

ERHD151 取扱説明書
(Ver 1.01.02.01.16/17 用)

Ver 1.6
2009/06/18

改訂履歴

Ver 0.0		新規作成
Ver 0.2		改訂
Ver 0.3		誤記修正
Ver 0.4		2-1：仕様 を追加 4-1：フロント外観と説明 スイッチと LED の説明を変更 8-*：画像および説明を変更 付録 C：サンプルスクリプトの変更 付録 E：対応 ATA コマンドの変更
Ver 0.5		8-*：画像および説明を変更 付録 C：サンプルスクリプトの変更 付録 E：対応 ATA コマンドの変更
Ver 0.6		6：画像および説明を変更 show コマンド変更 短縮コマンド追加
Ver 0.7		「1：はじめに」に“HDD 異常模擬装置”を追記 「3-1：フロント外観と説明」で FAILED の説明追加 「5-1：起動」の表の下に※を追記 「5-3：操作端末からの使い方」に追記 「5-4：エラー模擬設定」追記 「5-6：ステータスマonitoring」に性能低下を追記 「6：制御コマンド」を「6：コマンドリファレンス」に変更
Ver 1.0		「start」を「run」に変更 「ATA コマンド対応表」を変更
Ver 1.1		「6：コマンドリファレンス」に入力値範囲の記述追加
Ver 1.2		追加機能説明追加 ログ逐次表示コマンド「follow」追加 ランダムエラー発生条件「random_emu」追加 「cont」コマンド変更 5-7.Host 電源モニターを追加
Ver 1.3		ケース変更に伴う変更 誤植修正 システム構成追加
Ver 1.4		誤植修正

Ver 1.5		「Control レジスタ」を「Device Control レジスタ」に変更 「Device レジスタ」を「Device/Head レジスタ」に変更
Ver 1.6		入力範囲の修正 DEBUG コマンドを追加 ログ内容説明追加

目次

1 : はじめに.....	8
2 : 付属品	9
3 : システム概要	10
3-1 : システム構成.....	10
3-2 : フロント外観と説明.....	11
3-3 : リア外観と説明.....	13
4 : セットアップ	14
4-1 : セットアップの注意点.....	14
4-2 : SATA セットアップ	14
4-2-1 : HDD の接続.....	14
4-2-2 : ターゲットとの接続.....	14
4-2-3 : モード設定	15
4-3 : PATA 3.5 インチセットアップ	16
4-3-1 : HDD の接続.....	16
4-3-2 : ターゲットとの接続.....	16
4-3-3 : モード設定	17
4-4 : PATA 2.5 インチセットアップ	18
4-4-1 : HDD の接続.....	18
4-4-2 : ターゲットとの接続.....	19
4-4-3 : モード設定	20
4-5 : 操作端末と本機の接続.....	21
4-5-1 : 操作端末のシリアルポートセットアップ	21
4-5-2 : 操作端末のターミナルセットアップ	22
5 : 操作	23
5-1 : 起動.....	23
5-2 : 終了.....	24
5-3 : 操作端末からの使い方.....	25
5-3-1 : 時刻設定.....	28
5-3-2 : コマンド履歴.....	28
5-3-3 : ヘルプ.....	29
5-3-4 : ATA コマンド名リスト	29
5-4 : エラー模擬設定.....	30
5-4-1 : スクリプトファイルの記述.....	32
5-4-2 : スクリプトファイルの送信.....	33

5-4-3 : 登録したトリガーの確認	34
5-4-4 : エラー模擬動作の開始/終了	34
5-5 : ATA コマンドログ参照	35
5-5-1 : ログの保持情報	36
5-5-2 : 最新ログ表示	37
5-5-3 : 模擬ログ表示	40
5-5-4 : ログの保存	42
5-6 : ステータスマモニタリング	43
5-7 : Host 電源モニター	45
5-8 : 便利な使い方	46
6 : コマンドリファレンス	47
6-1 : エラー模擬コマンド	47
delay	47
error	47
all_abort	47
all_abort_unlock	48
busy	48
busy_unlock	48
6-2 : エラー模擬制御	49
trigger	49
tl (小文字 L)	50
delete	50
delete_all	50
enable	50
disable	51
cont	51
ignore	51
run	52
r	52
stop	52
s	52
count_reset	52
6-3 : トリガー条件	53
cmd_count_gt	53
cmd_count_le	53
time_expire	54

lba_range.....	54
recv_cmd.....	54
random_emu.....	55
6-4 : ログ参照.....	56
show.....	56
l (小文字 L).....	57
10 (小文字 L).....	57
11 (小文字 L).....	57
12 (小文字 L).....	58
13 (小文字 L).....	58
log_clear.....	59
lc (小文字 L).....	59
summary.....	59
y.....	59
6-5 : 汎用コマンド.....	60
echo.....	60
date.....	60
help.....	61
?.....	61
history.....	61
h.....	61
script.....	62
monitor.....	62
m.....	62
follow.....	63
f.....	63
ata_cmd.....	63
7 : 仕様.....	64
7-1 : 仕様.....	64
7-2 : HOST I/F 詳細 (リア SATA/PATA).....	65
7-3 : SLAVE I/F 詳細 (内部 SATA/PATA).....	65
7-4 : 外部通信 I/F 詳細 (リア SERIAL).....	65
8 : トラブルシューティング.....	66
8-1 : 電源 OFF ができない.....	66
8-2 : FAIL LED が点灯する.....	66
8-3 : 操作端末から正しく文字を入力できない.....	66

8-4: スクリプトファイルをダウンロードするとエラーと表示される	66
付録 A: 基板外観図	67
付録 B: 基板セッティング	68
①: 1/4	68
②: 2/4	69
③: 3/4	70
④: 4/4	71
付録 C: サンプルスクリプト	72
Read コマンドで UNC エラーを模擬	72
Write コマンドで CRC エラーを模擬	73
遅延処理模擬	74
複合エラー模擬	75
HDD ハングアップ (全コマンド無視)	76
付録 D: ATA コマンド対応表	77
付録 E: DEBUG コマンド	81

1 : はじめに

本書は、HDD 異常模擬装置 ERHD151（以降 本機）の取扱説明書です。
本機の Ver 01.01.02.01.16 に対応しております。
本機は、検査対象装置（以降 HOST / ターゲット）と HOST に本来接続されている HDD（以降 SLAVE）の間に接続し、SLAVE のエラーを模擬します。
本機の操作は、操作端末より行います。

また、本機では処理した ATA コマンドをロギングします。
最新の約 10000 個の ATA コマンドログと最近の 3 個のエラー模擬時点のログ
前後約 5000 個(約 10000 個)のログを保持します。

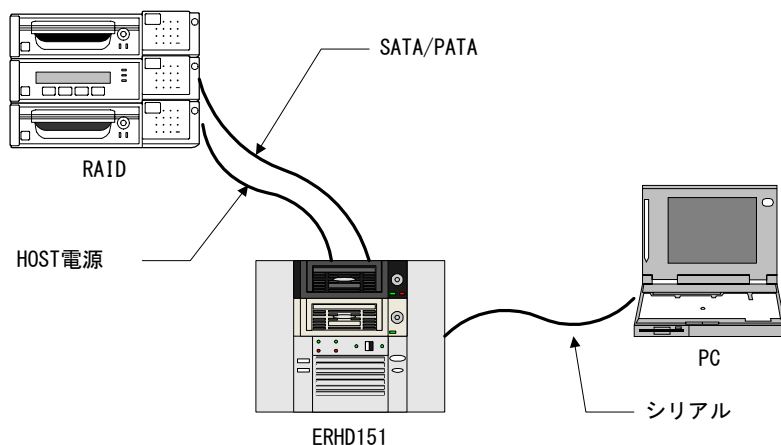
*いかなる場合においても、本機を使用したことにより本機の接続機器への損害及、
HDD のデータ破損に関しては免責とさせていただきます。

2 : 付属品

本体（基板/電源組み込み済み）	1 式
本体用電源ケーブル	1 本
SATA 信号ケーブル	1 本
4pin-SATA 電源変換ケーブル	1 本
4pin 電源延長ケーブル	1 本
SATA 電源延長ケーブル	1 本
80 芯 3.5 インチ用 PATA ケーブル	1 本
3.5-2.5 変換コネクタ	1 個
3.5-2.5 変換基板	1 枚
2.5 インチ用オス-オス変換基板	1 枚
3.5 インチ用オス-オス変換基板	1 枚
USB-シリアル変換器	1 個
ネジ	1 式

3：システム概要

3-1：システム構成



RAID：検査対象装置

ERHD151：本機

PC：ERHD151 制御用 PC

RAID より HDD を取り出し、ERHD151 のトレイにセットします。
この時、RAID より SATA/PATA 信号ケーブルと HDD 電源ケーブルを接続します。
RAID から見ると ERHD151 は HDD ドライブそのものに見えます。

エラー模擬等の処理は制御 PC より行い ERHD151 は RAID からのアクセス内容を記録し続けます。

エラー模擬のログやアクセスログは ERHD151 内部で保持していますが容量は限りがありますが、制御 PC 側へ常時出力する機能を使用することで長時間の運用も可能です。

3-2 : フロント外観と説明



LED

図番	名称	色	説明
	RUN	緑	本機 内部の H/W が正常に立ち上がると、点滅します。 HOST の電源ONで点灯、OFF で点滅します。
	HOST	緑	HOST 側 HDD アクセスランプです。 HOST 側に対して、BUSY / DREQ 状態で点灯します。
	FAIL	赤	起動時は点灯です。 本機が操作端末からのコマンド入力受付可能になると消灯します。 セッティング（後述）の不備がある場合、起動後も点灯したままになります。
	TRIG	赤	起動時は点灯です。 本機が操作端末からのコマンド入力受付可能になると消灯します。 その後、トリガーにヒットすると点灯します。 次のコマンド等受信すると消灯します。

			そのコマンドがトリガーにヒットすると、再度点灯します。
	SATA	緑	SATA モードで起動しています。
	PATA	緑	PATA モードで起動しています。
④	POWER LED	緑	本機に電源が供給されると点灯します。
⑤	アクセス LED	赤	パネルの HOST LED と同じです。 HOST 側 HDD アクセスランプです。 HOST 側に対して、BUSY / DREQ 状態で点灯します。

スイッチ

図番	名称	説明
①	電源スイッチ	本機の主電源スイッチです。 POWER OFF 時に 1 回押すと電源 ON、 POWER ON 時に 1 回押すと電源 OFF となります。 操作ミス等で本機がハングアップした場合、電源 OFF ができなくなる場合があります。 そのときは、本スイッチを長押しすれば電源が落ちます。
②	リセットスイッチ	本機のリセットスイッチです。 POWER スイッチ長押しでも本機の電源が切れなくなった場合のみ本スイッチを押します。 その後、電源スイッチを再度押します。 それ以外の場合は、操作しないで下さい。
③	動作モード	本機を SATA、PATA で動作させるための設定です。 スイッチを SATA 側に倒すと SATA モード、反対側が PATA モードです。 通電中にスイッチを操作してもモードの変更はできません。
⑥	キースイッチ	HDD トレイのロックスイッチです。 時計回りに回すとロックされ、反時計回りに回すとアンロックされます。 本スイッチの操作は必ず電源スイッチが OFF の時に操作してください。 またキーをロックしないとトレイには電源供給されません。

3-3 : リア外観と説明



コネクタ

図番	名称	説明
①	RS232C	操作端末とのシリアルケーブルを接続します。
②	SATA	SATA 信号ケーブルを接続します。(右側) POWER モニター用電源ケーブルを接続します。(左側)
③	PATA	PATA 信号ケーブルが本部分からでています。
④	電源	本機の電源ケーブルを接続します。

4：セッティング

4-1：セッティングの注意点

本機のセッティングを変更する場合、以下の2点について確認してから行ってください。

- a)本機の電源が切れていること (RUN ランプの消灯)
- b)本機に HOST を接続する場合には、本機並びに HOST の電源が切れていること

4-2：SATA セッティング

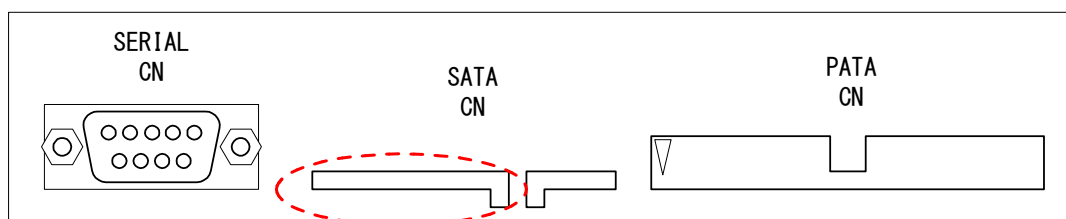
作業前にターゲットと本機の電源が切断されていることを確認してください。

4-2-1：HDD の接続

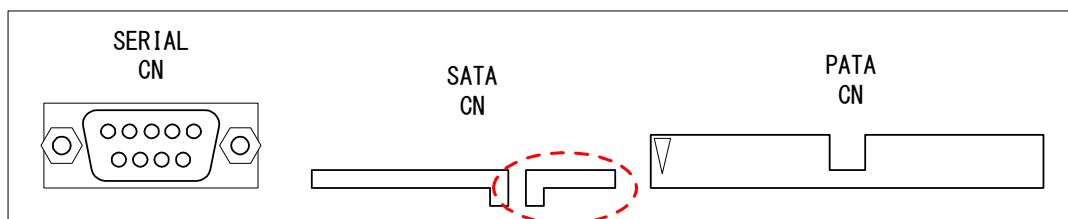
- ・ 5 インチベイラックよりキーロックを解除しトレイを取り出します。
- ・ トレイ内の HDD に SATA 信号ケーブルと SATA 電源ケーブルを接続します。
- ・ HDD をトレイに固定します (2.5 インチ HDD の場合は、付属のマウンタを使用して固定します)。
- ・ 5 インチベイラックにトレイを挿入しキーをロックします。

4-2-2：ターゲットとの接続

- ・ 4pin 電源ケーブルに 4pin-SATA 電源ケーブルを接続します。
- ・ ターゲットの電源と CN5B (SATA CN の左側の端子) を上記ケーブルで接続します。

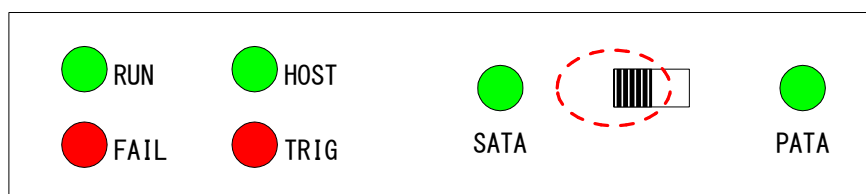


- ターゲットの SATA 信号コネクタと CN5A (SATA CN の右側の端子) を SATA 信号ケーブルで接続します。



4-2-3 : モード設定

- パネルのスイッチを SATA 側に ON します。



4-3 : PATA 3.5 インチセッティング

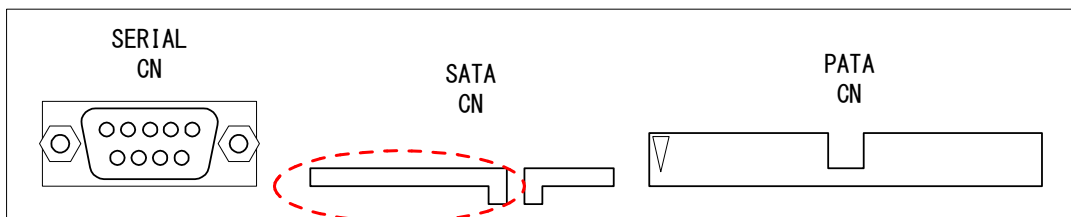
作業前にターゲットと本機の電源が切断されていることを確認してください。

4-3-1 : HDD の接続

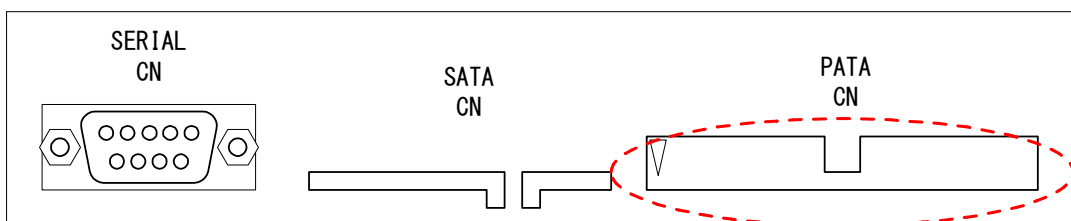
- ・ 5 インチベイラックよりキーロックを解除しトレイを取り出します。
- ・ トレイ内の 3.5 インチ HDD に 80 芯 3.5 インチ用 PATA ケーブルと電源ケーブルを接続します。
- ・ HDD をトレイに固定します。
- ・ 5 インチベイラックにトレイを挿入しキーをロックします。

4-3-2 : ターゲットとの接続

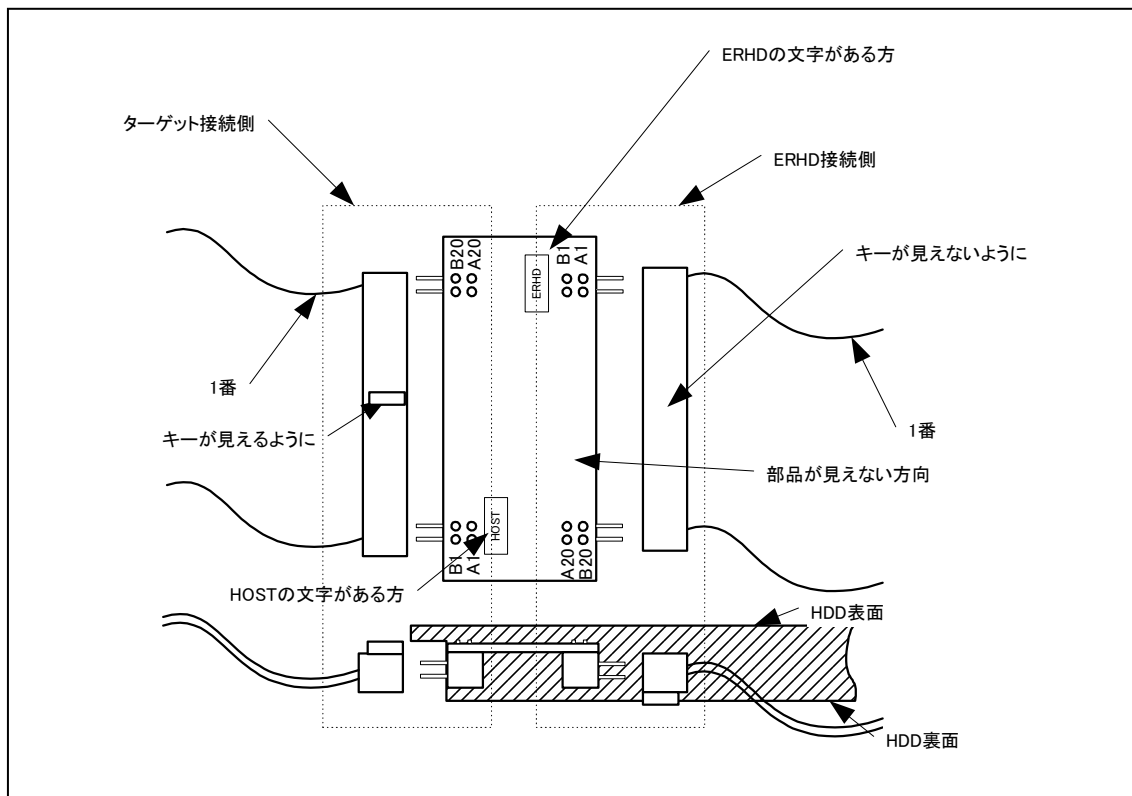
- ・ 4pin 電源ケーブルに 4pin-SATA 電源ケーブルを接続します。
- ・ ターゲットの電源と CN5B (SATA CN の左側の端子) を上記ケーブルで接続します。



- ・ ターゲットが「オスコネクタ」の場合、PATA CN からでているケーブルを直接ターゲットに接続します。



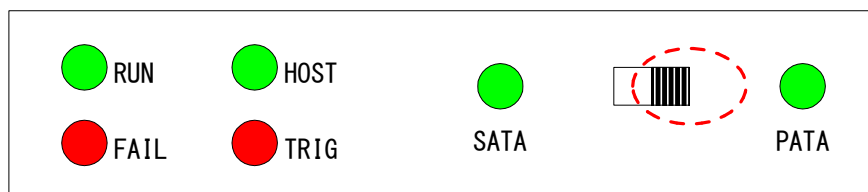
- ターゲットが「メスコネクタ」の場合、PATA CN からでてくるケーブルの先に下図のように 3.5 インチ用オス-オス変換基板を接続します。
変換基板は、HDD に対して下図のようなイメージになります。



- 変換基板の HOST 側を上図 HDD イメージになるようにターゲットと接続します。

4-3-3: モード設定

- パネルのスイッチを PATA 側に ON します。

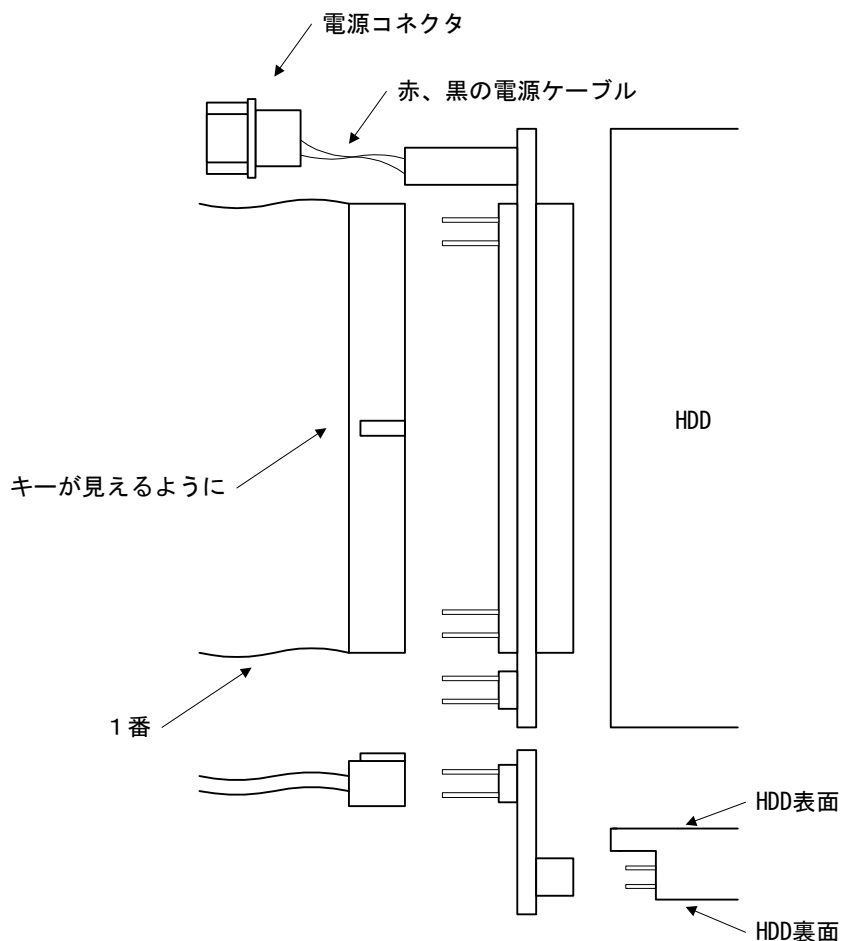


4-4 : PATA 2.5 インチセッティング

作業前にターゲットと本機の電源が切断されていることを確認してください。

4-4-1 : HDD の接続

- 5 インチベイラックよりキーロックを解除しトレイを取り出します。
- トレイ内の 80 芯 3.5 インチ用 PATA ケーブルの先に、下図のように 3.5-2.5 変換コネクタを接続します。
- トレイ内にアダプタを介して HDD をセットします。

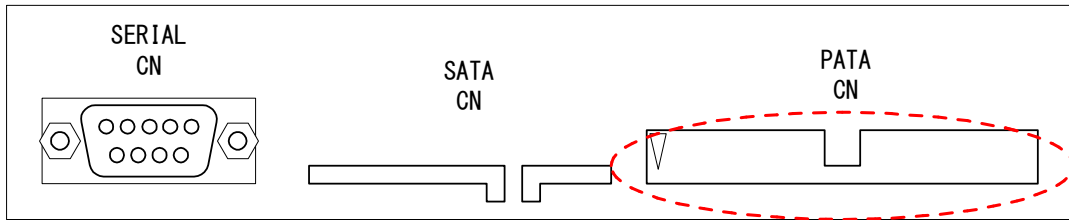


- HDD と上記ケーブルで接続します。
- トレイ内の電源ケーブルを電源コネクタと接続します。

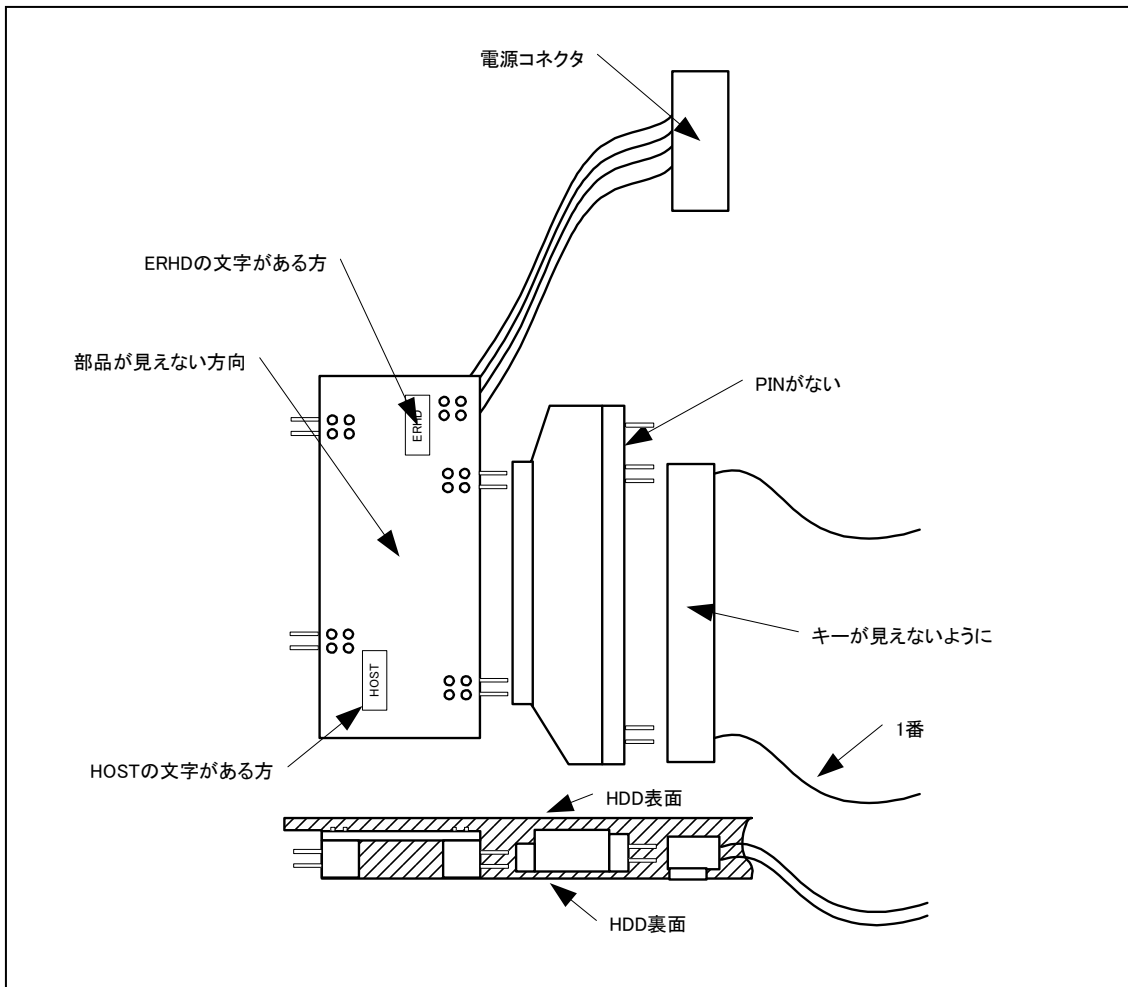
- HDD を 2.5 インチマウンタを使用してトレイに固定します。
- 5 インチベイラックにトレイを挿入しキーをロックします。

4-4-2 : ターゲットとの接続

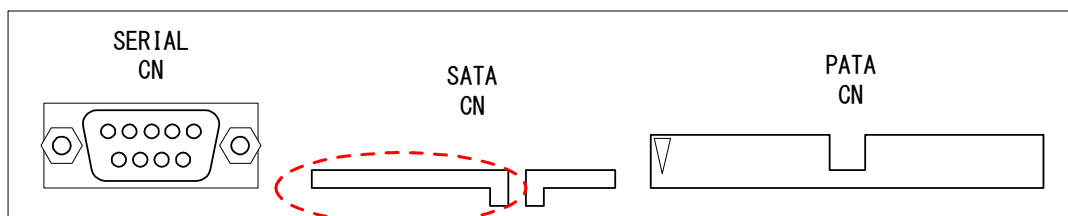
- PATA CN からでていたケーブルに下図のように変換コネクタを接続します。



変換基板は、HDD に対して下図のようなイメージになります。

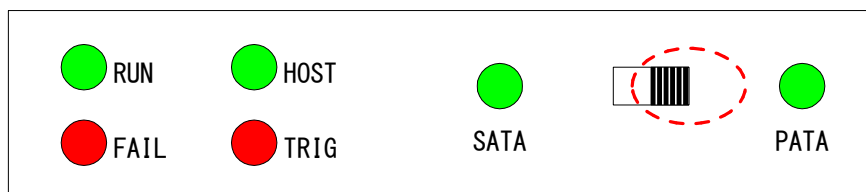


- 変換基板の HOST 側を上図 HDD イメージになるようにターゲットと接続します。
- 変換基板からでている 4pin 電源ケーブルに 4pin-SATA 電源ケーブルを接続します。
- CN5B (パネル SATA CN の左側の端子) に上記ケーブルを接続します。



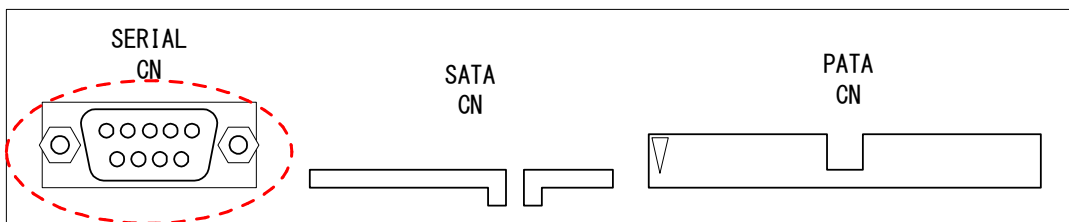
4-4-3 : モード設定

- パネルのスイッチを PATA 側に ON します。



4-5 : 操作端末と本機の接続

- 操作端末と本機を接続するには付属の USB-シリアル変換器を使用します。
また、RS232C ストレートケーブル（3 線式）を使用しての接続も可能です。
- 操作端末のシリアルポートと、本機の(SERIAL)とを上記ケーブルで接続します。



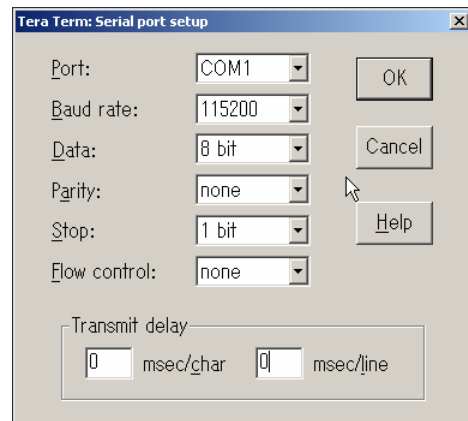
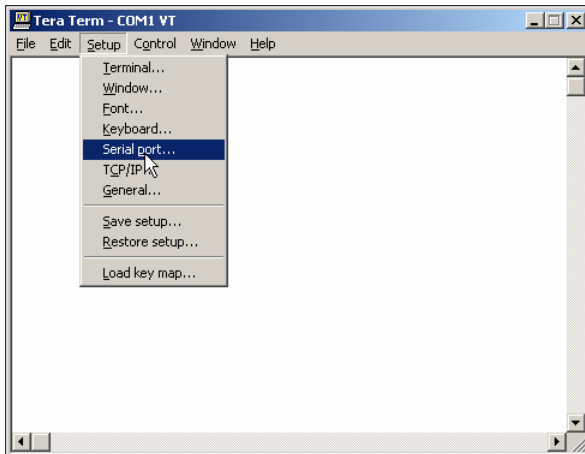
4-5-1 : 操作端末のシリアルポートセッティング

シリアルポートの設定値は下記のとおりです。

ボーレート	115200
データビット	8bit
パリティビット	無し
ストップビット	1 ビット
フロー制御	無し

TeraTerm Pro での設定

1. [setup]->[Serial Port]を選択します。
2. Port : お使いの PC のセッティングにあわせて選択して下さい。
3. その他の項目は、下図のように設定して下さい。
4. 設定が完了後、OK を押して下さい。



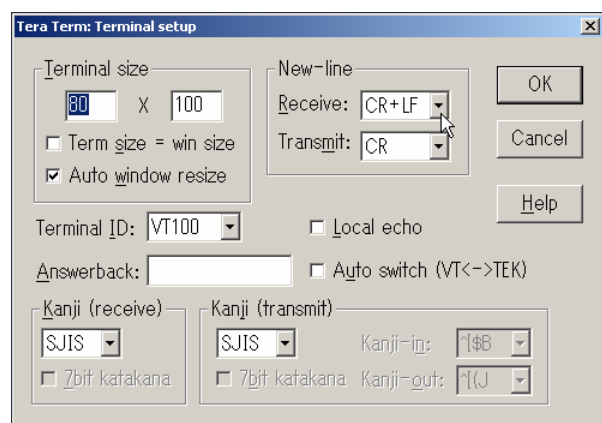
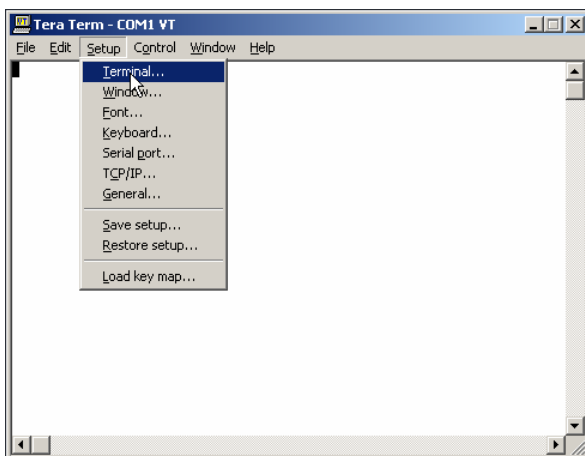
4-5-2 : 操作端末のターミナルセッティング

ターミナルの設定は下記のとおりです。

受信改行コード	LF 又は CR + LF
送信改行コード	LF 又は CR
端末タイプ	VT100 (互換)

TeraTerm Pro での設定

1. [setup]->[Terminal]を選択します。
2. [New-line]の項目を下図のように設定して下さい。
3. 設定が完了後、OK を押して下さい。



5 : 操作

5-1 : 起動

起動手順は以下のとおりです。

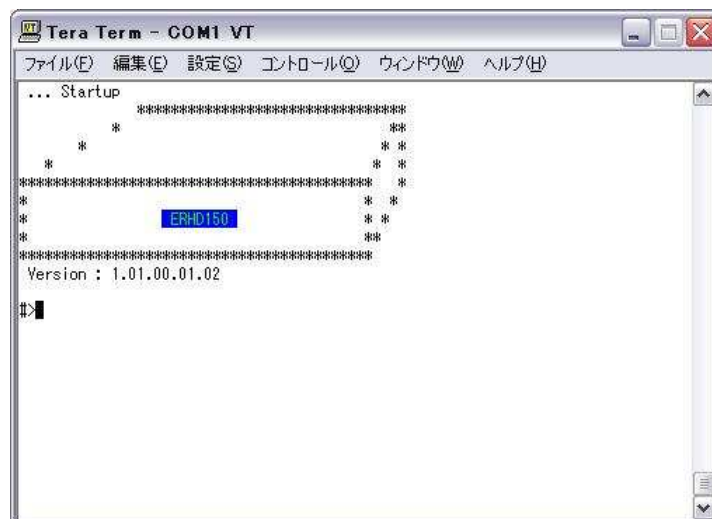
1. 本機に電源を入れます。
2. 本機の起動完了を待ちます。(下表)
3. 本機起動完了後にターゲットの電源を入れます。

本機の電源投入後の動作は以下のとおりです。

	起動中	正常起動	起動失敗
RUN LED[緑]	点滅	点滅	消灯 または 点滅
HOST LED[緑]	不定	不定	不定
TRIG LED[赤]	点灯	消灯	消灯 または 点灯
FAIL LED[赤]	点灯	消灯	消灯 または 点灯
POWER LED[緑]	点灯	点灯	消灯 または 点灯
アクセス LED[赤]	不定	不定	不定
操作端末	「Start up ...」表示	「#>」表示	なにも表示しない または 「#>」表示

※ : 「不定」のLEDは外部からの信号状態により変化します。

起動後、操作端末は下図のようになります。



正常起動しなかった場合には、すぐに本機の電源を落とし、セッティングに間違いがないか確認してください。

5-2 : 終了

終了手順は以下のとおりです。

1. ターゲットのシャットダウンをし、電源を切断します。
2. 本機リアの電源スイッチを押し、電源を切断します。
3. 終了後は本機の全ての LED が消灯します。

*操作ミス等により、本機がハングアップし電源が切断できなくなった場合、
電源ボタンを電源が切れるまで長押しして下さい。

*電源ボタン長押しでも切断できない場合、
リセットスイッチを押し、その後、電源ボタンを押します。

5-3 : 操作端末からの使い方

本機の制御は操作端末からのコマンド入力により行います。コマンドは次のルールで入力してください。

コマンド名 [オプション] [引数]

コマンド名には実行したいコマンドを入力します。コマンドに続けてオプション、引数をスペースで区切って入力します。オプション、引数は各コマンドによって異なるので「6 : コマンドリファレンス」を参照してください。

本機のコマンドは下表のように分類されます。

エラー模擬	エラー模擬の内容を設定するためのコマンド群
エラー模擬制御	エラー模擬を登録、制御するコマンド群
トリガー条件	エラー模擬を動作させる条件に使用するコマンド群
ログ参照	保持している ATA コマンドログを参照するためのコマンド群
汎用	本機への設定や制御を行うコマンド群

本機を制御するコマンドは下表のとおりです。各コマンドの詳細は「6：コマンドリファレンス」を参照してください。

エラー模擬		
コマンド	引数	コメント
delay	必須	ATA コマンド処理を遅延します。
error	必須	エラー応答を返します。
all_abort	無	すべてのコマンドをアボートします。
all_abort_unlock	無	all_abort を解除します。
busy	無	リクエストに対して busy ロックします。
busy_unlock	無	busy を解除します。
エラー模擬制御		
コマンド	引数	コメント
trigger	必須	設定するトリガーを指定します。
end_trigger	無	トリガーの入力を終了します。
setact	必須	設定するアクションを指定します
enable	必須	トリガーを有効にします。
disable	必須	トリガーを無効にします。
delete	必須	トリガーを削除します。
delete_all	無	全トリガーを削除します。
cont	必須	トリガーの繰り返し回数を指定します。
ignore	必須	トリガーの無視回数を指定します。
run	無	模擬動作を開始します。
stop	無	模擬動作を停止します。
count_reset	無	cmd_count_gt などと比較するカウンタをリセットします。
トリガー条件		
コマンド	引数	コメント
cmd_count_gt	必須	コマンド数が指定数を超えた後、真になります。
cmd_count_lt	必須	コマンド数が指定数を超えるまで真になります。
time_expire	必須	指定時間が経過すると真になります。
lba_range	必須	指定範囲へのアクセス要求を受信したら真になります。
recv_cmd	必須	指定のコマンドを受信したら真になります。
random_emu	必須	受信コマンド数に対して指定の頻度で真になります。
ログ参照		

コマンド	引数	コメント
show	オプション	ATA コマンドログを表示します。
log_clear	無	ATA コマンドログをクリアします。
summary	無	ATA コマンド毎の要求回数/エラー回数を表示します。
汎用		
コマンド	引数	コメント
echo	オプション	エコー
date	オプション	日時の表示、設定します。
help	無	ヘルプを表示します。
history	無	コマンドの履歴を表示します。
script	無	スクリプト入力を開始します。
monitor	無	ATA コマンドモニター開始/停止(トグル) します。
follow	無	ATA コマンド逐次表示を開始/停止(トグル)します。
ata_cmd	無	コマンド ID、コマンド名のリストを表示します。

短縮コマンド		
tl	無	“trigger -l”と同じです。
r	無	run と同じです。
s	無	stop と同じです。
l (小文字 L)	オプション	show と同じです。
l0 (小文字 L)	オプション	最新の ATA コマンドログを表示します。(show @0)
l1 (小文字 L)	オプション	トリガーログ 1 の ATA コマンドログを表示します。(show @1)
l2 (小文字 L)	オプション	トリガーログ 2 の ATA コマンドログを表示します。(show @2)
l3 (小文字 L)	オプション	トリガーログ 3 の ATA コマンドログを表示します。(show @3)
lc	無	log_clear と同じです。
y	無	summary と同じです。
?	無	help と同じです。
h	無	history と同じです。
m	無	monitor と同じです。
f	無	follow と同じです。

5-3-1 : 時刻設定

本機には時刻設定を保持する機能がありません。起動ごとに時刻設定を行う必要があります。

"date"コマンドにより時間を設定することができます。

例) #>date -s 2006/11/11 12:34:56

設定を省略した場合は起動時が「1970/1/1 00:00:00」となります。

TeraTerm のマクロにより起動時に設定する方法は後述の「便利な使い方」を参照してください。

5-3-2 : コマンド履歴

過去に実行したコマンドの履歴を 32 個保持します。

上矢印キー／下矢印キーで選択し、Enter キーで実行できます。

Ctrl+u で選択した履歴または入力した 1 行をキャンセル出来ます。

過去に実行したコマンドの履歴は"history"又は"h"コマンドで表示できます。



5-3-3 : ヘルプ

"help"又は"?"コマンドでコマンドのリストが表示されます。

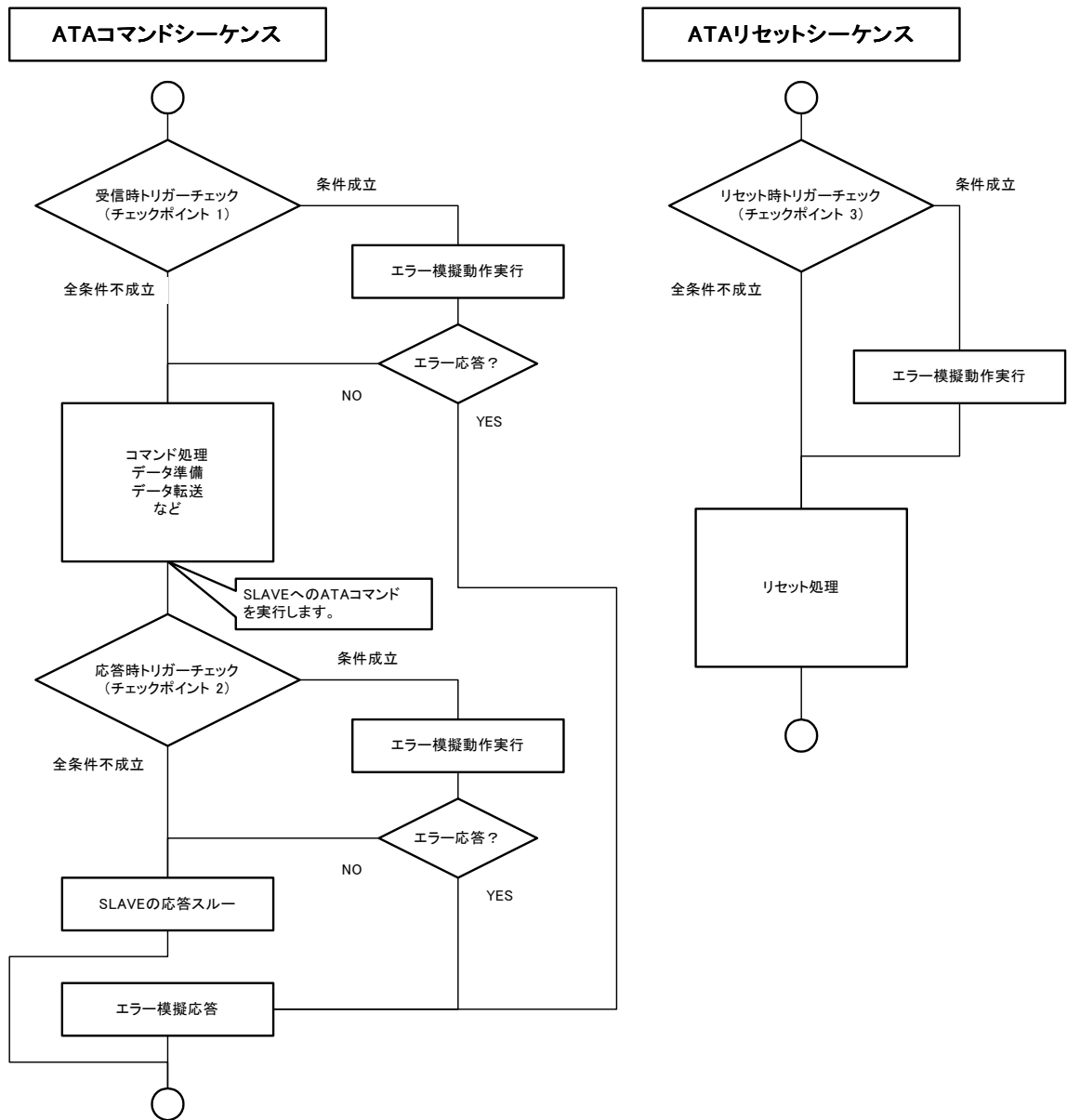
5-3-4 : ATA コマンド名リスト

"ata_cmd"コマンドで ATA コマンド名と ID のリストが表示されます。

表示されたコマンド名は "recv_cmd" コマンドで指定できます。

5-4 : エラー模擬設定

本機で模擬できるデバイスのエラーは ATA コマンドのシーケンス上で発生するエラーです。エラー模擬のシーケンスは下図のようになっています。



前図のように ATA コマンドシーケンス上に 3 箇所のエラー模擬を設定箇所があります。それぞれのチェックポイントにエラー模擬を設定します。

各チェックポイントでは登録されたエラー模擬を登録された順序で条件をチェックします。あるエラー模擬の条件が成立した場合、そのエラー模擬を行い以降のエラー模擬はチェックされません。

設定するエラー模擬は下の項目で構成されます。(以降 トリガー)

1. 前述のチェックポイント (受信時: -c1、応答時: -c2、リセット時: -c3)
2. 発生条件(コマンド数、経過時間、アドレス範囲など)
3. 模擬動作(エラー応答、処理遅延など)

スクリプトファイルでのトリガーの記述方法はしたのとおりです。

- ① “trigger”コマンドでチェックポイントを指定し、発生条件入力を始めます。
- ② 発生条件を指定します。(トリガー条件コマンドを使用します。)
- ③ “end_cond”で条件指定を終了します。
- ④ “setact 名前”で模擬動作指定を始めます。
- ⑤ 模擬動作を指定します。(エラー模擬コマンドを使用します。)
- ⑥ “end_action”で模擬動作指定を終了します。
- ⑦ “end_trigger”で本トリガーの指定を終了します。

トリガーの例

trigger -c1 2	① チェックポイント1に設定
recv_cmd(read_sector)	② 発生条件を指定
recv_cmd(read_sector_obsolete)	
recv_cmd(read_sector_ext)	
recv_cmd(read_multiple)	
recv_cmd(read_multiple_ext)	
recv_cmd(read_long)	
recv_cmd(read_long_obsolete)	
recv_cmd(read_dma)	
recv_cmd(read_dma_obsolete)	
recv_cmd(read_dma_ext) @read_unc_err	
end_cond	③ 条件指定終了
setact read_unc_err	④ 模擬動作指定開始
error UNC	⑤ 模擬動作指定
end_action	⑥ 模擬動作指定終了
end_trigger	⑦ トリガー指定終了

各コマンドの詳細は「6: コマンドリファレンス」を参照してください。

5-4-1 : スクリプトファイルの記述

本機のエラー模擬設定はスクリプトファイルを送信して行います。スクリプトファイルはテキストエディターで編集し、ターミナルソフトから ASCII モードでファイルを転送し実行できます。

スクリプトファイルのルールは以下の通りです。

- 1 行目を「script」で始める
- 最終行を「end_script」で終わる
- 1 行に 1 コマンドを記述する
- 1 行の中の「#」以降の文字列は無視される（コメント）なお、コメントとして認識されるのは、「script」～「end_script」になります。

スクリプトの例

```
script
stop
delete_all
trigger -c1 2
    recv_cmd(read_sector)
    recv_cmd(read_sector_obsolete)
    recv_cmd(read_sector_ext)
    recv_cmd(read_multiple)
    recv_cmd(read_multiple_ext)
    recv_cmd(read_long)
    recv_cmd(read_long_obsolete)
    recv_cmd(read_dma)
    recv_cmd(read_dma_obsolete)
    recv_cmd(read_dma_ext) @read_unc_err    # setact の名前と同じにします。
end_cond
setact read_unc_err    # @の名前と同じにします。
    error UNC          # エラー指定(CRC, UNC, ABRT, IDNF, TKONF, AMNF)
end_action
end_trigger
cont 2 1                # 上記の TrigNo. 2 に繰り返し回数を指定します。
ignore 2 1              # 上記の TrigNo. 2 に最初に無視する回数を指定します。
trigger -l              # 登録したトリガーを表示します。
run                     # 模擬動作をスタートします。
end_script
```


5-4-2 : スクリプトファイルの送信

TeraTerm のメニューバー「ファイル」「ファイル送信(S)...」であらかじめ作成しておいたスクリプトファイルを送信します。



受信した文字数、行数が表示されます。

「#>」が表示されると送信完了です。

5-4-3 : 登録したトリガーの確認

"trigger -l"又は"tl"コマンドを実行します。登録されているトリガーのサマリー（下図）が表示されます。

表示内容は下表のとおりです。

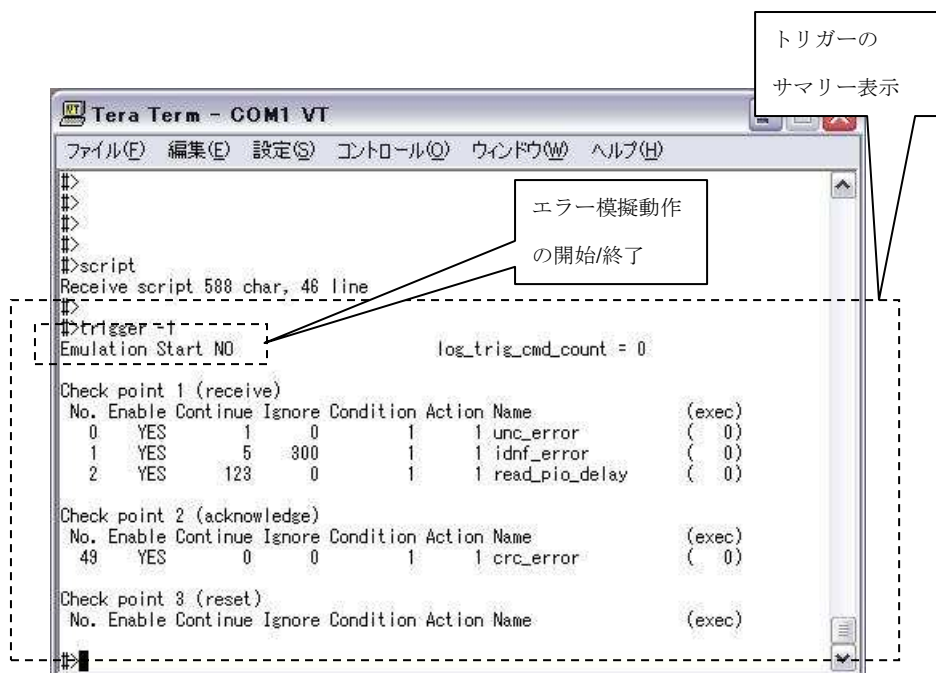
No.	登録されているトリガー番号
Enable	設定されたトリガーが有効状態の場合「YES」、無効状態の場合「NO」
Continue	繰り返し回数（模擬動作を開始している場合は残数）
Ignore	無視する回数（模擬動作を開始している場合は残数）
Condition	設定した条件の数
Action	設定されたアクションの数
Name	アクションの名前（先頭のみ）
(exec)	実行された回数

5-4-4 : エラー模擬動作の開始/終了

"run"又は"r"、"stop"又は"s"コマンドで行います。

"trigger -l"または"monitor"コマンドで状態を確認できます。（下図参照）

この操作は後述の ATA コマンドログに保持されます。



5-5 : ATA コマンドログ参照

本機では処理した ATA コマンドをロギングします。

最新の約 10000 個の ATA コマンドログ(以降 最新ログ)を保持します。

ロギングする内容は以下の通りです。

1. ATA レジスタ値(注：転送データは保持しません)
2. ATA コマンド受信時間 (ミリ秒単位)

最近の 3 個のエラー模擬時点のログ (以降 模擬ログ) を最新ログとは別途保持します。

模擬ログでは前後約 5000 個(約 10000 個)のログを保持します。ロギング内容は最新ログと共通です。

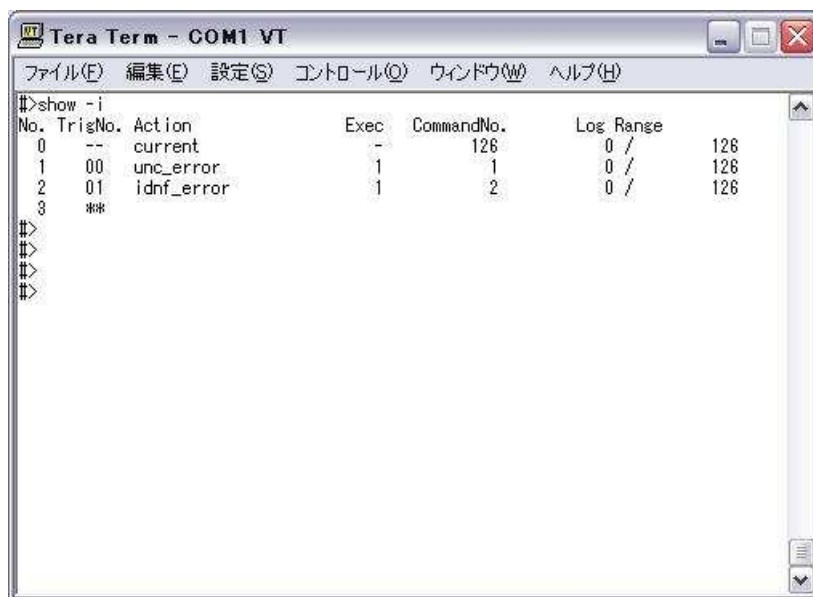
5-5-1 : ログの保持情報

"show -i"又は"l -i"コマンドでログの保持情報を表示します。

No.0 は最新ログの情報です。

No.1~3 はそれぞれ模擬ログの情報です。

該当ログが未保持の場合は「**」が表示されます。(下図の No.3)



表示内容は下表のとおりです。

No.	トリガーログ番号 (1 が最古で 3 が最新)
TrigNo.	エラー模擬を行ったトリガーの番号
Action	アクションの名前
Exec	このアクションの実行回数
CommandNo.	エラー模擬を行ったコマンドの番号
Log Range	各トリガーに対して保持しているコマンド番号の範囲

コマンド番号は最新ログを保持し始めたときからの通番です。”log_clear”コマンドでログをクリアした場合は 0 になります。

5-5-2 : 最新ログ表示

最新ログの表示には"show"コマンドを使用します。2種類の表示方法があります。

簡易表示(show @0 個数 /10 個数)

@0 の[0]は「5-5-1 : ログの保持情報」の No.0、最新ログを意味します。

```

Tera Term - COM1 VT
ファイル(F) 編集(E) 設定(S) コントロール(C) ウィンドウ(W) ヘルプ(H)
#>
#>
#>
#>show @0 15
Time          Cmd No. ID Command Name          LBA          Sect Err
01:56:38.456  112 ca WRITE_DMA        0000006001c7h 0008
01:56:38.456  113 ca WRITE_DMA        0000006001d7h 0008
01:56:38.456  114 ca WRITE_DMA        0000006001ffh 0008
01:56:38.456  115 ca WRITE_DMA        000000600227h 0008
01:56:38.457  116 ca WRITE_DMA        000000600237h 0008
01:56:38.457  117 ca WRITE_DMA        00000060024fh 0008
01:56:39.403  118 ca WRITE_DMA        00000140059fh 0008
01:56:39.403  119 ca WRITE_DMA        00000000620fh 0008

01:56:39.403  120 ca WRITE_DMA        00000000cb9fh 0008
01:56:39.404  121 ca WRITE_DMA        00000000d287h 0008
01:56:39.404  122 ca WRITE_DMA        00000000d7c7h 0008
01:56:40.403  123 ca WRITE_DMA        000000013627h 0008
01:56:40.403  124 ca WRITE_DMA        000000014a9fh 0008
01:56:42.255  125 ca WRITE_DMA        0000005e0047h 0008
01:56:42.256  126 ca WRITE_DMA        000000016877h 0008

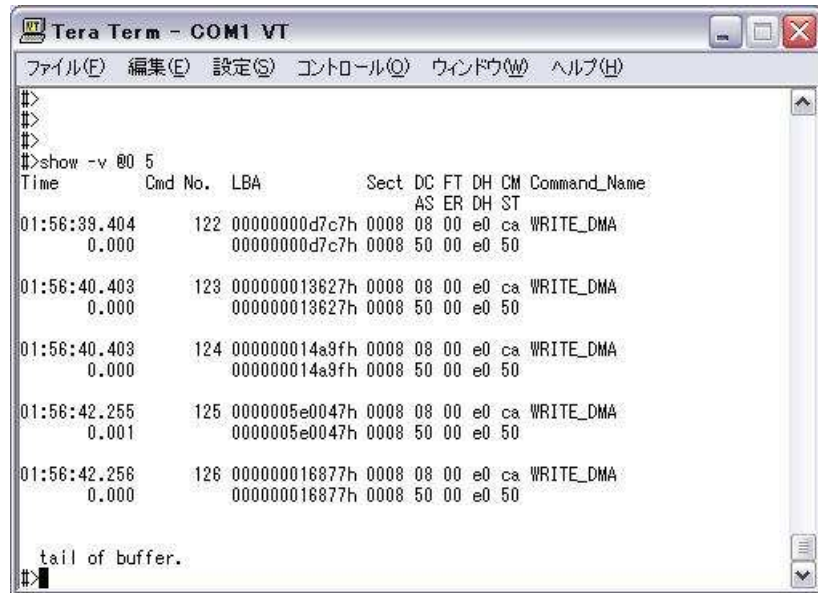
tail of buffer.
#>
  
```

表示内容は下表のとおりです。

Time	ATA コマンドを受信した時刻
Cmd No.	コマンド番号
ID.	Command レジスタ値 (ATA コマンド ID)
Command_Name	ATA のコマンド名称
LBA	要求 LBA 値
Sect	要求セクター数
Err	正常終了 : 無表示 SLAVE でエラー : E SLAVE が無応答 : ? エラー模擬実行時正常終了 : *00 (00 は trigger -cx XX で指定された XX 番号) エラー模擬実行時エラー : E*00 (00 は trigger -cx XX で指定された XX 番号)

	エラー模擬実行時無応答 = ?*00 (00 は trigger -cx XX で指定された XX 番号)
--	--

詳細表示(show -v @0 個数 / 10 個数)



```
Tera Term - COM1 VT
ファイル(F) 編集(E) 設定(S) コントロール(C) ウィンドウ(W) ヘルプ(H)
#>
#>
#>
#>show -v @0 5
Time      Cmd No.  LBA          Sect DC FT DH CM Command_Name
          AS ER DH ST
01:56:39.404 122 00000000d7c7h 0008 08 00 e0 ca WRITE_DMA
          0.000          00000000d7c7h 0008 50 00 e0 50
01:56:40.403 123 0000000013627h 0008 08 00 e0 ca WRITE_DMA
          0.000          0000000013627h 0008 50 00 e0 50
01:56:40.403 124 0000000014a9fh 0008 08 00 e0 ca WRITE_DMA
          0.000          0000000014a9fh 0008 50 00 e0 50
01:56:42.255 125 00000005e0047h 0008 08 00 e0 ca WRITE_DMA
          0.001          00000005e0047h 0008 50 00 e0 50
01:56:42.256 126 0000000016877h 0008 08 00 e0 ca WRITE_DMA
          0.000          0000000016877h 0008 50 00 e0 50

tail of buffer.
#>
```

表示内容は2行で1つのATAコマンドログです。

1行目：受信時のログ		
	Time	ATAコマンドを受信した時刻
	Cmd No.	コマンド番号
	LBA	要求LBAアドレス
	Sect	要求セクター数
	DC	Device Control レジスタ値
	FT	Feature レジスタ値
	DH	Device/Head レジスタ値
	CM	Command レジスタ値 (ATA コマンド ID)
	Command_Name	ATA のコマンド名称
2行目：応答時のログ		
	Time	ATAコマンド処理時間
	Cmd No.	エラー応答時のみ「ERROR」を表示
	LBA	要求LBAアドレス
	Sect	要求セクター数
	AS	Altanate Status レジスタ値
	ER	Error レジスタ値
	DH	Device/Head レジスタ値
	ST	Status レジスタ値
	Command_Name	エラー模擬実行時のみ「Trig_**」を表示 (**は trigger -cx XX で指定された XX 番号)

最後の2行に現在のコマンド数の情報を表示します。

内容は標準表示と同じです。

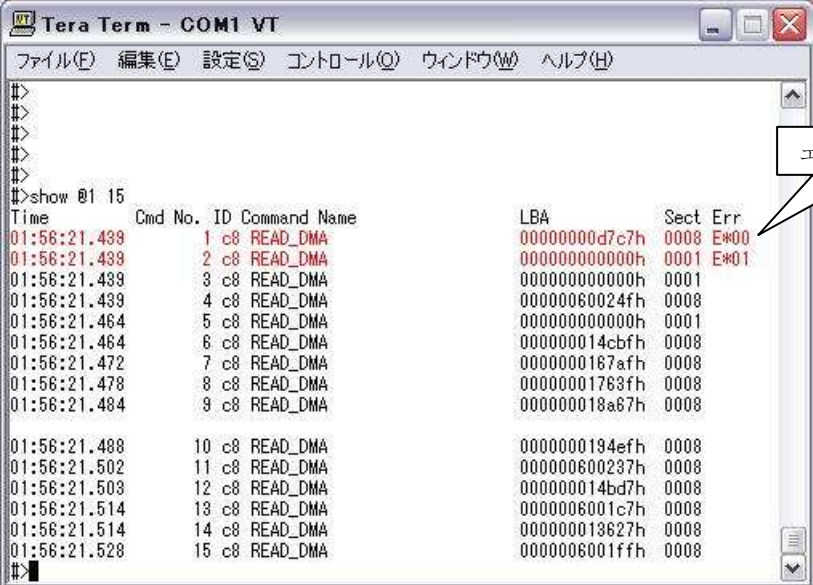
"show"コマンドの詳細は「6：コマンドリファレンス」を参照してください。

5-5-3 : 模擬ログ表示

模擬ログ表示には show コマンドを使用します。最近の模擬ログ 3 つを @1-3 で指定します。

簡易表示(show @1 個数 / l1 個数)

@1 の [1] は「5-5-1 : ログの保持情報」の No.1、模擬ログの 1 番目を意味します。



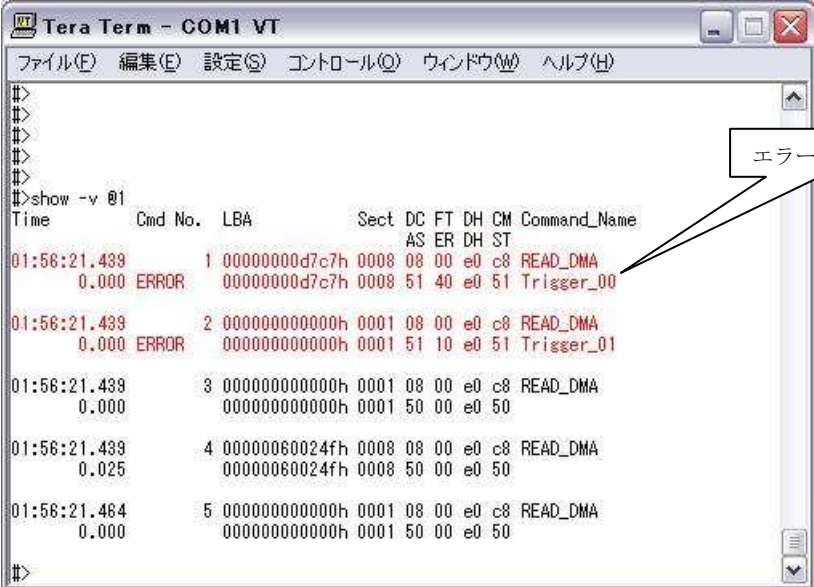
```
VT Tera Term - COM1 VT
ファイル(F) 編集(E) 設定(S) コントロール(Q) ウィンドウ(W) ヘルプ(H)
#>
#>
#>
#>
#>
#>
#>show @1 15
Time          Cmd No.  ID Command Name          LBA          Sect Err
01:56:21.439 1  c8  READ_DMA          00000000d7c7h 0008 E*00
01:56:21.439 2  c8  READ_DMA          000000000000h 0001 E*01
01:56:21.439 3  c8  READ_DMA          000000000000h 0001
01:56:21.439 4  c8  READ_DMA          00000060024fh 0008
01:56:21.464 5  c8  READ_DMA          000000000000h 0001
01:56:21.464 6  c8  READ_DMA          000000014cbfh 0008
01:56:21.472 7  c8  READ_DMA          0000000167afh 0008
01:56:21.478 8  c8  READ_DMA          00000001763fh 0008
01:56:21.484 9  c8  READ_DMA          000000018a67h 0008

01:56:21.488 10 c8  READ_DMA          0000000194efh 0008
01:56:21.502 11 c8  READ_DMA          000000600237h 0008
01:56:21.503 12 c8  READ_DMA          000000014bd7h 0008
01:56:21.514 13 c8  READ_DMA          0000006001c7h 0008
01:56:21.514 14 c8  READ_DMA          000000013627h 0008
01:56:21.528 15 c8  READ_DMA          0000006001ffh 0008
#>
```

エラー発生ログ

表示内容は前述の「5-5-2 : 最新ログ表示」を参照してください。

詳細表示(show -v @1 個数 /ll -v 個数)



```
VT Tera Term - COM1 VT
ファイル(F) 編集(E) 設定(S) コントロール(C) ウィンドウ(W) ヘルプ(H)
#>
#>
#>
#>
#>
#>show -v @1
Time      Cmd No.  LBA          Sect DC FT DH CM Command_Name
          AS ER DH ST
01:56:21.439 1 00000000d7c7h 0008 08 00 e0 c8 READ_DMA
0.000 ERROR 00000000d7c7h 0008 51 40 e0 51 Trigger_00
01:56:21.439 2 000000000000h 0001 08 00 e0 c8 READ_DMA
0.000 ERROR 000000000000h 0001 51 10 e0 51 Trigger_01
01:56:21.439 3 000000000000h 0001 08 00 e0 c8 READ_DMA
0.000      000000000000h 0001 50 00 e0 50
01:56:21.439 4 00000060024fh 0008 08 00 e0 c8 READ_DMA
0.025      00000060024fh 0008 50 00 e0 50
01:56:21.464 5 000000000000h 0001 08 00 e0 c8 READ_DMA
0.000      000000000000h 0001 50 00 e0 50
#>
```

エラー発生ログ

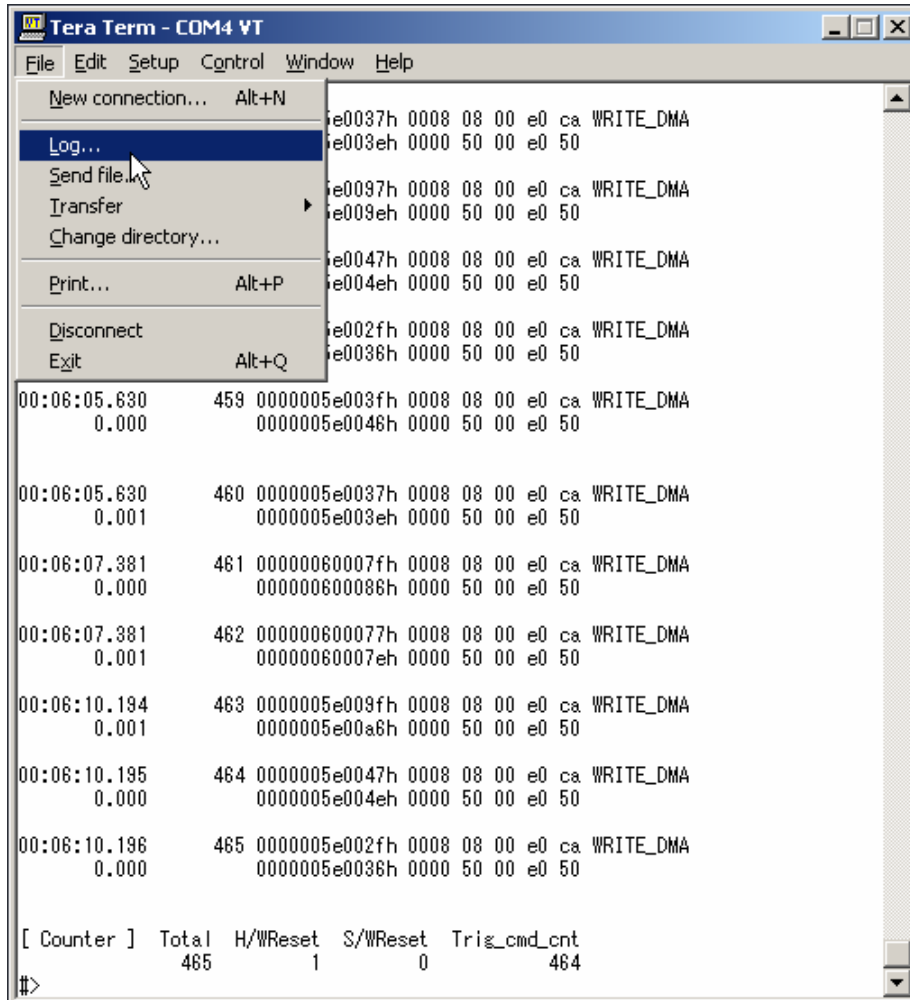
表示内容は前述の「5-5-2：最新ログ表示」を参照してください。

"show"コマンドの詳細は「6：コマンドリファレンス」を参照してください。

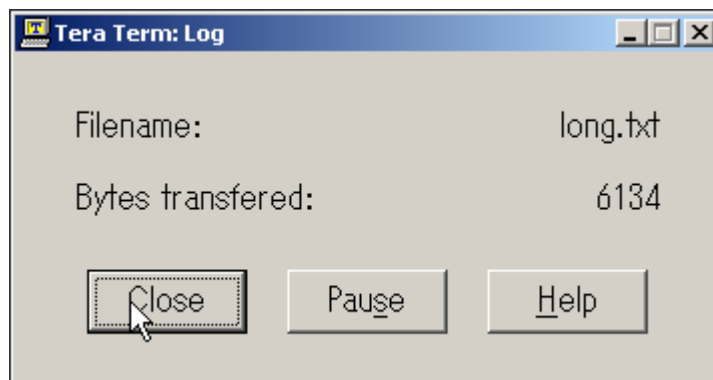
5-5-4 : ログの保存

TeraTerm のメニューバー「File」「Log...」でファイル名を入力します。

"show"コマンドでログを出力します。



TeraTerm のログウィンドウの"Close"ボタンを押して終了します。



5-6 : ステータスマモニタリング

"monitor"又は"m"コマンドでステータスマモニタリング(以降 モニタ)を起動することが出来ます。

モニタを起動すると操作端末が下図になります。

先頭から 15 行は表示専用エリアとなり入力は出来ません。

16 行目以降はコマンド入力が可能です。

モニタを起動中に"monitor"又は"m"コマンドを入力すると終了します。

* ステータスマモニタリングを起動中は HOST から見た ATA デバイスの性能が低下します。

The screenshot shows a terminal window titled "Tera Term - COM1 VT" with a menu bar (File, Edit, Settings, Control, Window, Help). The terminal content is as follows:

```
Emulation      Command count  H/W Reset  S/W Reset  TrigExec
START         1440         2         2         289
[ Recent ERROR ]
No. TrigNo. Action      Exec  CommandNo.  Log Range
1   00   unc_error      1    1186        1 / 1433
2   01   idnf_error    1    1187        1 / 1433
3   49   crc_error     1    1200        1 / 1433
[ Recent ATA Commands ]
Time      Cmd No. ID Command Name      LBA          Sect Err
00:20:23.973 1436 ca WRITE_DMA      0000005e1b77h 0008
00:20:23.973 1437 ca WRITE_DMA      0000005e0047h 0008
00:20:23.973 1438 ca WRITE_DMA      0000005e0037h 0008
00:20:28.892 1439 ca WRITE_DMA      000000049a8fh 0008
00:20:28.892 1440 ca WRITE_DMA      000000016877h 0008
TIME      00:20:31
#>
#>
#>
#>
#>
#>
#>
#>
#>
#>
#>
```

1～2行 各種カウンタ表示

表示内容は下表のとおりです。

Emulation	エラー模擬動作の開始
Command count.	受信した ATA コマンドの総数
H/W Reset	受信した H/W リセットの総数
S/W Reset	受信した S/W リセットの総数
TrigExec	発生させたエラー模擬の総数

3～7行 トリガー表示

表示内容は show コマンドのトリガーログ保持情報と同じです。

8～14行 ATA コマンド表示

表示内容は「5-5-2：最新ログ表示」の簡易表示と同じです。

5-7 : Host 電源モニター

リモート電源の OFF を検出したら、電源 OFF の模擬を行います。(Host へ応答を返しません。)

電源 OFF の模擬として、Host への応答レジスタを「0x7f」でフィルします(以降、電源 OFF 応答と略します)。コマンド要求を受けた場合は常に電源 OFF 応答を返します。このとき Host からの要求コマンドはログに保存します。

1. Hostに対してPower_offのレジスタ(0x7fフィル)を設定します
Hostからの要求をATAコマンドログに残します
2. Host側の転送をとめます
Slave処理中に電源OFFを検知した場合、Slave処理が終了してもHostへは応答を返しません
3. 電源OFFを検知した後は、Hostからの要求に対しPower_offのレジスタ(0x7fフィル)で応答し、Slaveへは要求を送出しません

電源 OFF 時の ERHD 状態と動作

ERHD 状態	電源 OFF 時動作
待機中	電源 OFF の作業
Host PIO 転送中	転送を完了し次の ATA コマンドから電源 OFF の模擬を行う。
Host DMA 転送中	転送を完了し次の ATA コマンドから電源 OFF の模擬を行う。 (データを送りきらないと Host との DMAC をリセットできないので転送中のコマンドは処理しきる。)
Slave PIO 転送中	Slave に対する処理は 1ATA コマンドの最後まで行う。
Slave DMA 転送中	ただし Slave 処理が終了しても Host には返さない。
Slave Nondate 発行中	

5-8 : 便利な使い方

TeraTerm の起動時に設定ファイル、マクロ実行を行います。

以下のファイルを TeraTerm をインストールしたフォルダに用意します。

erhd.ini(シリアルポート設定)

前述の「操作端末のセッティング (TeraTerm Pro)」で設定したものを保存してください。

erhd_settime.ttl(時刻設定マクロ)

```
;; Term macro for ERHD
;; 日時設定

PRONPT = '#>'

wait PRONPT
getdate nowdate
gettime nowtime
sendln 'date -s ' nowdate ' ' nowtime
```

ショートカットのプロパティでリンク先を下記のように変更します。

```
"C:¥Program Files¥TTERMPRO¥ttermpro.exe" /F=erhd.ini /M=erhd_settime.ttl
```

6 : コマンドリファレンス

6-1 : エラー模擬コマンド

delay

- 書式

delay *time*

- 説明

ATA コマンド処理を遅延します。

time には遅延時間(ミリ秒)を整数で指定します。指定可能な有効範囲は 1~59000 です。

S/W リセット又は H/W リセットで解除されます。

error

- 書式

error *err*

- 説明

err で指定されたエラー模擬します。Error レジスタの値を 16 進数(0x**)又は、次の文字列で指定します。

CRC	(0x80)	: CRC エラー
UNC	(0x40)	: UNC エラー
IDNF	(0x10)	: IDNF エラー
ABRT	(0x04)	: ABRT エラー
TK0NF	(0x02)	: TK0NF エラー
AMNF	(0x01)	: AMNF エラー

all_abort

- 書式

all_abort

- 説明

以降に受信する全てのコマンドにアボート応答します。

S/W リセット又は H/W リセットで解除されません。

all_abort_unlock

- 書式
all_abort_unlock
- 説明
all_abort コマンドを解除します。

busy

- 書式
busy
- 説明
以降に受信する全てのコマンドに **BSY** ステータスのまま応答しません。
S/W リセット又は H/W リセットで解除されません。

busy_unlock

- 書式
busy_unlock
- 説明
busy コマンドを解除します。

6-2 : エラー模擬制御

trigger

- 書式

trigger [option] *tnum*

トリガー条件 A &&

トリガー条件 B *action_name*

end_cond

setact *action_name*

・・・(模擬動作指定)

end_action

end_trigger

- 説明

tnum で指定された番号でトリガー条件を登録します。「&&」で AND 条件を指定できます。*action_name* を指定した後に AND 条件の指定はできません。何もない場合は OR 条件となります。トリガー条件が成立した時に *action_name* で指定された模擬動作を実行します。*action_name* は setact に指定する文字列と同じにしてください。

end_cond で条件入力を終了します。

setact で模擬動作を記述します。end_action 模擬動作入力を終了します。

end_trigger でトリガーの入力を終了します。

本コマンドは”trigger”内では使用できません。

設定可能な数は下記の表に従います

登録可能なトリガー数	50
指定可能な条件数	20
指定可能なアクション数	20
設定可能なコマンド数	20

- オプション

-c1 チェックポイントを 1/コマンド受信時に設定します。デフォルト値

-c2 チェックポイントを 2/コマンド応答時に指定します。

-c3 チェックポイントを 3/リセット受信時に指定します。

-l [list] トリガーの登録状態を表示します。*tnum* が指定された場合はそのトリガーの詳細情報を表示します。

tl (小文字 L)

- 書式
tl
- 説明
”trigger -l”と同じです。

delete

- 書式
delete *tnum*
- 説明
tnum で指定されたトリガーを削除します。 *tnum* に設定可能な有効範囲は 0～49 です。

delete_all

- 書式
delete_all
- 説明
すべてのトリガーを削除します。

enable

- 書式
enable *tnum*
- 説明
tnum で指定されたトリガーを有効にします。 *tnum* に設定可能な有効範囲は 0～49 です。

disable

- 書式
`disable tnum`
- 説明
`tnum` で指定されたトリガーを無効にします。`tnum` に設定可能な有効範囲は 0～49 です。

cont

- 書式
`cont tnum count`
- 説明
`tnum` で指定されたトリガーが `enable` になってから条件成立を繰り返す回数を `count` に設定します。トリガーが `disable` の場合、本カウントはカウントされません。
`count` の回数だけアクションを実行しトリガーは `disable` 状態になります。
`count` が 0 以下の時、設定は無効になります。`tnum` に設定可能な有効範囲は 0～49 で、`count` に指定可能な有効範囲は 1～99999999 です。

ignore

- 書式
`ignore tnum count`
- 説明
`tnum` で指定されたトリガーが `enable` になってから条件成立を無視する回数を `count` に設定します。トリガーが `disable` の場合、本カウントはカウントされません。
`count` の回数だけトリガー条件が成立しても、アクションは実行しません。
`count` が 0 以下の時、設定は無効になります。`tnum` に設定可能な有効範囲は 0～49 で、`count` に指定可能な有効範囲は 1～99999999 です。

run

- 書式
run
- 説明
エミュレーション動作を開始します。
本コマンドは”trigger”内では使用できません。

r

- 書式
r
- 説明
run コマンドと同じです。

stop

- 書式
stop
- 説明
エミュレーション動作を停止します。

s

- 書式
s
- 説明
stop コマンドと同じです。

count_reset

- 書式
count_reset
- 説明
トリガー条件コマンドで比較するカウンタをリセットします。

6-3 : トリガー条件

cmd_count_gt

- 書式

cmd_count_gt(*count*)

- 説明

トリガーが有効になった後の受信したコマンド数が *count* 以上の場合、真になります。

カウント対象となるコマンドはすべてのコマンドです。

count にはコマンド数を整数で指定します。*count* に指定可能な有効範囲は 1～99999999 です。0 以下が指定された場合は即時に条件が真になります。

count_reset コマンドで比較対象となるカウンタをリセットできます。

cmd_count_le

- 書式

cmd_count_le(*count*)

- 説明

トリガーが有効になった後の受信したコマンド数が *count* 以下の場合、真になります。

カウント対象となるコマンドはすべてのコマンドです。

count にはコマンド数を整数で指定します。*count* に指定可能な有効範囲は 1～99999999 です。0 以下が指定された場合は即時に条件が偽になります。

count_reset コマンドで比較対象となるカウンタをリセットできます。

time_expire

- 書式

`time_expire(time)`

- 説明

トリガーが有効になった後、システム内部時間が *time* 時間経過した場合、真になります。

time には経過時間（秒）を整数で指定します。*time* に指定可能な有効範囲は 1～99999 です。0 以下が指定された場合は即時に条件が真になります。

lba_range

- 書式

`lba_range (lower, upper)`

- 説明

Host からの要求アドレス(LBA)が *lower* 以上、*upper* 以下であった場合、真になります。

lower、*upper* は 16 進数(0x**)又は整数で指定します。Slave に接続した HDD の範囲内で指定します。

recv_cmd

- 書式

`recv_cmd(command)`

- 説明

現在受信している ATA コマンドが *command* である場合、真になります。

command には“ata_cmd” コマンドで表示された文字列か、値で入力する場合は ATA コマンド ID の範囲 0x00～0xff です。

random_emu

- 書式

random_emu (N)

- 説明

Nはエラー発生頻度を百分率（パーセント）を整数で指定します。100回の判定に対して不定期にN回条件が真になります。

他のトリガー条件と併用する例

```
recv_cmd(READ_DMA)
recv_cmd(READ_DMA_EXT) &&
random_emu(25) @ read_random_err
```

と記述した場合、指定したコマンド(READ_DMA, READ_DMA_EXT)を100回受信した内25回read_random_errを実行します。

[注意事項]

以下のようにrandom_emu()を先に指定すると、

```
random_emu(N) &&
lba_range(1000, 8000) @random_error
```

受信コマンドのうちNパーセントの要求をLBA判定し、合致すればエラー動作を行います。したがって、Nパーセントでエラーが模擬されるわけではありません。

6-4 : ログ参照

show

- 書式

show [option] [*start*] [*+num/-num*] [*count*]

- 説明

コマンドログを *start* (表示開始位置) で指定したコマンドから *count* 個表示します。表示開始位置 (*start*) が指定されなかった場合は前回の続きを表示します。

start は後述の文字列を指定します。 *+num* 又は *-num* で表示開始位置をコマンド数指定で移動できます。

count のデフォルト値は 20 です。

参照間でコマンドログがなくなった場合、最古のログから表示します。

参照中のトリガーログバッファがなくなった場合 (エラーが連続で発生した場合) には参照中のログはカレントログバッファに変更されます。

count 表示数を指定します。以降省略した場合は前回の表示数で表示します。デフォルトは 20 です。

start 表示開始位置を指定します。指定には下記の文字列を指定します。

@0 カレントログバッファの最新ログ(末尾から *count* 個表示)

@1 トリガーログバッファ 1 のトリガー発生ログ

@2 トリガーログバッファ 2 のトリガー発生ログ

@3 トリガーログバッファ 3 のトリガー発生ログ

最後のログ指定

@0h カレントログバッファの最古のログ

@1h トリガーログバッファ 1 の最古のログ

@2h トリガーログバッファ 2 の最古のログ

@3h トリガーログバッファ 3 の最古のログ

最新のログ指定

@1t トリガーログバッファ 1 の最新のログ(末尾から *count* 個表示)

@2t トリガーログバッファ 2 の最新のログ(末尾から *count* 個表示)

@3t トリガーログバッファ 3 の最新のログ(末尾から *count* 個表示)

±num 表示開始位置を現在方向 (+*num*) 又は過去方向 (-*num*) に移動させる数を指定します。

- オプション
 - v 詳細ログ表示
 - i ATA コマンドログ保持情報表示

l (小文字 L)

- 書式
l
- 説明
show コマンドと同じです。

l0 (小文字 L)

- 書式
l0
- 説明
show コマンドと同じです。引数 **start** が@0 固定です。
他の引数およびオプションは指定できます。

l1 (小文字 L)

- 書式
l1
- 説明
show コマンドと同じです。引数 **start** が@1 固定です。
他の引数およびオプションは指定できます。

12 (小文字 L)

- 書式

12

- 説明

show コマンドと同じです。引数 `start` が@2 固定です。
他の引数およびオプションは指定できます。

13 (小文字 L)

- 書式

13

- 説明

show コマンドと同じです。引数 `start` が@3 固定です。
他の引数およびオプションは指定できます。

log_clear

- 書式
log_clear
- 説明
すべてのログをクリアします。
コマンドカウントもクリアされます。

lc (小文字 L)

- 書式
lc
- 説明
log_clear コマンドと同じです。

summary

- 書式
summary
- 説明
ATA コマンド毎の要求回数/エラー回数/エラー模擬数を表示します。

y

- 書式
y
- 説明
summary コマンドと同じです。

6-5 : 汎用コマンド

echo

- 書式
`echo "test message"`
- 説明
引数で指定された文字列「*test message*」をコンソールに表示します。
模擬動作中は ATA コマンドログにロギングされます。

date

- 書式
`date [option]`
- 説明
システムの時間(日時)を表示、設定します。
オプションなしで現在時刻を表示します。
設定を省略した場合は起動時が「1970/01/01 00:00:00」となります。
- オプション
-s 日時日時を設定します。
例) #>date -s 2006/1/1 12:00:00

help

- 書式
help
- 説明
コンソールで使用できるコマンドのリストを表示します。

?

- 書式
?
- 説明
help コマンドと同じです。

history

- 書式
history
- 説明
以前に入力されたコマンドのリストを表示します。最大で 32 個表示します。

h

- 書式
h
- 説明
history コマンドと同じです。

script

- 書式

`script`

- 説明

スクリプト入力を開始します。スクリプトファイルの先頭に本コマンドを記述します。末尾行には `end_script` を記述します。

コンソールへの未送信(未入力)が約 10 秒間続くとタイムアウトします。

monitor

- 書式

`monitor`

- 説明

ATA コマンドモニタ画面を起動します。起動中に本コマンドを入力するとモニタ画面が終了します。

m

- 書式

`m`

- 説明

`monitor` コマンドと同じです。

follow

- 書式

follow

- 説明

本コマンド入力から以降のATAコマンドをコンソールに表示します。起動中に本コマンドを入力すると終了します。ATAコマンドのログ表示は逐次行われ表示フォーマットはshowコマンドと同等になります。

-

例)

```
00:52:16.951 133980 00000157974fh 0003 00 4f 41 c4 READ_MULTIPLE
00:52:16.973          7184a157974fh 0403 50 00 41 50
```

```
00:52:16.973 133981 0000014c3655h 0004 00 85 41 c8 READ_DMA
Trigger_01
00:52:16.973 ERROR 0000014c3655h 0004 51 40 41 51
```

- オプション

-v 詳細ログ表示

f

- 書式

f

- 説明

follow コマンドと同じです。

ata_cmd

- 書式

ata_cmd

- 説明

ATA コマンドリストを表示します。

表示されたコマンド名は“recv_cmd” コマンドで指定できます。

7 : 仕様

7-1 : 仕様

型式	ERHD 151
動作環境	5～50℃ 0%～80% (但し、結露無きこと)
保存環境	0～50℃ 0%～80% (但し、結露無きこと)
電源	AC 100V 50/60Hz
スイッチ	SATA/PATA モード切替スイッチ x1 POWER スイッチ x1 リセット スイッチ x1
LED	RUN 緑 HOST 緑 FAIL 赤 TRIG 赤 SATA 緑 PATA 緑 POWER 緑 アクセス 赤
HOST I/F	Serial ATA 1.0 x1 Parallel ATA x1 MAX UDMA 5 ATA 7 48bit LBA 対応
SLAVE I/F	Serial ATA 1.0 x1 Parallel ATA x1 UDMA 5 以上のみ対応 ATA 7 48bit LBA 対応
外部通信 IF	RS232C x 1 115200bps ストレート 3 線式

7-2 : HOST I/F 詳細 (リア SATA/PATA)

SATA モード : SATA 1.0 HOST コントローラの接続が可能です。

PATA モード : PATA の HOST コントローラの接続が可能です。

PATA の場合は DEVICE 0 限定動作となります。

DEVICE 1 モードの動作ならびに DEVICE 0/1 のパラレル接続は対応しておりません。

SATA/PATA のどちらか選択されたモードのみで動作可能です。

SATA、PATA 両コネクタに同時接続はできません。

また、後段の HDD 種別 (SATA/PATA) も HOST と同じ種別に限定されます。

ATA/ATAPI-7 準拠 UDMA 0~5 / MDMA 0~2 / PIO 0~4 をサポートします。

48bit LBA をサポートします。

7-3 : SLAVE I/F 詳細 (内部 SATA/PATA)

SATA モード : SATA 1.0 HDD の接続が可能です。

PATA モード : PATA HDD の接続が可能です。

PATA の場合は DEVICE 0 限定となります。

DEVICE 1 の接続ならびに、DEVICE 0/1 のパラレル接続は対応しておりません。

HOST の IF モードと同様のデバイスのみ接続可能です。

使用/未使用にかかわらず、PATA と SATA の HDD を同時に接続することはできません。

エラーを発生させない場合、通常 HOST からのコマンドに関しては本 IF に接続された HDD にコマンドを発行し、その応答結果が HOST に返されます。

HOST 側の UDMA 転送モードによらず、SLAVE 側は UDMA 5 で動作します。

HOST 側の MDMA 転送モードによらず、SLAVE 側は MDMA 2 で動作します。

ATA/ATAPI-7 準拠 UDMA 5 / MDMA 2 / PIO 0~4 をサポートします。

48bit LBA をサポートします。

SLAVE に接続されるデバイスは上記仕様を満たしているデバイスを接続してください。

また、identify device の DMA 動作モードは本機により模擬され HOST 側に動作モードが返信されます。

SLAVE デバイスが不良によりエラーを発生した場合は、模擬内容に関わらず HOST にエラーを返信します。

7-4 : 外部通信 I/F 詳細 (リア SERIAL)

RS232C 3 線式です。

HOST に発行するエラーパラメータ等は、本ポートを使用してオペレーション用端末と通信します。

8：トラブルシューティング

8-1：電源 OFF ができない

本機は、HDD 保護のため、電源 OFF 時に SLAVE の HDD を先に停止し本機の電源を落とします。

HDD に異常が発生した場合、または、本機が何らかの問題によりハングアップし電源 OFF ができなくなった場合は、

a) 電源スイッチを 5 秒以上長押しします。

それでも電源 OFF ができない場合は、

a) リアのリセットスイッチを押し、

b) 再度電源スイッチを押して下さい。

8-2：FAIL LED が点灯する

本機の電源を落とし以下の 3 点を確認して下さい。

a) フロントのモードスイッチがありますか？

b) SLAVE HDD が正常に接続されていますか？

c) SLAVE の HDD の UDMA モードを確認して下さい。

本機への接続可能な HDD はデータの転送速度が UDMA 5 でなければなりません。

転送速度の確認方法は、DISK を PC などに直接接続し BIOS でご確認頂けます。

BIOS の立ち上げ方と操作方法は、お使いの PC の取扱説明書をご覧ください。

* HOST 側の転送速度は UDMA 0~5 のすべてに対応しております。

8-3：操作端末から正しく文字を入力できない

本機へ HOST からのアクセスが集中している場合、操作端末からの情報を正確に受信できない場合があります。

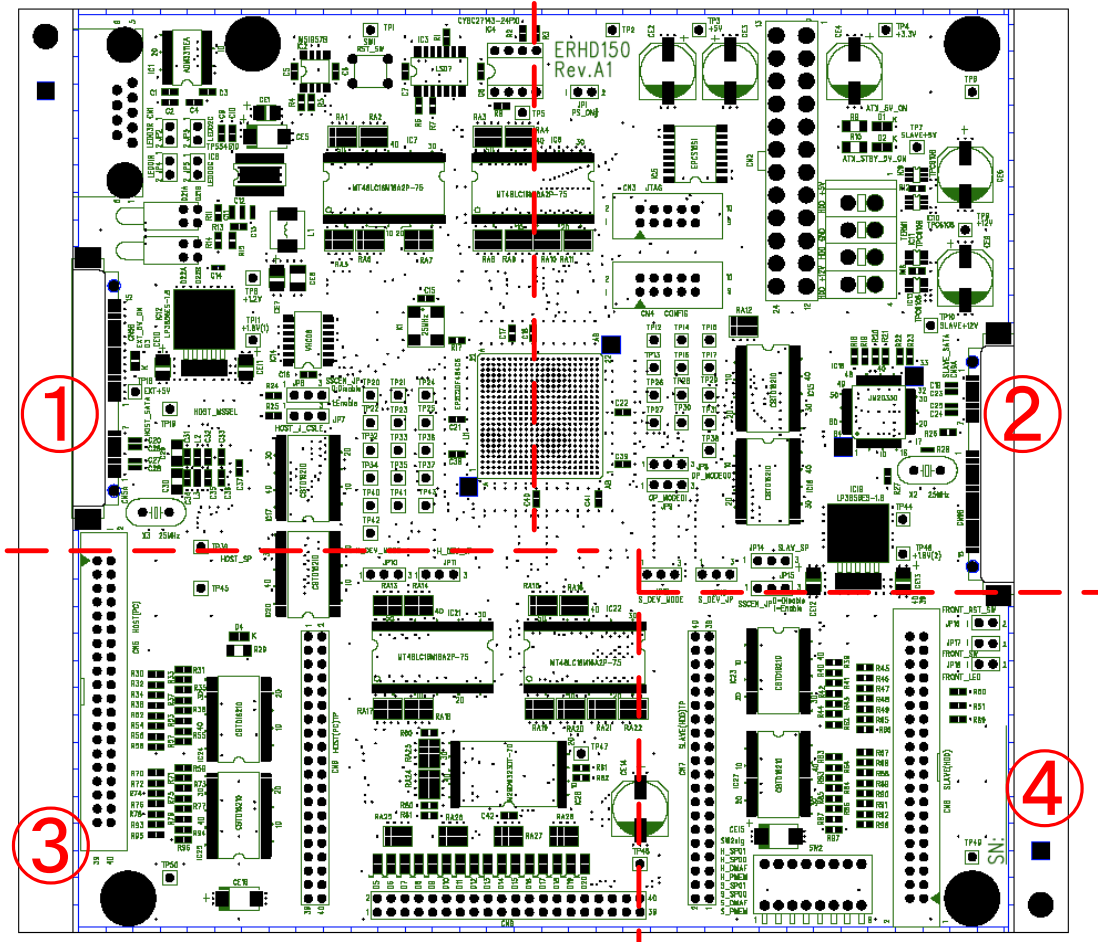
a) フロントの HOST LED が消灯、または、点滅している場合に再度行って下さい。

8-4：スクリプトファイルをダウンロードするとエラーと表示される

a) スクリプトファイルに間違いがないかも一度確認してみてください。

b) 上記 8-3 の対処を行って下さい。

付録 A : 基板外観図



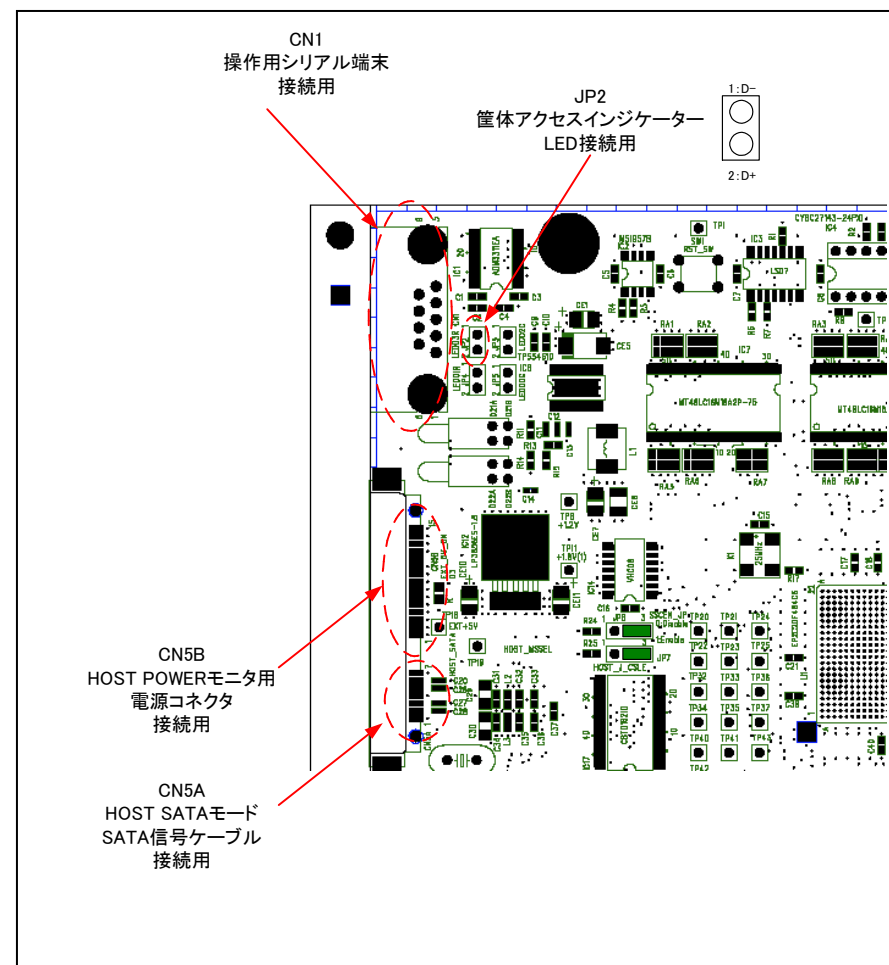
付録 B : 基板セッティング

① : 1/4

CN1 : 本機を操作するためのシリアル通信ポートです。
RS232C ストレートケーブルで接続します。

CN5A : 本機を SATA モードで動かす場合、
本コネクタに SATA 信号ケーブルを接続し、
HOST に接続します。

CN5B : 本機と HOST の GND レベルあわせる為に
HOST の POWER 状態をモニタする為に
HOST の電源ケーブルを接続します。
「すべての動作モード」で使用します。
4PIN-SATA 電源変換ケーブルを使用します。

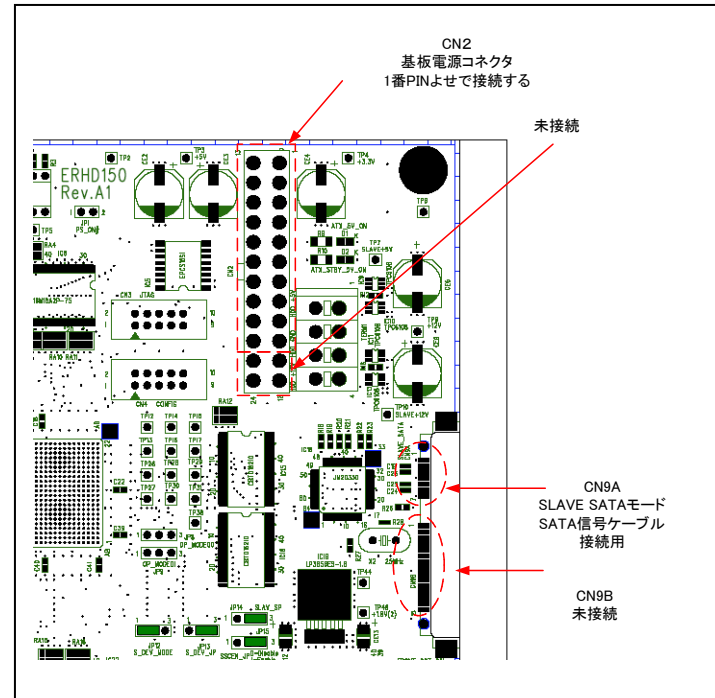


② : 2/4

CN2 : 本機の電源コネクタです。
1 番 pin 詰めで接続します。

CN9A : 本機を SATA モードで動かす場合、
本コネクタに SATA 信号ケーブルを接続し
SLAVE に接続します。
SLAVE への電源供給は、本機の電源ユニットから
直接 4pin-SATA 電源変換ケーブルを使用して
接続します。

CN5B : 使用しません。
基板上也結線されていません。

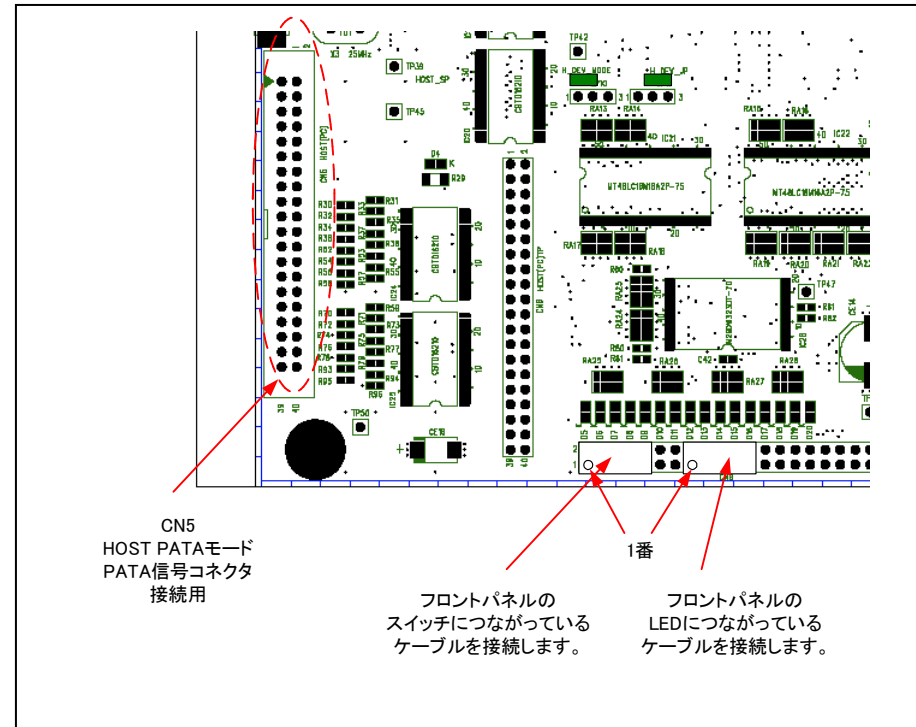


③ : 3/4

CN5 : 本機を PATA モードで動かす場合、
本コネクタに
80 芯 3.5 インチ用 PATA ケーブルの
MASTER 用コネクタを接続し
HOST に接続します。

HOST のコネクタがメス端子の場合
本機に接続したケーブルの先端 (HOST 用) に
オスメス変換基板を接続します。

HOST が 2.5 インチ I/F の場合
本機に接続したケーブルの先端に
3.5-2.5 変換コネクタを接続します。



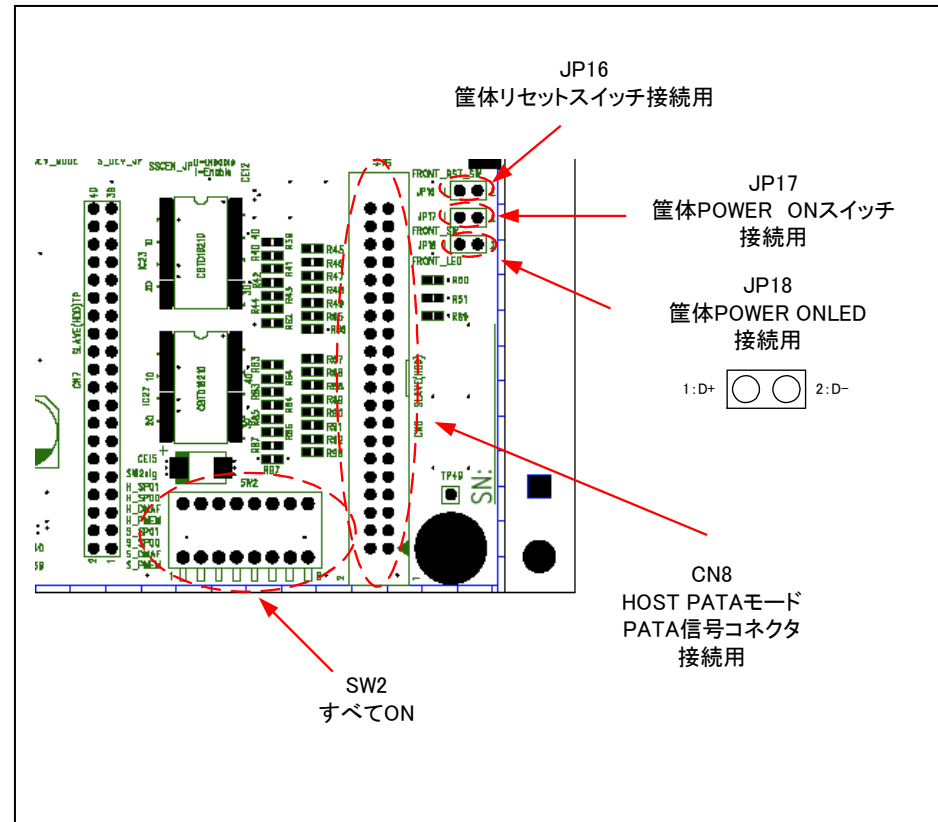
④ : 4/4

CN8 : 本機を PATA モードで動かす場合、
本コネクタに
80 芯 3.5 インチ用 PATA ケーブルの
HOST 用コネクタ (青色) を接続し
SLAVE に接続します。
SLAVE への電源供給は、本機の電源ユニットから
直接接続します。

SLAVE が 2.5 インチ I/F の場合
本機に接続したケーブルの先端に
3.5-2.5 変換コネクタを接続します。

SW2 : すべて ON (スイッチを下側に)

*その他の JP は図面の緑枠で塗りつぶしている状態が
出荷時設定で、設定の変更は行う必要はありません。



付録 C : サンプルスクリプト

Read コマンドで UNC エラーを模擬

```
script
stop
delete_all                # 以前のトリガーを全て削除します。

trigger -c1 2             # -c1:受信時, -c2:応答時, -c3:リセット時
    recv_cmd(read_sector) # Read 系のコマンドを条件にします。
    recv_cmd(read_sector_obsolete)
    recv_cmd(read_sector_ext)
    recv_cmd(read_multiple)
    recv_cmd(read_multiple_ext)
    recv_cmd(read_long)
    recv_cmd(read_long_obsolete)
    recv_cmd(read_dma)
    recv_cmd(read_dma_obsolete)
    recv_cmd(read_dma_ext) @read_unc_err # setact の名前と同じにします。
end_cond
    setact read_unc_err # @の名前と同じにします。
    error UNC # エラー指定(CRC, UNC, ABRT, IDNF, TKONF, AMNF)
end_action
end_trigger
cont 2 1 # 上記の TrigNo. 2 に繰り返し回数を指定します。
# ignore 2 1 # 上記の TrigNo. 2 に最初に無視する回数を指定します。
trigger -l # 登録したトリガーを表示します。
run # 模擬動作をスタートします。
end_script
```


Write コマンドで CRC エラーを模擬

```
script
stop
delete_all                # 以前のトリガーを全て削除します。

trigger -c2 3             # -c1:受信時, -c2:応答時, -c3:リセット時
    recv_cmd(write_sector)      # Write 系のコマンドを条件にします。
    recv_cmd(write_sector_obsolete)
    recv_cmd(write_sector_ext)
    recv_cmd(write_multiple)
    recv_cmd(write_multiple_ext)
    recv_cmd(write_long)
    recv_cmd(write_long_obsolete)
    recv_cmd(write_dma)
    recv_cmd(write_dma_obsolete)
    recv_cmd(write_dma_ext) @write_crc_err # setact の名前と同じにします。
end_cond
setact write_crc_err      # @の名前と同じにします。
    error crc             # エラー指定(CRC, UNC, ABRT, IDNF, TKONF, AMNF)
end_action
end_trigger
cont 3 1                  # 上記の TrigNo. 3 に繰り返し回数を指定します。
# ignore 3 1             # 上記の TrigNo. 3 に最初に無視する回数を指定します。
trigger -l                # 登録したトリガーを表示します。
run                       # 模擬動作をスタートします。
end_script
```

遅延処理模擬

```
script
stop
delete_all                # 以前のトリガーを全て削除します。

trigger -c1 0             # -c1:受信時, -c2:応答時, -c3:リセット時
    time_expire(0) @cmd_delay_emu# setact の名前と同じにします。
    end_cond
    setact cmd_delay_emu  # @の名前と同じにします。
        delay 1000       # 遅延時間はミリ秒で指定します。
    end_action
end_trigger
cont 0 1                  # 上記の TrigNo. 0 に繰り返し回数を指定します。
# ignore 0 1             # 上記の TrigNo. 0 に最初に無視する回数を指定します。
trigger -l                # 登録したトリガーを表示します。
run                       # 模擬動作をスタートします。
end_script
```

複合エラー模擬

```
stop
delete_all                # 以前のトリガーを全て削除します。

trigger -c1 8             # -c1:受信時, -c2:応答時, -c3:リセット時
    recv_cmd(read_sector) # Read系のコマンドを条件にします。
    recv_cmd(read_sector_obsolete)
    recv_cmd(read_sector_ext)
    recv_cmd(read_multiple)
    recv_cmd(read_multiple_ext)
    recv_cmd(read_long)
    recv_cmd(read_long_obsolete)
    recv_cmd(read_dma)
    recv_cmd(read_dma_obsolete)
    recv_cmd(read_dma_ext) @delay_unc_err # setactの名前と同じにします。
end_cond
setact delay_unc_err     # @の名前と同じにします。
    delay 5000           # 遅延時間はミリ秒で指定します。
    error unc            # エラー指定(CRC, UNC, ABRT, IDNF, TKONF, AMNF)
end_action
end_trigger
cont 8 1                 # 上記の TrigNo. 8 に繰り返し回数を指定します。
ignore 8 10              # 上記の TrigNo. 8 に最初に無視する回数を指定します。
trigger -l               # 登録したトリガーを表示します。
run                       # 模擬動作をスタートします。
end_script
```

HDD ハングアップ (全コマンド無視)

```
script
stop
delete_all                # 以前のトリガーを全て削除します。
trigger -c1 0             # -c1:受信時, -c2:応答時, -c3:リセット時
    cmd_count_gt(20) @busy_err # setact の名前と同じにします。
    end_cond
    setact busy_err        # @の名前と同じにします。
    busy                   # Halt 状態
    end_action
end_trigger
#cont 0 1                 # 上記の TrigNo. 0 に繰り返し回数を指定します。
# ignore 0 1              # 上記の TrigNo. 0 に最初に無視する回数を指定します。

# リセットを受信したら Halt 状態を解除するトリガー
trigger -c3 1             # -c1:受信時, -c2:応答時, -c3:リセット時
    cmd_count_gt(20) @busy_err_unlock # setact の名前と同じにします。
    end_cond
    setact busy_err_unlock # @の名前と同じにします。
    disable 0              # TrigNo. 0 を無効にします。
    busy_unlock
    end_action
end_trigger
cont 1 1                  # 上記の TrigNo. 1 に繰り返し回数を指定します。
# ignore 1 1              # 上記の TrigNo. 1 に最初に無視する回数を指定します。

count_reset               # トリガー用カウンタをリセットします。
trigger -l                # 登録したトリガーを表示します。
run                       # 模擬動作をスタートします。
end_script
```

付録 D : ATA コマンド対応表

本機が受け付けるコマンドのリストです。受け付けられたコマンドは SLAVE に対し実行されその結果を HOST に応答します。

下表の非サポートの ATA コマンドはアボートエラーを HOST に応答します。

下表の C1/C2 の欄は本機へのエラー模擬設定の可/不可です。

下表に記載の無い ATA コマンドも ID で指定することにより設定可能です。

ID	Name	プロトコル	サポート	C1	C2	備考
00	NOP	ND	○	○	○	
03	CFA REQUEST EXTENDED ERROR	ND	○	○	○	
08	DEVICE RESET	DR	×	○	×	
10	Recalibrate Obsolete	ND	○	○	○	
20	READ SECTOR	PI	○	○	×	
21	READ SECTOR Obsolete	PI	○	○	×	
22	READ LONG	PI	○	○	×	ECC データは改ざんしません。
23	READ LONG Obsolete	PI	○	○	×	ECC データは改ざんしません。
24	READ SECTOR EXT	PI	○	○	×	
25	READ DMA EXT	DM	○	○	○	
26	READ DMA QUEUED EXT	DMQ	×	○	×	
27	READ NATIVE MAX ADDRESS EXT	ND	○	○	○	
29	READ MULTIPLE EXT	PI	○	○	×	
2A	READ STREAM DMA EXT	DM	×	○	○	
2B	READ STREAM EXT	PI	○	○	×	
2F	READ LOG EXT	PI	○	○	×	
30	WRITE SECTOR	PO	○	○	○	
31	WRITE SECTOR Obsolete	PO	○	○	○	
32	WRITE LONG	PO	○	○	○	ECC データは改ざんしません。
33	WRITE LONG Obsolete	PO	○	○	○	ECC データは改ざんしません。
34	WRITE SECTOR EXT	PO	○	○	○	
35	WRITE DMA EXT	DM	○	○	○	
36	WRITE DMA QUEUED EXT	DMQ	×	○	×	
37	SET MAX ADDRESS EXT	ND	○	○	○	
39	WRITE MULTIPLE EXT	PO	○	○	○	
3A	WRITE STREAM DMA EXT	DM	×	○	×	

3B	WRITE STREAM EXT	PO	○	○	○	
3C	WRITE VERIFY Obsolete	ND	○	○	○	
3D	WRITE DMA FUA EXT	DM	×	○	○	
3E	WRITE DMA QUEUED FUA EXT	DMQ	×	○	×	
3F	WRITE LOG EXT	PO	○	○	○	
40	READ VERIFY SECTOR	ND	○	○	○	
41	READ VERIFY SECTOR Obsolete	ND	○	○	○	
42	READ VERIFY SECTOR EXT	ND	○	○	○	
51	CONFIGURE STREAM	ND	○	○	○	
70	SEEK Obsolete	ND	○	○	○	
87	CFA TRANSLATE SECTOR	PI	○	○	×	
90	EXECUTE DEVICE DIAGNOSTIC	DD	○	○	○	
91	INITIALIZE DEVICE PARAMETER	ND	○	○	○	
92	DOWNLOAD MICROCODE	PO	○	○	○	
94	STANDBY IMMEDIATE Obsolete	ND	×	○	×	
95	IDLE IMMEDIATE Obsolete	ND	×	○	×	
96	STANDBY Obsolete	ND	×	○	×	
97	IDLE Obsolete	ND	×	○	×	
98	CHECK POWER MODE Obsolete	ND	×	○	×	
99	SLEEP Obsolete	ND	×	○	×	
A0	PACKET	P	×	○	×	
A1	IDENTIFY PACKET DEVICE	PI	×	○	×	
A2	SERVICE	P/DMQ	×	○	×	
B0	SMART Disable Operations	ND	○	○	○	
B0	SMART Enable Operations	ND	○	○	○	
B0	SMART Enable/Disable Attribute Auto save	ND	○	○	○	
B0	SMART Execute Off-line Immediate	ND	○	○	○	
B0	SMART Read Attribute Values	PI	○	○	○	本機によるエラーは含まれません。
B0	SMART Read Log Sector	PI	○	○	○	本機によるエラーは含まれません。
B0	SMART Return Status	ND	○	○	○	
B0	SMART Write Log Sector	PO	○	○	○	
B1	DEVICE CONFIGURATION	-	×	○	×	
C0	CFA ERASE SECTORS	ND	○	○	○	
C4	READ MULTIPLE	PI	○	○	×	
C5	WRITE MULTIPLE	PO	○	○	○	

C6	SET MULTIPLE MODE	ND	○	○	○	
C7	READ DMA QUEUED	DMQ	×	○	×	
C8	READ DMA	DM	○	○	○	
C9	READ DMA Obsolete	DM	○	○	○	
CA	WRITE DMA	DM	○	○	○	
CB	WRITE DMA Obsolete	DM	○	○	○	
CC	WRITE DMA QUEUED	DMQ	×	○	×	
CD	CFA WRITE MULTIPLE WITHOUT ERASE	PO	○	○	○	
CE	WRITE MULTIPLE FUA EXT	PO	○	○	○	
D1	CHECK MEDIA CARD TYPE	ND	○	○	○	
DA	GET MEDIA STATUS	ND	○	○	○	
DE	MEDIA LOCK	ND	○	○	○	
DF	MEDIA UNLOCK	ND	○	○	○	
E0	STANDBY IMMEDIATE	ND	○	○	○	
E1	IDLE IMMEDIATE	ND	○	○	○	
E2	STANDBY	ND	○	○	○	
E3	IDLE	ND	○	○	○	
E4	READ BUFFER	PI	○	○	×	
E5	CHECK POWER MODE	ND	○	○	○	
E6	SLEEP	ND	○	○	○	
E7	FLUSH CACHE	ND	○	○	○	
E8	WRITE BUFFER	PO	○	○	○	
EA	FLUSH CACHE EXT	ND	○	○	○	
EC	IDENTIFY DEVICE	PI	○	○	×	
ED	MEDIA EJECT	ND	○	○	○	
EF	SET FEATURES	-	○	○	○	
F1	SECURITY SET PASSWORD	PO	○	○	○	
F2	SECURITY UNLOCK	PO	○	○	○	
F3	SECURITY ERASE PREPARE	ND	○	○	○	
F4	SECURITY ERASE UNIT	PO	○	○	○	
F5	SECURITY FREEZE LOCK	ND	○	○	○	
F6	SECURITY DISABLE PASSWORD	PO	○	○	○	
F8	READ NATIVE MAX ADDRESS	ND	○	○	○	
F9	SET MAX ADDRESS	ND	○	○	○	
**	その他	-	×	○	×	アボートエラーを応答します。

プロトコル

ND	: Non-data コマンド
PI	: PIO data-in コマンド
PO	: PIO data-out コマンド
DM	: DMA コマンド
DMQ	: DMA QUEUED コマンド
DR	: Device Reset コマンド
DD	: Execute Device Diagnostic コマンド
P	: PACKET コマンド

付録 E : DEBUG コマンド

本機には内部テスト用に DEBUG コマンドが用意されています。

fwup

ファームウェアのアップデートをします。

引数 : 無し

hddreset

SlaveHDD にソフトウェアリセットをかけます。

引数 : 無し

hddhwreset

SlaveHDD にハードウェアリセットをかけます。

引数 : 無し

identify

SlaveHDD に Identify コマンドを発行し、ERHD で保持している Identify データを更新します。

表示は HostPC に返す Identify データ、SlaveHDD から受け取った Identify データの順に表示されます。

引数 : 無し

identinfo

ERHD で保持している Identify データを表示します。

表示は HostPC に返す Identify データ、SlaveHDD から受け取った Identify データの順に表示されます。

引数 : 無し

setfeatures

SetFeatures コマンドの NonData プロトコルのサブコマンドを実行します。

引数 : 第一引数 = feature レジスタ値 (サブコマンド)

第二引数 = sector count レジスタ値 (設定値)

引数無しで実行するとコマンドヘルプが表示されます。

setmulti

SetMultiple コマンドを実行します。

引数 : 第一引数 = ブロック数

引数無しで実行するとコマンドヘルプが表示されます。

hddread

SlaveHDD からデータを読み込みます。

引数 : 第一引数 = ATA コマンドタイプ

第二引数 = 開始 LBA

第三引数 = セクター数

引数無しで実行するとコマンドヘルプが表示されます。

hddwrite

SlaveHDD にデータを書き込みます。

引数 : 第一引数 = ATA コマンドタイプ

第二引数 = 開始 LBA

第三引数 = セクター数

引数無しで実行するとコマンドヘルプが表示されます。

mdword

Word オフセットでメモリをダンプします。

引数 : 第一引数 = メモリアドレス

第二引数 = 表示数

引数無しで実行するとコマンドヘルプが表示されます。

mdlong

Long オフセットでメモリをダンプします。

引数 : 第一引数 = メモリアドレス

第二引数 = 表示数

引数無しで実行するとコマンドヘルプが表示されます。

mwlong

Long オフセットでメモリを書き込みます。

引数 : 第一引数 = メモリアドレス

第二引数 = 設定値

引数無しで実行するとコマンドヘルプが表示されます。

mcheck

データ用 SDRAM をメモリチェックします。

引数 : 無し

mfill

メモリフィル

引数 : 第一引数 = データパターン

第二引数 = メモリアドレス

第三引数 = 書き込み数

引数無しで実行するとコマンドヘルプが表示されます。

mcomp

メモリコンペア

引数 : 第一引数 = メモリアドレス 1

第二引数 = メモリアドレス 2

第三引数 = 比較数

引数無しで実行するとコマンドヘルプが表示されます。

sysinfo

システム情報を表示します。詳細なバージョン情報や電源 ON 時のシグネチャなどを表示します。

引数 : 無し

timer

タイマーの誤差計測

引数 : 第一引数 = 待ち時間

action_test

エラー模擬登録のテストルーチン呼び出し

引数 : 無し

logtest

ログ登録のテストルーチン呼び出し

引数 : 無し