

MT1000A ネットワークマスタプロ
OTDRモジュール
リモートスクリプティング取扱説明書

第8版

アンリツ株式会社

管理番号: M-W3859AW-8.0

-
- 製品を適切・安全にご使用いただくために、製品をご使用になる前に、本書を必ずお読みください。
 - 本書に記載以外の各種注意事項は、「MT1000A ネットワークマスタプロ OTDRモジュール取扱説明書」(M-W3810AW)に記載の事項に準じますので、そちらをお読みください。
 - 本書は製品とともに保管してください。

国外持出しに関する注意

本製品は日本国内仕様であり、外国の安全規格などに準拠していない場合もありますので、国外へ持ち出して使用された場合、当社は一切の責任を負いかねます。

本製品および添付マニュアル類は、輸出および国外持ち出しの際には、「外国為替及び外国貿易法」により、日本国政府の輸出許可や役務取引許可を必要とする場合があります。

また、米国の「輸出管理規則」により、日本からの再輸出には米国政府の再輸出許可を必要とする場合があります。

本製品や添付マニュアル類を輸出または国外持ち出しする場合は、事前に必ず当社の営業担当までご連絡ください。

輸出規制を受ける製品やマニュアル類を廃棄処分する場合は、軍事用途等に不正使用されないように、破碎または裁断処理していただきますようお願い致します。

2016年6月13日 (初版)
2021年3月5日 (第8版)

Copyright ©2016-2021, ANRITSU CORPORATION.

- ・予告なしに本書の製品操作・取り扱いに関する内容を変更することがあります。
- ・許可なしに本書の一部または全部を転載・複製することを禁じます。

Printed in Japan

はじめに

この取扱説明書は、ネットワークマスタプロの SCPI (Standard Commands for Programmable Instruments) コマンドについて説明しています。

注: 本書はネットワークマスタプロバージョン 11.00 のSCPIコマンドについて説明しています。

本書のいくつかのコマンドまたはクエリは、特定のハードウェアやソフトウェアオプションのインストールが必要です。これらオプションは購入が必要です。

本取扱説明書の表記法は次の規格を使用しています:

- IEEE: Std 488.2-1992
- SCPI: VERSION 1999.0 (SCPI Consortium)

目次

1	概要	9
1.1	Ethernetベースのリモート制御	9
1.1.1	ケーブルの接続	9
1.1.2	Ethernetリモート制御の設定	10
1.1.3	通信バッファ	10
1.2	プログラムメッセージ	11
1.2.1	プログラムメッセージユニット	11
1.2.2	プログラムヘッダー	12
1.2.3	プログラムデータ	12
1.2.4	プログラムメッセージターミネータ	14
1.2.5	複合プログラムメッセージ	14
1.2.6	シーケンシャル実行	14
1.3	レスポンスメッセージ	15
1.3.1	レスポンスデータ	15
1.3.2	レスポンスメッセージターミネータ	17
1.3.3	プロンプト	17
1.4	ステータス	18
1.4.1	IEEE488.2標準のステータスとSCPIにより定義されたレジスタ	18
1.4.2	ネットワークマスタ固有のステータスレジスタ	22
1.4.3	ステータスレジスタの読み取り, 書き込み, クリア	25
1.5	コントローラの例	27
1.5.1	PuTTY	27
1.6	定義	30
1.6.1	NaN (Not a Number)	30
1.6.2	→ Right Arrow	30
1.6.3	データビット (DB)	30
1.6.4	ポート番号 (論理ポート)	30
2	SCPI 適合情報	33
2.1	SCPIのバージョン	33
2.2	IEEE 488.2 必須コマンド	34
2.2.1	*CLS	34
2.2.2	*ESE	34
2.2.3	*ESR?	35
2.2.4	*IDN?	35
2.2.5	*OPC	35
2.2.6	*RST	36
2.2.7	*SRE	36
2.2.8	*STB?	37
2.2.9	*TST?	37
2.2.10	*WAI	37
2.3	SCPI System サブシステムコマンド	38
2.3.1	SYSTem:VERSion?	38
2.3.2	SYSTem:ERRor[:NEXT]?	38
2.3.3	SYSTem:ERRor:ADDitional[:MESSAge]	38
2.3.4	SYSTem:DATE	39
2.3.5	SYSTem:TIME	39
2.3.6	SYSTem:REBoot	39
2.3.7	SYSTem:GPS:NSATellites?	40

2.3.8	SYSTem:GPS:TIME?	40
2.3.9	SYSTem:GPS:LOCation?	40
2.3.10	SYSTem:GPS:POWER	40
2.3.11	SYSTem:GPS:POWER?	40
2.3.12	SYSTem:GPS:SYNCmode	40
2.3.13	SYSTem:GPS:SYNCmode?	41
2.3.14	SYSTem:GPS:ANTNpower	41
2.3.15	SYSTem:GPS:ANTNpower?	41
2.3.16	SYSTem:GPS:PWRElapsed?	41
2.3.17	SYSTem:GPS:LOCElapsed?	41
2.3.18	SYSTem:GPS:SYNElapsed?	42
2.3.19	SYSTem:COMMunicate:TERMinator	42
2.3.20	SYSTem:PROMpt	42
2.3.21	SYSTem:LOCal:CONTRol	42
2.3.22	SYSTem:NCONnectivity:COMunication:AUTH:TYPE	43
2.3.23	SYSTem:NCONnectivity:COMunication:BASic	43
2.3.24	SYSTem:NCONnectivity:COMunication:TURi	43
2.3.25	SYSTem:NCONnectivity:COMunication:SET	44
2.3.26	SYSTem:NCONnectivity:COMunication:LRETurn:RRESponse?	44
2.3.27	SYSTem:NCONnectivity:COMunication:LRETurn:RStatus?	44
2.3.28	SYSTem:NCONnectivity:COMunication:LRETurn:ERRor?	45
2.3.29	SYSTem:WAIT[:IDLE]	45
2.3.30	SYSTem:WAIT:DURation	45
2.4	SCPI Instrument サブシステムコマンド	46
2.4.1	アプリケーションサーバへの接続	46
2.4.2	複数アプリケーションの接続	46
2.4.3	マルチユーザでの接続	46
2.4.4	アプリケーションサーバの強制終了	46
2.4.5	INSTrument:STARt[:DEFault]	47
2.4.6	INSTrument:STARt:LAST	49
2.4.7	INSTrument:STARt:GUI	49
2.4.8	INSTrument:TERMinate	49
2.4.9	INSTrument:TERMinate:FORCe	49
2.4.10	INSTrument:COUNt?	50
2.4.11	INSTrument:CATalog?	50
2.4.12	INSTrument:STATe?	50
2.4.13	INSTrument:CONNect	50
2.4.14	INSTrument:CONNect:ALL	51
2.4.15	INSTrument:CONNect[:CATalog]?	51
2.4.16	INSTrument:DISConnect	51
2.4.17	INSTrument[:SELect]	51
2.4.18	INSTrument:ERRor[:NEXT]?	52
2.4.19	INSTrument:PORT?	52
2.4.20	INSTrument:PORT:FREE?	52
2.4.21	INSTrument:PORT:CATalog?	53
2.4.22	INSTrument:MODule:CATalog?	53
2.4.23	INSTrument:CTRL:NAME?	53
2.4.24	INSTrument:CTRL:SN?	53
2.4.25	INSTrument:CTRL:TRT?	53
2.4.26	INSTrument:CTRL:OPTion:CATalog?	53
2.4.27	INSTrument:MODule<Md>:NAME?	54
2.4.28	INSTrument:MODule<Md>:SN?	54
2.4.29	INSTrument:MODule<Md>:TRT?	54
2.4.30	INSTrument:MODule<Md>:OPTion:CATalog?	54
2.5	SCPI Status サブシステムコマンド	55
2.5.1	STATus:OPERation[:EVENT]?	55
2.5.2	STATus:OPERation:CONDition?	55
2.5.3	STATus:OPERation:ENABle	55
2.5.4	STATus:OPERation:PTRansition	56

2.5.5	STATus:OPERation:NTRansition	56
2.5.6	STATus:QUEStionable[:EVENT]?	57
2.5.7	STATus:QUEStionable:CONDition?	57
2.5.8	STATus:QUEStionable:ENABle	57
2.5.9	STATus:QUEStionable:PTRansition	58
2.5.10	STATus:QUEStionable:NTRansition	58
2.5.11	STATus:PORT[:EVENT]?	59
2.5.12	STATus:PORT:CONDition?	59
2.5.13	STATus:PORT:ENABle	59
2.5.14	STATus:PORT:PTRansition	59
2.5.15	STATus:PORT:NTRansition	60
2.5.16	STATus:PRESet	60
2.6	Mass Memory サブシステムコマンド	61
2.6.1	MMEMory:LOAD	61
2.6.2	MMEMory:STORe:STATe	61
2.6.3	MMEMory:STORe:DATA	62
2.6.4	MMEMory:DELeTe	62
2.6.5	MMEMory:DATA?	62
2.6.6	MMEMory:COPI	62
2.6.7	MMEMory:MOVE	63
2.6.8	MMEMory:INFO?	63
2.6.9	MMEMory:CATalog?	63
2.6.10	MMEMory:DCATalog?	63
2.6.11	MMEMory:MDIRectory	64
2.6.12	MMEMory:RDIRectory	64
2.6.13	MMEMory:SAVE	64
3	Standard OTDRアプリケーションコマンド	65
3.1	測定条件	65
3.1.1	OTDR:SOURce:PORT	65
3.1.2	OTDR:SOURce:TEST	65
3.1.3	OTDR:SOURce:WAVelength:AVAIlable?	66
3.1.4	OTDR:SOURce:WAVelength	66
3.1.5	OTDR:SOURce:RANge:AVAIlable?	66
3.1.6	OTDR:SOURce:RANge	66
3.1.7	OTDR:SOURce:RESO:AVAIlable?	67
3.1.8	OTDR:SOURce:RESO	67
3.1.9	OTDR:SOURce:PULSe:AVAIlable?	67
3.1.10	OTDR:SOURce:PULSe	67
3.1.11	OTDR:SOURce:AVERages:TIME	68
3.2	IOR/BSC	69
3.2.1	OTDR:SENSe:FIBer:IOR	69
3.2.2	OTDR:SENSe:FIBer:BSC	69
3.3	スプリッタ	70
3.3.1	OTDR:SOURce:SPLitter	70
3.4	ステータス	71
3.4.1	OTDR:SENSe:AVERages:TIME?	71
3.4.2	OTDR:SENSe:TRACe:READY?	71
3.5	測定機能	72
3.5.1	OTDR:SENSe:CONCheck	72
3.5.2	OTDR:SENSe:CONState?	72
3.5.3	OTDR:CONTinue	72
3.5.4	OTDR:SENSe:LIVCheck	72
3.5.5	OTDR:SENSe:FIBCheck	73
3.6	解析	74
3.6.1	OTDR:SENSe:PATCH:LAUnch	74
3.6.2	OTDR:SENSe:PATCH:LAUnch:CORRelation	74
3.6.3	OTDR:SENSe:PATCH:RECeive	74
3.6.4	OTDR:SENSe:PATCH:RECeive:CORRelation	75

3.6.5	OTDR:SENSe:ACURsor	75
3.6.6	OTDR:SENSe:BCURsor	75
3.6.7	OTDR:SENSe:LSALeft	76
3.6.8	OTDR:SENSe:LSARight	76
3.6.9	OTDR:SENSe:LOSS:MODE	77
3.6.10	OTDR:SENSe:ORL:MODE	77
3.6.11	OTDR:SENSe:ANALyze:PARAmeters	78
3.7	測定波形	79
3.7.1	OTDR:TRACe:PARAmeters?	79
3.7.2	OTDR:TRACe:ANALyze	79
3.7.3	OTDR:TRACe:ANALyze:ORL	79
3.7.4	OTDR:TRACe:MDLOss?	79
3.7.5	OTDR:TRACe:EELOss?	79
3.7.6	OTDR:TRACe:LOAD:TEXT?	80
3.7.7	OTDR:TRACe:HOFFset	80
4	OLTSアプリケーションコマンド	81
4.1	ロステストセット	81
4.1.1	OLTS:SOURce:POWer:WAVelength:AVAILable?	81
4.1.2	OLTS:SOURce:POWer:WAVelength	81
4.1.3	OLTS:SOURce:AM:FREQuency:AVAILable?	81
4.1.4	OLTS:SOURce:AM:FREQuency	82
4.1.5	OLTS:SOURce:POWer:STATe	82
4.1.6	OLTS:SENSe:POWer:WAVelength:AVAILable?	82
4.1.7	OLTS:SENSe:POWer:WAVelength	82
4.1.8	OLTS:SENSe:CORRection:COLLect:ZERO	83
4.1.9	OLTS:SENSe:AVERage:COUNt	83
4.1.10	OLTS:SENSe:POWer:REFerence	83
4.1.11	OLTS:SENSe:POWer:THReshold	84
4.1.12	OLTS:FETCh:POWer	84
5	測定	85
5.1	測定の開始と終了	85
5.1.1	MEASurement:APPLication?	85
5.1.2	MEASurement:STARt	85
5.1.3	MEASurement:STOP	85
5.1.4	MEASurement:RESult:SUMMARY?	85
A	サンプルスクリプト	87
A.1	ヒント	87
A.2	OTDR測定	88
A.3	HTTP GETリクエストを送信	89
A.4	シナリオ実行結果をアップロードするパラメータを設定	90

図目次

1.1	Ethernetを使用したシステム設定	9
1.2	コネクタパネル	9
1.3	リモート制御用のTCP Portの設定	10
1.4	プログラムメッセージ構造	11
1.5	プログラムメッセージユニット	11
1.6	レスポンスメッセージ構造	15
1.7	IEEE488.2標準のステータスとSCPIで定義されたレジスタ/キュー	18

1.8	サポートされる一部のインタフェースのためのネットワークマスタ固有のステータスレジスタ	22
1.9	インタフェースのアラームおよびエラーステータスレジスタの全般的な構造	23
1.10	ポートステータスレジスタの構造	23
1.11	ネットワークマスタ固有のステータスレジスタのレジスタモデル	24
1.12	PuTTYで [Implicit CR in every LF] をオンにする	27
1.13	機器の IP Address	28
1.14	PuTTYでの宛先の指定と接続の開始 (Open)	28
1.15	PuTTYとの接続の確立	29
1.16	データビット	30
1.17	OTDRモジュールのポート番号	30
2.1	INST:STAR は自動的に 接続して、開始アプリケーションサーバを選択します	47
2.2	別のクライアントセッションを 接続する前に、最初のクライアントセッションを切断する必要があります	47
2.3	複数のアプリケーションサーバに接続している場合は、クライアントセッションはどのアプリケーションサーバにアプリケーション固有コマンドを送信するか選択します	47

Chapter 1

概要

ネットワークマスタリモート制御機能では、内蔵Ethernetサービスインタフェースがサポートされます。ソフトウェア仕様は、SCPI（Standard Commands for Programmable Instruments）バージョン1999に基づくIEEE488.2標準に準拠しています。

ネットワークマスタは、外部コントローラに接続されると自動測定機器になります。



図 1.1: Ethernetを使用したシステム設定

1.1 Ethernetベースのリモート制御

1.1.1 ケーブルの接続

Ethernetサービスインタフェースのリモート制御オプションを使用するには、Ethernetケーブルを左側パネルのEthernetコネクタに接続します。



イーサネットサービスインタフェース

図 1.2: コネクタパネル

1.1.2 Ethernetリモート制御の設定

ポート番号

TCPポート番号を変更するには（有効な範囲については表 1.1を参照），[TCP Port] フィールドに数値を入力します（図 1.3参照）。

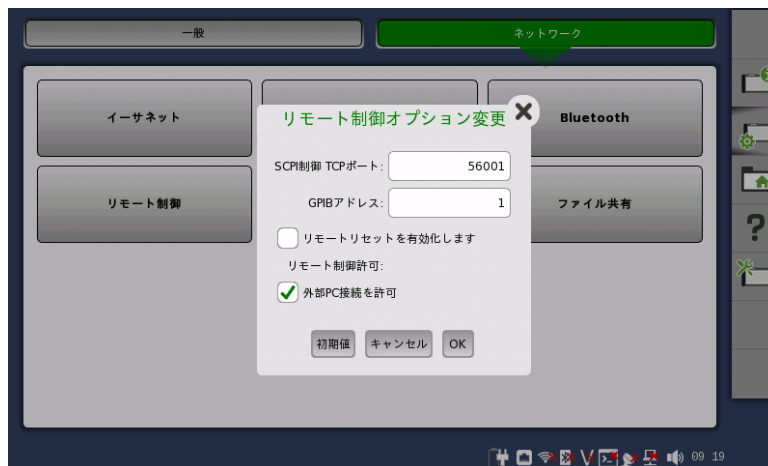


図 1.3: リモート制御用のTCP Portの設定

設定項目	説明	許容される範囲
ポート番号	TCPポート番号	1024～65535 （初 期 値: 56001）

表 1.1: 許容されるTCPポート範囲

1.1.3 通信バッファ

入出力ストリームがバッファされます。TCP受信バッファ（87380バイト）とTCP送信バッファ（16384バイト）に加えて，32エントリの共通コマンド/レスポンスバッファが2つのストリームにより共有されます。各バッファエントリには，最大4 KBの複合プログラムメッセージまたは最大64 KBのレスポンスメッセージを保持できます。

<ARBITRARY BLOCK PROGRAM DATA>として転送されるプログラムデータは内部バッファを通過しませんが，TCP受信バッファから内部ファイルシステムに直接ストリーミングされます。<DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>型のレスポンスデータと同様，内部ファイルシステムからTCP送信バッファに直接ストリーミングされます。

1.2 プログラムメッセージ

プログラムメッセージは、図 1.4に示すように、ネットワークマスタに送信されるリモートコマンドです。

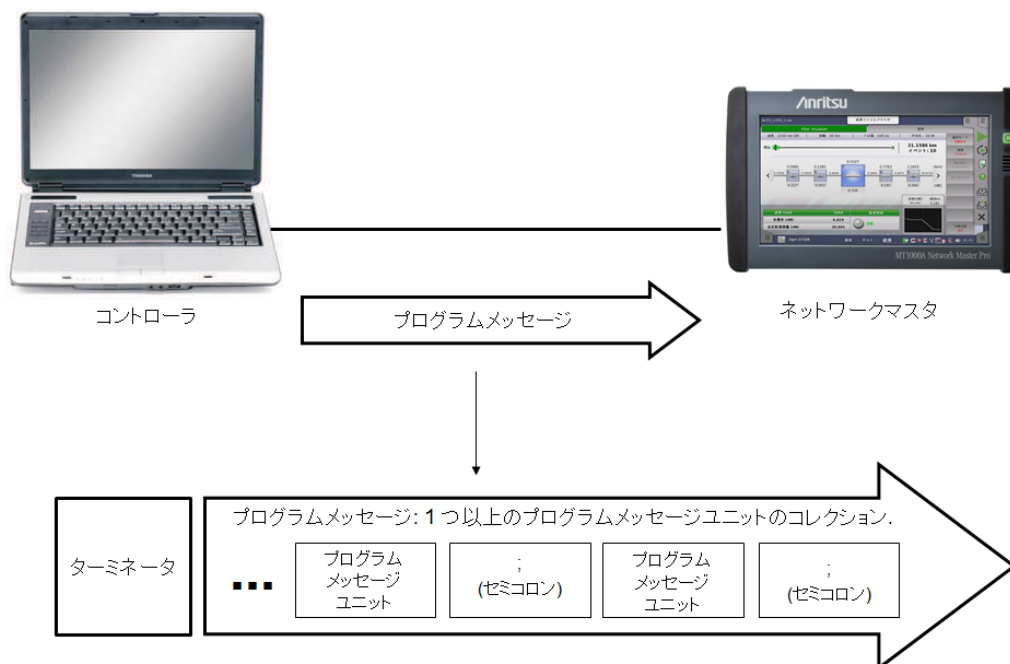


図 1.4: プログラムメッセージ構造

プログラムメッセージは、1つ以上のプログラムメッセージユニットをセミコロン (;) で区切った形で構成されます。セミコロンの前後に空白文字があっても無視されます（空白文字に意味はありません）。プログラムメッセージユニットの詳細については、1.2.1 節の「プログラムメッセージユニット」を参照してください。

プログラムメッセージがネットワークマスタに送信されると、末尾にターミネータが付加されます。ネットワークマスタは、ターミネータを検出することでプログラムメッセージを受信します。ターミネータについては、1.2.4 節の「プログラムメッセージターミネータ」を参照してください。

ネットワークマスタは、メッセージターミネータを含めて長さが最大4096文字のプログラムメッセージを処理できます。

1.2.1 プログラムメッセージユニット

プログラムメッセージユニットは、プログラムヘッダーとプログラムデータで構成されます。図 1.5を参照してください。

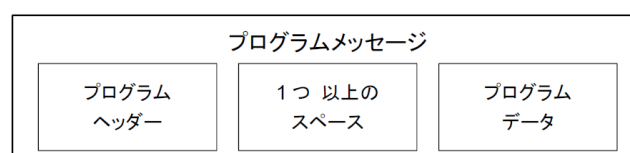


図 1.5: プログラムメッセージユニット

プログラムヘッダーとプログラムデータの間には1つ以上のスペースが必要です。ネットワークマスタは、スペースを使用してプログラムヘッダーとプログラムデータを認識します。プログラムヘッダーの前の1つ以上のスペースは無視されます。

1.2.2 プログラムヘッダー

プログラムヘッダーは、コントローラからネットワークマスタに送信されるコマンドメッセージユニットの機能を指定します。プログラムヘッダーには次の2つの種類があります。

- コマンドメッセージユニットのプログラムヘッダー。
- クエリメッセージユニットのプログラムヘッダー。コマンドメッセージユニットのヘッダーと似ていますが、常に疑問符「?」が後ろに付きます。

ネットワークマスタでは、IEEE488.2標準で定義された一般的なコマンドの一部がサポートされます。これらの一般的なコマンドは、常にアスタリスク「*」が先頭に付く点が特殊です（例: *IDN?）。他のすべてのコマンドは、「デバイス固有のコマンド」と呼ばれます。デバイス固有のコマンドは、複数の<program mnemonic>（これ以降は「ニーモニック」と呼びます）で構成されており、それぞれコロンの(:)で区切られます。

[:]<program mnemonic>[:<program mnemonic>]... 例: SYSTem:TIME

ニーモニックは、大文字と小文字で構成される文字列です。ニーモニックの大文字部分は、ニーモニックの短い形式とも呼ばれます。

- 長い形式のプログラムヘッダー: INSTrument:STARt
- 短い形式のプログラムヘッダー: INST:STAR

ネットワークマスタは、短い形式だけが送信された場合でもニーモニックを認識します。たとえば、SYSTが送信されると、ニーモニックSYSTemは通常のニーモニックとして認識されます。

このようにして、長い形式と短い形式のニーモニックを認識するために大文字と小文字が使用されます。ネットワークマスタは、プログラムヘッダーを読み取るとき大文字と小文字を区別しません。ただし、ネットワークマスタは短い形式のニーモニックか、長い形式の完全なニーモニックのみ受け入れます。したがって、SYSTeは有効なニーモニックではありません。次のプログラムヘッダーはすべて受け入れられ、同じと見なされます。

- SYSTEM:POWER:SOURCE?
- system:power:source?
- SySteM:PoWeR:SoUr?
- syst:POW:sour?

1.2.3 プログラムデータ

プログラムデータは、コマンドメッセージユニットで指定されたパラメータとしてプログラムヘッダーの後に送信されます。この操作マニュアルでは、プログラムデータの形式を示すため、表 1.2にある表記を使用しています。ほとんどはIEEE488.2標準で定義されています。

プログラムデータのタイプ	説明
<BOOLEAN PROGRAM DATA>	IEEE488.2で定義されています。 オン/オフ、有効/無効、はい/いいえを指定します。 オン/有効状態を指定するには、{ON 1}と設定します。 オフ/無効状態を指定するには、{OFF 0}と設定します。
<NUMERIC PROGRAM DATA>	IEEE488.2で定義されているように、<DECIMAL NUMERIC PROGRAM DATA>と<NON-DECIMAL NUMERIC PROGRAM DATA>で構成されます。 ネットワークマスタは、<NUMERIC PROGRAM DATA>の10進数エン트리と非10進数エントリの両方を受け入れます。
<DECIMAL NUMERIC PROGRAM DATA>	IEEE488.2で定義されています。 <NR1>,<NR2>と<NR3>の10進数値で構成されます。 <NR1> は、整数値です。 <NR2> は、固定小数点形式の数値です。

次のページへ続く...

... 前のページから続く

プログラムデータのタイプ	説明
	<NR3> は、浮動小数点形式の数値です。 例: <pre><NR1>: 123 <NR2>: -123.456 <NR3>: 1.23E-3</pre>
<NONDECIMAL NUMERIC PROGRAM DATA>	IEEE488.2で定義されています。 <HEXADECIMAL>, <OCTAL>, または<BINARY>のプログラムデータで構成されます。 詳細については以下を参照してください。
<HEXADECIMAL>	以下のIEEE488.2で定義された16進数形式に従います。 <code>#{H h}{A a B b C c D d E e F f <digit>}...</code> <digit> は、値の範囲が0x30~0x39 (10進数では48~57)のASCII文字です。つまり、0~9の数値です。 例: <pre>#h1234ABCD #Hfe1a9</pre>
<OCTAL>	次のように、IEEE488.2で定義された8進数形式に従います。 <code>#{Q q}{0 1 2 3 4 5 6 7}...</code> 例: <pre>#q12345670 #Q77</pre>
<BINARY>	以下のIEEE488.2で定義されたバイナリ形式に従います。 <code>#{B b}{0 1}...</code> 例: <pre>#b10101010 #B110</pre>
<STRING PROGRAM DATA>	IEEE488.2で定義されています。 1組のシングルコーテーション (') またはダブルコーテーション (") で囲まれた文字列。 例: <pre>"Network Master" 'Testing the network'</pre>
<CHARACTER PROGRAM DATA>	IEEE488.2で定義されています。 選択範囲の複数のニーモニックを指定します。プログラムヘッダーのニーモニックと同様、ニーモニック<CHARACTER PROGRAM DATA>には短い形式と長い形式があります。ネットワークマスタは、短い形式と長い形式の両方を受け入れます。詳細については、このマニュアルの後の章にあるネットワークマスタ固有のコマンドに関する詳しい説明を参照してください。

表 1.2: 許容されるプログラムデータ

1.2.4 プログラムメッセージターミネータ

プログラムメッセージターミネータは、プログラムメッセージの末尾であることを示しています。ネットワークマスタは、ターミネータを受信するとプログラムメッセージが完了したと見なし、メッセージの処理を開始します。ターミネータは、必ずプログラムメッセージの末尾に付加する必要があります。ネットワークマスタの場合、プログラムメッセージターミネータは次のとおりです。

[<WHITE SPACE>]{NL} (Ethernetベースのリモート制御の場合)

<WHITE SPACE> は、値の範囲が0x00～0x09または0x0B～0x20（10進数の0～9または11～32）の1つ以上のASCII文字です。これらの範囲には、ASCII制御文字とスペースが含まれます。ただし、NL（改行）を除きます。<WHITE SPACE>には、CR（0x0D）（10進数の13）が含まれているため、{CR}{NL}もEthernetベースのリモート制御のネットワークマスタによりターミネータとして解釈されます。これは、従来のモデルとの互換性を保つためです。

1.2.5 複合プログラムメッセージ

複合ヘッダーは、ネットワークマスタによりサポートされます。複合ヘッダーの使用例を次に示します。次のような3つのプログラムメッセージユニットがあるとします。

```
SYSTem:TIME?  
SYSTem:DATE?  
SYSTem:POWer:SOURce?
```

これらは、次のように1つのプログラムメッセージとして結合できます。

```
SYSTem:TIME?; :SYSTem:DATE?; :SYSTem:POWer:SOURce?
```

あるいは、単に次のようにすることもできます。

```
SYSTem:TIME?; DATE?; POWer:SOURce?
```

（2番目と3番目のプログラムデータユニットでは、ニーモニックSYST:を省略できます）

複合ヘッダーの詳細については、IEEE488.2標準の付録Aを参照してください。

1.2.6 シーケンシャル実行

ネットワークマスタは、一度に1つのプログラムメッセージユニットを処理し、プログラムメッセージ内の配置と同じ順序で処理します。現在のプログラムメッセージの処理が完了するまで、新しいプログラムメッセージの処理は開始しません。

1.3 レスポンスメッセージ

レスポンスメッセージは、クエリへのレスポンスとしてネットワークマスタからコントローラに送信されるメッセージです。図 1.6を参照してください。

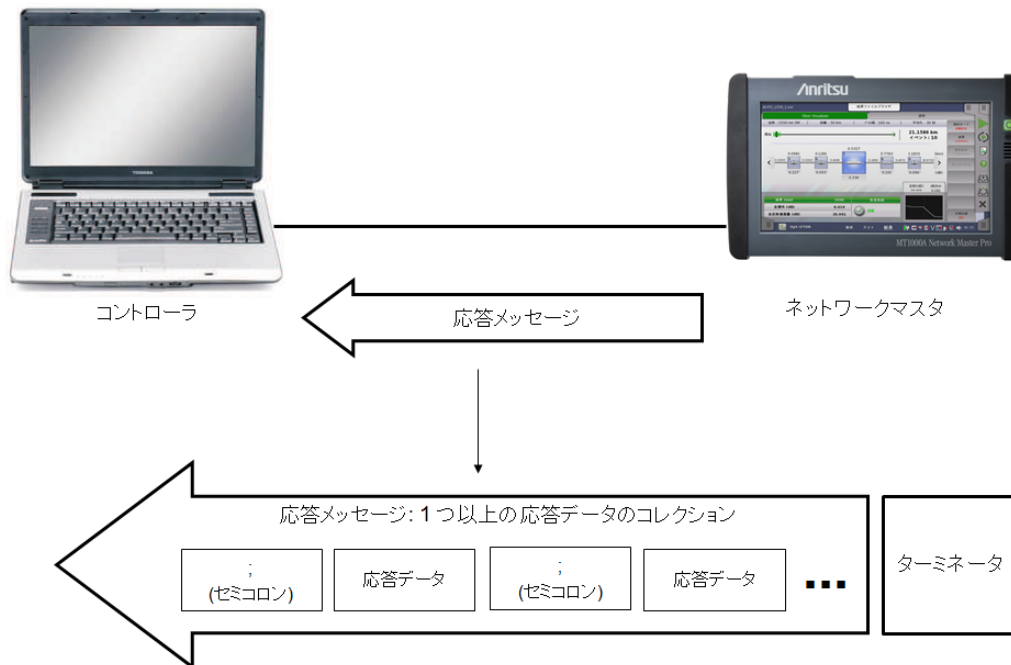


図 1.6: レスポンスメッセージ構造

レスポンスメッセージは、1つ以上のレスポンスデータで構成され、それぞれセミコロン (;) で区切られます。レスポンスメッセージは、レスポンスメッセージターミネータで終端処理されます。

1.3.1 レスポンスデータ

レスポンスデータは、コントローラから受信したクエリに対するレスポンスとしてネットワークマスタにより返されるデータです。表 1.3に、このマニュアルで使用されるレスポンスデータの例を示します。

レスポンスデータのタイプ	説明
<BOOLEAN RESPONSE DATA>	SCPI-99で定義されています。 オン/オフ、有効/無効、はい/いいえを指定します。 「1」が返された場合、オン/有効状態を示しています。 「0」が返された場合、オフ/無効状態を示しています。
<NR1 NUMERIC RESPONSE DATA>	IEEE488.2で定義されています。 10進数の整数値を指定します。 例: 123 -500
<NR2 NUMERIC RESPONSE DATA>	IEEE488.2で定義されています。 固定小数点形式の数値を指定します。 例: 123.45 -500.0

次のページへ続く...

... 前のページから続く

レスポンスデータのタイプ	説明
<NR3 NUMERIC RESPONSE DATA>	IEEE488.2で定義されています。 浮動小数点形式の数値を指定します。 例: 1.23E3 -5.67E-4
<HEXADECIMAL NUMERIC RESPONSE DATA>	以下のIEEE488.2で定義された16進数形式に従います。 #H{A B C D E F <digit>}... <digit> は、値の範囲が0x30~0x39（10進数の48~57）のASCII文字（0~9の数値）です。 例: #H0011EEFF
<BINARY NUMERIC RESPONSE DATA>	以下のIEEE488.2で定義されたバイナリ形式に従います。 #B{0 1}... 例: #B10101010
<STRING RESPONSE DATA>	IEEE488.2で定義されています。 1組のダブルコーテーション（"）で囲まれた文字列。 例: "Network Master - Testing the network."
<CHARACTER RESPONSE DATA>	IEEE488.2で定義されています。 選択範囲の複数のニーモニックを指定します。プログラムヘッダーのニーモニックと同様、ニーモニック<CHARACTER RESPONSE DATA>には短い形式と長い形式があります。ネットワークマスタは、常に短い形式を返します。詳細については、このマニュアルで後述する、ネットワークマスタ固有のコマンドに関する詳しい説明を参照してください。
<EXPRESSION RESPONSE DATA>	IEEE488.2で定義されています。 ネットワークマスタにより定義された一連の<RESPONSE DATA>要素。それぞれコンマ（,）で区切られ、括弧のセットで囲まれます。 例: (2,0.5),(3,0.25),(4,1.75) 詳細については、ネットワークマスタ固有のコマンドに関する詳しい説明を参照してください。
<DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>	IEEE488.2で定義されています。 このレスポンスデータ型は、機器がバイナリデータ（通常はPDFファイル）をコントローラにストリーミングする際に使用されます。次のように定義されます。 #<nonzero digit><digits><8 bit data bytes> , <nonzero digit> は、範囲が1~9の単一のASCII文字です。これは、<digits>の長さをバイト数で表しています。 <digits> は、範囲が0~9の複数のASCII文字です。これらの文字を合わせると、後続のデータバイト数の10進数表現になります。 例: #49137<9137 bytes of binary data>

表 1.3: ネットワークマスタのレスポンスデータ

1.3.2 レスponseメッセージターミネータ

レスponseメッセージターミネータは、レスponseメッセージの末尾であることを示しています。ネットワークマスタは、レスponseメッセージの末尾にターミネータを付加して、メッセージの末尾であることを示します。ネットワークマスタの場合、レスponseメッセージターミネータは{NL}です。

1.3.3 プロンプト

Ethernetベースのリモート制御の場合、プログラムメッセージのすべてのコマンドが完了したときに、ネットワークマスタでプロンプトを返すこともできます。プロンプトは、レスponseメッセージ（ある場合）の後に挿入されます。リモート制御インタフェースのコマンドラインに手動でコマンドを入力する場合は、プロンプトを有効にすると役立ちます。挿入されるプロンプトは次のとおりです。

SCPI:>

1.4 ステータス

1.4.1 IEEE488.2標準のステータスとSCPIにより定義されたレジスタ

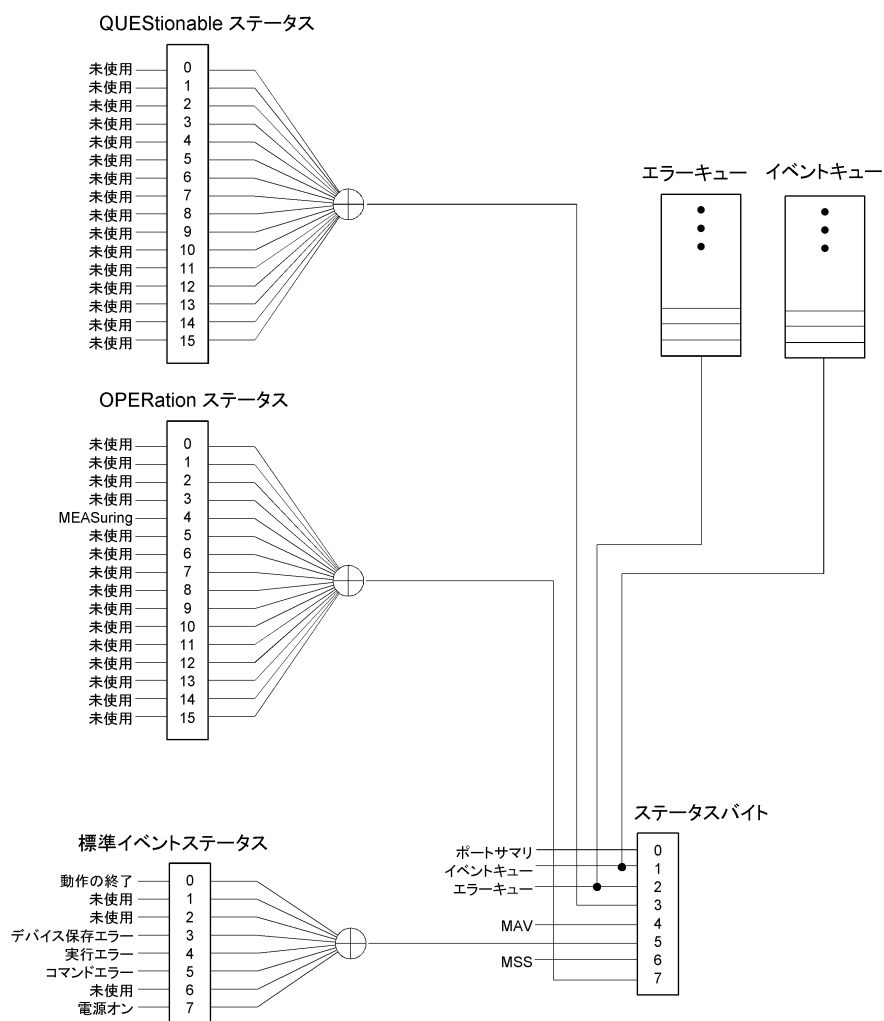


図 1.7: IEEE488.2標準のステータスとSCPIで定義されたレジスタ/キュー
($\oplus$: 論理OR)

ステータスバイト

Bit	名称	説明
0	ポート1	ポートイベントサマリレジスタのサマリメッセージビット。 2.5.16 節で説明されているSTATus:PRESetコマンドと、 2.5.13 節で説明されているSTATus:PORT:ENABLeコマンドを使用して、このサマリメッセージの生成を有効にします。
1	Event Queue	現在選択されているアプリケーションサーバーのイベントキューのサマリメッセージビット。 2.3.2 節で説明されているSYSTem:ERRor[:NEXT]?コマンドを使用して、メッセージを取得します。
2	Error Queue	接続されているすべてのアプリケーションサーバのためのエラーキューのサマリメッセージビット 2.4.18 節で説明されているINSTrument:ERRor[:NEXT]?コマンドを使用して、メッセージを取得します。
3	QUEStionable	クエスチョナブルステータスレジスタのサマリメッセージビット。 2.5.8 節で説明されているSTATus:QUEStionable:ENABLeコマンドを使用して、このサマリメッセージの生成を有効にします。
4	出力キュー	出力キューのサマリメッセージビット。
5	標準イベント	標準イベントステータスレジスタのサマリメッセージビット。 2.2.2 節で説明されている*ESEコマンドを使用して、このサマリメッセージの生成を有効にします。
6	マスターサマリ	マスターサマリステータスメッセージ。 2.2.7 節で説明されている*SREコマンドを使用して、このサマリメッセージの生成を有効にします。
7	OPERation	オペレーションステータスレジスタ。 2.5.2 節で説明されているSTATus:OPERation:ENABLeコマンドを使用して、このサマリメッセージの生成を有効にします。

表 1.4: ステータス・バイト・レジスタ内のビット（使用されていないビットはリストされません）

ステータス・バイト・レジスタの詳細については、37 ページの2.2.8 節を参照してください。

標準イベントステータス

すべての条件ビットは、設定された後すぐに0に戻ります。これは、ビットを確認するには、イベントレジスタを読み取る必要があることを意味します。デバイス依存エラー、実行エラー、およびコマンドエラーが何によってトリガされるかの詳細については、20 ページの「エラー/イベントキュー」セクションを参照してください。

標準イベントステータスの詳細については、34 ページの2.2.2 節を参照してください。

Bit	名称	説明
0	オペレーション完了	*OPCコマンドを受信すると、条件ビットが1に変更されます。
3	デバイス依存エラー	必要なSWまたはHWオプションがないか、エラー/イベントキューがいっぱいになると、条件ビットが1に変更されます。
4	実行エラー	コマンドが正しく実行されないと、条件ビットが1に変更されます。
5	コマンドエラー	不明なコマンドかエラーコマンドを受信すると、条件ビットが1に変更されます。
7	電源オン	外部電源が接続されると、条件ビットが1に変更されます。

表 1.5: 標準イベントステータスレジスタ内のビット
(使用されていないビットはリストされません)

エラー/イベントキュー

予期しないエラーやイベントが発生した場合、エラー/イベントキューにエントリが追加されます。このキューには、500のエラーまたはイベントを保持できます。キューがオーバーフローした場合、最も新しいイベントが破棄されます。キューが空ではない場合、ステータスバイトのビット2にあるサマリメッセージは1です。表 1.6に、キューに挿入される各種エラーおよびイベントの概要を示します。

イベント番号	エラーの説明
0	エラーなし（キューが空の場合）
コマンドエラー（コマンドエラービットが同時に設定されます）	
-100	コマンドエラー
-102	構文エラー
-104	データ型エラー
-115	予期しないパラメータ数
-130	接尾辞エラー
-131	無効な接尾辞
-138	接尾辞不許可
実行エラー（実行エラービットが同時に設定されます）	
-200	実行エラー
-220	パラメータエラー
-221	設定が競合
-222	データが範囲外
-224	無効なパラメータ値
-250	マスマスストレージエラー
デバイス依存エラー（デバイス依存エラービットが同時に設定されます）	
1	オプションなし
-350	キューのオーバーフロー

表 1.6: エラー/イベントキューで発生する可能性があるエラーとイベント

イベントキューの詳細については、52 ページの2.4.18 節を参照してください。
エラーキューの詳細については、38 ページの2.3.2 節を参照してください。

クエスチョナブルステータス

Bit	名称	説明
このレジスタ内のビットは本バージョンでは使用されていません。		

表 1.7: クエスチョナブルステータスレジスタ内のビット（使用されていないビットはリストされません）

クエスチョナブルステータスレジスタの詳細については、[57](#) ページの[2.5.6](#) 節を参照してください。

オペレーションステータス

Bit	名称	説明
4	Measuring	アプリケーションサーバが測定またはテストを実行している場合は、測定状態のビットは1に変更されます。測定またはテストが停止した場合は、測定状態のビットは0に変更されます。

表 1.8: オペレーションステータスレジスタ内のビット（使用されていないビットはリストされません）

オペレーションステータスレジスタの詳細については、[55](#) ページの[2.5.1](#) 節を参照してください。

1.4.2 ネットワークマスタ固有のステータスレジスタ

図 1.8に、ネットワークマスタ固有のステータスレジスタの構造を示します。

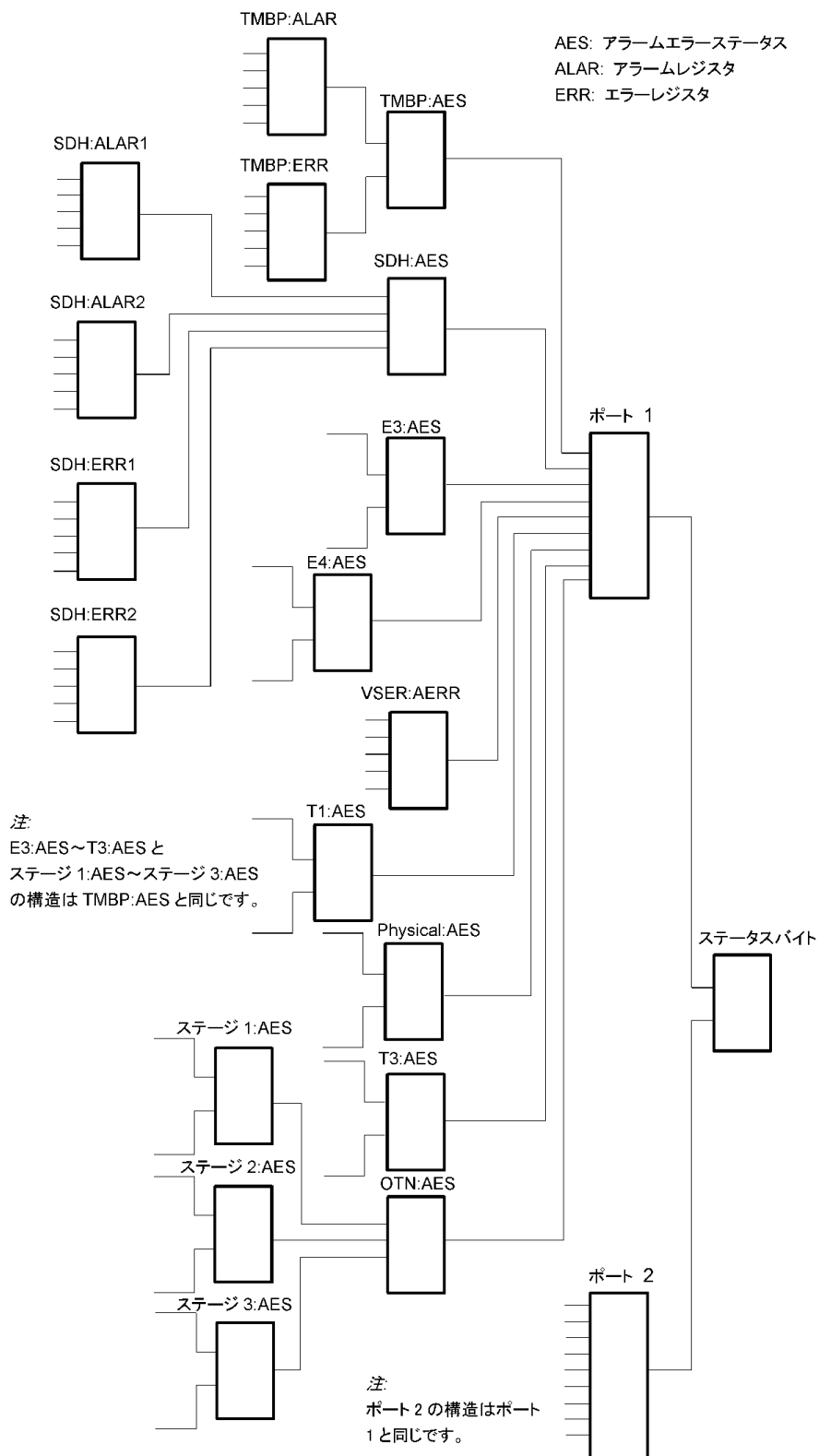


図 1.8: サポートされる一部のインタフェースのためのネットワークマスタ固有のステータスレジスタ

T1, OTN, 物理, およびT3の各インタフェースに同様のレジスタが存在します。

各インタフェースには、現在のアラームおよびエラーステータスを表す1つ以上のレジスタがあります。これらのアラームおよびエラーレジスタは、それぞれ全般的なインタフェースサマリレジスタ（AESummary）に要約されます。図 1.9を参照してください。各レジスタの正確なレイアウトは、各インタフェースの [Status] セクションにあります。ポートステータスレジスタは2つあります（ネットワークマスタの各ポ

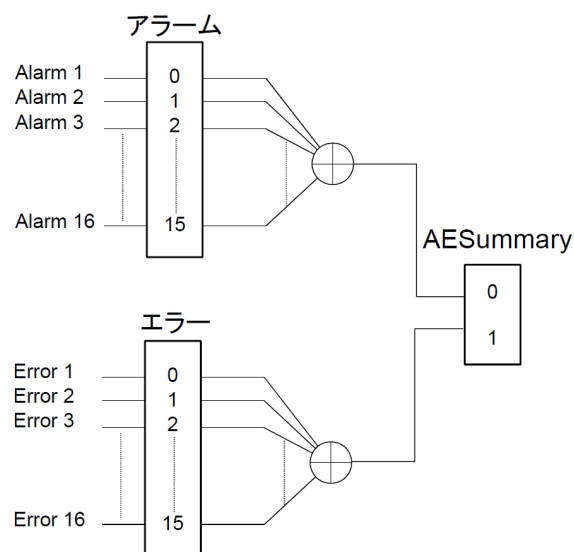


図 1.9: インタフェースのアラームおよびエラーステータスレジスタの全般的な構造

トに1つずつ)。ポートステータスレジスタには、アクティブなインタフェースのAESummaryレジスタに集約されます。ポートステータスレジスタは、ステータスバイトのビット0と1にも集約されます。図 1.10を参照してください。

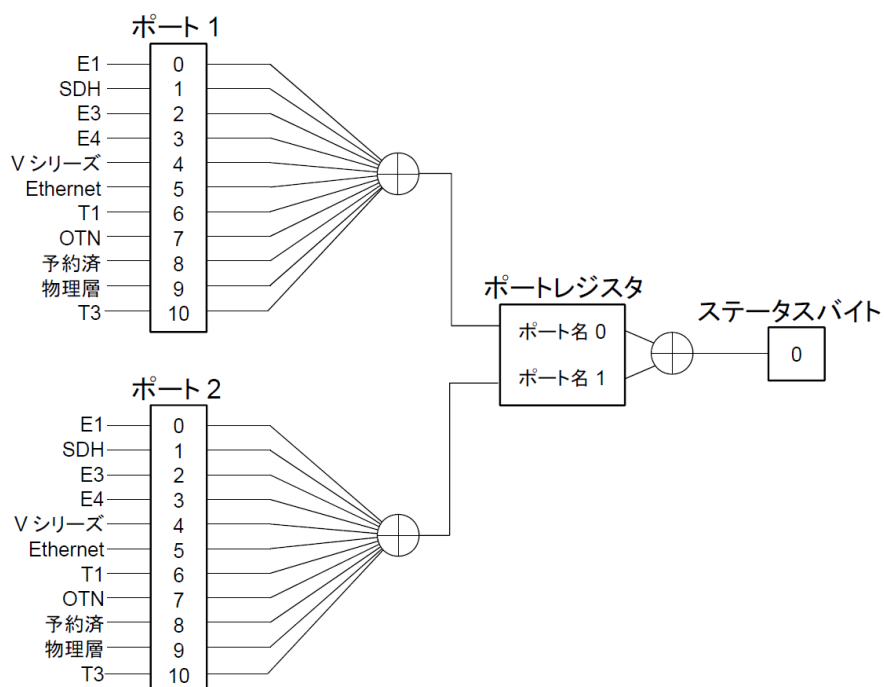


図 1.10: ポートステータスレジスタの構造

ネットワークマスタ固有のステータスレジスタはすべて、IEEE488.2のセクション11.4.2で定義されているレジスタモデルに従います。レジスタモデルについては、図 1.11に示します。

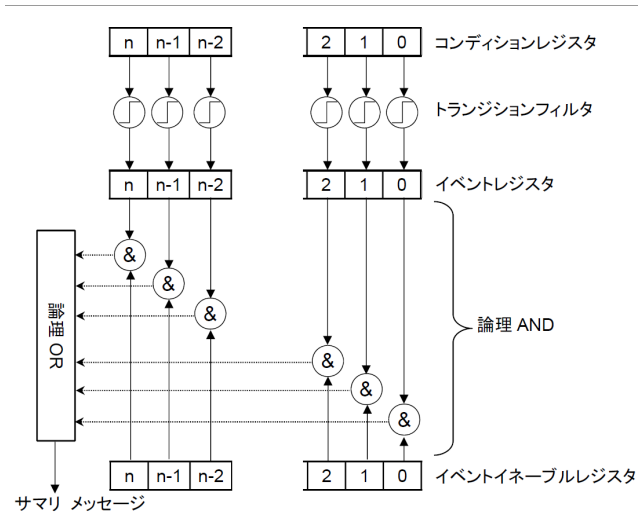


図 1.11: ネットワークマスタ固有のステータスレジスタのレジスタモデル

コンディションレジスタ

コンディションレジスタには、機器のリアルタイムステータスや、他のステータスレジスタのサマリメッセージビットが反映されます。

トランジションフィルタ

ネットワークマスタ固有のステータスレジスタのトランジションフィルタは、「ポジティブトランジション」条件にロックされます。これは、条件ビットが0から1に変化するとイベントが生成されることを意味します。トランジション条件を変更することはできません。

イベントレジスタ

イベントレジスタには、トランジションフィルタの出力が格納されます。これらのレジスタは、読み取り時にクリアされます。

イベントイネーブルレジスタ

イベントイネーブルレジスタは、読み取りまたは書き込みすることはできません。すべて0に設定されているとき、電源がオンになります。イベントイネーブルレジスタを有効にし、サマリメッセージの生成を有効にするには、STATus:PRESet コマンドを送信する必要があります。STATus:PRESet コマンドは、イベントイネーブルレジスタ内のすべてのビットを1に設定します。

サマリレジスタ

サマリレジスタ(AESummary)はアラーム、エラーイベントレジスタおよび、下位のサマリレジスタのイベントレジスタから要約されたステータスが格納されます。(図 1.9参照)。他のステータスレジスタと同様に、IEEE488.2のセクション11.4.2で定義されているレジスタモデルに従い、コンディションレジスタとイベントレジスタから構成されます。また、サマリレジスタは下位のイベントレジスタから構成されるため、ポジティブトランジションの条件にロックされます。

1.4.3 ステータスレジスタの読み取り，書き込み，クリア

次の2つの表に，各種ステータスレジスタとキューの読み取りと書き込みを行うことができるかどうかを示します。さらに，レジスタがクリアまたは有効化されるタイミングと方法も示します。

レジスタ		読み取り	書き込み
IEEE488.2標準ステータスレジスタ	ステータスバイト	*STB?	不可
	サービリクエストアイネーブル	*SRE?	*SRE
	標準イベントステータス	*ESR? 読み取り後，レジスタの中身はクリアされます。	不可
	標準イベントステータスイネーブル	*ESE?	*ESE
SCPIにより定義されたステータスレジスタ	エラー/イベントキュー	SYST:ERR? 読み取り後，エラー/イベントはキューから削除されます。	不可
	オペレーションイベント	STAT:OPER? 読み取り後，レジスタの中身はクリアされます。	不可
	オペレーションイネーブル	STAT:OPER:ENAB?	STAT:OPER:ENAB
	クエスチョナブルイベント	STAT:QUES? 読み取り後，レジスタの中身はクリアされます。	不可
	クエスチョナブルイネーブル	STAT:QUES:ENAB?	STAT:QUES:ENAB
ネットワークマスタ固有のステータスレジスタ	コンディション	<Interface>:STAT:<Port>:<Register>:COND?	不可
	トランジションフィルタ	不可	不可
	イベント	<Interface>:STAT:<Port>:<Register>? 読み取り後，レジスタの中身はクリアされます。	不可
	イネーブル	不可	不可

表 1.9: ステータスレジスタの読み取りと書き込み

レジスタ		*RST	*CLS	PowerOn	STAT:PRES
IEEE488.2 標準ステータスレジスタ	ステータスバイト	変更なし	クリア	クリア	変更なし
	サービスリクエストイネーブル	変更なし	変更なし	クリア	変更なし
	標準イベントステータス	変更なし	クリア	クリア	変更なし
	標準イベントステータスイネーブル	変更なし	変更なし	クリア	変更なし
SCPIにより定義されたステータスレジスタ	エラー/イベントキュー	変更なし	クリア	クリア	変更なし
	オペレーションイベント	変更なし	クリア	クリア	変更なし
	オペレーションイネーブル	変更なし	変更なし	クリア	変更なし
	クエスチョナブルイベント	変更なし	クリア	クリア	変更なし
	クエスチョナブルイネーブル	変更なし	変更なし	クリア	変更なし
ネットワークマスタ固有のステータスレジスタ	コンディション	変更なし	変更なし	クリア	変更なし
	トランジションフィルタ	変更なし	変更なし	変更なし	変更なし
	イベント	変更なし	クリア	クリア	変更なし
	イネーブル	変更なし	変更なし	クリア	有効 (すべて1)

表 1.10: 各種コマンド/イベントのステータスレジスタの動作

注意事項

サマリレジスタのコンディションレジスタは、「ポジティブトランジション」条件にロックされます。そのため、アラームやエラーが発生した状態で、レジスタをクリア(*CLS)すると、アラームやエラーが発生していてもレジスタは0にクリアされた状態になります。

1.5 コントローラの例

このセクションでは、コントローラをネットワークマスタ機器に接続する方法の1つの例について説明します。

1.5.1 PuTTY

PuTTYは、Raw TCP接続をサポートする無料のTelnet/SSHクライアントです。PuTTYを使用すると、機器への端末エミュレーションアクセスが可能になります。PuTTYを使用するときは、プロンプトを有効にすることをお勧めします。PuTTYでは、MEAS:EXPコマンドなどのファイルストリーミングはサポートされません。

PuTTYは、<http://www.chiark.greenend.org.uk/~sgtatham/putty/>からダウンロードできます。

セットアップ

1. PuTTYをインストールします。
2. PuTTYを起動します。
3. PuTTY Configurationの**Category:**→**Terminal**で、**[Implicit CR in every LF]** をオンにします。

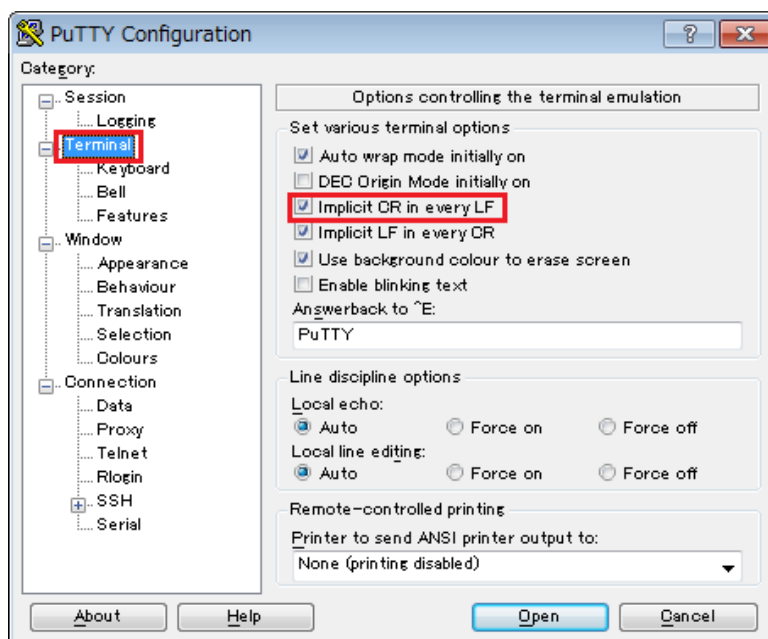


図 1.12: PuTTYで **[Implicit CR in every LF]** をオンにする

4. 機器のGUIで、機器のIP Address情報を見つけます。図 1.13を参照してください。次に、その情報をPuTTYに入力します。図 1.14を参照してください。

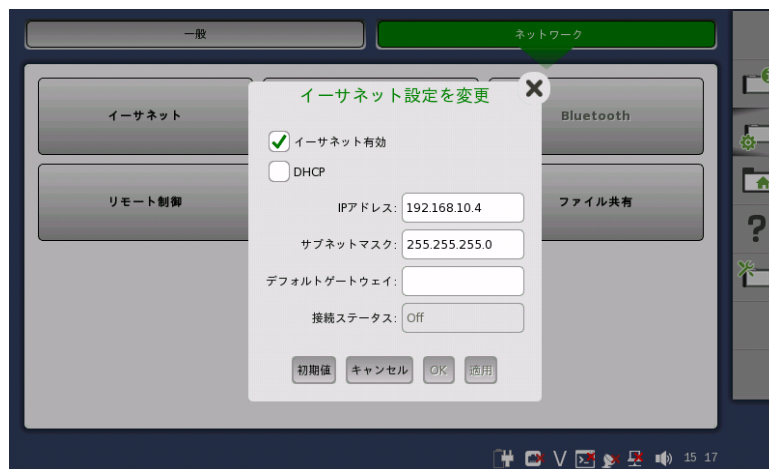


図 1.13: 機器のIP Address

5. PuTTYのPortフィールドに56001を入力し、Connection type にて、Raw を選択します。図 1.14で、[Open] ボタンをクリックします。

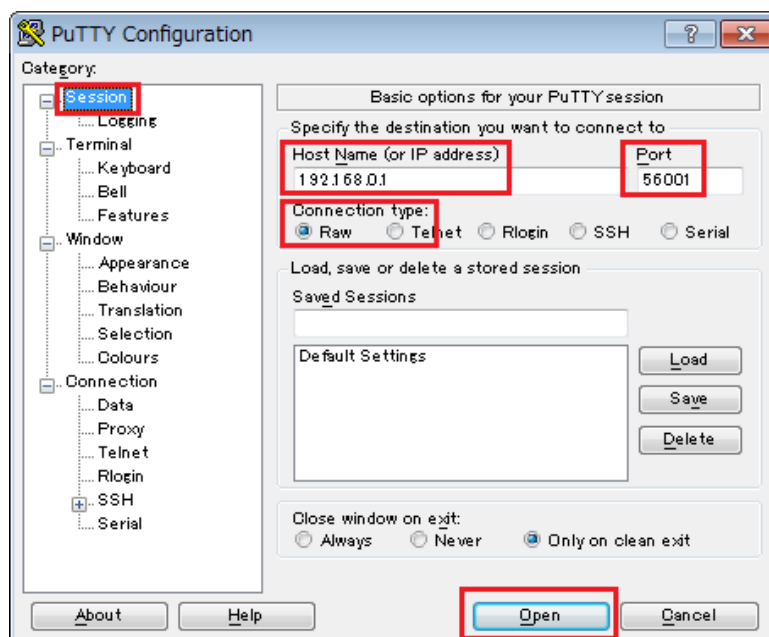


図 1.14: PuTTYでの宛先の指定と接続の開始 (Open)

6. ウィンドウが表示されます。図 1.15を参照してください。

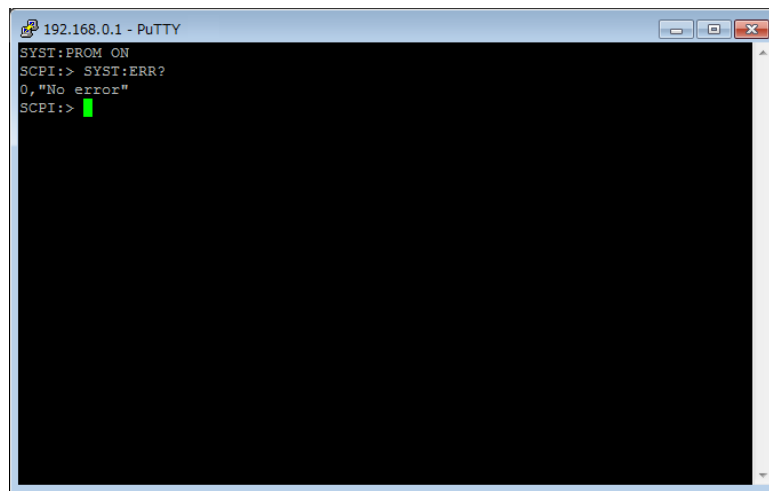


図 1.15: PuTTYとの接続の確立

1.6 定義

1.6.1 NaN (Not a Number)

NaNは、SCPI-99で定義されています。NaNは、IEEE 754での定義に従って9.91E37 (<NR3 NUMERIC RESPONSE DATA>) で表現されます。データがないことを表現するときもNaNが使用されます。

1.6.2 → Right Arrow

このドキュメントで使用される右矢印→には、次の2つの意味があります。

- 矢印の左側にはクエリがあり、右側には戻り値があります。
例: TMBP:RX1:PATT? → PRBS11

1.6.3 データビット (DB)

データビットはDB x と表現されます。ここで、 x はレジスタ内のビットインデックスを表しています。DB1は常にLSBです。

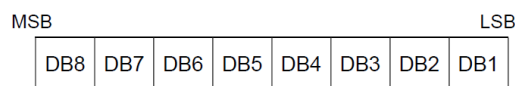


図 1.16: データビット

1.6.4 ポート番号 (論理ポート)

SCPIコマンドで指定するポート番号<Pt>には、起動したアプリケーションごとに定義された論理ポートの番号を使用します。論理ポートの番号は、各アプリケーションが使用する物理ポートのモジュール、ポートの小さい順に1から順に番号が割り当てられます。OTDRモジュールでは物理ポートはPORT1を指定します。

OTDRアプリケーション起動コマンドの例

- INST:STAR OTDR-OTDR,1-PORT1 (図 1.17)
物理ポート : 論理ポート
1-PORT1 : PORT1



図 1.17: OTDRモジュールのポート番号

注意事項

- MU100021Aでシングルモード、マルチモードを切り替えるときは65 ページの3.1.1 節を参照してください。

- MU100023Aにおいても物理ポートはPORT1を指定します。66 ページの3.1.4 節のコマンドによって波長を1310 nmあるいは1550 nmに設定するとPORT1, 1650 nmに設定するとPORT2が測定ポートとして選択されます。

Chapter 2

SCPI 適合情報

2.1 SCPIのバージョン

MT1000/MT1100リモート制御アプリケーションは，SCPI 1999.0に準拠しています。

2.2 IEEE 488.2 必須コマンド

2.2.1 *CLS

文法	*CLS
説明	本コマンドはステータス・バイト・レジスタにサマリ表示されるすべてのイベントレジスタをクリアします。エラーキューは空になります。標準イベントステータスイネーブルレジスタとサービスリクエストイネーブルレジスタは本コマンドによって変化しません。
パラメータ	無し
レスポンス	無し
例	*CLS
注	すべての実行中の (SCPI) セッションはそれぞれ標準レジスタを持っています。

2.2.2 *ESE

文法	*ESE <mask>
説明	本コマンドは標準イベント・ステータス・イネーブルレジスタのビットを設定します。イネーブルレジスタのビットを”1”に設定すると、標準イベント・ステータス・レジスタの対応するビットが有効になります。本レジスタは、電源投入時にクリアされます。 *RSTコマンドと*CLSコマンドは、標準イベント・ステータス・イネーブルレジスタには影響しません。
パラメータ	<mask> = <NUMERIC PROGRAM DATA> イネーブルマスクのビットと値: DB1 (1) = 操作完了 DB2 = 未使用 DB3 = 未使用 DB4 (8) = 機器依存エラー DB5 (16) = 実行エラー DB6 (32) = コマンドエラー DB7 = 未使用 DB8 (128) = Power On <i>MINimum=0, MAXimum=255</i>
レスポンス	無し
例	*ESE 16
注	すべての実行中のセッションはそれぞれ標準イベント・ステータス・イネーブル・レジスタを持っています。

文法	*ESE?
説明	標準イベント・ステータス・イネーブル・レジスタの内容を問い合わせます。
パラメータ	無し
レスポンス	<mask> = <NR1 NUMERIC RESPONSE DATA> イネーブルマスクのビットの値は、*ESEコマンドを参照してください。
例	*ESE? → 16
注	

2.2.3 *ESR?

文法	*ESR?
説明	標準イベント・ステータス・レジスタの内容を問い合わせます。標準イベント・ステータス・レジスタを読み取ると、値がクリアされます。
パラメータ	無し
レスポンス	<value> = <NR1 NUMERIC RESPONSE DATA> レジスタのビットと値は次のとおりです。 DB1 (1) = 操作完了 DB2 = 未使用 DB3 = 未使用 DB4 (8) = 機器依存エラー DB5 (16) = 実行エラー DB6 (32) = コマンドエラー DB7 = 未使用 DB8 (128) = 電源オン
例	*ESR? → 49
注	すべての実行中のセッションはそれぞれ標準イベント・ステータス・レジスタを持っています。

2.2.4 *IDN?

文法	*IDN?
説明	インタフェース経由で機器の識別情報を問い合わせます。
パラメータ	無し
レスポンス	<manufacturer>,<model>,<serial>,<version> = <ARBITRARY ASCII RESPONSE DATA>
例	*IDN? → Anritsu,MT1000,6123456789,1.00
注	

2.2.5 *OPC

文法	*OPC
説明	実行中のすべてのコマンド動作が終了したときに、標準イベント・ステータス・レジスタの動作の完了ビットを1に変化させるように設定します。
パラメータ	無し
レスポンス	無し
例	*OPC
注	すべての実行中のセッションはそれぞれ標準イベント・ステータス・レジスタを持っています。

文法	*OPC?
説明	実行中のすべてのコマンド動作が終了すると、機器の出力キューにASCII文字'1'が設定されます。
パラメータ	無し
レスポンス	<operation complete> = <NR1 NUMERIC RESPONSE DATA>
例	*OPC? → 1
注	現在のセッションによって接続されているアプリケーションサーバのみを同期します。

2.2.6 *RST

文法	*RST
説明	機器の設定を、機器内部に保存されているリセット状態（標準状態）にします。機器はコマンド待ちの状態になります。 起動中のアプリケーションはクローズされます。 以下は本コマンドによって変化しません。 -サービス・リクエスト・イネーブルレジスタ（SRE） -標準イベント・ステータス・レジスタ（ESR） -標準イベント・ステータス・イネーブルレジスタ（ESE） -機器に依存するすべてのステータス・イベント・レジスタ，またはステータス・イベント・イネーブルレジスタ
パラメータ	無し
レスポンス	無し
例	*RST
注	*RSTコマンドの動作はSCPI規格で規定されているコマンド動作に沿うものです。

2.2.7 *SRE

文法	*SRE <enable mask>
説明	サービスリクエスト・イネーブル・レジスタのビットを設定します。イネーブルレジスタのビットを” 1” に設定すると，対応するステータスバイトのビットが有効になります。また，ステータスバイトのマスタサマリステータスビットに対応するビット（DB7）が1に設定されます。本レジスタは，電源投入時にクリアされます。*RSTコマンドと*CLSコマンドは，サービスリクエストイネーブルレジスタには影響しません。
パラメータ	<enable mask> = <NUMERIC PROGRAM DATA> レジスタのビットと値 DB1 (1) = ポートイベントサマリ DB2 (2) = 現在選択中のアプリケーションサーバ用のエラー/イベントキューサマリ DB3 (4) = 接続されているすべてのアプリケーションサーバ用のエラー/イベントキューサマリ DB4 (8) = クエスチョナブルステータスサマリ DB5 (16) = メッセージあり（MAV） DB6 (32) = 標準イベントステータスサマリ（ESB） DB7 = 未使用 DB8 (128) = 実行ステータスサマリ <i>MINimum=0, MAXimum=255</i>
レスポンス	無し
例	*SRE 255
注	すべての実行中のセッションはそれぞれサービスリクエストイネーブルレジスタを持っています。

文法	*SRE?
説明	サービスリクエスト・イネーブル・レジスタの内容を問い合わせます。
パラメータ	無し
レスポンス	<mask> = <NR1 NUMERIC RESPONSE DATA> イネーブルマスクのビットの値は，*SREコマンドを参照してください。
例	*SRE? → 255
注	

2.2.8 *STB?

文法	*STB?
説明	ステータス・バイト・レジスタの内容を問い合わせます。STBレジスタのビットのどれかが設定され、サービスリクエスト・イネーブル・レジスタにあるビットと一致するように設定された場合、マスタサマリステータス（MSS）ビットが1になります。 *SRE を参照してください。MSSを含むステータス・バイト・レジスタのビットは、本コマンドにより変化しません。
パラメータ	無し
レスポンス	<value> = <NR1 NUMERIC RESPONSE DATA> レジスタのビットと値 DB1 (1) =ポートイベントサマリ DB2 (2) =アプリケーションサーバサマリ用のエラー/イベントキュー DB3 (4) =エラー/イベントキューサマリ DB4 (8) =クエスチョナブルステータスサマリ DB5 (16) =メッセージあり（MAV） DB6 (32) =標準イベントステータスサマリ（ESB） DB7 (64) =マスタサマリステータス（MSS） DB8 (128) =実行ステータスサマリ
例	*STB? → 7
注	

2.2.9 *TST?

文法	*TST?
説明	自己診断結果に異常があるかどうかをレスポンスします。
パラメータ	無し
レスポンス	<result> = <NR1 NUMERIC RESPONSE DATA> 0:自己診断エラー無し 1:自己診断エラー有り
例	*TST? → 0
注	自己診断は電源投入時に自動的に実行されます。

2.2.10 *WAI

文法	*WAI
説明	現在実行しているコマンド処理が終了するまで、他のコマンドの実行を保留します。 *WAI コマンドによって待っている間に、実行中のすべてコマンドは処理を終了します。
パラメータ	無し
レスポンス	無し
例	*WAI
注	*WAI は「オーバーラップコマンド」に対して有効ですが、現在のネットワークマスターにはオーバーラップコマンドはないため、有効に働くケースはありません。

2.3 SCPI System サブシステムコマンド

2.3.1 SYSTem:VERSion?

文法	SYSTem:VERSion?
説明	システムが従っているSCPIのバージョンを問い合わせます。
パラメータ	無し
レスポンス	<version> = <NR2 NUMERIC RESPONSE DATA>
例	SYST:VERS? → 1999.0
注	

2.3.2 SYSTem:ERRor[:NEXT]?

文法	SYSTem:ERRor[:NEXT]?
説明	エラーキューの一番古い情報を取り出します。取り出した情報はキューから削除されます。
パラメータ	無し
レスポンス	<error number> = <NR1 NUMERIC RESPONSE DATA> <description> = <STRING RESPONSE DATA>
例	SYST:ERR? → -222,"Data out of range"
注	すべての実行中のセッションはそれぞれエラーキューを持っています。 追加メッセージが、選択されたTESTまたはBOTHである場合は、アプリケーションサーバIDが各エラーメッセージに追加されます。 システムエラー用のアプリケーションサーバIDは-1になります。 エラーキューにエラーメッセージが含まれない場合、アプリケーションサーバIDは0に固定されます。 追加メッセージが、選択されたCOMMAndまたはBOTHである場合は、エラーコマンドが各エラーメッセージに追加されます。

2.3.3 SYSTem:ERRor:ADDITIONal[:MESSAge]

文法	SYSTem:ERRor:ADDITIONal[:MESSAge] <message>
説明	エラーメッセージに含まれる追加メッセージを選択します。
パラメータ	<message> = <CHARACTER PROGRAM DATA> NONE TEST COMMAnd BOTH
レスポンス	無し
例	SYST:ERR:ADD BOTH SYST:ERR? → -115,"Unexpected number of parameters:-1:INST:TERM"
注	本設定は現在のセッションのみに適用され、セッションが終了すると、初期値であるNONEに戻ります。 2.3.2 節 SYSTem:ERRor[:NEXT]? を参照してください。

文法	SYSTem:ERRor:ADDITIONal[:MESSAge]?
説明	エラーメッセージに含まれる追加メッセージを問い合わせます。
パラメータ	無し
レスポンス	<message> = <CHARACTER RESPONSE DATA>
例	SYST:ERR:ADD? → NON
注	

2.3.4 SYSTem:DATE

文法	SYSTem:DATE <year>,<month>,<day>
説明	機器のカレンダー（年月日）を設定します。
パラメータ	<year> = <NUMERIC PROGRAM DATA> <i>MINimum = 1997, MAXimum = 2036</i> <month> = <NUMERIC PROGRAM DATA> <i>MINimum = 1, MAXimum = 12</i> <day> = <NUMERIC PROGRAM DATA> <i>MINimum = 1, MAXimum = 31</i>
レスポンス	無し
例	SYST:DATE 2009,12,31
注	

文法	SYSTem:DATE?
説明	機器のカレンダー（年月日）を問い合わせます。
パラメータ	無し
レスポンス	<year>,<month>,<day> = <NR1 NUMERIC RESPONSE DATA>
例	SYST:DATE? → 2009,07,04
注	

2.3.5 SYSTem:TIME

文法	SYSTem:TIME <hour>,<minute>,<second>
説明	機器の時刻を設定します。
パラメータ	<hour> = <NUMERIC PROGRAM DATA> <i>MINimum = 0, MAXimum = 23</i> <minute> = <NUMERIC PROGRAM DATA> <i>MINimum = 0, MAXimum = 59</i> <second> = <NUMERIC PROGRAM DATA> <i>MINimum = 0, MAXimum = 59</i>
レスポンス	無し
例	SYST:TIME 23,59,59
注	

文法	SYSTem:TIME?
説明	機器の時刻を問い合わせます。
パラメータ	無し
レスポンス	<hour>,<minute>,<second> = <NR1 NUMERIC RESPONSE DATA>
例	SYST:TIME? → 15,45,03
注	

2.3.6 SYSTem:REBoot

文法	SYSTem:REBoot
説明	本コマンドは強制的に機器をリブートします。 リモートコントロールのTCPセッションも強制的にクローズされます。
パラメータ	無し
レスポンス	無し
例	SYST:REB
注	

2.3.7 SYSTem:GPS:NSATellites?

文法	SYSTem:GPS:NSATellites?
説明	GPSによって検出された衛星の数を問い合わせます。
パラメータ	無し
レスポンス	<count> = <NR1 NUMERIC RESPONSE DATA>
例	SYST:GPS:NSAT? → 5
注	GPSが無効の場合は"0"を返します。

2.3.8 SYSTem:GPS:TIME?

文法	SYSTem:GPS:TIME?
説明	GPS時刻を問い合わせます。
パラメータ	無し
レスポンス	<time> = <CHARACTER RESPONSE DATA>
例	SYST:GPS:TIME? → 2014-01-01T12:34:56
注	GPSが無効の場合は"0"を返します。

2.3.9 SYSTem:GPS:LOCation?

文法	SYSTem:GPS:LOCation?
説明	位置を問い合わせます。
パラメータ	無し
レスポンス	<location> = <NR2 NUMERIC RESPONSE DATA>
例	SYST:GPS:LOC? → 85 26.8444N, 22 20.4508E
注	GPSが無効の場合は"0"を返します。

2.3.10 SYSTem:GPS:POWEr

文法	SYSTem:GPS:POWEr <power>
説明	GPSモジュールのパワーを切り替えます。 (1 - ON, 0 - OFF).
パラメータ	<power> = <NUMERIC PROGRAM DATA> <i>MINimum</i> = 0, <i>MAXimum</i> = 1
レスポンス	無し
例	SYST:GPS:POWE 0
注	

2.3.11 SYSTem:GPS:POWEr?

文法	SYSTem:GPS:POWEr?
説明	GPSモジュールのパワーを問い合わせます。 (1 - ON, 0 - OFF).
パラメータ	無し
レスポンス	<power> = <NR1 NUMERIC RESPONSE DATA>
例	SYST:GPS:POWE? → 0
注	

2.3.12 SYSTem:GPS:SYNCmode

文法	SYSTem:GPS:SYNCmode <mode>
説明	GPSモジュールの同期モードを切り替えます。 (1 - GPS, 0 - EXT).
パラメータ	<mode> = <NUMERIC PROGRAM DATA> <i>MINimum</i> = 0, <i>MAXimum</i> = 1
レスポンス	無し
例	SYST:GPS:SYNC 0
注	

2.3.13 SYSTem:GPS:SYNCmode?

文法	SYSTem:GPS:SYNCmode?
説明	GPSモジュールの同期モードを問い合わせます。(1 - GPS, 0 - EXT).
パラメータ	無し
レスポンス	<mode> = <NR1 NUMERIC RESPONSE DATA>
例	SYST:GPS:SYNC? → 0
注	

2.3.14 SYSTem:GPS:ANTNpower

文法	SYSTem:GPS:ANTNpower <power>
説明	GPSモジュールのアンテナ供給電圧を切り替えます。(1 - 5V, 0 - 3.3V).
パラメータ	<power> = <NUMERIC PROGRAM DATA> <i>MINimum</i> = 0, <i>MAXimum</i> = 1
レスポンス	無し
例	SYST:GPS:ANTN 0
注	

2.3.15 SYSTem:GPS:ANTNpower?

文法	SYSTem:GPS:ANTNpower?
説明	GPSモジュールのアンテナ供給電圧を問い合わせます。(1 - 5V, 0 - 3.3V).
パラメータ	無し
レスポンス	<power> = <NR1 NUMERIC RESPONSE DATA>
例	SYST:GPS:ANTN? → 0
注	

2.3.16 SYSTem:GPS:PWRElapsed?

文法	SYSTem:GPS:PWRElapsed?
説明	GPSモジュールの電源供給時間を問い合わせます。
パラメータ	無し
レスポンス	<time> = <NR1 NUMERIC RESPONSE DATA>
例	SYST:GPS:PWRE? → 0
注	

2.3.17 SYSTem:GPS:LOCElapsed?

文法	SYSTem:GPS:LOCElapsed?
説明	GPSモジュールの発信器ロック時間を問い合わせます。
パラメータ	無し
レスポンス	<time> = <NR1 NUMERIC RESPONSE DATA>
例	SYST:GPS:LOCE? → 0
注	

2.3.18 SYSTem:GPS:SYNElapsed?

文法	SYSTem:GPS:SYNElapsed?
説明	GPSモジュールの同時期間を問い合わせます。
パラメータ	無し
レスポンス	<time> = <NR1 NUMERIC RESPONSE DATA>
例	SYST:GPS:SYNE? → 0
注	

2.3.19 SYSTem:COMMunicate:TERMinator

文法	SYSTem:COMMunicate:TERMinator <terminator>
説明	問い合わせに対するレスポンスに表示されるターミネータコードを設定します。
パラメータ	<terminator> = <CHARACTER PROGRAM DATA> 無し (GPIBのみ) LF CRLF
レスポンス	無し
例	SYST:COMM:TERM LF
注	本設定は現在のセッションのみに適用され、セッションが終了すると、初期値であるCRLFに戻ります。

文法	SYSTem:COMMunicate:TERMinator?
説明	問い合わせに対するレスポンスに表示されるターミネータコードを問い合わせます。
パラメータ	無し
レスポンス	<terminator> = <CHARACTER RESPONSE DATA>
例	SYST:COMM:TERM? → LF
注	

2.3.20 SYSTem:PROMpt

文法	SYSTem:PROMpt <enable>
説明	リモート制御インタフェースからの応答にプロンプトを追加するかを設定します。
パラメータ	<enable> = <BOOLEAN PROGRAM DATA>
レスポンス	無し
例	SYST:PROM 1
注	プロンプトの文字は、"SCPI:>"です。 本設定は現在のセッションのみに適用され、セッションが終了すると無効になります。

文法	SYSTem:PROMpt?
説明	プロンプトの表示設定を問い合わせます。
パラメータ	無し
レスポンス	<enable> = <BOOLEAN PROGRAM DATA>
例	SYST:PROM? → SCPI:>1
注	

2.3.21 SYSTem:LOCal:CONTRol

文法	SYSTem:LOCal:CONTRol <enable>
説明	リモート制御中にローカルGUI制御可能とするかを設定します。
パラメータ	<enable> = <BOOLEAN PROGRAM DATA>
レスポンス	無し
例	SYST:LOC:CONT 1
注	本設定は現在のセッションのみに適用され、セッションが終了すると無効になります。

文法	SYSTem:LOCal:CONTRol?
説明	ローカルGUI制御設定を問い合わせます。
パラメータ	無し
レスポンス	<enable> = <BOOLEAN PROGRAM DATA>
例	SYST:LOC:CONT? → 1
注	

2.3.22 SYSTem:NCONnectivity:COMunication:AUTH:TYPE

文法	SYSTem:NCONnectivity:COMunication:AUTH:TYPE <type>
説明	HTTPリクエストにおける認証方法を設定します。
パラメータ	<type> = <CHARACTER PROGRAM DATA> NONE: 認証を使用しません。 BASIC: Basic認証を使用します。 OAUTH20: OAuth 2.0を使用します。
レスポンス	無し
例	SYST:NCON:COM:AUTH:TYPE BASIC
注	

文法	SYSTem:NCONnectivity:COMunication:AUTH:TYPE?
説明	HTTPリクエストにおける認証方法を問い合わせます。
パラメータ	無し
レスポンス	<type> = <CHARACTER RESPONSE DATA>
例	SYST:NCON:COM:AUTH:TYPE? → BASIC
注	

2.3.23 SYSTem:NCONnectivity:COMunication:BASic

文法	SYSTem:NCONnectivity:COMunication:BASic <id>,<password>
説明	Basic認証におけるIDおよびパスワードを設定します。
パラメータ	<id> = <STRING PROGRAM DATA> <password> = <STRING PROGRAM DATA>
レスポンス	無し
例	SYST:NCON:COM:BAS "id","password"
注	IDおよびパスワードは128文字まで設定できます。

文法	SYSTem:NCONnectivity:COMunication:BASic?
説明	Basic認証におけるIDおよびパスワードを問い合わせます。
パラメータ	無し
レスポンス	<id> = <CHARACTER RESPONSE DATA> <password> = <CHARACTER RESPONSE DATA>
例	SYST:NCON:COM:BAS? → id,*****
注	パスワードのレスポンスは文字数分のアスタリスクになります。

2.3.24 SYSTem:NCONnectivity:COMunication:TURi

文法	SYSTem:NCONnectivity:COMunication:TURi <uri>
説明	シナリオ実行結果をアップロードするURIを設定します。
パラメータ	<uri> = <STRING PROGRAM DATA>
レスポンス	無し
例	SYST:NCON:COM:TUR "https://sample.net/upload"
注	URIは2048文字まで設定できます。

文法	SYSTem:NCONnectivity:COMunication:TURi?
説明	シナリオ実行結果をアップロードするURIを問い合わせます。
パラメータ	無し
レスポンス	<uri> = <CHARACTER RESPONSE DATA>
例	SYST:NCON:COM:TUR? → https://sample.net/upload
注	

2.3.25 SYSTem:NCONnectivity:COMunication:SET

文法	SYSTem:NCONnectivity:COMunication:SET <uri>
説明	指定したURIに対してHTTP GETリクエストを送信します。
パラメータ	<uri> = <STRING PROGRAM DATA>
レスポンス	無し
例	SYST:NCON:COM:SET "https://sample.net/config?argument=sample" SYST:NCON:COM:LRET:RST? → 0 SYST:WAIT:DUR 5 SYST:NCON:COM:LRET:RST? → 1 SYST:NCON:COM:LRET:ERR? → 0,"No error." SYST:NCON:COM:LRET:RRES? → {"key":"value"}
注	URIは2048文字まで設定できます。

2.3.26 SYSTem:NCONnectivity:COMunication:LRETurn:RRESponse?

文法	SYSTem:NCONnectivity:COMunication:LRETurn:RRESponse?
説明	送信したHTTP GETリクエストに対するレスポンスを問い合わせます。
パラメータ	無し
レスポンス	<response>=<CHARACTER RESPONSE DATA>
例	SYST:NCON:COM:LRET:RRES? → {"key":"value"}
注	<response>の文字数は、最大65636です。

2.3.27 SYSTem:NCONnectivity:COMunication:LRETurn:RSTatus?

文法	SYSTem:NCONnectivity:COMunication:LRETurn:RSTatus?
説明	送信したHTTP GETリクエストに対するレスポンスの受信が完了したかを問い合わせます。
パラメータ	無し
レスポンス	<status>=<NR1 NUMERIC RESPONSE DATA> 0: 未完了 1: 完了
例	SYST:NCON:COM:LRET:RST? → 1
注	

2.3.28 SYSTem:NCONnectivity:COMunication:LRETurn:ERRor?

文法	SYSTem:NCONnectivity:COMunication:LRETurn:ERRor?
説明	HTTPレスポンスの受信に失敗したときのエラーを問い合わせます。
パラメータ	無し
レスポンス	<error>=<NR1 NUMERIC RESPONSE DATA> <message>=<STRING RESPONSE DATA> 0: エラーなし 1: ネットワークマスターがクラウドサーバに接続されていません。 2: ネットワークが切断されました。 3: クラウドサーバにログインしてOAuth 2.0の認証を実行してください。 4: クラウドサーバで予期しないエラーが発生しました。 5: 指定したサーバへの接続に失敗しました。 6: 指定したサーバが予期しないエラーを返しました。 7: 受信したレスポンスのフォーマットが無効です。 8: Basic認証のIDまたはパスワードが正しくありません。 9: リクエストがタイムアウトしました。 10: MX109020A-003のライセンスが有効ではありません。
例	SYST:NCON:COM:LRET:ERR? → 0,"No error."
注	

2.3.29 SYSTem:WAIT[:IDLE]

文法	SYSTem:WAIT[:IDLE]
説明	機器がIDLE状態 (たとえば、測定していない、あるいはテストが保留中、起動中、読み込み中、蓄積中の状態) になるまで待機します。 また、設定の読み込みと保存が終了するまで待機します。
パラメータ	無し
レスポンス	無し
例	SYST:WAIT
注	本コマンドが正しく認識されるためには、アプリケーションサーバが接続されている必要があります。 本コマンドを使用するときは、リモートインタフェースが停止しないように注意してください。 ある状態では、機器がアイドル状態にもどるためにリモートコマンドやユーザ操作を必要とすることがあります。たとえば測定を実行しているときに測定を停止するには、MANualを設定します (MEAS:SET:STOP → MAN)。この場合、機器がアイドル状態にもどるには正面パネルのSTART/STOPボタンを押すか、MEASurement:STOPコマンドを送信します。 SYST:WAIT:IDLEを実行すると、MEASurement:STOPコマンドではアイドル状態に戻りません。 予期しない通信の停止が発生した場合は、一度リモート接続を閉じて再度接続してください。 予期しない通信コネクションを停止させるため *RSTコマンドを送信ください

2.3.30 SYSTem:WAIT:DURation

文法	SYSTem:WAIT:DURation <seconds>
説明	指定した秒数だけ待ちます。
パラメータ	<seconds> = <NUMERIC PROGRAM DATA> MINimum = 1, MAXimum = 3600
レスポンス	無し
例	SYST:WAIT:DUR 5
注	本コマンドが正しく認識されるためには、アプリケーションサーバが接続されている必要があります。

2.4 SCPI Instrument サブシステムコマンド

アプリケーション固有のSCPIコマンドを使用するには、アプリケーションサーバにクライアントセッションを接続する必要があります。接続されたアプリケーションサーバに、SCPIコマンドを送信することで、そのアプリケーションを制御することができます。

2.4.1 アプリケーションサーバへの接続

アプリケーションサーバにクライアントセッションを接続するためには、以下の方法があります。

- **INST:CONN:ALL** コマンドを使用する
未接続のアプリケーションサーバにクライアントセッションを接続します。
すでに他のクライアントセッションが接続されている場合、接続に失敗します。
- **INST:CONN** コマンドを使用する
すべての未接続のアプリケーションサーバにクライアントセッションを接続します。
- **INST:STAR** コマンドを使用する
新しくアプリケーションサーバを立ち上げ、そのサーバにクライアントセッションを接続します。

2.4.2 複数アプリケーションの接続

一つのクライアントセッションは、同時に複数のアプリケーションサーバを接続することができます。このとき、SCPIコマンドの送信先は、選択されたアプリケーションサーバになります。

SCPIコマンドを送信するアプリケーションサーバは、**INST:SEL**コマンドを使用して設定でき、選択されているアプリケーションサーバを確認するためには、**INST?**コマンドを使用します。

INST:STARコマンドで新しくアプリケーションサーバを起動した場合、SCPIコマンドの送信先は、新しく起動したアプリケーションサーバに変更されます。

2.4.3 マルチユーザでの接続

マルチユーザでNetwork Masterを使用する場合、複数のクライアントセッションが1つのNetwork Masterに接続することになります。このとき、アプリケーションサーバがほかのユーザに使用されていないか、注意する必要があります。

Network Masterでは、クライアントセッションに接続されたアプリケーションサーバに、他のクライアントセッションを新しく接続することはできません。

他のクライアントセッションが接続しているアプリケーションサーバに接続するためには、接続中のクライアントセッションが接続を切り、アプリケーションサーバを解放する必要があります。アプリケーションサーバを解放するためには、以下の方法があります

- **INST:DISC**コマンドを使用する
アプリケーションサーバに接続しているクライアントセッションから**INST:DISC**コマンドを送ることでサーバとの接続を切断できます。
- クライアントセッションを終了する
クライアントセッションが終了すると、接続していたすべてのアプリケーションサーバが解放されます。
ユーザの使用するPCからNetwork Masterとの接続を切断することで、クライアントセッションを終了することができます。

解放されたアプリケーションサーバは、**INST:CONN**コマンドあるいは、**INST:CONN:ALL**コマンドによって再度クライアントセッションを接続できるようになります。

2.4.4 アプリケーションサーバの強制終了

他のセッションに接続されているアプリケーションサーバは、別のセッションからのSCPIコマンドを受け付けません。

例外として、別のクライアントセッションであっても、**INST:TERM:FORC** コマンドは受け付けられ、他のセッションに接続されているアプリケーションサーバを終了することができます。

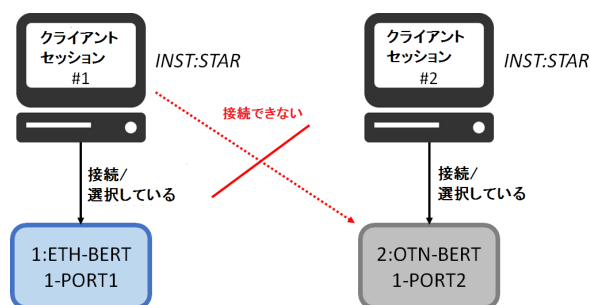


図 2.1: INST:STAR は自動的に 接続して、開始アプリケーションサーバを選択します

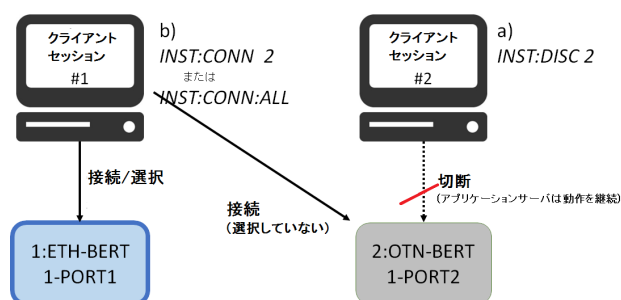


図 2.2: 別のクライアントセッションを 接続する前に、最初のクライアントセッションを切断する必要があります

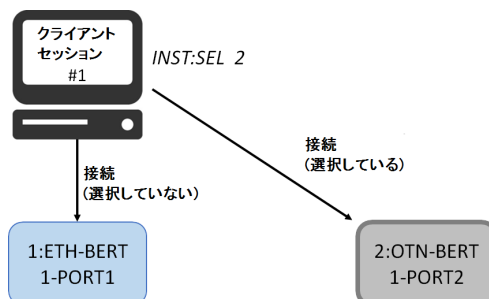


図 2.3: 複数のアプリケーションサーバに接続している場合は、クライアントセッションはどのアプリケーションサーバにアプリケーション固有コマンドを送信するか選択します

2.4.5 INSTRument:STARt[:DEFault]

文法	INSTRument:STARt[:DEFault] <app name>[, <port name>[, <port name>,...]]
説明	アプリケーションサーバを開始します。
パラメータ	<app name> = <CHARACTER PROGRAM DATA> TP-APS-OTN: OTN Automatic Protection Switching application. TP-APS-SDHPDH: SDH/PDH Automatic Protection Switching application. TP-APS-SDHPDH-OTN: SDH/PDH over OTN Automatic Protection Switching application. TP-BERT-CPRI: CPRI Bit Error Rate Test application. TP-BERT-CPRI-OTN: CPRI over OTN Bit Error Rate Test application. TP-BERT-ETH: Ethernet Bit Error Rate Test application. TP-BERT-ETH-OTN: Ethernet over OTN Bit Error Rate Test application. TP-BERT-FC: Fibre Channel Bit Error Rate Test application.

次のページに続く...

... 前のページから続く

	TP-BERT-FC-OTN: Fibre Channel over OTN Bit Error Rate Test application. TP-BERT-OTN: OTN Bit Error Rate Test application. TP-BERT-ROE: eCPRI/RoE Bit Error Rate Test application. TP-BERT-SDHPDH: PDH/SDH Bit Error Rate Test application. TP-BERT-SDHPDH-OTN: PDH/SDH over OTN Bit Error Rate Test application. TP-CABLE-ETH: Ethernet cable test application. TP-CHSTAT-ETH: Ethernet channel statistics application. TP-DISC-ETH: Ethernet discovery application. TP-MONGEN-ETH: Ethernet monitor/generate application. TP-MONGEN-ETH-OTN: Ethernet over OTN monitor/generate application. TP-NOFRAME-DEVICE: No frame device test (Unframed Bit Error Rate Test) application. TP-PASS-CPRI: CPRI pass-through application. TP-PASS-ETH: Ethernet pass-through application. TP-PERF-FC: Fibre Channel performance test application. TP-PERF-FC-OTN: Fibre Channel over OTN performance test application. TP-PING-ETH: Ethernet ICMP ping application. TP-REFL-ETH: Ethernet reflector application. TP-REFL-ETH-OTN: Ethernet over OTN reflector application. TP-REFL-FC: Fibre Channel reflector application. TP-REFL-FC-OTN: Fibre Channel over OTN reflector application. TP-RFC-ETH: Ethernet RFC-2544 test application. TP-RFC-ETH-OTN: Ethernet over OTN RFC-2544 test application. TP-RFC6349-ETH: Ethernet RFC-6349 test application. TP-RTD-OTN: OTN Round Trip Delay test application. TP-RTD-SDHPDH: SDH/PDH Round Trip Delay test application. TP-RTD-SDHPDH-OTN: SDH/PDH Round Trip Delay test application. TP-SAT-ETH: Ethernet Service Activation Test application. TP-SAT-ETH-OTN: Ethernet over OTN Service Activation Test application. TP-TRACE-ETH: Ethernet trace-route application. TP-SYNCTEST-ETH: Ethernet sync test application. OTDR-OTDR: Standard OTDR application. OTDR-OLTS: OLTS application.
	<port name> = <CHARACTER PROGRAM DATA> 1-PORT1: モジュール1のポート1 1-PORT2: モジュール2のポート1 2-PORT1: モジュール1のポート2 2-PORT2: モジュール2のポート2 1-PORT-SING1: モジュール1のポート1のみ (MU110011A/MU110012A/MU110013Aの場合) 1-PORT-MULT1: モジュール1のポート1 (TP-BERT-ROEで”25G eCPRI/RoE 2ポート開放”を使用する場合) 1-PORT-MULT2: モジュール1のポート2 (TP-BERT-ROEで”25G eCPRI/RoE 2ポート開放”を使用する場合) パラメータで指定した物理ポートは、アプリケーションサーバで論理ポートとして割り当てられます。 論理ポート番号は1から、アプリケーションサーバに割り当てたポートの数となります。
レスポンス	無し
例	INST:STAR TP-BERT-OTN,1-PORT1 INST? → 2
注	INSTrument:SElect?コマンドでアプリケーションのサーバIDを取得できます。 本コマンドでアプリケーションサーバを開始すると、起動したアプリケーションサーバは自動的に接続状態になり、今後発行されるコマンドの制御対象になります。 また、アプリケーションサーバは初期状態の設定 (DEFault) で起動します。

2.4.6 INStrument:STARt:LAST

文法	INStrument:STARt:LAST <app name>[, <port name>[, <port name>,...]]
説明	アプリケーションサーバを開始し、オートセーブされた設定を読み込みます。
パラメータ	指定できるパラメータは上記のINStrument:STARt[:DEFault] コマンドと同じです。
レスポンス	無し
例	INST:STAR:LAST TP-BERT-OTN,1-PORT1 INST? → 2
注	本コマンドでアプリケーションサーバを開始すると、起動したアプリケーションサーバは自動的に接続状態になり、今後発行されるコマンドの制御対象になります。 また、アプリケーションサーバは前回使用したときの設定で起動します。

2.4.7 INStrument:STARt:GUI

文法	INStrument:STARt:GUI <test index>
説明	リモート制御中のGUI表示を開始します。
パラメータ	<test index> = <NUMERIC PROGRAM DATA>
レスポンス	無し
例	INST:STAR TP-BERT-OTN,1-PORT1 INST? → 2 INST:STAR:GUI 2
注	最初にアプリケーションサーバに接続する必要があります。 サーバインデックスを省略した場合は、現在のアプリケーションサーバが対象となります。

2.4.8 INStrument:TERMinate

文法	INStrument:TERMinate <test index>
説明	アプリケーションサーバを終了します。
パラメータ	<test index> = <NUMERIC PROGRAM DATA>
レスポンス	無し
例	INST:STAR TP-BERT-OTN,1-PORT1 INST? → 2 INST:TERM 2
注	最初にアプリケーションサーバに接続する必要があります。 サーバインデックスを省略した場合は、現在のアプリケーションサーバが対象となります。

2.4.9 INStrument:TERMinate:FORCe

文法	INStrument:TERMinate:FORCe [<test index>]
説明	指定したアプリケーションサーバを強制的に終了します。
パラメータ	<test index> = <NUMERIC PROGRAM DATA>
レスポンス	無し
例	INST:STAR TP-BERT-OTN,1-PORT1 INST? → 2 INST:TERM:FORC 2
注	警告: このコマンドは他のセッションで接続されたアプリケーションを強制的に終了することができます。 サーバインデックスを省略した場合は、現在のアプリケーションサーバが対象となります。

2.4.10 INStrument:COUNT?

文法	INStrument:COUNT?
説明	アクティブなアプリケーションサーバの数を問い合わせます。
パラメータ	無し
レスポンス	<count> = <NR1 NUMERIC RESPONSE DATA>
例	INST:COUN? → 2
注	

2.4.11 INStrument:CATalog?

文法	INStrument:CATalog?
説明	アクティブなすべてのアプリケーションサーバのテストインデックス、アプリケーション名、およびポート名を問い合わせます。
パラメータ	無し
レスポンス	<test index> = <EXPRESSION RESPONSE DATA> 表示形式: (<test index>,<app name>,<port name>)
例	INST:CAT? → (1,TP-SAT-ETH,1-PORT1),(2,TP-BERT-SDHPDH,1-PORT2)
注	アプリケーションサーバが開始されていない場合は-1を返します。

2.4.12 INStrument:STATe?

文法	INStrument:STATe? <test index>
説明	任意のアプリケーションサーバに関するステータス情報を問い合わせます。
パラメータ	<test index> = <NUMERIC PROGRAM DATA>
レスポンス	<app name> = <CHARACTER RESPONSE DATA> <client connection> = <CHARACTER RESPONSE DATA> NON IP Address <select status> = <CHARACTER RESPONSE DATA> NON SELECTED <port name> = <CHARACTER RESPONSE DATA> 表示形式:文字リスト
例	INST:STAT? 1 → TP-BERT-OTN,192.168.128.21,SELECTED,1-PORT1,2-PORT2...
注	

2.4.13 INStrument:CONNect

文法	INStrument:CONNect <test index>
説明	クライアントセッションが既存のアプリケーションサーバに再接続できるようにします。
パラメータ	<test index> = <NUMERIC PROGRAM DATA>
レスポンス	無し
例	INST:CONN 1
注	本コマンドは、アプリケーションサーバが既存のどのクライアントセッションにも属していない場合にのみ正常に実行されます。 INStrument:CATalog? クエリを使用して、既存のすべてのアプリケーションサーバのリストを取得します。 コマンドが正常に実行されると、自動的にアプリケーションサーバが選択されます。

2.4.14 INSTrument:CONNect:ALL

文法	INSTrument:CONNect:ALL
説明	クライアントセッションが既存のすべてのアプリケーションサーバに再接続できるようにします。
パラメータ	無し
レスポンス	無し
例	INST:CONN:ALL
注	本コマンドは、少なくとも1つのアプリケーションサーバが既存のどのクライアントセッションにも属していない場合にのみ正常に実行されます。 複数のアプリケーションサーバに接続する場合は、最小のインデックス番号を持つアプリケーションサーバが選択されます。ただし、本コマンドの実行前にアプリケーションサーバがすでに選択されている場合、選択されたインデックスは変化しません。

2.4.15 INSTrument:CONNect[:CATalog]?

文法	INSTrument:CONNect[:CATalog]?
説明	現在のクライアントセッションに対するすべてのアプリケーションサーバのインデックスを問い合わせます。
パラメータ	無し
レスポンス	<test index> = <EXPRESSION RESPONSE DATA> 表示形式: 数値リスト
例	INST:CONN? → 0,1,...
注	現在のクライアントセッションに属しているアプリケーションサーバがない場合は、-1を返します。

2.4.16 INSTrument:DISConnect

文法	INSTrument:DISConnect <test index>
説明	アプリケーションサーバをクライアントセッションから切断します。
パラメータ	<test index> = <NUMERIC PROGRAM DATA>
レスポンス	無し
例	INST:DISC 1
注	現在のアプリケーションが切断されると、最小のアプリケーションサーバIDが選択されます。 リモートセッションが切断されると、現在接続しているすべてのアプリケーションサーバが自動的に切断されます。

2.4.17 INSTrument[:SElect]

文法	INSTrument[:SElect] <test index>
説明	現在のアプリケーションサーバを選択します。
パラメータ	<test index> = <NUMERIC PROGRAM DATA>
レスポンス	無し
例	INST:STAR TP-BERT-OTN,1-PORT1 INST? → 1 INST:STAR TP-BERT-OTN,1-PORT2 INST? → 2 INST 1
注	今後発行されるコマンドはすべて、現在のアプリケーションサーバが変更されるまで選択したアプリケーションサーバに送られます。

文法	INSTrument[:SElect]?
説明	現在のアプリケーションサーバのインデックスを問い合わせます。
パラメータ	無し
レスポンス	<count> = <NR1 NUMERIC RESPONSE DATA>
例	INST:STAR TP-BERT-OTN,1-PORT1 INST? → 1 INST:STAR TP-BERT-OTN,1-PORT2 INST? → 2 INST 1
注	現在のクライアントセッションで現在のアプリケーションサーバが選択されていない場合は-1を返します。

2.4.18 INSTrument:ERRor[:NEXT]?

文法	INSTrument:ERRor[:NEXT]?
説明	現在のアプリケーションサーバ用のイベントキューの一番古い情報を取り出します。取り出した情報はキューから削除されます。
パラメータ	無し
レスポンス	<description> = <STRING RESPONSE DATA>
例	INST:ERR? → "Signal abnormal"
注	アプリケーションサーバは独自のイベントキューを備えています。 このイベントキューは、セッションの終了時に破棄されません。 本コマンドを使用してサーバステータスを確認する前に、コントローラがステータス・バイト・レジスタのDB2 (イベントキューサマリ) をチェックして、DB2が設定されているかどうか確認することがあります。 ステータス・バイト・レジスタのDB2は、潜在的に多数のアプリケーションサーバからの入力を集約し、すべてのサーバのイベントキューが空になった場合にデータをクリアします。

2.4.19 INSTrument:PORT?

文法	INSTrument:PORT?
説明	現在のアプリケーションサーバ用に割り当てられたポートを問い合わせます。
パラメータ	無し
レスポンス	<port name> = <CHARACTER RESPONSE DATA> 表示形式: 文字リスト
例	INST:STAR TP-BERT-OTN,1-PORT1,1-PORT2 INST:PORT? → 1-PORT1,1-PORT2
注	INSTrument:SElect commandコマンドにより現在選択されているアプリケーションサーバを対象とします。 現在のクライアントセッションで現在のアプリケーションサーバが選択されていない場合はNONを返します。 現在のアプリケーションサーバ用にポートが割り当てられていない場合はNONを返します。

2.4.20 INSTrument:PORT:FREE?

文法	INSTrument:PORT:FREE? <app name>
説明	対象アプリケーションにおける利用可能なポートを問い合わせます。
パラメータ	<app name> = <CHARACTER PROGRAM DATA>
レスポンス	<port name> = <CHARACTER RESPONSE DATA> 表示形式: 文字リスト
例	INST:PORT:FREE? TP-BERT-OTN → 1-PORT1,1-PORT2
注	

2.4.21 INSTRument:PORT:CATalog?

文法	INSTRument:PORT:CATalog?
説明	機器のすべてのポートを問い合わせます。
パラメータ	無し
レスポンス	<port name> = <CHARACTER RESPONSE DATA> 表示形式:文字リスト
例	INST:PORT:CAT? → 1-PORT1,1-PORT2
注	SCPIクライアントユーザがいずれかのSTATus:PORT:コマンドを使用するとき、<bit>インデックスは、INSTRument:PORT:CATalog?によって返されるデータに一致します。

2.4.22 INSTRument:MODule:CATalog?

文法	INSTRument:MODule:CATalog?
説明	モジュール形名のリストを問い合わせます。
パラメータ	無し
レスポンス	{<module n>,*} = <CHARACTER RESPONSE DATA>
例	INST:MOD:CAT? → MU100010A,MU100011A
注	

2.4.23 INSTRument:CTRL:NAME?

文法	INSTRument:CTRL:NAME?
説明	コントローラの形名を問い合わせます。
パラメータ	無し
レスポンス	<model name> = <CHARACTER RESPONSE DATA>
例	INST:CTRL:NAME? → MT1000A
注	

2.4.24 INSTRument:CTRL:SN?

文法	INSTRument:CTRL:SN?
説明	コントローラのシリアルナンバーを問い合わせます。
パラメータ	無し
レスポンス	<serial number> = <CHARACTER RESPONSE DATA>
例	INST:CTRL:SN? → 1234567890
注	

2.4.25 INSTRument:CTRL:TRT?

文法	INSTRument:CTRL:TRT?
説明	コントローラの積算稼働時間 (秒)を問い合わせます。
パラメータ	無し
レスポンス	<time> = <NR1 NUMERIC RESPONSE DATA>
例	INST:CTRL:TRT? → 5000000
注	

2.4.26 INSTRument:CTRL:OPTion:CATalog?

文法	INSTRument:CTRL:OPTion:CATalog?
説明	コントローラに搭載されているオプションのリストを問い合わせます。
パラメータ	無し
レスポンス	{<option n>,*} = <CHARACTER RESPONSE DATA>
例	INST:CTRL:OPT:CAT? → MT1000A-303,MT1000A-005
注	

2.4.27 INSTRument:MODule<Md>:NAME?

文法	INSTRument:MODule<Md>:NAME?
説明	モジュールの形名を問い合わせます。
パラメータ	無し
レスポンス	<model name> = <CHARACTER RESPONSE DATA>
例	INST:MOD1:NAME? → MU100010A
注	

2.4.28 INSTRument:MODule<Md>:SN?

文法	INSTRument:MODule<Md>:SN?
説明	モジュールのシリアルナンバーを問い合わせます。
パラメータ	無し
レスポンス	<serial number> = <CHARACTER RESPONSE DATA>
例	INST:MOD1:SN? → 1234567890
注	

2.4.29 INSTRument:MODule<Md>:TRT?

文法	INSTRument:MODule<Md>:TRT?
説明	モジュールの積算稼働時間(秒)を問い合わせます。
パラメータ	無し
レスポンス	<time> = <NR1 NUMERIC RESPONSE DATA>
例	INST:MOD1:TRT? → 5000000
注	

2.4.30 INSTRument:MODule<Md>:OPTion:CATalog?

文法	INSTRument:MODule<Md>:OPTion:CATalog?
説明	モジュールに搭載されているオプションのリストを問い合わせます。
パラメータ	無し
レスポンス	{<option n>,}* = <CHARACTER RESPONSE DATA>
例	INST:MOD1:OPT:CAT? → MU100010A-001,MU100010A-002
注	

2.5 SCPI Status サブシステムコマンド

2.5.1 STATus:OPERation[:EVENT]?

文法	STATus:OPERation[:EVENT]?
説明	オペレーションイベントレジスタのデータを問い合わせます。本コマンドでデータを問い合わせると、オペレーションイベントレジスタはクリアされます。
パラメータ	無し
レスポンス	<value> = <NR1 NUMERIC RESPONSE DATA> レジスタのビットと値: DB1-DB4 = 未使用 DB5 (16) = Measuring DB6-DB16 = 未使用
例	STAT:OPER? → 16
注	すべての実行中のセッションにはそれぞれレジスタを持っています。セッションが開始したときにレジスタはクリアされます。

2.5.2 STATus:OPERation:CONDition?

文法	STATus:OPERation:CONDition?
説明	オペレーションコンディションレジスタのデータを問い合わせます。
パラメータ	無し
レスポンス	<value> = <NR1 NUMERIC RESPONSE DATA> レジスタのビットと値: DB1-DB4 = 未使用 DB5 (16) = Measuring DB6-DB16 = 未使用
例	STAT:OPER:COND? → 16
注	

2.5.3 STATus:OPERation:ENABLE

文法	STATus:OPERation:ENABLE <mask>
説明	オペレーションイベントレジスタのイネーブルマスクを設定します。
パラメータ	<mask> = <NUMERIC PROGRAM DATA> レジスタのビットと値 DB1-DB4 = 未使用 DB5 (16) = Measuring DB6-DB16 = 未使用 <i>MINimum = 0, MAXimum = 65535</i>
レスポンス	無し
例	STAT:OPER:ENAB 65535
注	

文法	STATus:OPERation:ENABLE?
説明	オペレーションイベントレジスタのイネーブルマスクを問い合わせます。
パラメータ	無し
レスポンス	<mask> = <NR1 NUMERIC RESPONSE DATA>
例	STAT:OPER:ENAB? → 16
注	

2.5.4 STATus:OPERation:PTRansition

文法	STATus:OPERation:PTRansition <mask>
説明	オペレーションイベントレジスタのポジティブトランジションフィルタを設定します。
パラメータ	<mask> = <NUMERIC PROGRAM DATA> レジスタのビットと値: DB1-DB4 = 未使用 DB5 (16) = Measuring DB6-DB16 = 未使用 <i>MINimum = 0, DEFault = 65535, MAXimum = 65535</i>
レスポンス	無し
例	STAT:OPER:PTR 16384
注	

文法	STATus:OPERation:PTRansition?
説明	オペレーションイベントレジスタのポジティブトランジションフィルタを問い合わせます。
パラメータ	無し
レスポンス	<mask> = <NR1 NUMERIC RESPONSE DATA>
例	STAT:OPER:PTR? → 16384
注	

2.5.5 STATus:OPERation:NTRansition

文法	STATus:OPERation:NTRansition <mask>
説明	オペレーションイベントレジスタのネガティブトランジションフィルタを設定します。
パラメータ	<mask> = <NUMERIC PROGRAM DATA> レジスタのビットと値: DB1-DB4 = 未使用 DB5 (16) = Measuring DB6-DB16 = 未使用 <i>MINimum = 0, MAXimum = 65535</i>
レスポンス	無し
例	STAT:OPER:NTR 16384
注	

文法	STATus:OPERation:NTRansition?
説明	オペレーションイベントレジスタのネガティブトランジションフィルタを問い合わせます。
パラメータ	無し
レスポンス	<mask> = <NR1 NUMERIC RESPONSE DATA>
例	STAT:OPER:NTR? → 16384
注	

2.5.6 STATus:QUEStionable[:EVENT]?

文法	STATus:QUEStionable[:EVENT]?
説明	クエスチョナブルイベントレジスタのデータを問い合わせます。本コマンドでデータを問い合わせると、クエスチョナブルイベントレジスタはクリアされます。
パラメータ	無し
レスポンス	<value> = <NR1 NUMERIC RESPONSE DATA> レジスタのビットと値: DB1-DB14 = 未使用 DB15 (16384) = Command Warning DB16 = 未使用
例	STAT:QUES? → 16384
注	すべての実行中のセッションにはそれぞれレジスタを持っています。セッションが開始したときにレジスタはクリアされます。

2.5.7 STATus:QUEStionable:CONDition?

文法	STATus:QUEStionable:CONDition?
説明	クエスチョナブルコンディションレジスタのデータを問い合わせます。
パラメータ	無し
レスポンス	<value> = <NR1 NUMERIC RESPONSE DATA> レジスタのビットと値: DB1-DB14 = 未使用 DB15 (16384) = Command Warning DB16 = 未使用
例	STAT:QUES:COND? → 16384
注	

2.5.8 STATus:QUEStionable:ENABle

文法	STATus:QUEStionable:ENABle <mask>
説明	クエスチョナブルイベントレジスタのイネーブルマスクを設定します。
パラメータ	<mask> = <NUMERIC PROGRAM DATA> レジスタのビットと値: DB1-DB14 = 未使用 DB15 (16384) = Command Warning DB16 = 未使用 <i>MINimum = 0, MAXimum = 65535</i>
レスポンス	無し
例	STAT:QUES:ENAB 16384
注	

文法	STATus:QUEStionable:ENABle?
説明	クエスチョナブルイベントレジスタのイネーブルマスクを問い合わせます。
パラメータ	無し
レスポンス	<mask> = <NR1 NUMERIC RESPONSE DATA>
例	STAT:QUES:ENAB? → 16384
注	

2.5.9 STATus:QUEStionable:PTRansition

文法	STATus:QUEStionable:PTRansition <mask>
説明	クエスチョナブルイベントレジスタのポジティブトランジションフィルタを設定します。
パラメータ	<mask> = <NUMERIC PROGRAM DATA> レジスタのビットと値: DB1-DB14 = 未使用 DB15 (16384) = Command Warning DB16 = 未使用 <i>MINimum = 0, DEFault = 65535, MAXimum = 65535</i>
レスポンス	無し
例	STAT:QUES:PTR 16384
注	

文法	STATus:QUEStionable:PTRansition?
説明	クエスチョナブルイベントレジスタのポジティブトランジションフィルタを問い合わせます。
パラメータ	無し
レスポンス	<mask> = <NR1 NUMERIC RESPONSE DATA>
例	STAT:QUES:PTR? → 16384
注	

2.5.10 STATus:QUEStionable:NTRansition

文法	STATus:QUEStionable:NTRansition <mask>
説明	クエスチョナブルイベントレジスタのネガティブトランジションフィルタを設定します。
パラメータ	<mask> = <NUMERIC PROGRAM DATA> レジスタのビットと値 DB1-DB14 = 未使用 DB15 (16384) = Command Warning DB16 = 未使用 <i>MINimum = 0, MAXimum = 65535</i>
レスポンス	無し
例	STAT:QUES:NTR 16384
注	

文法	STATus:QUEStionable:NTRansition?
説明	クエスチョナブルイベントレジスタのネガティブトランジションフィルタを問い合わせます。
パラメータ	無し
レスポンス	<mask> = <NR1 NUMERIC RESPONSE DATA>
例	STAT:QUES:NTR? → 16384
注	

2.5.11 STATus:PORT[:EVENT]?

文法	STATus:PORT[:EVENT]? <port name>
説明	ポートイベントレジスタのデータを問い合わせます。本コマンドでデータを問い合わせると、ポートイベントレジスタはクリアされます。
パラメータ	<port name> = <CHARACTER RESPONSE DATA> 表示形式: 文字リスト
レスポンス	<enablel> = <BOOLEAN RESPONSE DATA>
例	STAT:PORT? 1-PORT1 → 1
注	すべての実行中のセッションにはそれぞれレジスタを持っています。セッションが開始したときにレジスタはクリアされます。 <port name> は、INSTrument:PORT:CATalog?によって返されるポートの1つに一致します。

2.5.12 STATus:PORT:CONDition?

文法	STATus:PORT:CONDition? <port name>
説明	ポートコンディションレジスタのデータを問い合わせます。
パラメータ	<port name> = <CHARACTER RESPONSE DATA> 表示形式: 文字リスト
レスポンス	<enablel> = <BOOLEAN RESPONSE DATA>
例	STAT:PORT:COND? 1-PORT1 → 1
注	<port name> は、INSTrument:PORT:CATalog?によって返されるポートの1つに一致します。

2.5.13 STATus:PORT:ENABLE

文法	STATus:PORT:ENABLE <port name>,<enablel>
説明	ポートイベントレジスタのイネーブルマスクを設定します。
パラメータ	<port name> = <CHARACTER RESPONSE DATA> 表示形式: 文字リスト <enablel> = <BOOLEAN PROGRAM DATA>
レスポンス	無し
例	STAT:PORT:ENAB 1-PORT1,ON
注	<port name> は、INSTrument:PORT:CATalog?によって返されるポートの1つに一致します。

文法	STATus:PORT:ENABLE? <port name>
説明	ポートイベントレジスタのイネーブルマスクを問い合わせます。
パラメータ	<port name> = <CHARACTER RESPONSE DATA> 表示形式: 文字リスト
レスポンス	<enablel> = <BOOLEAN RESPONSE DATA>
例	STAT:PORT:ENAB? 1-PORT1 → 1
注	<port name> は、INSTrument:PORT:CATalog?によって返されるポートの1つに一致します。

2.5.14 STATus:PORT:PTRansition

文法	STATus:PORT:PTRansition <port name>,<enablel>
説明	ポートイベントレジスタのポジティブトランジションフィルタを設定します。
パラメータ	<port name> = <CHARACTER RESPONSE DATA> 表示形式: 文字リスト <enablel> = <BOOLEAN PROGRAM DATA>
レスポンス	無し
例	STAT:PORT:PTR 1-PORT1,ON
注	<port name> は、INSTrument:PORT:CATalog?によって返されるポートの1つに一致します。

文法	STATus:PORT:PTRansition? <port name>
説明	ポートイベントレジスタのポジティブトランジションフィルタを問い合わせます。
パラメータ	<port name> = <CHARACTER RESPONSE DATA> 表示形式:文字リスト
レスポンス	<enable> = <BOOLEAN RESPONSE DATA>
例	STAT:PORT:PTR? 1-PORT1 → 1
注	<port name> は, INSTRument:PORT:CATalog?によって返されるポートの1つに一致します。

2.5.15 STATus:PORT:NTRansition

文法	STATus:PORT:NTRansition <port name>,<enable>
説明	ポートイベントレジスタのネガティブトランジションフィルタを設定します。
パラメータ	<port name> = <CHARACTER RESPONSE DATA> 表示形式:文字リスト <enable> = <BOOLEAN PROGRAM DATA>
レスポンス	無し
例	STAT:PORT:NTR 1-PORT1,OFF
注	<port name> は, INSTRument:PORT:CATalog?によって返されるポートの1つに一致します。

文法	STATus:PORT:NTRansition? <port name>
説明	ポートイベントレジスタのネガティブトランジションフィルタを問い合わせます。
パラメータ	<port name> = <CHARACTER RESPONSE DATA> 表示形式:文字リスト
レスポンス	<enable> = <BOOLEAN RESPONSE DATA>
例	STAT:PORT:NTR? 1-PORT1 → 0
注	<port name> は, INSTRument:PORT:CATalog?によって返されるポートの1つに一致します。

2.5.16 STATus:PRESet

文法	STATus:PRESet
説明	機器に依存するステータスデータ構造に対して, PRESetコマンドはイネーブルレジスタをすべて1に設定し, ポジティブトランジションを検出するようにします。 SCPIで規定されるステータスデータ構造 (オペレーション, クエスチョナブル, およびポートステータス) に対して, PRESetコマンドはトランジションフィルタレジスタを設定し, ポジティブトランジションを検出するようにします。また, イネーブルレジスタを0に設定します。 本コマンドは, ステータス・バイト・レジスタ, 標準イベントステータスレジスタに影響しません。PRESetは, どのイベントレジスタ, およびどのエラー/イベントキューもクリアしません。
パラメータ	無し
レスポンス	無し
例	STAT:PRES
注	すべての実行中のセッションにはそれぞれレジスタを持っています。 本コマンドはすべての接続されているアプリケーションサーバのレジスタに影響します。

2.6 Mass Memory サブシステムコマンド

本コマンド群は以下ストレージエリアでのファイル/ディレクトリ操作を実行します。

ロケーション	詳細
Internal/	ネットワークマスタ内部ストレージ
Usb/	USB接続されたストレージ 本ロケーションはUSBストレージが接続された時のみアクセス可能です。
Internal/remote/	外部ネットワークデバイス 外部ネットワークデバイス接続方法はユーザマニュアルを参照ください。 本ロケーションは外部ネットワークデバイスが設定され、接続可能な場合のみアクセス可能です。

ファイルは上記ロケーションあるいはそのサブディレクトリにのみ作成可能です。

2.6.1 MMEemory:LOAD

文法	MMEemory:LOAD <file>
説明	現在のアプリケーションサーバで指定されたファイルを読み込みます。 読み込むファイルは設定または結果のデータの両方を含むまたはどちらかです。
パラメータ	<file> = <STRING PROGRAM DATA> 読み込むファイルのパスと名称
レスポンス	無し
例	MME:LOAD "Internal/otdr-settings.cfg"
注	本コマンドが正しく認識されるためには、アプリケーションサーバが接続されている必要があります。 アプリケーションサーバはIdle状態である必要があります。読み込むファイルの形式はアプリケーションサーバの形式と一致している必要があります。 ロードするファイルは Internal/ ディレクトリ、またはそのサブディレクトリに存在する必要があります。 USBストレージデバイスがマウントされている場合は、ファイルは Usb/ ディレクトリでアクセスできます。

2.6.2 MMEemory:STORe:STATe

文法	MMEemory:STORe:STATe <file>
説明	機器の指定されたファイルに現在の設定を保存します。
パラメータ	<file> = <STRING PROGRAM DATA> データを保存するファイルのパスと名称
レスポンス	無し
例	MME:STOR:STAT "Internal/my-otdr-settings.cfg"
注	本コマンドが正しく認識されるためには、アプリケーションサーバが接続されている必要があります。 アプリケーションサーバはIdle状態である必要があります。 USBストレージデバイスがマウントされている場合は、ファイルは Usb/ ディレクトリでアクセスできます。

2.6.3 MMEMory:STORe:DATA

文法	MMEMory:STORe:DATA <file>
説明	機器の指定されたファイルに現在の設定と結果のデータを保存します。
パラメータ	<file> = <STRING PROGRAM DATA> データを保存するファイルのパスと名称
レスポンス	無し
例	MMEM:STOR:DATA "Usb/my-otdr-result.res"
注	本コマンドが正しく認識されるためには、アプリケーションサーバが接続されている必要があります。 アプリケーションサーバはIdle状態である必要があります。 USBストレージデバイスがマウントされている場合は、ファイルは Usb/ ディレクトリでアクセスできます。

2.6.4 MMEMory:DELeTe

文法	MMEMory:DELeTe <file>
説明	ファイルを削除します。
パラメータ	<file> = <STRING PROGRAM DATA> 削除するファイルパス
レスポンス	無し
例	MMEM:DEL "Internal/report.pdf"
注	

2.6.5 MMEMory:DATA?

文法	MMEMory:DATA? <file>
説明	本コマンドはファイルのデータを取得します。
パラメータ	<file> = <STRING PROGRAM DATA> 取得するファイルパス
レスポンス	<DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> = #<nonzero digit><digits><8 bit data bytes>, where: <nonzero digit> は1バイトのASCIIキャラクタです。'1'-'9'。この値は <digits> の長さを示します。 <digits> は'0'-'9'のASCIIキャラクタで、後に続くデータバイト長を10進数で示します。
例	MMEM:DATA? "Internal/report.pdf" → #49137<9137 bytes of binary data>
注	本コマンドは複合コマンドで他コマンドと併用することはできません。

2.6.6 MMEMory:COpy

文法	MMEMory:COpy <source-file>,<destination-file>
説明	ファイルをコピーします。
パラメータ	<source-file> = <STRING PROGRAM DATA> コピー元のファイルパス <destination-file> = <STRING PROGRAM DATA> コピー先のファイルパス
レスポンス	無し
例	MMEM:COpy "Internal/report.pdf", "Usb/report.pdf"
注	

2.6.7 MMEemory:MOVE

文法	MMEemory:MOVE <old-file>,<new-file>
説明	ファイルをリネームあるいは移動します。
パラメータ	<old-file> = <STRING PROGRAM DATA> リネームあるいは移動するファイルパス <new-file> = <STRING PROGRAM DATA> リネームあるいは移動先のファイルパス
レスポンス	無し
例	MMEM:MOVE "Internal/report.pdf","Usb/report.pdf"
注	

2.6.8 MMEemory:INFO?

文法	MMEemory:INFO? <file>
説明	ファイルの情報を問い合わせます。
パラメータ	<file> = <STRING PROGRAM DATA> 問い合わせるファイルのパス
レスポンス	<file-date-time> = <STRING RESPONSE DATA> 最終変更時間 <file-size> = <NR1 NUMERIC RESPONSE DATA> ファイルサイズバイト
例	MMEM:INFO? "Internal/report.pdf" → "2015-05-29 16:02:20",9137
注	

2.6.9 MMEemory:CATalog?

文法	MMEemory:CATalog? <directory>[,<pattern>]
説明	ディレクトリ内のファイルリストを問い合わせます。
Parameters	<directory> = <STRING PROGRAM DATA> ディレクトリのパス <pattern> = <STRING PROGRAM DATA> オプションで*と?が使えます。
レスポンス	({<item>} + {,}*) = <EXPRESSION RESPONSE DATA> リストはダブルコーテーションでくくられて出力されます。
例	MMEM:CAT? "Internal/reports" → ("report.pdf","setup.cfg")
注	

2.6.10 MMEemory:DCATalog?

文法	MMEemory:DCATalog? <directory>
説明	ディレクトリ内のサブディレクトリを問い合わせます。
パラメータ	<directory> = <STRING PROGRAM DATA> ディレクトリのパス
レスポンス	({<directory>} + {,}*) = <EXPRESSION RESPONSE DATA> リストはダブルコーテーションでくくられて出力されます。
例	MMEM:DCAT? "Internal/" → ("diagnostics","favorites","logs","remote","screens","windowsinstaller")
注	

2.6.11 MMEMory:MDIRectory

文法	MMEMory:MDIRectory <directory>
説明	サブディレクトリを作成します。
パラメータ	<directory> = <STRING PROGRAM DATA> 作成するディレクトリのパス
レスポンス	無し
例	MMEM:MDIR "Internal/reports"
注	

2.6.12 MMEMory:RDIRectory

文法	MMEMory:RDIRectory <directory>[,<force>]
説明	ディレクトリを削除します。
パラメータ	<directory> = <STRING PROGRAM DATA> ディレクトリを指定します。 <force> = <BOOLEAN PROGRAM DATA> ON を設定するとディレクトリ内すべてが削除されます。
レスポンス	無し
例	MMEM:RDIR "Internal/reports"
注	無し

2.6.13 MMEMory:SAVE

文法	MMEMory:SAVE <enable>
説明	アプリケーションが終了するときに、現在設定の自動保存を行うかどうかを設定します。
パラメータ	<enable> = <BOOLEAN PROGRAM DATA> <i>DEFault = OFF</i>
レスポンス	無し
例	INST:STAR:LAST OTDR-OTDR,1-PORT1 MMEMory:SAVE ON INST? → 2 INST:TERM 2
注	

文法	MMEMory:SAVE?
説明	アプリケーションが終了するときに、現在設定の自動保存を行うかどうかを読み出します。
パラメータ	無し
レスポンス	<enable> = <NR1 NUMERIC RESPONSE DATA>
例	MMEMory:SAVE? → 1
注	

Chapter 3

Standard OTDRアプリケーションコマンド

Standard OTDRアプリケーションで使用可能なコマンドについて説明します。FTTA, およびConstructionアプリケーションはリモートコマンド制御に対応していません。

3.1 測定条件

3.1.1 OTDR:SOURce:PORT

文法	OTDR:SOURce:PORT <port>
説明	OTDR測定ポートを切り替えます。
パラメータ	<port>=<CHARACTER RESPONSE DATA> SM: シングルモード MM: マルチモード
レスポンス	無し
例	otdr:sour:port SM
注	“MM” の設定は MU100021Aのみ有効です。 MU100023Aでは、“SM”を指定してください。66 ページの3.1.4 節のコマンドによって波長を1310 nmあるいは1550 nmに設定するとPORT1, 1650 nmに設定するとPORT2が測定ポートとして選択されます。

文法	OTDR:SOURce:PORT?
説明	現在のOTDR出力ポート設定を問い合わせます。
パラメータ	無し
レスポンス	<port>=<CHARACTER RESPONSE DATA>
例	otdr:sour:port? → SM
注	

3.1.2 OTDR:SOURce:TEST

文法	OTDR:SOURce:TEST <mode>
説明	測定モードを切り替えます。
パラメータ	<mode>=<CHARACTER PROGRAM DATA> AUTO: 自動設定 MANUAL: 個別設定
レスポンス	無し
例	otdr:sour:test AUTO
注	

文法	OTDR:SOURce:TEST?
説明	現在の測定モードを問い合わせます。
パラメータ	無し
レスポンス	<mode> = <CHARACTER RESPONSE DATA>
例	otdr:sour:test? → AUTO
注	

3.1.3 OTDR:SOURce:WAVelength:AVailable?

文法	OTDR:SOURce:WAVelength:AVailable?
説明	OTDRモジュールの使用可能な波長を問い合わせます。
パラメータ	無し
レスポンス	<wavelength> = <NUMERIC RESPONSE DATA>
例	otdr:sour:wav:ava? → 1310, 1550
注	

3.1.4 OTDR:SOURce:WAVelength

文法	OTDR:SOURce:WAVelength <wavelength>
説明	測定に使用する波長 (nm) を設定します。 モジュールで使用可能な波長のみが設定できます。
パラメータ	<wavelength> = <NUMERIC PROGRAM DATA>
レスポンス	無し
例	otdr:sour:wav 1310
注	“ALL” の設定はできません。

文法	OTDR:SOURce:WAVelength?
説明	現在設定されている波長を問い合わせます。
パラメータ	無し
レスポンス	<wavelength> = <NUMERIC RESPONSE DATA>
例	otdr:sour:wav? → 1310
注	“ALL” が設定されているときは第1波長が返ります。

3.1.5 OTDR:SOURce:RANge:AVailable?

文法	OTDR:SOURce:RANge:AVailable?
説明	現在の波長で使用可能な距離レンジ (km) を問い合わせます。
パラメータ	無し
レスポンス	<range> = <NUMERIC RESPONSE DATA>
例	otdr:sour:ran:ava? → 5.0, 10.0, 20.0, 50.0, 100.0, 200.0, 300.0
注	

3.1.6 OTDR:SOURce:RANge

文法	OTDR:SOURce:RANge <range>
説明	測定するための距離レンジ (km) を設定します。 現在の波長で設定可能な距離レンジが設定できます。
パラメータ	<range> = <NUMERIC PROGRAM DATA>
レスポンス	無し
例	otdr:sour:ran 100
注	

文法	OTDR:SOURce:RANge?
説明	現在設定されている距離レンジ (km) を問い合わせます。
パラメータ	無し
レスポンス	<range> = <NUMERIC RESPONSE DATA>
例	otdr:sour:ran? → 50.0
注	

3.1.7 OTDR:SOURce:RESO:AVailable?

文法	OTDR:SOURce:RESO:AVailable?
説明	使用可能なサンプリング分解能を問い合わせます。
パラメータ	無し
レスポンス	<res> = <CHARACTER RESPONSE DATA>
例	otdr:sour:res:ava? → COARSE, MEDIUM, FINE
注	

3.1.8 OTDR:SOURce:RESO

文法	OTDR:SOURce:RESO <res>
説明	測定するための分解能を設定します。
パラメータ	<res> = <CHARACTER PROGRAM DATA>
レスポンス	無し
例	otdr:sour:res MEDIUM
注	

文法	OTDR:SOURce:RESO?
説明	現在設定されている分解能を問い合わせます。
パラメータ	無し
レスポンス	<res> = <CHARACTER RESPONSE DATA>
例	otdr:sour:res? → MEDIUM
注	

3.1.9 OTDR:SOURce:PULSe:AVailable?

文法	OTDR:SOURce:PULSe:AVailable?
説明	使用可能なパルス幅 (ns) を問い合わせます。
パラメータ	無し
レスポンス	<width> = <NUMERIC RESPONSE DATA>
例	otdr:sour:puls:ava? → 10, 20, 50, 100
注	

3.1.10 OTDR:SOURce:PULSe

文法	OTDR:SOURce:PULSe <width>
説明	測定で使用するパルス幅 (ns) を設定します。
パラメータ	<width> = <NUMERIC PROGRAM DATA>
レスポンス	無し
例	otdr:sour:puls 100
注	設定できるパルス幅は距離レンジと分解能に依存します。

文法	OTDR:SOURce:PULSe?
説明	現在設定されているパルス幅（ns）を問い合わせます。
パラメータ	無し
レスポンス	<width> = <NUMERIC RESPONSE DATA>
例	otdr:sour:puls? → 100
注	

3.1.11 OTDR:SOURce:AVERages:TIME

文法	OTDR:SOURce:AVERages:TIME <time>
説明	マニュアル測定のアベレージ時間（秒）を設定します。
パラメータ	<time> = <NUMERIC PROGRAM DATA>
レスポンス	無し
例	otdr:sour:aver:tim 120
注	測定実行中は設定できません。

文法	OTDR:SOURce:AVERages:TIME?
説明	現在設定されているアベレージ時間を問い合わせます。
パラメータ	無し
レスポンス	<time> = <NUMERIC RESPONSE DATA>
例	otdr:sour:aver:tim? → 120
注	

3.2 IOR/BSC

3.2.1 OTDR:SENSe:FIBer:IOR

文法	OTDR:SENSe:FIBer:IOR <ior>
説明	ファイバの群屈折率（IOR）を設定します。
パラメータ	<ior> = <NUMERIC PROGRAM DATA> 設定範囲: 1.300000～1.700000
レスポンス	無し
例	otdr:sens:fib:ior 1.45
注	設定は次の測定結果に対して反映されます。

文法	OTDR:SENSe:FIBer:IOR?
説明	設定された群屈折率（IOR）を問い合わせます。
パラメータ	無し
レスポンス	<ior> = <NUMERIC RESPONSE DATA>
例	otdr:sens:fib:ior? → 1.450000
注	

3.2.2 OTDR:SENSe:FIBer:BSC

文法	OTDR:SENSe:FIBer:BSC <bsc>
説明	ファイバの後方散乱光係数（BSC）を設定します。
パラメータ	<bsc> = <NUMERIC PROGRAM DATA> 設定範囲: -90.00～-40.00
レスポンス	無し
例	otdr:sens:fib:bsc -83.0
注	設定は次の測定結果に対して反映されます。

文法	OTDR:SENSe:FIBer:BSC?
説明	設定された後方散乱光係数（BSC）を問い合わせます。
パラメータ	無し
レスポンス	<bsc> = <NUMERIC RESPONSE DATA>
例	otdr:sens:fib:bsc? → -83.0
注	

3.3 スプリッタ

3.3.1 OTDR:SOURce:SPLitter

文法	OTDR:SOURce:SPLitter <number>[,<splitter1>[,<splitter2>[,<splitter3>]]]
説明	測定するファイバのスプリッタ数およびスプリッタ分岐数を設定します。
パラメータ	<number> = <CHARACTER PROGRAM DATA> NONE: スプリッタなし 1~3: 指定されたスプリッタ数を設定。 DETECT: スプリッタ数を自動検出します。 <splitter1>, <splitter2>, <splitter3>= <CHARACTER PROGRAM DATA> X2: 1×2 X4: 1×4 X8: 1×8 X16: 1×16 X32: 1×32 X64: 1×64 X128: 1×128 AUTO: 1×?? (分岐数を自動検出します。)
レスポンス	無し
例	otdr:sour:spl NONE otdr:sour:spl 1,X2 otdr:sour:spl 2,X4,X8 otdr:sour:spl 3,X4,X8,X16 otdr:sour:spl DETECT
注	

文法	OTDR:SOURce:SPLitter?
説明	現在設定されているスプリッタ数およびスプリッタ分岐数を問い合わせます。
パラメータ	無し
レスポンス	<number>[,<splitter1>[,<splitter2>[,<splitter3>]]] <number> = <CHARACTER RESPONSE DATA> <splitter1>, <splitter2>, <splitter3>= <CHARACTER RESPONSE DATA>
例	otdr:sour:spl? → NONE otdr:sour:spl? → 1,X2 otdr:sour:spl? → 2,X4,X8 otdr:sour:spl? → 3,X4,X8,X16 otdr:sour:spl? → DETECT
注	

3.4 ステータス

3.4.1 OTDR:SENSe:AVERages:TIME?

文法	OTDR:SENSe:AVERages:TIME?
説明	測定を開始してからの経過時間を問い合わせます。
パラメータ	無し
レスポンス	<time> = <NUMERIC RESPONSE DATA>
例	otdr:sens:aver:tim? → 28
注	

3.4.2 OTDR:SENSe:TRACe:READY?

文法	OTDR:SENSe:TRACe:READY?
説明	トレースデータが存在しているか問い合わせます。
パラメータ	無し
レスポンス	<status> = <BOOLEAN RESPONSE DATA> 1 = トレースデータが存在する。 0 = トレースデータが存在しない。
例	meas:star syst:wait:idle otdr:sens:trac:ready? → 1
注	測定完了を待つにはSYSTem:WAIT:IDLEコマンドを使用してください。

3.5 測定機能

3.5.1 OTDR:SENSe:CONCheck

文法	OTDR:SENSe:CONCheck <value>
説明	接続チェックの実行有無を設定します。
パラメータ	<value> = <BOOLEAN PROGRAM DATA> 1 = 接続チェックを実行する。 0 = 接続チェックを実行しない。
レスポンス	無し
例	otdr:sens:conc 1
注	

文法	OTDR:SENSe:CONCheck?
説明	接続チェックの実行有無を問い合わせます。
パラメータ	無し
レスポンス	<value> = <BOOLEAN RESPONSE DATA> 1 = 接続チェックを実行する。 0 = 接続チェックを実行しない。
例	otdr:sens:conc? → 1
注	

3.5.2 OTDR:SENSe:CONState?

文法	OTDR:SENSe:CONState?
説明	接続チェックの結果を問い合わせます。
パラメータ	無し
レスポンス	<res> = <CHARACTER RESPONSE DATA> NONE: 接続チェックが未実行。 Poor: 接続状態が悪い。 Fair: 接続状態が良い。 Good: 接続状態が良好。
例	otdr:sens:cons? → Good
注	

3.5.3 OTDR:CONTinue

文法	OTDR:CONTinue
説明	接続チェック後に測定を継続します。接続チェックの状態がFairまたはGoodのときに継続が可能です。
パラメータ	無し
レスポンス	無し
例	otdr:cont
注	

3.5.4 OTDR:SENSe:LIVCheck

文法	OTDR:SENSe:LIVCheck <value>
説明	通信光チェックの実行有無を設定します。
パラメータ	<value> = <BOOLEAN PROGRAM DATA> 1 = 通信光チェックを実行する。 0 = 通信光チェックを実行しない。
レスポンス	無し
例	otdr:sens:livc 1
注	

文法	OTDR:SENSe:LIVCheck?
説明	通信光チェックの実行有無を問い合わせます。
パラメータ	無し
レスポンス	<value> = <BOOLEAN RESPONSE DATA> 1 = 通信光チェックを実行する。 0 = 通信光チェックを実行しない。
例	otdr:sens:livc? → 1
注	

3.5.5 OTDR:SENSe:FIBCheck

文法	OTDR:SENSe:FIBCheck <value>
説明	ファイバ長チェックの実行有無を設定します。
パラメータ	<value> = <BOOLEAN PROGRAM DATA> 1 = ファイバ長チェックを実行する。 0 = ファイバ長チェックを実行しない。
レスポンス	無し
例	otdr:sens:fibc 1
注	

文法	OTDR:SENSe:FIBCheck?
説明	ファイバ長チェックの実行有無を問い合わせます。
パラメータ	無し
レスポンス	<value> = <BOOLEAN RESPONSE DATA> 1 = ファイバ長チェックを実行する。 0 = ファイバ長チェックを実行しない。
例	otdr:sens:fibc? → 1
注	

3.6 解析

3.6.1 OTDR:SENSe:PATCH:LAUnch

文法	OTDR:SENSe:PATCH:LAUnch <value>
説明	パッチコードの開始点を設定します。
パラメータ	<value> = <CHARACTER PROGRAM DATA> NONE = 開始点を設定しません。 EVENT1 = イベント1の位置を開始点に設定します。 EVENT2 = イベント2の位置を開始点に設定します。 EVENT3 = イベント3の位置を開始点に設定します。 <value> = <NR2 NUMERIC PROGRAM DATA> 開始点を距離で設定します。 設定範囲: 距離レンジの範囲内
レスポンス	無し
例	OTDR:SENSe:PATCH:LAUnch 10.0
注	

文法	OTDR:SENSe:PATCH:LAUnch?
説明	パッチコードの開始点の設定を問い合わせます。
パラメータ	無し
レスポンス	<value> = <CHARACTER RESPONSE DATA> <value> = <NR2 NUMERIC PROGRAM DATA>
例	OTDR:SENSe:PATCH:LAUnch? → EVENT2
注	

3.6.2 OTDR:SENSe:PATCH:LAUnch:CORRelation

文法	OTDR:SENSe:PATCH:LAUnch:CORRelation <value>
説明	パッチコードの開始点用相関係数を設定します。
パラメータ	<value> = <NUMERIC PROGRAM DATA>
レスポンス	無し
例	OTDR:SENSe:PATCH:LAUnch:CORRelation 10.0
注	パッチコードの開始点に距離を設定しているときのみ有効です。

文法	OTDR:SENSe:PATCH:LAUnch:CORRelation?
説明	パッチコードの開始点用相関係数を問い合わせます。
パラメータ	無し
レスポンス	<value> = <NUMERIC RESPONSE DATA>
例	OTDR:SENSe:PATCH:LAUnch:CORRelation? → 2.000
注	パッチコードの開始点に距離を設定しているときのみ有効です。

3.6.3 OTDR:SENSe:PATCH:RECeive

文法	OTDR:SENSe:PATCH:RECeive <value>
説明	遠端側パッチコードの終了点を設定します。
パラメータ	<value> = <CHARACTER PROGRAM DATA> NONE = 終了点を設定しません。 EVENT1 = イベント1の位置を終了点に設定します。 EVENT2 = イベント2の位置を終了点に設定します。 EVENT3 = イベント3の位置を終了点に設定します。 <value> = <NR2 NUMERIC PROGRAM DATA> 終了点を距離で設定します。 設定範囲: 距離レンジの範囲内
レスポンス	無し
例	OTDR:SENSe:PATCH:RECeive 10.0
注	

文法	OTDR:SENSe:PATCH:RECeive?
説明	遠端側パッチコードの終了点を問い合わせます。
パラメータ	無し
レスポンス	<value> = <CHARACTER RESPONSE DATA> <value> = <NR2 NUMERIC PROGRAM DATA>
例	OTDR:SENSe:PATCH:RECeive? → EVENT2
注	

3.6.4 OTDR:SENSe:PATCH:RECeive:CORRelation

文法	OTDR:SENSe:PATCH:RECeive:CORRelation <value>
説明	遠端側パッチコードの終了点用相関係数を設定します。
パラメータ	<value> = <NUMERIC PROGRAM DATA> 終了点の相関係数を設定します。
レスポンス	無し
例	OTDR:SENSe:PATCH:RECeive:CORRelation 10.0
注	パッチコードの終了点に距離を設定しているときのみ有効です。

文法	OTDR:SENSe:PATCH:RECeive:CORRelation?
説明	遠端側パッチコードの終了点用相関係数を問い合わせます。
パラメータ	無し
レスポンス	<value> = <NUMERIC RESPONSE DATA>
例	OTDR:SENSe:PATCH:RECeive:CORRelation? → 2.0
注	パッチコードの終了点に距離を設定しているときのみ有効です。

3.6.5 OTDR:SENSe:ACURsor

文法	OTDR:SENSe:ACURsor <value>
説明	Aカーソルの位置 (km) を設定します。
パラメータ	<value> = <NUMERIC PROGRAM DATA> 設定範囲: 設定された距離レンジの範囲
レスポンス	無し
例	otdr:sens:acur 20.5
注	

文法	OTDR:SENSe:ACURsor?
説明	Aカーソルの位置 (km) を問い合わせます。
パラメータ	無し
レスポンス	<value> = <NUMERIC RESPONSE DATA>
例	otdr:sens:acur? → 20.5
注	

3.6.6 OTDR:SENSe:BCURsor

文法	OTDR:SENSe:BCURsor <value>
説明	Bカーソルの位置 (km) を設定します。
パラメータ	<value> = <NUMERIC PROGRAM DATA> 設定範囲: 設定された距離レンジの範囲
レスポンス	無し
例	otdr:sens:bcursor 20.5
注	

文法	OTDR:SENSe:BCURsor?
説明	Bカーソルの位置 (km) を問い合わせます。
パラメータ	無し
レスポンス	<value> = <NUMERIC RESPONSE DATA>
例	otdr:sens:bcursor? → 20.5
注	

3.6.7 OTDR:SENSe:LSALeft

文法	OTDR:SENSe:LSALeft <start>,<stop>
説明	左側LSAマーカの開始, 終了位置 (km) を設定します。
パラメータ	<start> = <NUMERIC PROGRAM DATA> <stop> = <NUMERIC PROGRAM DATA> 設定範囲: -100.0～400.0
レスポンス	無し
例	otdr:sens:lsal 0.0, 0.5
注	開始位置は終了位置より小さい値にしてください。 距離レンジを超える値が設定されたときは,距離レンジの最大値になります。

文法	OTDR:SENSe:LSALeft?
説明	左側LSAマーカの開始位置, 終了位置 (km) を問い合わせます。
パラメータ	無し
レスポンス	<start>,<stop> <start> = <NUMERIC RESPONSE DATA> <stop> = <NUMERIC RESPONSE DATA>
例	otdr:sens:lsal? → 0.0, 0.5
注	

3.6.8 OTDR:SENSe:LSARight

文法	OTDR:SENSe:LSARight <start>,<stop>
説明	右側LSAマーカの開始, 終了位置 (km) を設定します。
パラメータ	<start> = <NUMERIC PROGRAM DATA> 設定範囲: -100.0～400.0 <stop> = <NUMERIC PROGRAM DATA> 設定範囲: -100.0～400.0
レスポンス	無し
例	otdr:sens:lsar 0.0, 0.5
注	開始位置は終了位置より小さい値にしてください。 距離レンジを超える値が設定されたときは,距離レンジの最大値になります。

文法	OTDR:SENSe:LSARight?
説明	右側LSAマーカの開始位置, 終了位置 (km) を問い合わせます。
パラメータ	無し
レスポンス	<start>,<stop> <start> = <NUMERIC RESPONSE DATA> <stop> = <NUMERIC RESPONSE DATA>
例	otdr:sens:lsar? → 0.0, 0.5
注	

3.6.9 OTDR:SENSe:LOSS:MODE

文法	OTDR:SENSe:LOSS:MODE <mode>
説明	損失測定を設定します。
パラメータ	<mode> = <CHARACTER PROGRAM DATA> SPLICE: 接続損失 (LSA) TP: 2点間損失 TPLSA: 2点間LSA DBKM: 伝送損失2PA DBKMLSA: dB/km LSA TPDBKM: 2点間損失, dB/km ORL: 全反射減衰量
レスポンス	無し
例	otdr:sens:loss:mode SPLICE
注	

文法	OTDR:SENSe:LOSS:MODE?
説明	現在の損失測定設定を問い合わせます。
パラメータ	無し
レスポンス	<mode> = <CHARACTER RESPONSE DATA>
例	otdr:sens:loss:mode? → SPLICE
注	

3.6.10 OTDR:SENSe:ORL:MODE

文法	OTDR:SENSe:ORL:MODE <mode>
説明	全反射減衰量の計算方法を設定します。
パラメータ	<mode> = <CHARACTER PROGRAM DATA> ACursor: カーソルA ORIGIN: 口元位置 FULL: 波形全体
レスポンス	無し
例	otdr:sens:orl:mode ACursor
注	

文法	OTDR:SENSe:ORL:MODE?
説明	現在の全反射減衰量の計算方法を問い合わせます。
パラメータ	無し
レスポンス	<mode> = <CHARACTER RESPONSE DATA>
例	otdr:sens:orl:mode? → ACursor
注	

3.6.11 OTDR:SENSe:ANALyze:PARAmeters

文法	OTDR:SENSe:ANALyze:PARAmeters <splice loss>,<reflectance>,<end loss>,<macro bend>,<splitter 1x2>,<splitter 1x4>,<splitter 1x8>,<splitter 1x16>,<splitter 1x32>,<splitter 1x64>,<splitter 1x128>
説明	自動検出のしきい値を設定します。
パラメータ	<splice loss> = <NUMERIC PROGRAM DATA> 設定範囲: 0.01～9.99 <reflectance> = <NUMERIC PROGRAM DATA> 設定範囲: -70.0～-20.0 <end loss> = <NUMERIC PROGRAM DATA> 設定範囲: 1～99 <Macro Bend> = <NUMERIC PROGRAM DATA> 設定範囲: 0.3～2.0 <Splitter Loss(1x2)> = <NUMERIC PROGRAM DATA> 設定範囲: 1.0～30.0 <Splitter Loss(1x4)> = <NUMERIC PROGRAM DATA> 設定範囲: 1.0～30.0 <Splitter Loss(1x8)> = <NUMERIC PROGRAM DATA> 設定範囲: 1.0～30.0 <Splitter Loss(1x16)> = <NUMERIC PROGRAM DATA> 設定範囲: 1.0～30.0 <Splitter Loss(1x32)> = <NUMERIC PROGRAM DATA> 設定範囲: 1.0～30.0 <Splitter Loss(1x64)> = <NUMERIC PROGRAM DATA> 設定範囲: 1.0～30.0 <Splitter Loss(1x128)> = <NUMERIC PROGRAM DATA> 設定範囲: 1.0～30.0
レスポンス	無し
例	otdr:sens:anal:par 0.05,-60.0,3,0.3,4.1,7.0,10.0,13.0,16.0,19.0,22.0
注	しきい値が変更されると自動的に解析を実行します。

文法	OTDR:SENSe:ANALyze:PARAmeters?
説明	現在設定されている自動検出しきい値を問い合わせます。
パラメータ	無し
レスポンス	<splice loss> = <NUMERIC RESPONSE DATA> 設定範囲: 0.01～9.99 <reflectance> = <NUMERIC RESPONSE DATA> 設定範囲: -70.0～-20.0 <end loss> = <NUMERIC RESPONSE DATA> 設定範囲: 1～99 <Macro Bend> = <NUMERIC RESPONSE DATA> 設定範囲: 0.3～2.0 <Splitter Loss(1x2)> = <NUMERIC RESPONSE DATA> 設定範囲: 1.0～30.0 <Splitter Loss(1x4)> = <NUMERIC RESPONSE DATA> 設定範囲: 1.0～30.0 <Splitter Loss(1x8)> = <NUMERIC RESPONSE DATA> 設定範囲: 1.0～30.0 <Splitter Loss(1x16)> = <NUMERIC RESPONSE DATA> 設定範囲: 1.0～30.0 <Splitter Loss(1x32)> = <NUMERIC RESPONSE DATA> 設定範囲: 1.0～30.0 <Splitter Loss(1x64)> = <NUMERIC RESPONSE DATA> 設定範囲: 1.0～30.0 <Splitter Loss(1x128)> = <NUMERIC RESPONSE DATA> 設定範囲: 1.0～30.0
例	otdr:sens:anal:par? → 0.05,-60.0,3,0.3,4.1,7.0,10.0,13.0,16.0,19.0,22.0
注	

3.7 測定波形

3.7.1 OTDR:TRACe:PARameters?

文法	OTDR:TRACe:PARameters?
説明	波形データ取得時の主なOTDRパラメータを問い合わせます。
パラメータ	無し
レスポンス	<wave> = <NUMERIC RESPONSE DATA> <range> = <NUMERIC RESPONSE DATA> <avg> = <NUMERIC RESPONSE DATA> <reso> = <NUMERIC RESPONSE DATA> <ior> = <NUMERIC RESPONSE DATA> <bsc> = <NUMERIC RESPONSE DATA>
例	otdr:trac:par? → 1310, 16.415554, 50, 6144, 0.656621, 1.467700, -78.500000
注	

3.7.2 OTDR:TRACe:ANALyze

文法	OTDR:TRACe:ANALyze
説明	解析を実行します。
パラメータ	無し
レスポンス	無し
例	otdr:trac:anal
注	測定中は解析を実行できません。

3.7.3 OTDR:TRACe:ANALyze:ORL

文法	OTDR:TRACe:ANALyze:ORL
説明	全反射減衰量の計算を実行します。
パラメータ	無し
レスポンス	無し
例	otdr:trac:anal:orl
注	測定中は全反射減衰量の計算を実行できません。

3.7.4 OTDR:TRACe:MDLOss?

文法	OTDR:TRACe:MDLOss?
説明	現在の損失測定設定に応じた損失値を問い合わせます。
パラメータ	無し
レスポンス	<loss> = <NUMERIC RESPONSE DATA> <dBloss> = <NUMERIC RESPONSE DATA> 損失測定設定が2点間損失,dB/km以外の時は-99.99になります。
例	otdr:trac:mdlo? → -4.610,-99.99
注	測定中は実行できません。

3.7.5 OTDR:TRACe:EELOss?

文法	OTDR:TRACe:EELOss?
説明	ファイバの全損失を問い合わせます。
パラメータ	無し
レスポンス	<loss> = <NUMERIC RESPONSE DATA>
例	otdr:trac:eeelo? → -4.610
注	測定中は実行できません。

3.7.6 OTDR:TRACe:LOAD:TEXT?

文法	OTDR:TRACe:LOAD:TEXT? [<start>[,<end>]]
説明	テキスト形式で波形データを取得します。
パラメータ	<start> = <NR2 NUMERIC PROGRAM DATA> 設定範囲: 現在設定されている距離レンジ (km) <end> = <NR2 NUMERIC PROGRAM DATA> 設定範囲: 現在設定されている距離レンジ (km)
レスポンス	(<parameters>+,*) = <EXPRESSION RESPONSE DATA>
例	<pre> otdr:trac:load:text? → #6140479 //転送データサイズ WL = 1310 nm //測定波長 FBR = SM //ポート設定 DR = 5 km //距離レンジ PW = 50 ns //パルス幅 AVG = 60 //アベレージ時間 IOR = 1.467700 //IOR BSC = -78.50 //後方散乱光係数 DATE = 08/13/09 //測定日 TIME = 10:19 PM //測定時刻 MXDB = 64 dB //dB Range RESO = 0.200 m //測定分解能 DX = 0.20440100621721 m //サンプリング分解能 PTS = 25001 //サンプリングポイント数 <... 波形データ ...> Events 1 //検出されたイベント数 Dist 1.0505 km //イベント位置 Type E //イベントタイプ Loss >3.00 dB //損失値 Reflectance N/A //反射減衰量 dB / km 1.801 dB //dB/Km Cumulative Loss 5.94 dB //累積損失 </pre>
注	測定中および測定結果が準備できていない時は実行できません。

3.7.7 OTDR:TRACe:HOFFset

文法	OTDR:TRACe:HOFFset?
説明	表示中の波形の距離オフセット値(km)を問い合わせます。
パラメータ	無し
レスポンス	<value> = < NR2 NUMERIC RESPONSE DATA>
例	otdr:trac:hoff? → -0.234
注	

Chapter 4

OLTSアプリケーションコマンド

OLTSアプリケーションで使用可能なコマンドについて説明します。FTTA, およびConstructionアプリケーションはリモートコマンド制御に対応していません。

4.1 ロステストセット

4.1.1 OLTS:SOURce:POWer:WAVelength:AVailable?

文法	OLTS:SOURce:POWer:WAVelength:AVailable?
説明	OLTSアプリケーションの光源に設定可能な波長を問い合わせます。
パラメータ	無し
レスポンス	<wavelength> = <NUMERIC RESPONSE DATA>
例	olts:sour:pow:wav:ava? → 1310, 1550
注	

4.1.2 OLTS:SOURce:POWer:WAVelength

文法	OLTS:SOURce:POWer:WAVelength <wavelength>
説明	光源の波長 (nm) を設定します。 モジュールで使用可能な波長のみが設定できます。
パラメータ	<wavelength> = <NUMERIC PROGRAM DATA>
レスポンス	無し
例	olts:sour:pow:wav 1310
注	“ALL” の設定はできません。

文法	OLTS:SOURce:POWer:WAVelength?
説明	光源の波長 (nm) を問い合わせます。
パラメータ	無し
レスポンス	<wavelength> = <NUMERIC RESPONSE DATA>
例	olts:sour:pow:wav? → 1310
注	“ALL” が設定されているときは第1波長が返ります。

4.1.3 OLTS:SOURce:AM:FREQuency:AVailable?

文法	OLTS:SOURce:AM:FREQuency:AVailable?
説明	OLTSアプリケーションの光源の出力モードに設定可能な周波数を問い合わせます。
パラメータ	無し
レスポンス	<value> = <CHARACTER PROGRAM DATA>
例	olts:sour:am:freq:ava? → CW, 270hz, 1khz, 2khz
注	

4.1.4 OLTS:SOURce:AM:FREQuency

文法	OLTS:SOURce:AM:FREQuency <value>
説明	光源の出力モードを設定します。
パラメータ	<value> = <CHARACTER PROGRAM DATA>
レスポンス	無し
例	olts:sour:am:freq CW
注	

文法	OLTS:SOURce:AM:FREQuency?
説明	光源の出力モードを問い合わせます。
パラメータ	無し
レスポンス	<freq> = <CHARACTER PROGRAM DATA _i >
例	olts:sour:am:freq? → CW
注	

4.1.5 OLTS:SOURce:POWer:STATe

文法	OLTS:SOURce:POWer:STATe <status>
説明	光源の消灯または点灯を設定します。
パラメータ	<status> = <NUMERIC PROGRAM DATA> 1 = 点灯。 0 = 消灯。
レスポンス	無し
例	olts:sour:pow:stat 1
注	

文法	OLTS:SOURce:POWer:STATe?
説明	光源が点灯しているか問い合わせます。
パラメータ	無し
レスポンス	<status> = <NUMERIC RESPONSE DATA> 1 = 点灯。 0 = 消灯。
例	olts:sour:pow:stat? → 1
注	

4.1.6 OLTS:SENSe:POWer:WAVelength:AVailable?

文法	OLTS:SENSe:POWer:WAVelength:AVailable?
説明	OLTSアプリケーションのパワーメータに設定可能な波長 (nm) を問い合わせます。
パラメータ	無し
レスポンス	<wavelength> = <NUMERIC RESPONSE DATA>
例	olts:sens:pow:wav:ava? → 850, 1300, 1310, 1490, 1550, 1625, 1650
注	

4.1.7 OLTS:SENSe:POWer:WAVelength

文法	OLTS:SENSe:POWer:WAVelength <wavelength>
説明	パワーメータの波長 (nm) を設定します。
パラメータ	<wavelength> = <NUMERIC PROGRAM DATA>
レスポンス	無し
例	olts:sens:pow:wav 1310
注	

文法	OLTS:SENSe:POWer:WAVelength?
説明	パワーメータの波長（nm）を問い合わせます。
パラメータ	無し
レスポンス	<wavelength> = <NUMERIC RESPONSE DATA>
例	olts:sens:pow:wav? → 1310
注	

4.1.8 OLTS:SENSe:CORRection:COLLect:ZERO

文法	OLTS:SENSe:CORRection:COLLect:ZERO
説明	パワーメータのゼロオフセットを実行します。
パラメータ	無し
レスポンス	無し
例	olts:sens:corr:coll:zero
注	

文法	OLTS:SENSe:CORRection:COLLect:ZERO?
説明	パワーメータのゼロオフセットの結果を問い合わせます。
パラメータ	無し
レスポンス	<status> = <CHARACTER RESPONSE DATA> Pass = 成功。 Fail = 失敗。
例	olts:sens:corr:coll:zero? → Pass
注	

4.1.9 OLTS:SENSe:AVERage:COUNt

文法	OLTS:SENSe:AVERage:COUNt <value>
説明	パワーメータの平均回数を設定します。
パラメータ	<value> = <NUMERIC PROGRAM DATA>
レスポンス	無し
例	olts:sens:aver:coun 1
注	

文法	OLTS:SENSe:AVERage:COUNt?
説明	パワーメータの平均回数を問い合わせます。
パラメータ	無し
レスポンス	<value> = <NUMERIC RESPONSE DATA>
例	olts:sens:aver:coun? → 1
注	

4.1.10 OLTS:SENSe:POWer:REFeRence

文法	OLTS:SENSe:POWer:REFeRence <reference>
説明	パワーメータの基準値（dBm）を設定します。
パラメータ	<reference> = <NUMERIC PROGRAM DATA>
レスポンス	無し
例	olts:sens:pow:ref -10.5
注	

文法	OLTS:SENSe:POWer:REference?
説明	パワーメータの基準値（dBm）を問い合わせます。
パラメータ	無し
レスポンス	<reference> = <NUMERIC RESPONSE DATA>
例	olts:sens:pow:ref? → -10.50
注	

4.1.11 OLTS:SENSe:POWer:THReshold

文法	OLTS:SENSe:POWer:THReshold <threshold>
説明	パワーメータの良否判定のしきい値（dB）を設定します。
パラメータ	<threshold> = <NUMERIC PROGRAM DATA>
レスポンス	無し
例	olts:sens:pow:thr -1.5
注	

文法	OLTS:SENSe:POWer:THReshold?
説明	パワーメータの良否判定のしきい値（dB）を問い合わせます。
パラメータ	無し
レスポンス	<threshold> = <NUMERIC RESPONSE DATA>
例	olts:sens:pow:thr? → -1.50
注	

4.1.12 OLTS:FETCh:POWer

文法	OLTS:FETCh:POWer?
説明	パワーメータのパワー（dBm）を問い合わせます。
パラメータ	無し
レスポンス	<power> = <NUMERIC RESPONSE DATA>
例	olts:fetc:pow? → -20.00
注	

Chapter 5

測定

5.1 測定の開始と終了

5.1.1 MEASurement:APPLication?

文法	MEASurement:APPLication?
説明	アプリケーションの種類を問い合わせます。
パラメータ	無し
レスポンス	<application> = <CHARACTER RESPONSE DATA> OTDR-OTDR: Standard OTDRアプリケーション OTDR-OLTS: OLTSアプリケーション
例	MEAS:APPL? → OTDR-OTDR
注	

5.1.2 MEASurement:STARt

文法	MEASurement:STARt
説明	測定を開始します。GUIの[START] ボタンを押したときと同じ動作です。
パラメータ	無し
レスポンス	無し
例	MEAS:STAR
注	

5.1.3 MEASurement:STOP

文法	MEASurement:STOP
説明	実行中の測定を停止します。GUIの[STOP] ボタンを押したときと同じ動作です。
パラメータ	無し
レスポンス	無し
例	MEAS:STOP
注	

5.1.4 MEASurement:REStult:SUMMARY?

文法	MEASurement:REStult:SUMMARY?
説明	測定結果の合否判定を問い合わせます。
パラメータ	無し
レスポンス	<summary> = <CHARACTER RESPONSE DATA> PASS = 合格 FAIL = 不合格
例	MEAS:RES:SUMM? → PASS
注	

Appendix A

サンプルスクリプト

この章ではOTDRをリモートコントロールする際のサンプルスクリプトを示します。

A.1 ヒント

既定の状態で実行開始できるようにするため、一般的にスクリプトの最初は次のコマンドで開始するのが良いとされています。このコマンドはすべてのアプリケーションを終了します。

```
*RST
```

A.2 OTDR測定

このサンプルではOTDR測定を実行します。SMポートに測定対象となるシングルモードファイバを接続してください。自動設定で測定を実行し測定結果を保存します。なお接続チェックが有効で、その結果がPoorの場合測定は行われず、ファイル保存がエラーになります。

```
*RST
INST:STAR OTDR-OTDR,1-PORT1
SYST:WAIT:IDLE

OTDR:SOUR:PORT SM
OTDR:SOUR:TES AUTO
OTDR:SOUR:WAV 1310

MEAS:STAR
SYST:WAIT:IDLE

OTDR:SENS:TRAC:READY?
MMEM:STOR:DATA "Usb/my-otdr-trace.sor"

SYST:ERR?
INST:TERM
```


A.3 HTTP GETリクエストを送信

このサンプルではHTTP GETリクエストを送信してレスポンスを受信します。

スクリプト	→	結果
<pre>SYST:NCN:COM:AUTH:TYPE BASIC SYST:NCN:COM:BAS "id","password" SYST:NCN:COM:SET "https://sample.net/config?argument=sample" SYST:NCN:COM:LRET:RST? SYST:WAIT:DUR 5 SYST:NCN:COM:LRET:RST? SYST:NCN:COM:LRET:ERR? SYST:NCN:COM:LRET:RRES?</pre>		<pre>0 1 0,"No error." {"key","Value"}</pre>

A.4 シナリオ実行結果をアップロードするパラメータを設定

このサンプルではシナリオ実行結果をアップロードするパラメータを設定します。

```
SYST:NCN:COM:AUTH:TYPE BASIC
SYST:NCN:COM:BAS "id","password"
SYST:NCN:COM:TUR "https://sample.net/upload"
```