

```
//This program pin is work for the product like https://www.flyrobo.in/finger\_print\_sensor  
//here wire connection is as  
//fingerprint sensor white wire - pin 3, yellow wire - pin 2, red wire - "+5v" , black wire - "GND"  
//servo wire - pin 4
```

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

```
SoftwareSerial mySerial(2, 3);
```

```
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
```

```
void setup()  
{  
  Serial.begin(9600);  
  while (!Serial); // For Yun/Leo/Micro/Zero/...  
  delay(100);  
  Serial.println("fingertest");  
  pinMode(12, OUTPUT);
```

```
  // 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) { delay(1); }  
  }
```

```
  finger.getTemplateCount();  
  Serial.print("Sensor contains "); Serial.print(finger.templateCount); Serial.println(" templates");  
  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.  
  digitalWrite(12, LOW);
```

```
}
```

```
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_PACKETRECEIVEERR:  
            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_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;  
}
```

```

}

// OK converted!
p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK) {
    Serial.println("Found a print match!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    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);

return finger.fingerID;
}

// 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!

    digitalWrite(12, HIGH);
    delay(3000);
    digitalWrite(12, LOW);

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

```

```
    return finger.fingerID;  
}
```