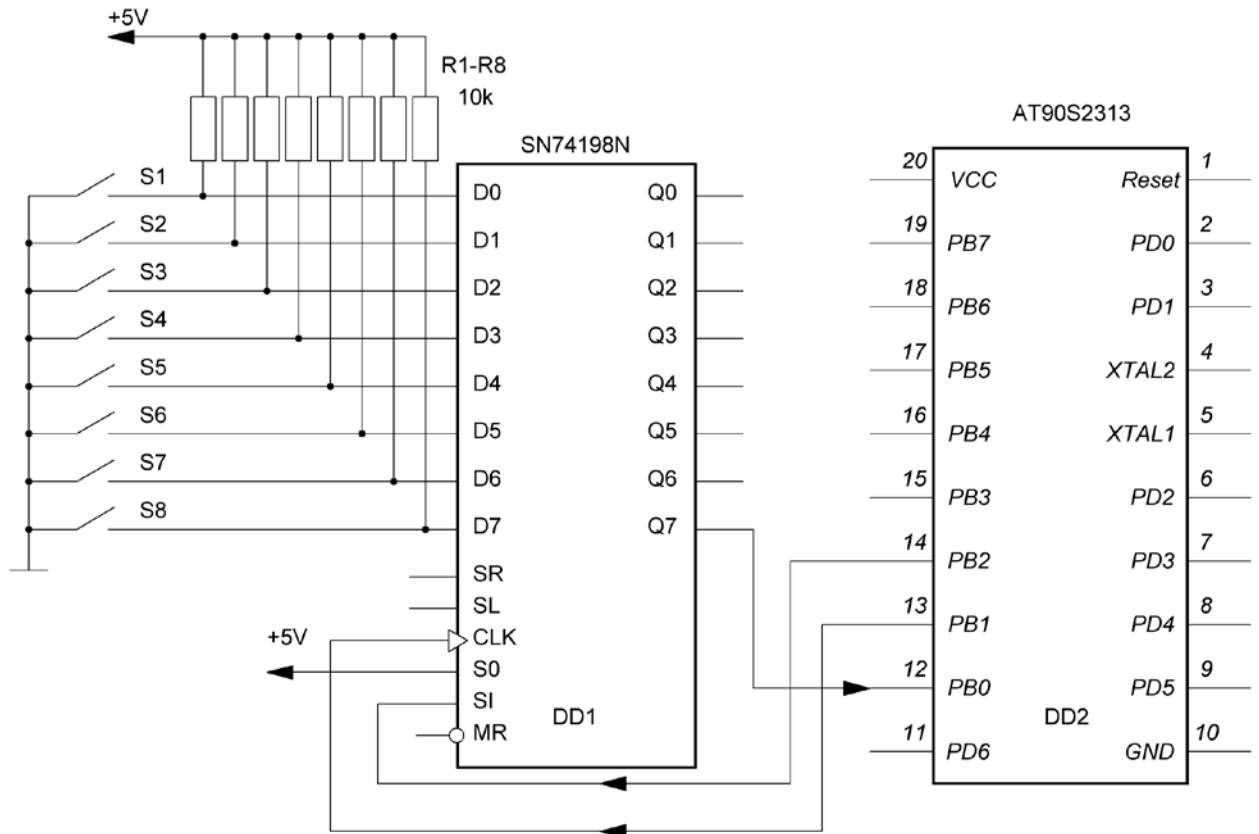


Рисунок 2.13 – Підключення клавіатури з використанням регістра зсуву

Вхідний і вихідний порт можна використовувати як в паралельному режимі, так і в послідовному.

D0 - D7 - вхід. Сюди подаються 8 сигналів з кнопок.

Q0 - Q7 - паралельний вихід. Використовується лише Q7. Інші не потрібні.

CLK - тактовий вхід. Усе в регістрі робиться тільки по тактам на цій ніжці (по наростаючому фронту). Якщо необхідно щось зробити, то подаємо високий рівень на ніжку CLK, а потім подаємо логічний нуль.

S0 і S1. Ці ніжки відповідають за поведінку виходу регістра (Q0 - Q7). По суті цей регістр може мати як послідовний, так і паралельний вихід. Коли на обидві ніжки подається високий логічний рівень завантажуються на вихід ті дані, які в даний момент знаходяться на вході (D0 - D7) за сигналом CLK.

Якщо S1=0, то за сигналом CLK, дані на вихідному порту зрушаться у бік старшого біта (біт з виходу Q0 перейде на вихід Q1, а біт, який раніше був на місці Q1 – на Q2 і так далі). Виконавши це 8 разів можна прочитати стан 8 кнопок у послідовному коді з Q7

Алгоритм дій

1. Ставимо S0 і S1 у високий рівень і формуємо сигнал CLK. На виході Q0 - Q7 фіксується необхідний байт (8 біт).

2. S1=0, S1=1 і формуємо сигнал CLK
3. Читаємо біт з ніжки Q7
4. Формуємо сигнал CLK.
5. Повертаємося на пункт 3 і повторюємо 7 разів.

Програмна частина

Напишемо функцію, яка опитує кнопки, і поміщати результат в регістр R17 і виводить номера натиснутих кнопок у двійковому коді на світлодіодний індикатор. Інвертуємо регістр R17, так як світлодіоди підключені анодом до порта PA0-7 МК ATMEGA16.

```
.equ  BTN_PORT  = PORTB
.equ  BTN_DDR   = DDRB
.equ  BTN_PIN   = PINB
.equ  BTN_DATA_IN = 0
.equ  BTN_HOLD  = 1
.equ  BTN_CLK   = 2
; Ініціалізація МК
SBI  BTN_DDR,BTN_HOLD      ; вихід HOLD
SBI  BTN_DDR,BTN_CLK      ; вихід CLK
SBI  BTN_PORT,BTN_DATA_IN ; вхід DATA_IN

LDI R18, 0xFF              ; Порт А на вихід. До нього
OUT  DDRA,R18             ; підключаються світлодіоди

;Основний нескінчений цикл:
main: RCALL btn_start      ; викликаємо функцію опитування клавіатури
      COM R17              ; інверсія байта
      OUT PORTA, R17      ; й запис його в порт А
      RJMP Main           ; повернення на початок
                          ; функцію опитування клавіатури

btn_start: SBI  BTN_PORT,BTN_HOLD      ; S1=1
           SBI  BTN_PORT,BTN_CLK      ; CLK=1
           CBI  BTN_PORT,BTN_CLK      ; CLK=0
           CBI  BTN_PORT,BTN_HOLD      ; S1=0

btn_again: LDI  R17,0                ; регістр накопичення результату.
           LDI  R16,8                ; лічильник. Повторюємо 8 раз
btn_loop:  LSL  R17                  ; якщо тут, перший раз, то дана команда
           ; з 0 нічого не зробить, інакше рухаємо
           ; всі біти ліворуч
           SBIC BTN_PIN,BTN_DATA_IN ; якщо на вхід прийшла 1,
           INC  R17                  ; записуємо 1 в молодший розряд R17

           SBI  BTN_PORT,BTN_CLK      ; CLK=1
           CBI  BTN_PORT,BTN_CLK      ; CLK=0
           DEC  R16                   ; зменшуємо лічильник
           BREQ btn_exit              ; якщо R16=0, то переходимо до btn_exit
           Rjmp btn_loop              ; інакше повторяємо цикл, де першою командою
           ; зсуваємо біти ліворуч

btn_exit:  RET
```

Розглянемо реалізацію 16 кнопок. Для цього необхідно застосувати додатковий регістр зсуву. На вхід SR DD2 подаємо сигнал з Q7 DD1. Сигнал на МК буде подаватись з Q7 DD2 (рисунок 2.14).

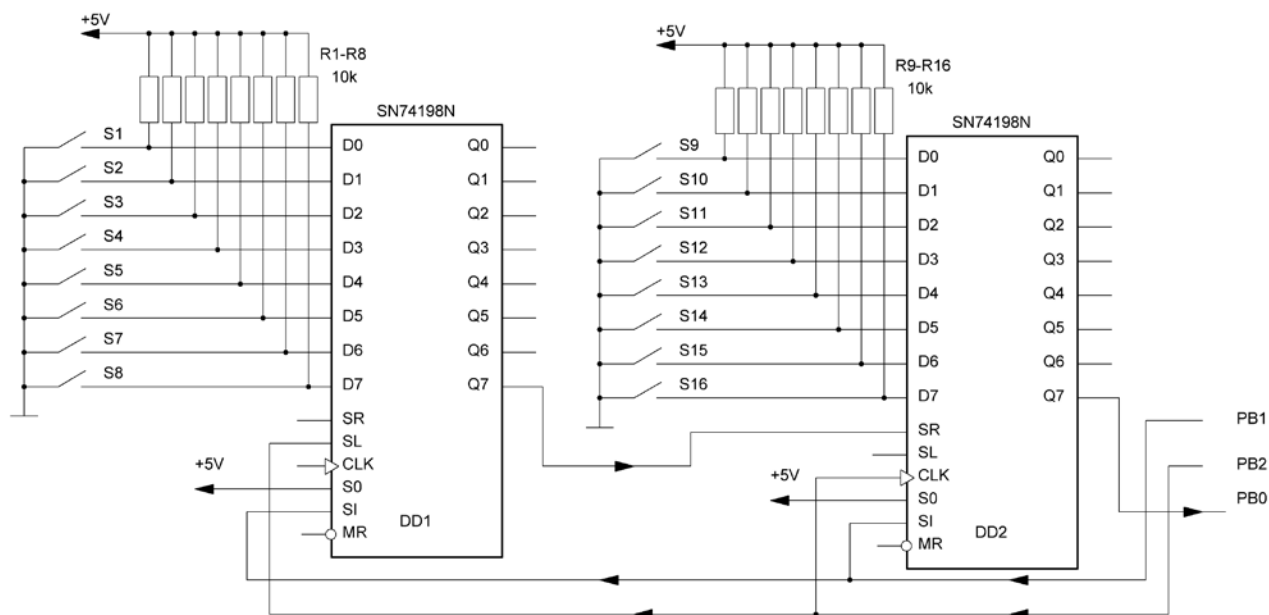


Рисунок 2.14 – Підключення 16 кнопок з використанням регістра зсуву

Алгоритм дій

1. Встановлюємо біт T в регістрі SREG (T=1 означає, що читається перший байт клавіатури, якщо T=0 – читання другого байта).
2. У циклі читаємо 8 біт з регістра зсуву.
3. Перевіряємо біт T:
 - якщо він встановлений, то зчитали перший байт, скидаємо біт T і повертаємося на пункт 2.
 - Якщо він скинутий, то зчитали другий байт

```

btn_start:  SBI   BTN_PORT,BTN_HOLD      ; S1=1
            SBI   BTN_PORT,BTN_CLK      ; CLK=1
            CBI   BTN_PORT,BTN_CLK      ; CLK=0
            CBI   BTN_PORT,BTN_HOLD      ; S1=0
            SET                               ; встановлення T=1 в регістрі SREG.
btn_again:  LDI   R17,0                  ; регістр накопичення результату.
            LDI   R18,8                  ; лічильник. Повторюємо 8 раз
btn_loop:   LSL   R17                    ; якщо тут, перший раз, то дана команда
            ; з 0 нічого не зробить, інакше рухаємо
            ; всі біти ліворуч
            SBIC  BTN_PIN,BTN_DATA_IN    ; якщо на вхід прийшла 1,
            INC   R17                    ; записуємо 1 в молодший розряд R17

            SBI   BTN_PORT,BTN_CLK      ; CLK=1
            CBI   BTN_PORT,BTN_CLK      ; CLK=0

            DEC   R18                     ; зменшуємо лічильник
            BREQ  btn_loop_end           ; якщо лічильник 0, то переходимо в
            ; btn_loop_end
    
```

```

                Rjmp  btn_loop      ; інакше повторяємо цикл, де першою командою
                                ; зсуваємо біти ліворуч
btn_loop_end:
                BRTC  btn_exit      ; якщо T=0, то виходимо з функції
                CLT                  ; інакше скидаємо біт T (будемо приймати 2 байт)

                MOV   R16,R17      ; зберігаємо перший прийнятий байт в R16
                RJMP  btn_again     ; та повертаємось до читання байта
btn_exit:      RET

```

<http://easyelectronics.ru/podklyuchenie-klaviatiry-k-mk-po-trem-provodom-na-sdvigovyh-registrax.html>

<http://easyelectronics.ru/avr-uchebnyj-kurs-procedura-skanirovaniya-klaviatiry.html#comments>

3 Оптичні датчики

Як оптичний датчик найчастіше виступає світло діод та фотоприймач, який називається оптронам. Випускаються оптрони з закритим (optoisolator) оптичним каналом (у монолітному виконанні) і відкритим оптичним каналом (щілинні і відбивальні оптрони).

3.1 Щілинний оптрон

На рисунку 3.1 показаний оптичний датчик – щілинний оптрон (slotted optical switch). Фототранзистор і направлений на нього світлодіод закріплені на пластиковій підставці і розділені проміжком так, що коли якийсь предмет рухається в зазорі, він перекриває світло між світлодіодом і датчиком. Щілинні оптрони часто використовуються для вимірювання швидкості двигуна за допомогою диска з прорізами, розміщеного на осі двигуна. Коли вісь обертається, диск перекриває світловий промінь. Інше застосування щілинного оптрона – це індикація того, відкриті чи закриті двері або, наприклад, кожух охоронного приладу. Прапорець на дверях, потрапляючи в щілину, блокує світло, коли двері закриваються. Механічна комп'ютерна миша також використовує щілинні оптрони.

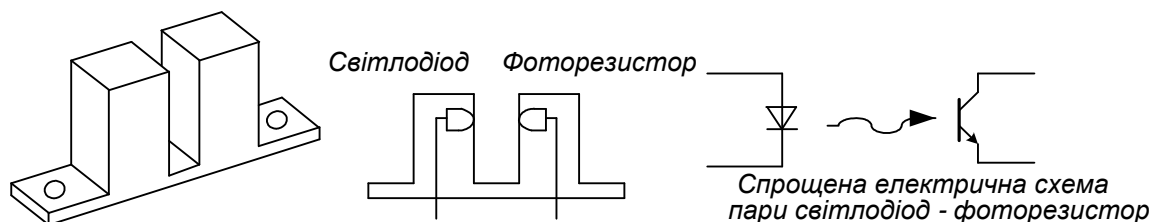


Рисунок 3.1 – Щілинний оптрон

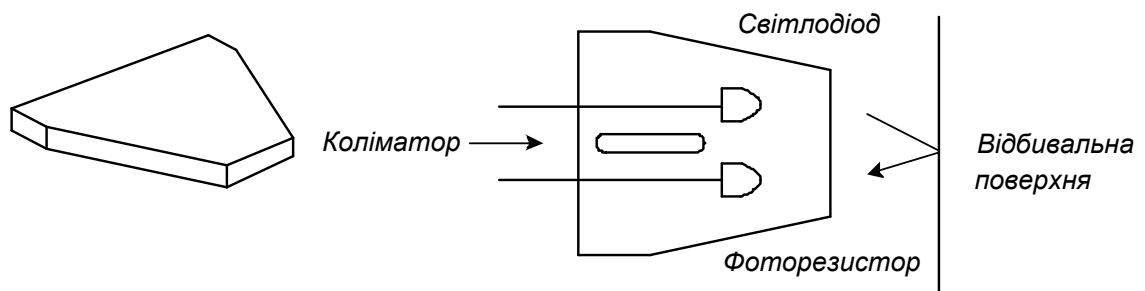


Рисунок 3.2 – Відбивальний оптрон

3.2 Відбивальний оптрон

На рисунку 3.2 показаний інший тип оптичного датчика – відбивальний оптрон (reflective sensor). Принцип роботи цього датчика такий же, як і щілинного, з тією різницею, що фототранзистор приймає відбите, а не пряме світло. Більшість датчиків відбиття характеризується фокусною відстанню – оптимальною відстанню, на якій має бути розміщений відбивальний об'єкт від датчика. Ця відстань дорівнює 0,254... 1,270 см (від 0,1 до 0,5 дюйма). Типове застосування відбивальних оптронів – це реєстрація обертання двигуна за нанесеними на його вісь темними мітками.

Коли вісь обертається, датчик реєструє зміну темних і відбивальних ділянок. Обидва типи оптронів мають схожі характеристики, які необхідно враховувати при проектуванні систем.

3.3 Швидкість спрацьовування оптронів

Оскільки фототранзистор є оптичним приладом, що діє досить повільно, то це обмежує максимальну швидкість реєстрації. Типовий час включення фототранзистора 8 мкс, а вимкнення 50 мкс. Ці часові параметри визначаються швидкістю носіїв під дією світла в переході база-емітер транзистора.

3.4 Коефіцієнт підсилення за струмом

Оптопара світлодіод-фототранзистор має обмежений коефіцієнт підсилення, зазвичай, менше одиниці. Відношення струму, що протікає в колекторі фото транзистора, до струму через світлодіод називається коефіцієнтом передачі за струмом (КПС) (Current Transfer Ratio, CTR). Типове значення КПС для щілинних оптронів складає 0,1. Це означає, що при струмі 10 мА, що протікає через світлодіод, струм колектора фототранзистора складе 1 мкА. КПС інколи задається як коефіцієнт, інколи наводиться у вигляді таблиці, що показує різні значення струмів колектора для різних величин струму світлодіода. КПС залежить від характеристик світлодіода і фототранзистора. КПС слід брати до уваги при створенні інтерфейсу між оптроном і мікропроцесорною системою. По-перше, якщо

оптрон приєднується до цифрового входу (рис. 3.3), вихід транзистора необхідно узгоджувати за логічними рівнями з входом цифрового пристрою.

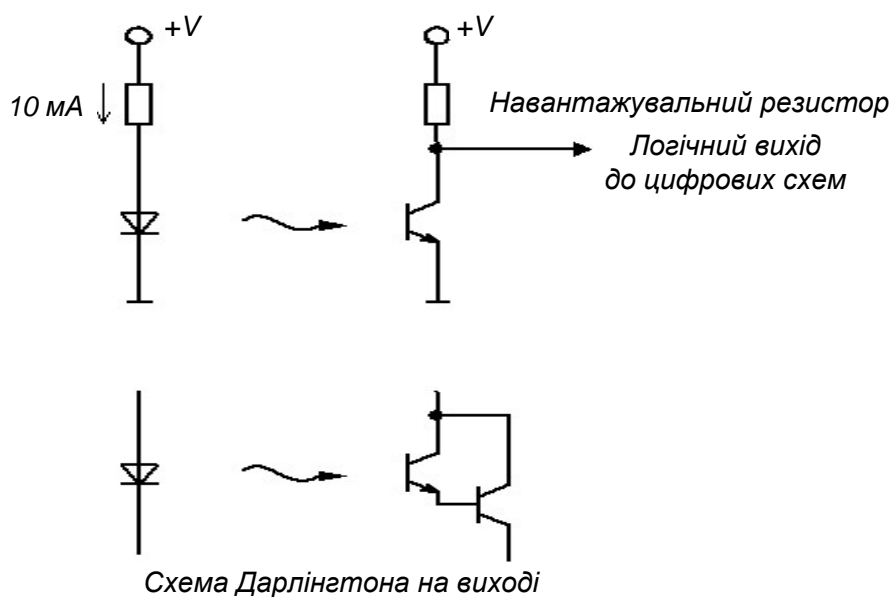


Рисунок 3.3 – Оптрони з цифровим виходом

Для швидкого насичення фототранзистора величину резистора навантаження слід обмежити. Наприклад, якщо через світлодіод протікає струм 10 мА, а мінімальна величина КПС дорівнює 0,1; то величину резистора навантаження слід вибрати 5 кОм. Менша величина резистора забезпечить кращу стійкість до шумів (через менший імпеданс) і, можливо, велику швидкість перемикавання, але не гарантує сумісності зі всіма пристроями, оскільки транзистор не зміг би пропускати достатній струм для забезпечення низького логічного рівня. Щоб знизити величину навантажувального резистора, можна використовувати оптопару з більш високим КПС або живити світлодіод великим струмом.

Випускаються оптрони зі складеним транзистором Дарлінгтона на виході, що забезпечує КПС > 1 , але швидкість перемикавання такого оптрона нижча і складає 20% від швидкості поодиноких транзисторів. До того ж напруга насичення транзистора Дарлінгтона більша, ніж поодинокого транзистора.

Відбивальні датчики також характеризуються КПС. Оскільки датчик розрахований на прийом відбитого світла, то КПС залежатиме від властивостей відбивальної поверхні і відстані до неї. КПС відбивального датчика задається відносно деякої стандартної відбивальної поверхні, розміщеної на характерній фокальній відстані.

3.5 Особливості роботи в інфрачервоному діапазоні

У більшості щілинних і відбивальних датчиків використовуються інфрачервоні світлодіоди і фототранзистори. Це означає, що реакція датчика

може відрізнятися від реакції у видимому діапазоні спектра. Об'єкти, що добре відбивають у видимому світлі, можуть відбивати менш ефективно в інфрачервоних (ІЧ) променях. Слід також враховувати завади, що вносяться природним і штучним освітленням.

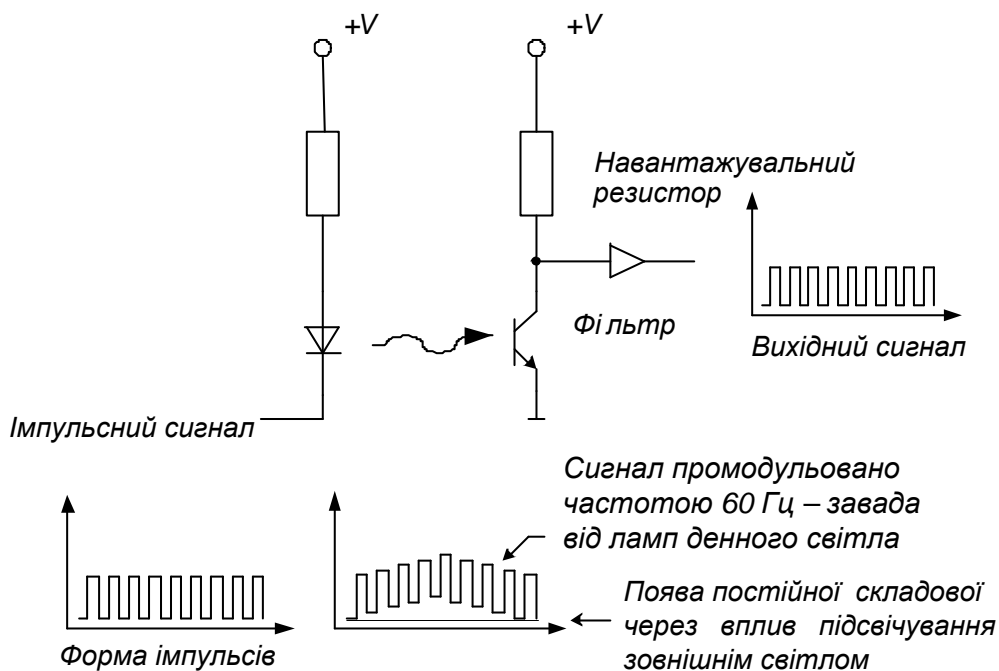


Рисунок 3.4 – Фільтрація сигналу оптичного датчика

На рисунку 3.4 показано, що оптрон, який працює з прямокутними імпульсами, може бути обладнаний фільтром для усунення подібних завад. У даному прикладі освітлювальний прилад додає до вихідного сигналу постійний зсув і коливання з частотою мережі 50 Гц. Після проходження сигналу через фільтр, налаштований на частоту паразитної модуляції мережі, ці завади вирізаються фільтром, і відфільтрований від завад сигнал може бути перетворений в цифровий. Фільтр може бути реалізований як за допомогою електроніки, так і програмно. ІЧ-промені широко застосовуються в побутовій техніці. У пульті дистанційного керування телевізора на ІЧ-променях використовується модуляція сигналу з частотою 40 кГц. Для забезпечення роботи з такою частотою в датчику застосовується високошвидкісний фотодіод.

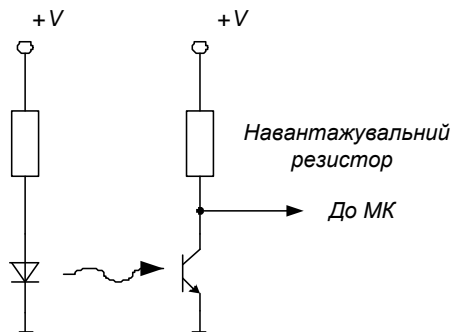
Подібна фільтрація має і недоліки. Один з недоліків – обмеження швидкості. Через значний час вмикання і вимикання фототранзистора частота модуляції, з якою пристрій ще працюватиме, обмежена величиною порядку 10 кГц.

Недолік також в тому, що в результаті фільтрації з'являється деяка затримка в часі, що досягає декількох циклів роботи механічної частини. Так, якщо навіть датчик розрахований на роботу з частотою 10 кГц, в цілому система зможе підтримувати не вище 1 кГц, бо розраховується за найповільнішою частотою. Крім того, якщо зовнішнє освітлення має сильний

вплив, то фототранзистор насичується і фільтрація не в змозі відновити вихідний сигнал. У даному випадку необхідно захистити фототранзистор від зовнішнього засвічування.

3.6 Дискретні оптичні датчики

У деяких пристроях інколи потрібне застосування дискретних оптичних датчиків: світлодіода і фототранзистора. Вони розраховані на роботу в інфрачервоному діапазоні, як і ті, що застосовуються в щілинних і відбивальних оптронах. Дискретні оптопари використовуються в системах, де великий об'єкт заступає шлях світловому променю між світлодіодом і фото транзистором, або там, де відстань дуже велика для використання оптрона.



Може бути необхідне застосування тригера Шмітта.

Навантажувальні резистори в колі світлодіода і фоторезистора вибираються з урахуванням КПС для встановлення необхідних логічних рівнів.

Датчики деяких систем мають достатньо низький КПС для застосування такої схеми.

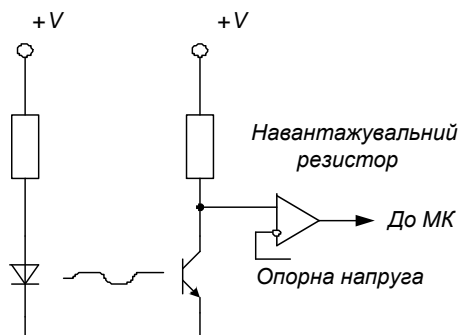
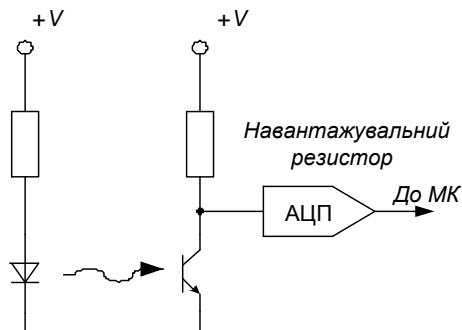


Схема забезпечує роботу оптичних приладів навіть у тому випадку, коли вихідний сигнал фоторезистора менший необхідного логічного рівня.

Рівень опорної напруги може бути змінений за допомогою потенціометра для встановлення напруги спрацьовування.



Найповільніша, оскільки кожний вимір потребує окремого аналого-цифрового перетворення.

Дана схема призводить до ускладнення всієї системи, оскільки програмне забезпечення повинно обробляти результати перетворення.

Така схема універсальна і може працювати з датчиками з різними КПС і оптичними системами.

Рисунок 3.5 – Схеми підключення оптичних датчиків до МК

Дискретні оптопари мають аналогічні схеми включення і використовуються так само, як розглянуті раніше оптрони. Проте існує

декілька додаткових зауважень. Оскільки відстань між світлодіодом і фототранзистором у такій системі більша, ніж в стандартній ІС, КПС, відповідно, нижчий. Часто з'являється необхідність підстроїти струм світлодіода або поріг чутливості для надійної роботи схеми. У деяких випадках може знадобитися лінза (об'єктив) на один з елементів оптопари для фокусування світла.

Необхідність фокусування – досить поширена проблема, що виникає при використанні дискретних елементів. Ця проблема буде особливо відчутна, якщо світлодіод і фототранзистор розташовані на різних механічних підставках.

У оптроні світлодіод і фототранзистор узгоджені за довжиною світлової хвилі. Хоча велика частина фототранзисторів і світлодіодів ІЧ-діапазону можуть працювати спільно, але узгодження максимуму їх спектральної чутливості все ж може відрізнитися. Отже, при використанні дискретних елементів вибирають світлодіод і фототранзистор, що працюють в одній і тій же області ІЧ-спектра.

На рисунку 3.5 показано три основні способи підключення оптичного датчика до МК. У всіх випадках світлодіод підключений через обмежувальний резистор, емітер фототранзистора з'єднується з землею, а колектор підключений до живлення через резистор навантаження і з'єднаний також з МК чи іншим пристроєм – компаратором або АЦП. Всі ці способи також ефективні і при інших підключеннях пристроїв керування, як, наприклад, з використанням МК для вмикання і вимикання світлодіода. Оптрони з закритим оптичним каналом працюватимуть і при іншій схемі підключення фототранзистора, наприклад, не за наведеною вище схемою зі спільним колектором, а за іншою – зі спільним емітером (живлення позитивної полярності подається на колектор, і сигнал знімається через резистор, що з'єднує емітер з землею).

На рисунку 3.6 показано, як оптрони з закритими оптичними каналами можуть бути використані для розв'язки двонаправленого сигналу між двома системами. На рисунку 3.6 показано, що вивід пристрою, що використовує інтерфейс *SPI*, потрібно ізолювати за постійним струмом від МП. Для забезпечення необхідної розв'язки використовуються два оптрони з закритими оптичними каналами. Вихід *SPI*-пристрою забезпечений буфером (для забезпечення необхідного струму світлодіода), і вихід оптрона *U1* переходить до стану низького рівня, коли *SPI*-пристрій встановлює на виводі низький логічний рівень. Вихід *U1* може бути підключений до порту МП.

Другий оптрон (*U2*) встановлює вивід порта в нуль, коли МП формує на світлодіоді сигнал низького рівня. Якщо МП не передає дані на вивід І/О, він повинен залишити світлодіод у вимкненому стані (*OFF state*), так, щоб *SPI*-пристрій міг подавати сигнали на цей вивід. Коли МП встановлює на виводі І/О низький логічний рівень, вихід оптрона *U2* також встановиться в низький рівень, щоб МП ігнорував передачі на виході *U2* в той час, коли він керує *SPI*-пристроєм. З іншого боку, може бути використаний у відповідь

