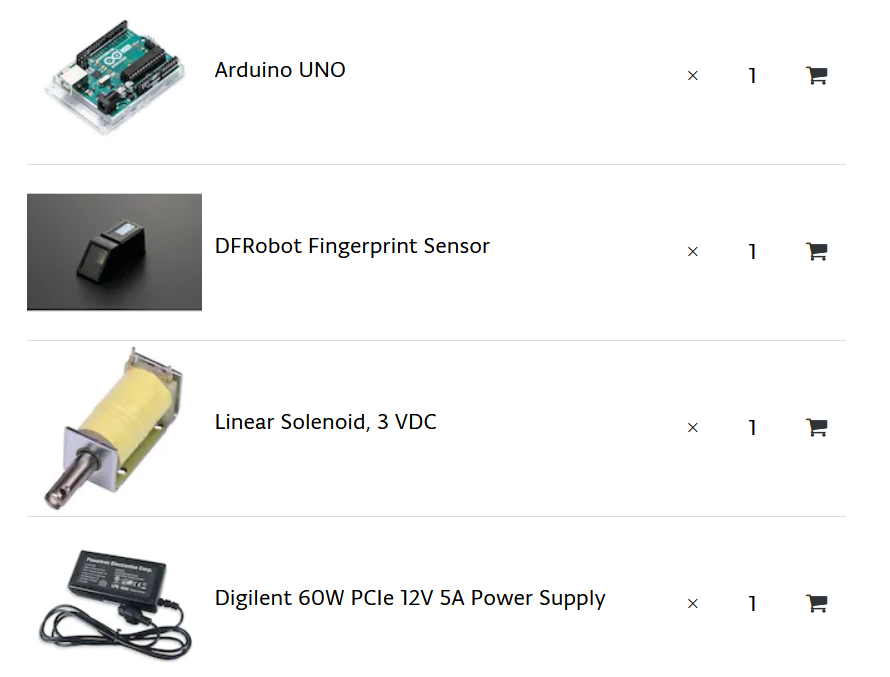
**Fingerprint door lock systém**

<https://create.arduino.cc/projecthub/diyprojectslab/diy-fingerprint-door-lock-system-040249>

<https://www.youtube.com/watch?v=k3Vl5-ODLAA>

Components and supplies:

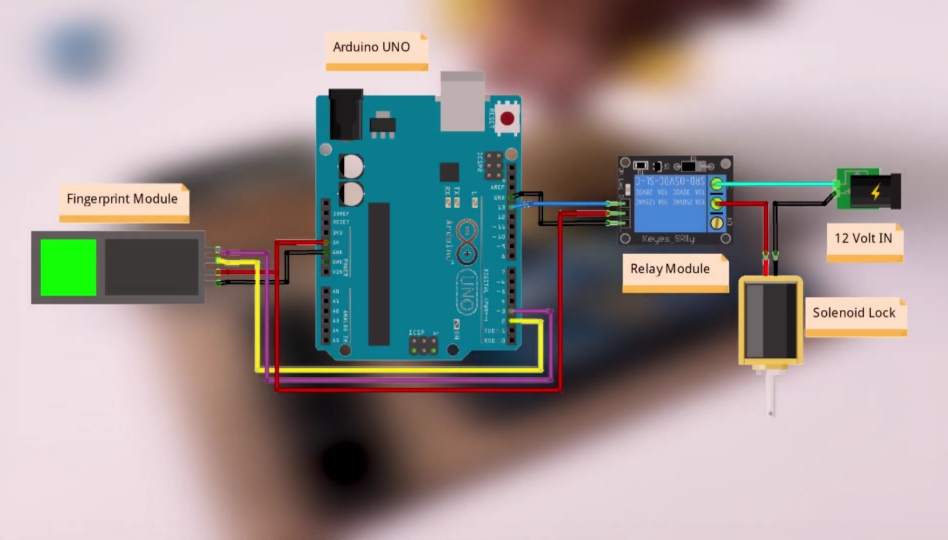


**Upload the Main Code**

1. Open arduino IDE.
2. Under the FILE menu, and Examples, find the adafruits finger print library.
3. Select the fingerprint example.
4. Upload the fingerprint example.
5. Open the Serial Monitor.
6. Select the baud rate to be 9600.
7. Place the finger you need to test on the sensor.

**Library:**

[**https://github.com/adafruit/Adafruit-Fingerprint-Sensor-Library**](https://github.com/adafruit/Adafruit-Fingerprint-Sensor-Library)



**Code**:

#include

int getFingerprintIDez();

// pin #2 is IN from sensor (GREEN wire)

// pin #3 is OUT from arduino (WHITE wire)

SoftwareSerial mySerial(2, 3);

Adafruit\_Fingerprint finger = Adafruit\_Fingerprint(&mySerial);

void setup()

{

Serial.begin(9600);

Serial.println(“fingertest”);

pinMode(13, OUTPUT);

digitalWrite(13, LOW);

// set the data rate for the sensor serial port

finger.begin(57600);

if (finger.verifyPassword()) {

Serial.println(“Found fingerprint sensor!”);

} else {

Serial.println(“Did not find fingerprint sensor :(“);

while (1);

}

Serial.println(“Waiting for valid finger…”);

}

void loop() // run over and over again

{

getFingerprintIDez();

delay(50); //don’t ned to run this at full speed.

}

uint8\_t getFingerprintID() {

uint8\_t p = finger.getImage();

switch (p) {

case FINGERPRINT\_OK:

Serial.println(“Image taken”);

break;

case FINGERPRINT\_NOFINGER:

Serial.println(“No finger detected”);

return p;

case FINGERPRINT\_PACKETRECIEVEERR:

Serial.println(“Communication error”);

return p;

case FINGERPRINT\_IMAGEFAIL:

Serial.println(“Imaging error”);

return p;

default:

Serial.println(“Unknown error”);

return p;

}

// OK success!

p = finger.image2Tz();

switch (p) {

case FINGERPRINT\_OK:

Serial.println(“Image converted”);

break;

case FINGERPRINT\_IMAGEMESS:

Serial.println(“Image too messy”);

return p;

case FINGERPRINT\_PACKETRECIEVEERR:

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;

}

// OK converted!

p = finger.fingerFastSearch();

if (p == FINGERPRINT\_OK) {

Serial.println(“Found a print match!”);

} else if (p == FINGERPRINT\_PACKETRECIEVEERR) {

Serial.println(“Communication error”);

return p;

} else if (p == FINGERPRINT\_NOTFOUND) {

Serial.println(“Did not find a match”);

return p;

} else {

Serial.println(“Unknown error”);

return p;

}

// found a match!

Serial.print(“Found ID #”); Serial.print(finger.fingerID);

Serial.print(” with confidence of “); Serial.println(finger.confidence);

}

// returns -1 if failed, otherwise returns ID #

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;

// found a match!

Serial.print(“Found ID #”); Serial.print(finger.fingerID);

Serial.print(” with confidence of “); Serial.println(finger.confidence);

digitalWrite(13, HIGH);

delay(3000);

digitalWrite(13, LOW);

return finger.fingerID;

}

#include

#include

int getFingerprintIDez();

// pin #2 is IN from sensor (GREEN wire)

// pin #3 is OUT from arduino (WHITE wire)

SoftwareSerial mySerial(2, 3);

Adafruit\_Fingerprint finger = Adafruit\_Fingerprint(&mySerial);

void setup()

{

Serial.begin(9600);

Serial.println(“fingertest”);

pinMode(13, OUTPUT);

digitalWrite(13, LOW);

// set the data rate for the sensor serial port

finger.begin(57600);

if (finger.verifyPassword()) {

Serial.println(“Found fingerprint sensor!”);

} else {

Serial.println(“Did not find fingerprint sensor :(“);

while (1);

}

Serial.println(“Waiting for valid finger…”);

}

void loop() // run over and over again

{

getFingerprintIDez();

delay(50); //don’t ned to run this at full speed.

}

uint8\_t getFingerprintID() {

uint8\_t p = finger.getImage();

switch (p) {

case FINGERPRINT\_OK:

Serial.println(“Image taken”);

break;

case FINGERPRINT\_NOFINGER:

Serial.println(“No finger detected”);

return p;

case FINGERPRINT\_PACKETRECIEVEERR:

Serial.println(“Communication error”);

return p;

case FINGERPRINT\_IMAGEFAIL:

Serial.println(“Imaging error”);

return p;

default:

Serial.println(“Unknown error”);

return p;

}

// OK success!

p = finger.image2Tz();

switch (p) {

case FINGERPRINT\_OK:

Serial.println(“Image converted”);

break;

case FINGERPRINT\_IMAGEMESS:

Serial.println(“Image too messy”);

return p;

case FINGERPRINT\_PACKETRECIEVEERR:

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;

}

// OK converted!

p = finger.fingerFastSearch();

if (p == FINGERPRINT\_OK) {

Serial.println(“Found a print match!”);

} else if (p == FINGERPRINT\_PACKETRECIEVEERR) {

Serial.println(“Communication error”);

return p;

} else if (p == FINGERPRINT\_NOTFOUND) {

Serial.println(“Did not find a match”);

return p;

} else {

Serial.println(“Unknown error”);

return p;

}

// found a match!

Serial.print(“Found ID #”); Serial.print(finger.fingerID);

Serial.print(” with confidence of “); Serial.println(finger.confidence);

}

// returns -1 if failed, otherwise returns ID #

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;

// found a match!

Serial.print(“Found ID #”); Serial.print(finger.fingerID);

Serial.print(” with confidence of “); Serial.println(finger.confidence);

digitalWrite(13, HIGH);

delay(3000);

digitalWrite(13, LOW);

return finger.fingerID;

}