R307 Fingerprint Module

R307 Fingerprint Module consists of optical fingerprint sensor, high-speed DSP processor, high-performance fingerprint alignment algorithm, high-capacity FLASH chips and other hardware and software composition, stable performance, simple structure, with fingerprint entry, image processing, fingerprint matching, search and template storage and other functions.

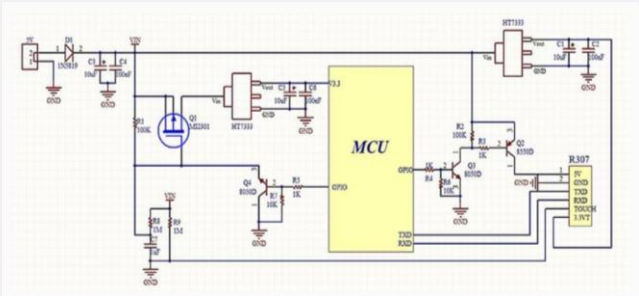
FEATURES:

- Perfect function: independent fingerprint collection, fingerprint registration, fingerprint comparison (1: 1) and fingerprint search (1: N) function.
- Small size: small size, no external DSP chip algorithm, has been integrated, easy to install, less fault.
- Ultra-low power consumption: low power consumption of the product as a whole, suitable for low-power requirements of the occasion.
- Anti-static ability: a strong anti-static ability, anti-static index reached 15KV above.
- Application development is simple: developers can provide control instructions, self-fingerprint application product development, without the need for professional knowledge of fingerprinting.
- Adjustable security level: suitable for different applications, security levels can be set by the user to adjust.
- Finger touch sensing signal output, low effective, sensing circuit standby current is very low, less than 5uA.

SPECIFICATIONS:

- Supply voltage: DC 4.2 ~ 6.0V
- Supply current: Working current: 50mA (typical) Peak current: 80mA
- Fingerprint image input time: <0.3 seconds
- Window area: 14x18 mm
- Matching method: Comparison method (1: 1)
- Search method (1: N)
- Characteristic file: 256 bytes
- Template file: 512 bytes
- Storage capacity: 1000 pieces
- Security Level: Five (from low to high: 1,2,3,4,5)
- Fake rate (FAR): <0.001%
- Refusal rate (FRR): <1.0%
- Search time: <1.0 seconds (1: 1000 hours, mean value)
- Host interface: UART \ USB1.1
- Communication baud rate (UART): (9600xN) bps Where N = 1 ~ 12 (default N = 6, ie 57600bps)
- Working environment: Temperature: -20 ℃ - +40 ℃ Relative humidity: 40% RH-85% RH (no condensation)
- Storage environment: Temperature: -40 ℃ - +85 ℃ Relative humidity: <85% H (no condensation)

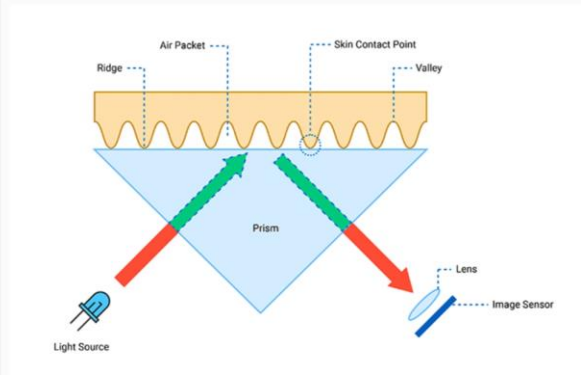
SCHEMATIC DIAGRAM:



INTERFACE DESCRIPTION:

- The R307 fingerprint module has two interface TTL UART and USB2.0, USB2.0 interface can be connected to the computer; RS232 interface is a TTL level, the default baud rate is 57600 , can be changed, refer to a communication protocol ; can And microcontroller, such as ARM, DSP and other serial devices with a connection, 3.3V 5V microcontroller can be connected directly.
- Needs to connect the computer level conversion, level conversion note , embodiments such as a MAX232 circuit.

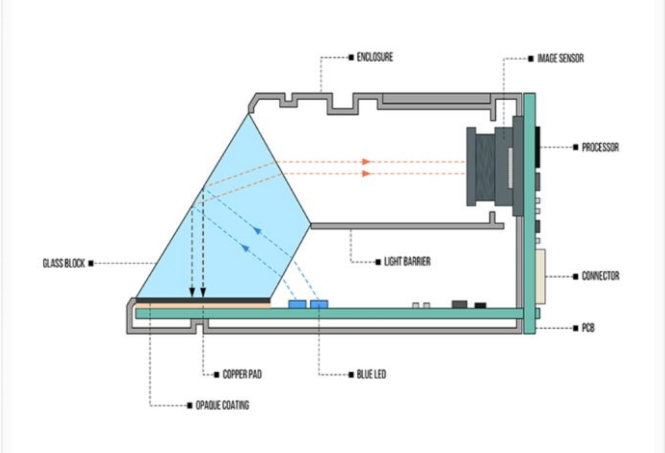
WORKING PRINCIPLE:



- An optical fingerprint scanner works based on the principle of Total Internal Reflection (TIR). In an optical fingerprint scanner, a glass prism is used to facilitate TIR. Light from an LED (usually blue color) is allowed to enter through one face of the prism at a certain angle for the TIR to occur. The reflected light exits the prism through the other face where a lens and an image sensor (essentially camera) are placed
- When there's no finger on the prism, the light will be completely reflected off from the surface, producing a plain image in the image sensor. When TIR occurs, a small amount of light leaked to the external medium and it is called the Evanescent Wave. Materials with different refractive indexes (RI) interact with the evanescent wave differently. When we touch a glass surface, only the ridges make good contact with it. The valleys remain separated from the surface by air packets. Our skin and air have different RIs and thus affect the evanescent field differently. This effect is called Frustrated Total Internal Reflection (FTIR). This effect alters the intensities of the internally reflected light and is

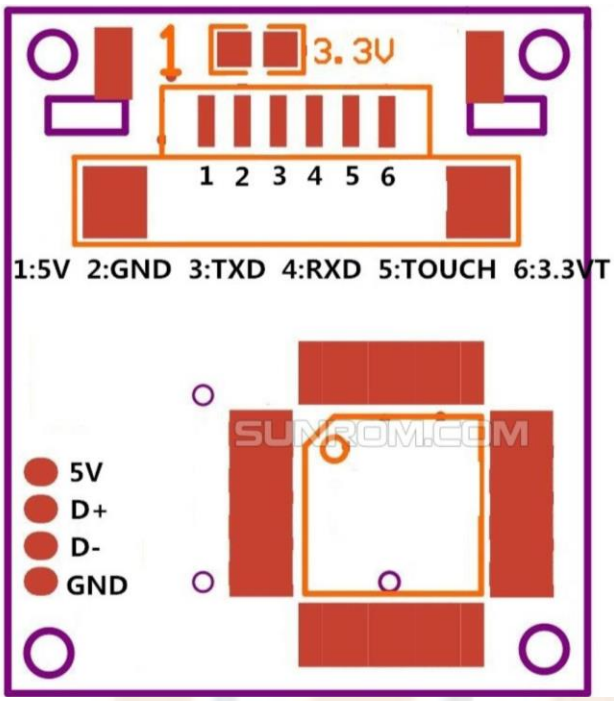
detected by the image sensor (see this image). The image sensor data is processed to produce a high contrast image which will be the digital version of the fingerprint.

- In capacitive sensors, which are more accurate and less bulky, there's no light involved. Instead, an array of capacitive sensors are arranged on the surface of the sensor and allowed to come in contact with the finger. The ridges and air packets affect the capacitive sensors differently. The data from the sensor array can be used to generate a digital image of the fingerprint.



- Above is a cross-sectional diagram to get understand the construction (illustrative only, not a physically exact one). Opening the module was easy; there are four Philips screws on the back. Unscrew them and it can remove the PCB. There are two PCBs; one arranged horizontally and one vertically (shown in washed green). These PCBs are connected by solder. The four blue LEDs and the touch sense pad are on the horizontal PCB. The vertical PCB has the image sensor, the processor and connector. When inserted, the touch sense pad comes in contact with the glass block above. The image sensor is soldered and glued. Strangely, I couldn't find any lens on it. May be it doesn't need one. The enclosure has an internal barrier to separate the light from the LEDs and the light coming out of the prism. On the bottom side of the prism a black epoxy is coated which gives a high-contrast background for the fingerprint image. To access the prism, just remove the cap on the front.

PIN FUNCTION:



Pin No	Pin Name	Description
1	5V	Regulated 5V DC
2	GND	Common Ground
3	TXD	Data output - Connect to MCU RX
4	RXD	Data Input - Connect to MCU TX
5	TOUCH	Active Low output when there is touch on sensor by finger
6	3.3V	Use this wire to give 3.3V to sensor instead of 5V

MEMORY AND REGISTORS:

1. Notepad - This is 512 bytes of the non-volatile flash memory. It is logically divided into 16 pages with 32 bytes each. Instructions GR_WriteNotepad and GR_ReadNotepad can be used to access this memory. When writing a page, it is taken as a whole, and the contents are replaced.

2. Image Buffer - Image buffer is used to store a BMP image of size 256 x 288, each pixel occupying a byte. This buffer is part of the RAM and the contents are lost when power is lost.

3. Character File Buffer - A character file is a processed high contrast image of a fingerprint. Two-character files from two consecutive scans are combined to form a template file which is the final version of the fingerprint that is stored in the fingerprint library (not to be confused with the Arduino library. Fingerprint library is the memory used to store up to 1000 fingerprints). The two-character file buffers are CharBuffer1 and CharBuffer2 each with size of 512 bytes.

4. Fingerprint Library - This is a section of the flash memory where 1000 fingerprint templates can be stored. Templates are arranged sequentially with numbering from 0 to N-1 (The manual says 0-N) where N is the capacity of the library determined by the size of the flash memory. There are instructions to store, process and delete templates from this memory. They will be explained later.

5. System Configuration Register - This is a 16-bytes long register bank containing operating parameters and status. Except the device address which takes up 4 bytes, rest of the parameters are 2 bytes (a word) in length. The command ReadSysPara can be used to read, and command SetSysPara can be used to write this register bank.

Name	**Description**	**Offset(word)**	**Size(word)**
Status Register	Contents of system status register	0	1
System Identifier Code	Fixed value: 0x0009	1	1
Library Size	Fingerprint library size	2	1
Security Level	Security level (1, 2, 3, 4, 5)	3	1
Device Address	32-bit device address	4	2
Data Packet Size	Size code (0, 1, 2, 3)	5	1
Baud Multiplier	N (baud = 9600*N bps)	6	1

The Status Register indicates the current operation status of the module, and comprises of the following

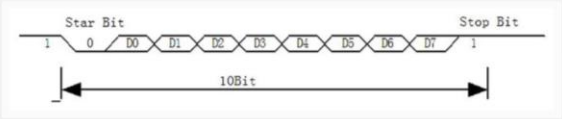
Bit Number	15-4	3	2	1	0
Description	Reserved	ImgBufStat	PWD	Pass	Busy

- where Busy = 1: system is executing commands; 0: system is free, Pass = 1: found a matching fingerprint; 0: fingerprint not found, PWD = 1: handshaking password verified; 0: password not verified, ImgBufStatus = 1: image buffer contains valid image; 0: image buffer does not have a valid image.
- System Identifier Code is a fixed value that determines the type of module. Its value is 0x0009 for R307.
- Library Size is the number of templates that can be stored in the module. The maximum value for this parameter is 1000 for R307.
- Security Value determines the threshold for fingerprint searching and matching. Its value can be from 1-5. When it is 1, the FAR (False Acceptance Rate) is the highest and FRR (False Recognition Rate) is the lowest. FAR is simply the number that determines how likely the module will identify a weakly matched fingerprint as positive. FRR is how likely the module will identify a wrong fingerprint as negative. At level 5, the FAR is the lowest and FFR is the highest. In this setting, it can be difficult to match your fingerprint.
- Device Address is a 32-bit value that holds the address of the module. The correct address is needed to communicate with the module. If you don't send the correct address, the module won't execute any commands. Device address can be modified with the command SetAddr. The factory programmed address is 0xFFFFFFFF. There's no methods specified in the manual to reset the address to default, so keep the address safe if you ever change it.
- Data Packet Size determines the maximum length of data content in a single packet. Its value can be 0-3 where 0 = 32 bytes, 1 = 64 bytes, 2 = 128 bytes and 3 = 256 bytes.
- Baud Multiplier sets the UART communication speed of the module. The minimum speed is 9600bps and can be set to up to 12 times of that which is 115200bps. The multiplier value N can be from 1-12 and the effective speed will be (9600*N) bps. The default baudrate is 57600bps.

COMMUNICATION PROTOCOL:

- Both UART and USB interfaces use a common serial communication protocol based on a packet format (the manual refers packets as "packages"). All data and commands are to

be sent as data packets and all responses from the module will also be packets. So we need to frame data and commands as packets before sending out, and must extract data from response packets. The UART frame format is 10 bit with 1 start bit, 1 stop bit and 8 data bits.



R307 UART frame format

The packet format is as follows (length in bytes is shown in brackets)

Header (2)	**Address (4)**	**Packet Identifier (1)**	**Packet Length (2)**	**Packet Content (Instruction/Data/Parameter)**	**Checksum (2)**

A packet is a group of many such bytes (or frames). Let's see the definitions of each part.

1.Header: This indicates the start of a packet. It has to be the fixed value 0xEF01. It is 2 bytes long and the high byte is always transferred first.

2.Address: This is the 32-bit address of the scanner module. The module will accept instructions only if the address we are sending matches the address stored in the module. The default address is 0xFFFFFFFF and can be modified with SetAddr instruction.

3.Packet Identifier: This determines what type of packet we're sending or receiving. It is 1 byte long and depending on the value the packet types can be,

0x01: The packet contains a command.

0x02: Data packet. A data packet must be followed by a command packet or acknowledge packet.

0x07: Acknowledge packet. It is sent by the module in response to a command.

0x08: End of data transfer packet. When we send large volume data such as an image, the data transfer will be terminated by this packet.

4.Packet Length: This is the total length of Packet Content and Checksum in bytes. Maximum length is 256 bytes and high byte is transferred first.

5.Packet Content: This can be data/command/parameters etc. of varying length. The Packet Length is the value that specifies the length of the data here in bytes.

6.Checksum: This is the arithmetic sum of all bytes in Packet Identifier, Packet Length and Packet Content. Overflowing bits are ignored. High byte is always transferred first.

- In order make the fingerprint scanner work, we must send instructions or commands in the form of packets. Each instruction is simply a 1-byte code that we must include in the packet. The module responds to each instruction with an acknowledgment packet that describes the result and status of command execution. Each instruction has a set of expected response codes found in the ACK packet that are called confirmation codes. Instructions and their byte codes are grouped according to their functions as shown below,

code	identifier	Description	Code	Identifier	Description
01H	GenImg	Collect finger image	0DH	Empty	to empty the library
02H	Img2Tz	To generate character file from image	0EH	SetSysPara	To set system Parameter
03H	Match	Carry out precise matching of two templates;	0FH	ReadSysPara	To read system Parameter
04H	Serach	Search the finger library	12H	SetPwd	To set password
05H	RegModel	To combine character files and generate template	13H	VfyPwd	To verify password
06H	Store	To store template;	14H	GetRandomCode	to get random code
07H	LoadChar	to read/load template	15H	SetAdder	To set device address
08H	UpChar	to upload template	17H	Control	Port control
09H	DownChr	to download template	18H	WriteNotepad	to write note pad
0AH	UpImage	To upload image	19H	ReadNotepad	To read note pad
0BH	DownImage	To download image	1BH	HiSpeedSearch	Search the library fastly
0CH	DeletChar	to delete tempates	1DH	TempleteNum	To read finger template numbers

Following is the list of confirmation codes.

- 0x00 - Command execution complete
- 0x01 - Error when receiving data package
- 0x02 - No finger on the sensor
- 0x03 - Failed to enroll the finger
- 0x04 - Failed to generate character file due to the over-disorderly fingerprint image
- 0x05 - Failed to generate character file due to the over-wet fingerprint image
- 0x06 - Failed to generate character file due to the over-disorderly fingerprint image
- 0x07 - Failed to generate character file due to lack of character point or over-smallness of fingerprint image
- 0x08 - Finger doesn't match
- 0x09 - Failed to find a matching finger
- 0x0A - Failed to combine the character files
- 0x0B - Addressing PageID is beyond the finger library
- 0x0C - Error when reading template from library or the template is invalid
- 0x0D - Error when uploading template
- 0x0E - Module can't receive the following data packages
- 0x0F - Error when uploading image
- 0x10 - Failed to delete the template
- 0x11 - Failed to clear finger library
- 0x13 - Wrong password
- 0x15 - Failed to generate the image
- 0x18 - Error when writing flash
- 0x19 - No definition error
- 0x21 - Password not verified
- 0x1A - Invalid register number
- 0x1B - Incorrect configuration of register
- 0x1C - Wrong notepad page number
- 0x1D - Failed to operate the communication port
- 0x41 - No finger on sensor when add fingerprint on second time
- 0x42 - Failed to enroll the finger for second fingerprint scan

- 0x43 - Failed to generate character file due to lack of character point or over-smallness of fingerprint image for second fingerprint scan
- 0x44 - Failed to generate character file due to the over-disorderly fingerprint image for second fingerprint scan
- 0x45 - Duplicate fingerprint
- Others - System reserved

OUTER DIMENSION: