

بسمه تعالی

آموزش راه اندازی و کار نویسی ماژول اثر انگشت R308

با سلام خدمت همه دوستان عزیز

این آموزش رو با توجه به کمبود منابع برای این ماژول تهیه کردم ، امیدوارم بتونه خلع موجود در این زمینه رو مرتفع کنه

نمایی از ماژول R308 :



این ماژول دارای مشخصات زیر هست :

Supply voltage: DC 4.2 ~ 6.0V

Supply Current:

Working Current: 110mA (typ)

Peak Current: 140mA

Fingerprint image time: <0.3 seconds

Window Size: 15×19 mm

Matching:

Than on the way (1: 1)

Search mode (1: N)

Template File: 512 bytes

Storage capacity: 500

Security level: five (from low to high: 1,2,3,4,5)

False Accept Rate (FAR): <0.001%

False Reject Rate (FRR): <1.0%

Search time: <1.0 seconds (1: 1000, the mean)

PC interface: RS232 (TTL logic level)

Baud rate (UART) :(9600×N) bps where N = 1 ~ 12 (default value N = 6, ie 57600bps)

Work Environment:

Temperature: -20 °C - + 40 °C

Relative Humidity: 40% RH-85% RH (non-condensing)

Storage environment:

Temperature: -40 °C - + 85 °C

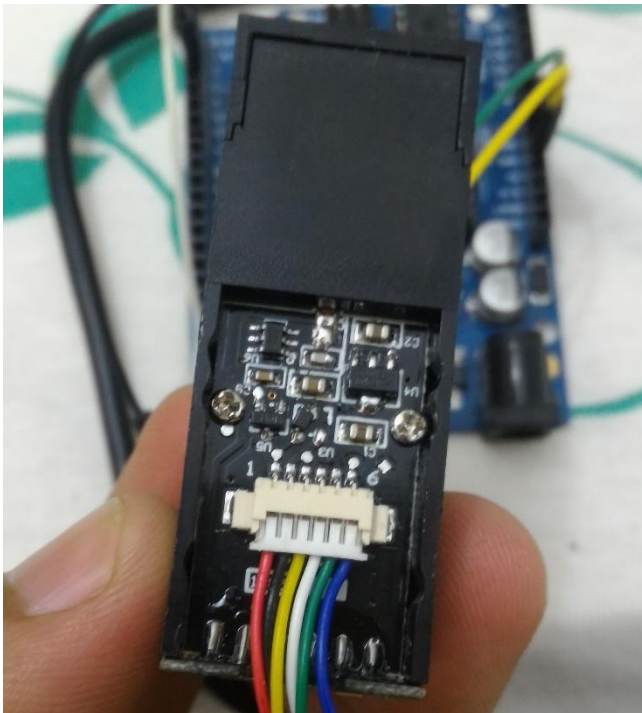
Relative humidity: <85% H (non-condensing)

در این ماژول اوج شلختگی رو شاهد خواهید بود ، سیم های رنگی استاندارد نبوده ، با کتابخانه استاندارد آردوینو هم دچار ارور می شود و مطالبی که همه ی آنها را در ادامه برای شما ذکر خواهم کرد .

وسایل مورد نیاز :

۱ – ماژول R308

۲ – Arduino MEGA 2560



خوب برای شروع کابلی که همراه ماژول عرضه

می شه رو زیر ماژول درون سوکت قرار بدید .

(مطابق شکل)

این نکته رو متذکر می شم که رنگ سیم ها به

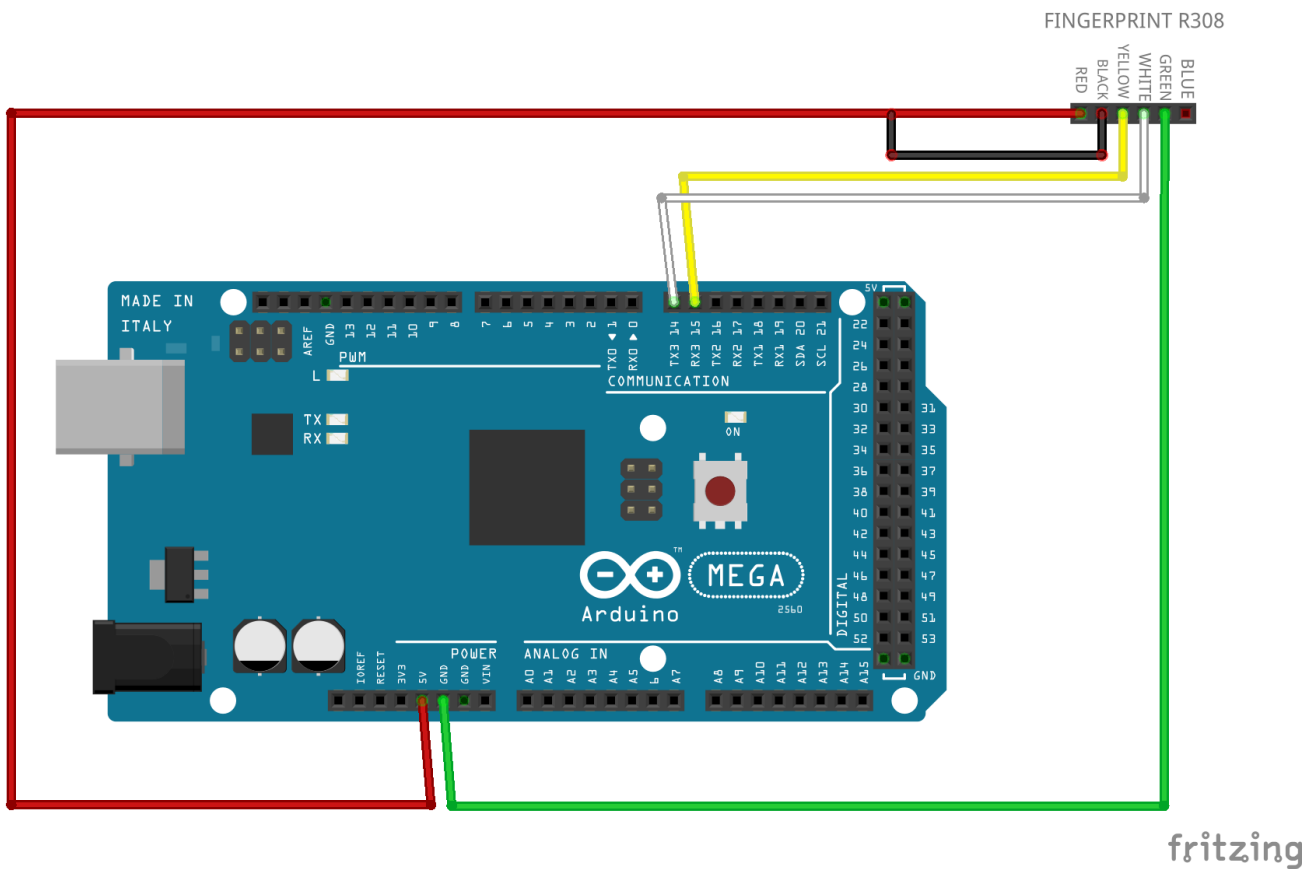
هیچ عنوان استاندارد نیستند . (یکی از این شلختگی

ها)

پایه ها و نوع کاربردهشون در جدول زیر آوردم

توضیحات	کاربرد	رنگ سیم
به +5 ولت آردوینو وصل گردد	تغذیه اثر انگشت	قرمز
به +5 ولت آردوینو وصل گردد	تغذیه ماژول	سیاه
به RXD آردوینو وصل گردد	TXD	زرد
به TXD آردوینو وصل گردد	RXD	سفید
به زمین متصل گردد	GND	سبز
بسته به نیاز می تونید استفاده کنید	سیگنال تشخیص لمس	آبی

نقشه اتصالات :



- نکته : توجه داشته باشد که آردوینو مگا ۲۵۶۰ دارای چهار درگاه سریال بوده ، درگاه صفر سیستمی بوده و بیشتر برای ارتباط برد با کامپیوتر مورد استفاده قرار می گیرد ، از این رو شما می توانید یکی از درگاه های سریال به غیر از درگاه صفر را برای ارتباط با ماژول انتخاب نمایید . البته شما می توانید از درگاه صفر استفاده نمایید ، به شرطی که نخواهید چیزی را بر روی کامپیوتر مانیتور نمایید .
- در برنامه ی نوشته شده توسط این حقیر از درگاه RXD3 و TXD3 برای ارتباط با ماژول استفاده گشته است .
- ولتاژ تغذیه ی ماژول هم طبق مشخصات دیتا شیت بین ۴٫۲ الی ۶ ولت است
- سیم آبی برای تشخیص لمس است ، طبق تست هایی که بنده به جای آوردم ، این پایه را به مبدل آنالوگ به دیجیتال (A0) متصل کردم ، مقدار مبدل بعد از برخورد انگشت با صفحه برابر با صفر خواهد شد ، بنابراین شما می توانید توسط این پایه طوری برنامه

نویسی نمایید که در صورت برخورد دست با صفحه ، عملیات شناسایی صورت پذیرد که این خود سبب صرفه جویی در مصرف انرژی می گردد .

به این مشخصات توجه نمایید

Power	DC 4.5V-6.0V	Interface	UART(TTL logical level)
Working current	Typical: 40mA Peak: 150mA	Matching Mode	1:1 and 1:N
Baud rate	(9600*N)bps, N=1~12 (default N=6)	Upper monitor	UART(3.3V TTL)
Image acquiring time	<0.5s	Template size	512 bytes
Storage capacity	120,240,500	Security level	5 (1, 2, 3, 4, 5(highest))
FAR	<0.001%	FRR	<0.5%
Average searching time	< 1s (1:1000)	Window dimension	18mm*22mm
Working environment	Temp: -20℃- +40℃	Storage environment	Temp: -40℃- +85℃
	RH: 10%-85%		RH: <90%
Outline Dimention	Size:55.5*21*20.5mm		

در این مشخصات عملا ذکر شده باند ریت دستگاه برابر با $9600 \times N$ بوده که به صورت پیشفرض N آن برابر با ۶ است ، پس اگر N را برابر با ۶ در نظر بگیریم ، حاصلضرب ۹۶۰۰ در ۶ برابر با ۵۷۶۰۰ خواهد بود ، اما در عمل اینگونه نیست ، شما برای ارتباط با ماژول باید باند ریت های زیادی را تست نمایید ، تا ببینید در کدام باند ریت ماژول به شما به درستی پاسخ می دهد .

ماژول بنده در زمان تست روی ۹۶۰۰ تنظیم شده بود ، شما هم از ۹۶۰۰ تست کنید تا ۱۱۵۲۰۰ ، ببینید کدام يك به شما جواب می دهد .

بررسی کامندها و نحوه ارتباط با ماژول :

```
void setup() {
```

```
    Serial.begin(57600);  
    Serial3.begin(57600);  
    delay(3000);
```

من برای مانیتورینگ اطلاعات از سریال صفر استفاده کردم و باند ریت اونو ۵۷۶۰۰ استفاده

کردم و و از درگاه سوم برای ارتباط با ماژول استفاده کردم ، قبلا هم گفتم که ماژول نرخ

باند ریت مشخصی نداره و باید خودتون مقدارشو با تست پیدا کنید (برای من به صورت پیش فرض روی ۹۶۰۰ تنظیم شده بود)

اما delay(1000) برای چیست ؟

طبق دیتا شیت ، پس از روشن شدن ماژول حدود ۵۰۰ میلی ثانیه طول می کشد که سنسور دستورات را قبول کند . به این مساله حتما دقت نمایید ، بعد از این تاخیر می توانید به راحتی از دستورات استفاده کنید .

برای شروع کار با ماژول لازمه که قبل از هر چیز رمز عبور رو تایید نمایید .

8. Verify password VfyPwd تایید رمز عبور

Description: Verify Module's handshaking password. (Refer to 4.6 for details)

Input Parameter: PassWord (4 bytes)

Return Parameter: Confirmation code (1 byte)

Instruction code: 13H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	4 byte	2 bytes
Header	Module address	Package identifier		Instruction code	Password	Checksum
0xEF01	xxxx	01H	07H	13H	PassWord	sum

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package Length	Confirmation code	Checksum
0xEF01	xxxx	07H	03H	xxH	sum

Note: Confirmation code = 00H: Correct password;

Confirmation code = 01H: error when receiving package;

Confirmation code = 13H: Wrong password;

در اینجا نحوه ارسال دستور به ماژول و فرمت داده ی برگشتی مشخص شده ، يك مورد رو کامل توضیح می دم ، بقیه موارد از همین روش تبعیت می کنه .

Command (or instruction) package format:

فرمت ارسال بسته ی دستورالعمل

Header

هدر ۲ بایت هست وهمیشه ثابت و در ابتدای فرمت دستور به کار می ره

دوبایت رو به این فرم ارسال می کنیم

```
Serial3.write(0xEF); Serial3.write(0x01);
```

Module address

۴ بایته که شامل آدرس ماژول می شود

- آدرس ماژول به طور پیش فرض 0xFFFFFFFF می باشد .

۴ بایت را به این فرم ارسال می کنیم

```
Serial3.write(0xFF);
```

```
Serial3.write(0xFF);
```

```
Serial3.write(0xFF);
```

```
Serial3.write(0xFF);
```

Package identifier يك بایت بوده و مقدار آن توسط دیتا شیت به ما داده شده است (0x01)

يك بایت را اینگونه ارسال می کنیم

```
Serial3.write(0x01);
```

بعدک مقدار 07H است که توسط خود دیتا شیت داده شده و ۲ بایت است .

```
Serial3.write(0x00); Serial3.write(0x07);
```

Instruction Code يك بایت بوده و مقدار آن 13H بوده و توسط خود دیتا شیت داده شده است .

```
Serial3.write(0x13);
```

PassWord چهار بایت بوده و مقدار پیش فرض آن برابر با 0X00000000 می باشد .

```
Serial3.write(0x00);
```

```
Serial3.write(0x00);
```

```
Serial3.write(0x00);
```

```
Serial3.write(0x00);
```

Checksum دو بایت می باشد ، که باید آن را محاسبه نماییم

مقادیر موجود را همیشه از Package identifier تا قبل از Checksum با هم جمع و حاصل را در

SUM مقدار دهی می کنیم

شروع می کنیم

	0	0	0	0	1	1	1	
Package identifier	0	0	0	0	0	0	0	1
	0	0	0	0	0	1	1	1
Instruction Code	0	0	0	1	0	0	1	1
PassWord	0	0	0	0	0	0	0	0
	0	0	0	1	1	0	1	1
								حاصل

حاصل بدست آمده برابر با 0X1B شد ، آن را به عنوان مقدار SUM در نظر می گیریم .

```
Serial3.write(0x00); Serial3.write(0x1B);
```

در صورت اینکه همه چیز درست باشد خروجی به فرم زیر خواهد بود

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package Length	Confirmation code	Checksum
0xEF01	xxxx	07H	03H	xxH	sum

Note: Confirmation code = 00H: Correct password;

Confirmation code = 01H: error when receiving package;

Confirmation code = 13H: Wrong password;

آن چیز که برای ما مهم است مقدار Confirmation code است

اگر برابر با 00 بود رمز داده شده درست است

اگر برابر با 01 بود اروری در دریافت اطلاعات پیش آمده

و اگر برابر با 13 بود یعنی پسورد وارد شده صحیح نمی باشد .

پس توجه داشته باشد برای تغییر مقادیر ماژول لازم است که رمز عبور را وارد و تایید کنیم

قالب توضیحات بالا می شود :

```
void setup() {
    Serial.begin(57600);
    Serial3.begin(57600);
    delay(3000);
    // -----Verify password-----
    Serial3.write(0xEF); Serial3.write(0x01);
    Serial3.write(0xFF); Serial3.write(0xFF); Serial3.write(0xFF);
    Serial3.write(0xFF);
    Serial3.write(0x01);
    Serial3.write(0x00); Serial3.write(0x07);
    Serial3.write(0x13);
    Serial3.write(0x00); Serial3.write(0x00); Serial3.write(0x00);
    Serial3.write(0x00);
    Serial3.write(0x00); Serial3.write(0x1B);
}

void loop() {
}
```

حال بعد از تایید رمز عبور می توانید تغییرات را اجاد نمایید

تغییر نرخ سریال (BAUD RATE) :

Set module system's basic parameter SetSysPara

Description: Operation parameter settings. (Refer to 4.4 for more information)

Input Parameter: Parameter number;

Return Parameter: Confirmation code (1 byte)

Instruction code: 0eH

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1byte	1byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Parameter number	Contents	Checksum
0xEF01	Xxxx	01H	05H	0eH	4/5/6	xx	sum

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
0xEF01	Xxxx	07H	03H	xxH	Sum

Note: Confirmation code=00H: parameter setting complete;

Confirmation code=01H: error when receiving package;

آنچه در اینجا مهم است یکی Parameter number و Contents است .

اگر می خواهید نرخ سریال را تغییر دهید Parameter number را برابر با چهار قرار داده و مقدار N را در Contents قرار دهید ، N چیست ، همان مقداری است که در ۹۶۰۰ ضرب شده و نرخ باند ریت را مشخص می نماید . (N می تواند مقداری بین ۱ تا ۱۲ داشته باشد)

```
void setup() {

    Serial.begin(57600);
    Serial3.begin(57600);
    delay(3000);
    // -----Verify password-----
    Serial3.write(0xEF); Serial3.write(0x01);
    Serial3.write(0xFF); Serial3.write(0xFF); Serial3.write(0xFF);
    Serial3.write(0xFF);
    Serial3.write(0x01);
    Serial3.write(0x00); Serial3.write(0x07);
    Serial3.write(0x13);
    Serial3.write(0x00); Serial3.write(0x00); Serial3.write(0x00);
    Serial3.write(0x00);
    Serial3.write(0x00); Serial3.write(0x1B);
    delay(1000);
    //-----SetSysPara-----
    Serial3.write(0xEF); Serial3.write(0x01);
    Serial3.write(0xFF); Serial3.write(0xFF);
    Serial3.write(0xFF);Serial3.write(0xFF);
    Serial3.write(0x01);
    Serial3.write(0x00); Serial3.write(0x05);
    Serial3.write(0x0e);
    Serial3.write(0x04); //Baud rate control
    Serial3.write(0x06); //57600
    Serial3.write(0x00); Serial3.write(0x1E);
    delay(1000);

}

void loop() {

}
```

در برنامه ک بالا من نرخ سریال را ۵۷۶۰۰ قرار دادم ، (N برابر ۶)

حال توضیحات مربوط به ذخیره اثر انگشت :

در برنامه ای که من در پایان نوشته هایم خواهم گذاشت ، برای هر کدام از کارهایی که برای

ثبت و ذخیره ی انگشتان قرار می گیرد تابعی تعریف نموده ام ، که آنها را مورد بررسی قرار

می دهیم .

```
Verifypassword();
delay(1000);
GetImage();
delay(1000);
Img2Tz(0X01);
delay(1000);
GetImage();
delay(1000);
```

```
Img2Tz (0X02);
delay(1000);
```

ابتدا تابع Verifypassword() را فراخوانی می کنیم

تابع :

```
void Verifypassword()
{
    // -----Verify password-----
    Serial3.write(0xEF); Serial3.write(0x01);
    Serial3.write(0xFF); Serial3.write(0xFF); Serial3.write(0xFF);
    Serial3.write(0xFF);
    Serial3.write(0x01);
    Serial3.write(0x00); Serial3.write(0x07);
    Serial3.write(0x13);
    Serial3.write(0x00); Serial3.write(0x00); Serial3.write(0x00);
    Serial3.write(0x00);
    Serial3.write(0x00); Serial3.write(0x1B);
}
```

همان دستورات تایید رمز عبور است

سپس با ۱ ثانیه تاخیر تابع GetImage() را فراخوانی می کنیم

تابع :

```
void GetImage()
{
    //-----GetImage1-----
    Serial3.write(0xEF); Serial3.write(0x01);
    Serial3.write(0xFF); Serial3.write(0xFF); Serial3.write(0xFF);
    Serial3.write(0xFF);
    Serial3.write(0x01);
    Serial3.write(0x00); Serial3.write(0x03);
    Serial3.write(0x01);
    Serial3.write(0x00); Serial3.write(0x05);
}
```

وظیفه ی این تابع جمع آوری تصویر انگشت است .

1. To collect finger image GetImage برای جمع آوری انگشت تصویر

Description: detecting finger and store the detected finger image in ImageBuffer while returning successfull confirmation code; If there is no finger, returned confirmation code would be "can't detect finger".

Input Parameter: none

Return Parameter: Confirmation code (1 byte)

Instuction code: 01H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Checksum
0xEF01	Xxxx	01H	03H	01H	05H

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
0xEF01	Xxxx	07H	03H	xxH	Sum

Note: Confirmation code=00H: finger collection success;

Confirmation code=01H: error when receiving package;

Confirmation code=02H: can't detect finger;

Confirmation code=03H: fail to collect finger;

پس از اعمال این دستور چراغ آبی رنگ روشن شده و به دنبال انگشت می گردد ، دقت داشت باشید زمانی که این دستور اجرا شود باید انگشت شما روی سنسور باشد و اگر نه چیزی را تشخیص نمی دهد . و چراغ خاموش می شود ،

در پکیج برگشتی هم آنچه که مهم است مقدار Confirmation code است . اگر 00 باشد عملیات جمع آوری انگشت موفقیت آمیز بوده ، اگر 01 باشد ارور در دریافت اطلاعات دریافتی داشته ، اگر 02 باشد یعنی نمی تواند انگشت را شناسایی نماید و 03 هم یعنی شکست عملیات جمع آوری .

مرحله ی بعد تابع ; Img2Tz (0X01) است
تابع :

```
void Img2Tz (byte id)
{
    // -----Img2Tz-----
    Serial3.write(0xEF); Serial3.write(0x01);
    Serial3.write(0xFF); Serial3.write(0xFF); Serial3.write(0xFF);
    Serial3.write(0xFF);
    Serial3.write(0x01);
    Serial3.write(0x00); Serial3.write(0x04);
    Serial3.write(0x02);
    Serial3.write(id); //ID
    Serial3.write(0x00); Serial3.write(0x07 + id);
}
```

این تابع از ما يك ID طلب می کند ، به این نکته توجه داشته باشید تمام عملیات تبدیل و تهیه قالب تمامی انگشت ها در id ها يك و دو صورت می گیرد و از سایر ID ها جهت این امر نمی توان استفاده نمود .

ID دریافتی توسط این تابع 0X01 یا 0X02 می باشد

(Img2Tz (0X01) با مقدار 0X01 تصویر انگشت دریافتی را به کارکتر تبدیل نموده و در آک دی شماره ی يك ذخیره می نماید .

2. To generate character file from image Genchar برای تولید فایل کاراکتر از تصویر

Description: to generate character file from the original finger image in ImageBuffer and store the file in CharBuffer1 or CharBuffer2.

Input Parameter: BufferID (character file buffer number)

Return Parameter: Confirmation code (1 byte)

Instuction code: 02H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Buffer number	Checksum
0xEF01	xxxx	01H	04H	02H	BufferID	sum

Note: BufferID of CharBuffer1 and CharBuffer2 are 1h and 2h respectively. Other values (except 1h, 2h) would be processed as CharBuffer2.

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
---------	--------	--------	---------	--------	---------



Header	Module address	Package identifier	Package length	Confirmation code	Checksum
0xEF01	xxxx	07H	03H	XxH	sum

Note: Confirmation code=00H: generate character file complete;

Confirmation code=01H: error when receiving package;

Confirmation code=06H: fail to generate character file due to the over-disorderly fingerprint image;

Confirmation code=07H: fail to generate character file due to lackness of character point or over-smallness of fingerprint image;

Confirmation code=15H: fail to generate the image for the lackness of valid primary image;

تابع بعدی باز هم دریافت اثر انگشت است با `GetImage()` و سپس دوباره با تابع `Img2Tz (0X02)` که این بار با `id` ، `0X02` می باشد و کارکتر تولید شده را در آک دی `0X02` قرار می دهد .

برای ادامه ی کار مشکلی که توک کار بهش بر خورد کردم این بود که از توابع به بعد ، جواب ارسالی از ماژول دریافت نمی شه ، برای حل مشکل کافیه یک بار درگاه سریال را باز و بسته کنید

```
Serial3.end();
Serial3.begin(57600);
```

حال نوبت به ارسال توابع می رسد که از `ID1` و `ID2` یک قالب درست کنیم از این رو از تابع `RegModel()` استفاده می کنیم .

تابع :

```
void RegModel()
{
    //-----RegModel-----
    Serial3.write(0xEF); Serial3.write(0x01);
    Serial3.write(0xFF); Serial3.write(0xFF); Serial3.write(0xFF);
    Serial3.write(0xFF);
    Serial3.write(0x01);
    Serial3.write(0x00); Serial3.write(0x03);
    Serial3.write(0x05);
    Serial3.write(0x00); Serial3.write(0x09);
}
```

}

4. To generate template RegModel1 برای تولید قالب

Description: To combine information of character files from CharBuffer1 and CharBuffer2 and generate a template which is stroed back in both CharBuffer1 and CharBuffer2.

Input Parameter: none

6

www.hzgrow.com



Return Parameter: Confirmation code (1 byte)

Instuction code: 05H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Checksum
0xEF01	xxxx	01H	03H	05H	09H

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
0xEF01	xxxx	07H	03H	xxH	sum

Note: Confirmation code=00H: operation success;

Confirmation code=01H: error when receiving package;

Confirmation code=0aH: fail to combine the character files. That's, the character files don't belong to one finger.

اگر قالب با موفقیت درست شود مقدار 00 را در قسمت Confirmation code شاهد خواهید بود

فرمت خروجی : 000A FF01FFFFFFFF703

• در اطلاعات دریافت شده از طرق مازول و نمایش بر روی درگاه سریال نرم افزار آردوینو

مقادیر يك بایتی در صورتی كه ۴ بیت دوم آن صفر باشد آن را نمایش نمی دهد (07 را

به صورت 7 نمایش می دهد) اما اگر ۲ بایتی باشد 0 آن هم نمایش داده خواهد شد.

و در پایان كار تابع Store(0X01) را برای ذخیره الكو مورد استفاده قرار می دهیم . این تابع

نیز ID می پذیرد تا بتواند قالب را در آك دك مشخص شده ذخیره كند . آك دك می تواند از يك

شروع شود .

تابع :

```
void Store(byte id)
{
```

```
//-----Store-----
```

```

Serial3.write(0xEF); Serial3.write(0x01);
Serial3.write(0xFF); Serial3.write(0xFF); Serial3.write(0xFF);
Serial3.write(0xFF);
Serial3.write(0x01);
Serial3.write(0x00); Serial3.write(0x06);
Serial3.write(0x06);
Serial3.write(id); //ID
Serial3.write(0x00); Serial3.write(id); //PAGEID
Serial3.write(0x00); Serial3.write(0X0D + id + id);
}

```

5. To store template Store برای ذخیره الگو

Description: to store the template of specified buffer (Buffer1/Buffer2) at the designated location of Flash library.

Input Parameter: BufferID(buffer number), PageID (Flash location of the template, two bytes with high byte front and low byte behind)

Return Parameter: Confirmation code (1 byte)

Instruction code: 06H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	buffer number	Location number	Checksum
0xEF01	xxxx	01H	06H	06H	BufferID	PageID	sum

Note: BufferID of CharBuffer1 and CharBuffer2 are 1h and 2h respectively. Other values (except 1h, 2h) would be processed as CharBuffer2.

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
0xEF01	Xxxx	07H	03H	xxH	sum

Note: Confirmation code=00H: storage success;

Confirmation code=01H: error when receiving package;

Confirmation code=0bH: addressing PageID is beyond the finger library;

Confirmation code=18H: error when writing Flash.

دقت داشته باشید مقدار BufferID و PageID يك مقدار داشته باشند ، كه اين در تابع پيش بينى شده است . اگر قالب با موفقيت درست شود مقدار 00 را در قسمت Confirmation code شاهد خواهيد بود.

تا اينجاى كار به راحتى مى توانيد يك انگشت را ذخيره نماييد ، براى ذخيره هاى بيشتر كافيست كارهاى بالا را دوباره تكرر نماييد و فقط آى دى (Store(0X01 را تغيير دهيد (انگشت اول ID=0X01 براى انگشت دوم ID=0X02 و)

جستجو و مطابقت انگشت :

برای این کار از تابع استفاده Search() می کنیم .

```
void Search()
{
    //-----Verify password-----
    Serial3.write(0xEF); Serial3.write(0x01);
    Serial3.write(0xFF); Serial3.write(0xFF); Serial3.write(0xFF);
    Serial3.write(0xFF);
    Serial3.write(0x01);
    Serial3.write(0x00); Serial3.write(0x07);
    Serial3.write(0x13);
    Serial3.write(0x00); Serial3.write(0x00); Serial3.write(0x00);
    Serial3.write(0x00);
    Serial3.write(0x00); Serial3.write(0x1B);
    delay(2000);
    //-----GetImage-----
    Serial3.write(0xEF); Serial3.write(0x01);
    Serial3.write(0xFF); Serial3.write(0xFF); Serial3.write(0xFF);
    Serial3.write(0xFF);
    Serial3.write(0x01);
    Serial3.write(0x00); Serial3.write(0x03);
    Serial3.write(0x01);
    Serial3.write(0x00); Serial3.write(0x05);
    delay(1000);
    // -----Img2Tz-----
    Serial3.write(0xEF); Serial3.write(0x01);
    Serial3.write(0xFF); Serial3.write(0xFF); Serial3.write(0xFF);
    Serial3.write(0xFF);
    Serial3.write(0x01);
    Serial3.write(0x00); Serial3.write(0x04);
    Serial3.write(0x02);
    Serial3.write(0x01); //ID
    Serial3.write(0x00); Serial3.write(0x08);
    delay(1000);
    // -----Search-----
    Serial3.write(0xEF); Serial3.write(0x01);
    Serial3.write(0xFF); Serial3.write(0xFF); Serial3.write(0xFF);
    Serial3.write(0xFF);
    Serial3.write(0x01);
    Serial3.write(0x00); Serial3.write(0x08);
    Serial3.write(0x04);
    Serial3.write(0x01); //ID
    Serial3.write(0x00); Serial3.write(0x01);
    Serial3.write(0x00); Serial3.write(0x0F);
    Serial3.write(0x00); Serial3.write(0x1E);
    delay(1000);
}
```

دستورات در بخش قبل توضیح داده شدند

ابتدا رمز را تایید ، سپس دستورات مربوط به جمع آوری انگشت را اعمال و پس از تبدیل آن به

کارکتر آن را در ID1 قرار می دهیم .

اما بخش آخر که مربوط به Search است

بدین گونه عمل می کنیم

```
// -----Search-----
```



```

Serial3.write(0xEF); Serial3.write(0x01);
Serial3.write(0xFF); Serial3.write(0xFF); Serial3.write(0xFF);
Serial3.write(0xFF);
Serial3.write(0x01);
Serial3.write(0x00); Serial3.write(0x08);
Serial3.write(0x04);
Serial3.write(0x01); //ID
Serial3.write(0x00); Serial3.write(0x01);
Serial3.write(0x00); Serial3.write(0x0F);
Serial3.write(0x00); Serial3.write(0x1E);
delay(1000);

```

3. To search finger library

Search

برای جستجوی کتابخانه انگشت

Description: to search the whole finger library for the template that matches the one in CharBuffer1 or CharBuffer2. When found, PageID will be returned.

Input Parameter: BufferID, StartPage (searching start address), PageNum (searching numbers)

Return Parameter: Confirmation code (1 byte), PageID (matching templates location)

Instruction code: 04H

Command (or instruction) package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes	2 bytes	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	buffer number	Parameter	Parameter	Checksum
0xEF01	xxxx	01H	08H	04H	BufferID	StartPage	PageNum	sum

Note: BufferID of CharBuffer1 and CharBuffer2 are 1h and 2h respectively. Other values (except 1h, 2h) would be processed as CharBuffer2.

Acknowledge package format:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes	2 bytes	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	页码	得分	Checksum
0xEF01	xxxx	07H	7	xxH	PageID	MatchScore	sum

Note 1: Confirmation code=00H: found the matching finger;

Confirmation code=01H: error when receiving package;

Confirmation code=09H: No matching in the library (both the PageID and matching score are 0);

2: The instruction doesn't affect the contents of the buffers.

توضیحات بخش Command (or instruction) package format

BufferID : مقدار id را طوری قرار می دهیم که هنگام جستجو از آن id شروع به پیدا کردن انگشت نماید که در برنامه id را برابر با يك قرار دادیم تا از id يك شروع به جستجو کند .

StartPage : مقدار StartPage را طوری قرار می دهیم که هنگام جستجو از آن Page شروع به پیدا کردن انگشت نماید که در برنامه شروع صفحه را برابر با يك قرار دادیم تا از صفحه اول شروع به جستجو کند .

PageNum : مقدار PageNum همان انتهاى صفحه است ، به اینصورت که ابتداى صفحه را با (StartPage) و انتهاى صفحه رو با PageNum مشخص می کنیم

در برنامه ابتدای صفحه از 0x01 تا 0x0f تعیین شده است .

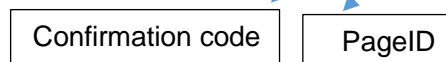
توضیحات بخش Acknowledge package format

آن چیز که مهم است PageID و Confirmation code است .

اگر انگشت مطابقت داشته باشد ، Confirmation code برابر با 00 خواهد شد و شماره ک id

مطابقت داشته با آن را بعد از این مقدار نمایش می دهد.

FF 01 FF FF FF FF 7 0 01



برنامه کامل :

در این برنامه برای ذخیره انگشت يك بار برنامه رو به فرم زیر کامپایل کنید

```
Verifypassword();  
delay(1000);  
GetImage();  
delay(1000);  
Img2Tz(0X01);  
delay(1000);  
GetImage();  
delay(1000);  
Img2Tz(0X02);  
delay(10000);  
//Search();  
//delay(1000);
```

بار دیگر برنامه رو به فرم زیر کامپایل کنید

```
//Verifypassword();  
//delay(1000);  
//GetImage();  
//delay(1000);  
//Img2Tz(0X01);  
//delay(1000);  
//GetImage();  
//delay(1000);  
//Img2Tz(0X02);  
//delay(10000);  
Search();  
delay(1000);
```

برنامه :

```
String f;  
byte a;  
int m = 1023;
```

```

int c = 0;
int OK = 0;

void Search();
void Store(byte id);
void RegModel();
void Img2Tz(byte id);
void GetImage();
void Verifypassword();
void setup() {

    Serial.begin(57600);
    Serial3.begin(57600);
    delay(3000);
    // -----Verify password-----
    //     Serial3.write(0xEF); Serial3.write(0x01);
    //     Serial3.write(0xFF); Serial3.write(0xFF); Serial3.write(0xFF);
    Serial3.write(0xFF);
    //     Serial3.write(0x01);
    //     Serial3.write(0x00); Serial3.write(0x07);
    //     Serial3.write(0x13);
    //     Serial3.write(0x00); Serial3.write(0x00); Serial3.write(0x00);
    Serial3.write(0x00);
    //     Serial3.write(0x00); Serial3.write(0x1B);
    //     delay(100);
    // //-----SetSysPara-----
    -----
    //     Serial3.write(0xEF); Serial3.write(0x01);
    //     Serial3.write(0xFF); Serial3.write(0xFF); Serial3.write(0xFF);
    Serial3.write(0xFF);
    //     Serial3.write(0x01);
    //     Serial3.write(0x00); Serial3.write(0x05);
    //     Serial3.write(0x0e);
    //     Serial3.write(0x04); //Baud rate control
    //     Serial3.write(0x06); //57600
    //     Serial3.write(0x00); Serial3.write(0x1E);
    //     delay(1000);
    // -----SetSysPara-----
    -----
    //     Serial3.write(0xEF); Serial3.write(0x01);
    //     Serial3.write(0xFF); Serial3.write(0xFF); Serial3.write(0xFF);
    Serial3.write(0xFF);
    //     Serial3.write(0x01);
    //     Serial3.write(0x00); Serial3.write(0x05);
    //     Serial3.write(0x0e);
    //     Serial3.write(0x06); //Data package length
    //     Serial3.write(0x00); //32
    //     Serial3.write(0x00); Serial3.write(0x1A); //256=1d 32=1A
    //     delay(1000);
    // #####
    #####
    Verifypassword();
    delay(1000);
    GetImage();
    delay(1000);
    Img2Tz(0X01);
    delay(1000);
    GetImage();
    delay(1000);
    Img2Tz(0X02);
    delay(1000);
}

```

```

    Search();
    delay(1000);
}

void loop() {

}

void serialEvent3() {
    if (Serial3.available() > 0) {
        c++;
        a = Serial3.read();
        f += String(a, HEX);
    }
    if (c == 12)
    {
        c = 0;

        Serial.print(f);
        if (f.substring(12, 16) == "0300")
        {
            OK++;
            Serial.println();
            Serial.println("OK");
            f = "";
            if (OK == 5)
            {
                Serial3.end();
                Serial3.begin(57600);
                RegModel();
                delay(1000);
                Store(0X02); // makane zakhire
                delay(1000);
                OK=0;
            }
        }
        else if (f.substring(12, 16) == "0320" || f.substring(12, 16) ==
"0370")
        {
            f = "";
            Serial.println();
            Serial.println("Error");
        } else if (f.substring(14, 17) == "900")
        {
            f = "";
            Serial.println();
            Serial.println("No matching");
        }
        else if (f.substring(14, 16) == "00")
        {
            f = "";
            Serial.println();
            Serial.println("matching");
        }
    }
}

void Verifypassword()
{
    // -----Verify password-----

```

```

    Serial3.write(0xEF); Serial3.write(0x01);
    Serial3.write(0xFF); Serial3.write(0xFF); Serial3.write(0xFF);
Serial3.write(0xFF);
    Serial3.write(0x01);
    Serial3.write(0x00); Serial3.write(0x07);
    Serial3.write(0x13);
    Serial3.write(0x00); Serial3.write(0x00); Serial3.write(0x00);
Serial3.write(0x00);
    Serial3.write(0x00); Serial3.write(0x1B);
}

void GetImage()
{
    //-----GetImage1-----
    Serial3.write(0xEF); Serial3.write(0x01);
    Serial3.write(0xFF); Serial3.write(0xFF); Serial3.write(0xFF);
Serial3.write(0xFF);
    Serial3.write(0x01);
    Serial3.write(0x00); Serial3.write(0x03);
    Serial3.write(0x01);
    Serial3.write(0x00); Serial3.write(0x05);
}

void Img2Tz(byte id)
{
    // -----Img2Tz-----
    Serial3.write(0xEF); Serial3.write(0x01);
    Serial3.write(0xFF); Serial3.write(0xFF); Serial3.write(0xFF);
Serial3.write(0xFF);
    Serial3.write(0x01);
    Serial3.write(0x00); Serial3.write(0x04);
    Serial3.write(0x02);
    Serial3.write(id); //ID
    Serial3.write(0x00); Serial3.write(0x07 + id);
}

void RegModel()
{
    //-----RegModel-----
    Serial3.write(0xEF); Serial3.write(0x01);
    Serial3.write(0xFF); Serial3.write(0xFF); Serial3.write(0xFF);
Serial3.write(0xFF);
    Serial3.write(0x01);
    Serial3.write(0x00); Serial3.write(0x03);
    Serial3.write(0x05);
    Serial3.write(0x00); Serial3.write(0x09);
}

void Store(byte id)
{
    //-----Store-----
    Serial3.write(0xEF); Serial3.write(0x01);
    Serial3.write(0xFF); Serial3.write(0xFF); Serial3.write(0xFF);
Serial3.write(0xFF);
    Serial3.write(0x01);
    Serial3.write(0x00); Serial3.write(0x06);
    Serial3.write(0x06);
    Serial3.write(id); //ID
    Serial3.write(0x00); Serial3.write(id); //PAGEID
    Serial3.write(0x00); Serial3.write(0x0D + id + id);
}

```

```

}

void Search()
{
    //-----Verify password-----
    Serial3.write(0xEF); Serial3.write(0x01);
    Serial3.write(0xFF); Serial3.write(0xFF); Serial3.write(0xFF);
    Serial3.write(0xFF);
    Serial3.write(0x01);
    Serial3.write(0x00); Serial3.write(0x07);
    Serial3.write(0x13);
    Serial3.write(0x00); Serial3.write(0x00); Serial3.write(0x00);
    Serial3.write(0x00);
    Serial3.write(0x00); Serial3.write(0x1B);
    delay(2000);
    //-----GetImage-----
    Serial3.write(0xEF); Serial3.write(0x01);
    Serial3.write(0xFF); Serial3.write(0xFF); Serial3.write(0xFF);
    Serial3.write(0xFF);
    Serial3.write(0x01);
    Serial3.write(0x00); Serial3.write(0x03);
    Serial3.write(0x01);
    Serial3.write(0x00); Serial3.write(0x05);
    delay(1000);
    // -----Img2Tz-----
    Serial3.write(0xEF); Serial3.write(0x01);
    Serial3.write(0xFF); Serial3.write(0xFF); Serial3.write(0xFF);
    Serial3.write(0xFF);
    Serial3.write(0x01);
    Serial3.write(0x00); Serial3.write(0x04);
    Serial3.write(0x02);
    Serial3.write(0x01); //ID
    Serial3.write(0x00); Serial3.write(0x08);
    delay(1000);
    // -----Search-----
    Serial3.write(0xEF); Serial3.write(0x01);
    Serial3.write(0xFF); Serial3.write(0xFF); Serial3.write(0xFF);
    Serial3.write(0xFF);
    Serial3.write(0x01);
    Serial3.write(0x00); Serial3.write(0x08);
    Serial3.write(0x04);
    Serial3.write(0x01); //ID
    Serial3.write(0x00); Serial3.write(0x01);
    Serial3.write(0x00); Serial3.write(0x0F);
    Serial3.write(0x00); Serial3.write(0x1E);
    delay(1000);
}

```

در پناه خدا موفق و پیروز باشید



GILELECTRONIC
THE BRIGHT FUTURE