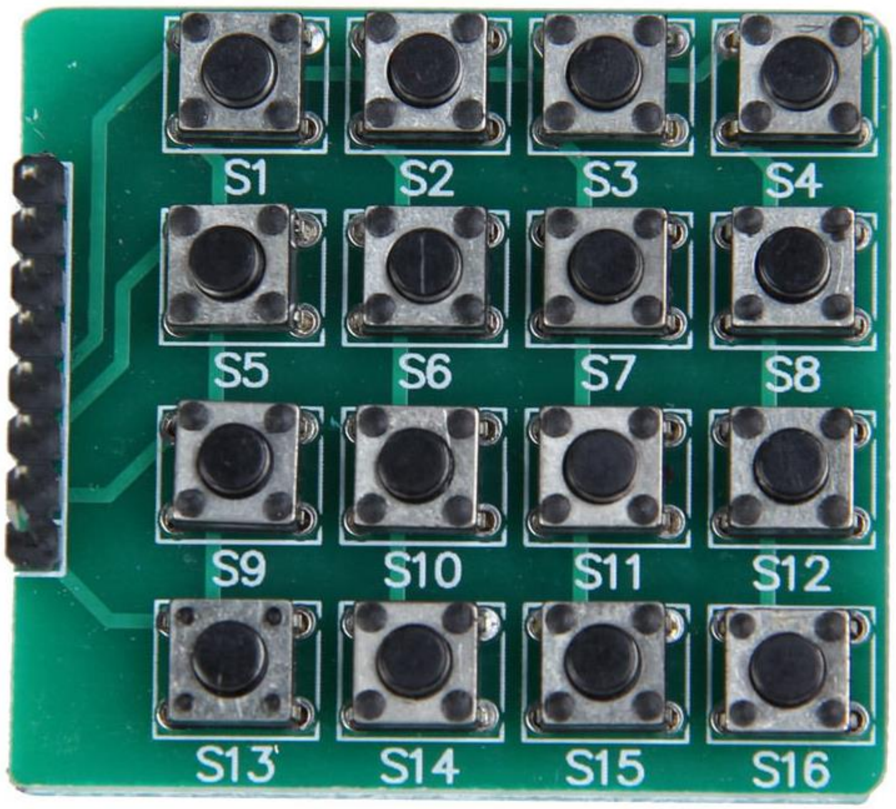
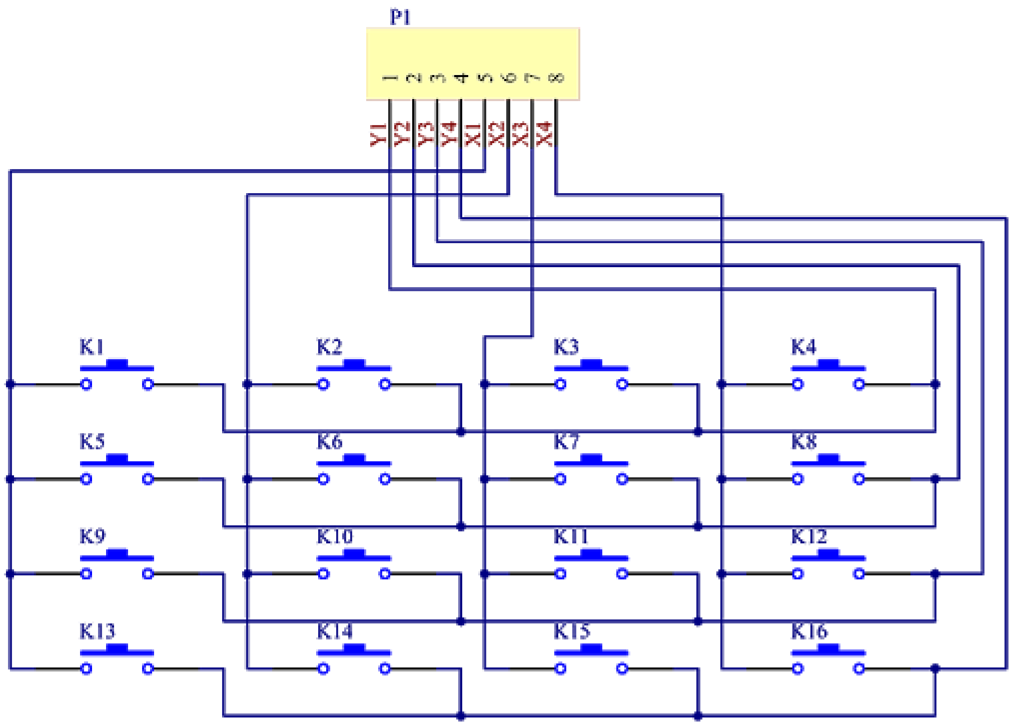
KEYBOARD MODULE



Whether it's a calculator or the keypad of a building, we commonly use numeric keypads. The 4×4 numeric keypad is a matrix of 16 buttons whose states can be detected by a microcontroller.

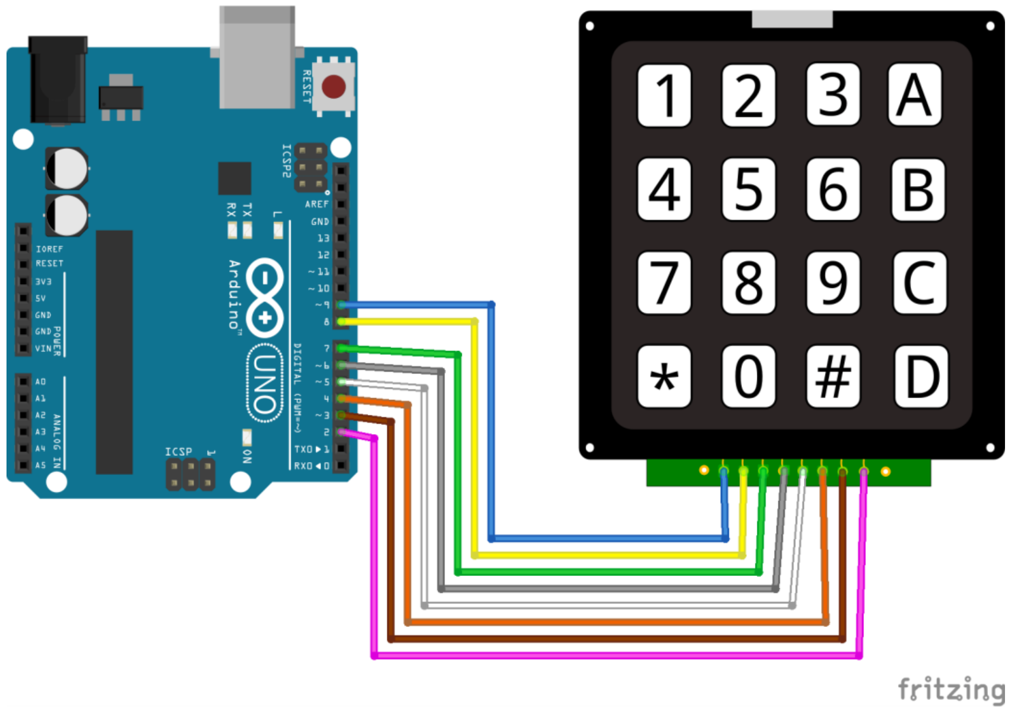
Principle of operation

Numerical keypad is a set of 16 buttons that are arranged in a matrix, i.e. all the buttons in a column are linked to one input and all the buttons in a row are linked to another. When a button is pressed, the input corresponding to the line is connected to the input corresponding to the column, which closes the circuit. The advantage of this type of assembly is that 16 buttons can be managed with only 8 microcontroller inputs.



Schematic

The numeric keypad uses 8 pins of the Arduino. It is possible to connect them to any pin. Pins 0 and 1, which are used for serial connection via the USB port, should be avoided.



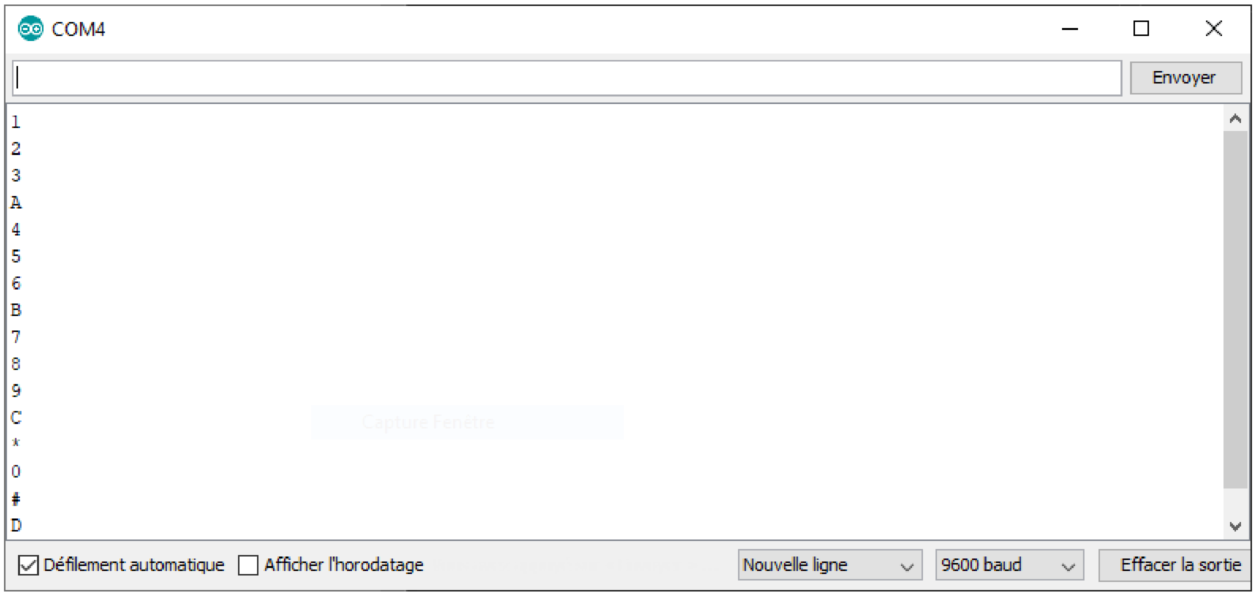
Code

To manage the numerical keypad, the principle is to switch each entry in the columns to the high state and to read the value of the entries corresponding to the lines. If a line is in the same state as the column, a button is pressed. In practice we can use the Keypad.h library, which allows us to manage a matrix of buttons of any size.

```
//Libraries
#include <Keypad.h>//https://github.com/Chris--A/Keypad

//Constants
#define ROWS 4
#define COLS 4

//Parameters
const char kp4x4Keys[ROWS][COLS]  = {{'1', '2', '3',
'A'}, {'4', '5', '6', 'B'}, {'7', '8', '9', 'C'}, {'*',
'0', '#', 'D'}};
byte rowKp4x4Pin [4] = {9, 8, 7, 6};
byte colKp4x4Pin [4] = {5, 4, 3, 2};

//Variables
Keypad kp4x4  = Keypad(makeKeymap(kp4x4Keys),
rowKp4x4Pin, colKp4x4Pin, ROWS, COLS);

void setup() {
  //Init Serial USB
  Serial.begin(9600);
  Serial.println(F("Initialize System"));
}

void loop() {
  readKp4x4();
}

void readKp4x4() { /* function readKp4x4 */
  //// Read button states from keypad
  char customKey = kp4x4.getKey();
  if (customKey) {
    Serial.println(customKey);
  }
}
```

Result

When a key on the keyboard is pressed, we observe that the associated character is correctly displayed in the serial monitor.



```
1
2
3
A
4
5
6
B
7
8
9
C
*
0
#
D
```

Key Features:

- **16-Button Layout**: Provides a wide range of input options for your projects.
- **Compact Design**: Perfect for space-constrained designs or compact systems.
- **Simple Integration**: Easily connects to Arduino and Raspberry Pi, with minimal setup required.
- **Durable Construction**: Designed for long-term reliability, even with repeated use.
- **Low Power Consumption**: Ideal for energy-efficient, battery-powered applications.

Applications:

- **Security Systems**: Create secure password-based access systems and entry controls.
- **Control Panels**: Build custom interfaces for a variety of electronics projects.
- **Robotics**: Great for input methods in robot control and automation applications.
- **DIY Electronics**: A must-have for hobbyists and creators building interactive projects.
- **Microcontroller Projects**: Works seamlessly with Arduino, Raspberry Pi, and other microcontroller platforms.

REFERENCE:

https://www.aranacorp.com/en/using-a-4x4-digital-keypad-with-arduino/