

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» (ДГТУ)**

Факультет: Информатика и вычислительная техника

Кафедра: Кибербезопасность информационных систем

**Отчёт по проекту**

**по дисциплине “Аппаратные средства вычислительной техники”**

Выполнил обучающийся гр.ВКБ31

Котелевец Кирилл Александрович

Проверил:

Доцент Болдырихин Н.В.

Ростов-на-Дону

2024

Тема: Аппаратный менеджер паролей.

Ход работы

Был написан следующий кол:

Листинг программы

|  |
| --- |
| #include <Keypad.h>  #include <LiquidCrystal.h>  #include <EEPROM.h>  LiquidCrystal lcd(A5, A4, A3, A2, A1, A0);  const byte ROWS = 4;  const byte COLS = 4;  char keys[ROWS][COLS] = {  {'1', '2', '3', 'A'},  {'4', '5', '6', 'B'},  {'7', '8', '9', 'C'},  {'\*', '0', '#', 'D'}  };  byte rowpins[ROWS] = {9, 8, 7, 6};  byte colpins[COLS] = {5, 4, 3, 2};  Keypad keypd = Keypad(makeKeymap(keys), rowpins, colpins, ROWS, COLS);  const int passwordLength = 8;  char masterPassword[passwordLength + 1] = "11111111";  const int eepromAddress = 0;  int failedAttempts = 0;  const int MAX\_FAILED\_ATTEMPTS = 3;  unsigned long lockoutTime = 0;  const unsigned long LOCKOUT\_DURATION = 60000; // 1 минута  String websites[] = {"Google", "Facebook", "Twitter", "LinkedIn", "GitHub"};  String usernames[] = {"user1", "user2", "user3", "user4", "user5"};  String passwords[] = {"pass1", "pass2", "pass3", "pass4", "pass5"};  const int numWebsites = 5;  int currentWebsite = 0;  bool isUnlocked = false;  bool isChangingPassword = false;  void setup() {  lcd.begin(16, 2);  Serial.begin(9600);  pinMode(10, OUTPUT); // Green  pinMode(11, OUTPUT); // Red  pinMode(12, OUTPUT); // Blue        lcd.print("Enter Password:");  lcd.setCursor(0, 1);  }  void loop() {  if (!isUnlocked) {  if (isChangingPassword) {  handleChangePassword();  } else {  handlePasswordEntry();  }  } else {  handleWebsiteSelection();  }  }  void handlePasswordEntry() {  static char enteredPassword[passwordLength + 1];  static int passwordIndex = 0;    char key = keypd.getKey();    if (key) {  if (key >= '0' && key <= '9' && passwordIndex < passwordLength) {  enteredPassword[passwordIndex++] = key;    lcd.print('\*');  } else if (key == 'B' && passwordIndex > 0) {    passwordIndex--;  lcd.setCursor(passwordIndex, 1);  lcd.print(' ');  lcd.setCursor(passwordIndex, 1);    } else if (key == '#' && passwordIndex == passwordLength) {    enteredPassword[passwordIndex] = '\0';  Serial.println(enteredPassword);  Serial.println(masterPassword);  if (strcmp(enteredPassword, masterPassword) == 0) {  failedAttempts = 0;  isUnlocked = true;  lcd.clear();  lcd.print("Access Granted");  digitalWrite(10, HIGH); // Green LED on  delay(1000);  digitalWrite(10, LOW);  lcd.clear();  displayWebsite();  } else {  failedAttempts++;    lcd.clear();  lcd.print("Wrong Password");  digitalWrite(11, HIGH); // Red LED on  delay(3000);  if (failedAttempts >= MAX\_FAILED\_ATTEMPTS) {  lcd.clear();  lcd.print("Locked out");  lockoutTime = millis();  while (millis() - lockoutTime < LOCKOUT\_DURATION) {  // Ждем, пока не пройдет время блокировки  }  failedAttempts = 0;  }  digitalWrite(11, LOW);  lcd.clear();  lcd.print("Enter Password:");  lcd.setCursor(0, 1);  }  passwordIndex = 0;  }  }  }  void handleChangePassword() {  static char newPassword[passwordLength + 1];  static char confirmPassword[passwordLength + 1];  static int newPasswordIndex = 0;  static bool confirming = false;    digitalWrite(12, HIGH); // Включаем синий светодиод в режиме смены пароля    char key = keypd.getKey();    if (key) {  if (key >= '0' && key <= '9' && newPasswordIndex < passwordLength) {  newPassword[newPasswordIndex++] = key;  lcd.print('\*');  } else if (key == 'B' && newPasswordIndex > 0) {  newPasswordIndex--;  lcd.setCursor(newPasswordIndex, 1);  lcd.print(' ');  lcd.setCursor(newPasswordIndex, 1);  } else if (key == '#' && newPasswordIndex == passwordLength) {  newPassword[newPasswordIndex] = '\0';  if (!confirming) {  lcd.clear();  lcd.print("Confirm password:");  lcd.setCursor(0, 1);  confirming = true;  newPasswordIndex = 0;  } else {  if (strcmp(newPassword, confirmPassword) == 0) {  // Пароли совпадают, сохраняем новый пароль  for (int i = 0; i < passwordLength; i++) {  EEPROM.write(eepromAddress + i, newPassword[i]);  masterPassword[i] = newPassword[i];  }  masterPassword[passwordLength] = '\0';    lcd.clear();  lcd.print("Password Changed");  delay(2000);  } else {  lcd.clear();  lcd.print("Passwords");  lcd.setCursor(0, 1);  lcd.print("do not match");  delay(2000);  }  lcd.clear();  lcd.print("Enter Password:");  lcd.setCursor(0, 1);  isChangingPassword = false;  confirming = false;  newPasswordIndex = 0;  }  }  digitalWrite(12, LOW); // Выключаем синий светодиод  }  void handleWebsiteSelection() {  char key = keypd.getKey();    if (key) {  switch (key) {  case '\*':  currentWebsite = (currentWebsite - 1 + numWebsites) % numWebsites;  displayWebsite();  break;  case '#':  currentWebsite = (currentWebsite + 1) % numWebsites;  displayWebsite();  break;  case 'A':  displayCredentials();  break;  case 'B':  isUnlocked = false;  lcd.clear();  lcd.print("Enter Password:");  lcd.setCursor(0, 1);  break;  case 'C':  isChangingPassword = true;  lcd.clear();  lcd.print("Mode chande true");  lcd.setCursor(0, 1);  lcd.print("Please Click B");  break;    }  }  }  void displayWebsite() {  lcd.clear();  lcd.print("S:");  lcd.print(websites[currentWebsite]);  lcd.setCursor(0, 1);  lcd.print("A:Show B:Back");  }  void displayCredentials() {  lcd.clear();  lcd.print(usernames[currentWebsite]);  lcd.setCursor(0, 1);  lcd.print(passwords[currentWebsite]);  delay(3000);  displayWebsite();  } |

На рисунке 1 представлена схема моей платы.

Реализует систему ввода паролей с использованием клавиатуры и ЖК-дисплея. Код позволяет пользователям вводить пароль, разблокирует доступ к меню выбора веб-сайтов после успешного ввода и управляет изменениями пароля, а также обрабатывает некорректные попытки входа с механизмом блокировки после нескольких неудачных попыток.

1. Инициализация и Настройка:

- ЖК-дисплей и клавиатура инициализируются.

- Определен мастер-пароль, а также отслеживаются неудачные попытки.

2. Главный Цикл:

- Программа проверяет, разблокирована ли система. Если нет, предоставляется возможность ввода пароля или изменения пароля.

3. Обработка Ввода Пароля:

- Пользователи могут вводить пароль с помощью клавиатуры. Каждый введенный знак отображается в виде звездочки на ЖК-дисплее.

- Кнопка # отправляет пароль, а кнопка B позволяет удалить последний введенный символ (функция "назад").

- Если пароль совпадает с заданным мастер-паролем, доступ предоставляется, и происходит переход к следующей функциональности (например, отображение вариантов веб-сайтов).

- Неверные попытки увеличивают счетчик failedAttempts, и после достижения максимального числа неудач система блокируется на определенный срок.

4. Дополнительные Функции:

- Код также включает массивы для веб-сайтов, имен пользователей и паролей, которые предположительно могут быть отображены пользователю после успешного входа

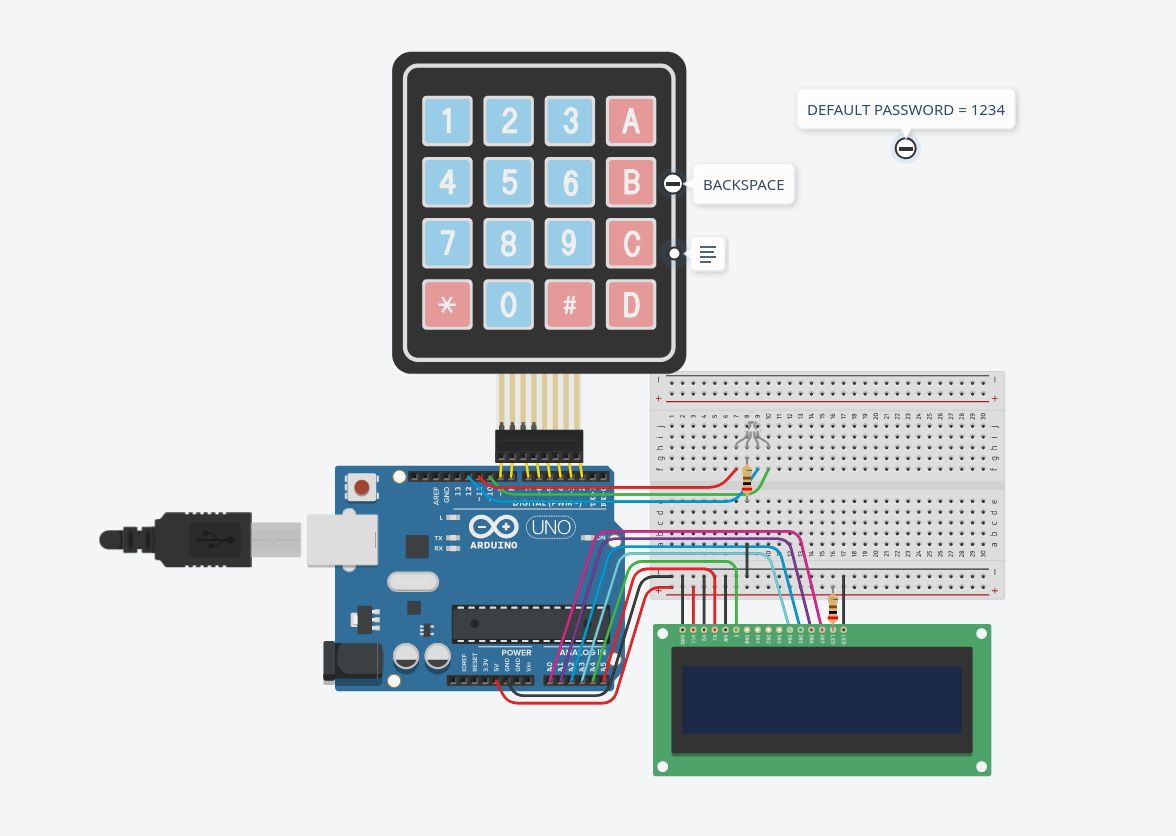


Рисунок 1