

Практическое занятие №2

Тема: «Подключение и программирование кнопочных массивов»

Цель работы: приобрести практические навыки по подключению и программированию кнопочных массивов на платформе Arduino.

Последовательность выполнения работы:

- Изучить теоретические сведения, приведенные в практическом занятии.
- Сделать монтажную и принципиальную схему в программе Fritzing. (напоминание: принципиальная схема формируется автоматически после создания монтажной).
- Собрать схемы желательно на макетной плате, иначе при отсутствии набора Arduino в web-приложениях (<https://wokwi.com/projects/new/arduino-uno> или <https://www.tinkercad.com/>) для приведенных примеров.
- Запрограммировать микроконтроллер согласно тексту, указанному в примере.
- Выполнить задание для самостоятельной работы.

Содержание отчета:

- Название практического занятия, его цель.
- Принципиальная и монтажная схема подключения к микроконтроллеру: скриншоты, в т.ч. если выполнение в web-приложении или фотографии, в случае наличия необходимых компонентов у вас (в наборе присутствуют).
- Написанный программный код для скетчей: вставить в отчет текстом, Courier New, 12 одинарный отступ без абзацев.
- Вывод о проделанной работе.

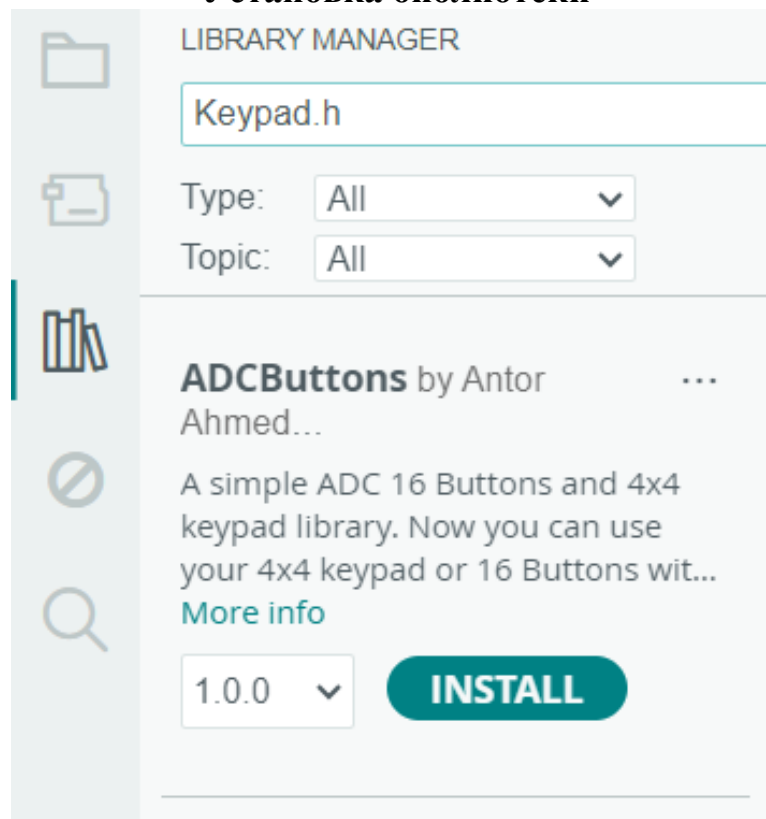
Теоретические сведения к заданию

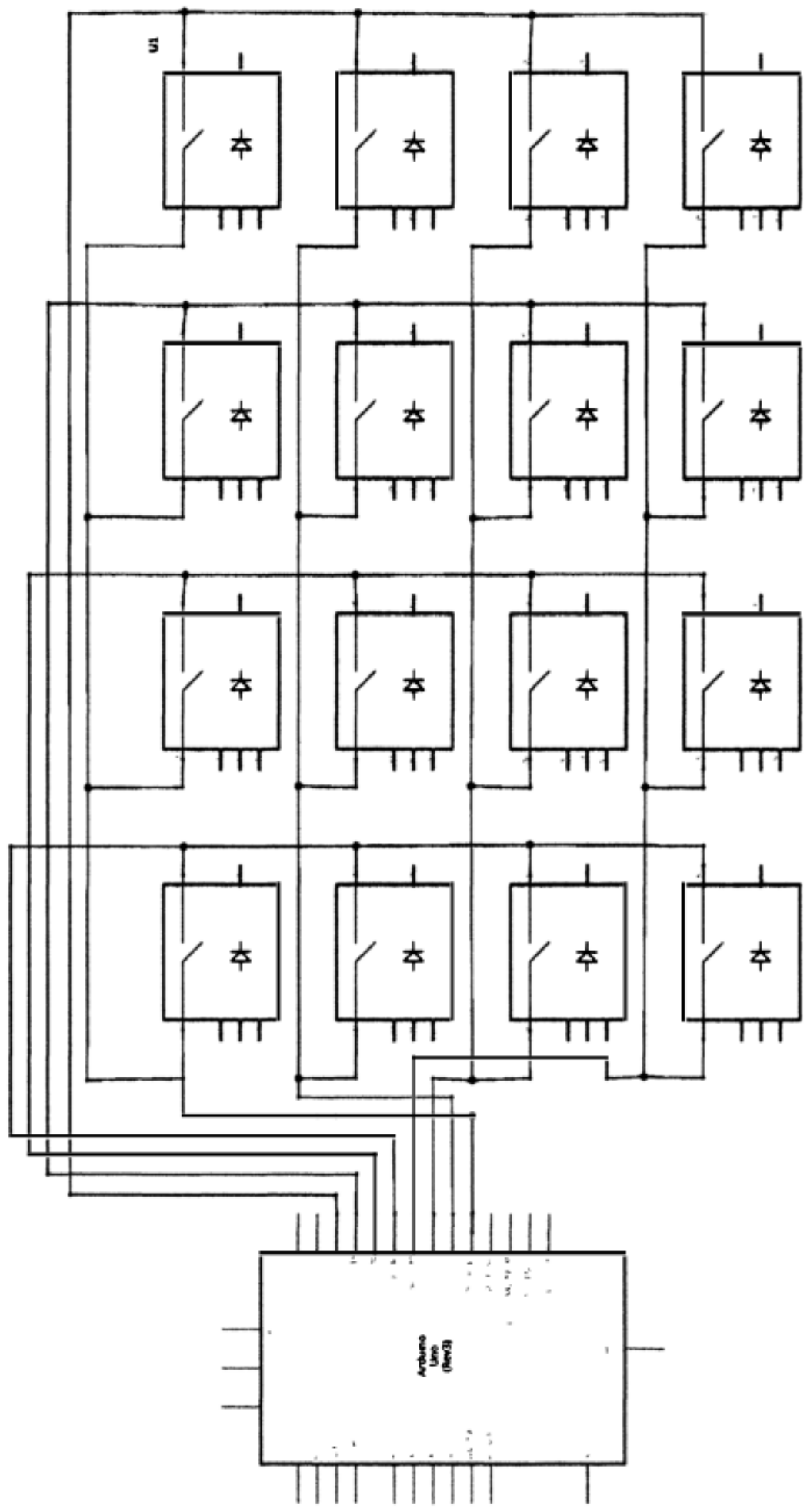
В этом практическом занятии нам **понадобятся**:

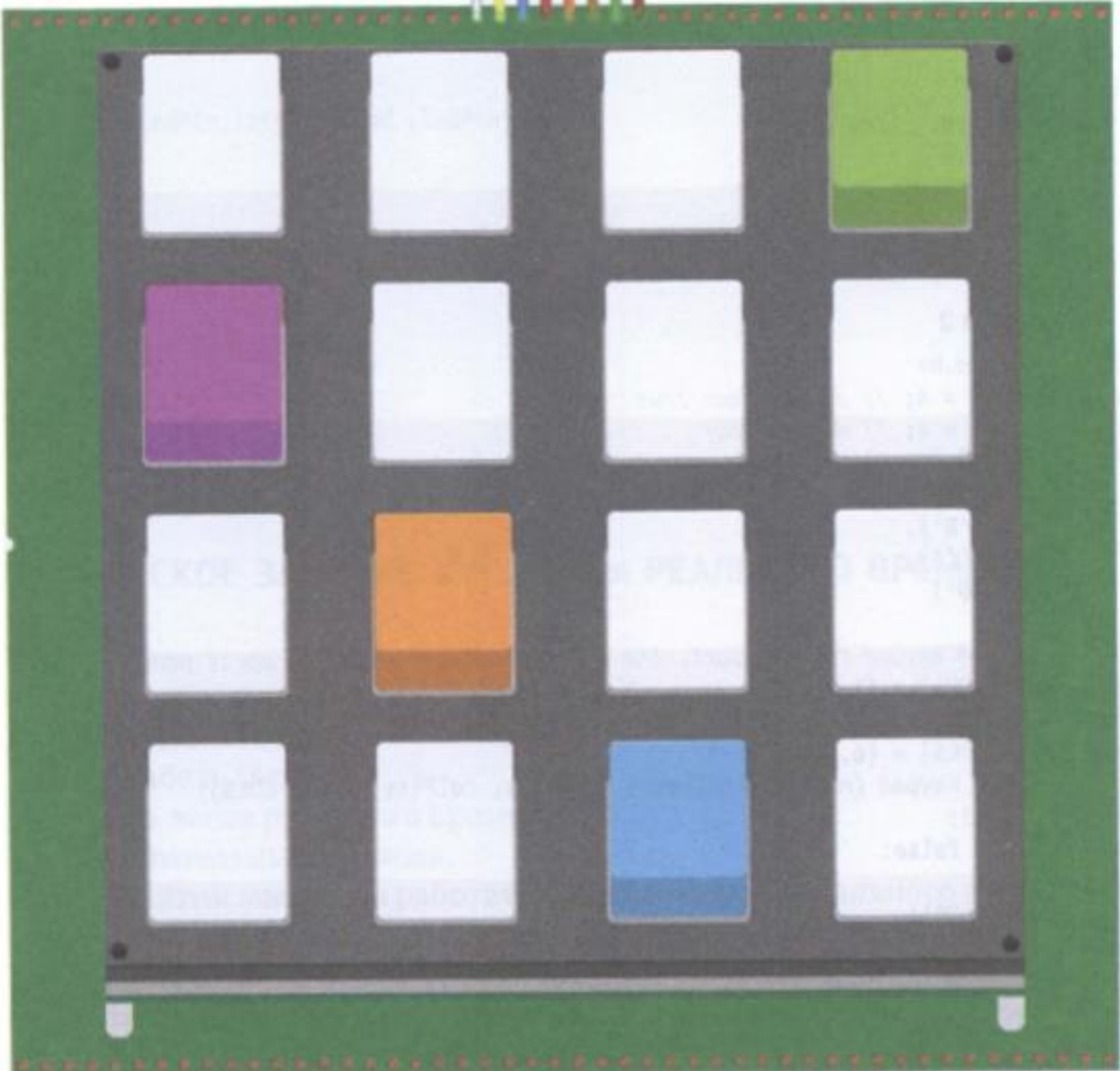
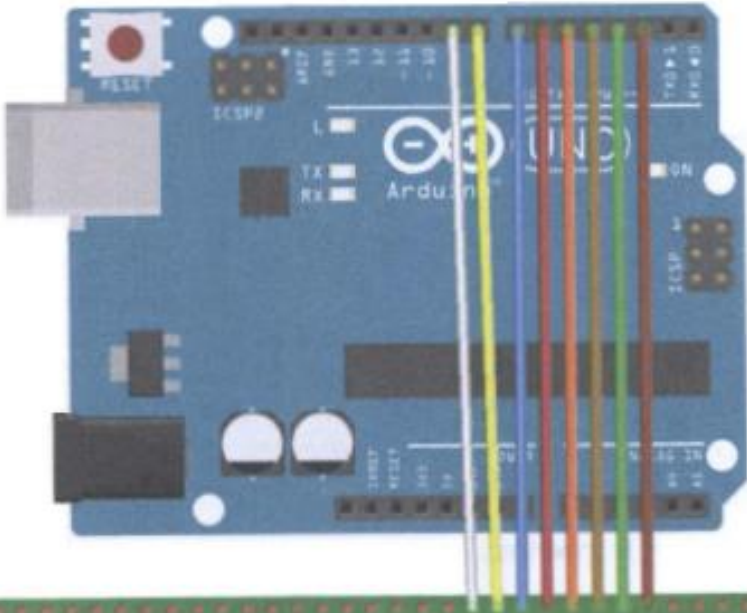
- плата Arduino Uno;
- USB-кабель (Am-Bm);
- кнопочный модуль 4×4;
- соединительные провода.

В этом занятии мы будем использовать кнопочный модуль 4×4 с 16 кнопками. В библиотеке компонентов Fritzing есть похожий компонент Buttonpad-4×4 в разделе Sparkfun-Electromechanical, однако внутри кнопки никак не соединены между собой и присутствуют светодиоды, которые так же висят в воздухе, как и кнопки, и вообще нам не нужны; поэтому для принципиальной схемы пришлось соединять кнопки между собой и выводить необходимые в занятии 8 выходов компонента. Также нам понадобится консоль, или монитор порта (**Ctrl+Shift+M**), чтобы смотреть результаты нажатия кнопок. Кроме того, нам понадобится библиотека Keypad.h, которую можно скачать с официального сайта Arduino по адресу [5]. Чтобы добавить библиотеку в проект Arduino IDE, нужно выбрать пункт меню **Скетч – Подключить библиотеку – Добавить .ZIP библиотеку...**, выбрать скачанный архив Keypad.zip и затем снова нажать **Скетч – Подключить библиотеку**, после чего выбрать библиотеку Keypad – она будет внизу списка. При этом в начало кода добавляется строка `#include <Keypad.h>`. Первая программа выводит символы в соответствии с нажатой кнопкой, вторая – задействует встроенный мини-светодиод на плате, который горит, пока нажата кнопка, ответственная за знак '*', и загорается или гаснет по нажатию кнопки, ответственной за символ '#'.

Установка библиотеки







Код программы №1

```
#include <Keypad.h>
const byte ROWS = 4; // define four rows
const byte COLS = 4; // define four
char keys [ROWS] [COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
// Connect 4 * 4 keypad row-bit port, the corresponding digital IO ports panel
byte rowPins [ROWS] = {2,3,4,5};
// Connect 4 * 4 buttons faithfully port, the corresponding digital IO ports panel
byte colPins [COLS] = {6,7,8,9};
// Call the function library function Keypad
Keypad keypad = Keypad (makeKeymap (keys), rowPins, colPins, ROWS, COLS);
void setup () {
  Serial.begin (9600);
}
void loop () {
  char key = keypad.getKey ();
  if (key!= NO_KEY) {
    Serial.println (key);
  }
}
```

Код программы №2

```
#include <Keypad.h>
const byte ROWS = 4; // define four rows
const byte COLS = 4; // define four
char keys [ROWS] [COLS] = {
  {'1', '2', '3','A'},
  {'4', '5', '6','B'},
  {'7', '8', '9','C'},
  {'*', '0','#','D'}
};
// Connect 4 * 4 keypad row-bit port, the corresponding digital IO ports panel
byte rowPins [ROWS] = {2,3,4,5};
// Connect 4 * 4 buttons faithfully port, the corresponding digital IO ports panel
byte colPins [COLS] = {6,7,8,9};
Keypad keypad = Keypad (makeKeymap (keys), rowPins, colPins, ROWS, COLS);
byte ledPin = 13;
boolean blink = false;
void setup () {
  Serial.begin (9600);
  pinMode (ledPin, OUTPUT); // sets the digital pin as output
  digitalWrite (ledPin, HIGH); // sets the LED on
  keypad.addEventListener (keypadEvent); // add an event listener for this keypad
}
void loop () {
  char key = keypad.getKey ();
  if (key!= NO_KEY) {
    Serial.println (key);
  }
}
```

```
}  
if (blink) {  
  digitalWrite (ledPin,! digitalRead (ledPin));  
  delay (100);  
}  
}  
// Take care of some special events  
void keypadEvent (KeypadEvent key) {  
  switch (keypad.getState ()) {  
  case PRESSED:  
    switch (key) {  
    case '#': digitalWrite (ledPin,!digitalRead (ledPin)); break;  
    case '*':  
      digitalWrite (ledPin,!digitalRead (ledPin));  
      break;  
    }  
    break;  
  case RELEASED:  
    switch (key) {  
    case '*':  
      digitalWrite (ledPin,!digitalRead (ledPin));  
      blink = false;  
      break;  
    }  
    break;  
  case HOLD:  
    switch (key) {  
    case '*': blink = true; break;  
    }  
    break;  
  }  
}
```