

## Лабораторная работа №4

### Тема: «Подключение матрицы кнопок к микроконтроллеру»

**Цель работы:** приобрести практические навыки по устранению дребезга контактов при подключении матричных клавиатур к микроконтроллерам.

### Последовательность выполнения работы:

1. Изучить теоретические сведения приведенные в лабораторной работе.
2. Собрать схемы на макетной плате для приведенных примеров.
3. Запрограммировать микроконтроллер согласно тексту программы указанному в примере.
4. Выполнить задание для самостоятельной работы.

## СОДЕРЖАНИЕ ОТЧЕТА

- Название лабораторной работы, ее цель.
- Задание на лабораторную работу (по варианту).
- Схемы подключения к микроконтроллеру.
- Программный код для скетчей.
- Вывод о проделанной работе.

### Теоретические сведения

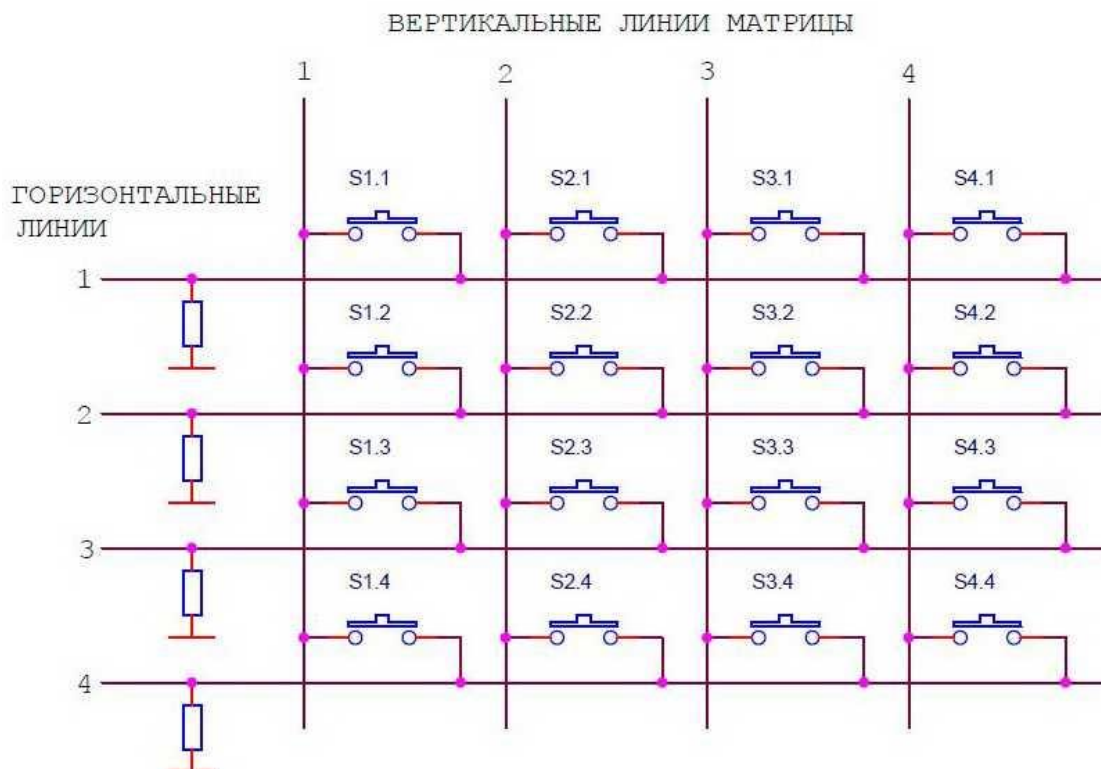
Научимся подключать к плате Ардуино матрицу кнопок, использовать для генерации звука стандартную функцию `tone()`.

В предыдущих лабораторных мы подключали кнопки к контроллеру, используя для каждой отдельный вывод. Если необходимо использовать в системе большое число кнопок, такое решение может оказаться неприемлемым. Просто не хватит выводов микроконтроллера.

Существует другой способ подключения кнопок к плате Ардуино – объединение кнопок в матрицу.

### Подключение матрицы кнопок к микроконтроллеру.

Вот пример схемы такой матрицы. Для подключения 16 кнопок требуется только 8 выводов микроконтроллера.



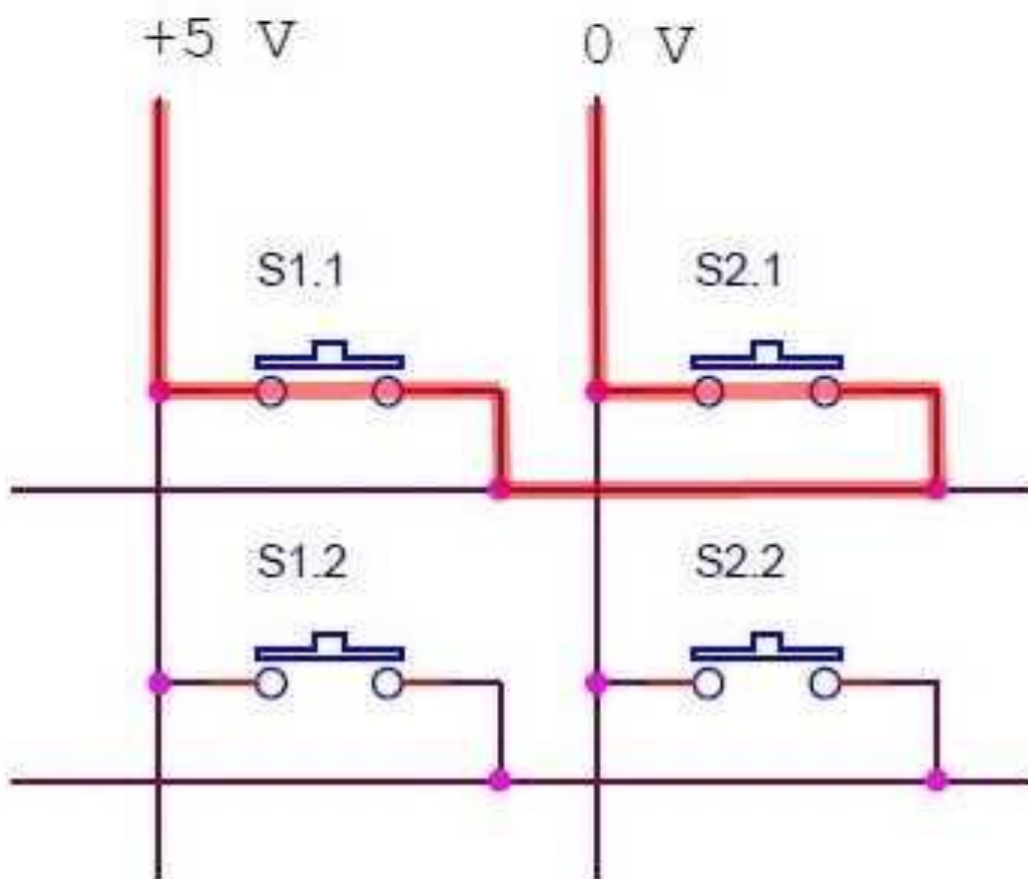
Кнопки подключены к вертикальным и горизонтальным линиям матрицы. Состояние кнопок для каждой вертикальной линии проверяется отдельно. На вертикальную линию подается сигнал высокого уровня (5 В) и считываются состояния горизонтальных линий. Высокий уровень в горизонтальной линии покажет, что соответствующая кнопка нажата. Далее проверяются остальные вертикальные линии.

Резисторы обеспечивают нулевое напряжение на входах микроконтроллера при разомкнутых кнопках.

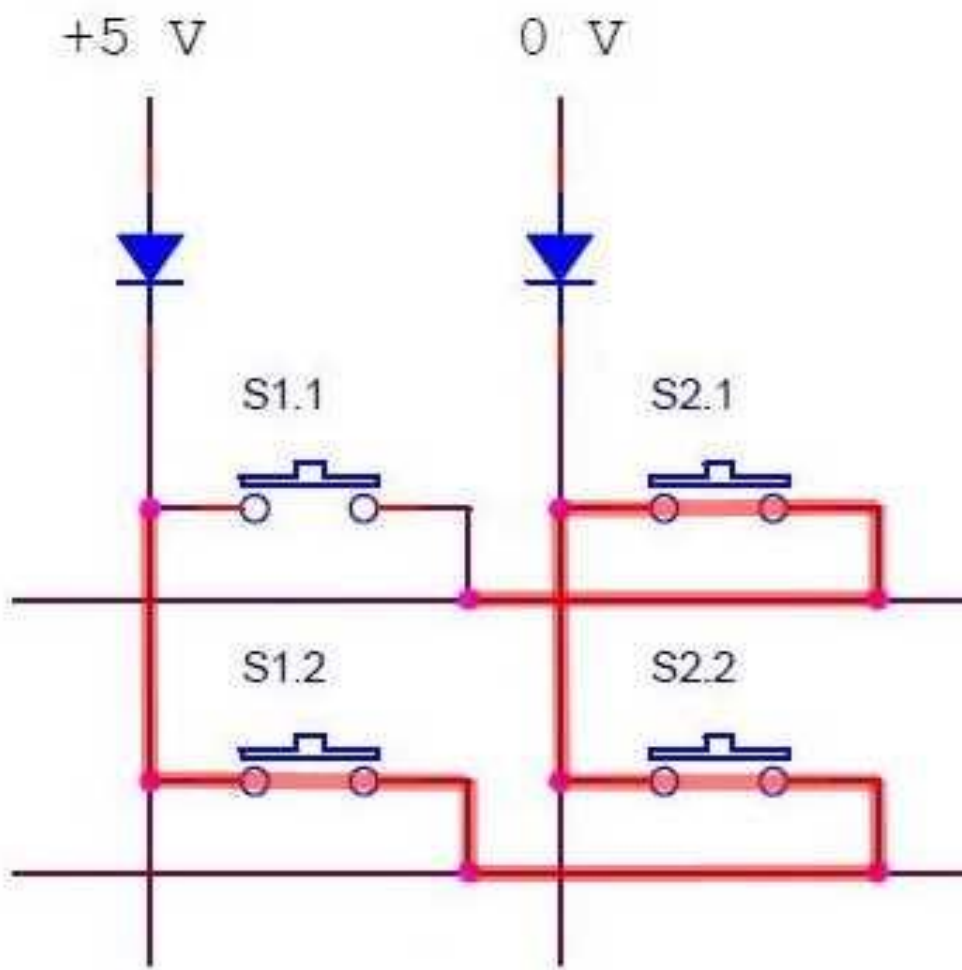
Такая схема часто используется для подключения матрицы кнопок к плате Ардуино в практических приложениях. Что меня удивляет. Ведь эта схема работает некорректно при одновременном нажатии двух и более кнопок из одной горизонтальной линии. В этом случае:

Результат будет непредсказуемым. Через горизонтальную линию и две нажатые кнопки замкнутся вертикальные линии с высоким и низким уровнями сигнала. Какая вертикальная линия перетянет, такой уровень и будет на горизонтальной линии.

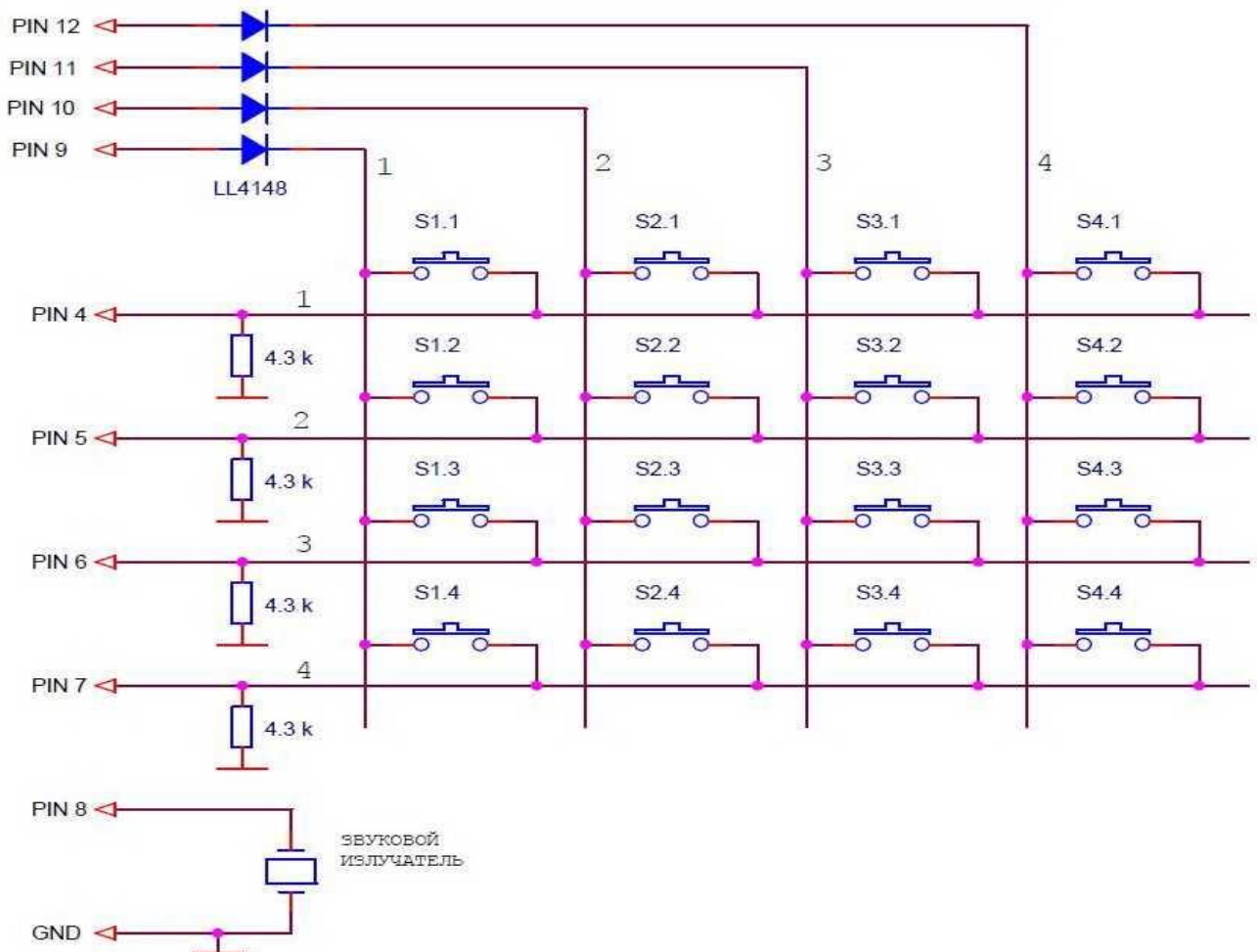
Произойдет замыкание выводов микроконтроллера с разными уровнями сигналов. В лучшем случае повысится потребляемый ток, и микроконтроллер будет нагреваться.



Для устранения этой проблемы достаточно подключить вертикальные линии через диоды. Тогда замыкание выводов контроллера невозможно при любом количестве одновременно нажатых кнопок. Также схема с диодами позволит корректно определять состояние матрицы с двумя одновременно нажатыми кнопками. При нажатых трех соседних кнопках произойдет замыкание четвертой.



Подключение матрицы кнопок к плате Ардуино.  
 Подключим матрицу кнопок 3x4 к плате Ардуино по такой схеме.



Хороший стиль - нажатие каждой кнопки сопровождать коротким звуковым сигналом. Для это подключим к плате звуковой пьезоизлучатель.

Для устранения дребезга кнопок используется способ ожидания стабильного состояния контактов, описанный в предыдущей лабораторной. В параллельном процессе должен регулярно вызываться метод scanState().

В результате формируются признаки массивов flagPress[4][4] и flagClick[4][4]:  
при нажатой кнопке flagPress= true;  
при отжатой кнопке flagPress= false;  
при нажатии на кнопку flagClick= true.

Объект типа MatrixKeys при создании имеет параметры:

номера выводов подключения вертикальных линий матрицы 1, 2, 3 ,4;

номера выводов подключения горизонтальных линий матрицы 1, 2, 3 ,4;

число подтверждений состояния контактов.

Пример создания объекта:

---

```
// создаем объект матрица кнопок keys  
// подключаем вертикальные линии к выводам 9, 10, 11, 12  
// подключаем горизонтальные линии к выводам 4, 5, 6, 7  
// число подтверждений состояния контактов 6  
MatrixKeys keys(9, 10, 11, 12, 4, 5, 6, 7, 6);
```

Конкретные выводы матрицы можно отключить, задав для них в конструкторе номера равные 255. Это может потребоваться при подключении матрицы меньшей размерности.

Для генерации звука по нажатию кнопок можно использовать стандартную функцию tone().

Стандартная функция генерации звука tone().

Функция генерирует на заданном выводе сигнал прямоугольной формы.

void tone(pin, frequency)

void tone(pin, frequency, duration)

Аргументы:

pin – номер вывода;

frequency – частота сигнала в Гц;

duration – длительность сигнала в миллисекундах.

Если длительность сигнала не задана третьим аргументом, то сигнал вырабатывается до тех пор пока не будет вызвана функция noTone().

Необходимо помнить, что для генерации сигнала функция tone() использует Таймер 2 платы Ардуино. Поэтому, если этот таймер уже используется в программе, например для формирования прерывания, то функция tone() приведет к конфликту обращения к таймеру 2.

Библиотека сканирования матричной клавиатуры 4 x 4.

В отличие от известной библиотеки Keypad эта библиотека:

значительно компактнее и быстрее;

ориентирована на использование в параллельном процессе;

надежно обрабатывает дребезг контактов, позволяет задавать число подтверждений состояния кнопок.

Напишем простую программу для проверки и демонстрации библиотеки MatrixKeys.h. Программа передает на компьютер по последовательному порту состояние матрицы кнопок:

нажатая кнопка отображается как "\*";

отжатая – ".";

момент нажатия кнопки (клик) отображается символом "=" в течение 0,5 секунд.

**ПРИМЕР**

```
// матрица кнопок
#include <MatrixKeys.h>
#include <MsTimer2.h>

// создаем объект матрица кнопок keys
// подключаем вертикальные линии к выводам 9, 10, 11, 12
// подключаем горизонтальные линии к выводам 4, 5, 6, 7
// число подтверждений состояния контактов 6
MatrixKeys keys(9, 10, 11, 12, 4, 5, 6, 7, 6);

void setup() {
  Serial.begin(9600); // инициализируем порт, скорость 9600
  MsTimer2::set(2, timerInterrupt); // задаем период прерывания по таймеру 2 мс
  MsTimer2::start(); // разрешаем прерывание по таймеру
}

void loop() {

  // перебор строк
  for (int i = 0; i < 4; i++) {

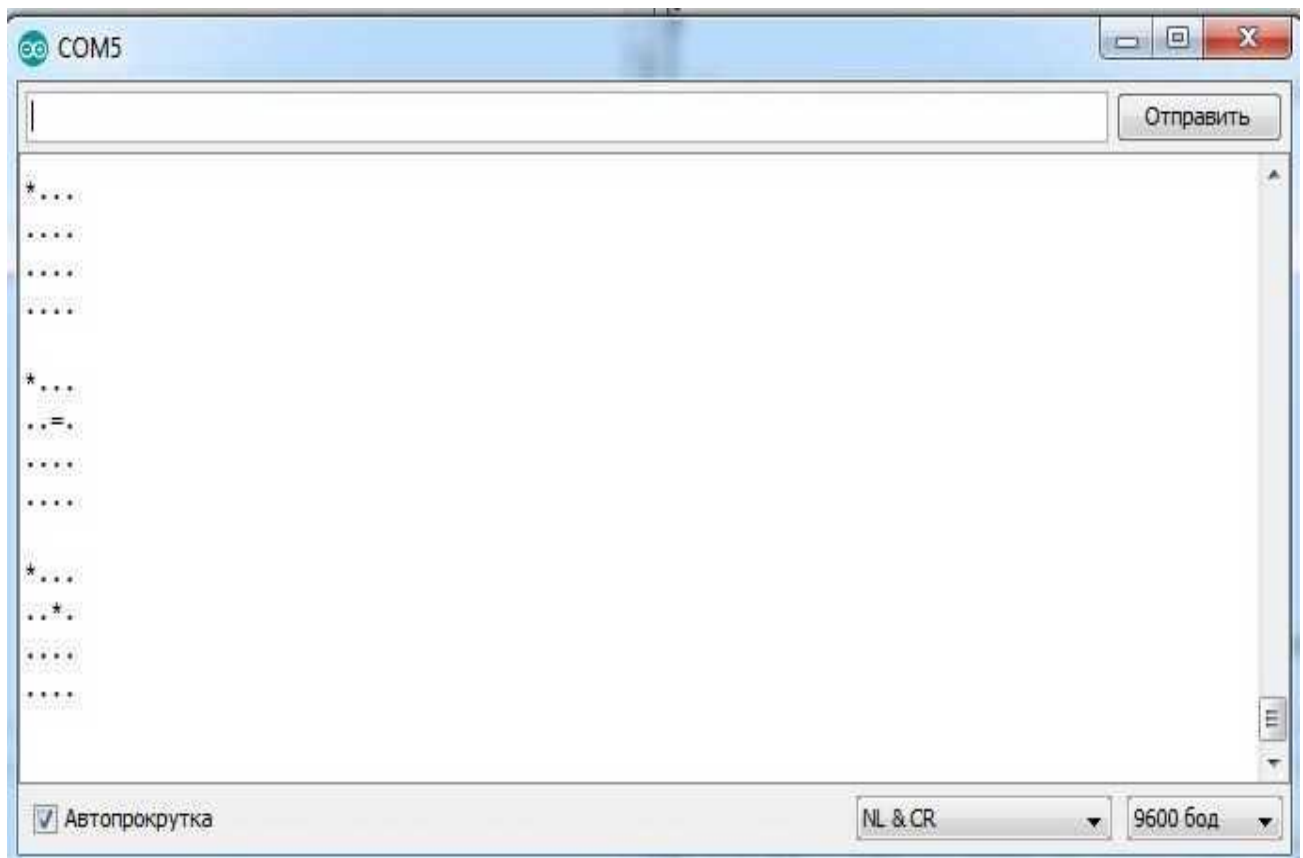
    // перебор столбцов
    for (int j = 0; j < 4; j++) {

      if (keys.flagClick[j][i] == true) { Serial.print("="); keys.flagClick[j][i]=false; tone
(8,1000,50); }
      else { if (keys.flagPress[j][i] == true) Serial.print("*"); else Serial.print("."); }
    }
    Serial.println(" ");
  }

  Serial.println(" ");
  delay(500);
}

//----- обработчик прерывания 2 мс
void timerInterrupt() {
  keys.scanState(); // сканирование матрицы
}
```

Загрузите программу в плату. Не забудьте установить библиотеки MatrixKeys.h и MsTimer2.h. Откройте монитор последовательного порта. На экране, каждые 0,5 секунды, будут пробегать блоки состояния матрицы.



Нажимая на кнопки матрицы можно проверить работу программы.

Если необходимо передавать код нажатой клавиши, то можно преобразовать состояние массива признаков кликов `keys.flagClick[4][4]` в коды кнопок `codKeys[4][4]`.

ПРИМЕР

```

// матрица кнопок
#include <MatrixKeys.h>
#include <MsTimer2.h>

// массив кодов кнопок
const char codKeys[4][4] =
{ {'1', '4', '7', '*'},
  {'2', '5', '8', '0'},
  {'3', '6', '9', '#'},
  {' ', ' ', ' ', ' ' }
};

// создаем объект матрица кнопок keys
// подключаем вертикальные линии к выводам 9, 10, 11, 12
// подключаем горизонтальные линии к выводам 4, 5, 6, 7
// число подтверждений состояния контактов 6
MatrixKeys keys(9, 10, 11, 12, 4, 5, 6, 7, 6);

void setup() {
  Serial.begin(9600); // инициализируем порт, скорость 9600
  MsTimer2::set(2, timerInterrupt); // задаем период прерывания по таймеру 2 мс
  MsTimer2::start(); // разрешаем прерывание по таймеру
}

void loop() {

  // вычисление кода нажатой кнопки
  // перебор столбцов
  for (int i = 0; i < 4; i++) {
    // перебор строк
    for (int j = 0; j < 4; j++) {
      if (keys.flagClick[i][j] == true) {
        keys.flagClick[i][j]=0;
        Serial.println(codKeys[i][j]);
      }
    }
  }
}

//----- обработчик прерывания 2 мс
void timerInterrupt() {
  keys.scanState(); // сканирование матрицы
}

```

Если загрузить программу в плату и открыть монитор последовательного порта, то при каждом нажатии кнопки в окне монитора будет отображаться символ, соответствующий кнопке.

При необходимости Вы легко сможете переделать библиотеку на матрицу кнопок других размерностей.

### **ЗАДАНИЕ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ**

1. Задайте произвольные символы и цифры, которые будут выводиться при нажатии кнопок матричной клавиатуры

2. Задайте произвольные фразы, которые будут выводиться при нажатии кнопок матричной клавиатуры (например: фамилию, имя, отчество, дату рождения, код группы и т.д.)