

A Tinkerer's Ramblings

Detailing technical projects using micro controllers, networking, c#, SQL, and other technologies to automate every day life.

Finger Print Scanner (GT-511C3) Library

Posted on [August 8, 2013](#)



12



I got a [GT-511C3 finger print scanner](#) Sparkfun as part of a project for controlling my front door lock. I had quite a bit of trouble making it work. below is a list of issues, and how I got through them:

1. It uses 3.3v logic instead of the Arduino's 5v. I used a voltage divider to keep the Arduino from damaging the board. Arduino tx – R1(560ohm) – FPS rx – R2(1kohm) – ground (see diagram)
2. The demo posted by Sparkfun shows it running at 115200, but the device defaults to 9600 baud. It is listed on the Sparkfun site as defaulting to 9600, but contradicted in their video. In any case, the device always boots at 9600 baud, and then you can change the baud rate from there.
3. The JST SH Jumper 4 Wire Assembly that you can buy from Sparkfun is not very breadboard friendly. This was annoying but not too hard to fix with a little soldering. In the video, he uses a much different connector. I looked around but could not find that one for sale on Sparkfun's site.
4. There is no library to support this device. I have spent my spare time in last few weeks working on a library to support this device. All of the basic functions are done and tested. The more advanced functions (like uploading/downloading templates/images/etc) are not implemented yet because i'm still working on how to deal with large amounts of data on a device that has 2k of ram.

So without further ado... Here is my library:

To use it, first download and install it in the Arduino IDE as you would any other library. There are lots of examples of this on the internet, so I won't go into that here.

There are a few objects in it, but the only one you really need to use is FPS_GT511C3. To create the object, declare it, and send it the RX and TX pins as parameters (don't forget the voltage divider or

logic level shifter:

```
FPS_GT511C3 fps(2, 3);
```

Inside setup you need to call open to initialize communications with the device and (optionally) turn on the serial debugging messages:

```
fps.UseSerialDebug = true;  
fps.Open();
```

One of the simplest things to do is to turn on the LED that is inside the device. It needs to be on to see the fingerprints anyways. I typically do this during setup so that all of the looping commands can see.

```
fps.SetLED(true);
```

Now that our FPS can see, we should check to see if there is a finger on it.

```
bool fingerpressed = fps.IsPressFinger();
```

Now that we know there is something to look at, lets capture an image of it. Note that the image is captured to the onboard memory for use with subsequent commands, and not returned to the Arduino. The boolean parameter is used to enable or disable high resolution reading. In general you use high resolution when enrolling (adding finger print ids), and low resolution when doing everyone else because of speed.

```
fps.CaptureFinger(false);
```

Lets check to see if that image is in the database on the fps. If it is, this command will return the ID number of it (a number between 0 and 199).

```
int id = fps.Identify1_N();
```

Enrolling fingerprints is a little more complex. You have to do the following steps below in order, and check for errors after each one.

1. Check for an ID slot that is unused until you find one: `fps.CheckEnrolled(enrollid);`
2. Start and enrollment with that ID: `fps.EnrollStart(enrollid);`
3. Wait for a finger press: `while(fps.IsPressFinger() == false) delay(100);`
4. Capture an image of that finger in high resolution: `fps.CaptureFinger(true);`
5. Call Enroll1 to use the image in memory: `fps.Enroll1();`
6. Wait for finger to be removed: `while(fps.IsPressFinger() == true) delay(100);`
7. Wait for the same finger press: `while(fps.IsPressFinger() == false) delay(100);`
8. Capture an image of that finger in high resolution: `fps.CaptureFinger(true);`

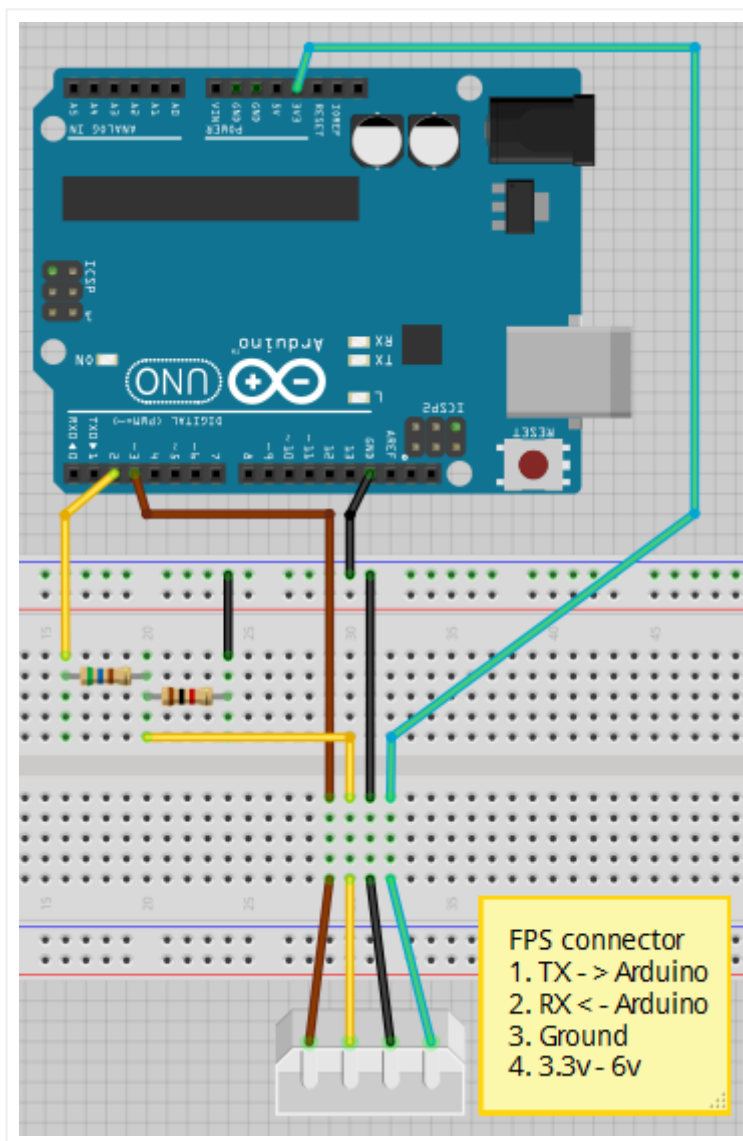
9. Call Enroll2 to use the image in memory: `fps.Enroll2();`
10. Wait for finger to be removed: `while(fps.IsPressFinger() == true) delay(100);`
11. Wait for the same finger press: `while(fps.IsPressFinger() == false) delay(100);`
12. Capture an image of that finger in high resolution: `fps.CaptureFinger(true);`
13. Call Enroll3 to use the image in memory: `fps.Enroll3();`
14. If the return value of Enroll3 is zero, the enrollment was successful

If all of this seems like too much, don't worry. I have included a few examples with the library. A basic blink sketch (blinks the LED on the FPS), an enrollment sketch, and an identification sketch.

You can download the library from me here: [FPS_GT511C3.Zip](#)

It is a work in progress, so if you have comments/suggestions please post them below =)

Here is a basic connection diagram that I made in Fritzing. I could not find the correct connector so I used a 4 pin connector that I found. It is not right, but illustrates the connections well enough. Do double check where pin one goes when connecting yours.



- This is how to connect the FPS (GT-511C3) to an Arduino using a voltage divider. Make sure that you double check where pin 1 is in your connections (the graphic for the connector may be misleading). Keep in mind that the logic level converters do a much better job of this. This configuration will likely not function properly at high speeds. It does work properly at the default of 9600 baud.

Update: Someone (sorry I can't read that language) used my library to make a finger print door lock. They added a LCD Panel, and a motion sensor as well for a great solution. The motion sensor turns on the LCD and FPS when someone approaches. That is a great way to keep from having them on all the time. http://lazygyu.net/blog/fingerprint_doorlock

Update 2: nodcah on Instructables made a garage door opener with my library too. They used an arduino to signal an attiny inside the garage, so it is nearly hackproof

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



You may also like:

SOCKET
SOCKET
SOCKET

Sockets Sockets
Sockets

FWL2013 Update 5
(6/20/2013)

FWL2013 Update 4
(6/19/2013)

FWL2013 Update 2
(6/10/2013)

FWL2013
UPDATE
(6/23/2013)

Really high(like from
space) level overview
of ASH

Webservices and iOS
Test app working

FWL2013 Update 3
(6/17/2013)

FWL2013 Update 6
(6/23/2013)

This entry was posted in [Arduino](#), [Door Control](#), [Smart Home](#) by [jhawley](#). Bookmark the [permalink \[http://wordpress.hawleyhosting.com/ramblings/?p=375\]](http://wordpress.hawleyhosting.com/ramblings/?p=375).

144 THOUGHTS ON "FINGER PRINT SCANNER (GT-511C3) LIBRARY"



Nicolas Dupoty

on **September 26, 2015 at 5:18 am** said:

Hi,

How can i check is the FPS is damaged or not ?

I forgot to add resistors to rx/tx :/



jhawley

on **October 28, 2015 at 4:31 pm** said:

Hook it up to a FTDI cable, and use the demo app.



Guido B

on **September 10, 2015 at 12:02 pm** said:

Hi, would it be possible to send the data from a fingerprint to a database on a pc via wifi, and then, when trying to identify the fingerprint, get that data from the pc database and do the identifying process?



jhawley

on **September 21, 2015 at 11:50 am** said:

You can get raw images from the scanner out to a pc to do the comparison, but you would have to write the comparison software yourself. reading the templates and images is not something that I was able to implement anyway due to serial IO and memory constraints. (see other comments here for an explanation). It might be possible with a non UNO controller, but I have not experimented with it.



Ricardo Marciano

on **February 21, 2015 at 4:00 pm** said:

Hello Hawley,

Your library is compatible with arduino mega?



jhawley

 on [February 27, 2015 at 10:54 am](#) said:

I have never tried it on a mega, but it will most likely work. you may need to try different pins than what are listed above.



Daniel Klotz
on [March 2, 2015 at 6:37 am](#) said:

Hey Josh,

First, thanks for the library.

Second: This thing is driving me nuts.

I have an example code from here

<http://www.homautomation.org/2014/10/11/playing-with-finger-print-scanner-fps-on-arduino/>

For Enrollment and whatnot.

I can get the Blink to work, that's it. When I use the Enrollment Sketch and open the Serial Monitor after uploading, I only get

FPS – Open

FPS – SEND: "55 AA 01 00 00 00 00 00 01 00 01 01"

And that's it. It appears that the FPS isn't starting the enroll as it's told to.

Any idea what I could do?

Your help is greatly appreciated!

~ D



jhawley
on [May 19, 2015 at 11:40 am](#) said:

Sorry for the slow response.

The OPEN command isn't getting a response. You should be getting another debug line that starts with "FPS – RECV". If you are not seeing any of those, then there is a problem with the communications. Are you telling it to use the same pins in both sketches?



Dhruvajyoti Singha

on **May 31, 2015 at 10:04 am** said:

Sir I am facing this problem too, so can you please tell me how to debug ts problem. I mean how to proceed to solve this problem.

Here is my email ID: dhruvajyoti17@gmail.com

Thank you.



Ted Chavalas

on **September 8, 2015 at 6:03 pm** said:

So... I have confirmed that using an Arduino Mega... I cannot get the GT-511C3 to talk back to the Arduino Mega with your library. Does not seem to matter which pins I assign using the FPS_GT511C3 fps(2, 3) command.

I have tried many different pin assignments. It will not receive data.

I have confirmed the GT511C3 is getting commands from the Arduino Mega. For example I can turn the finger reader LED on using the fps.SetLED(true) command and it light up. But then the program hangs because the Mega is not get ing any response back.

I think you need to try your library on a Mega board. I think you will find it does not work as expected.



Ted Chavalas

on **September 8, 2015 at 6:15 pm** said:

I just figured it out. On the Mega you cannot use pins 4 and 5 for the SoftwareSerial because they don't support the PinChange interrupt. As the RX pin you have to use one of the following: 10, 11, 12, 13, 14, 15, 50, 51, 52, 53, A8 (62), A9 (63), A10 (64), A11 (65), A12 (66), A13 (67), A14 (68) or A15 (69).

On the Mega it would be much more reliable to use one of the hardware serial interface instead of the clunky software emulation. But the library would have to change to take advantage of the Mega's UART support for the added serial ports.

For now I have it working. I am using A8 and A9 and it seems to be working OK.



jhawley

on **September 21, 2015 at 11:54 am** said:

Good job sorting that out!

You are right that the library would have to change. I built it with the minimum amount of references to SoftwareSerial so that it can be changed out pretty easily for that very reason. I figured that other serial libraries would be used for other controllers.



Dhruvajyoti Singha

on **June 4, 2015 at 10:44 pm** said:

i am having the same problem. How did you solved it? cane you please explain it to me



Ted Chavalas

on **September 8, 2015 at 5:08 pm** said:

I have been trying to get it running on an Arduino Mega as well with no luck. Your Library seems to work on an UNO but it does not seem to be receiving data. So it hangs after the send.



KamaRay

on **June 23, 2014 at 10:52 pm** said:

Dear Mr. Hawley,

In an ambitious attempt at a first arduous (something other than making a laser module blink, etc.) Arduino project, I purchased the Sparkfun's ADH-Tech FPS (C1). With my handy bread board, I linked it up to the Arduino board, following your precise diagram to the point. (I checked multiple times). Key difference is that I'm using a 330r, 220r, and 10r in series in place of the 560r, but does that make much of a difference? Now, here's where a problem arises. I can see the green light on the FPS, but the blink sketch (or any other example programs done by you, for that matter. Of course, the pins 4/5 were edited to 2/3 respectively) doesn't seem to work. Upon sending ANY serial command to through the serial monitor, the led on the FPS will quickly flash on, then back off, even if it's not in the code. It also doesn't seem to progress though your code— it completely ignored serial.prints. Have you ever seen this before, or have any assumptions on how to fix it. Do I own a faulty device? Did I not supply enough information? (If so, sorry, I'm still a seasoned noob) Thanks in advance, and I greatly appreciate your help.

Sincerely,

Kamaray



KamaRay

on **June 23, 2014 at 10:54 pm** said:

To provide a few more details:

– The Board is an Uno.

... And that's all I believe.

Also, is it possible to edit a comment?

Thanks again! 😊



jhawley

on **July 26, 2014 at 1:46 am** said:

Sorry for the late reply.

If you are using a combination of resistors in series, double check their total resistance value by measuring them as a set with your multimeter.

While I have never tried a C1, as far as i have heard, they use the same command set. The issue you describe smells like a hookup issue. have you tested your voltage divider to make sure that it is only delivering 3.3 volts?



hullo2chuks

on **June 13, 2014 at 12:34 pm** said:

Please I really need your assistance, am using raspberry pi, so memory is not an issue, I just need you to give me a code sketch on how i can implement the get template(template extraction) a code snippet will really do. thanks in anticipation of your prompt reply



jhawley

on **June 13, 2014 at 6:32 pm** said:

What you are asking for is not simple. You would need to implement code to parse the data packet that comes over the serial line after the response packet. See section section 2 for DataPacket format, and 5.23 (GetTemplate) on the datasheet.

Datasheet: http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/Biometric/GT-511C3_datasheet_V1%201_201304114.pdf



lazygyu

on **June 10, 2014 at 2:29 pm** said:

Hi!

I used your library to made a fingerprint doorlock. It works great. The library was very helpful.

I found this post by your trackback. And I want to say thank you. It is very nice. It will be harder If without your library.

By the way, the language is Korean. 😊 And sorry for my poor English.

Have a nice day!



jhawley

on **June 10, 2014 at 2:36 pm** said:

I'm glad you found it useful. I'm impressed at how you integrated the motion sensor. That is a great idea.

"And sorry for my poor English."

Your English is better than my Korean. lol



aaronpbrooks

on **June 10, 2014 at 11:39 am** said:

Hi Josh. Great library – many thanks for putting it together. I'm working on the template extraction functionality and was hoping you might shed some light on how you were planning on approaching this. I'm running this on a Teensy3.1 (great hardware if you don't already have one), which alleviates some of the memory constraints, and I've modified your library slightly to use with hardware serial, as opposed to SoftSerial. In any case, I don't have a lot of experience with C++, but I'm slowly starting to piece together the code that you have commented out. If you'd be willing to provide some more detail on how you might implement this, I would be extremely appreciative (even more than I am already). Thanks!!



jhawley

on **June 10, 2014 at 1:46 pm** said:

There are basically only 2 methods. It must be read in a continuous stream from the FPS because it sends without flow control. That means that either you must read it all into memory at once, or stream it through your device to somewhere else (network or some storage device).



aaronpbrooks

on **June 11, 2014 at 4:55 am** said:

Thanks Josh. That's exactly how I'm approaching this (both methods, actually – capture the stream in its entirety, and then split it up into chunks of 20 bytes in length, to be sent to another device). I guess I'm wondering, more specifically, how you were planning on using the "Data_Packet" class that you have commented out in your library. You did leave helpful comments with everything, I'm just trying to understand the intent behind each of the variables and functions and how they were meant to be strung together. I have

something that basically functions, but it's ugly and it bypasses most of the pseudo-code you had.

I know "beggars can't be choosers," and it's probably a real pain for you to keep coming back to this code after a year, but thanks for looking out for all of us noobs!



jhawley

on **June 14, 2014 at 1:39 pm** said:

When you send a GetTemplate request to the FPS, it replies with a normal response packet, and then follows that up with a data packet containing the template. The data packet has its own format (specified in the data sheet). I had originally thought that the SoftwareSerial buffer would just hold the extra data for me until got the next chunk, and I was going to read 128 bytes at a time. the problem is that because it is not hardware serial, that buffer is in memory (which I already am short on), so it just overruns and the data is lost on the way in from the FPS.

If you have a bunch of memory, I would strip out the ID numbers from the Data_Packet object, and then parse the incoming packet into it. Make sure that you increase the size of your incoming buffers appropriately too(inside SoftwareSerial).



Imescobar

on **April 26, 2014 at 10:34 am** said:

Hi Mr. Howley.

How to adjust the library to pcDuino FPS_GT511C3, I've been trying and I have not succeeded. thanks



Imescobar

on **April 26, 2014 at 10:38 am** said:

The pcDuino gives problems SoftwareSerial library.



jhawley

on **June 10, 2014 at 1:47 pm** said:

I would try asking on the pcdduino or arduino forums for an updated or different library that you can swap out with SoftwareSerial.

Pingback: [Lazygyu](#)



gcriniti

on **March 29, 2014 at 7:08 am** said:

Hi, Josh! I've been stuck on a problem with the FPS for some time now. I hope you can help. I want to use an Uno to communicate between the PFS and a wireless network board (WiFly from SparkFun). I cannot get both devices to work at the same time. I've stripped down my code to isolate the problem by simply turning on an LED when wireless info is received. The following code works perfectly UNTIL I enable the line that opens the FPS; fps.Open(). Then there is no more communication. Thanks for your help!

/*

WiFly2Uno2fps

When data arrives from the WiFly module this sketch turns on one of three LED's.

It also send an acknowledgement message back through the WiFly.

Hardware setup – FPS connected to:

digital pin 4(arduino rx, fps tx)

digital pin 5(arduino tx – 560ohm resistor – fps tx – 1000ohm resistor – ground)

this voltage divider brings the 5v tx line down to about 3.2v so we dont fry our fps

*/

#include

#include "SoftwareSerial.h"

SoftwareSerial WiFlySerial(18, 19); // RX, TX

FPS_GT511C3 fps(4, 5);

// WiFlySerial is the connection to the WiFly module

// Serial is the connection to the USB port

int greenled = 10;

int yellowled = 11;

int redled = 12;

char inChar;

```
void setup()
{
  // initialize serial connections
  Serial.begin(9600);
  WiFlySerial.begin(9600);

  pinMode(greenled, OUTPUT);
  pinMode(yellowled, OUTPUT);
  pinMode(redled, OUTPUT);
  //fps.Open();
  //fps.SetLED(true);
  delay(1200);
  //fps.SetLED(false);

}
void loop()
{
  while (WiFlySerial.available()) {
    // get the new byte:
    inChar = (char)WiFlySerial.read();
    if (inChar == 'g') {
      WiFlySerial.println("Green LED \r");
      digitalWrite(greenled, HIGH);
      delay(1000);
      digitalWrite(greenled, LOW);
    }
    if (inChar == 'y') {
      WiFlySerial.print("Yellow LED \r");
      digitalWrite(yellowled, HIGH);
      delay(1000);
      digitalWrite(yellowled, LOW);
    }
    if (inChar == 'r') {
      WiFlySerial.print("Red LED \r");
      digitalWrite(redled, HIGH);
      delay(1000);
      digitalWrite(redled, LOW);
    }
  }
}
```



gcriniti

on **March 29, 2014 at 7:29 am** said:

oops, that first include line adds the fps library:

```
#include
```

I guess my copy/paste didn't work so well.



jhawley

on **June 10, 2014 at 1:55 pm** said:

This is a limitation in the SoftwareSerial library. It can only listen to one device at a time.

You need to call this method before you try to get communicate with each device when using more than one.

<http://arduino.cc/en/Reference/SoftwareSerialListen>

My library calls listen each time it communicates with the device already, so you just need to add a `WiFlySerial.listen();` right after each of your calls to the FPS.



wbix

on **March 22, 2014 at 5:14 am** said:

Hi, thanks so much for posting this library. I only have one problem. When I try to compile, I get this error:

```
FPS_GT511C3/FPS_GT511C3.cpp.o: In function `FPS_GT511C3::Open()':  
/home/will/sketchbook/libraries/FPS_GT511C3/FPS_GT511C3.cpp:237: undefined reference  
to `operator new[](unsigned int)'  
/home/will/sketchbook/libraries/FPS_GT511C3/FPS_GT511C3.cpp:244: undefined reference  
to `operator new[](unsigned int)'  
collect2: ld returned 1 exit status
```

Thanks so much for your help.



jhawley

on **June 10, 2014 at 2:00 pm** said:

This typically happens when the linker is not setup right (not including proper standard libraries), or some messed with the new operator (happens more than you might think in libraries).



tesinasm

on **March 19, 2014 at 5:26 am** said:

Hello, i bought Fingerprint Scanner – 5V TTL (GT-511C1)..

i'm using your libraries, but i found problem : every time i want to save a fingerprint, first of all we read the fingerprint but i'm not that sure that it saves it; and when i want to read the next fingerprint it always reads in the first position (position 0). How can save more fingerprints to read them in a different position?



jhawley

on **June 10, 2014 at 2:02 pm** said:

Have a look through the sample code. It takes several operations to save a fingerprint, one of which allows you to specify the id number of the slot to save it in.



calwal

on **March 9, 2014 at 8:37 am** said:

Friend sorry to bother you , but I can already register id, I can verify that the registered id agree but I am not knowing how to write the rows to delete .
I tell my program to give you a look and see what is wrong .
thank you very much .

Please, I need your help too.

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

```
// Hardware setup – FPS connected to:  
// Digital pin 2 ( Arduino rx , tx fps )  
// Digital pin 3 ( arduino tx – 560ohm resistor – fps tx – 1000ohm resistor – ground )  
// This voltage divider brings the 5v tx line down to about 3.2v so we dont fry our fps
```

```
LiquidCrystal lcd ( 8,9,4,5,6,7 ) ;  
FPS_GT511C3 fps ( 2, 3 );
```

```
void setup ( )  
{
```

```
  lcd.begin ( 16 , 2 ) ;  
  lcd.setCursor (0,0) ;
```

```
  delay ( 100);  
  fps.Open ();  
  fps.SetLED (true ) ;  
  lcd.print ( " Biomaster " );
```

```
  ?
```

```
  ?
```

```
  ?
```

```
  delay ( 5000) ;  
  lcd.setCursor (0.1) ;
```

```
  ?
```

```
  int button ;
```

```
  ?
```

```
  button = analogRead ( 0 ) // Read the value of the analog port A0
```

```
  lcd.setCursor (0.1) ;  
  if ( button < 800 ) {  
    lcd.print ( " SETUP " ) ;
```

```
    delay ( 1000) ;
```

```
    lcd.clear ();
```

```
    delay ( 1000) ;
```

```
    Enroll ();
```

```
  }
```

```
  ?
```

```
}
```

```
Enroll void ( )
```

```
{
```

```
  // Enroll test
```

```
  // Find open enroll id
```

```
  enrollid int = 21 ;
```

```
  usedid bool = true ;
```

```
while ( usedid == true )
{
  usedid = fps.CheckEnrolled ( enrollid );
  if ( usedid == true ) enrollid + + ;
}
fps.EnrollStart ( enrollid );

// Enroll


lcd.print ( " P Touch Start" );
delay ( 100);
❖
lcd.println ( enrollid );
while ( fps.IsPressFinger ( ) == false ) delay ( 100 ) ;
bool bret = fps.CaptureFinger (true ) ;
IRET int = 0;
if ( bret != false )
{
  lcd.println ( " Remove " );
  delay ( 3000 ) ;
  lcd.clear ();
  fps.Enroll1 ();
  while ( fps.IsPressFinger ( ) == true ) delay ( 100 ) ;
  lcd.println ( "put the finger " );
  delay ( 3000 ) ;
  lcd.clear ();
  while ( fps.IsPressFinger ( ) == false ) delay ( 100 ) ;
  bret = fps.CaptureFinger (true ) ;
  if ( bret != false )
  {
    lcd.println ( " Remove " );
    delay ( 3000 ) ;
    lcd.clear ();
    fps.Enroll2 ();
    while ( fps.IsPressFinger ( ) == true ) delay ( 100 ) ;
    lcd.println ( "put the finger " );
    delay ( 3000 ) ;
    lcd.clear ();
    while ( fps.IsPressFinger ( ) == false ) delay ( 100 ) ;
    bret = fps.CaptureFinger (true ) ;
    if ( bret != false )
    {
      lcd.println ( " Remove " );
      delay ( 3000 ) ;
      lcd.clear ();
      IRET fps.Enroll3 = ();
```

```
if ( iret == 0 )
{
  lcd.println ( " Joined " );
  delay ( 5000 );
  ?
  lcd.begin ( 16 , 2 );
  lcd.setCursor (0,0);
  delay ( 100);
  ?
  lcd.print ( " Biomaster ");
  ?
  fps.Open ();
  fps.SetLED ( false );
  delay ( 1000 );
  ?
  ?
  ?
  ?
}
else
{
  lcd.print ( " Enrolling Failed with error code : " );
  lcd.println ( IRET );
}
}
else lcd.println ( "Failed to capture third finger " );
}
else lcd.println ( "Failed to capture second finger " );
}
else lcd.println ( "Failed to capture first finger " );
}

void loop ( )
{

  // Identify fingerprint test
  if ( fps.IsPressFinger ( ) )
  ?
  fps.Open ();
  fps.SetLED (true );

  // Identify fingerprint test
  if ( fps.IsPressFinger ( ) )
  {
    delay ( 500 );
    lcd.clear ();
```

```
fps.CaptureFinger ( false) ;  
int id = fps.Identify1_N ( ) ;  
if ( id < 200 )  
  
{  
  delay ( 500 ) ;  
  lcd.clear () ;  
  
  lcd.print ( " LIBRERADO ID " ) ;  
     
  lcd.println (id ) ;  
  delay ( 3000 ) ;  
  
}  
else  
{  
  delay ( 500 ) ;  
  lcd.clear () ;  
  
  lcd.println ( "Unauthorized " ) ;  
  delay ( 3000 ) ;  
  
}  
}  
else  
{  
  delay ( 100 ) ;  
  lcd.clear () ;  
  
  lcd.println ( " Put your finger " ) ;  
  
  lcd.setCursor (0.1) ;  
  lcd.print ( " Biomaster " ) ;  
  
}  
  
}
```



jhawley
on **March 9, 2014 at 1:47 pm** said:

I'm not sure exactly where you are going with that code, but if you want to delete an ID, it is:
fps.DeleteID(id);



calwal

on **March 11, 2014 at 4:58 am** said:

Hello friend, I created this code to use biometric access control reader how to open the door.

My problem is that I do not know how to write the part of deleter ID, could you send me an example of a code that contains the part of deleting an example written in ID.

Walter grateful

I do not know which position the code writing DeletelD, would be in the loop, it would be in the setup

Thank you in advance



calwal

on **March 13, 2014 at 1:51 pm** said:

Hello friend could send me an example of how to write the arduino part ID to delete because I can not write so that the compiler understands.

thankful Walter



jhawley

on **June 10, 2014 at 2:03 pm** said:

to delete the finger print in id slot one you would call it like this:

```
fps.DeleteID(1);
```



CJ545

on **March 3, 2014 at 2:56 pm** said:

Hey, super job making the finger scanner work with the Arduino, you have saved me alot of work! Thanks! 😊

But I have a problem when I upload the enroll code to my Arduino, no information appears on the serial debug screen? Am I doing something wrong, shouldn't it at least post something on the serial debug screen??



jhawley

on **March 4, 2014 at 1:52 pm** said:

what pins did you use to connect the FPS? If you tried to use 0 and 1 it disables the serial debugger



CJ545

on **March 4, 2014 at 11:37 pm** said:

I've used pin 4 and 5.

And I didn't have a connector to the FPS, so I have soldered it to the solder points underneath, could there be any problems with that?



jhawley

on **March 5, 2014 at 2:00 pm** said:

You will really need to double check your connections then. Make sure that the pins you soldered to on the board are the ones that you are expecting them to be. It is really easy to reverse them doing that. There are 5 points underneath, and only 4 pins in the connector, so I'm not sure if they are what you are expecting them to be.

Also, it would be really easy to damage the chips on the board next to the solder points because they are so close. I think that those are not solder points. I think they are contact points used by their QA department for testing.



ozioma

on **May 12, 2014 at 4:55 am** said:

I'm having the same issue. Im using the C1 version but they use the same protocol so it should work. The blink sketch is the only thing that works. I tried replacing the 3 false's to true and still no result on the serial monitor. The fingerprint has been soldered to the arduino. But since the blink sketch works then it means my connections are right. So please what's the problem?



jhawley

on **June 10, 2014 at 2:06 pm** said:

If the blink sketch works, then then you probably have it hooked up correctly. Try running the sample sketch that does enrolling.



ozioma

on **May 14, 2014 at 5:24 am** said:

@CJ545 Hello,were you able to solve your problem..cz im having the same issue .will need your help if we solved it



aqualily6

on **March 3, 2014 at 12:20 pm** said:

Hello,

I have been trying to figure out your libraries but cannot understand them. This is my first look into coding (I have an electronics background) and am really stuck. Could you give me an idiots guide to setting the software up, presuming absolute noob status please?

Thanks

Kim



jhawley

on **March 4, 2014 at 1:54 pm** said:

I would start here:

<http://arduino.cc/en/Guide/HomePage>

Then also read the guide to libraries:

<http://arduino.cc/en/Guide/Libraries#.UxZLWldV8E>



aqualily6

on **March 5, 2014 at 12:21 pm** said:

Thanks I will take a look now.



zachg007

on **February 23, 2014 at 7:30 am** said:

Thanks so much for your hard work! Quick question: Your diagram above shows the RX, TX connections on Arduino pins 2, 3 but the code in your zip file calls for pins 4, 5. Just wanted to confirm that is a mistake and that the wiring diagram is out of date.....or maybe I'm overlooking something.

Also, I'm trying to get the blink sketch to work on the C1 fingerprint scanner model. What code exactly needs to change to make this work for the C1 model?

Thanks again!



jhawley

on **March 4, 2014 at 1:56 pm** said:

The most current code is always on github. Assign the pins to whichever ones you actually hooked it up to.

All I know about the C1 is that it has less onboard memory, and can store less templates as a result. It probably uses the same protocols. You would need to compare the 2 datasheets to figure it out.



jdabronx

on **February 10, 2014 at 11:32 am** said:

First – a big thank you. The code you have posted here had been a tremendous help in getting my fingerprint scanner working while connected to my Raspberry Pi. I am basically reverse engineering your C++ code to Python and have successfully written the library methods to open, change baudrate, setLED and get responses back from the device.

I am now starting the enroll process which looks pretty straight forward. However, I have a question about the validate 1:N code in your example:

Do you need to turn on the LED before you can test if a finger is pressed? Or, can you turn on the LED after you detect that a finger has been placed on the sensor?

I thought I read that the order needs to be (1) turn on LED and (2) check if a finger has been pressed. Personally, I think it would be more user friendly to turn on the LED after a finger has been pressed to indicate that the device is now searching for a match.

As always, thank you for your assistance thus far and any help would be appreciated.



jhawley

on **February 10, 2014 at 1:30 pm** said:

That's awesome! I have dabbled in python a little, but not very much. I would love to see your code when it is done =) (And I could post it here too if you want)

I'm not sure if it would help you or not, but another user here changed out the serial pieces of my library and made it work on the RPi using C.

The camera in the FPS does not work very well without the light. If you want it to see anything (including if there is a finger there), you need to turn on the LED. I would suggest some thing else in your project that triggers the light/etc.

The order for testing if a finger works would be:

1. LED on.
2. IsPressFinger
3. CaptureFinger
4. Verify



jdabronx

on **February 10, 2014 at 1:47 pm** said:

Thanks for the fast reply. I thought about the C option on the RPi but decided against it since I wanted to use this project to learn Python as it seems to be the language of choice for the RPi (probably because of the Py/Pi thing).

I want to use the fingerprint scanner to control a garage door so I was hoping to initiate the scanning by a finger press. The other alternative is to either (a) always have the LED on (not sure if this is such a good idea) or (b) some

other method as you suggest to turn it on (maybe a motion sensor or nearby switch?) Anyway, thanks for the help — I'll share code once closer to completion.



jhawley

on **February 10, 2014 at 2:19 pm** said:

If you mounted the finger print scanner on a hinge inside a box, and put a small pushbutton behind it, you could fake triggering it by pushing your finger onto it.



gcriniti

on **January 22, 2014 at 4:06 pm** said:

Hey, Josh! Wow! Thanks for all the work you're doing for everybody! You're probably getting a lot more attention than you wanted from this. I, too, am having problems similar to other recent posts. The enroll routine seems to work, except that the id was always 0. I made the changes that you suggested, changing the "false" to "true" in three places. Now, my id is stuck at 4, although I still get a successful enroll, or so it says. Problem is, the verify does not work. I've enrolled most of my fingers now; soon will have to go with toes. Is there a way to determine which routine is not working; enroll or verify?



Ramez

on **February 8, 2014 at 8:37 am** said:

mine works good, have you changed the Baud rate?



jhawley

on **February 10, 2014 at 1:19 pm** said:

I did implement the change baud rate command, but due to several limitations of my hardware/project, I was not able to test it.



Ramez

 on [February 10, 2014 at 4:13 pm](#) said:

Greeting jhawley,

I was asking him that cause pushing the Baud rate higher than 9600 will lead to up-normal behavior due to (delay at resistors) in voltage step down technique used here (1k & 560ohm), So it may be the problem.



jhawley

on [February 11, 2014 at 1:58 pm](#) said:

oops! I replied to your comment from the wordpress dashboard, and did not see that it was a reply itself... doh!



jhawley

on [February 10, 2014 at 1:31 pm](#) said:

Yes. Load up the demo software and see what is enrolled on it.



Ramez

on [January 13, 2014 at 8:52 pm](#) said:

Sorry but I think there is a serious bug in your Enroll Code, It never saves the Enroll to the data-base and it's obviously as I put in the first Enroll step "Enroll1" a finger and in next step "Enroll2" another finger! and in "Enroll3" step another 3rd finger! and it says "Enrolling Successful"!! plus every time it says "Enroll # 1" which means enrollid is 1 and never goes to 2! also the "fps_IDfinger" code doesn't recognize it at all...I do appreciate your help Josh so maybe when you tried to fix the "empty id finder" something goes wrong?.

thanks in advance.



jhawley



on **January 15, 2014 at 1:23 pm** said:

Looks like both you and jethrock have the same issue. I'm replying to you both at the same time with a cut/paste of the same post =D

Are you using the same finger each time it asks for one? You have to use the same finger 3 times so that it has 3 pictures of the same fingerprint to compare when making a template.

Also, this section of the code does have a bug(change to github submitted)... below is the new version... basically just change the 3 falses to trues. if you have any trouble, delete it all except int enrollid=0; and just change the zero to the desired enrollid number.

```
// find open enroll id
int enrollid = 0;
bool usedid = true;
while (usedid == true)
{
    usedid = fps.CheckEnrolled(enrollid);
    if (usedid==true) enrollid++;
}
fps.EnrollStart(enrollid);
```



Ramez

on **January 15, 2014 at 2:27 pm** said:

oh thanks god...now it works like charm I've changed the 3 "false" to "true"
just a small question how to delete fps database or a specific IDfinger?

many thanks jhawley..



Ramez

on **January 15, 2014 at 7:06 pm** said:

O.K now I'd to surf more time on the header to know more about how to delete db. (^_^) but a small question here:

```
bool CheckEnrolled(int id);
return "1" for position in used not "3" can you please check that?
```

for people who want to delete specific ID just call under setup() :

```
fps.DeleteID(ID HERE)
```

To remove whole FPS database type:

```
fps.DeleteAll()
```

to make sure that finger data has been wiped:

```
int x = fps.GetEnrollCount();
```

```
Serial.print(x);
```

then open serial monitor and check if it's responding 0 that means finger database is empty.



jhawley

on **January 15, 2014 at 9:53 pm** said:

Yeah this post only covers the basics of the library, and not every single function.

I'm confused by your question. if you call `fps.CheckEnrolled(3)` it should return 1 if there is an enrollment at id 3, and zero if not.



Ramez

on **January 16, 2014 at 3:52 am** said:

sorry for confusion..I mean if I had I template already in ID say 10 and call the

```
CheckEnrolled(10);
```

the problem is it returns 1 "which means database full" not 3 "which means ID is in use"?

just a tiny question came to mind, How to repeatedly Enroll fingers (without resting my arduino each time to enter new enroll)...I mean as example after a successful enroll for ID 5 it moves on and asks me to enroll finger for ID 6 then after successful enroll operation for ID 6 it asks me to enroll finger for ID 7 and so on...till filling the whole 200 ID's, Any recommendation please?

many thanks in advance...



jhawley

on **January 16, 2014 at 8:53 am** said:

`bool CheckEnrolled(int)` returns a bool. It can never have a return value of 3. It returns a 1 (true, that ID number IS enrolled) or a 0 (false, that ID number is NOT enrolled).



Ramez

on **January 16, 2014 at 11:38 am** said:

oh yeah,now i got it...
really thanks for your time Mr.Jhawley.
Greeting from Egypt 😊



jethrock

on **January 12, 2014 at 8:58 pm** said:

hey there jhawley, Do you have an updated enroll example for arduino? because the one you've posted in github isn't working, the fingerprint scanner says "enrollment successful" but when I'm going to verify it, It doesn't read my fingerprint and another problem is it continues to enroll and enroll in the same ID but doesn't verify any of them. Thank you!



jhawley

on **January 15, 2014 at 1:11 pm** said:

Are you using the same finger each time it asks for one? You have to use the same finger 3 times so that it has 3 pictures of the same fingerprint to compare when making a template.

Also, this section of the code does have a bug(change to github submitted)... below is the new version... basically just change the 3 falses to trues. if you have any trouble, delete it all except `int enrollid=0;` and just change the zero to the desired enrollid number.

```
// find open enroll id
int enrollid = 0;
bool usedid = true;
while (usedid == true)
{
    usedid = fps.CheckEnrolled(enrollid);
    if (usedid==true) enrollid++;
}
```

```
}
fps.EnrollStart(enrollid);
```



calwal

on **March 8, 2014 at 1:19 pm** said:

Friend could help me,
I'm having trouble adding the part to delete ID.
grateful thanks



jhawley

on **March 8, 2014 at 3:46 pm** said:

there are 2 ways to delete from the fps.

fps.DeleteAll() deletes all of the ids on the device.

fps.DeleteID(1) would delete the ID in slot 1.



calwal

on **March 8, 2014 at 6:07 pm** said:

Friend sorry to bother you , but I can already register id, I can verify that the registered id agree but I am not knowing how to write the rows to delete .

I tell my program to give you a look and see what is wrong .

thank you very much .

```
# include " FPS_GT511C3.h "
```

```
# include " SoftwareSerial.h "
```

```
# include
```

```
// Hardware setup – FPS connected to:
```

```
// Digital pin 2 ( Arduino rx , tx fps )
```

```
// Digital pin 3 ( arduino tx – 560ohm resistor – fps tx – 1000ohm resistor – ground )
```

```
// This voltage divider brings the 5v tx line down to about 3.2v so we dont fry our fps
```

```
LiquidCrystal lcd ( 8,9,4,5,6,7 ) ;
```

```
FPS_GT511C3 fps ( 2, 3 );
```



```
void setup ( )
{

    lcd.begin ( 16 , 2 ) ;
    lcd.setCursor (0,0) ;

    delay ( 100);
    fps.Open ();
    fps.SetLED (true ) ;
    lcd.print ( " Biomaster ");
    ?
    ?
    ?
    delay ( 5000) ;
    lcd.setCursor (0.1) ;
    ?
    int button ;
    ?
    button = analogRead ( 0 ) // Read the value of the analog port A0
    lcd.setCursor (0.1) ;
    if ( button < 800 ) {
    lcd.print ( " SETUP " ) ;
    delay ( 1000) ;
    lcd.clear ();
    delay ( 1000) ;
    Enroll ();
    }

    ?
}

Enroll void ( )
{
    // Enroll test

    // Find open enroll id
    enrollid int = 21 ;
    usedid bool = true ;
    while ( usedid == true )
    {
        usedid = fps.CheckEnrolled ( enrollid );
        if ( usedid == true ) enrollid + + ;
    }
    fps.EnrollStart ( enrollid );
```

```
// Enroll
```

```
lcd.print ( " P Touch Start" );
delay ( 100);

❖
lcd.println ( enrollid );
while ( fps.IsPressFinger ( ) == false ) delay ( 100 ) ;
bool bret = fps.CaptureFinger (true ) ;
IRET int = 0;
if ( bret != false )
{
    lcd.println ( " Remove " ) ;
    delay ( 3000) ;
    lcd.clear ();
    fps.Enroll1 ();
    while ( fps.IsPressFinger ( ) == true ) delay ( 100 ) ;
    lcd.println ( "put the finger " ) ;
    delay ( 3000) ;
    lcd.clear ();
    while ( fps.IsPressFinger ( ) == false ) delay ( 100 ) ;
    bret = fps.CaptureFinger (true ) ;
    if ( bret != false )
    {
        lcd.println ( " Remove " ) ;
        delay ( 3000) ;
        lcd.clear ();
        fps.Enroll2 ();
        while ( fps.IsPressFinger ( ) == true ) delay ( 100 ) ;
        lcd.println ( "put the finger " ) ;
        delay ( 3000) ;
        lcd.clear ();
        while ( fps.IsPressFinger ( ) == false ) delay ( 100 ) ;
        bret = fps.CaptureFinger (true ) ;
        if ( bret != false )
        {
            lcd.println ( " Remove " ) ;
            delay ( 3000) ;
            lcd.clear ();
            IRET fps.Enroll3 = ();
            if ( iret == 0 )
            {
                lcd.println ( " Joined " ) ;
                delay ( 5000) ;
                ❖
                lcd.begin ( 16 , 2 ) ;
```

```

    lcd.setCursor (0,0) ;
    delay ( 100);

    ?
    lcd.print ( " Biomaster ");

    ?
    fps.Open ();
    fps.SetLED ( false) ;
    delay ( 1000) ;

    ?
    ?
    ?
    ?
    }
    else
    {
        lcd.print ( " Enrolling Failed with error code : " ) ;
        lcd.println ( IRET );
    }
}
else lcd.println ( "Failed to capture third finger " ) ;
}
else lcd.println ( "Failed to capture second finger " ) ;
}
else lcd.println ( "Failed to capture first finger " ) ;
}

void loop ( )
{

    // Identify fingerprint test
    if ( fps.IsPressFinger ( ) )
    ?
        fps.Open ();
        fps.SetLED (true ) ;

        // Identify fingerprint test
        if ( fps.IsPressFinger ( ) )
        {
            delay ( 500 ) ;
            lcd.clear ();

            fps.CaptureFinger ( false) ;
            int id = fps.Identify1_N ( ) ;
            if ( id < 200 )

```

```
{
  delay ( 500 ) ;
  lcd.clear ();

  lcd.print ( " LIBRERADO ID " ) ;

  lcd.println (id );
  delay ( 3000 ) ;

}
else
{
  delay ( 500 ) ;
  lcd.clear ();

  lcd.println ( "Unauthorized " ) ;
  delay ( 3000 ) ;

}
}
else
{
  delay ( 100);
  lcd.clear ();

  lcd.println ( " Put your finger " ) ;

  lcd.setCursor (0.1) ;
  lcd.print ( " Biomaster ");

}

}
```



jethrock

on **January 7, 2014 at 12:37 am** said:

hey jhawley, how can I get the database from the FPS through Arduino? Thank you!



jhawley

on **January 7, 2014 at 9:58 am** said:

That is a memory challenge that I have not solved yet. I do think that it is possible, but I have not had time to work on it.



totoman

on **January 6, 2014 at 1:58 pm** said:

Hey do you think the library would the with the Arduino Due, using one of the four hardware serial port?



jhawley

on **January 6, 2014 at 2:01 pm** said:

Yes, but you would need to replace a few lines of code. There are 4 or 5 lines that call the SoftwareSerial library, you would need to replace those with the appropriate lines for using the hardware. This has been done once already by another user to make it work on a Raspberry Pi.



jethrock

on **January 6, 2014 at 6:35 am** said:

Hello there jhawley! I just want to ask if this FPS is compatible with atmega 168? because it won't blink after I uploaded the FPS_BLINK sketch example. Even the enroll and idfinger sketch won't work but it uploads successfully to the atmega168. Thank you!



jhawley

on **January 6, 2014 at 8:48 am** said:

Are the pins on the atmega168 3.3v or 5v? In any case it sounds like your serial communication is the issue. If they are 3.3v already, then you do not need the voltage divider described above.



jethrock

on **January 6, 2014 at 6:22 pm** said:

the atmega 168 is 5v. When I use the sample program for windows by sparkfun, it works perfectly. Why would that be?



jhawley

on **January 7, 2014 at 10:00 am** said:

Sounds like something is pinned wrong then. Maybe you have the rx and tx backward? does your voltage divider properly limit voltage?



Roland

on **December 23, 2013 at 3:01 pm** said:

Hi

I'm working with a GT-511C1, similar to the C3, and wondering if you can help me on the data format. I'm sending characters directly at the moment but will be using a pic.

To turn the led on, what do I actually send? The checksum has me confused. I see it as;

55h

AAh

1 (ID)

0

1 (parameter, LED on)

0

0

0

12h (turn on led)

0

checksum

Is the checksum? in decimal;

$85 + 170 + 1 + 1 + 18 = 275$

$= 255 + 20$

FFh 14h

Which order are they sent in?

So if I send;

55h

AAh

1

0

1

0

0

0

12h

0

FFh

14h

should the LED turn on?

Regards

Roland



jhawley

on **December 24, 2013 at 11:55 am** said:

To turn the LED on it should be:

55 AA 01 00 01 00 00 00 00 12 00 13 01

it looks like your conversion of 275 to hex is what was wrong. In little endian it literally converts like this (13h) $19 + (01h) 256 = 275$. The 01 counts as 256 because it is in the 9th bit position.



Roland

on **December 26, 2013 at 12:36 pm** said:

Hi jhawley

Thanks for the reply. I'm still having no luck at all. I did realise that error, but still no working comms.

I actually think their product spec is pathetic. Why not use all-ascii

commands like the AT+ command set? Anyway if you could please bear with me..

I connected USB directly on the pads on the rear, and the module works fine with their demo prog. Then, with the plug connector, I made connection via a FTDI-USB-TTL-5V with a 5V-3V3 interface, and using another comms program.

So now I have both USB's plugged in, talking to it with both interfaces, and can use their demo, but sending 232 does not work. What I find very odd is that the reply from the device looks nothing like what is in the manual. I send;

```
55 aa 01 00 01 00 00 00 00 12 00 13 01
```

but the reply is a barrage of info;

```
00
```

```
0d 0a
```

```
0d 52 31 30 20 30 30 30 30 30 30 30 20 35 31 32 0a
```

```
0d 0a
```

```
0d 52 31 30 20 30 30 30 30 30 30 30 20 35 31 32 0a
```

```
0d 0a
```

```
0d 52 31 30 20 30 30 30 30 30 30 30 20 35 31 32 0a
```

```
0d 0a
```

```
0d 52 31 30 20 30 30 30 30 30 30 30 20 35 31 32 0a
```

```
0d 0a
```

in fact 14 lines of the above.

in ASCII it looks like;

```
R1000000000512
```

x 14 times

So, what I'm seeing is an ack of 52h which is no-where in the manual. I see similar replies on my 232 interface if I use their working demo.

Unfortunately, I cannot see exactly what is being sent via the direct USB, but whatever I'm sending is not doing the trick.

Is there a way to use the direct USB link as a HID connection?

Any suggestions? I'm stuck.

Regards

Roland



jhawley

on **December 26, 2013 at 12:53 pm** said:

They used this format because it is more consistent across multiple commands, and takes less code/memory on the FPS.

None of those bytes conform to any of the comm spec. Are you operating at the wrong baud speed? The demo is a little misleading. It makes it look like the default baud rate is 115200, when in fact it is 9600.

Also, the ACK packet should look like this:

```
55 AA 01 00 01 00 00 00 30 00 31 01
```

or NACK:

```
55 AA 01 00 01 00 00 00 31 00 32 01
```



Roland

on [December 26, 2013 at 1:22 pm](#) said:

Thanks jhawley, only saw your reply now.

Maybe I should write up after my project. There's precious little info on using this device, in spite of the numerous vendors selling it.

It would be nice if the manufacturer could release a few firmware updates, to make it simpler to use, like you've done. Like send LEDON, in plain text, and led comes on...

Thanks for the help.

Regards

Roland



Roland

on [December 26, 2013 at 1:15 pm](#) said:

Yeeha !!

Ok, the gospel command to turn the LED on is;

```
55 AA 01 00 01 00 00 00 12 00 13 01
```

and the reply is;

```
55 aa 01 00 00 00 00 00 30 00 30 01
```

There was one too many 00 in the command. Also, I had to unplug both USB's to reset it, and only use the TTL connection.

Phew, finally on my way after three days of futility.

Regards

Roland



jhawley

on **December 26, 2013 at 2:28 pm** said:

Good! I'm glad you got it to work. You might consider using my library, All you need to change is the 5 lines of code (or so) that actually do the sending/receiving on the serial lines. I had one other user do that for a Raspberry Pi.



safiyyah

on **April 15, 2014 at 12:46 pm** said:

Hi

I am also using the GT511 C1 module and I am having a lot of trouble getting it to communicate with the arduino Uno. What suggestions do you have that will enable me to allow communication to occur between the two modules?



NoviceEngineers

on **December 4, 2013 at 11:08 am** said:

I am using the enroll example but it does not seem to be communicating with the fps because I can use 3 different fingers and the enrollment still be successful. Also it does not save the successfully enrolled finger to the fps because when I use the verifyid example it says "finger not found" PLEASE HELP



jhawley

on [December 4, 2013 at 11:09 am](#) said:

Does the blink sketch work?



NoviceEngineers

on [December 5, 2013 at 10:48 am](#) said:

Yes the blink sketch works but that is the only sketch that successfully works



jhawley

on [December 5, 2013 at 3:58 pm](#) said:

If the FPS is blinking, then the serial communications are working properly.

Why would you use 3 different fingers for enrollment? It does not do verification when you enroll. That would just confuse it, and make a corrupt template. Try changing the enrollment ID and to another slot and then enroll again.

Find this section in the code:

```
// find open enroll id
```

```
int enrollid = 0;
```

Change the enrollid to another value (between 0 and 199) to enroll in a different slot.

It looks like there is a bug in my code that finds an empty enrollid too...

Change:

```
if (okid==false) okid++;
```

To:

```
if (okid==false) enrollid++;
```

Edit: Updated the code on github and in the zip file here.



NoviceEngineers

on [December 6, 2013 at 5:56 am](#) said:

Is there any way to randomize the enroll id, if the enroll id 0 is already used is there a way to change the enroll id a number

that is not being used? Also is there anyway to assign your own custom enroll id like say by a name or something other than just 1-199?



jhawley

on **December 6, 2013 at 2:11 pm** said:

One of the first lines of the Enroll function in the sample code assigns the enroll id. Just change the value.



NoviceEngineers

on **December 6, 2013 at 10:39 am** said:

Also the verifyid code is not working and correctly identifying my finger



jhawley

on **December 6, 2013 at 2:12 pm** said:

What about it is not working?



gcriniti

on **November 27, 2013 at 1:14 pm** said:

Hi, JHawley! Thanks for posting your work, it's a tremendous help. I'm just getting started with Arduino and the finger print scanner. I've downloaded and installed your library and it works! I've only tried the blink sketch so far, but will be using the others soon. Looking ahead, I will need to transfer images to/from a computer, and through the Arduino. This

seems to be something that you haven't been able to master yet. Are you still working on it? Can this be done? Do we need to look at other hardware?



jhawley

on **November 27, 2013 at 2:35 pm** said:

The FPS does not have flow control on its serial lines. That means that it will just shove the bits across as fast as it can. The Arduino has 2,048 bytes of ram and the image file that you are requesting from the FPS is 52,116 bytes. This means that at best, the Arduino can only be a pass through to another device for the image. While you could possibly write code to make this happen, you would likely not be able to do much else with the Arduino because of the ram consumed just buffering and forwarding the info. Also, timing would become an issue. Any network or other delay, and the data from the FPS would just overrun the buffer, and corrupt your image. It is not an impossible task, but realistically it is beyond the comfortable limits of an Arduino.

Now the fingerprint templates that it generates are much smaller (only 498 bytes), but they are still a quarter of the available ram. I have been experimenting with capturing those. Getting it through SoftwareSerial has been the issue with no flow control. I plan to sort that part out, but I have not really had a lot of time to work on it.

I am looking in to making it work with a Raspberry Pi, and a Netduino plus 2. Both of those MCUs have a lot more ram.



tawanda

on **June 22, 2014 at 7:03 am** said:

Hi have you managed to make this fingerprint reader communicate with the raspberry pi using python? I am struggling to make it work, can you help please.



jhawley

on **June 22, 2014 at 12:09 pm** said:

There was another user that adapted my library to work on an RPi, but not using python.



Franet

on **November 25, 2013 at 2:46 pm** said:

Hi Hawley!

I just got this Module like a month ago, and I've been having issues making it work with my Arduino...I think it's my TX & RX connection that's wrong, coz the FPS module is powered on.

The RX & TX pins on the FPS requires 3.3V to work right?? ... but according to your diagram above, your TX jumper from the Arduino was just connected directly to the FPS without the 3.3V required.

Can you please help explain better the TX & RX connection between the Arduino board & the FPS module using the circuit diagram...

Thanks!



jhawley

on **November 25, 2013 at 5:39 pm** said:

I think you read my diagram wrong. The TX on the FPS -> RX on the Arduino does not need a voltage divider because the FPS only sends up to 3.3v which is well below the 5v for the Arduino.

Longer explanation:

Serial works on high vs low voltage. low is usually 0v (or close to it). High on an Arduino is 5v, but it also reads high at voltages greater than 2.4v.

The 3.3v TX going from the FPS module directly to the 5v Arduino RX does not need amplification because 3.3v is enough for the hardware to read "high".

The 5v TX going from the Arduino to the 3.3v RX on the FPS can be damaged by 5v, so we put a voltage divider in between.

When the Arduino sends 5v on the line the voltage divider lowers that to 3.3v, and the FPS reads that as high without taking damage.



Franet

on **December 2, 2013 at 3:55 am** said:

Hi JHayley!

Thanks for the reply... Finally gotten around 2.8V output to my FPS RX pin.

I've downloaded you library, but I'm having issues making the "FPS_Blink.ino" sketch work...

The FPS Led isn't blinking.

According to this line of code:

```
FPS_GT511C3 fps(4, 5);  
SoftwareSerial lcd(2, 3);
```

I connected my FPS TX to pin 4 (On Arduino Board) & FPS RX to pin 5 (on Arduino board)

What about the SoftwareSerial lcd (2,3); line ??

I do not understand that line. What's it supposed to do??

Thanks!



jhawley

on **December 2, 2013 at 11:23 am** said:

Why are you only getting 2.8v? I was getting 3.22v. You should test your voltage divider.

First, unhook all of the wires from your FPS.

Next, unhook the wire from the digital pin and connect it to the +5v pin on the arduino. Now test the wire that would go to the FPS (if it were not unhooked) with a volt meter. If it does not read ~3.2v, then something is wrong with it. Maybe the resistors are reversed, damaged, or the wrong values? In any case, you should be able to play with it until you can get it to work.

Alternately you could just buy a logic level shifter from Sparkfun

<https://www.sparkfun.com/products/11978>

I'm confused as to where you are seeing that line. I opened the download and looked in all of the sketches. I do not see it anywhere. In any case, it is not related to the FPS library.



Franet

on **December 5, 2013 at 4:04 am** said:

Hi JHaylew...

I did exactly what you told me to do...
And I got a value around 3.16V.

When I connected my FPS back to it, I was getting a value around 2.87V again...

Don't know what's wrong...
Pls what else can I do??
This is my Final Year Project...

I downloaded your Library from GitHub, and the content in the FPS_Blink.ino in the examples folder is this:

```
/*  
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  
  
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 "LIB_GT511C3.h"  
#include "SoftwareSerial.h"
```

```
// Hardware setup – FPS connected to:  
// digital pin 4(arduino rx, fps tx)  
// digital pin 5(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(4, 5);  
SoftwareSerial lcd(2, 3);
```

```
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);  
}
```

Can u see the "SoftwareSerial lcd(2, 3);" ??
That was what I was asking about the other time...



jhawley
on **December 5, 2013 at 8:55 am** said:

Ahh yes I see it now. I was using that to output to a LCD backpack when I was testing the code and left it in.

Delete this line:
SoftwareSerial lcd(2, 3);

Change this line:
#include "LIB_GT511C3.h"

TO:
#include "FPS_GT511C3.h"

RE: the voltage, I never did measure it with the FPS attached.
If you are getting 3.16v without it attached, the voltage divider is connected properly.



Franet
on **December 16, 2013 at 4:16 pm** said:

Hello JHaylew!

Thanks for your help so far...

Still having the same issues...I'm beginning to think the fingerprint scanner is faulty (though it's powered on)

Pls I need to draw my circuit diagram for the FPS, Arduino board & LCD display I plan to use...
Something similar to the circuit diagram you drew above.

Pls can u give me a link to download that type of software you used to draw your circuit diagram above?

Thanks!



NoviceEngineers

on **November 13, 2013 at 7:41 am** said:

Hello there fellow tinkerer,

My team and I have ran into a standstill. We would like to know if you could assist us with your knowledge.

Our team is trying to implement the fingerprint scanner into a locker that we also want to have a locking mechanism. But we want to have the fingerprint scanner send the locking mechanism a signal in which it will open the locker.

We would like to know if you knew what our next step would be to achieve that goal. Also we would like to know if we would need to have to know how to program to finish this project.



jhawley

on **November 14, 2013 at 8:48 am** said:

What you are asking about is really 2 separate things that need to be bridged by a micro controller (like an Arduino). First you need the micro controller to be talking to the FPS (so that it knows when an authorized finger is pressed). Second you need the lock hardware and any other hardware to interface with that.

Some programming would be involved. Nothing overly complex though. Have a look at <http://arduino.cc/en/Guide/HomePage>



mx5jon

on **November 10, 2013 at 3:20 am** said:

Hi jhawley

I'm going to try to adapt for Raspberry Pi as this has loads more memory etc, also there is a good serial library from Gordon Henderson.

<https://projects.drogon.net/raspberry-pi/wiringpi/serial-library/>

Could you email me privately to discuss?

I guess you can access my email address from my registration?.

What you've done seems so much more usable than the manufacturers supplied software!.

Regards

mx5jon



ngwako

on **November 2, 2013 at 5:09 pm** said:

hi can the fps for C3 library also work for gt-c511c1 and arduino uno ? cos i get this error
"Arduino: 1.5.3 (Windows 7), Board: "Arduino Uno"

FPS_Blink.ino:1:25: error: LIB_GT511C3.h: No such file or directory

FPS_Blink:9: error: 'FPS_GT511C3' does not name a type

FPS_Blink.ino: In function 'void setup()':

FPS_Blink:14: error: 'fps' was not declared in this scope

FPS_Blink.ino: In function 'void loop()':

FPS_Blink:22: error: 'fps' was not declared in this scope

This report would have more information with
"Show verbose output during compilation"
enabled in File > Preferences.
"

please help and thanks in advance



jhawley

on **November 3, 2013 at 1:40 pm** said:

Looks like you have a problem with the compiler. Check and make sure that all of the library files are in the right locations.



Rajendra

on **October 7, 2013 at 10:54 pm** said:

Hello jhawley,

Thanks for nice blog.

We just ordered GT-511C3 for our project.

I was having few queries regarding it power supply:

- 1) Can i connect it this module directly to USB. Is it possible to use same USB power for powering this module?
- 2) In case of RS232 can we obtain the power from RS232 pins or do we need to power it separately.?
- 3) Do we need any level shifter to be use while testing this device through PC. (USB or RS232 connection) ?

Regards



jhawley

on **October 8, 2013 at 11:02 am** said:

- 1) The specs and data sheet seem to imply that you can hook this up to USB, but do not say how. I don't know enough about the internals of USB to be able to answer you. I do know that USB is capable of supplying a lot of power, so you will likely not need separate power when using it.
- 2) Power must be supplied to different pins than TX/RX

3) Again this depends on the specs. You should look up what voltage those levels run at.

If USB to a computer is the way you want to go, you need a 5v FTDI breakout from [like this one](#)

You can see it done [here](#).



mx5jon

on **November 10, 2013 at 3:09 am** said:

I think from contacting the manufacturer that there is a different model for USB, suggest you contact them directly. That was where I also found out about the 5v supply but 3.3v i/o requirement (it's nit at all clear in their documentation). Here is the manufacturers web page: <http://www.adh-tech.com.tw/?9,gt-511c2>



pi_fan

on **January 5, 2014 at 4:58 pm** said:

Hi,

Thanks so much for this work!

I have a GT-511C3 FPS. I bought an FTDI breakout board and soldered the leads from the FPS.

The USB from the FTDI is connected to a Raspberry Pi.

The USB shows up on the Pi as

"Bus 001 Device 008: ID 0403:6001 Future Technology Devices International, Ltd FT232 USB-Serial (UART) IC"

I tried setting the voltage to 3.3v as well as 5V on the FTDI board.

The green light on the FPS is on. I assume it means it's receiving the power.

However, the open command doesn't work.

The FTDI board has six leads, whereas the FPS has only 4 leads (Power, GND, rx, tx). I connected those four pins to the corresponding pins on the FTDI board, while leaving the other two (CTS and DTR) on FTDI unfilled.

My first question is – is this correct?

Because if this is incorrect, then the FTDI board is not sending the data to

the FPS.

Do you know of any 4-pin FTDI board?

Thanks for your help!



pi_fan

on **January 6, 2014 at 8:45 am** said:

Hi,

I found my problem. The TX and RX from the FPS need to be connected (soldered) to RX and TX of the FTDI board. That fixed my problem.

With this, the FPS is working from Raspberry Pi thru USB using PiWiring library.

Hope it might help others.



jhawley

on **January 6, 2014 at 9:18 am** said:

FTDI is by definition 6 pins. The last 2 pins are for serial flow control, which the FPS does not use, and it is OK not to use them. If you want to check if your FTDI board works, plug it in to a computer and use the demo software to see if your connections are right.

If you are connecting it to a RPi, you could consider using the UART pins instead of USB too. The RPi is already 3.3v, so you do not need a voltage divider.

See this link for more info (UART is at the bottom):

<https://sites.google.com/site/semilleroadt/raspberry-pi-tutorials/gpio>

Pins: <http://jeffskinnerbox.files.wordpress.com/2012/12/mini-uart.jpg>



jeff

on **October 2, 2013 at 6:19 pm** said:

Arduino from damaging the board. Arduino tx – R1(560ohm) – FPS rx – R2(1kohm) – ground
(see diagram)



jeff

on **October 1, 2013 at 8:59 pm** said:

where is the diagram?



jhawley

on **October 1, 2013 at 10:20 pm** said:

of what?



dnkupfer

on **October 30, 2013 at 9:37 am** said:

Hi jhawley,

at the beginning of your post, in point 1. you said the following:

1. It uses 3.3v logic instead of the Arduino's 5v. I used a voltage divider to keep the Arduino from damaging the board. Arduino tx – R1(560ohm) – FPS rx – R2(1kohm) – ground (see diagram).

“(see diagram)” is suppose to point to a diagram of the wiring described in point 1 but I couldn't find any diagram in your post.

Any help will be much appreciated.

Many thanks



jhawley

on **October 30, 2013 at 11:54 am** said:

ahh now i see. I meant to go back and make one... but forgot about it. I'll add it to the post above.



Ed Fish

on [September 30, 2013 at 1:31 am](#) said:

Hi thanks for the great article. Was wondering if it is possible to access the actual hexadecimal template of the finger and store that information?



jhawley

on [October 1, 2013 at 11:09 am](#) said:

It is possible to do so. The templates are 498 bytes, and so could be captured by an Arduino. It would take some careful memory planning though, as it is roughly a quarter of your ram. You would also need to address the issue that SoftwareSerial's buffer is only 64 bytes. Because of the memory involved, I did not include this command in the library(which is more general purpose). The command is detailed on page 28 of the data sheet.



Josue

on [September 23, 2013 at 12:35 pm](#) said:

Hi, I have a couple of questions:

when you scan (not when you enroll), your finger, what type of data saved in its memory the FPS (string, char, image, bytes, hexadecimal, binary code)?

And how to extract that data?

I guess that the library should be modified to make it but , what can you tell me about?



jhawley

on [September 23, 2013 at 2:36 pm](#) said:

When you capture the finger using CaptureFinger(), It stores the image in the memory on the FPS. The best way I can think of to extract the data is to use GetImage. It is not currently implemented in my library because of memory issues (the resulting serial data comes in one stream, and is much more than the arduino can handle). If I could get the serial buffer not to overrun, I could extract this data in segments and shoot it out a network port or something.

The device DOES support this, and you can see it in use in sparkfun's video.



Josue

on [September 24, 2013 at 4:52 pm](#) said:

just hypothetically if the library were removed other features you think would have enough room for at least GetRawImage function (), these gets an image that is 19 200 bytes.... after all Binary sketch size is 32256 arduino

regards



jhawley

on [September 24, 2013 at 7:43 pm](#) said:

Not even close. The arduino only has 2048 bytes of ram to work with. Complicating matters is the fact that the SoftwareSerial library that I'm using to talk to it only has a 64 byte buffer. I'm thinking about adjusting the baud rate down to 2400 or so. Then I may be able to keep the buffer clear. It will not be fast, but it will function. I'm still testing with it.



Josue

on [September 25, 2013 at 2:29 pm](#) said:

then, from what I understand at this point is impossible to obtain the raw image of the fingerprint from the arduino.



jhawley

on [September 29, 2013 at 9:27 pm](#) said:

Not impossible, very difficult, and the image must be transferred through the arduino. Still need to work out timings, and also have a destination that can accept it as fast as the FPS can send it.



Juan

on **September 5, 2013 at 6:19 pm** said:

Hi,

I'm using the enroll and the identification sketch, but it seems like it does not save in the db the fingerprint.

It says that the fingerprint was sucessfully saved, but when i try to identify it, it says "Not found".

Any ideas?



jhawley

on **September 9, 2013 at 9:24 am** said:

Well first lets split the problem in half. make a simple sketch that calls
`int x = fps.GetEnrollCount();`
and then output x to the debug window. If it is zero, then the problem is with enrollment. if it is more than zero, then the problem is with identification.



wendell

on **September 2, 2013 at 3:47 am** said:

Hello, Hawley

I am confused on how to connect the FPS to arduino

On the front page it says:

Arduino from damaging the board. Arduino tx – R1(1kohm) – FPS rx – R2(560ohm) – ground (see diagram)

On the zip file it says:

digital pin 5(arduino tx – 560ohm resistor – fps tx – 1000ohm resistor – ground)
this voltage divider brings the 5v tx line down to about 3.2v so we dont fry our fps

which one is the right one?

Sorry but still new to this stuffs.



jhawley

on **September 9, 2013 at 9:13 am** said:

@wendell Hmm... that is interesting. Clearly one of them is backwards. An easy way to test it would to use an volt meter on it. (Arduino **+5v** – 560ohm resistor – volt meter red – 1000ohm resistor – ground) and see if it is somewhere near 3.3 volts. I think when I did that check i got 3.2v. If it is much farther than 0.1v off, reverse the resistors and try it again. I am at work at the moment, but I will check my setup tonight and let you know.



jhawley

on **September 11, 2013 at 6:46 pm** said:

The zip file was correct. the 1k goes to ground, and the 560 goes to the Arduino. I must have just reversed them when I was posting this.



Daniel Beaulieu

on **August 29, 2013 at 1:58 am** said:

Major thanks for the blog post. Really getting excited about read more. Really Cool.



Olu

on [August 28, 2013 at 4:42 am](#) said:

When this module identifies a person, can it send out a specific signal, to indicate that particular person. e.g user 11 swiped, user 13 etc, so if one is interfacing it with an MCU, one can decode who swiped in particular



jhawley

on [September 23, 2013 at 2:38 pm](#) said:

Yes it does.

From the blog post above:

Lets check to see if that image is in the database on the fps. If it is, this command will return the ID number of it (a number between 0 and 199).

```
int id = fps.Identify1_N();
```



Daniel Beaulieu

on [August 19, 2013 at 6:13 am](#) said:

Getting excited about reading much more. Great article. Much many thanks. Really Great.