

## Идентификация с помощью отпечатка пальца на базе Arduino

*Кизянов Антон Олегович*

*Приамурский государственный университет имени Шолом-Алейхема*

*Студент*

### Аннотация

В данной статье описан процесс создания системы идентификации по отпечатку пальца. Для создания используется плата Arduino и датчик отпечатка пальца. Созданный макет в полной мере отражает устройство систем идентификации и позволяет создать более сложные варианты.

**Ключевые слова:** Fingerprint sensor, Arduino, Internet of things

### Arduino-based fingerprint identification

*Kizyanov Anton Olegovich*

*Sholom-Aleichem Priamursky State University*

*student*

### Abstract

This article describes the process of creating a fingerprint identification system. To create a used Arduino board and fingerprint sensor. The created layout fully reflects the device identification systems and allows you to create more complex options.

**Keywords:** Fingerprint sensor, Arduino, Internet of things

Считыватели отпечатков пальцев более безопасны, надежны и удобны, чем обычные традиционные пароли, хотя и их можно подобрать. Система распознавания отпечатков пальцев более тесно связана с конкретным пользователем, чем, например, карта доступа, которую можно украсть. Программное обеспечение считывателя отпечатков пальцев позволяет регистрировать до десяти отпечатков пальцев на устройство.

При предъявлении авторизованного отпечатка пальца программное обеспечение передает соответствующие имена и пароли для входа в систему совместимым приложениям и веб-сайтам, что позволяет выполнять вход без клавиатуры. Если программное обеспечение обнаруживает, что конкретный отпечаток не совпадает с его базой данных, оно отклоняет доступ.

Цель исследования – создать систему идентификации по отпечатку пальца на базе Arduino.

Ранее этим вопросом интересовались Р.Р. Бастанов, А.Н. Калитаев, В.Д. Тутарова, развивали тему «Алгоритм создания шаблона отпечатка пальца» [1], в которой обсуждается алгоритм создания шаблона отпечатка пальца для современных биометрических систем, состоящий из этапов

считывания, восстановления и бинаризации, скелетизации и постобработки изображения и выявления минуций. Применение данного алгоритма позволит идентифицировать человека по его уникальным биологическим параметрам. В.И. Волчихин, В.А. Фунтиков, А.И. Иванов, М.В. Секретов, А.В. Майоров, А.Ю. Малыгин с темой «Способ классификации рисунков отпечатка пальца при аутентификации личности» [2], а подробнее про область биометрической аутентификации личности, построенной на сохранении в тайне применяемого при аутентификации биометрического образа в виде рисунка отпечатка пальца. Техническим результатом является сохранение в тайне рисунка отпечатка пальца и его биометрических параметров, используемых при биометрической аутентификации, и обеспечение условий, при которых по данным биометрической аутентификации нельзя найти хозяина биометрии, при одновременном сокращении среднего времени загрузки сервера (средства) аутентификации. Р.Н. Киндеркнехт опубликовал статью «Обработка изображения в задаче идентификации по отпечатку пальца» [3] рассказал про структуру отпечатка пальца и методы обработки изображения для построения шаблона. Идентификация по изображению осуществляется с использованием геометрических преобразований, статистических методов и методов цифровой обработки сигналов.

В этом проекте доступ к отпечатку пальца будет осуществляться путем считывания отпечатка пальца с использованием датчика отпечатка пальца [4], который можно увидеть на рисунке 1.



Рис. 1 Датчик отпечатка пальца

Система будет работать следующим образом, есть реле, которое выключено пока датчик отпечатков не подтвердил личность, а когда подтверждение прошло успешно, реле включается и позволяет запустить совершенно другую систему, совсем не связанную с первой.

В собранном виде это выглядит как на рисунке 2.

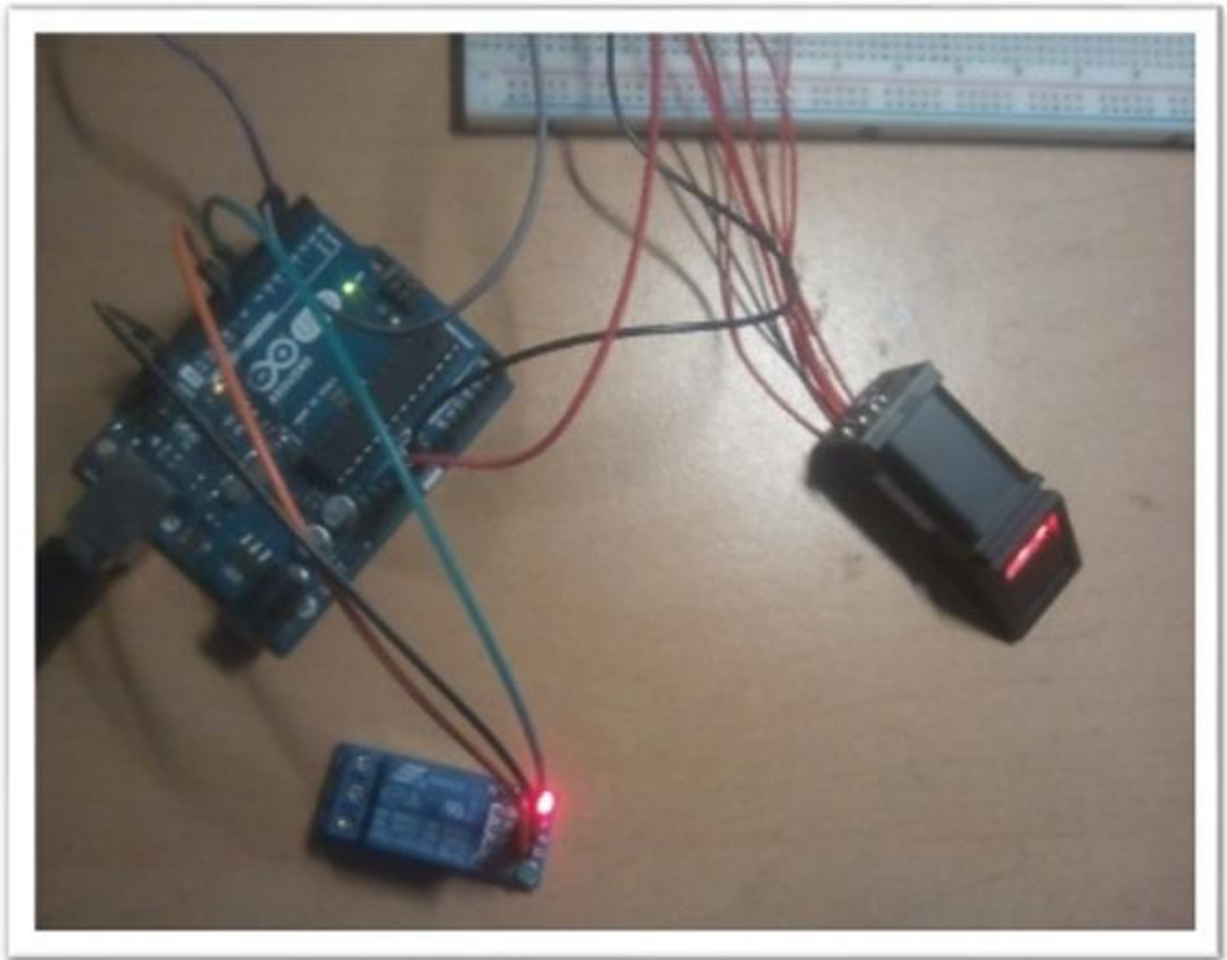


Рис. 2 Собранная схема Arduino и датчика отпечатков

Сначала нужно определить библиотеки.

```
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
```

Определить идентификатор чтения и функцию процесса регистрации.

```
uint8_t id;
uint8_t getFingerprintEnroll();
```

Определить последовательную связь с устройством.

```
SoftwareSerial mySerial(2, 3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
```

Объявить экземпляр датчика.

```
//Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial1);
```

Проверить состояние датчика.

```
void setup()
{
  while (!Serial);
  delay(500);
```

Отобразить подтверждение датчика.

```
Serial.begin(9600);
Serial.println("Adafruit Fingerprint sensor enrollment");
// set the data rate for the sensor serial port
finger.begin(57600);
```

Определить датчик, если он обнаруживает.

```
if (finger.verifyPassword()) {
  Serial.println("Found fingerprint sensor!");
} else {
  Serial.println("Did not find fingerprint sensor :(");
  while (1);
}
}
uint8_t readnumber(void) {
  uint8_t num = 0;
  boolean validnum = false;
  while (1) {
    while (! Serial.available());
    char c = Serial.read();
    if (isdigit(c)) {
      num *= 10;
      num += c - '0';
      validnum = true;
    } else if (validnum) {
      return num;
    }
  }
}
```

Показать регистрационный номер.

```
void loop()
{
  Serial.println("Ready to enroll a fingerprint! Please Type in the ID # you want to
  save this finger as...");
  id = readnumber();
  Serial.print("Enrolling ID #");
  Serial.println(id);
  while (! getFingerprintEnroll() );
}
```

Функция для регистрации выглядит следующим образом.

```
uint8_t getFingerprintEnroll() {
  int p = -1;
  Serial.print("Waiting for valid finger to enroll as #"); Serial.println(id);
  while (p != FINGERPRINT_OK) {
    p = finger.getImage();
```

```
switch (p) {
case FINGERPRINT_OK:
Serial.println("Image taken");
break;
case FINGERPRINT_NOFINGER:
Serial.println(".");
break;
case FINGERPRINT_PACKETRECEIVEERR:
Serial.println("Communication error");
break;
case FINGERPRINT_IMAGEFAIL:
Serial.println("Imaging error");
break;
default:
Serial.println("Unknown error");
break;
}
```

Если датчик успешно считывает изображение, выводится следующее.

```
p = finger.image2Tz(1);
switch (p) {
case FINGERPRINT_OK:
Serial.println("Image converted");
break;
case FINGERPRINT_IMAGEMESS:
Serial.println("Image too messy");
return p;
case FINGERPRINT_PACKETRECEIVEERR:
Serial.println("Communication error");
return p;
case FINGERPRINT_FEATUREFAIL:
Serial.println("Could not find fingerprint features");
return p;
case FINGERPRINT_INVALIDIMAGE:
```

Если он не может найти функции отпечатка пальца, выводится следующее: Serial.println («Не удалось найти функции отпечатка пальца»).

```
return p;
default:
Serial.println("Unknown error");
return p;
}
```

Обновить датчик отпечатков пальцев.

```
Serial.println("Remove finger");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
p = finger.getImage();
}
Serial.print("ID "); Serial.println(id);
p = -1;
Serial.println("Place same finger again");
```

```
while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
    case FINGERPRINT_OK:
        Serial.println("Image taken");
        break;
    case FINGERPRINT_NOFINGER:
        Serial.print(".");
        break;
    case FINGERPRINT_PACKETRECEIVEERR:
        Serial.println("Communication error");
        break;
    case FINGERPRINT_IMAGEFAIL:
        Serial.println("Imaging error");
        break;
    default:
        Serial.println("Unknown error");
        break;
    }
}
```

Изображение для датчика отпечатка пальца.

```
p = finger.image2Tz(2);
switch (p) {
case FINGERPRINT_OK:
    Serial.println("Image converted");
    break;
case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    return p;
case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Communication error");
    return p;
case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    return p;
default:
    Serial.println("Unknown error");
    return p;
}
```

Если изображение корректно сохранилось, выводится следующее.

```
Serial.print("Creating model for #"); Serial.println(id);

p = finger.createModel();
if (p == FINGERPRINT_OK) {
    Serial.println("Prints matched!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Communication error");
    return p;
} else if (p == FINGERPRINT_ENROLLMISMATCH) {
    Serial.println("Fingerprints did not match");
    return p;
}
```

```

    } else {
Serial.println("Unknown error");
return p;
    }

```

Показать результат работы датчика.

```

Serial.print("ID "); Serial.println(id);
p = finger.storeModel(id);
if (p == FINGERPRINT_OK) {
Serial.println("Stored!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
Serial.println("Communication error");
return p;
} else if (p == FINGERPRINT_BADLOCATION) {
Serial.println("Could not store in that location");
return p;
} else if (p == FINGERPRINT_FLASHERR) {
Serial.println("Error writing to flash");
return p;
} else {
Serial.println("Unknown error");
return p;
}
}

```

Если открыть консоль, там будет приглашение к вводу идентификационного номера.

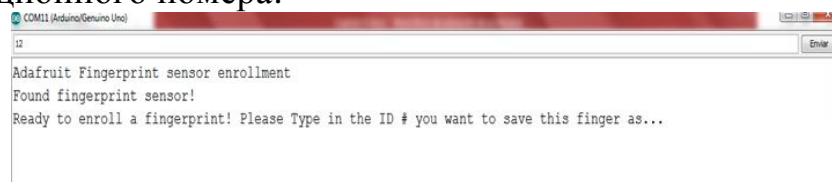


Рис. 3 приглашение к вводу номера

На рисунке 4 показан процесс считывания изображения с датчика отпечатка.

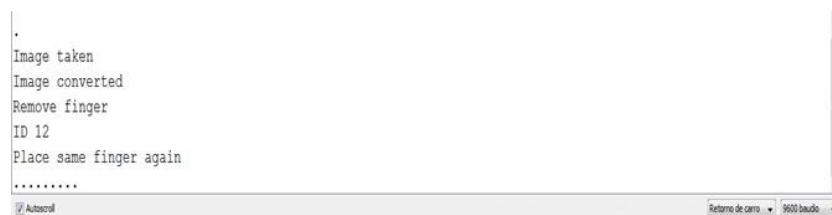


Рис. 4 Процесс считывания отпечатка пальца

На рисунке 5 считывание прошло успешно.

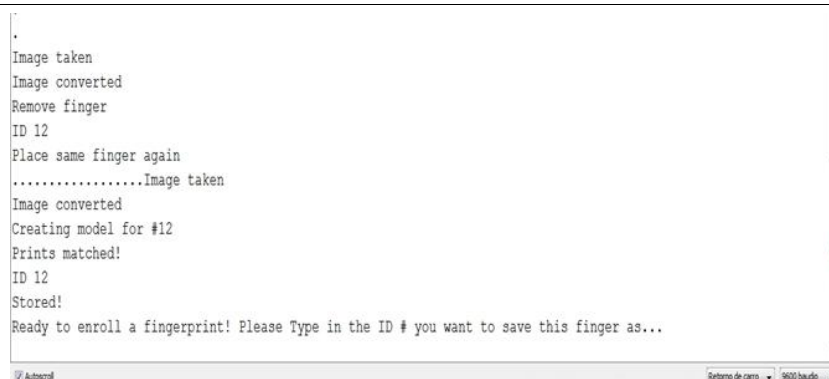


Рис. 5 Рисунок отпечатка пальца успешно сохранен

### Вывод

Дорогостоящие охранные системы с виду кажутся сложными и запутанными, но в их основе лежат те же принципы что излагаются в данной статье. Помимо датчиков отпечатков существует ещё масса датчиков, позволяющих считывать, сравнивать и регистрировать множество процессов и явлений.

### Библиографический список

1. Бастанов Р.Р., Калитаев А.Н., Тутарова В.Д. Алгоритм создания шаблона отпечатка пальца // Актуальные проблемы современной науки, техники и образования 2013. № 71 С. 79-82. URL: <https://elibrary.ru/item.asp?id=20798491> (Дата обращения: 21.12.2018)
2. Волчихин В.И., Фунтиков В.А., Иванов А.И., Секретов М.В., Майоров А.В., Малыгин А.Ю. Способ классификации рисунков отпечатка пальца при аутентификации личности // патент № 2473125, 2013. URL: <https://elibrary.ru/item.asp?id=18859157> (Дата обращения: 21.12.2018)
3. Киндеркнехт Р.Н. Обработка изображения в задаче идентификации по отпечатку пальца // Научные исследования и разработки молодых ученых 2014. № 2 С. 94-98. URL: <https://elibrary.ru/item.asp?id=22623378> (Дата обращения: 21.12.2018)
4. Датчик отпечатка пальца URL: [https://www.gearbest.com/development-boards/pp\\_70487.html?lkid=11637831](https://www.gearbest.com/development-boards/pp_70487.html?lkid=11637831) (Дата обращения 21.12.2018)