

Разработка простого меню в LCD панели на базе микроконтроллера Arduino

Вавилов Егор Дмитриевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье описан процесс создания макета простейшего меню на LCD панели. Для создания используется плата Arduino, LCD панель, макетная плата, резисторы и светодиод. Разработанное меню может в дальнейшем быть доработано для использования в сложных проектах

Ключевые слова: Arduino, микроконтроллер, LCD панель

Development of a simple menu in LCD panel based on the Arduino microcontroller

Vavilov Yegor Dmitrievich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article describes the process of creating a simple menu layout on the LCD panel. The Arduino board, LCD panel, layout board, resistors and LED are used for creation. The developed menu can be further modified for use in complex projects.

Keywords: Arduino, microcontroller, LCD panel

Для некоторых сложных проектов на Arduino иногда может потребоваться возможность выбора нескольких опций, лучше всего, когда это можно сделать визуально, т.е. использовать специальное меню, например на LCD панели.

Цель исследования – создать простейшее меню на LCD панели.

Ранее этим вопросом интересовались Я.Н. Стецюк и М.В. Слива описывали тему «Работа с графическими экранами и микроконтроллерами (на основе платформы Arduino)» [1] в которой они рассказали о своей работе с LCD экраном. Рассказали об использовании LCD панелей и меню в сложных системах Д.Г. Волохов и А.В. Прядилов в своей работе «Многофункциональная система управления на базе Arduino» [2], а также А.А. Миронова показала «Использование автономных математических тренажеров на платформе Arduino в обучении школьников» [3], где использовала LCD панель в одном из примеров.

Для этого потребуется:

- Плата Arduino
- Соединительные провода
- Несколько кнопок (или мембрана с кнопками)
- 1 LCD-Экран
- 2 резистора 220 Ом
- 1 резистор 1 Ом
- 1 светодиод
- Макетная плата

Схема подключения представлена на рисунке 1.

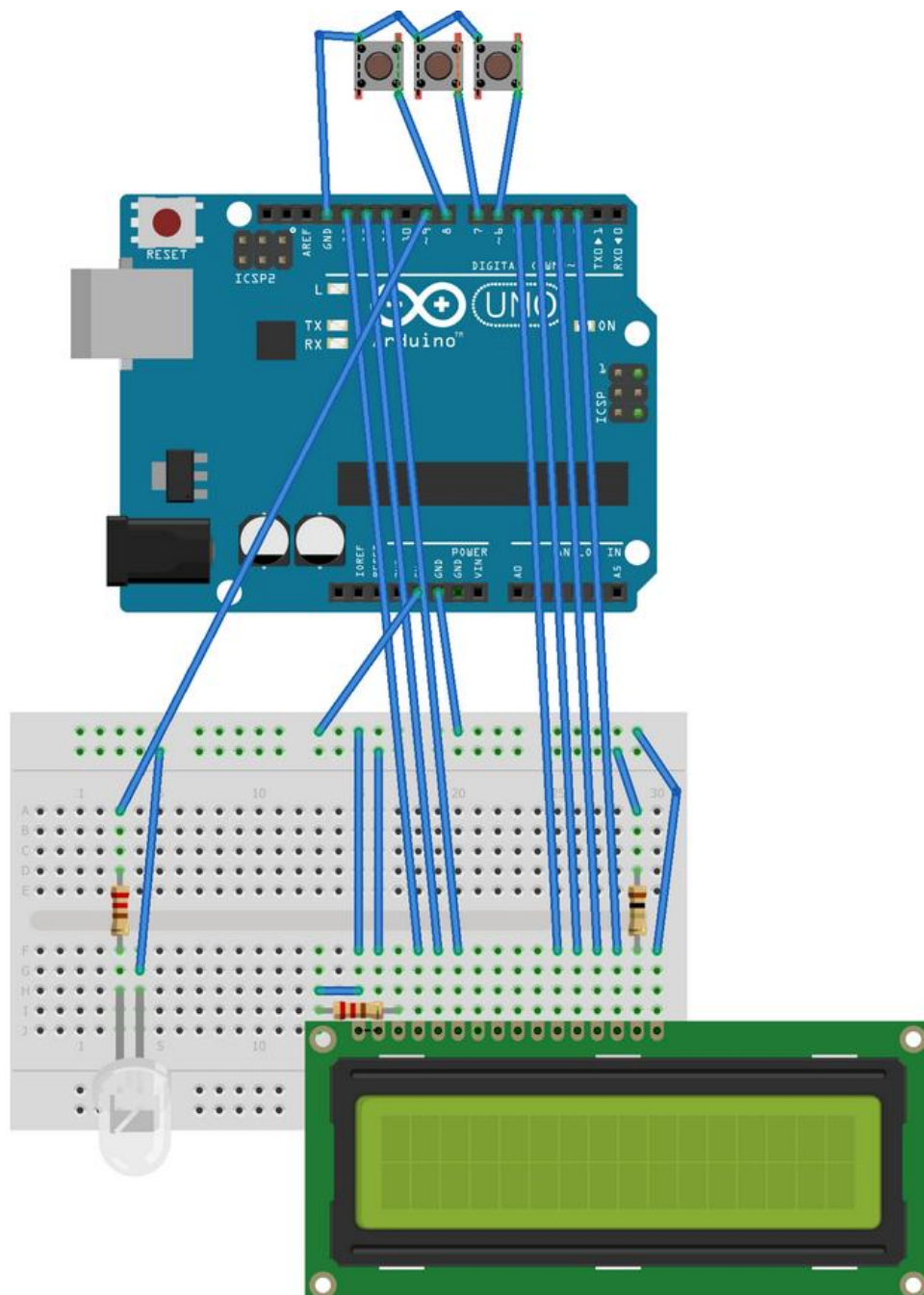


Рисунок 1. Схема подключения к плате Arduino

Идея проекта выводить на экран возможные, запрограммированные опции кода и выполнять только их при выборе.

В написанной программе будет создано меню для управления светодиодом, как пример возможного меню. При нажатии кнопок будет осуществляется либо перемещение по меню (вверх/вниз), либо выполнять опцию из меню. В написанном коде осуществляются следующие возможные опции меню для светодиода:

- Включение
- Отключение
- Постепенное угасание и зажигание
- Мигание

Листинг программы можно увидеть на рисунке 2 и 3.

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 10, 5, 4, 3, 2);
byte buttons[] = {6,7,8};
const byte nrButtons = 3;
int menuSize = 10;
String menu[] = {
  "Menu",
  "Menu>LED",
  "Menu>LED>Off",
  "Menu>LED>On",
  "Menu>LED>Fade",
  "Menu>LED>Blink",
  "Menu>LCDlight",
  "Menu>Nothing1",
  "Menu>Nothing2",
  "Menu>Nothing3"
};
int t;
byte pressedButton, currentPos, currentPosParent, possiblePos[20], possiblePosCount, possiblePosScroll = 0;
String parent = "";
int brightness;
int LCDtoggle = 0;
int fadeAmount = 5;
unsigned long timer = (millis() / 10);
int ledMode = 0;
void updateMenu () {
  possiblePosCount = 0;
  while (possiblePosCount == 0) {
    for (t = 1; t < menuSize; t++) {
      if (mid(menu[t], 1, inStrRev(menu[t], ">")-1).equals(menu[currentPos])) {
        possiblePos[possiblePosCount] = t;
        possiblePosCount = possiblePosCount + 1;
      }
    }
    parent = mid(menu[currentPos], 1, inStrRev(menu[currentPos], ">")-1);
    currentPosParent = 0;
    for (t = 0; t < menuSize; t++) {
      if (parent == menu[t]) {currentPosParent = t;}
    }
    if (possiblePosCount == 0) {
      switch (currentPos) {
        case 2:
        case 3:
        case 4:
        case 5://Choose between 2,3,4,5
          for (t = 2; t < 6; t++) {
            if (mid(menu[t], len(menu[t]), 1) == "**") {
              menu[t] = mid(menu[t], 1, len(menu[t])-1);
            }
          }
          menu[currentPos] = menu[currentPos] + "**";
          break;
        case 6:
          if (mid(menu[currentPos], len(menu[currentPos]), 1) == "**") {
            menu[currentPos] = mid(menu[currentPos], 1, len(menu[currentPos])-1);
          } else {
            menu[currentPos] = menu[currentPos] + "**";
          }
          break;
      }
    }
    switch (currentPos) {
      case 2:
        ledMode = 0;
        break;
      case 3:
        ledMode = 1;
        break;
    }
  }
}
```

Рисунок 2. Листинг программы меню

```

        case 4:
            brightness = 128;
            ledMode = 2;
            break;
        case 5:
            brightness = 0;
            ledMode = 3;
            break;
        case 6:
            LCDtoggle = abs(LCDtoggle - 1);
            digitalWrite(13,LCDtoggle);
            break;
    }
    currentPos = currentPosParent;
}
}
lcd.clear();
lcd.setCursor(0,0); lcd.print (mid(menu[currentPos],inStrRev(menu[currentPos],">")+1,len(menu[currentPos])-inStrRev(menu[currentPos],">")));
lcd.setCursor(0,1); lcd.print (mid(menu[possiblePos[possiblePosScroll]],inStrRev(menu[possiblePos[possiblePosScroll]],">")+1,len(menu[possiblePos[possiblePosScroll]])-inStrRev(menu[possiblePos[possiblePosScroll]],">")));
}
byte checkButtonPress() {
    byte bP = 0;
    byte rBp = 0;
    for (t = 0; t<nrButtons;t++) {
        if (digitalRead(buttons[t]) == 0) {bP = (t + 1);}
    }
    rBp = bP;
    while (bP != 0) { // wait while the button is still down
        bP = 0;
        for (t = 0; t<nrButtons;t++) {
            if (digitalRead(buttons[t]) == 0) {bP = (t + 1);}
        }
    }
    return rBp;
}
void setup() {
    lcd.begin(16,2);
    lcd.clear();
    for (t=0;t<nrButtons;t++) {
        pinMode(buttons[t],INPUT_PULLUP);
    }
    pinMode(13,OUTPUT);
    pinMode(9,OUTPUT);
    lcd.setCursor(0,0); lcd.print("HELLO");
    delay(1000);
    updateMenu();
}
void loop() {
    switch (ledMode) {
        case 0:
            analogWrite(9, 0);
            break;
        case 1:
            analogWrite(9, 255);
            break;
        case 3:
            if ((millis() / 10) - timer > 30) {
                timer = (millis() / 10);
                brightness = abs(brightness - 255);
                timer = (millis() / 10);
                analogWrite(9, brightness);
            }
            break;
        case 2:
            if ((millis() / 10) - timer > 3) {
                timer = (millis() / 10);
                analogWrite(9, brightness);
                brightness = brightness + fadeAmount;
                if (brightness <= 0 || brightness >= 255) {
                    fadeAmount = -fadeAmount;
                }
            }
            break;
    }
    pressedButton = checkButtonPress();
    if (pressedButton !=0) {
        switch (pressedButton) {
            case 1:
                possiblePosScroll = (possiblePosScroll + 1) % possiblePosCount;
                break;
            case 2:
                currentPos = possiblePos[possiblePosScroll];
                break;
            case 3:
                currentPos = currentPosParent;
                possiblePosScroll = 0;
                break;
        }
        updateMenu();
    }
}
}

```

Рисунок 3. Продолжение листинга программы

Результат работы можно увидеть на рисунке 4.

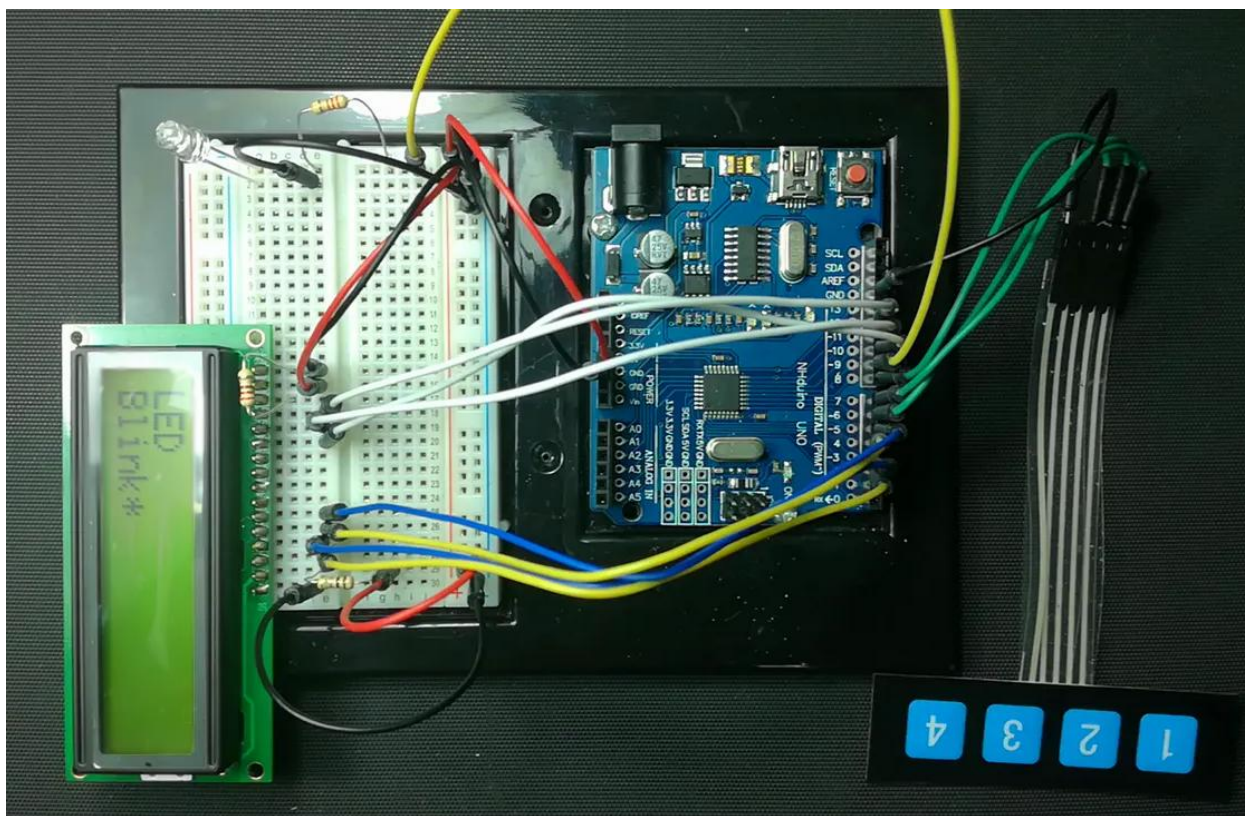


Рисунок 4. Макет меню LCD-экрана на базе платы Arduino

Вывод.

Результатом статьи стал прототип работающего меню для управления режимами работы светодиода. Такую простейшую схему легко повторить, так как она состоит из базовых модулей. При желании данное меню можно легко доработать и использовать в более сложных проектах.

Библиографический список

1. Стецюк Я.Н., Слива М.В. Работа с графическими экранами и микроконтроллерами (на основе платформы Arduino) // В сборнике: Культура, наука, образование: проблемы и перспективы материалы VI международной научно-практической конференции. 2017. С. 220-225.
2. Волохов Д.Г., Прядилов А.В. Многофункциональная система управления на базе Arduino // В сборнике: Прикладная математика и информатика: современные исследования в области естественных и технических наук Материалы IV научно-практической международной конференции (школы-семинара) молодых ученых: в 2 частях. 2018. С. 360-364.
3. Миронова А.А. Использование автономных математических тренажеров на платформе Arduino в обучении школьников // В книге: Молодежь XXI века: образование, наука, инновации Материалы VII Всероссийской студенческой научно-практической конференции с международным участием. Под редакцией Т.А. Василенко. 2018. С. 131-133.