

Л7.2 Кнопки та датчики

За допомогою кнопок і простих датчиків в мікропроцесорну систему управління надходить різна інформація, яка використовується для зміни алгоритму роботи програми. Схема підключення контактного датчика до МК наведена на рис. 7.8. 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.

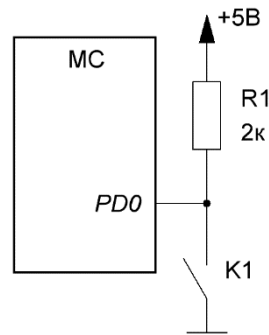


Рис. 7.8. Підключення контактного датчика до МК

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 рис. 7.9.

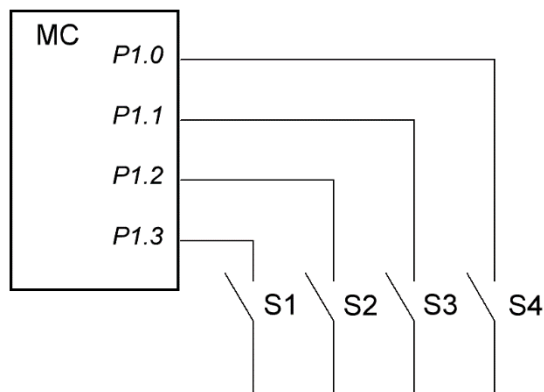


Рис. 7.9. Підключення кнопок або простих датчиків до МК

У схемі, що зображена рис. 7.10, при натисканні 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.

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

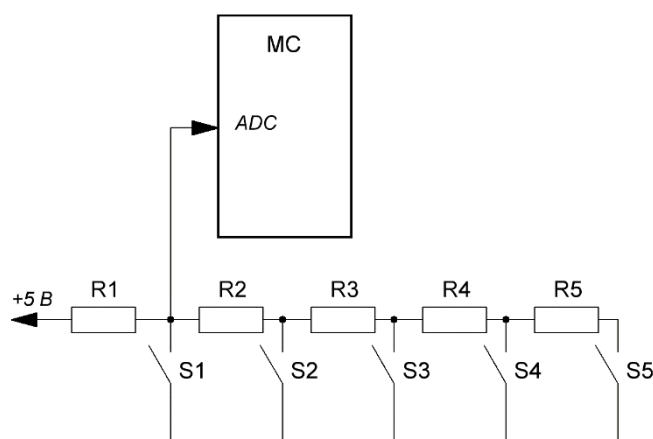


Рис. 7.10. Підключення кнопок зміною напруги на аналоговому вході МК

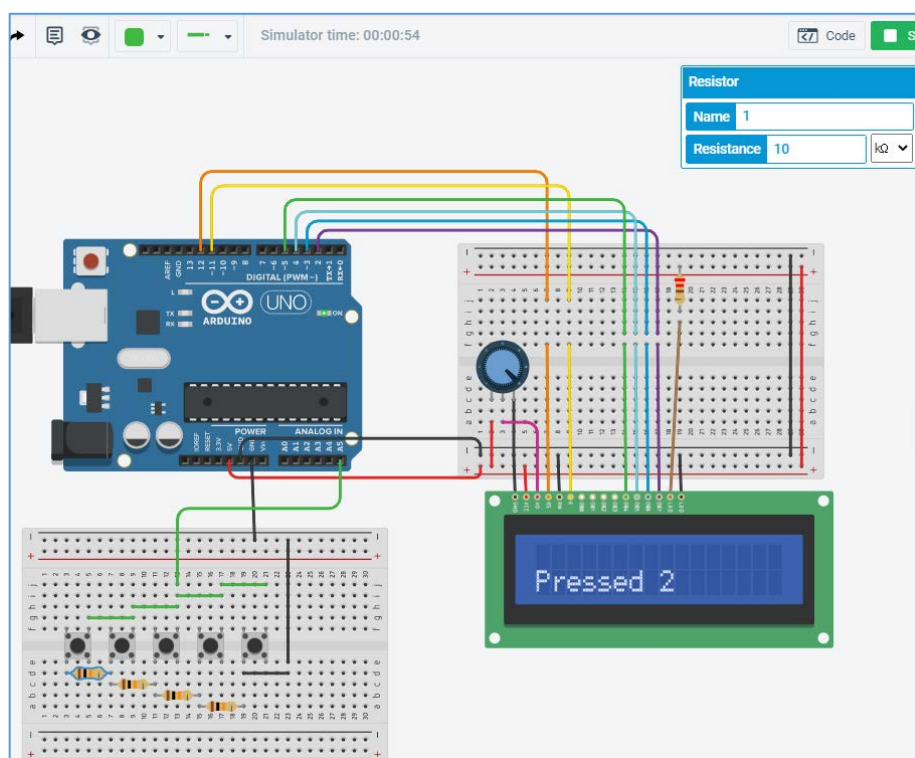


Рис. 7.11. Підключення кнопок зміною напруги на аналоговому вході A5 Arduino

Лістинг програми до проекту, що зображений на рис. 7.11

```

#include <LiquidCrystal.h>
LiquidCrystal lcd (12, 11, 5, 4, 2, 3);
int a=0;

void setup (){
  lcd.begin (16, 2);
  pinMode (A5, INPUT_PULLUP); // sets analog pin for input
}
// returns the button number pressed, or zero for none pressed
int readButtons (int pin) // int pin is the analog pin number to
read
{
  int b,c = 0;
  c=analogRead (pin); // get the analog value
  if (c>1000) {
    b=0; // buttons have not been pressed
  }
  else if (c>440 && c<470){
    b=1; // button 1 pressed
  }
  else if (c<400 && c>370) {
    b=2; // button 2 pressed
  }
  else if (c>280 && c<310) {
    b=3; // button 3 pressed
  }
  else if (c>150 && c<180){
    b=4; // button 4 pressed
  }
  else if (c<20) {
    b=5; // button 5 pressed
  }
  return b;
}

void loop () {
  a=readButtons (19);
  lcd.clear ();
  if (a==0) // no buttons pressed
  {
    lcd.setCursor (0,1);
    lcd.print ("Press a button");
  }
  else if (a>0) // someone pressed a button!
  {
    lcd.setCursor (0,1);
    lcd.print ("Pressed button");
    lcd.print (a);
  }
  delay (1000); // give the human time to read LCD
}

```

Блок-схема 12-клавішної клавіатури зі скануванням показана на рис.7.12. Клавіші розташовані у вузлах матриці, у якої чотири лінії рядків і три лінії стовпців. На лінії стовпців 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 +5 В через резистори). Якщо ж клавіша натискається, і на лінії стовпця, відповідного натиснутій клавіші, все ще нуль, то лінія рядка також стає рівною нулю. Знаючи номери стовпця і рядка, можна отримати позицію натиснутої клавіші.

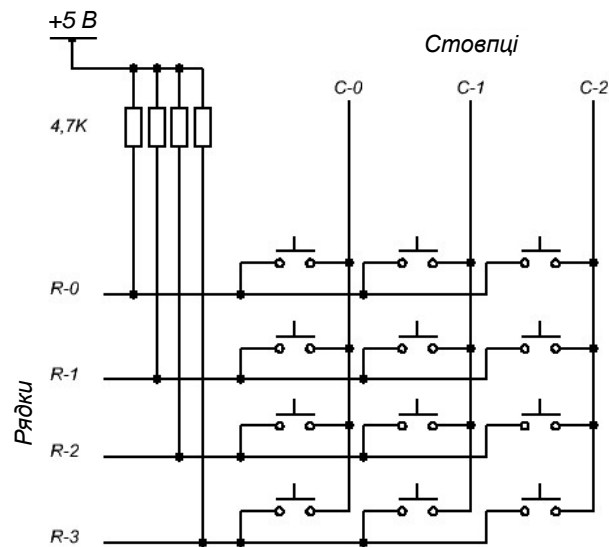


Рис. 7.12. Матрична клавіатура 4×3

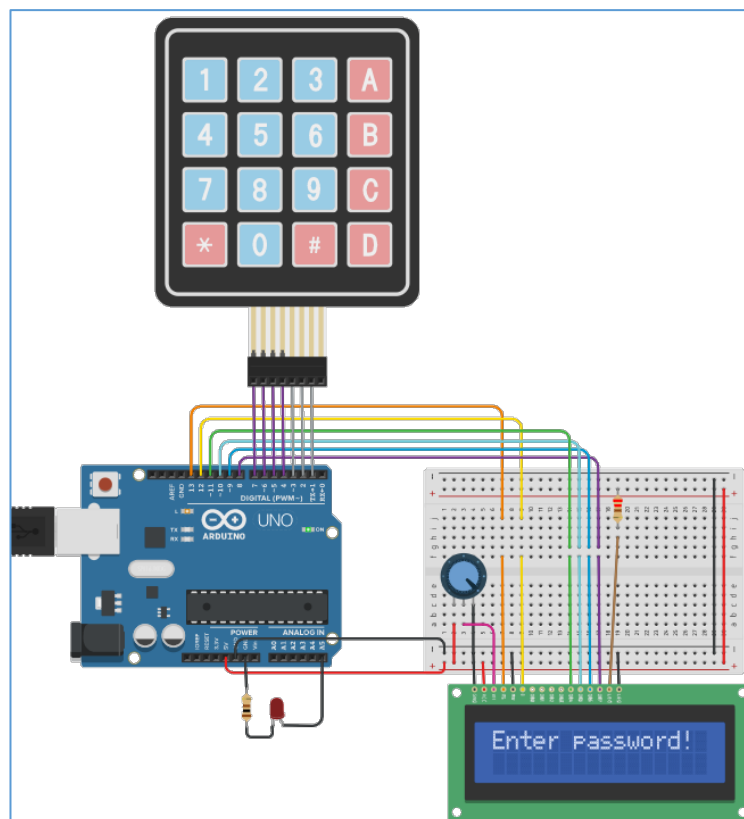


Рис. 7.13. Реалізація проекту кодового замку з матричною клавіатурою

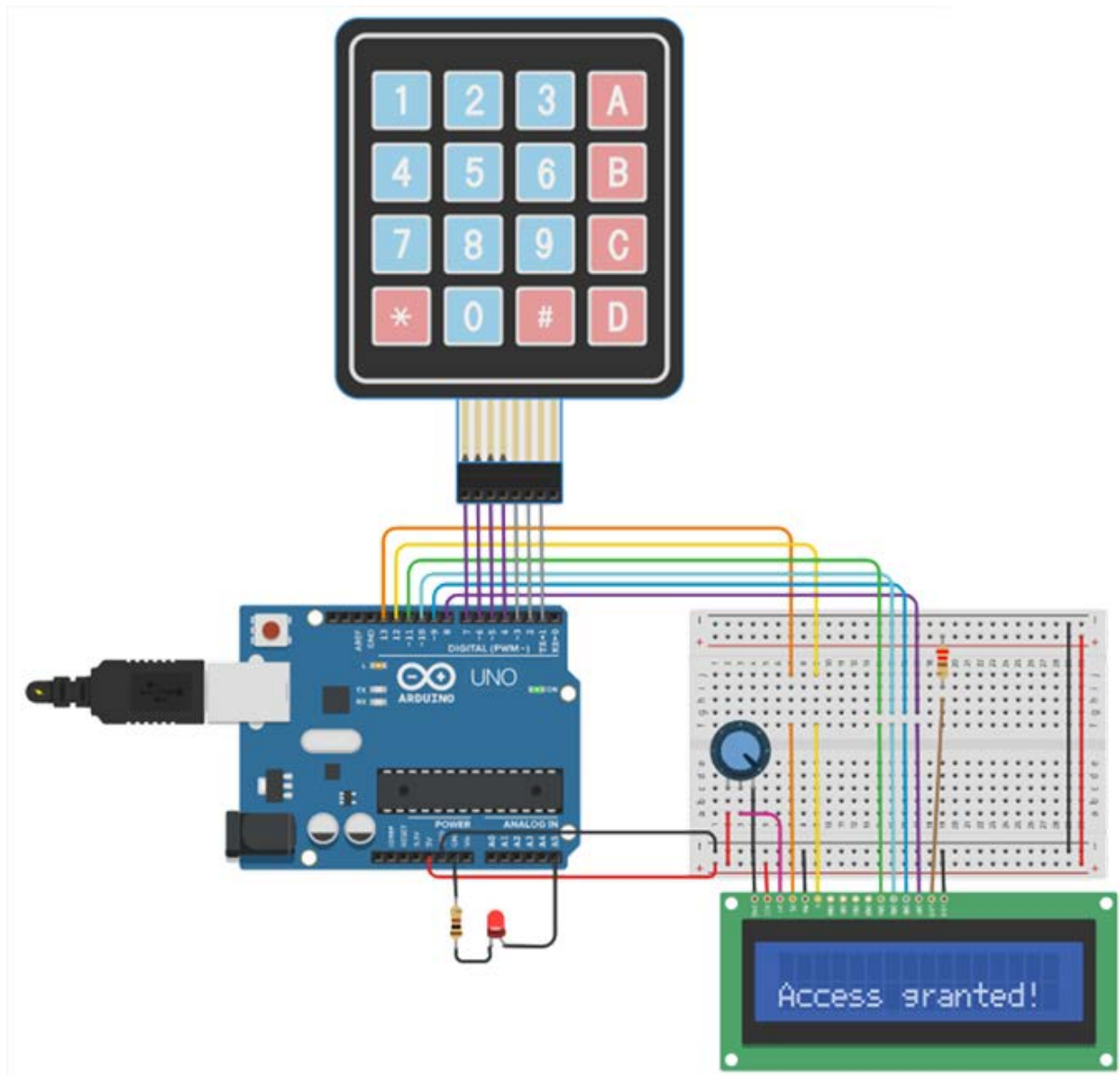


Рис. 7.14. Реалізація проекту кодового замку з матричною клавіатурою. Введений код вірний

Для роботи з матричною бібліотекою застосовується бібліотека <Keypad.h>. Для роботи з клавіатурою потрібно створити об'єкт Keypad та вказати виводи рядків та стовпців, кількість рядків та стовпців

```
Keypad(makeKeymap(userKeymap), row[], col[], rows, cols)
```

Для визначення кнопки, що натиснута використовується функція

```
char getKey()
```

```
#include <LiquidCrystal.h>
#include <Key.h>
#include <Keypad.h>
```

```
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(13, 12, 11, 10, 9, 8);
```

```

const byte ROWS = 4; // число рядків клавіатури
const byte COLS = 3; // число стовпців клавіатури
char hexaKeys[ROWS][COLS] = {
    {'1','2','3'},
    {'4','5','6'},
    {'7','8','9'},
    {'*','0','#'}}
};

byte rowPins[ROWS] = {7, 6, 5, 4}; // виводи керування рядками
byte colPins[COLS] = {3, 2, 1}; // виводи керування стовпцями
char pass[4] = {'7', '3', '1', '5'}; // вірний пароль
char buttons[5]; // масив натиснутих кнопок
int k = 0; // лічильник натиснень

Keypad customKeypad=Keypad(makeKeymap(hexaKeys),rowPins,
colPins,ROWS,COLS);

void setup() {
    // set up the LCD's number of columns and rows:
    lcd.begin(16, 2);
    // Print a message to the LCD.
    //lcd.print ("Press any key!");
    pinMode (A5, OUTPUT);
}

void loop() {
    lcd.setCursor(0, 0); // курсор на початок першого рядку
    lcd.print("Enter password!");
    lcd.setCursor(0, 1); // курсор на початок другого рядка
    char customKey = customKeypad.getKey();
    if (customKey) {
        buttons[k] = customKey; // зберігаємо значення кнопки у масиві
        lcd.setCursor(k, 1);
        lcd.print('*'); // виводимо символ '*' замість значення кнопки
        k = k + 1; // збільшуємо лічильник натиснень на 1
        if (k == 4) {
            if(buttons[0]==pass[0]&&buttons[1]==pass[1]&&buttons[2]==
pass[2]&&buttons[3]==pass[3]){
                lcd.clear();
                lcd.setCursor(0, 1);
                lcd.print("Access granted!"); // якщо збіг паролю
                digitalWrite (A5, HIGH);
                delay (1000);
                digitalWrite (A5, LOW);
                lcd.clear();
                k=0;
            }
            else {

```

```

        lcd.clear();
        lcd.setCursor(0, 1);
        lcd.print("Access denied!"); // якщо пароль не вірний
        lcd.clear();
        k=0;
    }
}
}
}
}

```

У клавіатурі з кодуванням застосовують спеціалізовані мікросхеми, які виявляють натискання клавіші і передають її код. Прикладом такого пристрою є мікросхема MM74C922 (National Semiconductors).

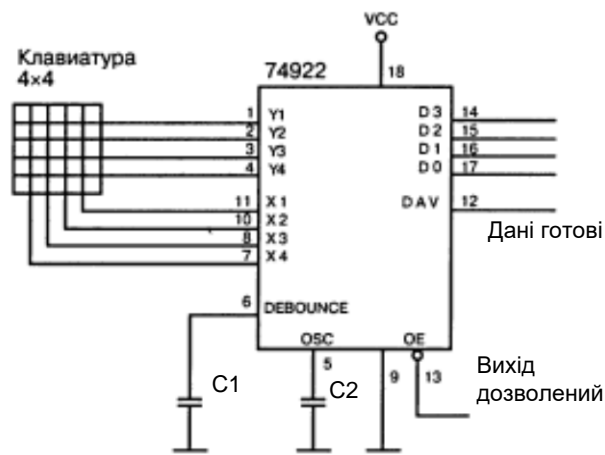


Рис.7.15. Клавіатура 4x4 на ІМС 74922

Сенсорна кнопка реалізується досить просто. Від порту мікроконтролера на плюс живлення підключається резистор великого опору (pullup). До даного порту також підключається площадка, що проводить струм (сенсорна «кнопка»), яка ізолюється від прямого дотику діелектриком (скотч, лак, самоклеїтка)

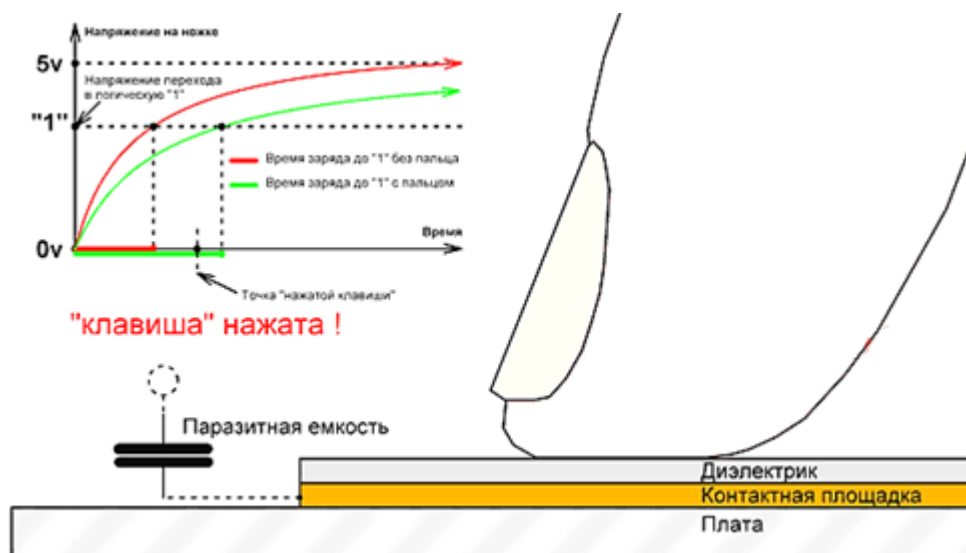


Рис. 7.16. Конструкція та принцип роботи сенсорної кнопки

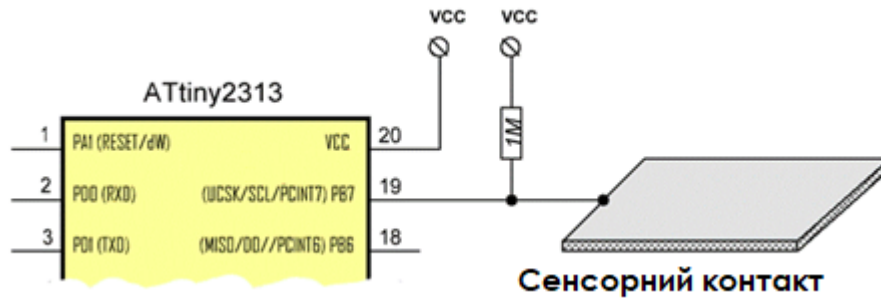


Рис. 7.17. Підключення сенсорної кнопки до мікроокнтролера

Алгоритм визначення натиснення кнопки

1. Порт переводиться на вивід (1 -> DDRxn) та скидається в нуль (0 -> PORTxn);
2. Порт переводиться на ввід, у Z-стан (0 -> DDRxn). Так як на порту був 0, а елементи електричного кола порту мають певну ємність, то починається процес заряду цієї ємності через зовнішній резистор підтяжки;
3. З моменту переведення порту на ввід починається відлік часу з контролем стану порту (PINxn), як тільки на порту появиться рівень «1» (PINxn = 1) – зупиняється лічильник;
4. Значення лічильнику є ємністю ніжки у відносних одиницях. За величиною даного значення можна визначити чи є дотик до сенсору чи ні.