

УДК 004.457

ОСОБЕННОСТИ АППАРАТНО-ПРОГРАММНОГО ВЗАИМОДЕЙСТВИЯ СИСТЕМЫ КОНТРОЛЯ ПОСЕЩАЕМОСТИ СТУДЕНТОВ НА ОСНОВЕ АНАЛИЗА ОТПЕЧАТКОВ ПАЛЬЦЕВ

К.А. АПАНЧЁНОК
(Представлено: М.В. МАТЮШ)

Рассматривается работа устройства системы контроля посещаемости студентов с использованием отпечатков пальцев. Представлен способ взаимодействия клиентского приложения для компьютера под управлением Windows и устройства на платформе ArduinoUno. Дано подробное описание передачи данных от приложения к устройству и обратно.

В разрабатываемом приложении и проектированном устройстве требовалось осуществить передачу данных между собой. В качестве канала передачи данных использовался последовательный СОМпорт. В Arduino имеется уже встроенный преобразователь USB-COM, поэтому работа устройство принимает и отправляет данные посредством СОМпорта [1].

Устройство по умолчанию находится в режиме ожидания команды. Всего устройство может обработать набор необходимых для полноценной работы команд (табл. 1). Каждая команда предполагает дальнейшее взаимодействие со сканирующим устройством, т.е. обмен данными между Arduino и сканером отпечатков. Для того чтобы сократить время передачи и обработки команды, было решено использовать номера в качестве команд, вместо их названия.

Таблица 1. – Команды и их функции

Команда	Функция
1	Получение кода отпечатка пальца
2	Передача и запись кода отпечатка пальца в сканер
3	Сканирование пальца и поиск совпадения
4	Очистка БД сканера

При выполнении *первой команды* на сканер отправляется команда считывания отпечатка пальца и записи результата в первый буфер сканера, затем снова команда считывания отпечатка пальца и запись во второй буфер. После отправляется команда на создание усредненного шаблона отпечатка (для того, чтобы ввести погрешность при последующем поиске совпадений с данным отпечатком) и в конце – команда передачи шаблона со сканера на Arduino и команда передачи шаблона с Arduino в приложение.

При выполнении *второй команды* Arduino ожидает приема кода отпечатка пальца определенной длины и номера для записи шаблона в память сканера. Затем Arduino отправляет команду передачи шаблона на сканер и номера для записи в необходимую ячейку памяти.

При выполнении *третьей команды* на сканер отправляется команда для сканирования отпечатка, записи его в буфер и поиск совпадения в базе сканера. Сканер, в свою очередь, отправляет обратно результат, который потом дублируется на отправку в приложение.

При выполнении *четвертой команды* на сканер отправляется команда, которой устройство очищает энергонезависимую память сканера, а также временные буферы.

Таким образом, пользовательское приложение отправляет команду и ожидает приема метасимволов, означающих выполнение команды, а затем продолжает работу с пользователем (псевдокод 1).

Псевдокод 1 – функция отправки команды поиска совпадения

```
String findMatch() {
    String ret = null;
    try {
        data2 = "";
        serialPort.writeString("3\n"); //отправкакоманды 3
        while (true) {sleep(1);
            data = serialPort.readString(); //чтениеответныхданных
            if (scan == 0) { break;}
            if (data != null) {data2 += data;}
```

```

    if (data2.contains("$#")) { //проверка на наличие мета-символов
        int a = data2.indexOf("ID=") + 3;
        int b = data2.indexOf("$");
        ret = data2.substring(a, b); //получениеномерасовпадения
        break; }
    if (data2.contains("Did not find a match")) { break; } //проверкананесовпадение
}
} catch (SerialPortException ex) { Logger.getLogger(Window.class.getName()).log(Level.SEVERE,
null, ex);
} catch (InterruptedException ex) {Logger.getLogger(Window.class.getName()).log(Level.SEVERE,
null, ex);}
data2 = ""; //очистка буфера приема данных
returnret; } //вывод результата функции

```

На устройстве Arduino обрабатываемая команда выполняется и выдает результат как при нахождении совпадения (рис. 1), так и при его отсутствии (рис. 2).

```

FingerprintTest...#
Success!#
SWITCH MODE#
SWITCH = 3
MatchMode#
Scan finger#
Taked#
Converted#
Found a print match!#
Found ID=1#
SWITCH MODE#

```

Рисунок 1. – Результат выполнения команды поиска совпадений при нахождении отпечатка в БД

```

SWITCH MODE#
SWITCH = 3
MatchMode#
Scan finger#
Taked#
Converted#
Did not find a match#
SWITCH MODE#

```

Рисунок 2. – Результат выполнения команды поиска отпечатка при отсутствии совпадения

Разработанное устройство выполнено на базе аппаратной платформы ArduinoUno и сканера отпечатков пальцев ZFM-20 [2]. Данный сканер позволяет получать шаблоны отпечатков за время < 1 секунды, хранить ввстроенной памяти до 1600 значений. Аппаратная часть представляет собой комплекс из двух этих устройств, которые взаимодействуют между собой и связываются с программой на ПК. Связь между ПК и устройством осуществлена посредством COM-порта [3]. Сканер по умолчанию настроен на прием и передачу информации в шестнадцатиричном коде путем формирования пакетов данных (табл. 2).

Таблица 2. – Формат построения пакета данных

Заголовок (2 байта) Обычно значение EF01H	Адрес (4 байта)	Идентификатор пакета данных (1 байт) 01H – командный; 02H – пакет данных; 07H – ответный пакет; 08H – завершающий пакет данных	Размер пакета (2 байта)	Содержание пакета (размер задается в настройках сканера. Максимум 256 байт)	Контрольная сумма (2 байта)
--	-----------------	---	-------------------------	---	-----------------------------

Для корректной и более удобной работы со сканером использована библиотека AdufruitFingerPrintLib [4], которая позволяет использовать большинство функций данного сканера без формирования пакетов данных вручную. На Arduino реализованы функции, необходимые для обмена данными с ПК и сканером.

При включении Arduino происходит авторизация сканера (инициализация порта обмена данными со сканером и проверка его на готовность работы) и подключения с ПК (открытие порта для обмена данными с ПК), затем идет ожидание получения одной из команд от ПК, перечисленных выше. Рассмотрим каждую функцию подробнее в рамках устройства.

Функция получения шаблона отпечатка пальца при выполнении отправляет на сканер команду начала сканирования и ожидает сканирования отпечатка. После того как палец был просканирован, его скан загружается в один из двух буферов сканера для дальнейшей обработки. Затем сканер снова переходит в режим сканирования и записывает второй скан в другой буфер. Это необходимо для построения усредненного значения шаблона отпечатка пальца. Такой шаблон будет распознаваться сканером даже при неполном соответствии (70–80%), не допуская при этом ошибок при поиске отпечатка в БД. После составления шаблона на сканер подается команда для выгрузки шаблона со сканера на Arduino, которая, в свою очередь, отправляет его через СОМ-порт на ПК (псевдокод 2).

Псевдокод 2 – функция получения шаблона отпечатка пальца со сканера

```
int cas1(){
  Serial.println("GetCharFromIMG mode");
  Serial.println("Scan finger#");
  int p = -1;
  p = GetImage(1); //отправка команды получения скана пальца
  if(p != -1){
    delay(2);
    p = ImageToTz(1); //конвертирование изображения в массив символов и запись в буфер №1
    delay(1000);
    while (p != FINGERPRINT_NOFINGER) {
      p = finger.getImage(); //отправка команды получения скана пальца для проверки на наличие
      пальца на сканере (проверка ввода)
    }
    p = GetImage(1); //отправка команды получения скана пальца
    if(p != -1){
      delay(2);
      p = ImageToTz(2); //конвертирование изображения в массив символов и запись в буфер № 2
      p = createModel(); //создание усредненного шаблона отпечатка пальцев при совпадении
      более 90% (то есть при скане одного и того же пальца) и запись результата в буфер № 1
      if(p == 0)
      {
        downloadModel(1); //если создание успешно, то отправляется команда получения значения
        из буфера № 1
      }
      else{
        Serial.println("NOTMATCH"); } } //иначе, вывод сообщения «НЕТ СОВПАДЕНИЯ»
```

Функция добавления шаблона отпечатка во внутреннюю память устройства при вызове запускает прием данных шаблона отпечатка пальца с ПК, затем принимает уникальный идентификатор данного отпечатка. Затем запускается функция отправки шаблона отпечатка в буфер сканера. После выполняется функция записи данного отпечатка в память по указанному ID (псевдокод 3).

Псевдокод 3 – функция добавления шаблона отпечатка в память

```
int cas2(void){
  Serial.println("DownloadCharToCharbuf mode");
  uint8_t packet[688];
  Serial.println("Start recive char...#");
  int p = 0;
  p = reciveChar(packet); //получение данных с СОМ-порта от ПК
  Serial.println("End recive#");
```

```

Serial.println("Enter ID#");
int id = readnumber(); //получениеидентификатора
Serial.print("ID=");
Serial.println(id);
Serial.println("Upload#");
uploadModel(1, packet); //передачашаблонаотпечаткапальцавбуфер №1
Serial.println("Storing#");
storeModel(id); //записьзначениябуфера №1 впамятьподномеромid
}

```

Функция поиска совпадений по скану при выполнении отправляет команду захвата отпечатка пальца со сканера. Затем при наличии значения пальца конвертируется в буфер № 1, и отправляется команда получения номера совпадения (псевдокод 4).

Псевдокод 4 – поиск совпадения в памяти устройства

```

int cas3(void){
Serial.println("MatchMode#");
Serial.println("Scan finger#");
int p = -1;
p = GetImage2(); //получение скана пальца
if(p != -1){
delay(2);
p = ImageToTz(1); //конвертирование изображения в символьное представление
getMatch(); //поиск совпадения о сходстве отпечатка с уже имеющимися в памяти
}
}

```

Функция очистки памяти сканера предназначена для очистки шаблонов перед загрузкой новых шаблонов какой-либо группы (псевдокод 5).

Псевдокод 5 – функция очистки памяти сканера

```

intcas4(void){
Serial.println("EmptyDataBase");
emptyBase();
}

```

ЛИТЕРАТУРА

1. Работа с COM-портом Arduino [Электронный ресурс]. – 2017. – Режим доступа: <http://arduino.ru/forum/programmirovanie/rabota-s-com-portom>. – Дата доступа: 10.05.2017.
2. ZFM-20dingerprintmoduledatasheet [Электронный ресурс]. – 2017. – Режим доступа: https://github.com/adafruit/Adafruit-Fingerprint-Sensor-Library/blob/master/documentation/ZFM-20_Fingerprint_Module.pdf. – Дата доступа: 10.05.2017.
3. Работа с COM-портом в Java [Электронный ресурс]. – 2017. – Режим доступа: <https://ru.stackoverflow.com/questions/401234/java-%D0%A7%D1%82%D0%B5%D0%BD%D0%B8%D0%B5-%D1%81-com-%D0%BF%D0%BE%D1%80%D1%82%D0%B0>. – Дата доступа: 10.05.2017.
4. Arduino library for interfacing to the fingerprint sensor [Электронный ресурс]. – 2017. – Режим доступа: <https://github.com/adafruit/Adafruit-Fingerprint-Sensor-Library/>. – Дата доступа: 10.05.2017.