

MX269036A
測定ソフトウェア for MediaFLO
取扱説明書
リモート制御編

第2版

- ・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、MS2690A/MS2691A/MS2692A シグナルアナライザ取扱説明書（本体 操作編）および MX269036A 測定ソフトウェア for MediaFLO 取扱説明書（操作編）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。




アンリツ株式会社

安全情報の表示について

当社では人身事故や財産の損害を避けるために、危険の程度に応じて下記のようなシグナルワードを用いて安全に関する情報を提供しています。記述内容を十分理解して機器を操作するようにしてください。

下記の表示およびシンボルは、そのすべてが本器に使用されているとは限りません。また、外観図などが本書に含まれるとき、製品に貼り付けたラベルなどがその図に記入されていない場合があります。

本書中の表示について

- | | | |
|---|-----------|--|
|  | 危険 | 回避しなければ、死亡または重傷に至る切迫した危険状況があることを警告しています。 |
|  | 警告 | 回避しなければ、死亡または重傷に至る恐れがある潜在的危険について警告しています。 |
|  | 注意 | 回避しなければ、軽度または中程度の人体の傷害に至る恐れがある潜在的危険、または、物的損害の発生のみが予測されるような危険状況について警告しています。 |

機器に表示または本書に使用されるシンボルについて

機器の内部や操作箇所付近に、または本書に、安全上または操作上の注意を喚起するための表示があります。

これらの表示に使用しているシンボルの意味についても十分理解して、注意に従ってください。



禁止行為を示します。丸の中や近くに禁止内容が描かれています。



守るべき義務的行為を示します。丸の中や近くに守るべき内容が描かれています。



警告や注意を喚起することを示します。三角の中や近くにその内容が描かれています。



注意すべきことを示します。四角の中にその内容が書かれています。



このマークを付けた部品がリサイクル可能であることを示しています。

MX269036A

測定ソフトウェア for MediaFLO

取扱説明書 リモート制御編

2009年（平成21年）10月14日（初 版）

2010年（平成22年）3月30日（第2版）

- ・予告なしに本書の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。

Copyright © 2009-2010, ANRITSU CORPORATION

Printed in Japan

国外持出しに関する注意

1. 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。
2. 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。

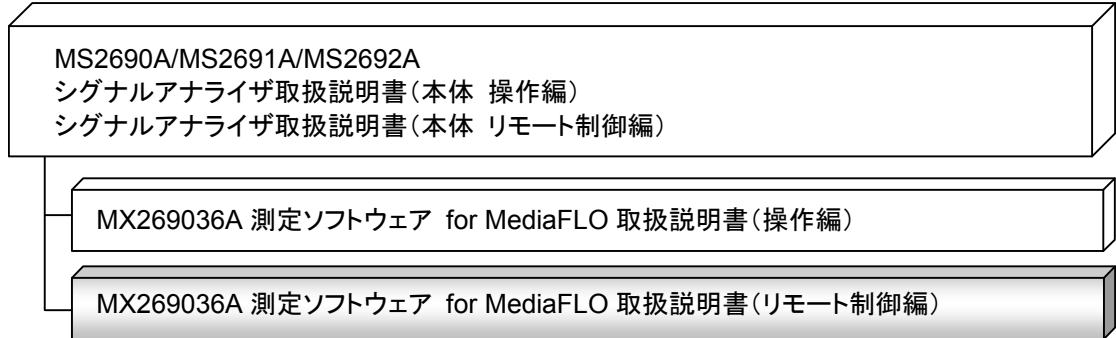
商標・登録商標

MediaFLO は Qualcomm 社の登録商標です。

はじめに

■取扱説明書の構成

MX269036A 測定ソフトウェア for MediaFLO の取扱説明書は、以下のように構成されています。



- シグナルアナライザ 取扱説明書(本体 操作編)
- シグナルアナライザ 取扱説明書(本体 リモート制御編)

本体の基本的な操作方法, 保守手順, 共通的な機能, 共通的なリモート制御などについて記述しています。

- MX269036A 測定ソフトウェア for MediaFLO 取扱説明書(操作編)

MX269036A 測定ソフトウェア for MediaFLO の操作方法について記述しています。

- MX269036A 測定ソフトウェア for MediaFLO 取扱説明書
(リモート制御編) <本書>

MX269036A 測定ソフトウェア for MediaFLO のリモート制御について記述しています。

目次

はじめに	I
第 1 章 概要	1-1
1.1 概要	1-2
1.2 基本的な制御の流れ	1-3
1.3 Native モードでの使用について	1-11
1.4 数値プログラムデータの設定について	1-14
第 2 章 SCPI デバイスメッセージ詳細	2-1
2.1 アプリケーションの選択	2-4
2.2 基本パラメータの設定	2-10
2.3 システムパラメータの設定	2-19
2.4 共通測定機能	2-40
2.5 Modulation 測定機能	2-48
2.6 測定結果の保存機能	2-55

第 3 章	SCPI ステータスレジスタ	3-1
3.1	測定状態の読み出し	3-2
3.2	STATus:QUESTIONable レジスタ.....	3-3
3.3	STATus:OPERation レジスタ.....	3-13

1
2
3

この章では、MX269036A 測定ソフトウェア for MediaFLO(以下、本アプリケーション)のリモート制御の概要について説明します。


1.1	概要	1-2
1.1.1	インタフェース	1-2
1.1.2	制御対象のアプリケーションについて	1-2
1.2	基本的な制御の流れ	1-3
1.2.1	初期設定	1-5
1.2.2	基本パラメータの設定	1-7
1.2.3	Modulation 共通の設定	1-8
1.2.4	Modulation 測定	1-10
1.3	Native モードでの使用について	1-11
1.4	数値プログラムデータの設定について	1-14

1.1 概要

本アプリケーションは、MS2690A/MS2691A/MS2692A シグナルアナライザ(以下、本器)を通じて、外部コントローラ(PC)からリモート制御コマンドによる制御を行うことができます。本アプリケーションのリモート制御コマンドは、SCPI 形式によって定義されています。

1.1.1 インタフェース

本器では、リモート制御用のインタフェースとして、GPIB, Ethernet, および USB に対応しています。同時に使用できるインタフェースはこのうちの 1 つです。

インタフェースは、本器が Local 状態のときに外部コントローラ(PC)から通信開始のコマンドを受信したものに自動的に決定されます。インタフェースが決定されると、本器は Remote 状態になります。正面パネルの  Remote が点灯している状態は Remote 状態を、消灯している状態は Local 状態を示します。

インタフェースの設定方法など、リモート制御の基本的な説明については、「MS2690A/MS2691A/MS2692A シグナルアナライザ取扱説明書(本体 リモート制御編)」を参照してください。

1.1.2 制御対象のアプリケーションについて

本器で使用できるリモート制御コマンドには、本器自体またはすべてのアプリケーションに対して適用されるコマンド(以下、共通コマンド)と、アプリケーション固有のコマンドの 2 種類があります。共通コマンドは、現在選択されているアプリケーションの種類によらず、実行できます。一方、アプリケーション固有のコマンドは、制御対象のアプリケーションに対してのみ有効で、制御対象でないアプリケーションが選択されている場合は、エラーになるか、制御対象のアプリケーションに対して実行されません。

本器では、複数のアプリケーションを同時に起動させることができます。このうち、同時に実行させることができるアプリケーションは、1 つのハードウェアリソースに対して 1 つのみです。本アプリケーションは、RF Input のリソースを使用して入力信号の測定を行います。したがって、本アプリケーションを、シグナルアナライザ機能など、同じリソースを使用するアプリケーションと同時に実行することはできません。本アプリケーション固有の機能をリモート制御で実行するときは、本アプリケーションが起動された状態で、本アプリケーションを選択するという操作をする必要があります。なお、MS2690A/MS2691A/MS2692A オプション 020 ベクトル信号発生器(以下、オプション 020)など、本アプリケーションが使用しないリソースを単独で利用するアプリケーションとは同時に実行することができます。

1.2 基本的な制御の流れ

この節では、本アプリケーションを使用した MediaFLO 信号の測定の基本的なリモート制御コマンドプログラミングの方法について説明します。

図 1.2-1 は、基本的な制御の流れを示しています。実行する測定機能の順序は入れ替えることができますが、測定に対して適用されるパラメータの設定と測定機能の種類、および測定実行の順番は入れ替えることができません。

概要

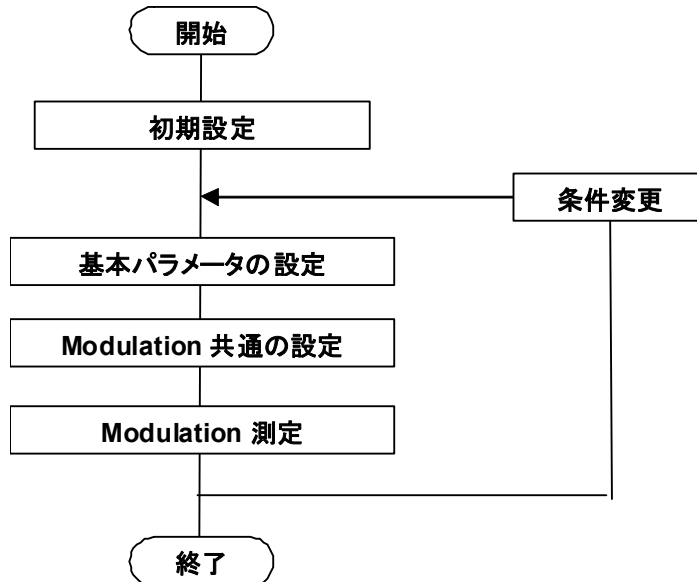



図 1.2-1 基本的な制御の流れ


(1) 初期設定

通信インタフェースの初期化、通信モードの設定、アプリケーションの起動と選択、およびパラメータの初期化などを行います。

 1.2.1 初期設定


(2) 基本パラメータの設定

キャリア周波数や入力レベルなど、すべての測定に共通に適用されるパラメータを設定します。

 1.2.2 基本パラメータの設定

(3) Modulation 共通の設定

本アプリケーションで実行する Modulation 測定機能に共通するパラメータを設定します。トリガ、変調方式、帯域の設定などが含まれます。

 1.2.3 Modulation 共通の設定

(4) Modulation 測定

本アプリケーションで実行する測定機能を順番に実行します。はじめに、測定機能を選択します。次に、測定機能ごとに、ストレージモードなどを設定し、測定の実行と測定結果の読み出しを行います。

 1.2.4 Modulation 測定

1.2.1 初期設定

測定器とそのアプリケーションを使用するための準備を行います。初期設定には、次の処理が含まれます。

- (1) 通信インタフェースの初期化
コマンドの送受信を開始するため、使用しているリモート制御インタフェースの初期化を行います。詳細は、お使いのインタフェースの取扱説明書を参照してください。
- (2) 言語モードとレスポンス形式の設定
通信に使用する言語モードとレスポンス形式を設定します。詳細は、『MS2690A/MS2691A/MS2692A シグナルアナライザ取扱説明書(本体リモート制御編)』を参照してください。
- (3) アプリケーションの起動
使用するアプリケーションを起動します。
- (4) アプリケーションの選択
使用するアプリケーションを選択します。
- (5) すべてのパラメータと状態の初期化
すべてのパラメータと状態を初期設定に戻します。
- (6) 測定モードの設定
初期化を行ったあとは連続測定になっているため、必要に応じてシングル測定に切り替えます。



図 1.2.1-1 初期設定の流れとコマンド例

1.2.2 基本パラメータの設定

キャリア周波数や入力レベルなど、すべての測定に共通するパラメータを設定します。基本パラメータには、次のものが含まれます。

- (1) Carrier Frequency
- (2) RF Spectrum
- (3) Input Level (Reference Level・Attenuator)
- (4) Level Offset
- (5) Pre-Amp (オプション)



図 1.2.2-1 基本パラメータの設定の流れとコマンド例

1.2.3 Modulation共通の設定

本アプリケーションで実行する Modulation 測定機能に共通するパラメータを設定します。特に明記がない限り、パラメータの設定順序に制限はありません。

- (1) Trigger
 - (a) Trigger Switch
 - (b) Trigger Source
 - (c) Trigger Slope
 - (d) Trigger Delay
- (2) Channel Bandwidth
- (3) Tracking Mode
- (4) Modulation
- (5) SPC Presence
- (6) FFT Size
- (7) Cyclic Prefix
- (8) Slot to Interlace
- (9) PPC Presence
- (10) PPC / RS Number
- (11) Wide-area Data Symbols
- (12) WID
- (13) LID
- (14) Channel Estimation

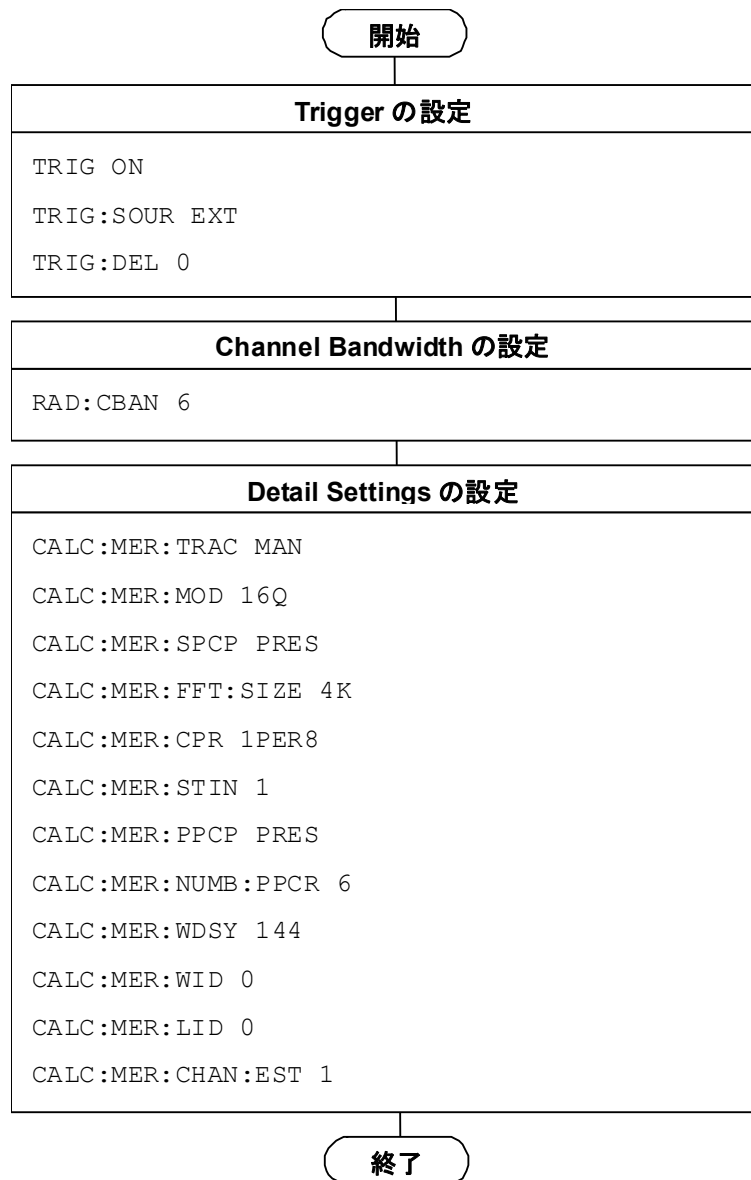


図 1.2.3-1 Modulation 共通の設定の流れとコマンド例

1.2.4 Modulation測定

以下の順に Modulation 測定を実行します。

- (1) 測定パラメータの設定
Modulation 測定に対してのみ適用されるパラメータです。
 - (a) End Symbol
 - (b) Storage
- (2) 測定の実行と測定結果の読み出し

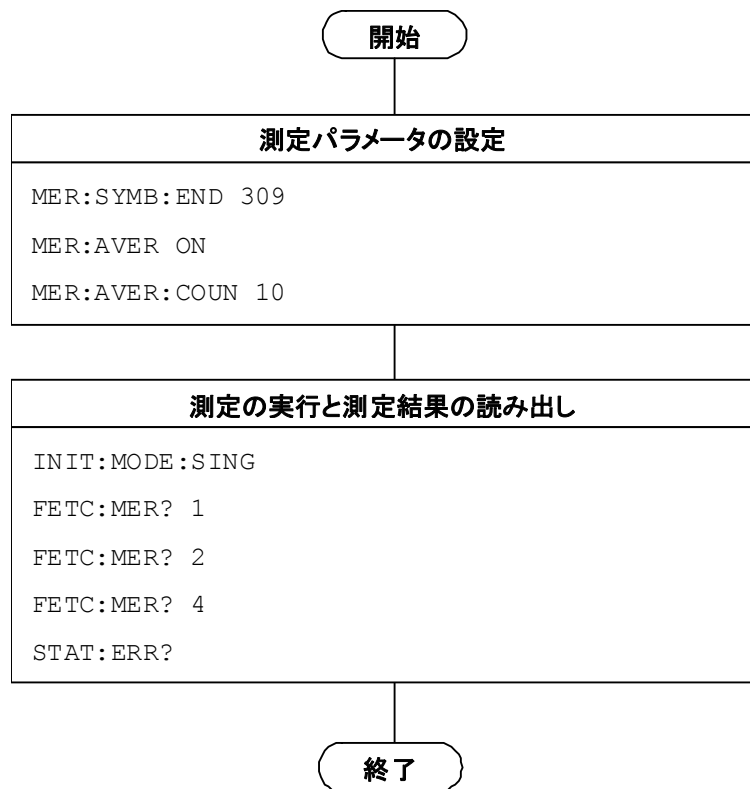


図 1.2.4-1 Modulation 測定の流れとコマンド例

1.3 Native モードでの使用について

本器では、リモート制御コマンドの文法・書式の種類を「言語モード」と定義しています。本器の言語モードには、SCPI モードと Native モードがあります。

(1) SCPI モード

SCPI (ver1999.0)で定義された文法・書式に準拠したコマンドを処理するモードです。プログラミング時にロングフォーム・ショートフォーム形式の文字列や角括弧 ([]) 定義文字列のスキップなどが利用できます。

コマンド `SYST:LANG SCPI` を送信すると、SCPI モードになります。

(2) Native モード

本器独自の定義形式によるコマンドを処理するモードです。特に明記がない限り、コマンドヘッダ部分は固定文字列です。アプリケーションのコマンドが SCPI モードでのみ定義されている場合、読み替えルールに従って変換した文字列が Native モードにおけるコマンドになります。SCPI モードの文法、つまり、プログラミング時にロングフォーム・ショートフォーム形式の文字列や角括弧 ([]) 定義文字列のスキップなどは使用できません。

注:

Native モードでは、`STATus:QUEStionable` レジスタおよび `STATus:OPERation` を使用することはできません。コマンドを読み替えルールに従って Native モードに変換した場合でも同様です。

コマンド `SYST:LANG NAT` を送信すると、Native モードになります。

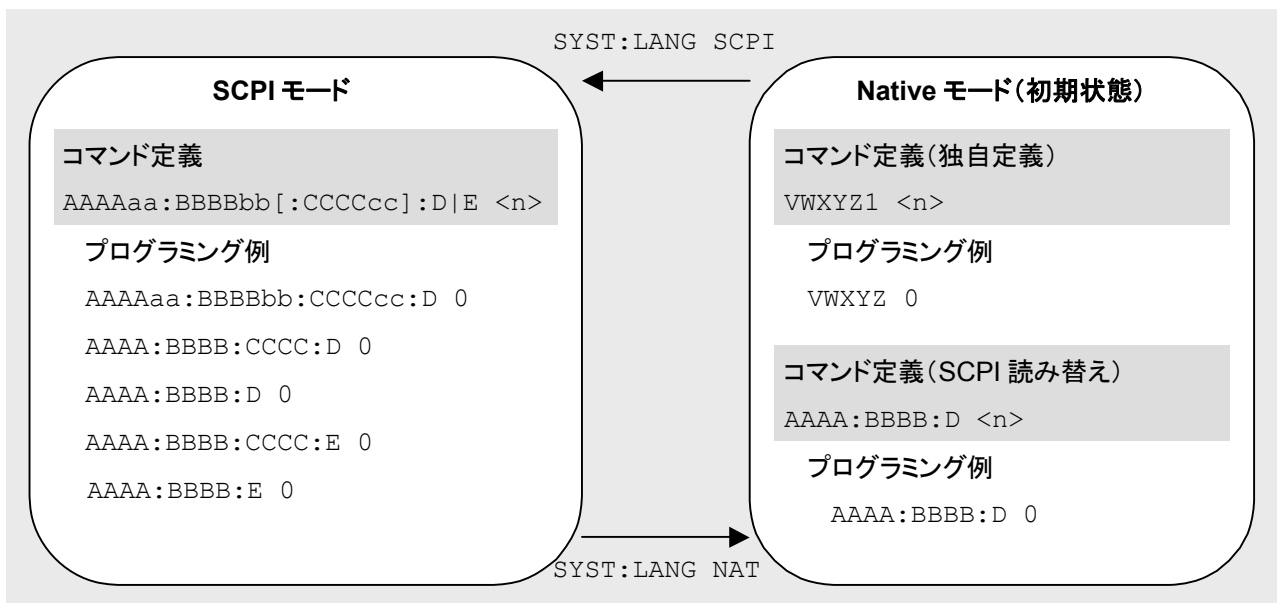


図 1.3-1 SCPI モードと Native モード

本アプリケーションは、SCPI モードのコマンドでのみ定義されています。本アプリケーションの制御を、Native モードで行う場合は、本書で定義されたコマンドを下

記の①～⑤のルールに従って、Native モードに読み替えて使用してください。

読み替えルール

- ① SCPI モードのプログラムヘッダ中の数値パラメータを引数の先頭に移動します。1 種類の値しか取らないもので、かつ省略可能なものは省略します。1 種類の値しか取らないもので、かつ省略不可能なものはそのままにします。
- ② 複数のノードを選択できる場合は先頭のものを使用します。
- ③ 省略できる階層があれば省略します。
- ④ ロングフォーム表記をすべてショートフォーム表記にします。
- ⑤ 先頭の“:”は省略します。

例 1

:CALCulate:MARKer[1]|2[:SET]:CENTER

を Native モードに読み替える

- ① プログラムヘッダ中の数値パラメータを引数の先頭に移動します。

:CALCulate:MARKer[1]|2[:SET]:CENTER

↓

:CALCulate:MARKer[:SET]:CENTER <integer>

(<integer>は 1 または 2 の数値を取る引数を表しています)

- ② 省略できる階層があれば省略します。

:CALCulate:MARKer[:SET]:CENTER <integer>

↓

:CALCulate:MARKer:CENTER <integer>

- ③ ロングフォーム表記をすべてショートフォーム表記にします。

:CALCulate:MARKer:CENTER <integer>

↓

:CALC:MARK:CENT <integer>

- ④ 先頭の“:”は省略します。

:CALC:MARK:CENT <integer>

↓

CALC:MARK:CENT <integer>

例 2

[:SENSe]:BPOWer|:TXPower[:STATe]?

を Native モードに読み替える

- ① 複数のノードを選択できる場合は先頭のものを使用します。

[:SENSe]:BPOWer|:TXPower[:STATe]?

↓

[:SENSe]:BPOWer[:STATe]?

- ② 省略できる階層があれば省略します。

[:SENSe]:BPOWer[:STATe]?

↓

:BPOWer?

- ③ ロングフォーム表記をすべてショートフォーム表記に変更します。

:BPOWer?

↓

:BPOW?

- ④ 先頭の“:”は省略します。

:BPOW?

↓

BPOW?

例 3

:FETCh|:MER[n]?

を Native モードに読み替える

- ① プログラムヘッダ中の数値パラメータを引数の先頭に移動します。

:FETCh:MER[n]?

↓

:FETCh:MER? <integer>

(<integer>は整数を表しています)

- ② ロングフォーム表記をすべてショートフォーム表記にします。

:FETCh:MER? <integer>

↓

:FETC:MER? <integer>

- ③ 先頭の“:”は省略します。

:FETCh:MER? <integer>

↓

FETC:MER? <integer>

- ④ 引数の数値を設定します。

FETCh:MER? <integer>

↓

FETC:MER? 1

1.4 数値プログラムデータの設定について

SCPI モードでは、数値プログラムデータ(数値型パラメータ)の設定に対して、次のキャラクタプログラムを使用することができます。

(1) DEFault

数値プログラムデータに対して DEFault を指定すると、対象のパラメータは初期値に設定されます。

(2) MINimum

数値プログラムデータに対して MINimum を指定すると、対象のパラメータは最小値に設定されます。

(3) MAXimum

数値プログラムデータに対して MAXimum を指定すると、対象のパラメータは最大値に設定されます。

本アプリケーションにおいて、DEFault, MINimum, MAXimum が使用できる数値プログラムデータは、次の表記で示されたパラメータです。

```
<freq>
<real>
<rel_power>
<integer>
<time>
```

第2章 SCPI デバイスメッセージ詳細

この章では、本アプリケーションの機能を実行する SCPI リモート制御コマンドの詳細な仕様を、機能別に説明します。IEEE488.2 共通デバイスメッセージおよびアプリケーション共通デバイスメッセージの詳細な仕様は、『MS2690A/MS2691A/MS2692A シグナルアナライザ取扱説明書（本体 リモート制御編）』を参照してください。

2

SCPI デバイスメッセージ詳細

2.1	アプリケーションの選択	2-4
2.1.1	アプリケーションの起動	2-5
	:SYSTem:APPLication:LOAD MEDIAFLO	2-5
	:SYSTem:APPLication:UNLoad MEDIAFLO	2-5
2.1.2	アプリケーションの選択	2-6
	:INSTrument[:SELeCt] MEDIAFLO CONFIG	2-6
	:INSTrument[:SELeCt]?	2-7
	:INSTrument:SYSTem MEDIAFLO,[ACTive] INACTive MINimum	2-8
	:INSTrument:SYSTem? MEDIAFLO	2-8
2.1.3	初期化	2-9
	:INSTrument:DEFault	2-9
	:SYSTem:PRESet	2-9
2.2	基本パラメータの設定	2-10
2.2.1	Carrier Frequency	2-11
	[:SENSE]:FREQuency:CENTer <freq>	2-11
	[:SENSE]:FREQuency:CENTer?	2-12
2.2.2	RF Spectrum	2-13
	[:SENSE]:MER:SPECtrum NORMAl REVerse	2-13
	[:SENSE]:MER:SPECtrum?	2-13
2.2.3	Input Level	2-14
	[:SENSE]:POWER[:RF]:RANGe:ILEVel <real>	2-14
	[:SENSE]:POWER[:RF]:RANGe:ILEVel?	2-15
2.2.4	Level Offset	2-16
	:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel:OFFSet <rel_power>	2-16
	:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel:OFFSet?	2-16
2.2.5	Level Offset State	2-17
	:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel:OFFSet:STATe OFF ON 0 1	2-17
	:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel:OFFSet:STATe?	2-17
2.2.6	Pre Amp	2-18
	[:SENSE]:POWER[:RF]:GAIN[:STATe] OFF ON 0 1	2-18
	[:SENSE]:POWER[:RF]:GAIN[:STATe]?	2-18
2.3	システムパラメータの設定	2-19
2.3.1	Channel Bandwidth	2-20
	[:SENSE]:RADio:CBANDwidth 5 6 7 8	2-20
	[:SENSE]:RADio:CBANDwidth?	2-20
2.3.2	Tracking Mode	2-21
	:CALCulate:MER:TRACking MANUal ESTimation	2-21
	:CALCulate:MER:TRACking?	2-21
2.3.3	Modulation	2-22

:CALCulate:MER:MODulation QPSK 16Qam ER_6.....	2-22
:CALCulate:MER:MODulation?.....	2-23
2.3.4 SPC Presence	2-24
:CALCulate:MER:SPCPresence ABSent PREsent 0 1.....	2-24
:CALCulate:MER:SPCPresence?	2-24
2.3.5 FFT Size	2-25
:CALCulate:MER:FFT:SIZE 2K 4K 8K.....	2-25
:CALCulate:MER:FFT:SIZE?	2-25
2.3.6 Cyclic Prefix.....	2-26
:CALCulate:MER:CPRefix 1PER16 1PER8 3PER16 1PER4.....	2-26
:CALCulate:MER:CPRefix?	2-27
2.3.7 Slot to Interlace.....	2-28
:CALCulate:MER:STINterlace 1 2.....	2-28
:CALCulate:MER:STINterlace?.....	2-28
2.3.8 PPC Presence	2-29
:CALCulate:MER:PPCPresence ABSent PREsent 0 1.....	2-29
:CALCulate:MER:PPCPresence?	2-29
2.3.9 PPC / RS (Reserved OFDM Symbol) Number.....	2-30
:CALCulate:MER:NUMBer:PPCRs 2 6 10 14	2-30
:CALCulate:MER: NUMBer:PPCRs?	2-31
2.3.10 Wide-area Data Symbols.....	2-32
:CALCulate:MER:WDSYmbols <integer>	2-32
:CALCulate:MER:WDSYmbols?	2-33
2.3.11 WID	2-34
:CALCulate:MER:WID <integer>	2-34
:CALCulate:MER:WID?	2-34
2.3.12 LID	2-35
:CALCulate:MER:LID <integer>	2-35
:CALCulate:MER:LID?	2-35
2.3.13 Channel Estimation	2-36
:CALCulate:MER:CHANnel:ESTimation OFF ON 0 1	2-36
:CALCulate:MER:CHANnel:ESTimation?	2-36
2.3.14 End Symbol	2-37
[:SENSe]:MER:SYMBol:END <integer>	2-37
[:SENSe]:MER:SYMBol:END?.....	2-37
2.3.15 Frame Select	2-38
[:SENSe]:MER:FSElect MANual OIS 1 2 3 4.....	2-38
[:SENSe]:MER:FSElect?	2-39
2.4 共通測定機能	2-40
2.4.1 測定と制御.....	2-41
:INITiate:CONTInuous OFF ON 0 1	2-41
:INITiate:CONTInuous?.....	2-41
:INITiate:MODE:CONTInuous.....	2-42
:INITiate:MODE:SINGle	2-42
:INITiate[:IMMEdiate].....	2-42
2.4.2 Trigger Switch.....	2-43
:TRIGger[:SEQuence][:STATe] OFF ON 0 1	2-43

:TRIGger[:SEQuence][:STATe]?	2-43
2.4.3 Trigger Source	2-44
:TRIGger[:SEQuence]:SOURce EXTeRnal[1] SG	2-44
:TRIGger[:SEQuence]:SOURce?	2-45
2.4.4 Trigger Slope	2-46
:TRIGger[:SEQuence]:SLOPe POSitive NEGative	2-46
:TRIGger[:SEQuence]:SLOPe?	2-46
2.4.5 Trigger Delay	2-47
:TRIGger[:SEQuence]:DELay <time>	2-47
:TRIGger[:SEQuence]:DELay?	2-47
2.5 Modulation 測定機能	2-48
2.5.1 Measure	2-49
:FETCh:MER[n]?	2-49
2.5.2 Storage Mode	2-52
[:SENSe]:MER:AVERage[:STATe] OFF ON AMAXimum 0 1 2	2-52
[:SENSe]:MER:AVERage[:STATe]?	2-52
2.5.3 Storage Count	2-53
[:SENSe]:MER:AVERage:COUNT <integer>	2-53
[:SENSe]:MER:AVERage:COUNT?	2-53
2.5.4 Graph View Setting	2-54
:CALCulate:MER:WINDow2:MODE EACH AVERage	2-54
:CALCulate:MER:WINDow2:MODE?	2-54
2.6 測定結果の保存機能	2-55
:MMEMory:STORe:RESult [<filename>[,<device>]]	2-56
:MMEMory:STORe:RESult:MODE XML CSV	2-57
:MMEMory:STORe:RESult:MODE?	2-57

2.1 アプリケーションの選択

アプリケーションの起動・選択・初期化などのアプリケーションのセットアップに関するデバイスメッセージは表 2.1-1 のとおりです。

表 2.1-1 アプリケーションの選択

機能	デバイスメッセージ
Load Application	:SYSTem:APPLication:LOAD MEDIAFLO
Unload Application	:SYSTem:APPLication:UNLoad MEDIAFLO
Application Switch	:INSTrument[:SElect] MEDIAFLO CONFIG
	:INSTrument[:SElect]?
Application Status	:INSTrument:SYSTem MEDIAFLO, [ACTive] INACTive MINimum
	:INSTrument:SYSTem? MEDIAFLO
Initialization	:INSTrument:DEFault
	:SYSTem:PRESet

2.1.1 アプリケーションの起動

:SYSTem:APPLication:LOAD MEDIAFLO

Load Application

機能

本アプリケーションを起動します。

コマンド

```
:SYSTem:APPLication:LOAD MEDIAFLO
```

詳細

本機能により、インストールされているアプリケーションが起動し、Application Switch メニューに登録されます。

制御対象のアプリケーションが Config のときに設定できます。

本アプリケーションを終了した直後は 30 秒程度、間隔を置いてから起動してください。

使用例

本アプリケーションを起動する

```
SYST:APPL:LOAD MEDIAFLO
```

:SYSTem:APPLication:UNLoad MEDIAFLO

Unload Application

機能

本アプリケーションを終了します。

コマンド

```
:SYSTem:APPLication:UNLoad MEDIAFLO
```

詳細

本機能により、起動中のアプリケーションが終了し、Application Switch メニューから削除されます。

制御対象のアプリケーションが Config のときに設定できます。

使用例

本アプリケーションを終了する

```
SYST:APPL:UNL MEDIAFLO
```

2.1.2 アプリケーションの選択

:INSTrument[:SElect] MEDIAFLO|CONFIG

Application Switch

機能

制御対象のアプリケーションを選択します。

コマンド

```
:INSTrument[:SElect] <apl_name>
```

パラメータ

<apl_name>	アプリケーション
MEDIAFLO	本アプリケーション
CONFIG	Config

使用例

制御対象を本アプリケーションに切り替える
INST MEDIAFLO

:INSTrument[:SElect]?

Application Switch Query

機能

制御対象のアプリケーションを読み出します。

クエリ

```
:INSTrument[:SElect]?
```

レスポンス

```
<apl_name>
```

パラメータ

<apl_name>	アプリケーション
MEDIAFLO	本アプリケーション
SIGANA	シグナルアナライザ
SPECT	スペクトラムアナライザ
CONFIG	Config

詳細

本アプリケーションの測定機能を選択しているときは、MEDIAFLO が返ります。

使用例

```
制御対象のアプリケーションを読み出す
INST?
> MEDIAFLO
```

:INSTrument:SYSTem MEDIAFLO,[ACTive]|INACTive|MINimum

Application Switch And Window Status

機能

本アプリケーションのウィンドウ状態を選択します。

コマンド

```
:INSTrument:SYSTem MEDIAFLO,<window>
```

パラメータ

<window>	ウィンドウの状態
ACTive	アクティブ状態
INACTive	非アクティブ状態
MINimum	最小化された状態
省略時	アクティブ状態

使用例

本アプリケーションのウィンドウ状態をアクティブ状態に設定します
INST:SYST MEDIAFLO,ACT

:INSTrument:SYSTem? MEDIAFLO

Application Switch And Window Status Query

機能

本アプリケーションの状態を読み出します。

クエリ

```
:INSTrument:SYSTem? MEDIAFLO
```

レスポンス

```
<status>,<window>
```

パラメータ

<status>	本アプリケーションの状態
CURR	実行中で制御対象である
RUN	実行中で制御対象でない
IDLE	起動しているが、実行されていない状態
UNL	起動されていない状態
<window>	ウィンドウの状態
ACT	アクティブ状態
INAC	非アクティブ状態
MIN	最小化された状態
NON	ウィンドウが表示されていない状態

使用例

本アプリケーションの状態を読み出す
INST:SYST? MEDIAFLO
> CURR, ACT

2.1.3 初期化

:INSTrument:DEFault

Preset Current Application

機能

現在選択しているアプリケーションの設定と状態を初期化します。

コマンド

```
:INSTrument:DEFault
```

使用例

現在選択しているアプリケーションの設定と状態を初期化する
INST:DEF

:SYSTem:PRESet

Preset Current Application

機能

現在選択しているアプリケーションの設定と状態を初期化します。

:INSTrument:DEFault を参照してください。

使用例

現在選択しているアプリケーションの設定と状態を初期化する
SYST:PRES

2.2 基本パラメータの設定

周波数・レベルなどの本アプリケーションにおいて共通に適用されるパラメータ設定に関するデバイスメッセージは表 2.2-1 のとおりです。

表 2.2-1 基本パラメータの設定

パラメータ	デバイスメッセージ
Carrier Frequency	<code>[[:SENSE]:FREQUENCY:CENTER <freq></code>
	<code>[[:SENSE]:FREQUENCY:CENTER?</code>
RF Spectrum	<code>[[:SENSE]:MER:SPECTRUM NORMAL REVERSE</code>
	<code>[[:SENSE]:MER:SPECTRUM?</code>
Input Level	<code>[[:SENSE]:POWER[:RF]:RANGE:ILEVEL <real></code>
	<code>[[:SENSE]:POWER[:RF]:RANGE:ILEVEL?</code>
Level Offset	<code>:DISPLAY:WINDOW[1]:TRACE:Y[:SCALE]:RLEVEL:OFFSET <rel_power></code>
	<code>:DISPLAY:WINDOW[1]:TRACE:Y[:SCALE]:RLEVEL:OFFSET?</code>
Level Offset State	<code>:DISPLAY:WINDOW[1]:TRACE:Y[:SCALE]:RLEVEL:OFFSET:STATE OFF ON 0 1</code>
	<code>:DISPLAY:WINDOW[1]:TRACE:Y[:SCALE]:RLEVEL:OFFSET:STATE?</code>
Pre-Amp State	<code>[[:SENSE]:POWER[:RF]:GAIN[:STATE] OFF ON 0 1</code>
	<code>[[:SENSE]:POWER[:RF]:GAIN[:STATE]?</code>

2.2.1 Carrier Frequency

`[[:SENSe]:FREQuency:CENTer <freq>`

Carrier Frequency

機能

被測定信号のキャリア周波数を設定します。

コマンド

`[[:SENSe]:FREQuency:CENTer <freq>`

パラメータ

<freq>	キャリア周波数
範囲	30 MHz～6 GHz (MS2690A) 30 MHz～13.5 GHz (MS2691A) 30 MHz～26.5 GHz (MS2692A)
分解能	1 Hz
サフィックスコード	HZ, KHZ, KZ, MHZ, MZ, GHZ, GZ 省略した場合は Hz として扱われます。
初期値	207 MHz

使用例

キャリア周波数を 1.000 GHz に設定する

`FREQ:CENT 1.000GHZ`

[:SENSe]:FREQuency:CENTer?

Carrier Frequency Query

機能

被測定信号のキャリア周波数を読み出します。

クエリ

```
[ :SENSe ] :FREQuency:CENTer?
```

レスポンス

```
<freq>
```

パラメータ

<freq>	キャリア周波数
範囲	30 MHz～6 GHz (MS2690A) 30 MHz～13.5 GHz (MS2691A) 30 MHz～26.5 GHz (MS2692A)
分解能	1 Hz
	Hz 単位の値を返します。

使用例

```
キャリア周波数を読み出す  
FREQ:CENT?  
> 6000000000
```

2.2.2 RF Spectrum

[:SENSe]:MER:SPECTrum NORMal|REVerse

RF Spectrum

機能

入力信号のスペクトラム反転を設定します

コマンド

[:SENSe]:MER:SPECTrum <mode>

パラメータ

<mode>	スペクトラム反転
NORMal	行わない (初期値)
REVerse	行う

使用例

スペクトラム反転を行うに設定する
MER:SPEC REV

[:SENSe]:MER:SPECTrum?

RF Spectrum Query

機能

入力信号のスペクトラム反転の設定を読み出します。

クエリ

[:SENSe]:MER:SPECTrum?

レスポンス

<mode>

パラメータ

<mode>	スペクトラム反転
NORM	行わない
REV	行う

使用例

スペクトラム反転の設定を読み出す
MER:SPEC?
> REV

2.2.3 Input Level

`[[:SENSe]:POWer[:RF]:RANGe:ILEVel <real>`

Input Level

機能

RF 信号の入力レベルを設定します。

コマンド

`[[:SENSe]:POWer[:RF]:RANGe:ILEVel <real>`

パラメータ

<code><real></code>	入力レベル値
範囲	($-60.00 + \text{Level Offset}$) ~ ($30.00 + \text{Level Offset}$) dBm (Pre-Amp が Off の場合) ($-80.00 + \text{Level Offset}$) ~ ($10.00 + \text{Level Offset}$) dBm (Pre-Amp が On の場合)
分解能	0.01 dB
単位	1 dBm
サフィックスコード	DBM 省略した場合は dBm として扱われます。
初期値	-10.00 dBm

詳細

MS2690A/MS2691A/MS2692A オプション 008 6 GHz プリアンプ (以下, オプション 008) が未搭載のときは, Off の設定範囲となります。

使用例

入力レベルを 0 dBm に設定する
`POW:RANG:ILEV 0`

[:SENSe]:POWer[:RF]:RANGe:ILEVel?

Input Level Query

機能

RF 信号の入力レベルを読み出します。

クエリ

[:SENSe]:POWer[:RF]:RANGe:ILEVel?

レスポンス

<real>

パラメータ

<real>

範囲

入力レベル値

 $(-60.00 + \text{Level Offset}) \sim (30.00 + \text{Level Offset})$
 dBm (Pre-Amp が Off の場合)

 $(-80.00 + \text{Level Offset}) \sim (10.00 + \text{Level Offset})$
 dBm (Pre-Amp が On の場合)

分解能

0.01 dB

dBm 単位の値を返します。

使用例

入力レベルを読み出す

POW:RANG:ILEV?

> -15.00

2.2.4 Level Offset

:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel:OFFSet <rel_power>

Level Offset Value

機能

入力レベルのオフセット値を設定します。

コマンド

```
:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel:OFFSet  
<rel_power>
```

パラメータ

<rel_power>	オフセット値
範囲	-99.99~+99.99 dB
分解能	0.01 dB
サフィックスコード	DB
	省略した場合は dB として扱われます。
初期値	0 dB

使用例

入力レベルのオフセット値を+10 dB に設定する
DISP:WIND:TRAC:Y:RLEV:OFFS 10

:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel:OFFSet?

Level Offset Value Query

機能

入力レベルのオフセット値を読み出します。

クエリ

```
:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel:OFFSet?
```

レスポンス

```
<rel_power>
```

パラメータ

<rel_power>	オフセット値
範囲	-99.99~+99.99 dB
分解能	0.01 dB

使用例

入力レベルのオフセット値を読み出す
DISP:WIND:TRAC:Y:RLEV:OFFS?
> 10.00

2.2.5 Level Offset State

```
:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel:OFFSet:STATe OFF|ON|0|1
```

Level Offset State

機能

入力レベルのオフセット機能の有効・無効を設定します。

コマンド

```
:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel:OFFSet:STATe
<switch>
```

パラメータ

<switch>	入力レベルのオフセット機能の有効・無効
OFF 0	無効にする (初期値)
ON 1	有効にする

使用例

入力レベルのオフセット機能を有効にする
 DISP:WIND:TRAC:Y:RLEV:OFFS:STAT ON

```
:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel:OFFSet:STATe?
```

Level Offset State Query

機能

入力レベルのオフセット機能の有効・無効を読み出します。

クエリ

```
:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel:OFFSet:STATe?
```

レスポンス

```
<switch>
```

パラメータ

<switch>	入力レベルのオフセット機能の有効・無効
0	無効
1	有効

使用例

入力レベルのオフセット機能の有効・無効を読み出す
 DISP:WIND:TRAC:Y:RLEV:OFFS:STAT?
 > 1

2.2.6 Pre Amp

`[[:SENSe]:POWer[:RF]:GAIN[:STATe] OFF|ON|0|1`

Pre Amp

機能

Pre-Amp の On・Off を設定します。

コマンド

`[[:SENSe]:POWer[:RF]:GAIN[:STATe] <switch>`

パラメータ

<code><switch></code>	Pre-Amp の On・Off
<code>OFF 0</code>	Off (初期値)
<code>ON 1</code>	On

詳細

オプション 008 が未搭載のとき本コマンドは無効です。

使用例

Pre-Amp を On に設定する
`POW:GAIN ON`

`[[:SENSe]:POWer[:RF]:GAIN[:STATe]?`

Pre Amp Query

機能

Pre-Amp の On・Off を読み出します。

クエリ

`[[:SENSe]:POWer[:RF]:GAIN[:STATe]?`

レスポンス

`<switch>`

パラメータ

<code><switch></code>	Pre-Amp の On・Off
<code>0</code>	Off
<code>1</code>	On

詳細

オプション 008 が未搭載のときは常に Off の値を返します。

使用例

Pre-Amp の設定を読み出す
`POW:GAIN?`
`> 1`

2.3 システムパラメータの設定

測定対象の通信システムに関するデバイスメッセージは表 2.3-1 のとおりです。

表 2.3-1 システムパラメータの設定

パラメータ	デバイスメッセージ
Channel Bandwidth	<code>[:SENSE] :RADio :CBANdwidth 5 6 7 8</code>
	<code>[:SENSE] :RADio :CBANdwidth?</code>
Tracking Mode	<code>:CALCulate :MER :TRACking MANual ESTimation</code>
	<code>:CALCulate :MER :TRACking?</code>
Modulation	<code>:CALCulate :MER :MODulation QPSK 16Qam ER_6</code>
	<code>:CALCulate :MER :MODulation?</code>
SPC Presence	<code>:CALCulate :MER :SPCPresence ABSent PREsent 0 1</code>
	<code>:CALCulate :MER :SPCPresence?</code>
FFT Size	<code>:CALCulate :MER :FFT :SIZE 2K 4K 8K</code>
	<code>:CALCulate :MER :FFT :SIZE?</code>
Cyclic Prefix	<code>:CALCulate :MER :CPRefix 1PER16 1PER8 3PER16 1PER4</code>
	<code>:CALCulate :MER :CPRefix?</code>
Slot to Interlace	<code>:CALCulate :MER :STINterlace 1 2</code>
	<code>:CALCulate :MER :STINterlace?</code>
PPC Presence	<code>:CALCulate :MER :PPCPresence ABSent PREsent 0 1</code>
	<code>:CALCulate :MER :PPCPresence?</code>
PPC / RS Number	<code>:CALCulate :MER :NUMBer :PPCRs 2 6 10 14</code>
	<code>:CALCulate :MER :NUMBer :PPCRs?</code>
Wide-area Data Symbols	<code>:CALCulate :MER :WDSymbols <integer></code>
	<code>:CALCulate :MER :WDSymbols?</code>
WID	<code>:CALCulate :MER :WID <integer></code>
	<code>:CALCulate :MER :WID?</code>
LID	<code>:CALCulate :MER :LID <integer></code>
	<code>:CALCulate :MER :LID?</code>
Channel Estimation	<code>:CALCulate :MER :CHANnel :ESTimation OFF ON 0 1</code>
	<code>:CALCulate :MER :CHANnel :ESTimation?</code>
End Symbol	<code>[:SENSE] :MER :SYMBol :END <integer></code>
	<code>[:SENSE] :MER :SYMBol :END?</code>
Frame Select	<code>[:SENSE] :MER :FSElect MANual OIS 1 2 3 4</code>
	<code>[:SENSE] :MER :FSElect?</code>

2.3.1 Channel Bandwidth

`[[:SENSe]:RADio:CBANdwidth 5|6|7|8`

Channel Bandwidth

機能

被測定信号の帯域を設定します。

コマンド

```
[[:SENSe]:RADio:CBANdwidth <mode>
```

パラメータ

<mode>	被測定信号の帯域
5	5 MHz 帯域信号として解析する
6	6 MHz 帯域信号として解析する (初期値)
7	7 MHz 帯域信号として解析する
8	8 MHz 帯域信号として解析する

使用例

被測定信号の帯域を 5 MHz に設定する

```
RAD:CBAN 5
```

`[[:SENSe]:RADio:CBANdwidth?`

Channel Bandwidth Query

機能

帯域の設定を読み出します。

クエリ

```
[[:SENSe]:RADio:CBANdwidth?
```

レスポンス

```
<mode>
```

パラメータ

<mode>	被測定信号の帯域
5	5 MHz 帯域信号として解析
6	6 MHz 帯域信号として解析
7	7 MHz 帯域信号として解析
8	8 MHz 帯域信号として解析

使用例

被測定信号の帯域の設定を読み出す

```
RAD:CBAN?
```

```
> 5
```

2.3.2 Tracking Mode

:CALCulate:MER:TRACking MANual|ESTimation

Tracking Mode

機能

Tracking Mode の種類を設定します。

コマンド

```
:CALCulate:MER:TRACking MANual|ESTimation <mode>
```

パラメータ

<mode>	Tracking Mode
MANual	Tracking Mode を手動で設定する (初期値)
ESTimation	Tracking Mode を変調方式推定に設定する

使用例

Tracking Mode を変調方式推定に設定する
 CALC:MER:TRAC EST

:CALCulate:MER:TRACking?

Tracking Mode Query

機能

Tracking Mode の種類を読み出します。

クエリ

```
:CALCulate:MER:TRACking?
```

レスポンス

```
<mode>
```

パラメータ

<mode>	Tracking Mode
MAN	Tracking Mode を手動で設定
EST	Tracking Mode を変調方式推定に設定

使用例

Tracking Mode の設定を読み出す
 CALC:MER:TRAC?
 > EST

2.3.3 Modulation

:CALCulate:MER:MODulation QPSK|16Qam|ER_6

Modulation

機能

変調方式を設定します。

コマンド

:CALCulate:MER:MODulation <mode>

パラメータ

<mode>	変調方式
QPSK	変調方式を QPSK として解析する
16Qam	変調方式を 16QAM/Layered (ER=4) として解析する (初期値)
ER_6	変調方式を Layered (ER=6.25) として解析する

詳細

Tracking Mode が Manual のときに設定できます。

使用例

変調方式を QPSK に設定する
CALC:MER:MOD QPSK

:CALCulate:MER:MODulation?

Modulation Query

機能

変調方式を読み出します。

クエリ

:CALCulate:MER:MODulation?

レスポンス

<mode>

パラメータ

<mode>	変調方式
QPSK	変調方式を QPSK として解析
16Q	変調方式を 16QAM/Layered (ER=4) として解析
ER_6	変調方式を Layered (ER=6.25) として解析
***	変調方式を推定して解析

詳細

Tracking Mode が Manual 以外のときは "***" を返します。

使用例

```

変調方式を読み出す
CALC:MER:MOD?
> QPSK

```

2.3.4 SPC Presence

:CALCulate:MER:SPCPresence ABSent|PREsent|0|1

SPC Presence

機能

SPCの有無を設定します。

コマンド

```
:CALCulate:MER:SPCPresence <switch>
```

パラメータ

<switch>	SPCの有無
ABSent 0	Absent (無効)
PREsent 1	Present (有効) (初期値)

使用例

SPCを無効に設定する
CALC:MER:SPCP ABS

:CALCulate:MER:SPCPresence?

SPC Presence Query

機能

SPCの有無の設定を読み出します。

クエリ

```
:CALCulate:MER:SPCPresence?
```

レスポンス

```
<switch>
```

パラメータ

<switch>	SPCの有無
ABS	Absent (無効)
PRES	Present (有効)

使用例

SPCの有無の設定を読み出す
CALC:MER:SPCP?
> ABS

2.3.5 FFT Size

:CALCulate:MER:FFT:SIZE 2K|4K|8K

FFT Size

機能

FFT Size を設定します。

コマンド

:CALCulate:MER:FFT:SIZE <mode>

パラメータ

<mode>	FFT Size
2K	FFT Size を 2K として解析する
4K	FFT Size を 4K として解析する (初期値)
8K	FFT Size を 8K として解析する

詳細

SPC Presence が Present のときに設定できます。

使用例

FFT Size を 2K に設定する
 CALC:MER:FFT:SIZE 2K

:CALCulate:MER:FFT:SIZE?

FFT Size Query

機能

FFT Size を読み出します。

クエリ

:CALCulate:MER:FFT:SIZE?

レスポンス

<mode>

パラメータ

<mode>	FFT Size
2K	FFT Size を 2K として解析
4K	FFT Size を 4K として解析
8K	FFT Size を 8K として解析

詳細

SPC Presence が Absent のときは 4K を返します。

使用例

FFT Size を読み出す
 CALC:MER:FFT:SIZE?
 > 2K

2.3.6 Cyclic Prefix

:CALCulate:MER:CPRefix 1PER16|1PER8|3PER16|1PER4

Cyclic Prefix

機能

Cyclic Prefix を設定します。

コマンド

:CALCulate:MER:CPRefix <mode>

パラメータ

<mode>	Cyclic Prefix
1PER16	Cyclic Prefix を 1/16 として解析する
1PER8	Cyclic Prefix を 1/8 として解析する (初期値)
3PER16	Cyclic Prefix を 3/16 として解析する
1PER4	Cyclic Prefix を 1/4 として解析する

詳細

SPC Presence が Present のときに設定できます。

使用例

Cyclic Prefix を 1/16 に設定する
CALC:MER:CPR 1PER16

:CALCulate:MER:CPRefix?

Cyclic Prefix Query

機能

Cyclic Prefix を読み出します。

クエリ

:CALCulate:MER:CPRefix?

レスポンス

<mode>

パラメータ

<mode>	Cyclic Prefix
1PER16	Cyclic Prefix を 1/16 として解析
1PER8	Cyclic Prefix を 1/8 として解析
3PER16	Cyclic Prefix を 3/16 として解析
1PER4	Cyclic Prefix を 1/4 として解析

詳細

SPC Presence が Absent のときは 1PER8 を返します。

使用例

```

Cyclic Prefix を読み出す
CALC:MER:CPR?
> 1PER16

```

2.3.7 Slot to Interlace

:CALCulate:MER:STINterlace 1|2

Slot to Interlace

機能

Slot to Interlace を設定します。

コマンド

```
:CALCulate:MER:STINterlace <mode>
```

パラメータ

<mode>	Slot to Interlace
1	Slot to Interlace を Pattern1 として解析する (初期値)
2	Slot to Interlace を Pattern2 として解析する

詳細

SPC Presence が Present のときに設定できます。

使用例

Slot to Interlace を Pattern2 に設定する
CALC:MER:STIN 2

:CALCulate:MER:STINterlace?

Slot to Interlace Query

機能

Slot to Interlace を読み出します。

クエリ

```
:CALCulate:MER:STINterlace?
```

レスポンス

```
<mode>
```

パラメータ

<mode>	Slot to Interlace
1	Slot to Interlace を Pattern1 として解析
2	Slot to Interlace を Pattern2 として解析

詳細

SPC Presence が Absent のときは 1 を返します。

使用例

Slot to Interlace を読み出す
CALC:MER:STIN?
> 2

2.3.8 PPC Presence

:CALCulate:MER:PPCPresence ABSent|PREsent|0|1

PPC Presence

機能

PPC の有無を設定します。

コマンド

```
:CALCulate:MER:PPCPresence <switch>
```

パラメータ

<switch>	PPC の有無
ABSent 0	Absent (無効)
PREsent 1	Present (有効) (初期値)

使用例

PPC を無効に設定する
 CALC:MER:PPCP ABS

:CALCulate:MER:PPCPresence?

PPC Presence Query

機能

PPC の有無の設定を読み出します。

クエリ

```
:CALCulate:MER:PPCPresence?
```

レスポンス

```
<switch>
```

パラメータ

<switch>	PPC の有無
ABS	Absent (無効)
PRES	Present (有効)

使用例

PPC の有無の設定を読み出す
 CALC:MER:PPCP?
 > ABS

2.3.9 PPC / RS (Reserved OFDM Symbol) Number

:CALCulate:MER:NUMBER:PPCRs 2|6|10|14

PPC / RS (Reserved OFDM Symbol) Number

機能

PPC または RS Number を設定します。

コマンド

:CALCulate:MER:NUMBER:PPCRs <mode>

パラメータ

<mode>	PPC / RS Number
2	PPC / RS Number を 2 として解析する
6	PPC / RS Number を 6 として解析する (初期値)
10	PPC / RS Number を 10 として解析する
14	PPC / RS Number を 14 として解析する

詳細

SPC Presence が Absent のときに設定できます。

ただし、PPC Presence が Present のときは 2 に設定できません。

使用例

PPC / RS Number を 10 に設定する

CALC:MER:NUMB:PPCR 10

:CALCulate:MER: NUMBER:PPCRs?

PPC / RS (Reserved OFDM Symbol) Number Query

機能

PPC または RS Number を読み出します。

クエリ

:CALCulate:MER:NUMBER:PPCRs?

レスポンス

<mode>

パラメータ

<mode>	PPC / RS Number
0	PPC / RS Number を使用しない
2	PPC / RS Number を 2 として解析
6	PPC / RS Number を 6 として解析
8	PPC / RS Number を 8 として解析
10	PPC / RS Number を 10 として解析
14	PPC / RS Number を 14 として解析
16	PPC / RS Number を 16 として解析

詳細

SPC Presence が Present でかつ、PPC Presence が Absent のとき、0 を返します。

SPC Presence が Present でかつ、PPC Presence が Present のとき、FFT Size により以下のように返します。

FFT Size が 2K のとき 16 を返します。

FFT Size が 4K および 8K のとき 8 を返します。

使用例

```

PPC / RS Number を読み出す
CALC:MER:NUMB:PPCR?
> 10

```

2.3.10 Wide-area Data Symbols

:CALCulate:MER:WDSYmbols <integer>

Wide-area Data Symbols

機能

Wide-area Data Symbols を設定します。

コマンド

:CALCulate:MER:WDSYmbols <integer>

パラメータ

<integer> Wide-area Data Symbols

範囲

SPC Presence が Absent のとき,

$$0 \sim \frac{(\text{Superframe Symbol} - 18 - \text{PPC} / \text{RS Number})}{4} \quad -4$$

SPC Presence が Present のとき,

0 ~ (TIA-1099-A 規格に準拠)

分解能 1

サフィックスコード なし

初期値 288

使用例

Wide-area Data Symbols を 150 に設定する

CALC:MER:WDSY 150

:CALCulate:MER:WDSYmbols?

Wide-area Data Symbols Query

機能

Wide-area Data Symbols を読み出します。

クエリ

:CALCulate:MER:WDSYmbols?

レスポンス

<integer>

パラメータ

<integer> Wide-area Data Symbols

範囲

SPC Presence が Absent のとき,

$$0 \sim \frac{(\text{Superframe Symbol} - 18 - \text{PPC} / \text{RS Number})}{4} - 4$$

SPC Presence が Present のとき,

0 ~ (TIA-1099-A 規格に準拠)

分解能

1

使用例

Wide-area Data Symbols を読み出す

CALC:MER:WDSY?

> 150

2.3.11 WID

:CALCulate:MER:WID <integer>

WID

機能

WID を設定します。

コマンド

:CALCulate:MER:WID <integer>

パラメータ

<integer>	WID
範囲	0~15
分解能	1
サフィックスコード	なし
初期値	15

使用例

WID を 1 に設定する
CALC:MER:WID 1

:CALCulate:MER:WID?

WID Query

機能

WID を読み出します。

クエリ

:CALCulate:MER:WID?

レスポンス

<integer>

パラメータ

<integer>	WID
範囲	0~15
分解能	1

使用例

WID を読み出す
CALC:MER:WID?
> 1

2.3.12 LID

:CALCulate:MER:LID <integer>

LID

機能

LID を設定します。

コマンド

:CALCulate:MER:LID <integer>

パラメータ

<integer>	LID
範囲	0~15
分解能	1
サフィックスコード	なし
初期値	10

使用例

LID を 1 に設定する
 CALC:MER:LID 1

:CALCulate:MER:LID?

LID Query

機能

LID を読み出します。

クエリ

:CALCulate:MER:LID?

レスポンス

<integer>

パラメータ

<integer>	LID
範囲	0~15
分解能	1

使用例

LID を読み出す
 CALC:MER:LID?
 > 1

2.3.13 Channel Estimation

:CALCulate:MER:CHANnel:ESTimation OFF|ON|0|1

Channel Estimation

機能

Channel Estimation 機能の On・Off を設定します。

コマンド

```
:CALCulate:MER:CHANnel:ESTimation <switch>
```

パラメータ

<switch>	Channel Estimation の On・Off
OFF 0	Off
ON 1	On (初期値)

使用例

Channel Estimation を Off にする
CALC:MER:CHAN:EST OFF

:CALCulate:MER:CHANnel:ESTimation?

Channel Estimation Query

機能

Channel Estimation 機能の設定を読み出します。

クエリ

```
:CALCulate:MER:CHANnel:ESTimation?
```

レスポンス

```
<switch>
```

パラメータ

<switch>	Channel Estimation の On・Off
0	Off
1	On

使用例

Channel Estimation の設定を読み出す
CALC:MER:CHAN:EST?
> 0

2.3.14 End Symbol

[:SENSe]:MER:SYMBOL:END <integer>

End Symbol

機能

解析終了シンボルを設定します。

コマンド

[:SENSe]:MER:SYMBOL:END <integer>

パラメータ

<integer>	解析終了シンボル
範囲	FRAME OIS の終端シンボル～ (Superframe Symbol – PPC/SPC Symbol – 1)
分解能	1
サフィックスコード	なし
初期値	17Symbol

使用例

解析終了シンボルを 20 に設定する
MER:SYMB:END 20

[:SENSe]:MER:SYMBOL:END?

End Symbol Query

機能

解析終了シンボルを読み出します。

クエリ

[:SENSe]:MER:SYMBOL:END?

レスポンス

<integer>

パラメータ

<integer>	解析終了シンボル
範囲	FRAME OIS の終端シンボル～ (Superframe Symbol – PPC/SPC Symbol – 1)
分解能	1

使用例

解析終了シンボルを読み出す
MER:SYMB:END?
> 20

2.3.15 Frame Select

`[[:SENSe]:MER:FSElect MANual|OIS|1|2|3|4`

Frame Select

機能

Frame Select を設定します。
Frame Select の設定を変更することにより、解析終了シンボルの値が変わります。

コマンド

`[[:SENSe]:MER:FSElect <mode>`

パラメータ

<mode>	Frame Select
MANual	解析終了シンボルを手動で設定する
OIS	解析終了シンボルを FRAME OIS までにする (初期値)
1	解析終了シンボルを FRAME 1 までにする
2	解析終了シンボルを FRAME 2 までにする
3	解析終了シンボルを FRAME 3 までにする
4	解析終了シンボルを FRAME 4 までにする

使用例

Frame Select を FRAME 1 に設定する
`MER:FSEL 1`

[:SENSe]:MER:FSElect?

Frame Select Query

機能

Frame Select を読み出します。

クエリ

[:SENSe]:MER:FSElect?

レスポンス

<mode>

パラメータ

<mode>

MAN

Frame Select

解析終了シンボルを手動で設定

OIS

解析終了シンボルが **FRAME OIS** まで
(初期値)

1

解析終了シンボルが **FRAME 1** まで

2

解析終了シンボルが **FRAME 2** まで

3

解析終了シンボルが **FRAME 3** まで

4

解析終了シンボルが **FRAME 4** まで

使用例

Frame Select を読み出す

MER:FSEL?

> 1

2.4 共通測定機能

各測定機能に共通する操作を行うデバイスメッセージは表 2.4-1 のとおりです。

表 2.4-1 共通測定機能

機能	デバイスメッセージ
Continuous Measurement	:INITiate:CONTinuous OFF ON 0 1
	:INITiate:CONTinuous?
	:INITiate:MODE:CONTinuous
Single Measurement	:INITiate:MODE:SINGLE
Initiate	:INITiate[:IMMediate]
Trigger Switch	:TRIGger[:SEQuence][:STATe] ON OFF 1 0
	:TRIGger[:SEQuence][:STATe]?
Trigger Source	:TRIGger[:SEQuence]:SOURce EXTernal[1] SG
	:TRIGger[:SEQuence]:SOURce?
Trigger Slope	:TRIGger[:SEQuence]:SLOPe POSitive NEGative
	:TRIGger[:SEQuence]:SLOPe?
Trigger Delay	:TRIGger[:SEQuence]:DELay <time>
	:TRIGger[:SEQuence]:DELay?

注:

トリガの設定は、各アプリケーションに保持され、アプリケーション内での各測定機能に対しては共通に適用されます。

2.4.1 測定と制御

:INITiate:CONTinuous OFF|ON|0|1

Continuous Measurement

機能

測定モードを設定します。

コマンド

```
:INITiate:CONTinuous <switch>
```

パラメータ

<switch>	測定モード
0 OFF	シングル測定
1 ON	連続測定 (初期値)

詳細

On 設定時は連続測定を開始します。**Off** 設定時はシングル測定になり測定は開始しません。

使用例

連続測定を実行する
INIT:CONT ON

:INITiate:CONTinuous?

Continuous Measurement Query

機能

測定モードを読み出します。

クエリ

```
:INITiate:CONTinuous?
```

レスポンス

```
<switch>
```

パラメータ

<switch>	測定モード
0	シングル測定
1	連続測定

使用例

測定モードを読み出す
INIT:CONT?
> 1

:INITiate:MODE:CONTInuous

Continuous Measurement

機能

連続測定を開始します。

コマンド

```
:INITiate:MODE:CONTInuous
```

使用例

連続測定を開始する
INIT:MODE:CONT

:INITiate:MODE:SINGle

Single Measurement

機能

シングル測定を開始します。

コマンド

```
:INITiate:MODE:SINGle
```

使用例

シングル測定を開始する
INIT:MODE:SING

:INITiate[:IMMediate]

Initiate

機能

現在の測定モードで測定を開始します。

コマンド

```
:INITiate[:IMMediate]
```

使用例

現在の測定モードで測定を開始する
INIT

2.4.2 Trigger Switch

:TRIGger[:SEQuence][:STATe] OFF|ON|0|1

Trigger Switch

機能

トリガ待ちの On・Off を設定します。

コマンド

```
:TRIGger[:SEQuence][:STATe] <switch>
```

パラメータ

<switch>	トリガ待ちの On・Off
OFF 0	Off (初期値)
ON 1	On

使用例

トリガ待ちに設定する
TRIG ON

:TRIGger[:SEQuence][:STATe]?

Trigger Switch Query

機能

トリガ待ちの On・Off を読み出します。

クエリ

```
:TRIGger[:SEQuence][:STATe]?
```

レスポンス

```
<switch>
```

パラメータ

<switch>	トリガ待ちの On・Off
0	Off
1	On

使用例

トリガ待ち設定を読み出す
TRIG?
> 0

2.4.3 Trigger Source

:TRIGger[:SEQuence]:SOURce EXTernal[1]| SG

Trigger Source

機能

トリガ信号源を選択します。

コマンド

:TRIGger[:SEQuence]:SOURce <mode>

パラメータ

<mode>	トリガ信号源
EXTernal[1]	外部入力 (External) (初期値)
SG	SG マーカ (SG Marker)

詳細

SG マーカはオプション 020 を搭載時のみ選択できます。

使用例

トリガ信号源を外部入力に設定する

TRIG:SOUR EXT

:TRIGger[:SEQuence]:SOURce?

Trigger Source Query

機能

トリガ信号源を読み出します。

クエリ

```
:TRIGger[:SEQuence]:SOURce?
```

レスポンス

```
<mode>
```

パラメータ

<mode>	トリガ信号源
EXT	外部入力 (External)
SG	SG マーカ (SG Marker)

詳細

SG マーカはオプション 020 を搭載時のみ返します。

使用例

```
トリガ信号源を読み出す  
TRIG:SOUR?  
> EXT
```

2.4.4 Trigger Slope

:TRIGger[:SEQuence]:SLOPe POSitive|NEGative

Trigger Slope

機能

トリガの検出方法（立ち上がり・立ち下がり）を設定します。

コマンド

```
:TRIGger[:SEQuence]:SLOPe <mode>
```

パラメータ

<mode>	トリガの検出方法
POSitive	立ち上がりのエッジで検出する（初期値）
NEGative	立ち下がりエッジで検出する

使用例

トリガの立ち上がりで検出する
TRIG:SLOP POS

:TRIGger[:SEQuence]:SLOPe?

Trigger Slope Query

機能

トリガの検出方法（立ち上がり・立ち下がり）を読み出します。

クエリ

```
:TRIGger[:SEQuence]:SLOPe?
```

レスポンス

```
<mode>
```

パラメータ

<mode>	トリガの検出方法
POS	立ち上がりのエッジで検出する
NEG	立ち下がりエッジで検出する

使用例

トリガの検出方法を読み出す
TRIG:SLOP?
> POS

2.4.5 Trigger Delay

:TRIGger[:SEQuence]:DELay <time>

Trigger Delay

機能

トリガ発生点からキャプチャを開始するまでの遅延時間を設定します。

コマンド

```
:TRIGger[:SEQuence]:DELay <time>
```

パラメータ

<time>	トリガ発生点からキャプチャ開始までの遅延時間
範囲	-2~2 seconds
分解能	50 nanoseconds
サフィックスコード	NS, US, MS, S 省略した場合は second として扱われます。
初期値	0 seconds

使用例

トリガ遅延時間を 20 ms に設定する
TRIG:DEL 20MS

:TRIGger[:SEQuence]:DELay?

Trigger Delay Query

機能

トリガ発生点からキャプチャを開始するまでの遅延時間を読み出します。

クエリ

```
:TRIGger[:SEQuence]:DELay?
```

レスポンス

```
<time>
```

パラメータ

<time>	トリガ発生点からキャプチャ開始までの遅延時間
範囲	-2~2 seconds
分解能	50 nanoseconds second 単位の値を返します。

使用例

トリガ遅延時間を読み出す
TRIG:DEL?
> 0.02000000

2.5 Modulation 測定機能

この節では、Modulation 測定に関するデバイスメッセージについて説明します。

Modulation 測定の結果読み出しに関するデバイスメッセージは表 2.5-1 のとおりです。

表 2.5-1 Modulation 測定機能

機能	デバイスメッセージ
Fetch	:FETCh:MER[n]?

Modulation 測定でのパラメータ設定に関するデバイスメッセージは表 2.5-2 のとおりです。

表 2.5-2 Modulation 測定のパラメータの設定

パラメータ	デバイスメッセージ
Storage Mode	[:SENSe] :MER:AVERage [:STATe] OFF ON AMAXimum 0 1 2
	[:SENSe] :MER:AVERage [:STATe] ?
Storage Count	[:SENSe] :MER:AVERage:COUNT <integer>
	[:SENSe] :MER:AVERage:COUNT?
Graph View Setting	:CALCulate:MER:WINDow2:MODE EACH AVERage
	:CALCulate:MER:WINDow2:MODE?

2.5.1 Measure

:FETCh:MER[n]?

Modulation Query

機能

Modulation 測定の結果を読み出します。

クエリ

:FETCh:MER[n]?

レスポンス

表 2.5-3 を参照してください。

詳細

未測定またはエラーの場合には、“-999.0”を返します。ただし、Frequency Error の場合は“999999999999”を返します。

使用例

Modulation 測定の結果を読み出す
FETC:MER? 1

クエリのパラメータ n に対するレスポンスは表 2.5-3 のとおりです。

表 2.5-3 Modulation 測定結果のレスポンス

n	Result Mode	レスポンス
1 または省略	A/B	次の順にコンマ (,) 区切りで返します。 1. Frequency Error (Average) Hz 単位 2. Frequency Error (Maximum) Hz 単位 3. Frequency Error PPM (Average) ppm 単位 4. Frequency Error PPM (Maximum) ppm 単位 5. Output Power (Average) dBm 単位 6. Output Power (Maximum) dBm 単位 7. Transmitter Timing (Average) ns 単位 8. Transmitter Timing (Maximum) ns 単位
2	A/B	次の順にコンマ (,) 区切りで返します。 1. Carrier Suppression (Average) dB 単位 2. Carrier Suppression (Maximum) dB 単位 3. MER DATA (Average) dB 単位 4. MER DATA (Maximum) dB 単位 5. MER TDM1 (Average) dB 単位 6. MER TDM1 (Maximum) dB 単位 7. MER WIC (Average) dB 単位 8. MER WIC (Maximum) dB 単位 9. MER LIC (Average) dB 単位 10. MER LIC (Maximum) dB 単位 11. MER TDM2 (Average) dB 単位 12. MER TDM2 (Maximum) dB 単位 13. MER PPC (Average) dB 単位 14. MER PPC (Maximum) dB 単位 15. MER SPC (Average) dB 単位 16. MER SPC (Maximum) dB 単位 17. MER Total (Average) dB 単位 18. MER Total (Maximum) dB 単位 19. MER DATA & OIS Channel (Average) dB 単位 20. MER DATA & OIS Channel (Maximum) dB 単位 21. MER Pilot (Average) dB 単位 22. MER Pilot (Maximum) dB 単位
4	A/B	MER (rms) vs Subcarrier グラフの表示データを次の順にコンマ (,) 区切りで返します。 1. サブキャリア 0 の MER (rms) 2. サブキャリア 1 の MER (rms) ... N. N-1 番目サブキャリアの MER (rms) N: FFT Size で決まるサブキャリア数

Result Mode の詳細は、『MS2690A/MS2691A/MS2692A シグナルアナライザ取扱説明書 (本体 リモート制御編) 』の“:SYSTem:RESult:MODE”を参照してください。

2.5.2 Storage Mode

`[[:SENSe]:MER:AVERage[:STATe] OFF|ON|AMAXimum|0|1|2`

Storage Mode

機能

Storage Mode を設定します。

コマンド

`[[:SENSe]:MER:AVERage[:STATe] <mode>`

パラメータ

<mode>	Storage Mode
OFF 0	Off (初期値)
ON 1	Average
AMAXimum 2	Average & Max

使用例

Storage Mode を Average に設定する
`MER:AVER ON`

`[[:SENSe]:MER:AVERage[:STATe]?`

Storage Mode Query

機能

Storage Mode の設定を読み出します。

クエリ

`[[:SENSe]:MER:AVERage[:STATe]?`

レスポンス

<mode>

パラメータ

<mode>	Storage Mode
0	Off
1	Average
2	Average & Max

使用例

Storage Mode の設定を読み出す
`MER:AVER?`
> 1

2.5.3 Storage Count

`[[:SENSe]:MER:AVERage:COUNT <integer>`

Storage Count

機能

Storage Count を設定します。

コマンド

`[[:SENSe]:MER:AVERage:COUNT <integer>`

パラメータ

<code><integer></code>	Storage Count
範囲	2~9999
分解能	1
初期値	10

使用例

Storage Count を 10 に設定する
`MER:AVER:COUN 10`

`[[:SENSe]:MER:AVERage:COUNT?`

Storage Count Query

機能

Storage Count を読み出します。

クエリ

`[[:SENSe]:MER:AVERage:COUNT?`

レスポンス

`<integer>`

パラメータ

<code><integer></code>	Storage Count
範囲	2~9999
分解能	1

使用例

Storage Count を読み出す
`MER:AVER:COUN?`
`> 10`

2.5.4 Graph View Setting

:CALCulate:MER:WINDow2:MODE EACH|AVERage

MER vs Subcarrier View

機能

MER vs Subcarrier グラフに表示する結果の種類を設定します。

コマンド

```
:CALCulate:MER:WINDow2:MODE <mode>
```

パラメータ

<mode>	表示結果の種類
EACH	Symbol Number で指定されたシンボルに対する MER
AVERage	表示範囲のシンボル間で平均化された MER (初期値)

使用例

Symbol Number で指定されたシンボルに対する MER を表示する
CALC:MER:WIND2:MODE EACH

:CALCulate:MER:WINDow2:MODE?

MER vs Subcarrier View Query

機能

MER vs Subcarrier グラフに表示する結果の種類を読み出します。

クエリ

```
:CALCulate:MER:WINDow2:MODE?
```

レスポンス

```
<mode>
```

パラメータ

<mode>	表示結果の種類
EACH	Symbol Number で指定されたシンボルに対する MER
AVER	表示範囲のシンボル間で平均化された MER

使用例

MER vs Subcarrier グラフに表示する結果の種類を読み出す
CALC:MER:WIND2:MODE?
> EACH

2.6 測定結果の保存機能

測定結果を保存するデバイスメッセージは表 2.6-1 のとおりです。

表 2.6-1 測定結果保存機能

機能	デバイスメッセージ
Save All Results	:MMEMory:STORe:RESult [<filename>[,<device>]]
Save as Type	:MMEMory:STORe:RESult:MODE XML CSV
	:MMEMory:STORe:RESult:MODE?

:MMEMory:STORe:RESult [<filename>[,<device>]]

Save All Results Data

機能

測定結果をファイルに保存します。

コマンド

```
:MMEMory:STORe:RESult [<filename>[,<device>]]
```

パラメータ

<filename>	<p>対象ファイル名 ダブルコーテーション(" ")またはシングルクォーテーション(' ')で囲まれた 32 文字以内の文字列 以下の文字は使用できません。 ¥ / : * ? " ' \ < > 省略時のファイル名は“MEDIAFLO 日付_連番.xml” となります。 MEDIAFLO20080617_00.xml</p>
<device>	<p>ドライブ名 A, B, D, E, F, ... 省略時は D ドライブとなります。</p>

詳細

ファイル名省略時にファイル名付加される連番は、00～99 までです。99 まで使用している場合はそれ以上のファイルの保存はできません。

保存したファイルは、指定したドライブの以下のディレクトリにあります。

```
¥Anritsu Corporation¥Signal Analyzer¥User Data¥Measurement Results¥MediaFLO
```

フォルダ内のファイル数の上限は 100 ファイルです。

使用例

“TEST”というファイル名で測定結果を本器内蔵ハードディスクに保存する
 MMEM:STOR:RES "TEST",D

:MMEMory:STORe:RESult:MODE XML|CSV

Save as Type

機能

保存ファイルの種類を設定します。

コマンド`:MMEMory:STORe:RESult:MODE <mode>`**パラメータ**

<code><mode></code>	ファイルの種類
XML	xml 形式 (初期値)
CSV	csv 形式

使用例

保存ファイルの種類を csv 形式に設定する
`MMEM:STOR:RES:MODE CSV`

:MMEMory:STORe:RESult:MODE?

Save as Type Query

機能

保存ファイルの種類を読み出します。

クエリ`:MMEMory:STORe:RESult:MODE?`**レスポンス**`<mode>`**パラメータ**

<code><mode></code>	ファイルの種類
XML	xml 形式
CSV	csv 形式

使用例

保存ファイルの種類を読み出す
`MMEM:STOR:RES:MODE?`
`> CSV`

第3章 SCPI ステータスレジスタ

この章では、アプリケーションの状態を読み出すための SCPI コマンドとステータスレジスタについて説明します。

3.1	測定状態の読み出し.....	3-2
	:STATus:ERRor?	3-2
3.2	STATus:QUEStionable レジスタ	3-3
	:STATus:QUEStionable[:EVENT]?	3-5
	:STATus:QUEStionable:CONDition?	3-5
	:STATus:QUEStionable:ENABle <integer>	3-6
	:STATus:QUEStionable:ENABle?	3-6
	:STATus:QUEStionable:NTRansition <integer>	3-7
	:STATus:QUEStionable:NTRansition?	3-7
	:STATus:QUEStionable:PTRansition <integer>	3-8
	:STATus:QUEStionable:PTRansition?	3-8
	:STATus:QUEStionable:MEASure[:EVENT]?	3-9
	:STATus:QUEStionable:MEASure:CONDition?	3-9
	:STATus:QUEStionable:MEASure:ENABle <integer>	3-10
	:STATus:QUEStionable:MEASure:ENABle?	3-10
	:STATus:QUEStionable:MEASure:NTRansition <integer>	3-11
	:STATus:QUEStionable:MEASure:NTRansition?	3-11
	:STATus:QUEStionable:MEASure:PTRansition <integer>	3-12
	:STATus:QUEStionable:MEASure:PTRansition?	3-12
3.3	STATus:OPERation レジスタ	3-13
	:STATus:OPERation[:EVENT]?	3-14
	:STATus:OPERation:CONDition?	3-14
	:STATus:OPERation:ENABle <integer>	3-15
	:STATus:OPERation:ENABle?	3-15
	:STATus:OPERation:NTRansition <integer>	3-16
	:STATus:OPERation:NTRansition?	3-16
	:STATus:OPERation:PTRansition <integer>	3-17
	:STATus:OPERation:PTRansition?	3-17

3.1 測定状態の読み出し

:STATus:ERRor?

Measurement Status Query

機能

測定状態を読み出します。

クエリ

:STATus:ERRor?

レスポンス

<status>

パラメータ

<status> 測定状態
 値 = bit0 + bit1 + bit2 + bit3 + bit4 + bit5 + bit6
 + bit7 + bit8 + bit9 + bit10 + bit11 + bit12
 + bit13 + bit14 + bit15

bit0 : $2^0 = 1$	未測定
bit1 : $2^1 = 2$	レベルオーバー
bit2 : $2^2 = 4$	(未使用)
bit3 : $2^3 = 8$	(未使用)
bit4 : $2^4 = 16$	(未使用)
bit5 : $2^5 = 32$	(未使用)
bit6 : $2^6 = 64$	(未使用)
bit7 : $2^7 = 128$	(未使用)
bit8 : $2^8 = 256$	(未使用)
bit9 : $2^9 = 512$	(未使用)
bit10 : $2^{10} = 1024$	(未使用)
bit11 : $2^{11} = 2048$	(未使用)
bit12 : $2^{12} = 4096$	(未使用)
bit13 : $2^{13} = 8192$	(未使用)
bit14 : $2^{14} = 16384$	(未使用)
bit15 : $2^{15} = 32768$	(未使用)

範囲 0~255

詳細

正常終了時は 0 が返ります。

使用例

測定状態を読み出す
 STAT:ERR?
 > 0

3.2 STATUS:QUESTIONABLE レジスタ

QUESTIONABLE ステータスレジスタの階層構造は、図 3.2-1、表 3.2-1、図 3.2-2、表 3.2-2 のとおりです。

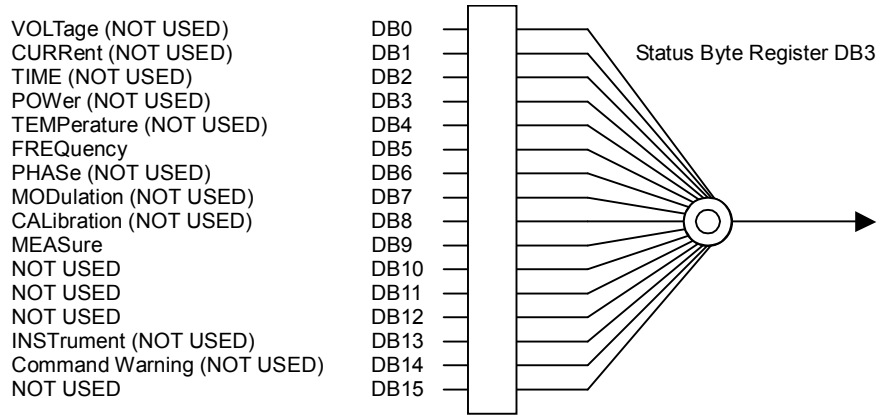


図 3.2-1 QUESTIONABLE ステータスレジスタ

表 3.2-1 QUESTIONABLE ステータスレジスタのビット定義

ビット	定義
DB5	Reference Clock の Unlock
DB9	QUESTIONABLE Measure レジスタサマリ

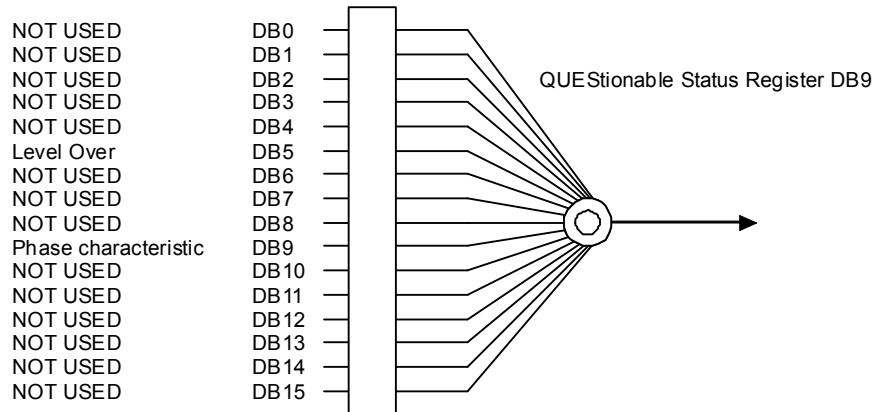


図 3.2-2 QUESTIONABLE Measure レジスタ

表 3.2-2 QUESTIONABLE Measure レジスタのビット定義

ビット	定義
DB5	レベルオーバー
DB9	位相特性がばらついています

QUESTIONable ステータスレジスタに関するデバイスメッセージは表 3.3-3 のとおりです。

表 3.3-3 QUESTIONable ステータスレジスタに関するデバイスメッセージ

機能	デバイスメッセージ
Questionable Status Register Event	:STATus:QUESTIONable[:EVENT]?
Questionable Status Register Condition	:STATus:QUESTIONable:CONDition?
Questionable Status Register Enable	:STATus:QUESTIONable:ENABle <integer>
	:STATus:QUESTIONable:ENABle?
Questionable Status Register Negative Transition	:STATus:QUESTIONable:NTRansition <integer>
	:STATus:QUESTIONable:NTRansition?
Questionable Status Register Positive Transition	:STATus:QUESTIONable:PTRansition <integer>
	:STATus:QUESTIONable:PTRansition?
Questionable Measure Register Event	:STATus:QUESTIONable:MEASure[:EVENT]?
Questionable Measure Register Condition	:STATus:QUESTIONable:MEASure:CONDition?
Questionable Measure Register Enable	:STATus:QUESTIONable:MEASure:ENABle <integer>
	:STATus:QUESTIONable:MEASure:ENABle?
Questionable Measure Register Negative Transition	:STATus:QUESTIONable:MEASure:NTRansition <integer>
	:STATus:QUESTIONable:MEASure:NTRansition?
Questionable Measure Register Positive Transition	:STATus:QUESTIONable:MEASure:PTRansition <integer>
	:STATus:QUESTIONable:MEASure:PTRansition?

:STATus:QUEStionable[:EVENT]?

Questionable Status Register Event

機能

QUEStionable ステータスレジスタのイベントレジスタを読み出します。

クエリ

:STATus:QUEStionable[:EVENT]?

レスポンス

<integer>

パラメータ

<integer>	イベントレジスタのビット総和
分解能	1
範囲	0~65535

使用例

```

QUEStionable ステータスレジスタのイベントレジスタを読み出す
STAT:QUES?
> 0

```

:STATus:QUEStionable:CONDition?

Questionable Status Register Condition

機能

QUEStionable ステータスレジスタのコンディションレジスタを読み出します。

クエリ

:STATus:QUEStionable:CONDition?

レスポンス

<integer>

パラメータ

<integer>	コンディションレジスタのビット総和
分解能	1
範囲	0~65535

使用例

```

QUEStionable ステータスレジスタのコンディションレジスタを読み出す
STAT:QUES:COND?
> 0

```

:STATus:QUEStionable:ENABle <integer>

Questionable Status Register Enable

機能

QUEStionable ステータスレジスタのイベントイネーブルレジスタを設定します。

コマンド

```
:STATus:QUEStionable:ENABle <integer>
```

パラメータ

<integer>	イベントイネーブルレジスタのビット総和
分解能	1
範囲	0~65535

使用例

QUEStionable ステータスレジスタのイベントイネーブルレジスタに 16 を設定する
STAT:QUES:ENAB 16

:STATus:QUEStionable:ENABle?

Questionable Status Register Enable Query

機能

QUEStionable ステータスレジスタのイベントイネーブルレジスタを読み出します。

クエリ

```
:STATus:QUEStionable:ENABle?
```

レスポンス

```
<integer>
```

パラメータ

<integer>	イベントイネーブルレジスタのビット総和
分解能	1
範囲	0~65535

使用例

QUEStionable ステータスレジスタのイベントイネーブルレジスタを読み出す
STAT:QUES:ENAB?
> 16

:STATus:QUEStionable:NTRansition <integer>

Questionable Status Register Negative Transition

機能

QUEStionable ステータスレジスタのトランジションフィルタ（負方向変化）を設定します。

コマンド

```
:STATus:QUEStionable:NTRansition <integer>
```

パラメータ

<integer>	トランジションフィルタ（負方向変化）のビット総和
分解能	1
範囲	0~65535

使用例

QUEStionable ステータスレジスタのトランジションフィルタ（負方向変化）に 16 を設定する
 STAT:QUES:NTR 16

:STATus:QUEStionable:NTRansition?

Questionable Status Register Negative Transition Query

機能

QUEStionable ステータスレジスタのトランジションフィルタ（負方向変化）を読み出します。

クエリ

```
:STATus:QUEStionable:NTRansition?
```

レスポンス

```
<integer>
```

パラメータ

<integer>	トランジションフィルタ（負方向変化）のビット総和
分解能	1
範囲	0~65535

使用例

QUEStionable ステータスレジスタのトランジションフィルタ（負方向変化）を読み出す
 STAT:QUES:NTR?
 > 16

:STATus:QUEStionable:PTRansition <integer>

Questionable Status Register Positive Transition

機能

QUEStionable ステータスレジスタのトランジションフィルタ（正方向変化）を設定します。

コマンド

```
:STATus:QUEStionable:PTRansition <integer>
```

パラメータ

<integer>	トランジションフィルタ（正方向変化）のビット総和
分解能	1
範囲	0~65535

使用例

QUEStionable ステータスレジスタのトランジションフィルタ（正方向変化）に 16 を設定する
STAT:QUES:PTR 16

:STATus:QUEStionable:PTRansition?

Questionable Status Register Positive Transition Query

機能

QUEStionable ステータスレジスタのトランジションフィルタ（正方向変化）を読み出します。

クエリ

```
:STATus:QUEStionable:PTRansition?
```

レスポンス

```
<integer>
```

パラメータ

<integer>	トランジションフィルタ（正方向変化）のビット総和
分解能	1
範囲	0~65535

使用例

QUEStionable ステータスレジスタのトランジションフィルタ（正方向変化）を読み出す
STAT:QUES:PTR?
> 16

:STATus:QUEStionable:MEASure[:EVENT]?

Questionable Measure Register Event

機能

QUEStionable Measure レジスタのイベントレジスタを読み出します。

クエリ

`:STATus:QUEStionable:MEASure[:EVENT]?`

レスポンス

`<integer>`

パラメータ

<code><integer></code>	イベントレジスタのビット総和
分解能	1
範囲	0~65535

使用例

QUEStionable Measure レジスタのイベントレジスタの内容を読み出す

```
STAT:QUES:MEAS?
> 0
```

:STATus:QUEStionable:MEASure:CONDition?

Questionable Measure Register Condition

機能

QUEStionable Measure レジスタのコンディションレジスタを読み出します。

クエリ

`:STATus:QUEStionable:MEASure:CONDition?`

レスポンス

`<integer>`

パラメータ

<code><integer></code>	コンディションレジスタのビット総和
分解能	1
範囲	0~65535

使用例

QUEStionable Measure レジスタのコンディションレジスタの内容を読み出す

```
STAT:QUES:MEAS:COND?
> 0
```

:STATus:QUEStionable:MEASure:ENABle <integer>

Questionable Measure Register Enable

機能

QUEStionable Measure レジスタのイベントイネーブルレジスタを設定します。

コマンド

```
:STATus:QUEStionable:MEASure:ENABle <integer>
```

パラメータ

<integer>	イベントイネーブルレジスタのビット総和
分解能	1
範囲	0~65535

使用例

QUEStionable Measure レジスタのイベントイネーブルレジスタに 16 を設定する
STAT:QUES:MEAS:ENAB 16

:STATus:QUEStionable:MEASure:ENABle?

Questionable Measure Register Enable Query

機能

QUEStionable Measure レジスタのイベントイネーブルレジスタを読み出します。

クエリ

```
:STATus:QUEStionable:MEASure:ENABle?
```

レスポンス

```
<integer>
```

パラメータ

<integer>	イベントイネーブルレジスタのビット総和
分解能	1
範囲	0~65535

使用例

QUEStionable Measure レジスタのイベントイネーブルレジスタを読み出す
STAT:QUES:MEAS:ENAB?
> 16

:STATus:QUEStionable:MEASure:NTRansition <integer>

Questionable Measure Register Negative Transition

機能

QUEStionable Measure レジスタのトランジションフィルタ（負方向変化）を設定します。

コマンド

```
:STATus:QUEStionable:MEASure:NTRansition <integer>
```

パラメータ

<integer>	トランジションフィルタ（負方向変化）のビット総和
分解能	1
範囲	0~65535

使用例

QUEStionable Measure レジスタのトランジションフィルタ（負方向変化）に 16 を設定する

```
STAT:QUES:MEAS:NTR 16
```

:STATus:QUEStionable:MEASure:NTRansition?

Questionable Measure Register Negative Transition Query

機能

QUEStionable Measure レジスタのトランジションフィルタ（負方向変化）を読み出します。

クエリ

```
:STATus:QUEStionable:MEASure:NTRansition?
```

レスポンス

```
<integer>
```

パラメータ

<integer>	トランジションフィルタ（負方向変化）のビット総和
分解能	1
範囲	0~65535

使用例

QUEStionable Measure レジスタのトランジションフィルタ（負方向変化）を読み出す

```
STAT:QUES:MEAS:NTR?
```

```
> 16
```

:STATus:QUEStionable:MEASure:PTRansition <integer>

Questionable Measure Register Positive Transition

機能

QUEStionable Measure レジスタのトランジションフィルタ（正方向変化）を設定します。

コマンド

```
:STATus:QUEStionable:MEASure:PTRansition <integer>
```

パラメータ

<integer>	トランジションフィルタ（正方向変化）のビット総和
分解能	1
範囲	0～65535

使用例

QUEStionable Measure レジスタのトランジションフィルタ（正方向変化）に 16 を設定する

```
STAT:QUES:MEAS:PTR 16
```

:STATus:QUEStionable:MEASure:PTRansition?

Questionable Measure Register Positive Transition Query

機能

QUEStionable Measure レジスタのトランジションフィルタ（正方向変化）を読み出します。

クエリ

```
:STATus:QUEStionable:MEASure:PTRansition?
```

レスポンス

```
<integer>
```

パラメータ

<integer>	トランジションフィルタ（正方向変化）のビット総和
分解能	1
範囲	0～65535

使用例

QUEStionable Measure レジスタのトランジションフィルタ（正方向変化）を読み出す

```
STAT:QUES:MEAS:PTR?
```

```
> 16
```

3.3 STATus:OPERation レジスタ

OPERation ステータスレジスタの階層構造は図 3.3-1, 表 3.3-1 のとおりです。

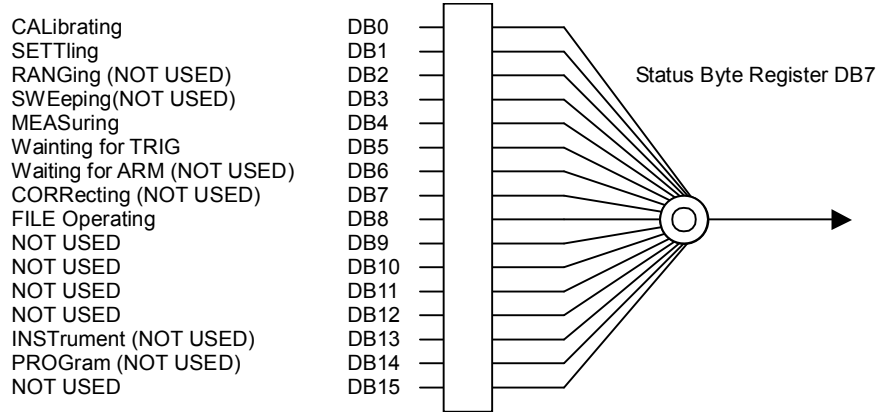


図 3.3-1 OPERation ステータスレジスタ

表 3.3-1 OPERation ステータスレジスタの定義

ビット	定義
DB0	CAL 実行中
DB1	ウォームアップメッセージ表示中
DB4	測定中 (トリガ待ち含む, Continuous 中は常に 1 となります)
DB5	トリガ待ち中
DB8	ファイル操作中

OPERation ステータスレジスタに関するデバイスメッセージは表 3.3-2 のとおりです。

表 3.3-2 OPERation ステータスレジスタに関するデバイスメッセージ

機能	デバイスメッセージ
Operation Status Register Event	:STATus:OPERation[:EVENT]?
Operation Status Register Condition	:STATus:OPERation:CONDition?
Operation Status Register Enable	:STATus:OPERation:ENABLE <integer>
	:STATus:OPERation:ENABLE?
Operation Status Register Negative Transition	:STATus:OPERation:NTRansition <integer>
	:STATus:OPERation:NTRansition?
Operation Status Register Positive Transition	:STATus:OPERation:PTRansition <integer>
	:STATus:OPERation:PTRansition?

:STATus:OPERation[:EVENT]?

Operation Status Register Event

機能

OPERation ステータスレジスタのイベントレジスタを読み出します。

クエリ

:STATus:OPERation[:EVENT]?

レスポンス

<integer>

パラメータ

<integer> イベントレジスタのビット総和
分解能 1
範囲 0~65535

使用例

OPERation ステータスレジスタのイベントレジスタを読み出す
STAT:OPER?
> 0

:STATus:OPERation:CONDition?

Operation Status Register Condition

機能

OPERation ステータスレジスタのコンディションレジスタを読み出します。

クエリ

:STATus:OPERation:CONDition?

レスポンス

<integer>

パラメータ

<integer> コンディションレジスタのビット総和
分解能 1
範囲 0~65535

使用例

OPERation ステータスレジスタのコンディションレジスタを読み出す
STAT:OPER:COND?
> 0

:STATus:OPERation:ENABle <integer>

Operation Status Register Enable

機能

OPERation ステータスレジスタのイベントイネーブルレジスタを設定します。

コマンド

`:STATus:OPERation:ENABle <integer>`

パラメータ

<integer>	イベントイネーブルレジスタのビット総和
分解能	1
範囲	0~65535

使用例

OPERation ステータスレジスタのイベントイネーブルレジスタに 16 を設定する
 STAT:OPER:ENAB 16

:STATus:OPERation:ENABle?

Operation Status Register Enable Query

機能

OPERation ステータスレジスタのイベントイネーブルレジスタを読み出します。

クエリ

`:STATus:OPERation:ENABle?`

レスポンス

<integer>

パラメータ

<integer>	イベントイネーブルレジスタのビット総和
分解能	1
範囲	0~65535

使用例

OPERation ステータスレジスタのイベントイネーブルレジスタを読み出す
 STAT:OPER:ENAB?
 > 16

:STATus:OPERation:NTRansition <integer>

Operation Status Register Negative Transition

機能

OPERation ステータスレジスタのトランジションフィルタ（負方向変化）を設定します。

コマンド

```
:STATus:OPERation:NTRansition <integer>
```

パラメータ

<integer>	トランジションフィルタ（負方向変化）のビット総和
分解能	1
範囲	0~65535

使用例

OPERation ステータスレジスタのトランジションフィルタ（負方向変化）に 16 を設定する
STAT:OPER:NTR 16

:STATus:OPERation:NTRansition?

Operation Status Register Negative Transition Query

機能

OPERation ステータスレジスタのトランジションフィルタ（負方向変化）を読み出します。

クエリ

```
:STATus:OPERation:NTRansition?
```

レスポンス

```
<integer>
```

パラメータ

<integer>	トランジションフィルタ（負方向変化）のビット総和
分解能	1
範囲	0~65535

使用例

OPERation ステータスレジスタのトランジションフィルタ（負方向変化）を読み出す
STAT:OPER:NTR?
> 16

:STATus:OPERation:PTRansition <integer>

Operation Status Register Positive Transition

機能

OPERation ステータスレジスタのトランジションフィルタ（正方向変化）を設定します。

コマンド

```
:STATus:OPERation:PTRansition <integer>
```

パラメータ

<integer>	トランジションフィルタ（正方向変化）のビット総和
分解能	1
範囲	0~65535

使用例

OPERation ステータスレジスタのトランジションフィルタ（正方向変化）に 16 を設定する

```
STAT:OPER:PTR 16
```

:STATus:OPERation:PTRansition?

Operation Status Register Positive Transition Query

機能

OPERation ステータスレジスタのトランジションフィルタ（正方向変化）を読み出します。

クエリ

```
:STATus:OPERation:PTRansition?
```

レスポンス

```
<integer>
```

パラメータ

<integer>	トランジションフィルタ（正方向変化）のビット総和
分解能	1
範囲	0~65535

使用例

OPERation ステータスレジスタのトランジションフィルタ（正方向変化）を読み出す

```
STAT:OPER:PTR?
> 16
```

