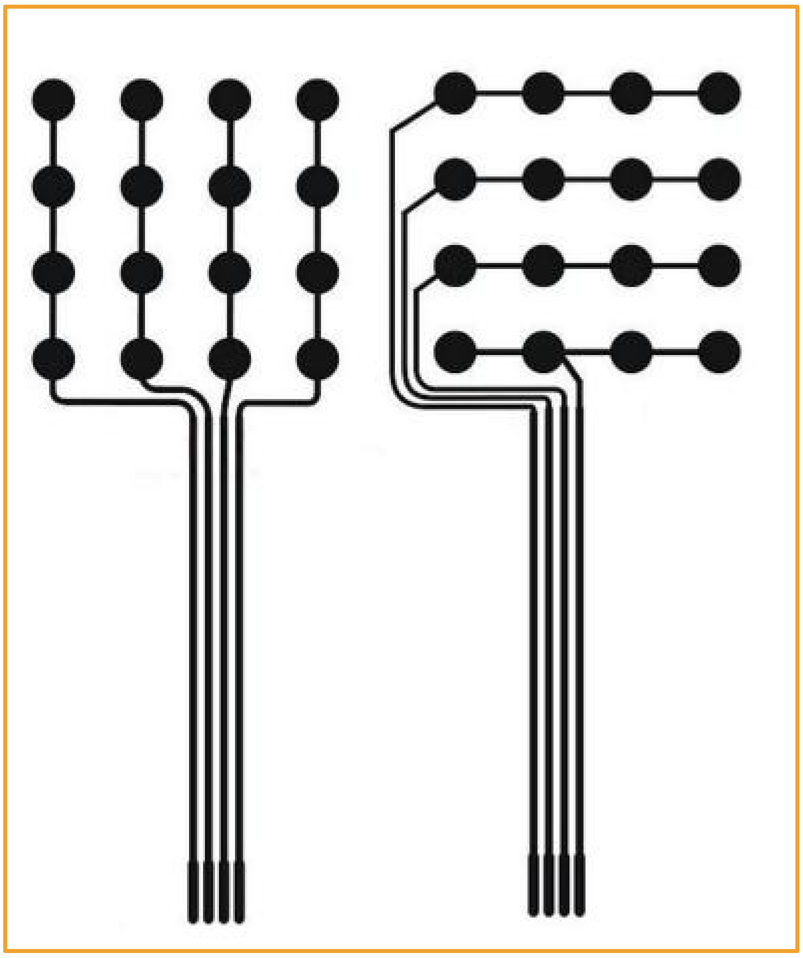
Matrix keyboard experiment

Introduction to Matrix keyboard

The matrix keyboard used in this experiment contains 16 keys in total, 4 rows and 4 columns. The 4 keys of each raw are linked together to form a line, and so does each column, so there will be a total of 8 lines, namely, 4 rows and 4 columns.

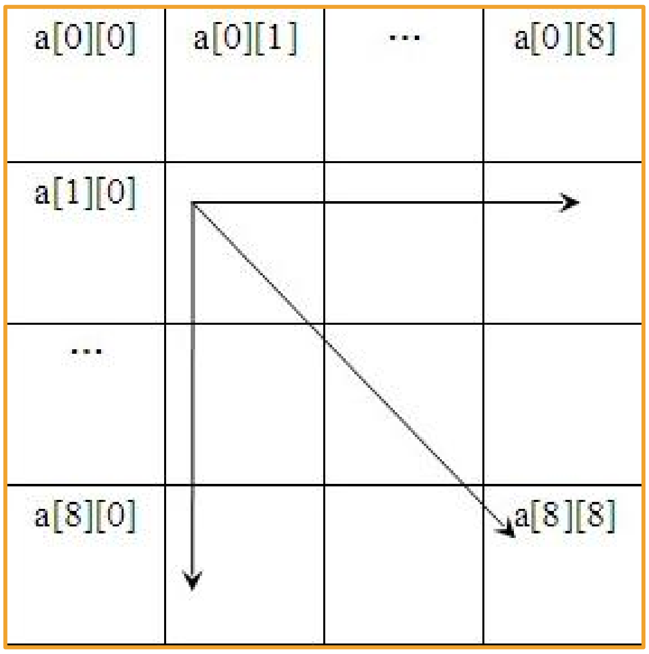
4 * 4 Matrix Keyboard Schematic Diagram



Detection Principle

First we send a low level to a column, the rest of the columns remain high level, then immediately take turns to detect each row. If low level is not detected, it turns out that no keys of the very column are pressed, then we continue taking turns to send low level to the rest of the columns and scan them; If low level is detected in a row, then there must be a pressed key on the row, while the low level column is the one where a key is pressed. When the row and column are determined, the key is affirmed. The speed of scanning and detection by Arduino is fast enough, so you do not have to worry about missing the keys you pressed.

In the program, we define and use a two-dimensional array of characters. An array is designed to number a simple type of data, a[0], a[1], a[2], a[3].... However, this data type is not convenient in some occasions. For example, we want to define an array, recording the results of 9 x 9 multiplication table. The best way to record the results of the 99 multiplication table is not to define an array of length 81, but an array of 9 rows and 9 columns. Therefore, the reference array has two subscripts which are row and column. This kind of array is called two dimensional array, and so on, three dimensional array etc.



Obviously, two dimensional array is such convenient for recording key characters data. For example, we need to know the data of the second row and third column on the keyboard as long as we reference into hexaKeys[1][2] in the defined two dimensional array hexaKeys. Because counting from zero, not one. At the beginning of the definition of the two-dimensional array, we can assign it values. Taking the above 4 x 4 keyboard as an example, the format is:

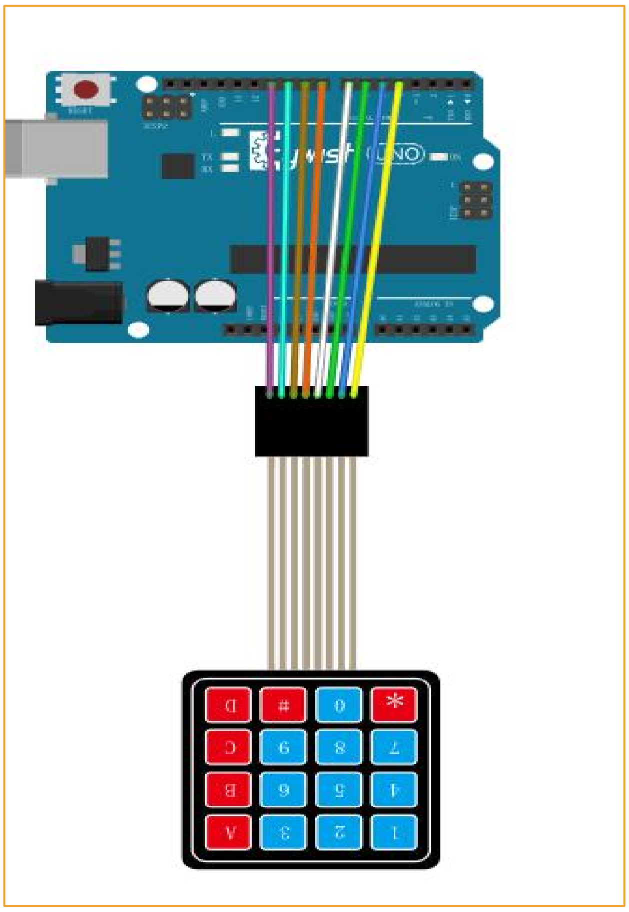
```
char hexaKeys[ROWS][COLS] = {
{'1','2','3','A'},
{'4','5','6','B'},
{'7','8','9','C'},
{'*','0','#','D'}
};
```

Component List

- Keywish Arduino UNO R3 mainboard
- Breadboard
- USB cable
- Matrix Keyboard*1
- Several jumper wires

Wiring of Circuit

arduino Uno	Matrix keypad
4	1
5	2
6	3
7	4
8	5
9	6
10	7
11	8

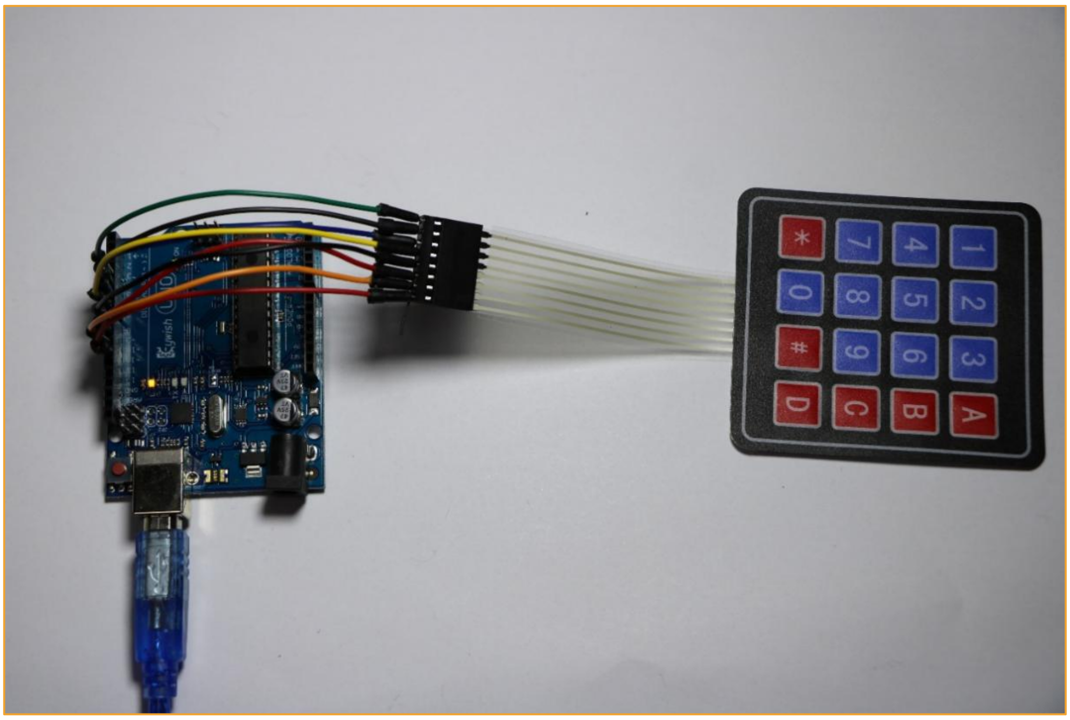

RESET
ICSP2
TX
RX
UNO
ICSP
ANALOG IN

Code

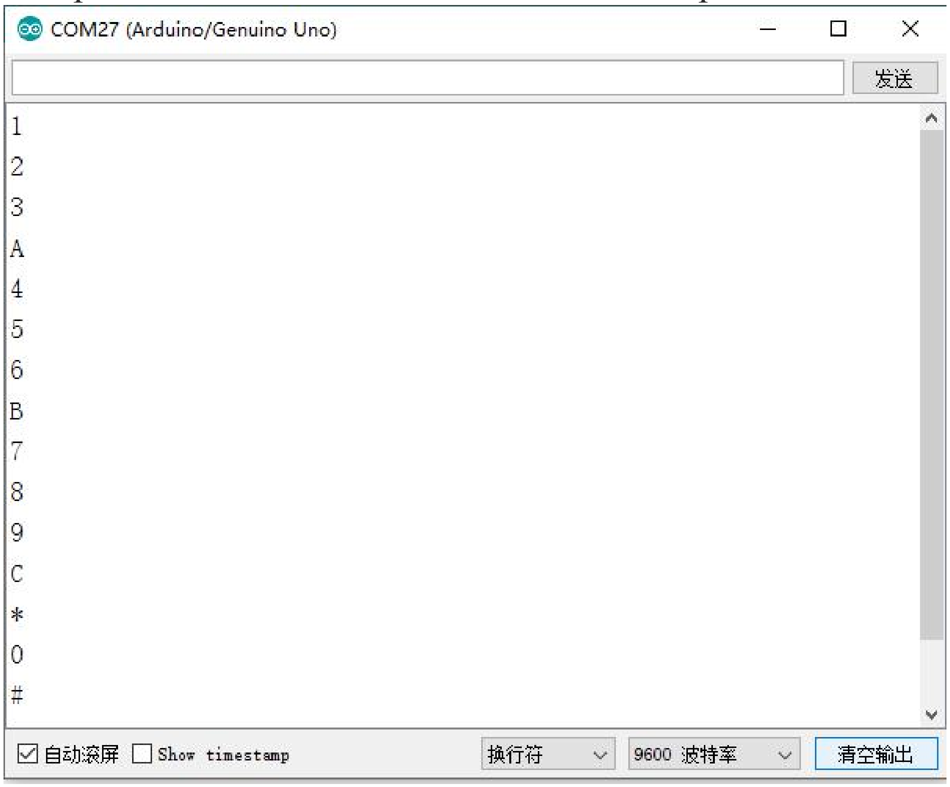
```
#include "Keypad.h"

#define   ROW_1   4
#define   ROW_2   5
#define   ROW_3   6
#define   ROW_4   7
#define   COL_1   8
#define   COL_2   9
#define   COL_3   10
#define   COL_4   11

const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns

char hexaKeys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
//connect to the row pinouts of the keypad
byte rowPins[ROWS] = {ROW_1, ROW_2, ROW_3, ROW_4};
//connect to the column pinouts of the keypad
byte colPins[COLS] = {COL_1, COL_2, COL_3, COL_4};

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

```
void setup(){
   int i ;
    for(i=0 ; i< ROWS ; i++)
    {
        pinMode(rowPins[i],OUTPUT);
        pinMode(colPins[i],OUTPUT);
    }
    Serial.begin(115200);
}

void loop(){
    char customKey = customKeypad.getKey();
    if (customKey){
        Serial.println(customKey);
    }
}
```

Experiment Result



The result of this experiment is that we can see the button we pressed on the serial port.



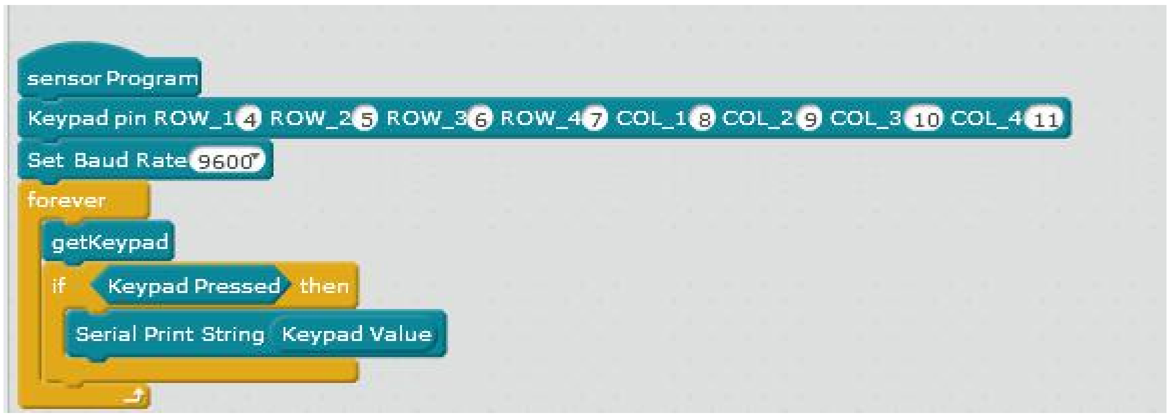
COM27 (Arduino/Genuino Uno)

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

☑ 自动滚屏 ☐ Show timestamp 换行符 9600 波特率

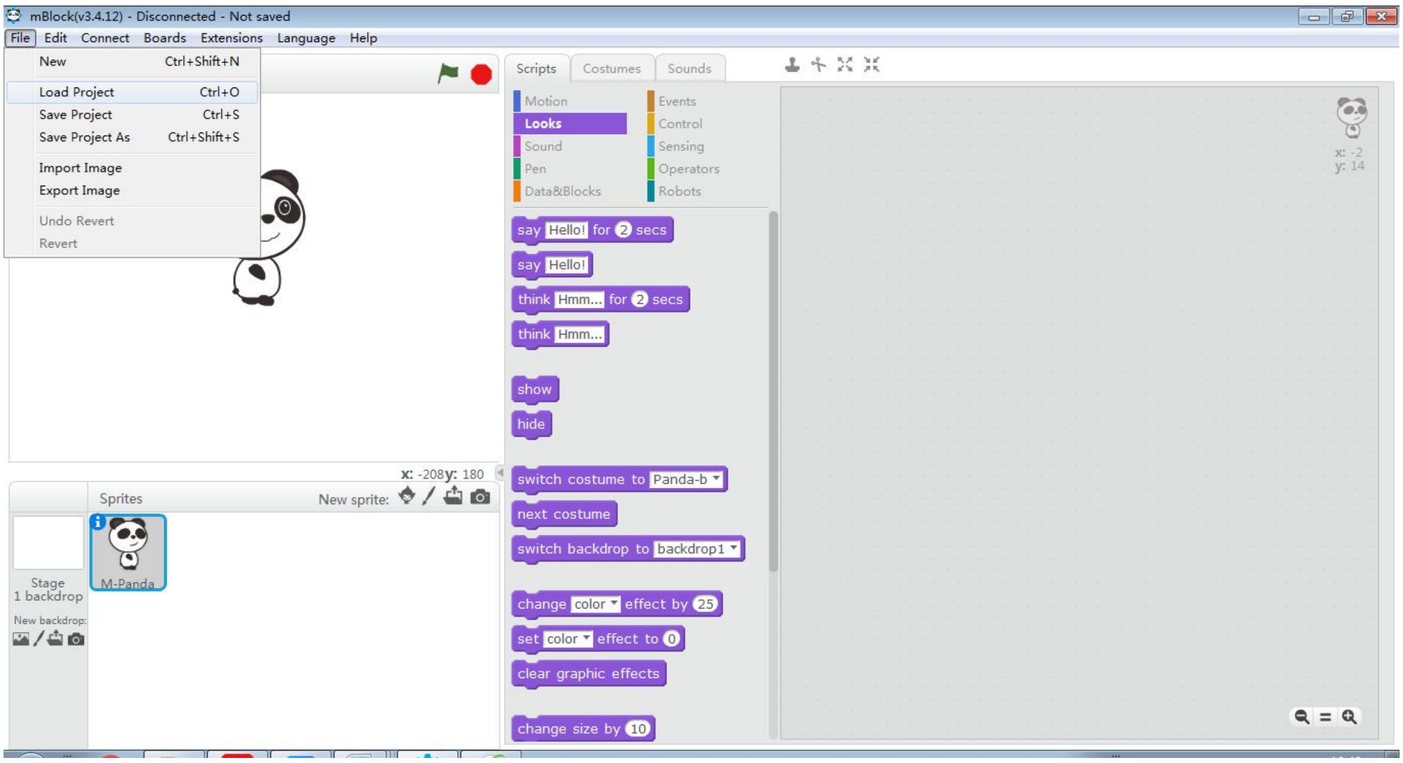
MBlock graphical programming program

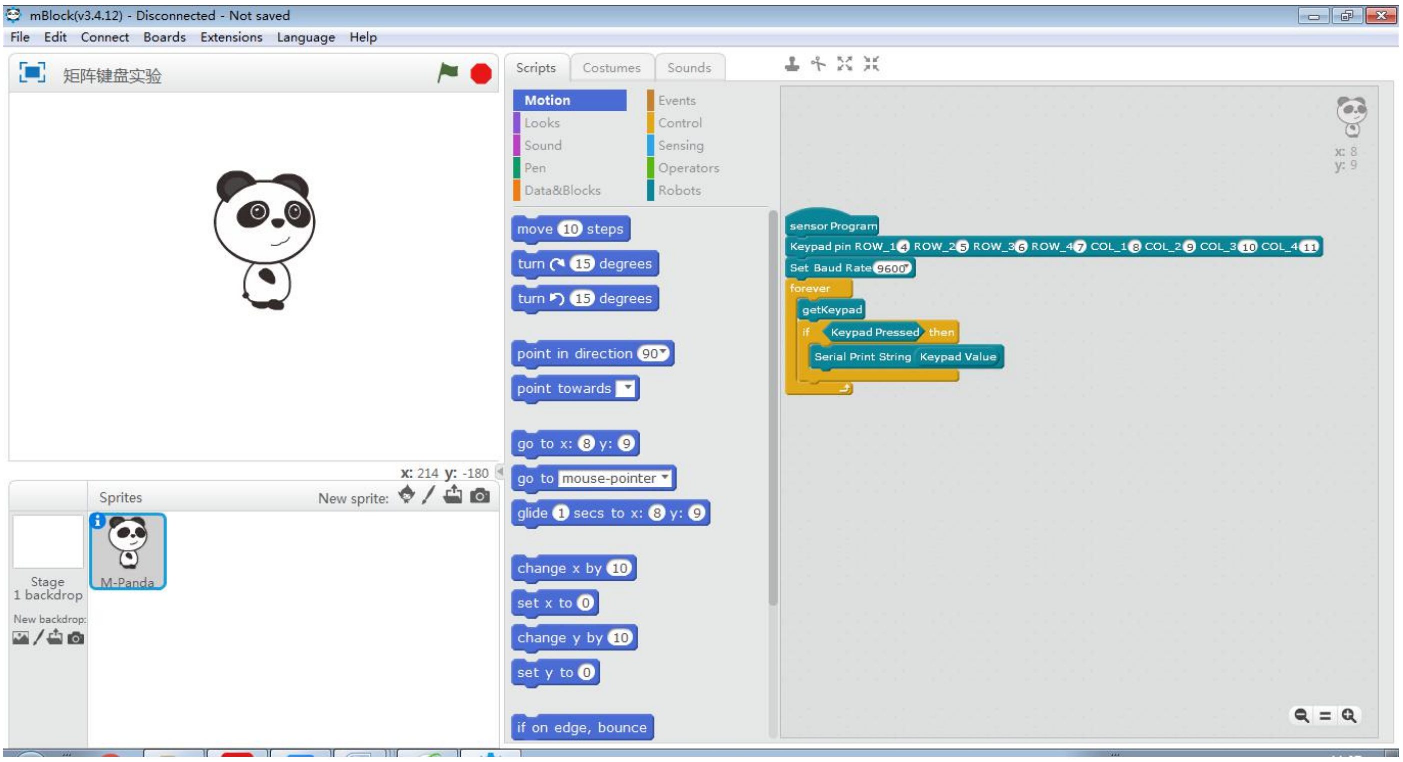
The program prepared by mBlock is shown in the figure below:



```
sensor Program
Keypad pin ROW_1 4 ROW_2 5 ROW_3 6 ROW_4 7 COL_1 8 COL_2 9 COL_3 10 COL_4 11
Set Baud Rate 9600
forever
    getKeypad
    if Keypad Pressed then
        Serial Print String Keypad Value
```

You can also open the program file directly with mblock, which is a. Sb2 file. Here are the steps to open it:





```
sensor Program
Keypad pin ROW_1 4 ROW_2 5 ROW_3 6 ROW_4 7 COL_1 8 COL_2 9 COL_3 10 COL_4 11
Set Baud Rate 9600
forever
  getKeypad
  if Keypad Pressed then
    Serial Print String Keypad Value
```