

HT580
通信プログラミング
マニュアル

ユニテック・ジャパン株式会社

2008年5月初版

概要

本マニュアルはHT580とPCの通信プログラム開発をするためのハンドブックです。基本的にはPT630/HT630/MR350/MR250 シリーズのUnitech標準のESC通信に従っていますが、サポートされているコマンドはターミナルの構成によって異なります。

目次

目次	2
1. ホスト ESC コマンド	4
1.1. コントロールコマンド	5
1.1.1. ハードリセット (ESC H)	5
1.1.2. アボート (ESC A)	5
1.1.3. 消去 (ESC E ファイル名)	5
1.2. HT580 からの情報問い合わせ	6
1.2.1. ディレクトリ (ESC D)	6
1.2.2. メモリサイズを得る (ESC G)	6
1.2.3. ファイルがあるかどうかとファイルのサイズをチェックする (ESC J ファイル名)	6
1.2.4. ターミナルIDとバージョン番号を得る (ESC v)	6
1.2.5. オンラインの急速チェック (Bell addr)	6
1.3. 設定コマンド (ESC C)	7
1.3.1. 通信設定 (ESC C)	7
1.3.2. ターミナルアドレスID変更 ID (ESC 5 ID)	7
1.3.3. HT580 ブザーの音量設定 (ESC N)	7
1.3.4. 日付/時刻設定 (ESC M)	7
1.4. 単一のデータ通信	8
1.4.1. HT580 に一つのデータレコードを送信(ESC 0)	8
1.4.2. HT580から一つのデータレコードを得る (STX Addr)	8
1.5. ファイル通信	8
1.5.1. ダウンロード (ESC L ファイル名)	8
1.5.2. アップロード (ESC U ファイル名)	8
2. マルチポイントプロトコル	10
2.1. プロトコルフォーマットの概要	10
2.1.1. データ送信中のデータ変換ルール:	11
2.1.2. ホスト伝送	11
2.1.3. ターミナル伝送	11
2.1.4. プロトコル操作	12
2.1.5. コマンド	12
2.2. コマンドセットプロトコルの概要	13
2.2.1. ESC 0 アプリケーションデータ	13
2.2.2. ESC 5 - マルチポイントアドレスのセット	13
2.2.3. ESC A ソフトリセット、再スタート、またはアボート	13
2.2.4. ESC C - HT580に通信設定テーブルを書く	14
2.2.5. ESC D - RAM ディスクのディレクトリをホストに読む	15
2.2.6. ESC E - HT580 ディレクトリからファイルを消去	15
2.2.7. ESC G - HT580の RAM サイズを得る	15

2.2.8	ESC H	ハードリセットと電源の初期化	15
2.2.9	ESC J	ファイルがあるか無いかをチェック	16
2.2.10	ESC L	HT580にファイルを転送 (ダウンロード)	16
2.2.11	ESC z	ホストから HT580へのファイル転送を取り消す	16
2.2.12	ESC M	HT580にデータと時刻を書き込む	16
2.2.13	ESC N	ブザーの音量をセット	16
2.2.14	ESC U	HT580からファイルを転送 (ファイルアップロード)	17
2.2.15	ESC y	HT580からファイル転送を停止	17
2.2.16	ESC v	ターミナルIDとバージョン番号を得る	17
3.		プログラミングの方法	18
3.1		Formcaching(フォームキャッシング)	18
3.2		Easy Job	18
3.3		通信プログラムのプログラミング	18

1. ホスト ESC コマンド

ホスト通信には三つのクラスがあります。

(1) ホストがHT580にコントロール/設定コマンドを送信

ほとんどすべてのHT580 設定と動作はコントロールコマンドによりホストから制御することができます。設定コマンドは通信コントロールテーブルのようなシステムテーブルをセットアップするために使用されます。コントロールコマンドはHT580の動作の中止、HT580のリセット、アプリケーションプログラムの実行をHT580に指示、あるいは動作に関する他の機能をを実行するために使用されます。

設定コマンドは通常、初期化のプロセス中にホストシステムによって出されます。しかし、コントロールコマンドは、通常の動作中もしくは回復操作中にいつでも出すことができます。

(2) ホストがHT580にデータを要求

HT580システムデータとアプリケーションデータの二種類のデータがホストシステムによって要求されます。アプリケーションデータはアプリケーションプログラムによって生成される情報で、システムデータはファイルシステム、システムパラメータ等のHT580の実行に関する情報です。HT580 は使用されるプロトコルによってデータを転送します。

(3) ホストとHT580間のデータ転送

実行形式のプログラムファイルとデータファイルはホストからHT580へダウンロード、もしくはHT580からホストへアップロードされます。ファイル転送はポイントツーポイントまたはマルチポイント接続でデータを移動するために特別なファイル転送プロトコルを使用します。

以下のセクションでは、ホスト通信コマンドとそれらの機能を簡単に説明します。そして、各機能を詳しく見ていきます。

1.1. コントロールコマンド

1.1.1. ハードリセット (ESC H)

ハードリセットコマンドはすべてのHT580 RAMメモリの内容を消去します。そして主要なハードウェア機能のテストを実行します。以前にHT580に記憶、もしくはホストによってダウンロードされたプログラムまたはデータはメモリから消去されます。標準システムパラメータがFlash ROMから復元されます。ハードリセットコマンドはFlash ROMに保存されたプログラムファイルを消去することはありません。

ハードリセットコマンドはパラメータあるいは値を持っていません。

1.1.2. アポート (ESC A)

アポートは "ソフトリセット" コマンドです。HT580 は実行を停止し、レディ(Ready)モードに戻ります。HT580 メモリに保存されたプログラムとデータは保持されます。システムパラメータは変更されません。アポートコマンドはパラメータあるいは値を持っていません。

1.1.3. 消去 (ESC E ファイル名)

消去コマンドはHT580のメモリからファイルを消去します。ファイルがあり、ファイルが消去された場合、ACK 応答が返ります。そうでなければ、NAK応答がHT580で生成されます。消去コマンドはファイル名のパラメータがあります。

1.2. HT580 からの情報問い合わせ

1.2.1. ディレクトリ (ESC D)

ディレクトリコマンドはHT580メモリのファイルリストを戻すようにHT580に指示します。

ESC D/ROM

1.2.2. メモリサイズを得る (ESC G)

HT580の全メモリサイズ、プログラム実行メモリ、そして空き容量を得ます。

ESC G/ROM

1.2.3. ファイルがあるかどうかとファイルのサイズをチェックする (ESC J ファイル名)

HT580にファイルがあるかどうかをチェックします。ファイルがある場合、そのサイズを戻します。

1.2.4. ターミナルIDとバージョン番号を得る (ESC v)

このコマンドは、ターミナルIDとバージョン番号を得ることができます。標準のターミナルIDは“:HT580 V1.xx”です。

1.2.5. オンラインの急速チェック (Bell addr)

このコマンドは、接続されたHT580の通信が可能かどうかをチェックするために使用されます。PCはHT580が通信可能であることを確認するためにデータの応答を得ます。

1.3. 設定コマンド (ESC C)

1.3.1. 通信設定 (ESC C)

通信設定コマンドは以下のフォーマットです。

ESC C comtable

このコマンドは、ホストから "comtable" データ構造体を取り、二つのHT580内部通信コントロールテーブルの一つに書き込みます。"comtable" 中の構造体は、2.2.4で説明している COM_CONFIG typedef と同じでなければなりません。

新しい通信コントロールテーブルはESC Cコマンドが受け取られるとすぐに有効になります。HT580はその新しいパラメータで対応する通信ポートを初期化します。例えば、ESC CがRS-232ポートを9600 bpsから1200 bpsの転送速度に変更するように指示した場合、HT580はESC Cコマンドを受け取った後は1200 bpsになります。次のホスト通信は1200 bpsを使用します。

1.3.2. ターミナルアドレスID変更 ID (ESC 5 ID)

ターミナルに新しいターミナルIDを指定します。ターミナルIDが変更されたら、ターミナルをリセットすることなしにすぐ有効になります。

1.3.3. HT580 ブザーの音量設定 (ESC N)

このコマンドは、HT580のブザー音量をリモートで設定するために使用されます。

1.3.4. 日付/時刻設定 (ESC M)

日付/時刻設定はコマンド以下のフォーマットです。

ESC M datetime

このコマンドは、ホストシステムがHT580のリアルタイムクロック機能を初期化することができます。パラメータ datetime は以下の表記によるASCII文字列です。

yyyymmddhhmmss

最初の4文字は年です。次の二文字は月で、1月は01です。月の後は日、時間(24時フォーマット)、分と秒です。

例えば、コマンド ESC M 200709262345 は、HT580 の時計を2007年9月26日6、時刻 11:45 PMに設定します。HT580 はESC M コマンドが受け取るとすぐにリアルタイムクロックを設定します。

1.4. 単一のデータ通信

1.4.1. HT580 に一つのデータレコードを送信(ESC 0)

このコマンドはデータ文字列をHT580に送信するために使用されます。HT580 はEasyJob またはターミナルモードになるまでシステムバッファに残します。次のレコードは、前のデータがEasyJobまたはターミナルモードで取り込むまでHT580に送ることはできません。

1.4.2. HT580から一つのデータレコードを得る (STX Addr)

HT580 はPCから呼び出すまでデータを直接PCに送ることはできません。HT580では、システムはHT580のアプリケーションがデータを送信する場合、ポーリングされるデータを持つために一つのレコードデータをバッファに持っています。PCはこのコマンドをHT580がデータを返すために使用する必要があります。

1.5. ファイル通信

1.5.1. ダウンロード (ESC L ファイル名)

ダウンロードコマンドは、ホストシステムからHT580にバイナリ実行形式プログラムまたはデータファイルを転送するために使用されます。HT580 がダウンロードコマンドを受信した場合、ホストにACK 応答を返し、そしてすぐにファイル受信状態に入ります。ファイル受信状態は、前もって割り当てたホストプロトコルkermitによって決まります。ホストシステムは、ACK応答が来るとすぐにファイルの転送を開始します。

ダウンロードコマンドは一つのパラメータ、ファイル名があります。アップロードコマンドは、ダウンロードコマンドと逆の機能を実行します。HT580からホストシステムへのデータファイル転送に使用します。

1.5.2. アップロード (ESC U ファイル名)

HT580 がアップロードコマンドを受信した場合、ファイル転送状態になり、指定されたデータファイルの送信を開始します。ファイル

転送プロトコルは、前もって割り当てたホストプロトコルkermitによって決まります。ホストシステムは、ホストシステムは、アップロードコマンドを送出後データファイルの受信を待ちます。

アップロードコマンドは、一つのパラメータ、ファイル名がありません。

2. マルチポイントプロトコル

マルチポイントでは、HT580 はホストコンピュータとRS232、Bluetoothまたはモデムを通して調歩同期シリアルマルチドロッププロトコルを使用します。ターミナルプロトコルは、以下のフォーマットのコマンドと応答があります。

2.1 プロトコルフォーマットの概要

<u>シンボル</u>	<u>説明</u>
=>	ホストからターミナルへ送信
<=	ターミナルからホストへ送信
ADDR	ターミナルアドレス (A-Y,0-6) + 80H
CMD	ターミナルへのネットワークコマンド、2 バイト、 A-F、0-9
CS1	チェックサム、第一バイト
CS2	チェックサム、第二バイト

チェックサムは、送信データ、ADDR、そしてデータブロック長 (STX と ETXを除く)の各バイトを加算することによって計算されます。CS1 は上位ニブル (4 ビット) +40H そして CS2 は下位ニブル +40Hです。

例: ファイル名 A.EXEをロードするためのコマンド

STX ESC L A . E X E CS1 CS2 ADDR

データブロック = ESC L A . E X E (STXを除く)

データブロック長 = 7

CS = ESC + L + A + . + E + X + E + ADDR + 7

CS1 = CS 上位ニブル + 40H

CS2 = CS 下位ニブル + 40H

ASCII データ文字とその値は以下の通りです。

STX 0x02

ETX 0x03

ACK 0x06
 NAK 0x15
 DC1 0x11
 ESC 0x1B
 EOT 0x04

プロトコル制御文字を含む最大フレームサイズは128文字です。プロトコル制御文字 STX と ETX の透過伝送は \ (逆スラッシュ) 文字を前に置くことによってなされます。 \ 文字の透過伝送は二つの \ 文字を続けることによって行われます。

2.1.1 データ送信中のデータ変換ルール:

- 1) 1バイトデータを2バイトデータに変換

\ <u>00 hex -- 1F hex</u> <u>A0 hex -- FF hex</u>	変換	\ <u>80 hex -- 9F hex</u> <u>20 hex -- 7F hex</u>	変換	\ <u>80 hex -- 9F hex</u> <u>20 hex -- 7F hex</u>
---------------------------------------------------------	----	---------------------------------------------------------	----	---------------------------------------------------------

(DC hexを除く)
- 2) 変換なしに元のデータとして送信される1バイトデータ

<u>他のコード</u>	変換無し
--------------	------

2.1.2 ホスト伝送

<u>送信</u>	<u>フォーマット</u>
ポール	STX, ADDR
ホストデータ	STX, CMD, data, CS1, CS2, ADDR
承認	ACK
非 ACK	NAK

2.1.3 ターミナル伝送

<u>送信</u>	<u>フォーマット</u>
ターミナルデータ	STX, data, CS1, CS2, ETX

承認	ACK
非 ACK	NAK

2.1.4 プロトコル操作

ターミナルプロトコルはマルチポイントのストップアンドウエイトプロトコルとして行われます。ターミナルは一つのフレームだけを送信し応答を待ちます。

以下のシナリオはリンク送信(ポール)の代表例です:

* ターミナルはホストに送るデータがありません。

=> STX ADDR

<= EOT - 送信するデータがない場合

* ターミナルはホストに送るデータがあります。

=> STX ADDR

<= STX <データ> CS1 CS2 ETX - データがある場合

=> ACK - データを正しく受信、または
NAK - エラーが発生

* ホストは一つのポールサイクルでコマンドを送信し、ターミナルからの応答を受け、そしてターミナルコマンド応答の受信を承認します。

=> STX, CMD, パラメータ, ... CS1, CS2, ADDR

<= ACK - データを正しく受信、そして応答が不要、あるいは
NAK - エラーが発生、あるいは送信する応答データがある場合

<= コマンド応答とデータ

=> ACK - コマンド応答がターミナルから送られ、ホストで正しく受信、あるいは
NAK - コマンド応答でエラーが発生

2.1.5 コマンド

以下のコマンドはダウンロード、診断、そしてアプリケーションのデータ転送のためにサポートされています。各コマンドとそのパラメータは、送信の前にフレーム化されます。応答はADDR フィールドのない同じフォーマットです。

STX, CMD, パラメータ, ... CS1, CS2, ADDR

2.2 コマンドセットプロトコルの概要

2.2.1 ESC 0 - アプリケーションデータ

=> STX ESC 0 <データ> CS1 CS2 ADDR
<= ACK または NAK

HT580 は一つの入力アプリケーションデータを保持するために一つのフレームしか持たず、そしてフレームは保持しているデータがターミナルアプリケーションプログラムによって読み込まれるまで有効にはなりません。従って、ターミナルがデータを受信したかどうかを確実に知るために、ターミナルがACKまたはNAKを返したかどうかをチェックすることが重要です。

2.2.2 ESC 5 - マルチポイントアドレスのセット

=> STX ESC '5' <addr> CS1 CS2 ADDR
ここで <addr> = 'A'~'Y','0'~'6'
<= STX ESC '5' <Retcode> CS1 CS2 ETX
ここで <Retcode>= 0x00 OKの場合
0x01 エラー
=> ACK または NAK

2.2.3 ESC A - ソフトリセット、再スタート、またはアポート

このコマンドはHT580 のプログラム実行を終了します。

=> STX ESC A CS1 CS2 ADDR
<= ACK または NAK

2.2.4 ESC C - HT580に通信設定テーブルを書く

=> STX ESC C <comtable> CS1 CS2 ADDR
<= STX ESC C <Retcode> CS1 CS2 ETX
=> ACK または NAK

<comtable> の型定義

```
typedef struct { BYTE    baud_rate;  
                  BYTE    stop_bit;  
                  BYTE    data_bit;  
                  BYTE    parity;  
                  BYTE    protocol;  
                  BYTE    address;  
                  WORD    timeout;  
} COM_CONFIG;
```

HT580 ターミナルは、ホストポートを通してホストと通信します。通信速度は110から 38.4K bpsの間で設定することができます。 .

baud_rate:	'0'	= 110 bps (bits per second)
	'1'	= 150
	'2'	= 300
	'3'	= 600
	'4'	= 1200
	'5'	= 2400
	'6'	= 4800
	'7'	= 9600 (標準値)
	'8'	= 19200
	'9'	= 38400
	'0'	= 57600
stop_bit:	'1'	= 1ストップビット (標準値)
	'2'	= 2ストップビット
data_bit:	'7'	= 7 データビット
	'8'	= 8 データビット (標準値)
parity:	'N'	= パリティなし (標準値)
	'O'	= 奇数パリティ
	'E'	= 偶数パリティ
protocol:	'M'	= マルチポイント (標準値)
	'F'	= プロトコル無し
address:	'A'	= マルチポイントモードのターミナルアドレス(標準値)

各 HT580 はマルチポイントモードを使用する場合はユニークな通信アドレスを持つことができます。アドレスはポーリング機能を実行するためにホストによって使用されます。文字 'A'-'Y' と '0'-'6' は各ターミナルのアドレスIDを指定するために使用されません。

timeout:	'02'	= ポーリングタイムアウト、2サイクル (標準値)
		'02'-'FF' 16進数

この設定値は、通信タイムアウトのために指定されます。HT580 指定したタイムアウトサイクル中にホストシステムから応答を受信しなかった場合、HT580 は通信が失敗したとして通信を中止します。タイムアウト値がゼロに設定されると、タイムアウトチェックはHT580では行われません。

2.2.5 ESC D - RAM ディスクのディレクトリをホストに読む

=> STX ESC D CS1 CS2 ADDR
<= STX ESC D <ディレクトリデータ> CS1 CS2
ETX (または NAK)
=> ACK または NAK

2.2.6 ESC E - HT580 ディレクトリからファイルを消去

=> STX ESC E <filename> CS1 CS2 ADDR
<= STX ESC E <RetCode> CS1 CS2 ETX
ここで <RetCode>:
00 消去できた場合
01 ファイルがない場合
=> ACK または NAK

2.2.7 ESC G - HT580の RAM サイズを得る

=> STX ESC G CS1 CS2 ADDR
<= STX ESC G <xxx> <yyy> <zzz> CS1 CS2 ETX
ここで <xxx>= 全メモリ容量(Kバイト)
<yyy>= 使用中のメモリ(Kバイト)
<zzz>= 空きメモリ容量(Kバイト)
=> ACK または NAK

2.2.8 ESC H - ハードリセットと電源の初期化

このコマンドはHT580ターミナルをリセットし、実行中のアプリケーションは停止します。電源投入時のテストを実行し、正常に完了したら、HT580 は次のコマンドを実行できます。

=> STX ESC H CS1 CS2 ADDR
<= ACK または NAK

2.2.9 ESC J - ファイルがあるか無いかをチェック

=> STX ESC J <filename> CS1 CS2 ADDR
<= STX ESC J <Retcode> CS1 CS2 ETX
ここで Retcode = 00 ファイルがある場合
 = 01 ファイルがない場合
=> ACK または NAK

2.2.10 ESC L - HT580にファイルを転送 (ダウンロード)

=> STX ESC L <filename> CS1 CS2 ADDR
<= ACK or NAK
... ファイル転送が終了するまで次の二つのステップをループする
... => STX ESC 'Y' <データ> CS1 CS2 ADDR
<= ACK または NAK

=> STX ESC Z CS1 CS2 ADDR (データ送信の終了)
<= ACK または NAK

2.2.11 ESC z - ホストから HT580へのファイル転送を取り消す

このコマンドは、ユーザが連続してファイルのダウンロードをしたくない場合、現在のファイルダウンロードを終了するために使用されます。キャンセルした後、HT580のファイルはこの時点で終わります。

=> STX ESC z <ファイル名> CS1 CS2 ADDR
<= ACK または NAK

2.2.12 ESC M - HT580にデータと時刻を書き込む

=> STX ESC M <datetime> CS1 CS2 ADDR
ここで <datetime>
 yyyyymmddhhmmss のASCII 文字列
<= STX ESC M <Retcode> CS1 CS2 ETX
=> ACK または NAK

2.2.13 ESC N - ブザーの音量をセット

=> STX ESC 'N' <n> CS1 CS2 ADDR
ここで <n> = '0' 低音量
 '5' 中音量
 '9' 高音量
<= ACK または NAK

2.2.14 ESC U - HT580からファイルを転送 (ファイルアップロード)

=> STX ESC U <ファイル名> CS1 CS2 ADDR
<= ACK または NAK または
EOT (ファイルがない場合)
... ファイル転送が終了するまで以下の二つの
ステップをループします。
=> STX ESC 'Y' CS1 CS2 ADDR
<= STX ESC 'Y' <data> CS1 CS2 ETX
<= ACK または NAK
<= STX ESC Z CS1 CS2 ETX (ファイル転送終了)
=> ACK または NAK

2.2.15 ESC y - HT580からファイル転送を停止

このコマンドは、ユーザが連続してファイルのアップロードをしたくない場合、現在のファイルアップロードを終了するために使用されます。

=> STX ESC U <ファイル名> CS1 CS2 ADDR
<= ACK または NAK または
EOT (ファイルがない場合)

2.2.16 ESC v ターミナルIDとバージョン番号を得る

=> STX ESC V <devtable> CS1 CS2 ADDR
<= STX ESC V <terminal_id and ver> CS1 CS2 ETX
=> ACK または NAK

3. プログラミングの方法

プログラミングの方法は大きく二つの方法があります。1. HT580のプログラミング、そして2. HT580とホストコンピュータ間の通信プログラムです。以下のヒントは、これらのアプリケーションを速やかかつ効率的に開発するためのアプリケーションとユーティリティについて説明しています。

3.1 Formcaching(フォームキャッシング)

HT580 は、Formcachingによって簡単にプログラムを作ることができます。詳細は HT580 の操作マニュアルをご覧ください。

3.2 Easy Job

Easy Job はWindows ペースのプログラムジェネレータです。HT580 CD ROMに入っており、このソフトウェアの操作ガイドもあります。

3.3通信プログラムのプログラミング

第2章では、HT580とホストコンピュータ間のマルチプロトコルについて詳しく説明しました。ユーザは通信プログラムの開発、もしくはアプリケーションプログラムに組み込むためにこのプロトコルをお使い下さい。通信プロトコルに従って通信プログラムの作成をすることが難しいと感じられる場合があるかもしれません。Unitech は詳細な通信プロトコルを知らなくてもアプリケーションプログラムを開発できるように通信ユーティリティライブラリを用意しています。以下の URL からダウンロードすることができます。

http://w3.adc.unitech.com.tw/pub/cs/software/multi32/multi_dll.zip

これは Windows バージョンの通信プログラム、通信ライブラリとサンプルプログラムです。

Windows プラットフォームでは、Windows 通信プログラムの開発またはアプリケーションプログラムに通信機能を組み込むための通信機能を提供する 16ビット/32ビット通信 DLL を提供しています。ユーザはこの DLL を Visual C/C++、 Visual Basic または Delphi から呼び出すことができます。