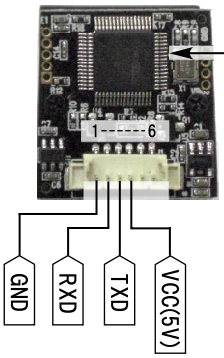
指紋を検出して一致するか照合

UART接続・光学式 指紋センサモジュール

☆UART接続/光学式指紋センサモジュール
☆マイコンから簡単なコマンドを送るだけで、指紋の取り込みや特徴抽出、一致検出や検索など指紋データ処理が簡単に行えます。
☆電源電圧：DC 5V

モジュール 信号線の配置



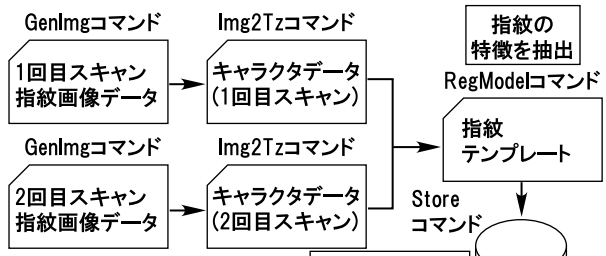
UART 通信パラメータ
◎通信速度：57600bps
◎データビット：8ビット
◎ストップビット：1ビット
◎パリティビット：なし

※5番/6番ピンは開放で使用してください。

指紋照合のしくみ

(1)指紋の特徴抽出と登録

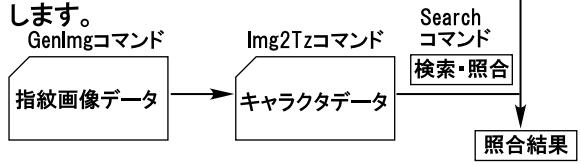
登録する指紋を2回スキャンして、データ変換の後特徴を抽出します



(2)指紋の一致検出(照合)

指紋テンプレートを保存(登録)

登録しておいた指紋テンプレートと、新たにスキャンした指紋データを照合します。



オーディオ・マイコン・メカトロ・電子パーツ
アジット
年中無休・営業時間：AM11:00~PM8:00
〒556-0005 大阪市浪速区日本橋4-6-7
[TEL] 06-6644-4555 / [FAX] 06-6644-1744
[HP] http://digit.kyohritsu.com
[Blog] http://blog.digit-parts.com [Twitter] @0666444555

通信のしかたとコマンドの詳細については、ZFM-20のデータシートを参照してください。

コマンドのフォーマット

指紋センサモジュールとマイコンとの間の通信は、下図のフォーマットのパケットを送受信することで行います。
※パケットのデータは全てバイナリです(16進で表記しています)
※複数バイトの項目は、上位のバイトを先に送ります。

◎モジュールへのコマンド（例：GenImgコマンド）

EF	01	FFFFFFFF	01	03	1D	0005
ヘッダ		モジュールアドレス(1) 4バイト	パケット種別(2) 1バイト	データ長 (3) 2バイト	コマンド 可変長	チェックサム (5) 2バイト

◎モジュールからの応答（例：指紋スキャン成功）

EF	01	FFFFFFFF	07	0003	00	000A
ヘッダ		モジュールアドレス(1) 4バイト	パケット種別(2) 1バイト	データ長 (3) 2バイト	結果コード/データ(4) 可変長	チェックサム(5) 2バイト

(1)モジュールアドレス（4バイト）
指紋センサのモジュールアドレスです。デフォルトのアドレスは「0xFFFFFFFF」です。(SetAddrコマンドで変更できます)

(2)パケット種別（1バイト）
送受信するパケットの種別を表します：

0x01：モジュールへのコマンド
0x02：データパケット
0x07：応答パケット(コマンド実行の結果コード)
0x08：データ終了を示すパケット

※ UpImageコマンドなど、データの転送を行うコマンドを実行した場合、応答パケットの後で、データのパケットの転送が行われます。データのパケットの転送が終わると、データ終了を示すパケットが送受信されます。

(3)データ長（2バイト）
データ/コマンドの長さに2(チェックサムの長さ)を足したものです。

(4)結果コード（1バイト）※主要なもののみ示します
モジュールがコマンドを実行した結果を表します。コマンドの実行に成功した場合は「00」を返してきます。※詳細については、ZFM-20のデータシートを参照してください。

	結果コード	概要
1	0x00	コマンドの実行が成功した
2	0x01	データパケットの受信に失敗した
3	0x02	センサに指が触れられていない
4	0x03	指紋の取り込みに失敗した
5	0x06	指紋画像が悪くてキャラクタデータが作成できなかった
6	0x07	指紋の特徴が抽出できずキャラクタデータが作成できなかった
7	0x08	照合の結果、指紋が一致しなかった
8	0x09	照合の結果、一致する指紋が見つからなかった
9	0x0A	テンプレートデータの作成に失敗した
10	0x0B	ページIDが指紋データライブラリの容量をオーバーした
11	0x0C	指紋データライブラリからテンプレートデータが読めなかった
13	0x0E	データパケットが受け付けられる状態でなかった
14	0x0F	指紋画像の送信に失敗した
17	0x15	有効な指紋画像がなく、指紋イメージが生成できなかった

(5)チェックサム（2バイト）
パケット種別、データ長、データ/コマンドの各バイトの値を足したものです。※オーバーフロー分は無視します。

コマンド表は次のページにあります。

コマンド一覧表

※コマンドとその実行結果の詳細については、ZFM-20のデータシートを参照してください。

指紋の取得/特徴抽出/照合関係のコマンド					
コマンド名	コマンド	パラメータ			機能
GenImg	0x01				指紋画像の取り込み
Img2Tz	0x02	バッファ番号(1/2)			指紋画像からキャラクタデータを作成
Match	0x03				バッファ上の指紋の一致を照合
Search	0x04	バッファ番号(1/2)	開始ページ番号(2バイト)	ページ数(2バイト)	フラッシュメモリから一致する指紋を探す
RegModel	0x05				キャラクタデータからテンプレートを作成
Store	0x06	バッファ番号(1/2)	ページ番号(2バイト)		テンプレートをフラッシュメモリに保存
LoadChar	0x07	バッファ番号(1/2)	ページ番号(2バイト)		フラッシュメモリからテンプレートを取り出す
UpChar	0x08	バッファ番号(1/2)			キャラクタデータをホストに送信
DownChar	0x09	バッファ番号(1/2)			ホストからキャラクタデータを受信
UpImage	0x0A				取り込んだ指紋画像をホストに送信
DownImage	0x0B				ホストから指紋画像を受信
DeletChar	0x0C	ページ番号(2バイト)	テンプレート数(2バイト)		フラッシュメモリからテンプレートを削除
Empty	0x0D				フラッシュメモリ上のテンプレートを全部削除
TemplateNum	0x1D	なし			有効なテンプレート番号の読み出し
システム関係のコマンド					
コマンド名	コマンド	パラメータ			機能
SetSysPara	0x0E	レジスタ番号(4,5,6)	設定データ		指定したレジスタを設定
ReadSysPara	0x0F	なし			システムレジスタの読み出し
GetRandomCode	0x14				4バイトの乱数を発生
SetAddr	0x15	アドレス(4バイト)			モジュールアドレスの変更
handshake	0x17	0x00			通信の確認
WriteNotepad	0x18	ページ番号(1～15)	書き込みデータ(32バイト)		内部メモ帳メモリへの書き込み
ReadNotepad	0x19	ページ番号(1～15)			内部メモ帳メモリからの読み出し(32バイト)

プログラム例

指紋センサモジュールにコマンドを送信する部分と、センサモジュールからの応答を受信する部分のプログラム例です。

コマンド送信部

```
void cmd_tx(unsigned char pkg_id,int len_data,unsigned
char *buf_data)
{
        int i,k;
        unsigned int icksum;
        unsigned char cbuf[8];
        /* header and module address */
        tx_buf[0]=0xEF;
        tx_buf[1]=0x01;
        tx_buf[2]=0xff;
        tx_buf[3]=0xff;
        tx_buf[4]=0xff;
        tx_buf[5]=0xff;
        icksum=0;
        tx_buf[6]=pkg_id;
        icksum=(unsigned int)tx_buf[6];
        tx_buf[7]=(unsigned int)(len_data+2)>>8;
        icksum=icksum+(unsigned int)tx_buf[7];
        tx_buf[8]=(unsigned int)(len_data+2)&0x00ff;
        icksum=icksum+(unsigned int)tx_buf[8];
        k=9;
        for(i=0;i<len_data;i++)
        {
                tx_buf[k]=*buf_data;
                icksum=icksum+(unsigned int)tx_buf[k];
                buf_data++;
                k++;
        }
        tx_buf[k]=icksum>>8;
        tx_buf[k+1]=icksum&0x00ff;
        tx_str(k+2,tx_buf);
}
```

応答受信部

```
int cmd_rx(unsigned char *buf_data)
{
        int nchar;
        int i;
        unsigned char moji;
        unsigned char cbuf[8];
        *buf_data=uart_rx();
         buf_data++;
        *buf_data=uart_rx();
        buf_data++;
        *buf_data=uart_rx();
        buf_data++;
        *buf_data=uart_rx();
        buf_data++;
        *buf_data=uart_rx();
        buf_data++;
        *buf_data=uart_rx();
        buf_data++;
        *buf_data=uart_rx();
        buf_data++;
        *buf_data=uart_rx(); buf_data++;
        moji=uart_rx();
        nchar=(moji<<8);
        *buf_data=moji; buf_data++;
        moji=uart_rx();
        nchar=nchar+moji;
        *buf_data=moji; buf_data++;
        for(i=0;i<nchar;i++)
        {
                *buf_data=uart_rx(); buf_data++;
        }
}
```