CEUR-WS.org/Vol-2864/paper23.pdf

Simple Autonomous Security System Based On Arduino UNO Platform and Fingerprint Scanner Module: A Study Case

Volodymyr Rusyn[a], Sergey Subbotin[b] and Aceng Sambas[c]

[a] *Yuriy Fedkovych Chernivtsi National University, Kotsybynsky str., 2, Chernivtsi, 58012, Ukraine*
[b] *National university “Zaporizhzhia polytechnic”, Zhukovsky str., 64, Zaporizhzhia, 69063, Ukraine*
[c] *Universitas Muhammadiyah Tasikmalaya, Jl. Tamansari No. KM 2,5, Mulyasari, Kec. Tamansari, Tasikmalaya, Jawa Barat, 46196, Indonesia*

Abstract

In this paper, we presented a simple autonomous security system. This security system based on the Arduino Uno and fingerprint scanner module. The Arduino was connected to a computer through the USB port and programmed using a language similar to C++. Programming code (sketch) was uploaded into Arduino using program software ArduinoIDE. Connection scheme and experimental results are presented and can be used for education.

Keywords 1

Fingerprint scanner, security system, Arduino

1. Introduction

In the modern world, biometric information protection is one of the most effective methods for its preservation. Thanks to the process of authenticating a person, that is, comparing his characteristics with the characteristics previously entered into the system, it is possible to determine with maximum accuracy whether a given person has access to the requested information or still does not. Nowadays, everyone cares about how to protect their data as efficiently as possible. In an effort to secure personal information, mankind has been developing more and more security systems for many years. Biometric protection turned out to be one of the most effective such systems.

Biometric security technologies are becoming more and more everyday attributes of the ordinary person's life. In recent years, they have become extremely widespread in the production of mobile technology: fingerprint scanners, face geometry scanners, and voice recognizers are built into smartphones [1-7]. All this makes the life of an ordinary user much more convenient, quieter and safer. Precisely the same devices are now very often equipped with personal computers, smart homes and many other computerized equipment.

Biometric systems are also quite popular among enterprises, where the entrance to the territory or access to information is open to a large, but limited number of people. At the checkpoints of such enterprises, specialized equipment is usually installed that allows you to identify “your” person through the iris of the eye, the geometry of the hand, face or through the drawing of papillary lines on the pads of the fingers, and sometimes all together.

The introduction of biometrics opens up a sea of different possibilities in banking, making life much easier not only for the bank, but also for its customers. By introducing biometric technologies, bank managers want to reduce the number of data thefts, cases of fraud and greatly simplify the procedure for a client's work with an ATM.

In this paper, we proposed a simple autonomous security system that based on the Arduino Uno and fingerprint scanner module that can be used for education.

CMIS-2021: The Fourth International Workshop on Computer Modeling and Intelligent Systems, April 27, 2021, Zaporizhzhia, Ukraine
EMAIL: rusyn_v@ukr.net (V. Rusyn); subbotin@zntu.edu.ua (S. Subbotin); acenx.bts@gmail.com (A. Sambas)
ORCID: 0000-0001-6219-1031 (V. Rusyn); 0000-0001-5814-8268 (S. Subbotin); 0000-0002-1623-0770 (A. Sambas)
© 2021 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

2. ARDUINO board

The Arduino can be compared to a computer motherboard, which also has a processor under the cooler. The motherboard with the processor performs the same functions as the Arduino Uno. Other devices are connected to it: the display, disk drives, hard drives, and all other peripherals. You can also connect many different devices [8].

The uniqueness of the Arduino is that now even an inexperienced radio amateur will be able to create various complex digital devices, without going into theory. Previously, for example, this required programmers, deep knowledge of digital electronics, programming. Arduino was created so that almost any user could create their own interesting and useful device.

Fig. 1 shows Arduino Uno that we used for our project.

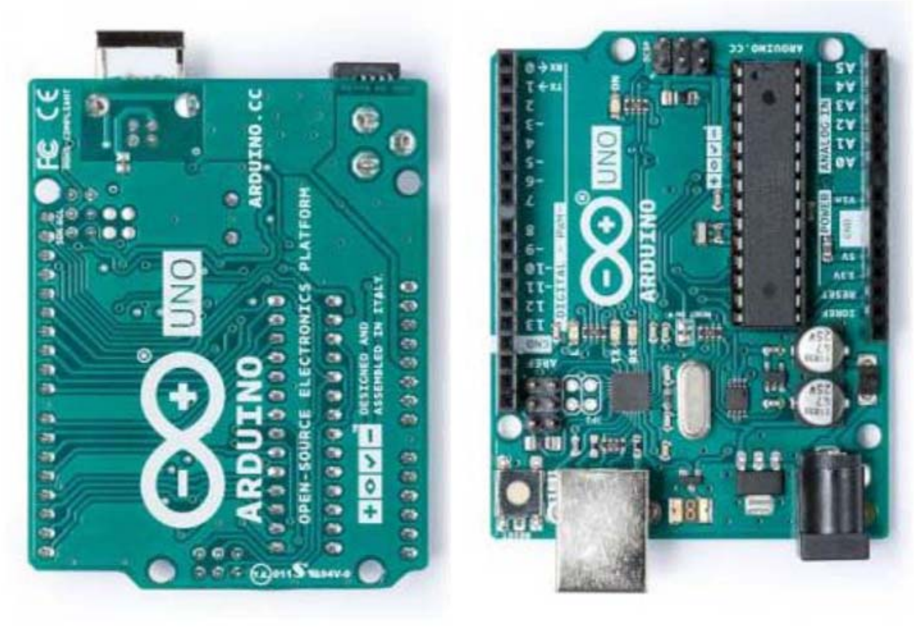


Figure 1: Arduino Uno board

Arduino Uno controller built on ATmega328. The platform has 16 MHz crystal oscillator, 6 analog inputs, 14 digital inputs / outputs (6 of which can be used as PWM outputs), a USB connector, an ICSP connector, a power connector and a reset button. For work, we must connect the platform to the computer via a USB cable, or apply power using battery or AC/DC adapter.

The Arduino Uno board has 3 ways to connect power on board: via USB, via an external power connector and via the Vin connector, brought out to one of the combs on the side. The platform has a built-in stabilizer on board, which allows not only to automatically select the power source, but also to equalize the current to stable 5 Volts, which the controller needs to work.

External power can be supplied either directly from the USB port of the computer, or from any AC / DC power supply via the power connector or USB.

The board has several pins that allow powering connected sensors, sensors and actuators from it. All of these findings are labeled:

- Vin – power input, used to receive power from an external source. Through the data output, only power is supplied to the board, it is impossible to receive power from there for external devices. It is recommended to apply a voltage in the range from 7 V to 20 V to the Vin input, in order to avoid overheating and burning of the built-in stabilizer.
- 5 V – a five-volt voltage source for powering external devices. When the board receives power from any other source (USB, power connector or Vin), you can always get a stable voltage of 5 Volts on this pin. It can be displayed on a breadboard or fed directly to the desired device.

- 3V3 – source of 3.3 Volt voltage for powering external devices. Works on the same principle as the 5 V pin. From this leg, you can also output the voltage to the breadboard, or apply it directly to the required sensor / sensor.
- GND – pin for ground connection. Needed to create a closed circuit when connected to Vin, 5 V or 3V3 pins. In all cases, the GND leg must be output as a minus, otherwise the circuit will not be closed and the power (both external and internal) will not be supplied.

The Arduino Uno platform has an ATmega328 microcontroller on board, which has Flash, SRAM and EEPROM memory:

- FLASH – 32 kB (0.5 kB is used for bootloader storage SRAM (RAM).
- 2 kB EEPROM.
- 1 kB (available with EEPROM library).

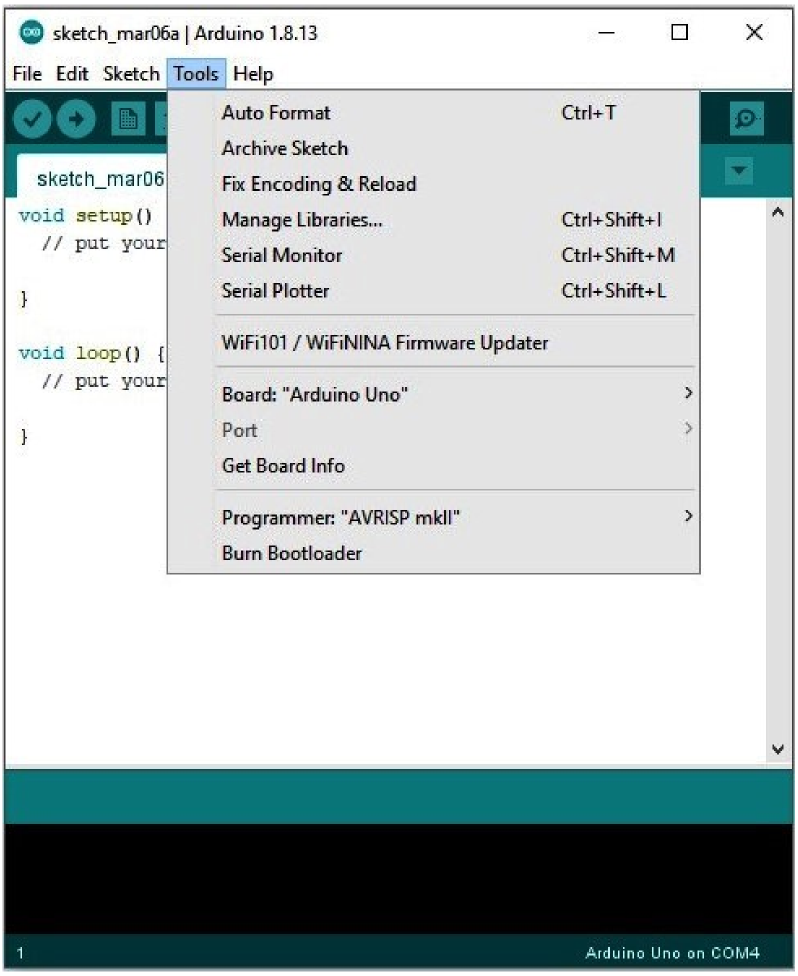
3. Program realization

The Arduino integrated development environment is an environment in which an Arduino board can be programmed.

A written program or code is called SKETCH. In this work, the Arduino IDE is used as an environment in which the Arduino Uno program is written, compiled and uploaded on the Arduino board as depicted in Fig. 2.

The Arduino can be connected to a computer through the USB port and programmed using a language similar to C++.

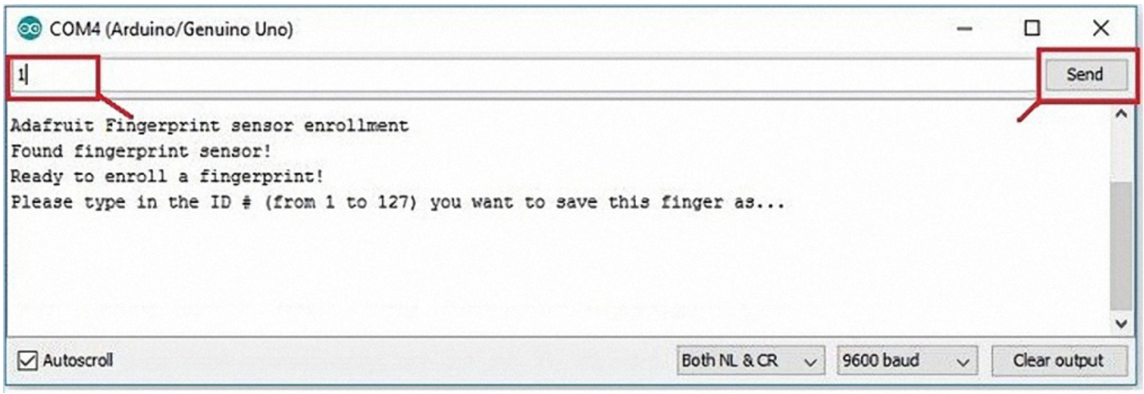
In the first stage, we must control the fingerprint sensor module with the Arduino. It is by using the Adafruit library for this sensor.


sketch_mar06a | Arduino 1.8.13
File
Edit
Sketch
Tools
Help
sketch_mar06
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Uno"
Port
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
1
Arduino Uno on COM4

```
void setup()
  // put your

}

void loop() {
  // put your

}
```

Figure 2: Arduino IDE

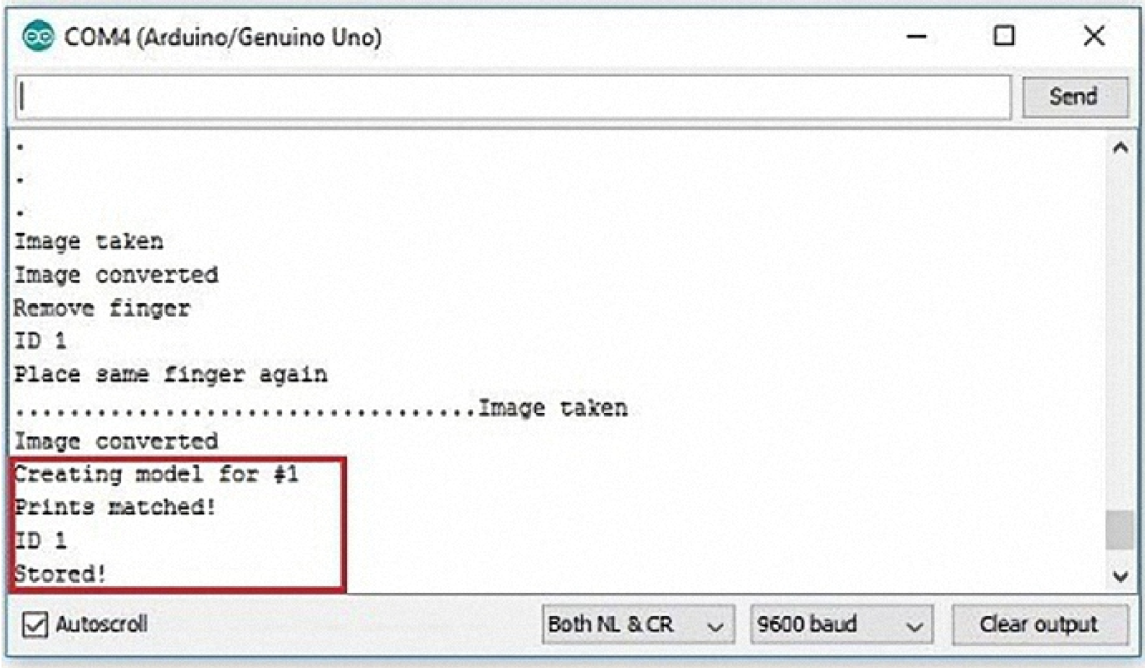
Having the fingerprint sensor module wired to the Arduino, we must follow the next steps to enroll a new fingerprint. Make sure you have installed the Adafruit Fingerprint Sensor library previously (Fig. 3).



```
Adafruit Fingerprint sensor enrollment
Found fingerprint sensor!
Ready to enroll a fingerprint!
Please type in the ID # (from 1 to 127) you want to save this finger as...
```

Figure 3: Enroll a new fingerprint

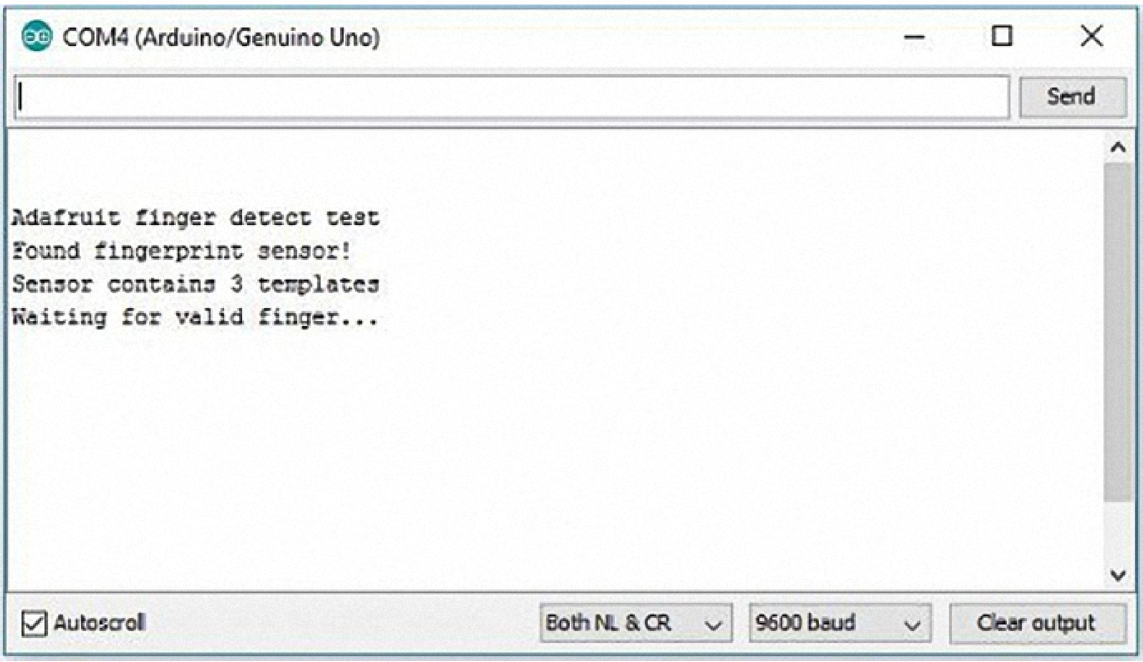
You will be asked to place the same finger twice on the scanner. If you get the "Prints matched!" message, the fingerprint was successfully stored. If not, we repeat the process, until was succeed (Fig. 4).



```
.
.
.
Image taken
Image converted
Remove finger
ID 1
Place same finger again
.................................Image taken
Image converted
Creating model for #1
Prints matched!
ID 1
Stored!
```

Figure 4: Interface that demonstrate that a fingerprint was successfully stored

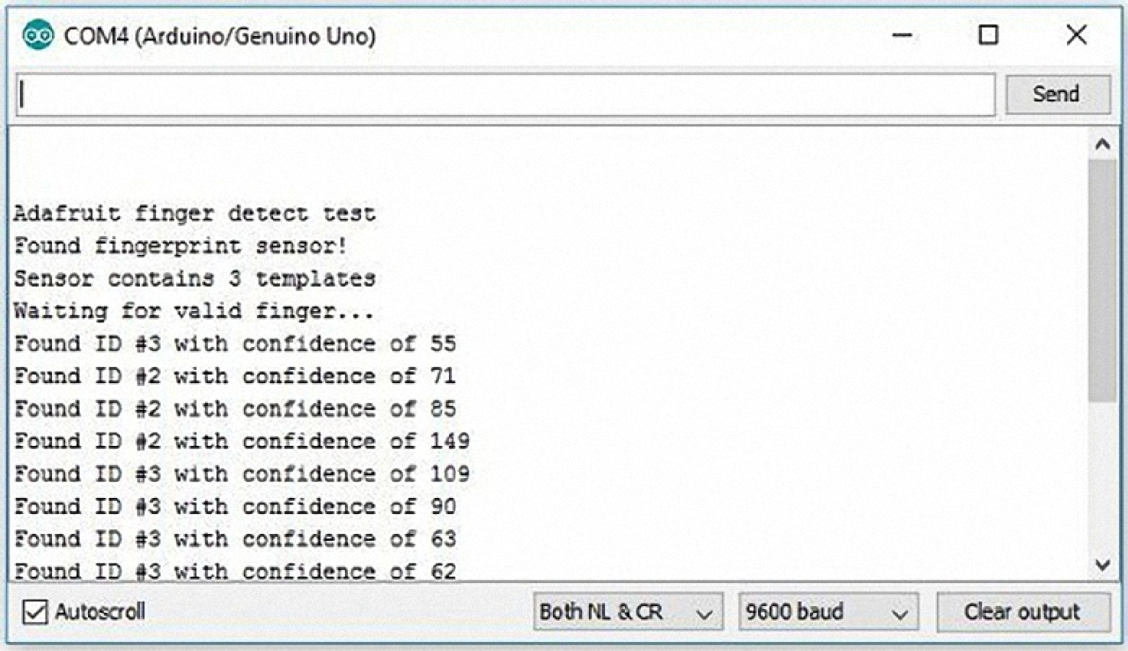
Now, we should have several fingerprints saved on different IDs (Fig. 5).



```
Adafruit finger detect test
Found fingerprint sensor!
Sensor contains 3 templates
Waiting for valid finger...
```

Figure 5: Interface that demonstrate that a sensor contains three templates

Next stage: place the finger on the scan for identified . On the serial monitor, we can see the ID user that matches the fingerprint. It also shows the confidence – the higher the confidence, the similar the fingerprint is with the stored fingerprint (Fig. 6).

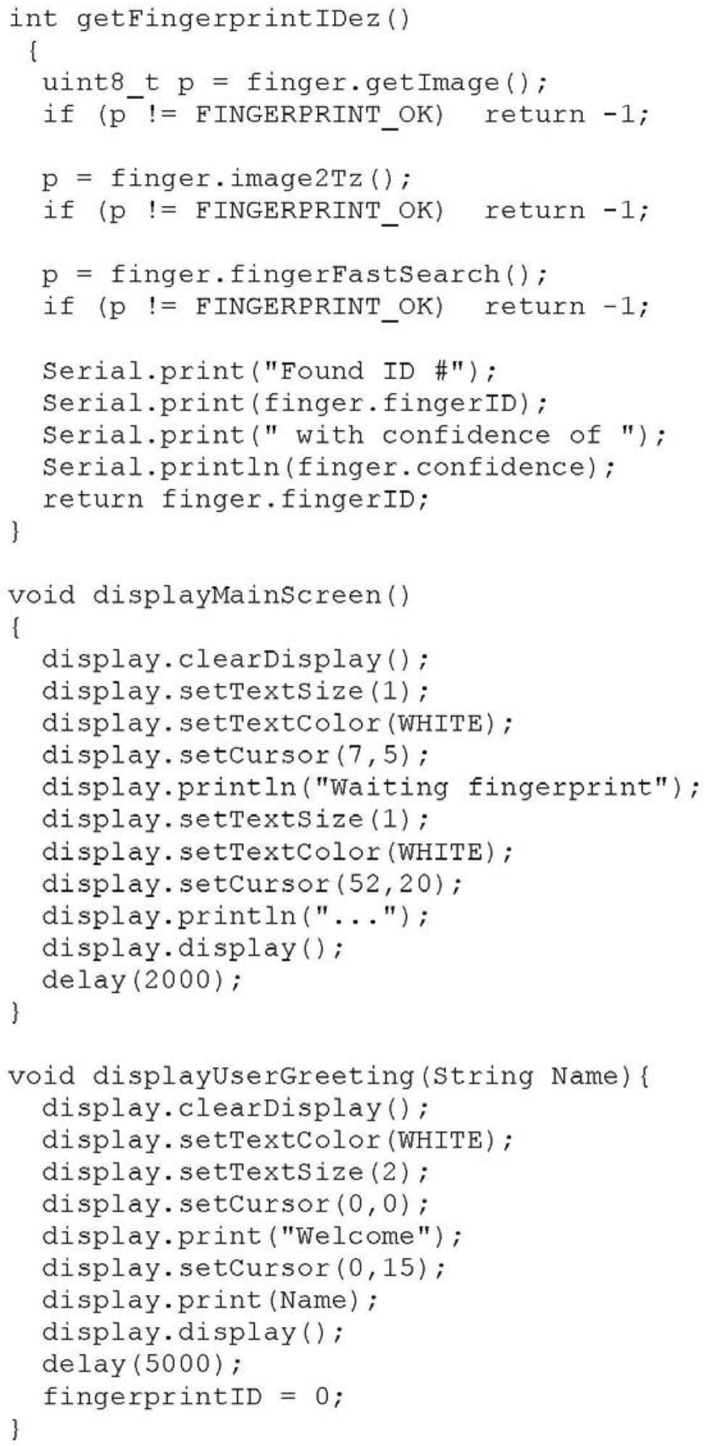


```
Adafruit finger detect test
Found fingerprint sensor!
Sensor contains 3 templates
Waiting for valid finger...
Found ID #3 with confidence of 55
Found ID #2 with confidence of 71
Found ID #2 with confidence of 85
Found ID #2 with confidence of 149
Found ID #3 with confidence of 109
Found ID #3 with confidence of 90
Found ID #3 with confidence of 63
Found ID #3 with confidence of 62
```

Figure 6: Interface that demonstrate ID with confidence

The programming code:

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define OLED_RESET 4
Adafruit_SSD1306 display(OLED_RESET);

#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2, 3);

Adafruit_Fingerprint finger =
Adafruit_Fingerprint(&mySerial);
int fingerprintID = 0;
String IDname;

void setup(){
  Serial.begin(9600);
  finger.begin(57600);

  if (finger.verifyPassword()) {
    Serial.println("Fingerprint sensor found!");
  }
  else {
    Serial.println("Fingerprint sensor did not find");
    while (1) { delay(1); }
  }

  Wire.begin();
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  displayMainScreen();
}

void loop(){
  displayMainScreen();
  fingerprintID = getFingerprintIDez();
  delay(50);
  if(fingerprintID == 1 || fingerprintID == 2 ||
fingerprintID == 3)
 {
    IDname = "Volodymyr";
    displayUserGreeting(IDname);
  }
  else if(fingerprintID == 2){
    IDname = "Sergey";
    displayUserGreeting(IDname);
  }
  else if(fingerprintID == 3){
    IDname = "Aceng";
    displayUserGreeting(IDname);
  }
}
```



```
int getFingerprintIDez()
 {
  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK)  return -1;

  p = finger.image2Tz();
  if (p != FINGERPRINT_OK)  return -1;

  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK)  return -1;

  Serial.print("Found ID #");
  Serial.print(finger.fingerID);
  Serial.print(" with confidence of ");
  Serial.println(finger.confidence);
  return finger.fingerID;
}

void displayMainScreen()
{
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(7,5);
  display.println("Waiting fingerprint");
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(52,20);
  display.println("...");
  display.display();
  delay(2000);
}

void displayUserGreeting(String Name){
  display.clearDisplay();
  display.setTextColor(WHITE);
  display.setTextSize(2);
  display.setCursor(0,0);
  display.print("Welcome");
  display.setCursor(0,15);
  display.print(Name);
  display.display();
  delay(5000);
  fingerprintID = 0;
}
```

4. Experimental realization

Fig. 7 shows fingerprint sensor module that made fingerprint recognition more accessible and easy to add to different projects. This means that is super easy to make fingerprint collection, search, registration and comparison. This module work with any microcontroller or system with TTL serial and come with FLASH memory to store the fingerprints. This fingerprint sensor module can be added to security systems, door locks, time attendance systems, and much more.

Specifications of the fingerprint sensor module:

- Voltage supply: DC 3.6 to 6.0 V.
- Current supply: < 120mA.
- Backlight color: green.
- Interface: UART.
- Bad rate: 9600.
- Safety level: five (from low to high: 1,2,3,4,5).
- False Accept Rate (FAR): < 0.001 % (security level 3).
- False Reject Rate (FRR): < 1.0 % (security level 3).
- Able to store 127 different fingerprints.

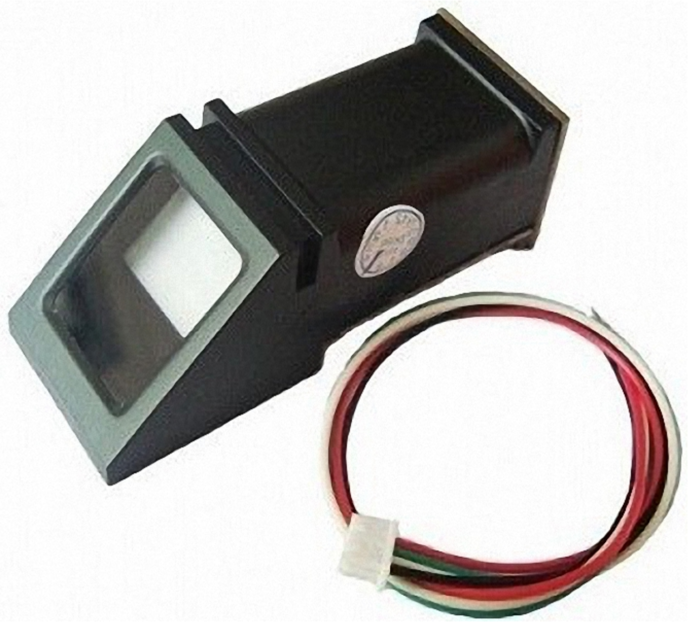


Figure 7: Fingerprint sensor module

Fig. 8 shows pins of the fingerprint sensor module.

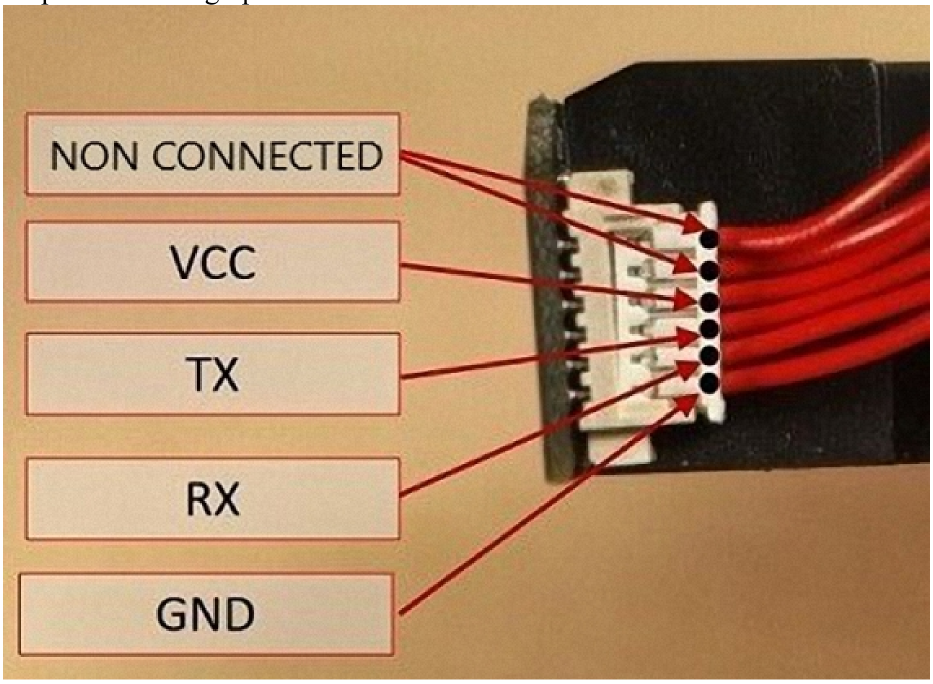


Figure 8: Pins of the fingerprint sensor module

The fingerprint sensor module used in this project came with really thin wires, so soldering breadboard-friendly wires was needed. We recommend using different colors according to the pin function.

In our case:

- DNC – white wires.
- VCC – red wire.
- TX – blue wire.
- RX – green wire.
- GND – black wire.

The connection scheme is quite simple and it's rather difficult to make a mistake (Fig. 9).

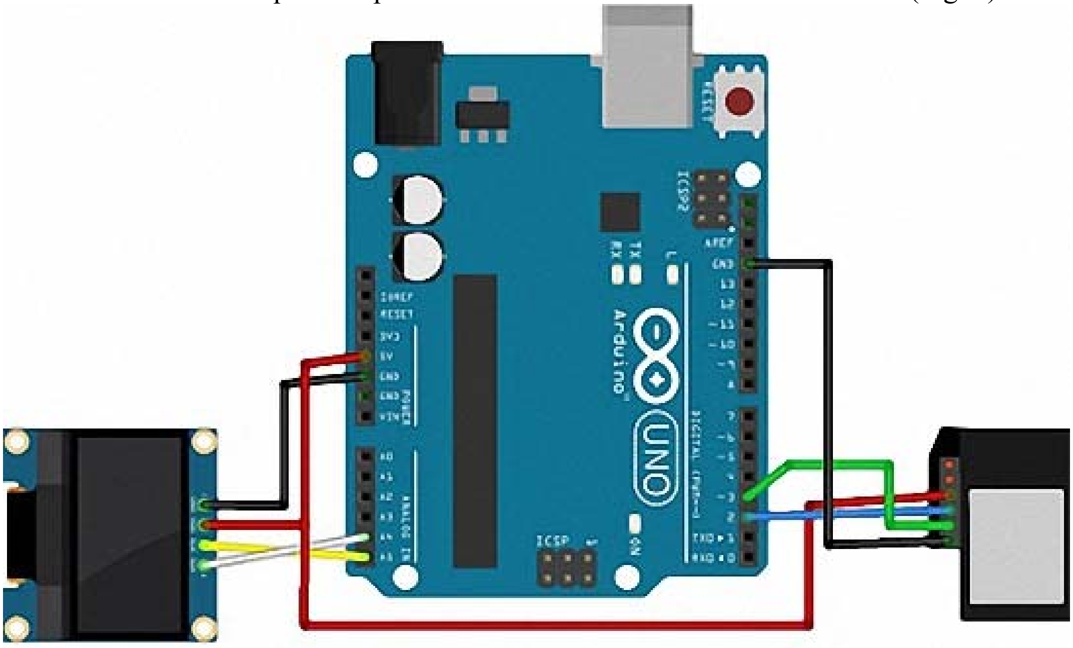


Figure 9: The connection scheme

Fig. 10 shows connection scheme for experimental demonstration of the simple autonomous security system.

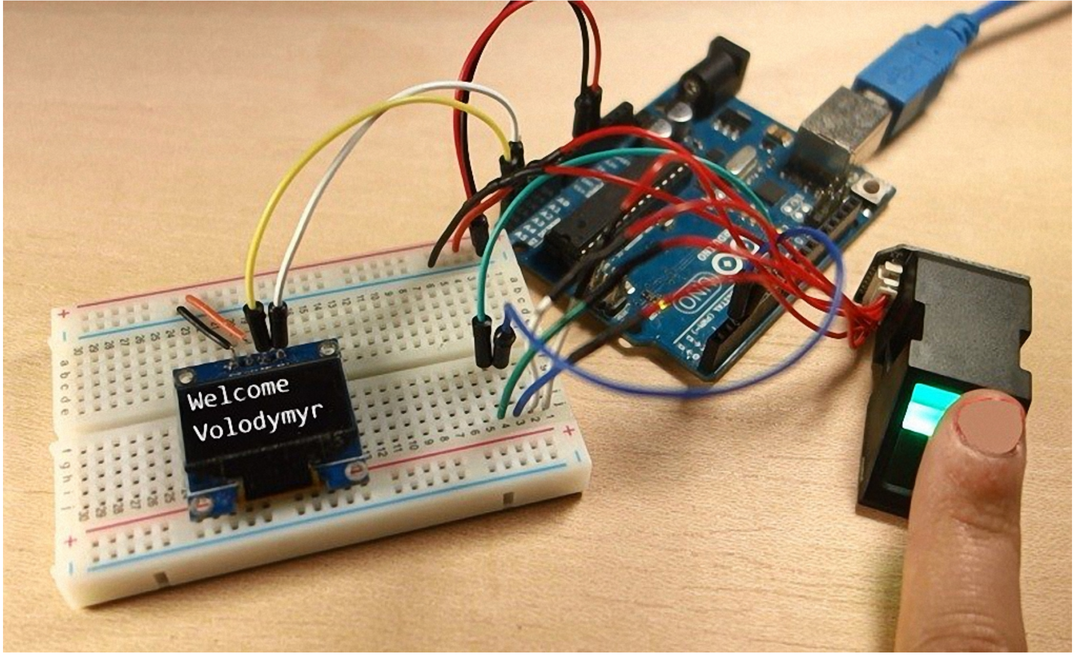


Figure 10: Connection scheme for experimental demonstration

5. Conclusions

The simple autonomous security system that can be used for individual or different commercial security as example personal data, based on the fingerprint scanner module and Arduino Uno.

The Arduino was connected to a computer through the USB port and programmed using a language similar to C++. Programming code (sketch) was uploaded into Arduino using program software ArduinoIDE. Connection scheme and experimental results are also presented.

6. References

[1] Y. Elmir, O. Ghazaoui, F. Boukenni, Multimodal Biometrics System's Resistance to noise (Fingerprint and Voice). CEUR Workshop Proceedings 942 (2012) 25-28.

[2] S. Xia, Y. Liu, G. Yuan, M. Zhu, Z. Wang, Indoor fingerprint positioning based on Wi-Fi: an overview, ISPRS Int. J. Geo-Inf. 6 (2017) 135.

[3] H. Suining, S. H. Gary Chan, Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons, IEEE Commun. Surv Tut, 18 (2016) 466-490.

[4] G. Li, C. Busch and B. Yang, "A novel approach used for measuring fingerprint orientation of arch fingerprint," 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, 2014, pp. 1309-1314. doi: 10.1109/MIPRO.2014.6859770

[5] S. P. Sandip and P. H. Zope, "Selective review of fingerprint enhancement, classification and matching techniques," 2015 IEEE Bombay Section Symposium (IBSS), Mumbai, 2015, pp. 1-6. doi: 10.1109/IBSS.2015.7456656

[6] G. Caso, L. De Nardis, On the Applicability of Multi-wall Multi-floor Propagation Models to WiFi Fingerprinting Indoor Positioning, Future Access Enablers for Ubiquitous and Intelligent Infrastructures: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Springer, 2015.

[7] Tomas Trainys, Algimantas Venčkauskas. Encryption Keys Generation Based on Bio-Cryptography Finger Vein Method. CEUR Workshop Proceedings 2145 (2018) 106-111.

[8] Rusyn V, Subbotin S., Sambas A., Analysis and Experimental Realization of the Logistic Map Using Arduino Pro Mini. CEUR Workshop Proceedings 2608 (2020) 300-310.

View publication stats