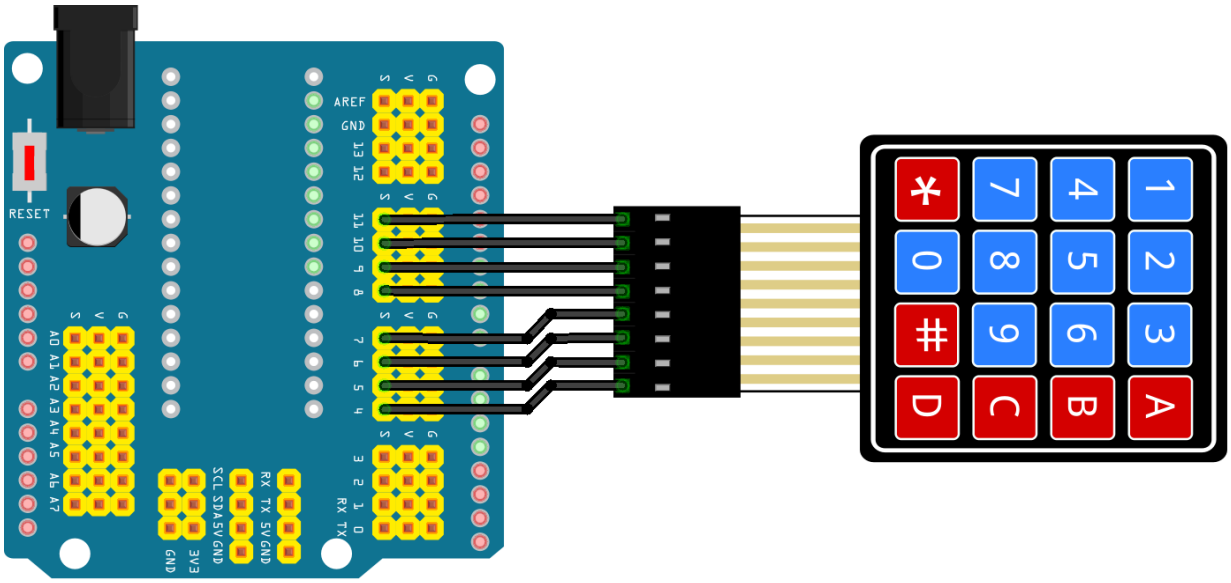
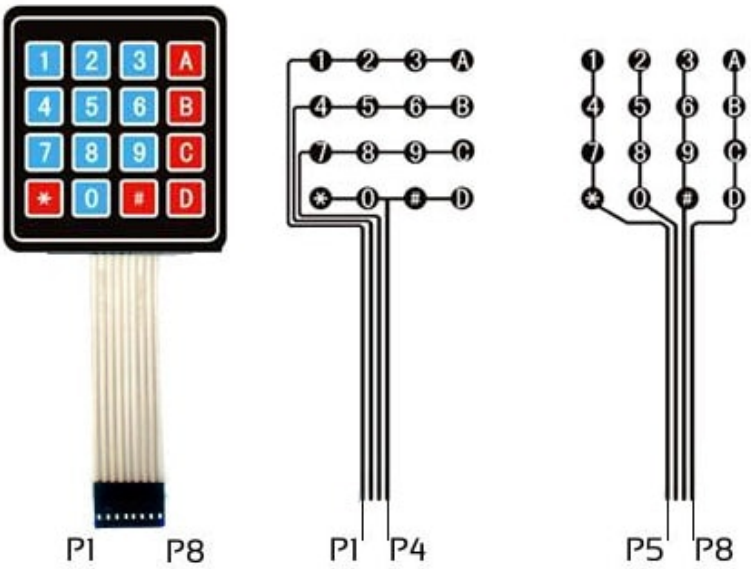
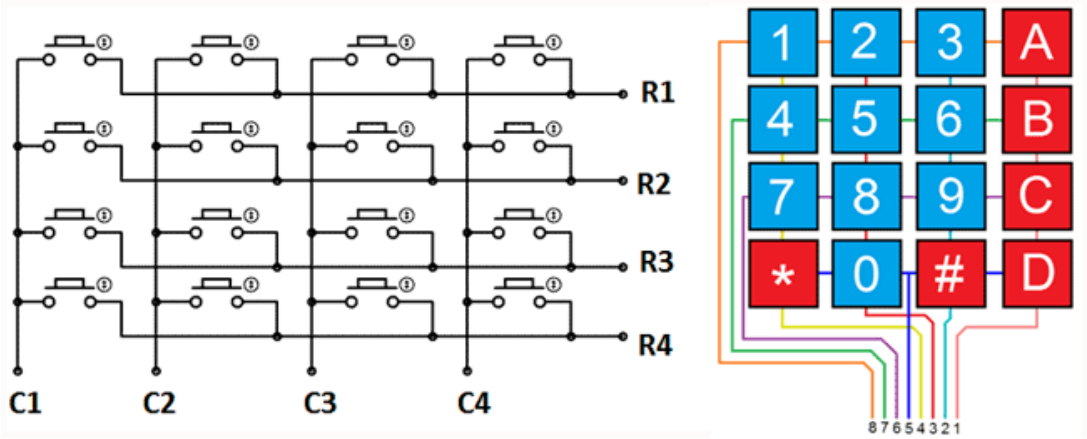
Task.1. Connect the circuit as shown in the picture:



Inside a matrix keyboard, all switches are connected to each other with conductive traces forming a matrix of a 4×4 grid. Using 16 individual push buttons requires 17 input pins - one for each key and a ground pin in order to make them work. With matrix arrangement, the required number of Arduino pins, to scan through the pad, equals 8 - four for columns and four for rows.





The reading procedure is as follows:

1. The Arduino board sets all the column and row lines to input.
2. It picks a row and sets it HIGH.
3. After that, it checks the column lines one at a time.
4. If the column connection stays LOW, the button on the row has not been pressed.
5. If it goes HIGH, the microcontroller knows which row was set HIGH, and which column was detected HIGH when checked.
6. Finally, it knows which button was pressed that corresponds to the detected row & column.

Keypad is a library for using matrix keypads with Arduino. The user can add this library to the Arduino IDE using *Library Manager* (*Sketch -> Include Library -> Manage Libraries…*).

```
#include <Keypad.h>

#define BAUDRATE 115200
#define ROWS 4
#define COLS 4
char keys[ROWS][COLS] = {
```

```
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'D','0','E','F'}
};
byte rowPins[ROWS] = {11,10,9,8};
byte colPins[COLS] = {7,6,5,4};

Keypad keyb = Keypad(makeKeymap(keys),rowPins,colPins,ROWS,COLS);

void setup() {
 Serial.begin(BAUDRATE); }

void loop() {
 char key = keyb.getKey();
 if(key) {
  Serial.print("Key pressed: ");
  Serial.println(key);
 }
 delay(100);
 }
```

Task.2. Replace the *loop()* function in the Task 1 program with the following code.

```
void loop() {
char key = keyb.getKey();
if(key) {
  switch(key) {
    case '1':
     Serial.println("Key 1");
     break;

    case 'A':
     Serial.println("Key A");
     break;

    default:
     break; }
 }
}
```

Task.3. Add the LCD to Your project. Display all button codes on the LCD.

Task.4. Using the circuit from Task no 1 create a program for the Arduino board that allows You to move a selected character (eg. '*') on the LCD screen. All places on the LCD should be accessible for the character.

Task.5. The *`getState()`* method returns the current state of any of the keys. The four states are *`IDLE`*, *`PRESSED`*, *`RELEASED`*, and *`HOLD`*. Propose a program that presents how this method works.

Task.6. Build a prototype of an alarm keypad. It should be able to arm and disarm the system. The information about successful arming/disarming should be sent to the computer. Implement basic interface in Node-Red.

For those interested:

1. Arduino Playground:

   playground.arduino.cc/Code/Keypad/

2. Last Minute Engineers tutorial:

   lastminuteengineers.com/arduino-keypad-tutorial/