

## DIY Fingerprint Scanning Garage Door Opener

by **nodcah** on December 21, 2013

### Table of Contents

DIY Fingerprint Scanning Garage Door Opener .....	1
Intro: DIY Fingerprint Scanning Garage Door Opener .....	2
Step 1: Materials .....	2
Step 2: The Circuit .....	4
Step 3: Assemble the Serial LCD Kit .....	5
Step 4: Assembling the Circuit Boards .....	5
Step 5: Programing the ATmega328 .....	6
File Downloads .....	6
Step 6: Setting Up the Fingerprint Scanner .....	7
File Downloads .....	8
Step 7: Programing the ATtiny85 .....	8
File Downloads .....	9
Step 8: The Final Code .....	9
File Downloads .....	13
Step 9: The 3D Printed Case .....	13
File Downloads .....	14
Step 10: Prepare the Garage .....	14
Step 11: Testing! .....	15
Related Instructables .....	16
Advertisements .....	16
Comments .....	16



Author: **nodcah**

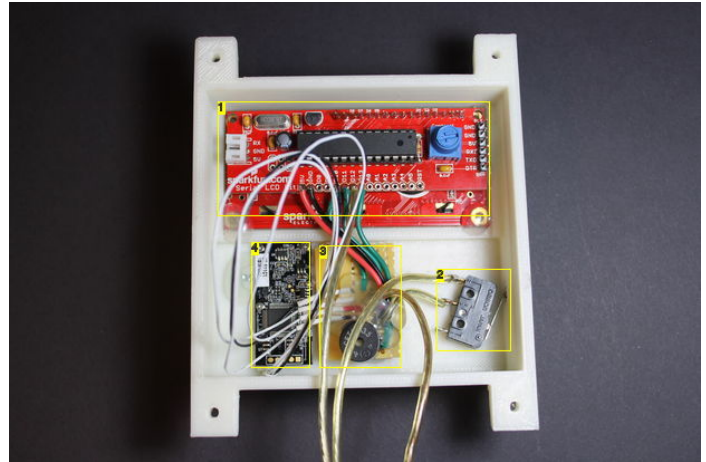
I am currently a sophomore in high school and a member of a FIRST robotics team. I love to tinker on my own, which is mostly inspired by the amazing people on this website.

## Intro: DIY Fingerprint Scanning Garage Door Opener

As a person without a car, I don't need to carry keys around everywhere I go. Because of this, I've been locked out of my own house several times. It's a pain to wait for someone with a key, so I thought I would do something about it.

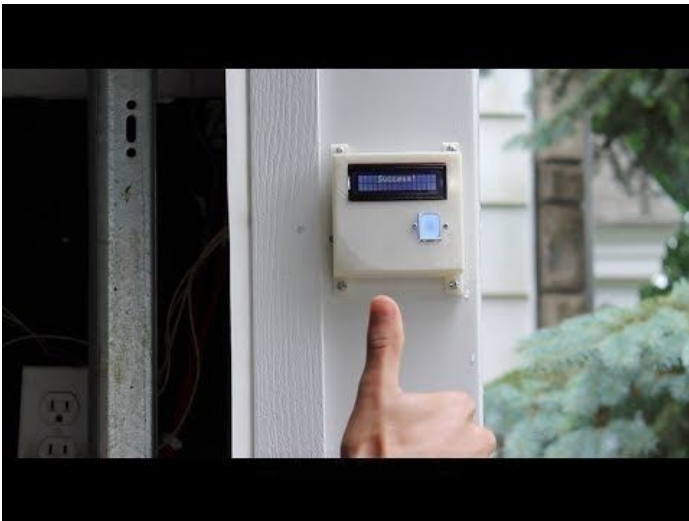
This project is my way of solving this problem, while getting the chance to interface with an awesome **fingerprint scanner** (aka: FPS).

Also, this module **isn't restricted to just garage doors**, for you can create different kinds of simple motorized locks to suit your needs.



### Image Notes

1. The serial LCD kit
2. Limit switch to detect when the case is closed
3. Circuit board with buzzer
4. Fingerprint scanner



## Step 1: Materials

### Electronics:

Part	Supplier
Fingerprint scanner (and JST connector)	
Serial LCD kit (w/ATmega328)	
ATtiny85	
PNP transistor	

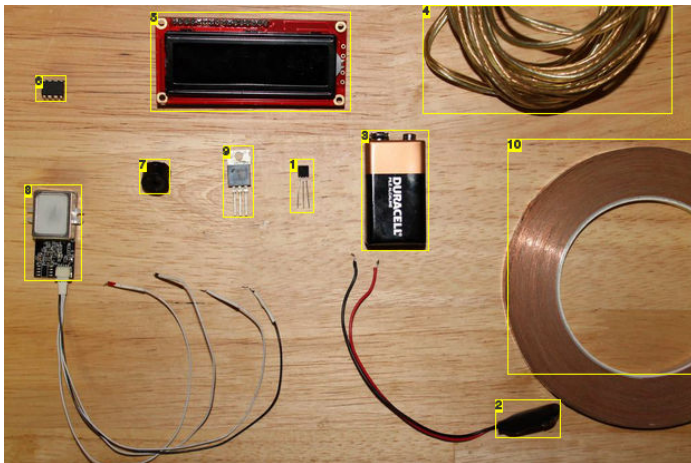
<http://www.instructables.com/id/DIY-Fingerprint-Scanning-Garage-Door-Opener/>

Buzzer	
Speaker wire	
3D printed case	See step 9 for files
Copper tape	
5V voltage regulator	
9V battery	
9V battery connector	
SPDT limit switch	

Here is a list of almost all of the parts (It's a Sparkfun wishlist).

#### Tools:

- Soldering iron/solder
- Electrical tape
- Hook up wire/ jumpers
- Wire cutter/stripper
- Perfboard
- Assorted resistors
- Screws
- Drill
- A few LEDs for testing
- 5V FTDI board ( )
- Hot glue gun
- Access to a 3D printer
- Optional: IC holder (8 pin for ATtiny and 28 pin for ATmega)
- Optional: Another Arduino board/10uF capacitor (see step 5 for details)



#### Image Notes

1. NPN transistor
2. 9V battery clip
3. 9V battery
4. Wire
5. Serial LCD (I've already assembled it)
6. Attiny85
7. Piezo buzzer
8. GT-511C1 fingerprint scanner
9. 7805 voltage regulator
10. Copper tape

## Step 2: The Circuit

The serial LCD kit sold by Sparkfun comes with an ATmega328 to control the LCD. The ATmega has extra processing power to be used for other tasks besides controlling the LCD. Because of this, we can use it as an Arduino to communicate with the fingerprint scanner, send an ATtiny85 commands, control the LCD, and use a buzzer to play tones.

To prevent the module from running continuously, I've added a limit switch to detect when the case is closed. If it's closed, power will not be supplied to it (saves battery power).

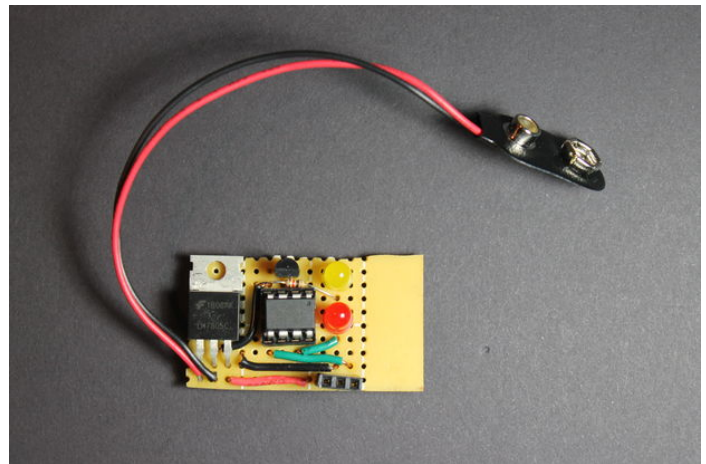
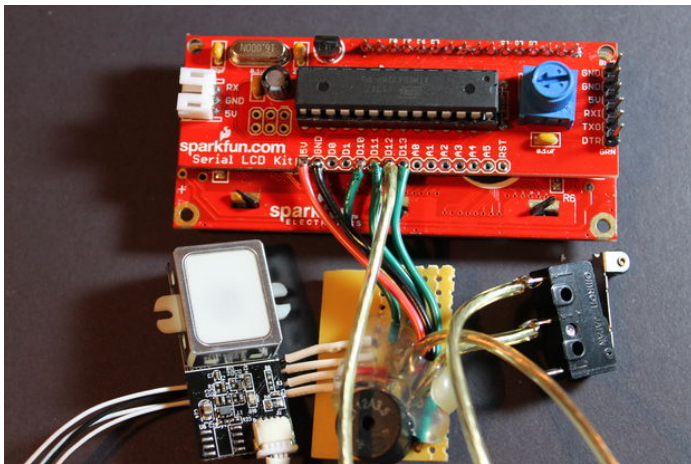
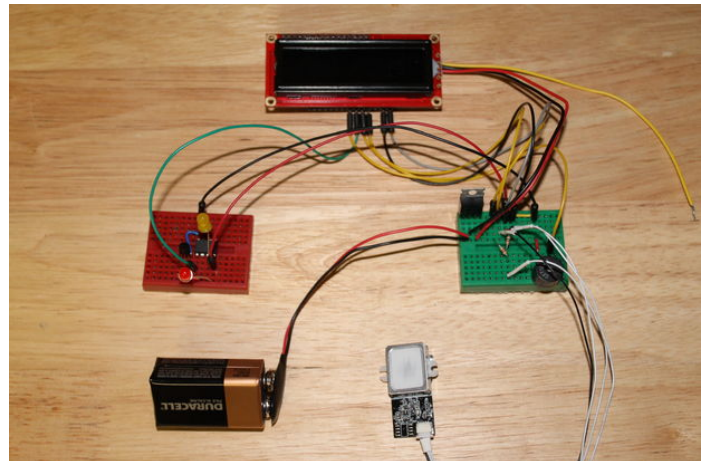
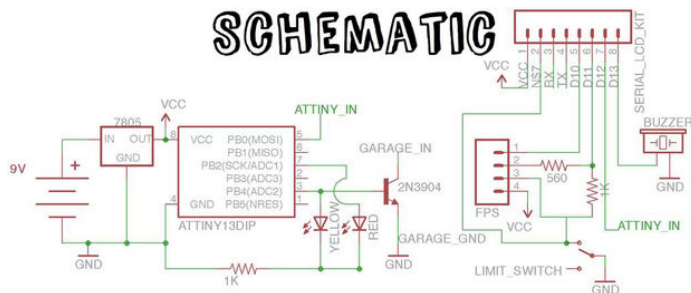
Important note: The fingerprint scanner communicates at a 3.3V level, so it is recommended to use a voltage divider to bring the signal from the ATmega to 3.2V. The voltage divider consists of a 560 $\Omega$  resistor between D10/FPS pin 2 and a 1K $\Omega$  resistor between GND/FPS pin 2.

Serial LCD Pinout:

D10	FPS pin 2 (through voltage divider)
D11	FPS pin 1 (black wire)
D12	ATtiny85
D13	Buzzer

ATtiny85 Pinout:

Pin 5 (0 in code)	Input from ATmega
Pin 3 (4 in code)	Transistor/yellow LED
Pin 7 (2 in code)	Indicator LED

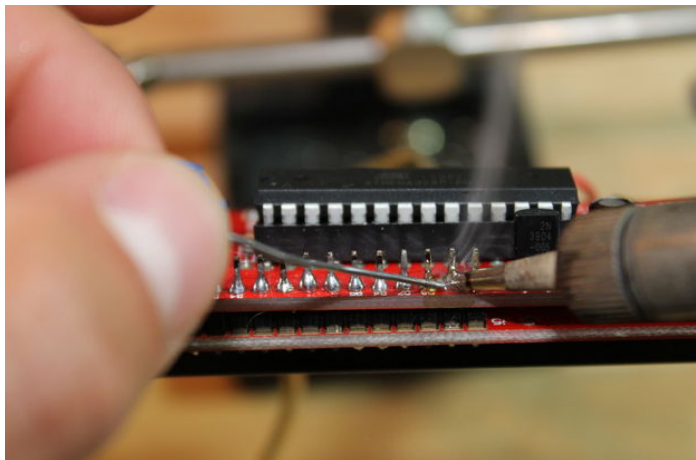




### Step 3: Assemble the Serial LCD Kit

Title says it all... This is a nice little kit to solder (I, personally, **love** to solder).

You can optionally solder a 28 pin IC holder to the board, which will allow you to take the ATmega out and use it again in another non-LCD project.

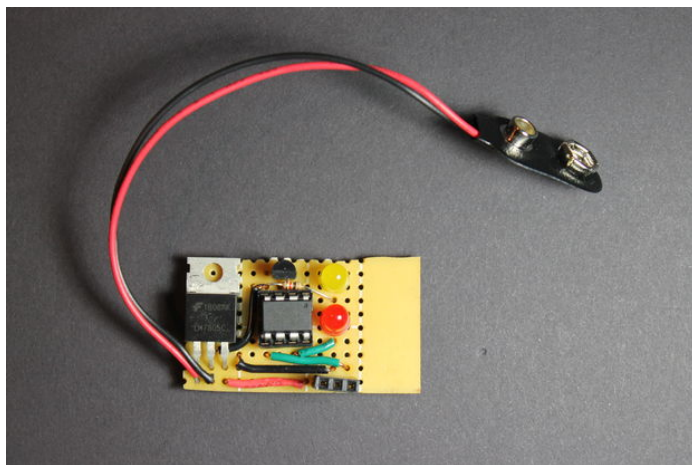
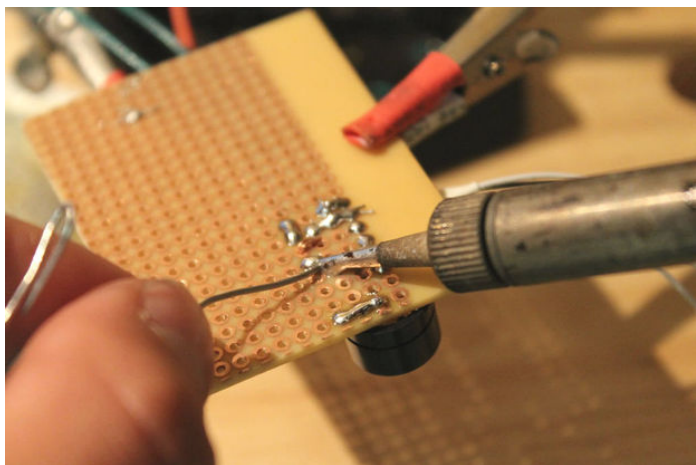
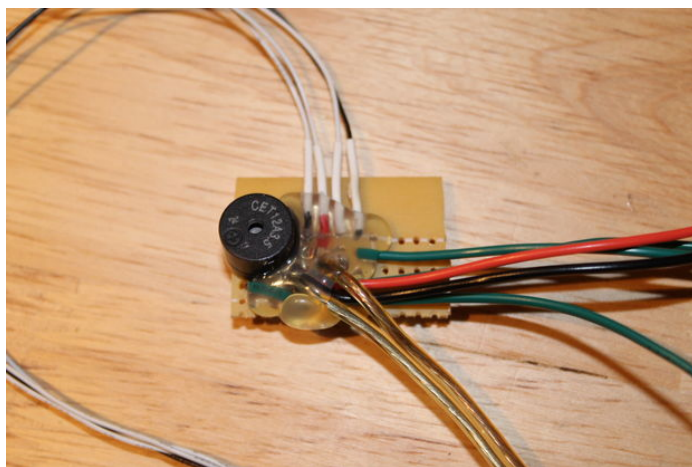
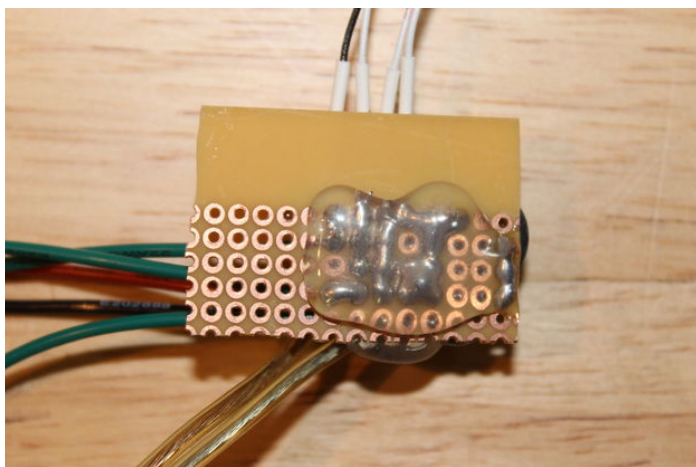


### Step 4: Assembling the Circuit Boards

The arrangement of the board is up to you, but remember to try to keep the FPS' wires facing the same direction so they don't break (they are really thin).

Next, I covered the the top and bottom with hot glue for both support and insulation. Using a high temperature hot glue is fine (nothing was burned/melted/ruined for me).

As with the main board, solder everything on the ATtiny's board together and optionally insulate/support it with hot glue. The voltage regulator might get a bit hot, so it would probably be a good idea not to let any hot glue get near it. You also might want to avoid covering the ATtiny in case you decide to take it out or reprogram it.



## Step 5: Programing the ATmega328

As mentioned in step 2, the ATmega328 has enough processing power and pins to drive the LCD while driving other things. To take advantage of this, you will need to have some way to program the chip.

If you own an Arduino Uno or Duemilanove, you can simply take off the chip already on the board and replace it with the one provided in the kit. Alternatively, you can use Sparkfun's FTDI Basic Breakout (5V) and solder headers to the side (see the pictures of step 3 for details).

Also, you need to upload the code as a "Duemilanove w/ ATmega328."

See below for an example sketch to make sure it is working.

### Code:

#### LCD Test:

```
//LCDTestExample by Nodcah
//A simple sketch to make sure your Serial LCD Kit from Sparkfun
//is working

#include "LiquidCrystal.h"

LiquidCrystal lcd(2,3,4,5,6,7,8);

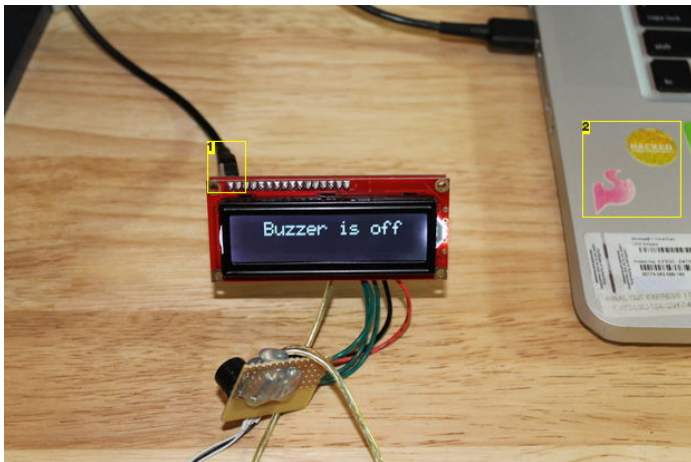
void setup() {
  pinMode(9, OUTPUT); //the backlight
  pinMode(13, OUTPUT); //the buzzer

  lcd.begin(16, 2); //16 chars wide, 2 tall

  digitalWrite(9, HIGH); //set the backlight to HIGH

  lcd.print(" Hello world! "); //use spaces to center the text
  delay(2000);
}

void loop() {
  //buzzer turns on and off and its status is displayed on the LCD
  lcd.clear();
  lcd.print(" Buzzer is on ");
  tone(13, 262, 1000);
  delay(1000);
  lcd.clear();
  lcd.print(" Buzzer is off ");
  delay(1000);
}
```



### Image Notes

1. FTDI basic breakout going to my computer
2. Sparkfun and Arduino!!

### File Downloads



**LCDTestExample.ino** (685 bytes)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'LCDTestExample.ino']

## Step 6: Setting Up the Fingerprint Scanner

For communicating with the FPS, I will use [this Arduino library](#) by Josh Hawley (direct download for the library [here](#)).

To make sure communication with your fingerprint scanner is working, I would upload [this blink example](#).

The fingerprint scanner has its own memory to store the fingerprint data. So, after you have verified the fps is working, upload [this example sketch](#) to add your fingerprint to the database under id #0. Open the serial console and simply follow the instructions.

### Code:

#### Blink Example:

```
/*
Library example for controlling the GT-511C3 Finger Print Scanner (FPS)
Created by Josh Hawley, July 23rd 2013
Licensed for non-commercial use, must include this license message
basically, Feel free to hack away at it, but just give me credit for my work =)
TLDR; Wil Wheaton's Law

This simple sketch turns the LED on and off similar to the Arduino blink sketch.
It is used to show that communications are working.
*/

#include "FPS_GT511C3.h"
#include "SoftwareSerial.h"

//Hardware setup - FPS connected to:
//digital pin 10(arduino rx, fps tx)
//digital pin 11(arduino tx - 560ohm resistor fps tx - 1000ohm resistor - ground)
//this brings the 5v tx line down to about 3.2v so we dont fry our fps

FPS_GT511C3 fps(10, 11);

void setup(){
  Serial.begin(9600);
  fps.UseSerialDebug = true; // so you can see the messages in the serial debug screen
  fps.Open();
}

void loop(){
  // FPS Blink LED Test
  fps.SetLED(true); // turn on the LED inside the fps
  delay(1000);
  fps.SetLED(false); // turn off the LED inside the fps
  delay(1000);
}
```

#### Enroll Example:

```
/*
FPS_Enroll.ino - Library example for controlling the GT-511C3 Finger Print Scanner (FPS)
Created by Josh Hawley, July 23rd 2013
Licensed for non-commercial use, must include this license message
basically, Feel free to hack away at it, but just give me credit for my work =)
TLDR; Wil Wheaton's Law
*/

#include "FPS_GT511C3.h"
#include "SoftwareSerial.h"

//Hardware setup - FPS connected to:
//digital pin 10(arduino rx, fps tx)
//digital pin 11(arduino tx - 560ohm resistor fps tx - 1000ohm resistor - ground)
//this brings the 5v tx line down to about 3.2v so we dont fry our fps

FPS_GT511C3 fps(10, 11);

void setup(){
  Serial.begin(9600);
  delay(100);
  fps.Open();
  fps.SetLED(true);
  Enroll();
}

void Enroll(){
  // Enroll test
  // find open enroll id
  int enrollid = 0;
  fps.EnrollStart(enrollid);

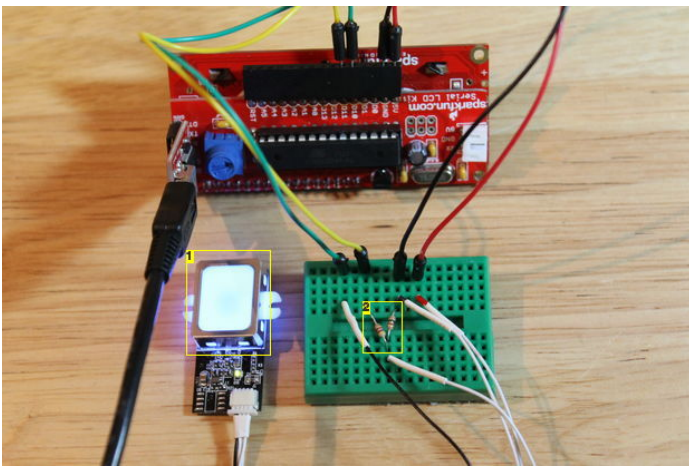
  // enroll
  Serial.print("Press finger to Enroll #");
  Serial.println(enrollid);
  while(fps.IsPressFinger() == false) delay(100);
  bool bret = fps.CaptureFinger(true);
  int iret = 0;
  if (bret != false)
  {
    Serial.println("Remove finger");
    fps.Enroll1();
    while(fps.IsPressFinger() == true) delay(100);
    Serial.println("Press same finger again");
    while(fps.IsPressFinger() == false) delay(100);
    bret = fps.CaptureFinger(true);
    if (bret != false)
    {
      Serial.println("Remove finger");
    }
  }
}
```

```

fps.Enroll2();
while(fps.IsPressFinger() == true) delay(100);
Serial.println("Press same finger yet again");
while(fps.IsPressFinger() == false) delay(100);
bret = fps.CaptureFinger(true);
if (bret != false)
{
  Serial.println("Remove finger");
  iret = fps.Enroll3();
  if (iret == 0)
  {
    Serial.println("Enrolling Successfull");
  }
  else
  {
    Serial.print("Enrolling Failed with error code:");
    Serial.println(iret);
  }
}
else Serial.println("Failed to capture third finger");
}
else Serial.println("Failed to capture second finger");
}
else Serial.println("Failed to capture first finger");
}

void loop(){
  delay(100000);
}

```



#### Image Notes

1. It is running the blink sketch right now
2. The voltage divider

#### File Downloads



**FPSBlinkExample.ino** (1 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'FPSBlinkExample.ino']



**FPSEnrollExample.ino** (2 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'FPSEnrollExample.ino']

### Step 7: Programing the ATtiny85

The ATtiny85 is basically a cheap and small Arduino condensed onto one chip (aka: one of the best things ever)! It can be programmed with another Arduino, including the ATmega328 in the serial LCD kit.

In this project, it will be used to execute very simple commands: check for a signal from the ATmega and open the garage door if the signal is legitimate.

To program it, connect it as seen in the picture above. Then, download all of the required files and follow the instructions by High-Low Tech.

After uploading this code, pin 13 on the Arduino (build-in LED) should be set to HIGH to signify that the code is working.

#### Code:

##### Final Code:

```

//fpsAttiny by Nodcah
//Recieves a brief signal from the main module to close a relay

void setup(){
  pinMode(2,OUTPUT); //indicator led through 10K resistor
  pinMode(4,OUTPUT); //trasistor pin that opens the garage
  pinMode(0,INPUT); //input
  digitalWrite(2, HIGH); //indicator LED
}

```

<http://www.instructables.com/id/DIY-Fingerprint-Scanning-Garage-Door-Opener/>

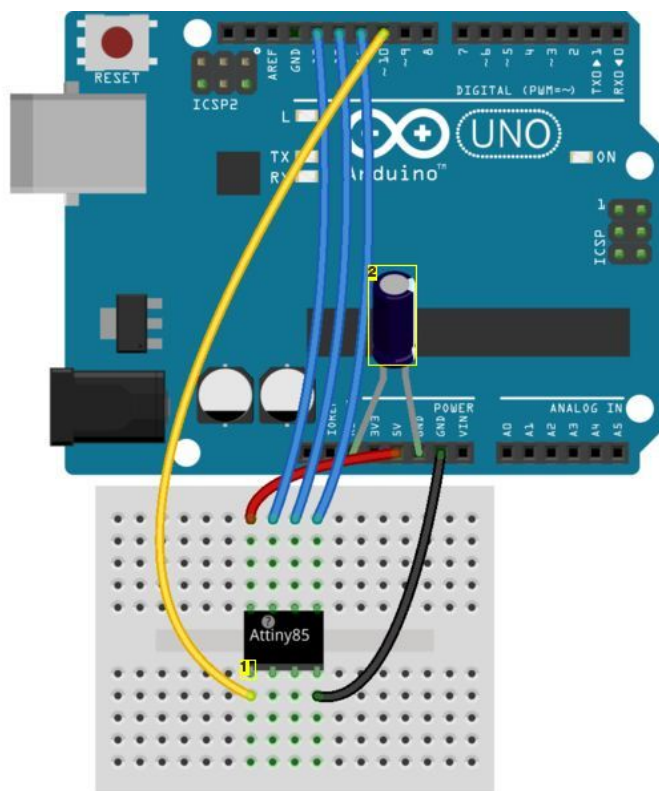


```

void loop(){

  if(digitalRead(0)){ //simple pattern that will be sent to trigger...
    delay(18);
    if(digitalRead(0)==false){
      delay(9);
      if(digitalRead(0)){
        delay(20);
        if(digitalRead(0)==false){
          digitalWrite(4, HIGH); //the transistor to "press" the button
          delay(1000);
          digitalWrite(4,LOW);
          digitalWrite(2,LOW);
          delay(1000);
          digitalWrite(2, HIGH);
        }
      }
    }
  }
}

```



Made with Fritzing.org

#### Image Notes

1. Pin 1 on ATtiny85
2. 10µF capacitor

## File Downloads



**FPSAttiny.ino** (767 bytes)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'FPSAttiny.ino']

### Step 8: The Final Code

Below is an Arduino program I have written for this project using the FPS and LCD libraries. I've done my best to write comments in code to describe what each part does, but if you have any questions, feel free to ask me!

After this code is uploaded, everything should be working. Now all that needs to be done it to integrate it!

#### Code:

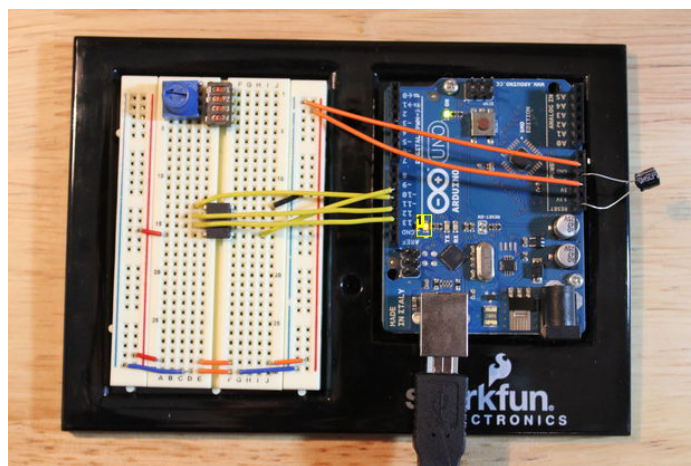
##### Code for ATmega238:

```

/*FPSGarageDoorOpener by Nodcah
*
*FPS_GT511C3 library created by Josh Hawley, July 23rd 2013
*Licensed for non-commercial use, must include this license message
*basically, Feel free to hack away at it, but just give me credit for my work =)
*TLDR: Wil Wheaton's Law
*
*Opens a garage door if the scanned fingerprint is in
*the fps' database of prints.

```

<http://www.instructables.com/id/DIY-Fingerprint-Scanning-Garage-Door-Opener/>



#### Image Notes

1. LED on pin 13 is on means it's working

```

*/
#include "LiquidCrystal.h" //for the screen
#include "FPS_GT511C3.h" //the fps (fingerprint scanner) library
#include "SoftwareSerial.h" //used by fps library

//Setting up the pins for the LCD and the fps
LiquidCrystal lcd(2, 3, 4, 5, 6, 7, 8); //pinouts for LCD
FPS_GT511C3 fps(10, 11); //RX, TX

boolean isFinger = false; //true if the fps detects a finger on the scanner
int timer = 0; //this is for when there is too long of a delay, it turns off

//output pins
const int buzzerPin = 13;
const int backlightPin = 9;
const int attinyPin = 12;

void setup(){
  //set outputs
  pinMode(buzzerPin, OUTPUT);
  pinMode(backlightPin, OUTPUT);
  pinMode(attinyPin, OUTPUT);

  //for debugging
  //Serial.begin(9600);
  fps.UseSerialDebug = false; //set to true for fps debugging through serial

  //initializing the libraries
  lcd.begin(16,2);
  digitalWrite(backlightPin, HIGH); //the LCD backlight
  fps.Open();
  fps.SetLED(true); //the fps LED

  //boot up sound
  for(int i=0; i<30; i++){
    tone(buzzerPin, 50+10*i, 30);
    delay(30);
  }
  tone(buzzerPin, 350);

  //print starting message
  lcd.print("Put your finger "); //the command to print to the LCD
  lcd.setCursor(0, 1); //sets the cursor to the 0th column in the 1st row
  lcd.print(" on the scanner ");
  delay(150);
  noTone(buzzerPin); //stops the startup sound
}

void loop(){
  //scan and identify the finger when one is put on it
  waitForFinger();
  timer = 0; //resets timer for timeout
  lcd.clear(); //clears the screen and sets the cursor to 0,0
  fps.CaptureFinger(false); //captures the finger for identification
  int id = fps.IdentifyL_N(); //identifies print and stores the id
  if(id < 20){
    lcd.print(" Access granted "); //success message
    lcd.setCursor(0,1);

    //one line personalized messages
    //messages can't be more than 16 chars!!!
    switch(id){
      case 0:
        lcd.print("    Hi self!    ");
        break;
      case 1:
        lcd.print("    Hey Bro!    ");
        break;
      case 2:
        lcd.print("Wat up homeslice");
        break;
      case 3:
        lcd.print("    Hi Mom!    ");
        break;
      case 4:
        lcd.print("    Hi person 4    ");
        break;
      case 5:
        lcd.print("    Hi Auntie!    ");
        break;
      case 6:
        lcd.print("    Hi person 6    ");
        break;
    }
    tone(buzzerPin, 262, 1000);
    delay(1500);

    //sends simple pattern to attiny inside garage (for security)
    digitalWrite(attinyPin, HIGH);
    delay(10);
    digitalWrite(attinyPin, LOW);
    delay(10);
    digitalWrite(attinyPin, HIGH);
    delay(20);
    digitalWrite(attinyPin, LOW);
    delay(2000);

    lcd.clear();
    lcd.print("Don't forget to ");
  }
}

```

```

lcd.setCursor(0,1);
lcd.print(" shut me off! ");
delay(2000);

waitForFinger(); //tap to continue to enroll

while(true){
  //save a new fingerprint

  //prints message to lcd
  lcd.clear();
  lcd.print(" So you want to ");
  lcd.setCursor(0,1);
  lcd.print("scan a new one? ");
  delay(2000);

  //Copied and slightly modified from the enroll example:
  int enrollid;

  //choosing which id to overwrite/create
  //release your finger when you want to write to the id printed on the screen

  waitForFinger(); //waits for the fps to be pressed
  if(fps.IsPressFinger() == true){
    lcd.clear();
    lcd.print("      1?      ");
    delay(1000);
    if(fps.IsPressFinger() == true){
      lcd.clear();
      lcd.print("      2?      ");
      delay(1000);
      if(fps.IsPressFinger() == true){
        lcd.clear();
        lcd.print("      3?      ");
        delay(1000);
        if(fps.IsPressFinger() == true){
          lcd.clear();
          lcd.print("      4?      ");
          delay(1000);
          if(fps.IsPressFinger() == true){
            lcd.clear();
            lcd.print("      5?      ");
            delay(1000);
            if(fps.IsPressFinger() == true){
              lcd.clear();
              lcd.print("      6?      ");
              delay(1000);
              if(fps.IsPressFinger() == true){
                lcd.clear();
                lcd.print("      7?      ");
                delay(1000);
                if(fps.IsPressFinger() == true){
                  lcd.clear();
                  lcd.print("      8?      ");
                  delay(1000);
                  if(fps.IsPressFinger() == true){
                    lcd.clear();
                    lcd.print("      9?      ");
                    delay(1000);
                    if(fps.IsPressFinger() == true){
                      lcd.clear();
                      lcd.print("     10?      ");
                      delay(1000);
                      enrollid = 10; //can be expanded to up to 20
                    }
                    else enrollid = 9;
                  }
                  else enrollid = 8;
                }
                else enrollid = 7;
              }
              else enrollid = 6;
            }
            else enrollid = 5;
          }
          else enrollid = 4;
        }
        else enrollid = 3;
      }
      else enrollid = 2;
    }
    else enrollid = 1;
  }

  //warning if there is already data in this id slot
  if(fps.CheckEnrolled(enrollid)){
    lcd.clear();
    lcd.print(" Warning! ID #");
    lcd.print(enrollid);
    lcd.setCursor(0,1);
    lcd.print(" has data. OK? ");

    waitForFinger(); //waits for the fps to be pressed

    fps.DeleteID(enrollid); //delete data
    delay(100);
  }

  //Enroll

```

```

fps.EnrollStart(enrollid);
lcd.clear();
lcd.print("Place finger to ");
lcd.setCursor(0,1);
lcd.print("enroll #");
lcd.print(enrollid); //prints id that is being enrolled

    waitForFinger(); //waits for the fps to be pressed

//captures the finger and saves to memory three times for accurate data
bool bret = fps.CaptureFinger(true); //high quality pic for enrollment
int iret = 0; //error stuff

if (bret != false){ //first enroll
    lcd.clear();
    lcd.print(" Remove finger ");
    fps.Enroll1();
    while(fps.IsPressFinger() == true) delay(100); //waits until no finger
    delay(100);
    lcd.clear();
    lcd.print(" Press again ");
    waitForFinger(); //waits for the fps to be pressed
    bret = fps.CaptureFinger(true);

    if (bret != false){ //second enroll
        lcd.clear();
        lcd.print(" Remove finger ");
        fps.Enroll2();
        while(fps.IsPressFinger() == true) delay(100);
        delay(100);
        lcd.clear();
        lcd.print("Press yet again ");
        waitForFinger();
        bret = fps.CaptureFinger(true);

        if (bret != false){ //third enroll
            lcd.clear();
            lcd.print(" Remove finger ");
            while(fps.IsPressFinger() == true) delay(100);
            iret = fps.Enroll3();
            if (iret == 0){ //checks to see if there are any errors
                lcd.clear();
                lcd.print(" Success! ");
                delay(2000);
                beep(); //shuts arduino off
            }
            else{ //if the enrollment fails in any way
                lcd.clear();
                lcd.print("Fail. Try again ");
                delay(1000);
            }
        }
        lcd.clear();
        lcd.print(" Failed 3rd "); //error on 3rd
        delay(1000);
    }
    lcd.clear();
    lcd.print(" Failed 2nd "); //error on 2nd
    delay(1000);
}
lcd.clear();
lcd.print(" Failed 1st "); //error on 1st
delay(1000);
}
}

else{
    lcd.print("Fingerprint is"); //if print isn't recognized
    lcd.setCursor(0,1);
    lcd.print(" unverified ");
    delay(2000);
    lcd.clear();
    lcd.print("Please try again");
    lcd.setCursor(0,1);
    lcd.print("Use your pointer"); //I scanned everyone's pointer finger
    delay(500);
}
delay(250);
}

void beep(){
    //beeps in hopes of someone closing the case
    lcd.clear();
    lcd.print("Please close the");
    lcd.setCursor(0,1);
    lcd.print(" case! ");
    for(int i=0;i<10;i++){
        tone(buzzerPin, 262, 500);
        delay(1000);
    }
    delay(10000); //wait for someone to close the case
    //if no one does, shut everything off
    lcd.clear();
    digitalWrite(backlightPin, LOW);
    fps.SetLED(LOW);
    while(true) delay(10000);
}

```



```

void waitForFinger(){
  timer = 0; //resets the timer everytime this function starts
  while(fps.IsPressFinger() == false){ //timeout
    timer++;
    delay(100);
    if (timer>=80){
      timer = 0; //reset timer
      break;
    }
  }
  if(fps.IsPressFinger() == false)beep();
  timer = 0; //resets the timer everytime this function ends
}

```

## File Downloads



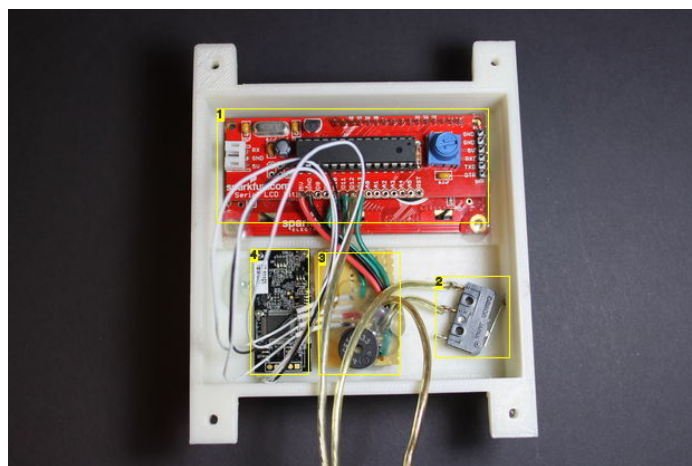
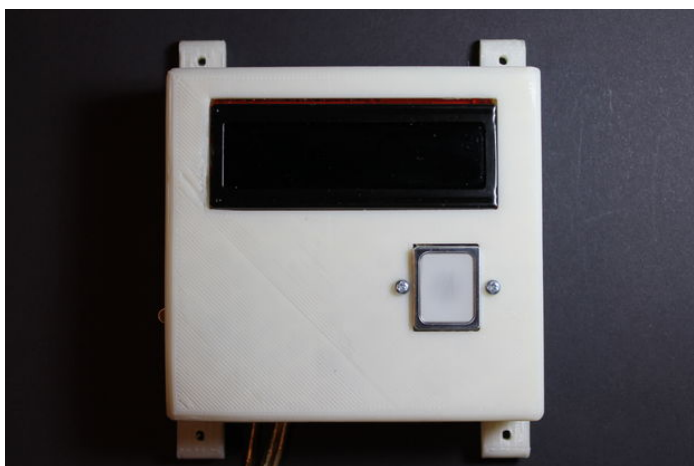
**FPSGarageDoorOpener.zip** (4 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'FPSGarageDoorOpener.zip']

### Step 9: The 3D Printed Case

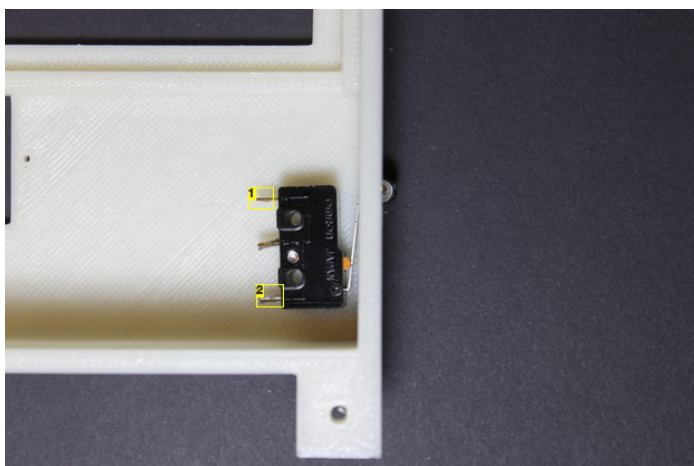
To turn on the module, the case will need to be slid up, triggering the limit switch. As shown by the pictures, the limit switch needs to be wired to the common terminal (C), and the normally closed (NC) terminal.

Then, everything is glued to the case with hot glue. The limit switch is positioned with a slight tilt to make it easier to press.



#### Image Notes

1. The serial LCD kit
2. Limit switch to detect when the case is closed
3. Circuit board with buzzer
4. Fingerprint scanner



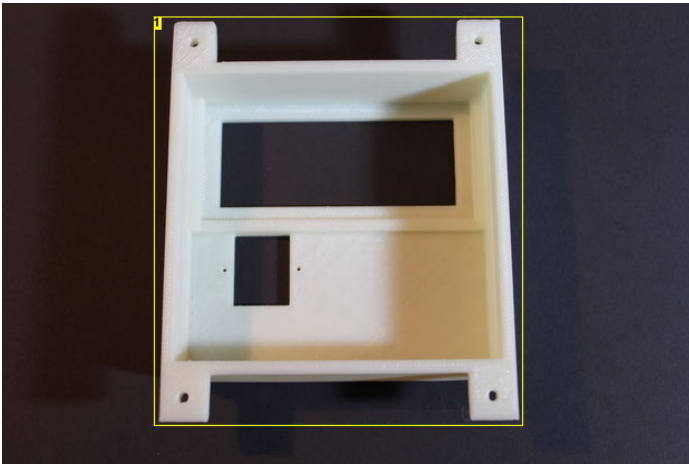
#### Image Notes

1. Normally closed (NC)
2. Common (C)



#### Image Notes

1. Spaces for the main case's supports



#### Image Notes

1. 3D printed main case

#### File Downloads



**FPSMainCase.STL** (127 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'FPSMainCase.STL']



**FPSCover.STL** (71 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'FPSCover.STL']

#### Step 10: Prepare the Garage

To open the garage door I wired my module to the button that normally opens the garage. Instead of a physical connection being made, the module uses a NPN transistor to "press" the button.

The wires should first be measured and cut to size, leaving a little extra wire just to be safe. Then, the hard part: soldering the wires from the button to the FPS module (shown in the pictures as an animated GIF). The wires should next be wrapped with a generous amount of tape.

To get the signal from the ATmega outside of the garage to the ATtiny inside the garage, three wires (power, ground and signal) will need to be fed through the wall. On my garage, there was a piece of wood that I just drilled right through (see the pictures).

Finally, screw on the case and boot it up!



#### Image Notes

1. The button that came with the garage
2. Where the solder connection is made
3. The destination



#### Image Notes

1. I remembered safety glasses this time!
2. My super handy gas-powered soldering iron



#### Image Notes

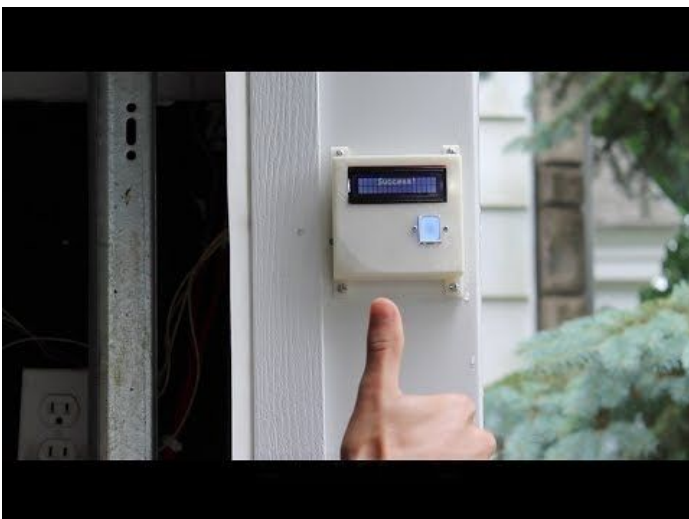
1. 3/8" hole

### Step 11: Testing!

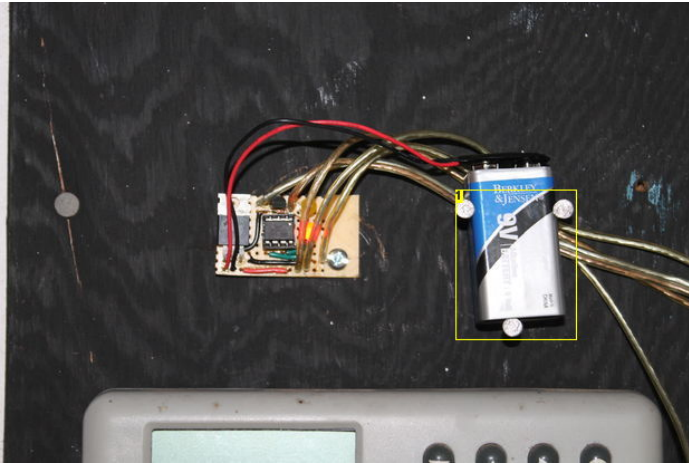
Now is the fun part! Use the module's built-in enroll feature so family/friends can open the garage. Then, create personalized messages for each one! Watch the video for a visual explanation of functionality.

If you've made it this far, please consider voting for me in the sensors contest. Every one of the prizes will help me to continue my love for learning electronics. Also, isn't a fingerprint scanner an awesome sensor?

Thanks! â







**Image Notes**  
 1. A simple 9V battery holder using three nails

**Related Instructables**



**Arduino LCD Thermometer / temperature control with TMP36GZ Temp sensor added 2 chanel relay, added RTC ds1307 , added a case.based on WWC Guide .** by citin



**Arduino LCD Thermometer with LM35 Temp Sensor** by WWC



**Pocket Distance Scanner** by Baybren



**LCD display of temperature in C or F by choosing the type with an IR remote** by GreyHat231



**Salvaging Liquid Crystal Displays (LCDs)** by Josehf Murchison



**Digital Thermometer With Arduino & DS18B20** by Rahul Mitra

**Comments**