

MS2690A/MS2691A/MS2692A  
および  
MS2830A/MS2840A/MS2850A  
シグナルアナライザ  
取扱説明書  
位相雑音測定機能  
リモート制御編

第 10 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 および MS2830A/MS2840A/MS2850A シグナルアナライザ 取扱説明書(位相雑音測定機能 操作編)に記載の事項に順じますので、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た上で機器を操作してください。

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があることを示します。



**警告**

回避しなければ、死亡または重傷に至るおそれがある潜在的な危険があることを示します。



**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おそ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があることを示します。

## 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の近くに、または本書に、安全上および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。

MS2690A/MS2691A/MS2692A および MS2830A/MS2840A/MS2850A

シグナルアナライザ

取扱説明書 位相雑音測定機能 リモート制御編

2008年（平成20年）4月1日（初版）

2017年（平成29年）10月30日（第10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 © 2008-2017, ANRITSU CORPORATION

Printed in Japan

## 国外持ち出しに関する注意

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。

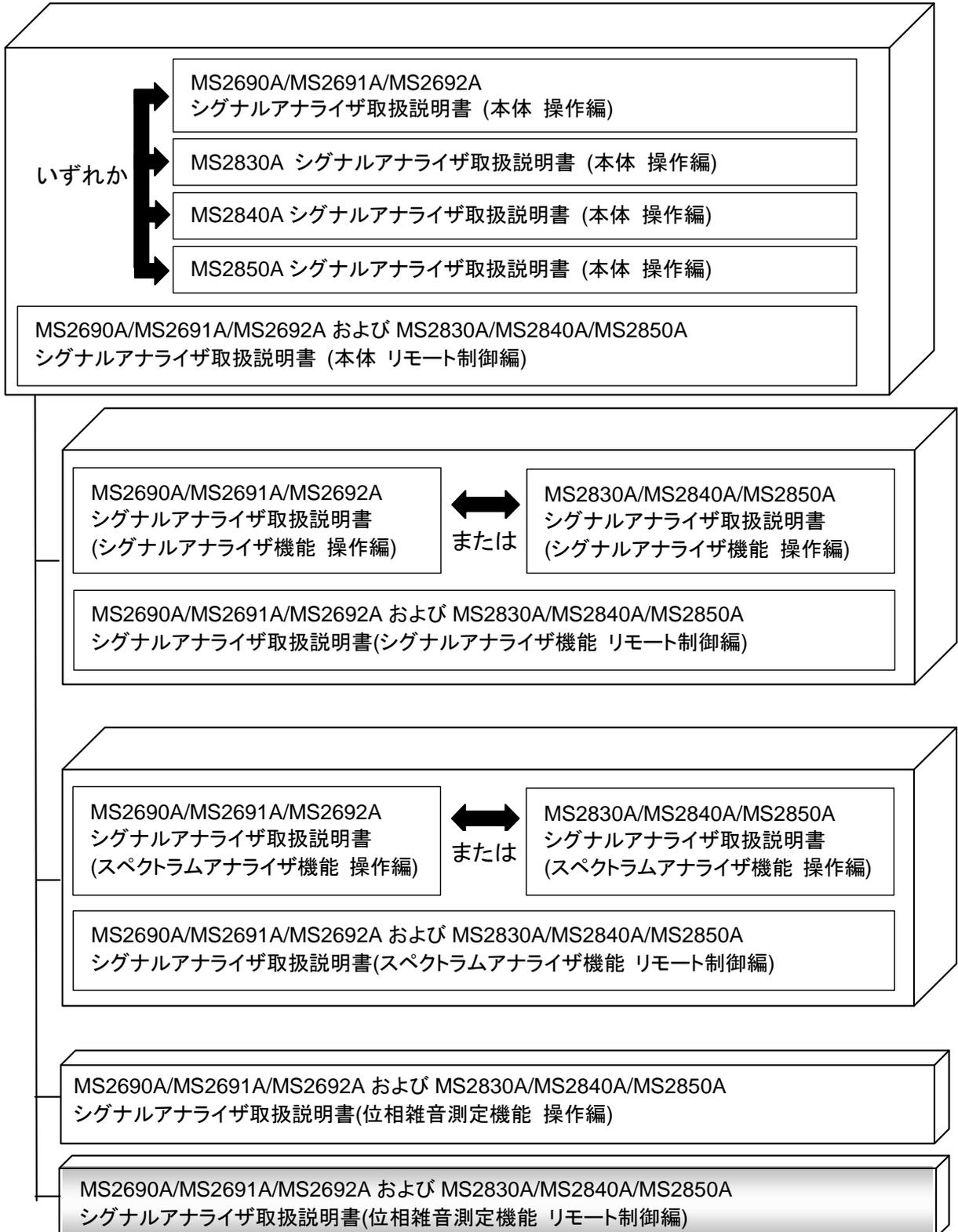
輸出規制を受ける製品やマニュアル類を廃棄処分する場合は、軍事用途等に不正使用されないように、破碎または裁断処理していただきますようお願い致します。



# はじめに

## ■取扱説明書の構成

MS2690A/MS2691A/MS2692A, MS2830A, MS2840A および MS2850A シグナルアナライザの取扱説明書は、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。

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

シグナルアナライザ機能の基本的な操作方法, 機能, リモート制御などについて記述しています。

- シグナルアナライザ 取扱説明書(スペクトラムアナライザ機能 操作編)
- シグナルアナライザ 取扱説明書(スペクトラムアナライザ機能 リモート制御編)

スペクトラムアナライザ機能の基本的な操作方法, 機能, リモート制御などについて記述しています。

- シグナルアナライザ 取扱説明書(位相雑音測定機能 操作編)
- シグナルアナライザ 取扱説明書(位相雑音測定機能 リモート制御編)〈本書〉

位相雑音測定機能の基本的な操作方法, 機能, リモート制御など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 Native モードでの使用について.....	1-3
1.3 数値プログラムデータの設定について .....	1-5
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.....	2-11
2.3 ユーティリティ機能.....	2-31
2.4 共通測定機能 .....	2-35
2.5 ログプロット測定機能.....	2-40
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、位相雑音測定機能(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	インターフェース.....	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	Native モードでの使用について .....	1-3
1.3	数値プログラムデータの設定について .....	1-5

## 1.1 概要

本アプリケーションは、MS2690/MS2691/MS2692A、MS2830A、MS2840A または MS2850A 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 および MS2830A/MS2840A/MS2850A 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、ベクトル信号発生器オプションなど、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 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。

Configuration 画面において、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、コマンドヘッダー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。

Configuration 画面において、コマンド `SYST:LANG NAT` を送信すると、Native モードになります。

#### 注:

Native モードでは、`STATus:QUEStionable` レジスタおよび `STATus:OPERation` レジスタ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変換した場合でも同様です。

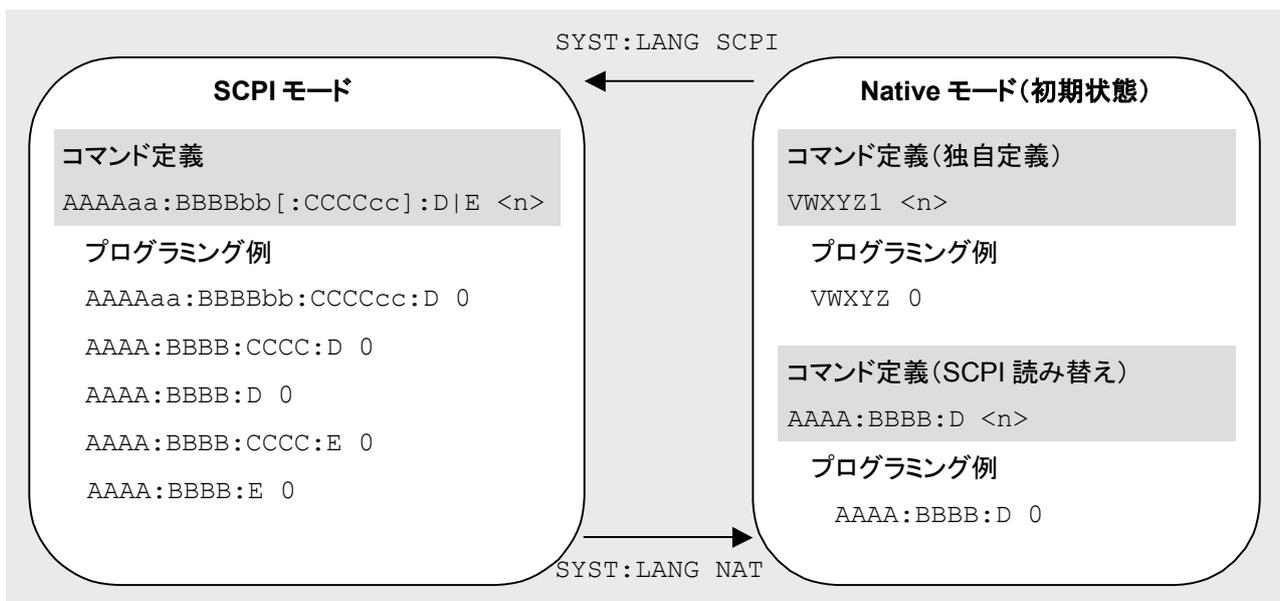


図1.2-1 SCPI モードと Native モード

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ー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

:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel <real> を Native モードのコマンドに読み替える

- ① プログラムヘッダー中の数値パラメータが 1 種類の値しかとらないもので、省略可能なものは省略します。

```
:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel <real>  
↓  
:DISPlay:WINDow:TRACe:Y:[SCALe]:RLEVel <real>
```

- ② 省略できる階層があれば省略します。

```
:DISPlay:WINDow:TRACe:Y[:SCALe]:RLEVel <real>  
↓  
:DISPlay:WINDow:TRACe:Y:RLEVel <real>
```

- 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。

```
:DISPlay:WINDow:TRACe:Y:RLEVel <real>  
↓  
:DISP:WIND:TRAC:Y:RLEV <real>
```

- ④ 先頭の“:”は省略します。

```
:DISP:WIND:TRAC:Y:RLEV <real>  
↓  
DISP:WIND:TRAC:Y:RLEV <real>
```

## 1.3 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_ampl>  
<rel_power>  
<integer>  
<rel_freq>
```



## 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 および MS2830A/MS2840A/MS2850A 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。

2.1	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 PNOISE	2-5
	:SYSTem:APPLIcation:UNLoad PNOISE	2-5
2.1.2	アプリケーションの選択	2-6
	:INSTrument[:SElect] PNOISE SIGANA SPECT CONFIG	2-6
	:INSTrument[:SElect]?	2-7
	:INSTrument:SYSTem PNOISE,[ACTive]  INACTive MINimum	2-8
	:INSTrument:SYSTem? PNOISE	2-9
2.1.3	初期化	2-10
	:INSTrument:DEFault	2-10
	:SYSTem:PRESet	2-10
2.2	基本パラメータの設定	2-11
2.2.1	Carrier Frequency	2-12
	[:SENSe]:FREQuency:CENTer <freq>	2-12
	[:SENSe]:FREQuency:CENTer?	2-12
2.2.2	Start Offset	2-13
	[:SENSe]:FREQuency:OFFSet:STARt <rel_freq>	2-13
	[:SENSe]:FREQuency:OFFSet:STARt?	2-14
2.2.3	Stop Offset	2-15
	[:SENSe]:FREQuency:OFFSet:STOP <rel_freq>	2-15
	[:SENSe]:FREQuency:OFFSet:STOP?	2-16
2.2.4	Reference Level	2-17
	:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel <real>	2-17
	:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel?	2-18
2.2.5	RF Attenuator	2-19
	[:SENSe]:POWer[:RF]:ATTenuation <rel_amp>	2-19
	[:SENSe]:POWer[:RF]:ATTenuation?	2-20
2.2.6	RF Attenuator Auto/Manual	2-21
	[:SENSe]:POWer[:RF]:ATTenuation:AUTO OFF ON 0 1	2-21
	[:SENSe]:POWer[:RF]:ATTenuation:AUTO?	2-21
2.2.7	Level Offset	2-22
	:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel:OFFSet <rel_power>	2-22
	:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel:OFFSet?	2-22
2.2.8	Level Offset State	2-23
	:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel:OFFSet:STATe OFF ON 0 1	2-23
	:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel:OFFSet:STATe?	2-23
2.2.9	Pre Amp	2-24
	[:SENSe]:POWer[:RF]:GAIN[:STATe] OFF ON 0 1	2-24
	[:SENSe]:POWer[:RF]:GAIN[:STATe]?	2-24

2.2.10	Log Scale Line .....	2-25
	:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:LINE <line> .....	2-25
	:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:LINE? .....	2-25
2.2.11	Reference Value .....	2-26
	:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RVALue <real> .....	2-26
	:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RVALue? .....	2-26
2.2.12	External Mixer .....	2-27
	[:SENSe]:MIXer[:STATe] ON OFF 1 0 .....	2-27
	[:SENSe]:MIXer[:STATe]? .....	2-28
	[:SENSe]:MIXer:BAND VHP EHP .....	2-29
	[:SENSe]:MIXer:BAND? .....	2-30
2.3	ユーティリティ機能 .....	2-31
2.3.1	Erase Warm Up Message .....	2-32
	:DISPlay:ANNotation:WUP:ERASe .....	2-32
2.3.2	Display Title .....	2-32
	:DISPlay:ANNotation:TITLe[:STATe] OFF ON 0 1 .....	2-32
	:DISPlay:ANNotation:TITLe[:STATe]? .....	2-33
2.3.3	Title Entry .....	2-33
	:DISPlay:ANNotation:TITLe:DATA <string> .....	2-33
	:DISPlay:ANNotation:TITLe:DATA? .....	2-34
2.4	共通測定機能 .....	2-35
2.4.1	測定と制御 .....	2-36
	:INITiate:CONTInuous OFF ON 0 1 .....	2-36
	:INITiate:CONTInuous? .....	2-36
	:INITiate:MODE:CONTInuous .....	2-37
	:INITiate:MODE:SINGle .....	2-37
	:INITiate[:IMMediate] .....	2-37
	:CONFigure? .....	2-38
2.4.2	Save Result Data .....	2-39
	:MMEMory:STORe:TRACe [<filename>[,<device>]] .....	2-39
2.5	ログプロット測定機能 .....	2-40
2.5.1	Measure .....	2-42
	:CONFigure:LPLot .....	2-42
	:INITiate:LPLot .....	2-42
	:FETCh:LPLot[n]? .....	2-43
	:READ:LPLot[n]? .....	2-44
	:MEASure:LPLot[n]? .....	2-44
2.5.2	Average Count .....	2-45
	[:SENSe]:LPLot:AVERAge:COUNt <integer> .....	2-45
	[:SENSe]:LPLot:AVERAge:COUNt? .....	2-45
2.5.3	Phase Noise Optimization .....	2-46
	[:SENSe]:FREQuency:SYNThesis[:STATe] 0 1 2 3 .....	2-46
	[:SENSe]:FREQuency:SYNThesis[:STATe]? .....	2-47
2.5.4	Marker Mode .....	2-48
	:CALCulate:LPLot:MARKer[1] 2 3 4 5 6 7 8:MODE NORMal INTEgralnoise RMSNoise JITTer RESidualfm OFF .....	2-48

:CALCulate:LPLot:MARKer[1] 2 3 4 5 6 7 8:MODE? .....	2-49
2.5.5 Analysis Width .....	2-50
:CALCulate:LPLot:MARKer[1] 2 3 4 5 6 7 8:WIDTH:START <rel_freq> .....	2-50
:CALCulate:LPLot:MARKer[1] 2 3 4 5 6 7 8:WIDTH:START? .....	2-51
:CALCulate:LPLot:MARKer[1] 2 3 4 5 6 7 8:WIDTH:STOP <rel_freq> .....	2-52
:CALCulate:LPLot:MARKer[1] 2 3 4 5 6 7 8:WIDTH:STOP? .....	2-53
2.5.6 Marker Frequency Offset .....	2-54
:CALCulate:LPLot:MARKer[1] 2 3 4 5 6 7 8:X <rel_freq> .....	2-54
:CALCulate:LPLot:MARKer[1] 2 3 4 5 6 7 8:X? .....	2-55
2.5.7 Marker Result .....	2-56
:CALCulate:LPLot:MARKer[1] 2 3 4 5 6 7 8:Y? .....	2-56
:CALCulate:LPLot:MARKer[1] 2 3 4 5 6 7 8:VALue? .....	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 PNOISE
Unload Application	:SYSTem:APPLication:UNLoad PNOISE
Application Switch	:INSTrument[:SElect] PNOISE
	:INSTrument[:SElect]?
Application Status	:INSTrument:SYSTem PNOISE, [ACTive] INACTive MINimum
	:INSTrument:SYSTem? PNOISE
Initialization	:INSTrument:DEFault
	:SYSTem:PRESet

### 2.1.1 アプリケーションの起動

#### :SYSTem:APPLication:LOAD PNOISE

##### 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 PNOISE
```

###### 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するようになり、Application Switch メニューに登録されます。

###### 使用例

本アプリケーションを起動する  
SYST:APPL:LOAD PNOISE

#### :SYSTem:APPLication:UNLoad PNOISE

##### 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 PNOISE
```

###### 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。

###### 使用例

本アプリケーションを終了する  
SYST:APPL:UNL PNOISE

## 2.1.2 アプリケーションの選択

:INSTrument[:SElect] PNOISE|SIGANA|SPECT|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
PNOISE	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

本機能により、制御対象のアプリケーションが選択されます。

### 使用例

制御対象を本アプリケーションに切り替える

INST PNOISE

**: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**

<code>&lt;apl_name&gt;</code>	アプリケーション
<code>PNOISE</code>	本アプリケーション
<code>SIGANA</code>	シグナルアナライザ
<code>SPECT</code>	スペクトラムアナライザ
<code>CONFIG</code>	Config

**使用例**

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

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

Application 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 <apl\_name>, <window>

パラメータ

<apl_name>	アプリケーション
PNOISE	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	<b>Config</b>
<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する

INST:SYST PNOISE,ACT

**:INSTrument:SYSTem? PNOISE**

## Application 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? PNOISE
```

## 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
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
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? PNOISE
> CURR,ACT
```

### 2.1.3 初期化

#### :INSTrument:DEFault

##### Initialization

##### 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

##### Initialization

##### 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

```
:SYSTem:PRESet
```

##### 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	[ :SENSe ] :FREQuency:CENTer <freq>
	[ :SENSe ] :FREQuency:CENTer?
Frequency Start Offset	[ :SENSe ] :FREQuency:OFFSet:START <rel_freq>
	[ :SENSe ] :FREQuency:OFFSet:START?
Frequency Stop Offset	[ :SENSe ] :FREQuency:OFFSet:STOP <rel_freq>
	[ :SENSe ] :FREQuency:OFFSet:STOP?
Reference Level	:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel <real>
	:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel?
RF Attenuator	[ :SENSe ] :POWer[:RF]:ATTenuation <rel_ampl>
	[ :SENSe ] :POWer[:RF]:ATTenuation?
RF Attenuator Auto/Manual	[ :SENSe ] :POWer[:RF]:ATTenuation:AUTO ON OFF 1 0
	[ :SENSe ] :POWer[:RF]:ATTenuation:AUTO?
Level Offset	:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel:OFFSet <rel_power>
	:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel:OFFSet?
Level Offset State	:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel:OFFSet:STATE OFF ON 0 1
	:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel:OFFSet:STATE?
Pre-Amp State	[ :SENSe ] :POWer[:RF]:GAIN[:STATe] OFF ON 0 1
	[ :SENSe ] :POWer[:RF]:GAIN[:STATe]?
Scale Lines	:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:LINE <line>
	:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:LINE?
Reference Value	:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RVALue <real>
	:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RVALue?
External Mixer Mode	[ :SENSe ] :MIXer[:STATe] ON OFF 1 0
	[ :SENSe ] :MIXer[:STATe]?
External Mixer Band	[ :SENSe ] :MIXer:BAND VHP EHP
	[ :SENSe ] :MIXer:BAND?

## 2.2.1 Carrier Frequency

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

Carrier Frequency

### 機能

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	10 MHz～本体の上限値
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	2.000 GHz

### 使用例

キャリア周波数を 2.000 GHz に設定する  
`FREQ:CENT 2.000GHZ`

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

Carrier Frequency Query

### 機能

設定した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	10 MHz～本体の上限値
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?`  
> 2000000000

## 2.2.2 Start Offset

`[[:SENSe]:FREQuency:OFFSet:STARt <rel_freq>`

Frequency Start Offset

### 機能

オフセット周波数のスタートを設定します。

### コマンド

`[[:SENSe]:FREQuency:OFFSet:STARt <rel_freq>`

### パラメータ

<freq>	スタートオフセット周波数
範囲	10 Hz, 100 Hz, 1 kHz
サフィックスコード	HZ, KHZ
	省略した場合は Hz として扱われます。
初期値	10 Hz

### 使用例

オフセット周波数のスタートを 100 Hz に設定する

`FREQ:OFFS:STAR 100HZ`

## [[:SENSe]:FREQuency:OFFSet:STARt?

Frequency Start Offset Query

### 機能

オフセット周波数のスタートを読み出します。

### コマンド

```
[[:SENSe]:FREQuency:OFFSet:STARt?
```

### レスポンス

```
<freq>
```

### パラメータ

<freq>	スタートオフセット周波数
範囲	10 Hz, 100 Hz, 1 kHz
サフィックスコード	なし, Hz 単位の値を返します。

### 使用例

```
オフセット周波数のスタートを読み出す  
FREQ:OFFS:STAR?  
> 1000
```

### 2.2.3 Stop Offset

[[:SENSe]:FREQuency:OFFSet:STOP <rel\_freq>

Frequency Stop Offset

#### 機能

オフセット周波数のストップを設定します。

#### コマンド

```
[[:SENSe]:FREQuency:OFFSet:STOP <rel_freq>
```

#### パラメータ

<rel_freq>	ストップオフセット周波数
範囲	100 kHz, 1 MHz, 10 MHz
サフィックスコード	HZ, KHZ, MHZ
	省略した場合は Hz として扱われます。
初期値	10 MHz

#### 使用例

オフセット周波数のストップを 1 MHz に設定する

```
FREQ:OFFS:STOP 1MHZ
```

## [[:SENSe]:FREQuency:OFFSet:STOP?

Frequency Stop Offset Query

### 機能

オフセット周波数のストップを読み出します。

### コマンド

```
[[:SENSe]:FREQuency:OFFSet:STOP?
```

### レスポンス

```
<rel_freq>
```

### パラメータ

<rel_freq>	ストップオフセット周波数
範囲	100 kHz, 1 MHz, 10 MHz
サフィックスコード	なし, Hz 単位の値を返します。

### 使用例

```
オフセット周波数のストップを読み出す  
FREQ:OFFS:STOP?  
> 1000000
```

## 2.2.4 Reference Level

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

Reference Level

### 機能

リファレンスレベルを設定します。

### コマンド

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

### パラメータ

<real>	リファレンスレベル
範囲	-120.00+Offset~50.00+Offset (Pre-Amp Off) -120.00+Offset~30.00+Offset (Pre-Amp On)
分解能	0.01 dB
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	0 dBm

### 詳細

MS2830A-008/108/068/168, MS2840A-008/108/068/168/069/169,  
MS2850A-068/168 プリアンプが未搭載のときは, Pre-Amp Off の設定範囲となります。

### 使用例

リファレンスレベルを-15.00 dBm に設定する  
DISP:WIND:TRAC:Y:RLEV -15.00

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

Reference Level Query

機能

リファレンスレベルを読み出します

クエリ

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

レスポンス

<real>

パラメータ

<real>	リファレンスレベル
範囲	-120.00+Offset~50.00+Offset (Pre-Amp Off) -120.00+Offset~30.00+Offset (Pre-Amp On)
分解能	0.01 dB
	dBm 単位の値を返します。

使用例

リファレンスレベルを読み出す  
DISP:WIND:TRAC:Y:RLEV?  
> -15.00

## 2.2.5 RF Attenuator

`[:SENSe]:POWer[:RF]:ATTenuation <rel_ampl>`

RF Attenuator

### 機能

アッテネータを設定します。

### コマンド

`[:SENSe]:POWer[:RF]:ATTenuation <rel_ampl>`

### パラメータ

<code>&lt;rel_ampl&gt;</code>	アッテネータ値
範囲	0~60 dB
分解能	2 dB ステップ
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	10 dB

### 使用例

アッテネータを 10 dB に設定する  
`POW:ATT 10`

## [[:SENSe]:POWer[:RF]:ATTenuation?

RF Attenuator Query

### 機能

アッテネータを読み出します。

### クエリ

```
[[:SENSe]:POWer[:RF]:ATTenuation?
```

### レスポンス

```
<rel_ampl>
```

### パラメータ

<rel_ampl>	アッテネータ値
範囲	0～60 dB
分解能	2 dB
	dB 単位の値を返します。

### 使用例

```
アッテネータ値を読み出す  
POW:ATT?  
> 10
```

## 2.2.6 RF Attenuator Auto/Manual

```
[:SENSe]:POWer[:RF]:ATTenuation:AUTO OFF|ON|0|1
```

RF Attenuator Auto/Manual

### 機能

アッテネータの自動設定を設定します。

### コマンド

```
[:SENSe]:POWer[:RF]:ATTenuation:AUTO <switch>
```

### パラメータ

<switch>	自動設定
0 OFF	自動設定を Off にする
1 ON	自動設定を On にする (初期値)

### 使用例

アッテネータの自動設定を有効にする  
 POW:ATT:AUTO ON

```
[:SENSe]:POWer[:RF]:ATTenuation:AUTO?
```

RF Attenuator Auto/Manual Query

### 機能

アッテネータの自動設定を読み出します。

### コマンド

```
[:SENSe]:POWer[:RF]:ATTenuation:AUTO?
```

### レスポンス

```
<switch>
```

### パラメータ

<switch>	自動設定
0	自動設定が Off
1	自動設定が On

### 使用例

アッテネータの自動設定を読み出す  
 POW:ATT:AUTO?  
 > 1

## 2.2.7 Level Offset

:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RLEVel:OFFSet <rel\_power>

Level Offset Value

### 機能

リファレンス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

リファレンスレベルのオフセット値を 0.5 dB に設定する  
DISP:WIND:TRAC:Y:RLEV:OFFS 0.5

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

Level Offset Value Query

### 機能

リファレンス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

リファレンスレベルのオフセット値を読み出す  
DISP:WIND:TRAC:Y:RLEV:OFFS?  
> 0.50

## 2.2.8 Level Offset State

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

Level Offset State

### 機能

リファレンス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>	リファレンス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

リファレンス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

リファレンス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>	リファレンス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

リファレンス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

## 2.2.9 Pre Amp

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

Pre Amp State

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

### 詳細

MS2830A-008/108/068/168, MS2840A-008/108/068/168/069/169, MS2850A-068/168 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 State 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>&lt;switch&gt;</code>	Pre-Amp の On・Off
<code>0</code>	Off
<code>1</code>	On

### 詳細

MS2830A-008/108/068/168, MS2840A-008/108/068/168/069/169, MS2850A-068/168 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`

## 2.2.10 Log Scale Line

:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:LINE <line>

Scale Lines

### 機能

レベル軸の Log Scale Line を設定します。

### コマンド

```
:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:LINE <line>
```

### パラメータ

<line>	Line の値
範囲	10, 16
初期値	10

### 使用例

Log Scale Line を 16 に設定する  
 DISP:WIND:TRAC:Y:SCAL:LINE 16

:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:LINE?

Scale Lines

### 機能

レベル軸の Log Scale Line を読み出します。

### コマンド

```
:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:LINE?
```

### レスポンス

<line>

### パラメータ

<line>	Line の値
範囲	10, 16

### 使用例

レベル軸の Log Scale Line を読み出す  
 DISP:WIND:TRAC:Y:LINE?  
 > 16

## 2.2.11 Reference Value

:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RVALue <real>

Reference Value

機能

レベル軸の上限値を設定します。

コマンド

:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RVALue <real>

パラメータ

<real>	上限値の値
範囲	-140~-50 (Log Scale Line=10 の場合) -170~-20 (Log Scale Line=16 の場合)
分解能	10 dBc/Hz
サフィックスコード	DB
	dB is used when omitted.
初期値	-50 (Log Scale Line=10 の場合) -20 (Log Scale Line=16 の場合)

使用例

Reference Value を-70 に設定する  
DISP:WIND:TRAC:Y:SCAL:RVAL -70

:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RVALue?

Scale Lines

機能

レベル軸の上限値を読み出します。

コマンド

:DISPlay:WINDow[1]:TRACe:Y[:SCALe]:RVALue?

レスポンス

<real>

パラメータ

<real>	Reference Value
範囲	-140~-50 (Log Scale Line=10 の場合) -170~-20 (Log Scale Line=16 の場合)

使用例

レベル軸の上限値を読み出す  
DISP:WIND:TRAC:Y:RVALue?  
> -140

## 2.2.12 External Mixer

[[:SENSe]:MIXer[:STATe] ON|OFF|1|0

External Mixer Mode

### 機能

内部ミキサと外部ミキサを選択します。

### コマンド

[[:SENSe]:MIXer[:STATe] <switch>

### パラメータ

<switch>	外部ミキサスイッチ
ON 1	外部ミキサを選択します。
OFF 0	内部ミキサを選択します。

### 詳細

MS2830A-044/045 搭載時, 有効です。

MS2840A-044/046 搭載時, 有効です。

MS2850A は Option にかかわらず有効です。

外部ミキサ関連のパラメータ設定はスペクトラムアナライザ機能を使用します。

詳細は『MS2690A/MS2691A/MS2692A および MS2830A/MS2840A/MS2850A  
スペクトラムアナライザ 取扱説明書 (スペクトラムアナライザ機能 リモート編)』  
「2.20 外部ミキサの設定」を参照してください。

### 使用例

外部ミキサを使用する  
MIX ON

## [:SENSe]:MIXer[:STATe]?

External Mixer Mode Query

### 機能

現在のミキサモードを読み出します。

### クエリ

[:SENSe]:MIXer[:STATe]?

### レスポンス

<status>

### パラメータ

<status>	外部ミキサスイッチ
1	外部ミキサを使用します。
0	内部ミキサを使用します。

### 詳細

MS2830A-044/045 搭載時, 有効です。

MS2840A-044/046 搭載時, 有効です。

MS2850A は Option にかかわらず有効です。

外部ミキサ関連のパラメータ設定はスペクトラムアナライザ機能を使用します。

詳細は『MS2690A/MS2691A/MS2692A および MS2830A/MS2840A/MS2850A  
スペクトラムアナライザ 取扱説明書 (スペクトラムアナライザ機能 リモート編)』

「2.20 外部ミキサの設定」を参照してください。

### 使用例

現在の外部ミキサスイッチ状態を読み出す

MIX?

> 1

## [:SENSe]:MIXer:BAND VHP|EHP

External Mixer Band

## 機能

高性能導波管ミキサのバンドを選択します。

## コマンド

[:SENSe]:MIXer:BAND &lt;band&gt;

## パラメータ

<band>	高性能導波管ミキサバンド
VHP	Band VHP (50.0~75.0 GHz, 8+) (初期値)
EHP	Band EHP (60.0~90.0 GHz, 12-)

## 詳細

MS2830A-044/045 搭載時, 有効です。

MS2840A-044/046 搭載時, 有効です。

MS2850A は Option にかかわらず有効です。

高性能導波管ミキサを使用する場合, 本体スペクトラムアナライザ機能を使用して外部ミキサのパラメータを設定します。

詳細は『MS2690A/MS2691A/MS2692A および MS2830A/MS2840A/MS2850A  
 スペクトラムアナライザ 取扱説明書 (スペクトラムアナライザ機能 リモート編)』  
 「2.20 外部ミキサの設定」を参照してください。

## 使用例

高性能導波管ミキサのバンドを Band EHP にする  
 MIX:BAND EHP

## [[:SENSe]:MIXer:BAND?

External Mixer Band Query

### 機能

現在の高性能導波管ミキサのバンドを読み出します。

### クエリ

```
[[:SENSe]:MIXer:BAND?
```

### レスポンス

```
<band>
```

### パラメータ

<band>	高性能導波管ミキサバンド
VHP	Band VHP (50.0~75.0 GHz, 8+)
EHP	Band EHP (60.0~90.0 GHz, 12-)

### 詳細

MS2830A-044/045 搭載時, 有効です。

MS2840A-044/046 搭載時, 有効です。

MS2850A は Option にかかわらず有効です。

### 使用例

現在の外部ミキサバンドを読み出す

```
MIX:BAND?
```

```
> EHP
```

## 2.3 ユーティリティ機能

測定対象のユーティリティ機能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 ユーティリティ機能

機能	デバイスメッセージ
Erase Warm Up Message	:DISPlay:ANNotation:WUP:ERASe
Display Title	:DISPlay:ANNotation:TITLe[:STATe] ON OFF 1 0
	:DISPlay:ANNotation:TITLe[:STATe]?
Title Entry	:DISPlay:ANNotation:TITLe:DATA <string>
	:DISPlay:ANNotation:TITLe:DATA?

### 2.3.1 Erase Warm Up Message

:DISPlay:ANNotation:WUP:ERASe

Erase Warm Up Message

機能

起動直後に表示されるウォームアップメッセージを消去します。

コマンド

:DISPlay:ANNotation:WUP:ERASe

使用例

ウォームアップメッセージを消去する

DISP:ANN:WUP:ERAS

### 2.3.2 Display Title

:DISPlay:ANNotation:TITLe[:STATe] OFF|ON|0|1

Display Title

機能

タイトル表示の On・Off を設定します。

コマンド

:DISPlay:ANNotation:TITLe[:STATe] <switch>

パラメータ

<switch>	タイトル表示の On・Off
OFF 0	Off
ON 1	On (初期値)

使用例

タイトルを表示する

DISP:ANN:TITL ON

**:DISPlay:ANNotation:TITLe[:STATe]?**

Display Title Query

**機能**

タイトル表示の On・Off を読み出します。

**クエリ****:DISPlay:ANNotation:TITLe[:STATe]?****レスポンス**

&lt;switch&gt;

**パラメータ**

<switch>	タイトル表示の On・Off
1	On
0	Off

**使用例**

タイトル表示の設定を読み出す  
DISP:ANN:TITL?  
> 1

**2.3.3 Title Entry****:DISPlay:ANNotation:TITLe:DATA <string>**

Title Entry

**機能**

タイトル文字列を設定します。

**コマンド****:DISPlay:ANNotation:TITLe:DATA <string>****パラメータ**

<string>	ダブルコーテーション (“ ”) またはシングルコーテーション (‘ ’) で囲まれた 32 文字以内の文字列
----------	---

**使用例**

タイトル文字列を設定する  
DISP:ANN:TITL:DATA 'TEST'

## :DISPlay:ANNotation:TITLe:DATA?

Title Entry Query

### 機能

タイトル文字列を読み出します。

### クエリ

```
:DISPlay:ANNotation:TITLe:DATA?
```

### レスポンス

```
<string>
```

### パラメータ

```
<string>
```

ダブルコーテーション (“ ”) またはシングルコーテーション (‘ ’) で囲まれた 32 文字以内の文字列

### 使用例

```
タイトル文字列を読み出す  
DISP:ANN:TITL:DATA?  
> TEST
```

## 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]
Configure	:CONFigure?
Save Result Data	:MMEMory:STORe:TRACe [<filename>[,<device>]]

## 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?  
> 0

**: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
```

## 詳細

非同期コマンドです。連続測定中の同期制御には対応していないので注意してください。

## 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]
```

## 詳細

非同期コマンドです。連続測定中の同期制御には対応していないので注意してください。

## 使用例

現在の測定モードで測定を開始する  
INIT

## :CONFigure?

Configure Query

### 機能

現在の測定機能の名前を読み出します。

### クエリ

```
:CONFigure?
```

### レスポンス

```
<mode>
```

### パラメータ

<mode>	測定機能
LPLot	ログプロット測定

### 使用例

```
現在の測定機能を読み出す  
CONF?  
> LPL
```

## 2.4.2 Save Result Data

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

Save Result Data

### 機能

測定結果を CSV ファイルに保存します。

### コマンド

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

### パラメータ

<filename>	<p>保存するファイル名 ダブルコーテーション (“ ”) またはシングルコーテーション ( ‘ ’ ) で囲まれた 32 文字以内の文字列 ( 拡張子は除く ) 。 下記の文字は使用できません。 ¥ / : * ? “ ” \ ‘ &lt; &gt;   省略時のファイル名は “WaveData 日付_連番.csv” となります。</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に付加される連番は、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
  ¥Trace Data¥Phase Noise
```

フォルダ内のファイル数の上限は 1000 ファイルです。

### 使用例

測定結果ファイル “trace” を E ドライブに保存する

```
MMEM:STOR:TRAC "trace",E
```

## 2.5 ログプロット測定機能

この節では、ログプロット測定に関するデバイスメッセージについて説明します。

ログプロット測定の実行、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 ログプロット測定機能

機能	デバイスメッセージ
Configure	:CONFigure:LPLot
Initiate	:INITiate:LPLot
Fetch	:FETCh:LPLot [n] ?
Read/Measure	:READ:LPLot [n] ?
	:MEASure:LPLot [n] ?

表 2.5-1のパラメータ n に対するレスポンスは表 2.5-2のとおりです。

表2.5-2 ログプロット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	Average Count に対する平均値を次の順にコンマ (,) 区切りで返します。 1. Carrier Power[dBm] 2. Carrier Frequency[Hz] 注 設定したキャリア周波数と周波数エラーの測定結果を加算した結果となります 3. -999.0 4. -999.0 5. -999.0 6. 最小オフセット周波数における位相雑音レベル[dBc/Hz] 7. 最大オフセット周波数における位相雑音レベル[dBc/Hz]
2	A/B	ログプロットグラフの表示データ数を返します。
3	A/B	ログプロットグラフの表示データをコンマ (,) 区切りで返します。単位:dBc/Hz

ログプロット測定のパラメータ設定に関するデバイスメッセージは、表 2.5-3のとおりです。

表2.5-3 ログプロット測定のパラメータ設定

パラメータ	デバイスメッセージ
Average Count	<code>[:SENSe]:LPLot:AVERage:COUNT &lt;integer&gt;</code>
	<code>[:SENSe]:LPLot:AVERage:COUNT?</code>
Phase Noise Optimization	<code>[:SENSe]:FREQuency:SYNThesis[:STATe] &lt;integer&gt;</code>
	<code>[:SENSe]:FREQuency:SYNThesis[:STATe]?</code>

ログプロット測定でのマーカの設定・マーカ位置の値の読み出しに関するデバイスメッセージは、表 2.5-4のとおりです。

表2.5-4 ログプロット測定のマーカに関するデバイスメッセージ

パラメータ	デバイスメッセージ
Marker Mode	<code>:CALCulate:LPLot:MARKer[n]:MODE &lt;mode&gt;</code>
	<code>:CALCulate:LPLot:MARKer[n]:MODE?</code>
Marker Width Start	<code>:CALCulate:LPLot:MARKer[n]:WIDTh:START &lt;rel_freq&gt;</code>
	<code>:CALCulate:LPLot:MARKer[n]:WIDTh:START?</code>
Marker Width Stop	<code>:CALCulate:LPLot:MARKer[n]:WIDTh:STOP &lt;rel_freq&gt;</code>
	<code>:CALCulate:LPLot:MARKer[n]:WIDTh:STOP?</code>
Marker Freq. Offset	<code>:CALCulate:LPLot:MARKer[n]:X &lt;rel_freq&gt;</code>
	<code>:CALCulate:LPLot:MARKer[n]:X?</code>
Marker Result	<code>:CALCulate:LPLot:MARKer[n]:Y?</code>
Marker Result Value	<code>:CALCulate:LPLot:MARKer[1] 2 3 4 5 6 7 8:VALue?</code>

## 2.5.1 Measure

### :CONFigure:LPLot

#### Configure

##### 機能

ログプロット測定機能を選択します。測定は実行しません。

##### コマンド

```
:CONFigure:LPLot
```

##### 使用例

ログプロット測定を選択する  
CONF:LPL

### :INITiate:LPLot

#### Initiate

##### 機能

ログプロット測定を実行します。

##### コマンド

```
:INITiate:LPLot
```

##### 使用例

ログプロット測定を実行する  
INIT:LPL

:FETCh:LPLot[n]?

Fetch

機能

ログプロット測定の結果を読み出します。

クエリ

:FETCh:LPLot [n]?

レスポンス

```

<carrier power>,<carrier freq>,
-999.0,-999.0,-999.0,
<level_lower>,<level_upper>
    (n=1 または省略時)
<data_point>
    (n=2)
<level_1>,<level_2>,<level_3>,
.....
    (n=3)

```

パラメータ

<carrier power>	キャリアパワー
<carrier freq>	キャリア周波数
<level_lower>	最小オフセット周波数における位相雑音レベル
<level_upper>	最大オフセット周波数における位相雑音レベル
<data_point>	表示データ数
<level_n>	オフセット周波数ポイント <b>n</b> における位相雑音レベル

使用例

```

ログプロット測定の結果を読み出す
FETC:LPL?
> 0.00,2000000000,-999.0,-999.0,-999.0,-50.00,-150.00

```

### :READ:LPLot[n]?

Read

#### 機能

現在の設定値でログプロット測定のシングル測定を実行したあと、結果を読み出します。

#### クエリ

:READ:LPLot [n] ?

#### レスポンス

:FETCh:LPLot [n] ?を参照してください。

#### 使用例

ログプロット測定を実行し、結果を読み出す  
READ:LPL?

#### 関連コマンド

下記コマンドと同一の機能です。  
:MEASure:LPlot [n] ?

### :MEASure:LPLot[n]?

Measure

#### 機能

現在の設定値でログプロット測定のシングル測定を実行したあと、結果を読み出します。

#### クエリ

:MEASure:LPLot [n] ?

#### レスポンス

:FETCh:LPLot [n] ?を参照してください。

#### 使用例

ログプロット測定を実行し、結果を読み出す  
MEAS:LPL?

#### 関連コマンド

下記コマンドと同一の機能です。  
:READ:LPLot [n] ?

## 2.5.2 Average Count

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

Average Count

### 機能

Average Count を設定します。

### コマンド

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

### パラメータ

<code>&lt;integer&gt;</code>	Average Count
範囲	1~999
分解能	1
初期値	1

### 使用例

Average Count を 10 に設定する  
`LPL:AVER:COUN 10`

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

Average Count Query

### 機能

Average Count の設定を読み出します。

### クエリ

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

### レスポンス

`<integer>`

### パラメータ

<code>&lt;integer&gt;</code>	Average Count
範囲	1~999
分解能	1

### 使用例

Average Count の設定を読み出す  
`LPL:AVER:COUN?`  
`> 50`

## 2.5.3 Phase Noise Optimization

[**:SENSe**]:**FREQuency**:**SYNTHeSis**[**:STATe**] 0|1|2|3

Phase Noise Optimization

### 機能

位相雑音測定最適化モードを選択します。

### コマンド

[**:SENSe**]:**FREQuency**:**SYNTHeSis**[**:STATe**] <mode>

### パラメータ

<mode> 位相雑音測定最適化モード

[MS2840A, MS2850A の場合]

- |   |  |
|---|--|
| 0 | キャリアに対し近傍の位相雑音測定と遠方の位相雑音測定それぞれに最適な Loop Filter を自動的に切り替えて測定します(Auto)。(初期値) |
| 1 | キャリアに対し近傍の位相雑音測定に最適な Loop Filter で測定します (Best Close-in)。                   |
| 2 | キャリアに対し遠方の位相雑音測定に最適な Loop Filter で測定します (Best Wide-offset)。                |
| 3 | キャリアに対し近傍と遠方をバランスよく位相雑音測定できる Loop Filter で測定します (Balance)。                 |

[MS269xA, MS2830A の場合]

- |   |   |
|---|---|
| 3 | キャリアに対し近傍と遠方をバランスよく位相雑音測定できる Loop Filter を選択して測定します (Balance)。(初期値, 固定) |
|---|---|

### 使用例

位相雑音測定最適化モードを遠方位相雑音測定最適化に設定する

**FREQ:SYNT 2**

[:SENSe]:FREQuency:SYNTHeSis[:STATe]?

Phase Noise Optimization

## 機能

位相雑音測定の最適化モードを読み出します。

## クエリ

[:SENSe]:FREQuency:SYNTHeSis[:STATe]?

## レスポンス

&lt;mode&gt;

## パラメータ

&lt;mode&gt; 位相雑音測定の最適化モード

[MS2840A, MS2850A の場合]

- 0 キャリアに対し近傍の位相雑音測定と遠方の位相雑音測定それぞれに最適な Loop Filter を自動的に切り替えて測定します (Auto)。
- 1 キャリアに対し近傍の位相雑音測定に最適な Loop Filter で測定します (Best Close-in)。
- 2 キャリアに対し遠方の位相雑音測定に最適な Loop Filter で測定します (Best Wide-offset)。
- 3 キャリアに対し近傍と遠方をバランスよく位相雑音測定できる Loop Filter で測定します (Balance)。

[MS269xA, MS2830A の場合]

- 3 キャリアに対し近傍と遠方をバランスよく位相雑音測定できる Loop Filter を選択して測定します (Balance)。(固定)

## 使用例

位相雑音測定の最適化モードを読み出す

FREQ:SYNT?

&gt; 2

## 2.5.4 Marker Mode

:CALCulate:LPLot:MARKer[1|2|3|4|5|6|7|8]:MODE

NORMal|INTEgralnoise|RMSNoise|JITTer|RESidualfm|OFF

Marker Mode

### 機能

ログプロット測定におけるマーカモードを設定します。

### コマンド

:CALCulate:LPLot:MARKer[n]:MODE <mode>

### パラメータ

<n>	マーカ番号
1	マーカ 1 を対象とする
2	マーカ 2 を対象とする
3	マーカ 3 を対象とする
4	マーカ 4 を対象とする
5	マーカ 5 を対象とする
6	マーカ 6 を対象とする
7	マーカ 7 を対象とする
8	マーカ 8 を対象とする
省略時	マーカ 1 を対象とする

### 注

Marker Mode を Integral Noise, RMS Noise, Jitter または Residual FM に設定している場合, マーカは指定している Active Marker に固定されます。

<mode>	マーカモード
NORMal	Normal
INTEgralnoise	Integral Noise
RMSNoise	RMS Noise
JITTer	Jitter
RESidualfm	Residual FM
OFF	Off
初期値	Normal (マーカ 1~7) Off (マーカ 8)

### 使用例

マーカ 8 のマーカモードを Normal に設定する  
 CALC:LPL:MARK8:MODE NORM

:CALCulate:LPLot:MARKer[1]|2|3|4|5|6|7|8:MODE?

Marker Mode Query

#### 機能

ログプロット測定におけるマーカモードを読み出します。

#### クエリ

:CALCulate:LPLot:MARKer[n]:MODE?

#### レスポンス

<mode>

#### パラメータ

<n>	マーカ番号
1	マーカ 1 を対象とする
2	マーカ 2 を対象とする
3	マーカ 3 を対象とする
4	マーカ 4 を対象とする
5	マーカ 5 を対象とする
6	マーカ 6 を対象とする
7	マーカ 7 を対象とする
8	マーカ 8 を対象とする
省略時	マーカ 1 を対象とする

<mode>	マーカモード
NORM	Normal
INT	Integral Noise
RMSN	RMS Noise
JITT	Jitter
RES	Residual FM
OFF	Off

#### 使用例

マーカ 2 のマーカモードを読み出す  
 CALC:LPL:MARK2:MODE?  
 > NORM

## 2.5.5 Analysis Width

:CALCulate:LPLot:MARKer[1]|2|3|4|5|6|7|8:WIDTh:STARt <rel\_freq>

Marker Width Start

### 機能

ログプロット測定におけるマーカ範囲の開始位置を設定します。

### コマンド

:CALCulate:LPLot:MARKer[n]:WIDTh:STARt <rel\_freq>

### パラメータ

<n>	マーカ番号
1	マーカ 1 を対象とする
2	マーカ 2 を対象とする
3	マーカ 3 を対象とする
4	マーカ 4 を対象とする
5	マーカ 5 を対象とする
6	マーカ 6 を対象とする
7	マーカ 7 を対象とする
8	マーカ 8 を対象とする
省略時	マーカ 1 を対象とする
<freq>	周波数オフセット
範囲	10 Hz～10 MHz
分解能	1 Hz (10 Hz～100 Hz)
	10 Hz (100 Hz～1 kHz)
	100 Hz (1 kHz～10 kHz)
	1 kHz (10 kHz～100 kHz)
	10 kHz (100 kHz～1 MHz)
	100 kHz (1 MHz～10 MHz)
初期値	1 kHz
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。

### 使用例

マーカ 1 のマーカ範囲の開始位置を 5 kHz に設定する

CALC:LPL:MARK1:WIDT:STAR 5KHZ

:CALCulate:LPLot:MARKer[1]|2|3|4|5|6|7|8:WIDTh:STARt?

Marker Width Start Query

#### 機能

ログプロット測定におけるマーカ範囲の開始位置を読み出します。

#### クエリ

:CALCulate:LPLot:MARKer[n]:WIDTh:STARt?

#### レスポンス

<rel\_freq>

#### パラメータ

<n>	マーカ番号
1	マーカ 1 を対象とする
2	マーカ 2 を対象とする
3	マーカ 3 を対象とする
4	マーカ 4 を対象とする
5	マーカ 5 を対象とする
6	マーカ 6 を対象とする
7	マーカ 7 を対象とする
8	マーカ 8 を対象とする
省略時	マーカ 1 を対象とする
<rel_freq> 範囲	周波数オフセット Start Offset~Stop Offset
分解能	1 Hz (10 Hz~100 Hz)
	10 Hz (100 Hz~1 kHz)
	100 Hz (1 kHz~10 kHz)
	1 kHz (10 kHz~100 kHz)
	10 kHz (100 kHz~1 MHz)
	100 kHz (1 MHz~10 MHz)
サフィックスコード	なし, Hz 単位の値を返します。

#### 使用例

マーカ 1 のマーカ範囲の開始位置を読み出す

CALC:LPL:MARK1:WIDT:STAR?

> 5000

:CALCulate:LPLot:MARKer[1]|2|3|4|5|6|7|8:WIDTh:STOP <rel\_freq>

Marker Width Stop

機能

ログプロット測定におけるマーカ範囲の終了位置を設定します。

コマンド

:CALCulate:LPLot:MARKer[n]:WIDTh:STOP <rel\_freq>

パラメータ

<n>	マーカ番号
1	マーカ 1 を対象とする
2	マーカ 2 を対象とする
3	マーカ 3 を対象とする
4	マーカ 4 を対象とする
5	マーカ 5 を対象とする
6	マーカ 6 を対象とする
7	マーカ 7 を対象とする
8	マーカ 8 を対象とする
省略時	マーカ 1 を対象とする
<freq>	周波数オフセット
範囲	Start Offset～Stop Offset
分解能	1 Hz (10 Hz～100 Hz)
	10 Hz (100 Hz～1 kHz)
	100 Hz (1 kHz～10 kHz)
	1 kHz (10 kHz～100 kHz)
	10 kHz (100 kHz～1 MHz)
	100 kHz (1 MHz～10 MHz)
初期値	100 kHz
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。

使用例

マーカ 1 のマーカ範囲の終了位置を 500 kHz に設定する  
 CALC:LPL:MARK1:WIDT:STOP 500KHZ

:CALCulate:LPLot:MARKer[1]|2|3|4|5|6|7|8:WIDTh:STOP?

Marker Width Stop Query

#### 機能

ログプロット測定におけるマーカ範囲の終了位置を読み出します。

#### クエリ

:CALCulate:LPLot:MARKer[n]:WIDTh:STOP?

#### レスポンス

<rel\_freq>

#### パラメータ

<n>	マーカ番号
1	マーカ 1 を対象とする
2	マーカ 2 を対象とする
3	マーカ 3 を対象とする
4	マーカ 4 を対象とする
5	マーカ 5 を対象とする
6	マーカ 6 を対象とする
7	マーカ 7 を対象とする
8	マーカ 8 を対象とする
省略時	マーカ 1 を対象とする
<rel_freq> 範囲	周波数オフセット Start Offset～Stop Offset
分解能	1 Hz (10 Hz～100 Hz)
	10 Hz (100 Hz～1 kHz)
	100 Hz (1 kHz～10 kHz)
	1 kHz (10 kHz～100 kHz)
	10 kHz (100 kHz～1 MHz)
	100 kHz (1 MHz～10 MHz)
サフィックスコード	なし, Hz 単位の値を返します。

#### 使用例

マーカ 1 のマーカ範囲の終了位置を読み出す  
 CALC:LPL:MARK1:WIDT:STOP?  
 > 500000

## 2.5.6 Marker Frequency Offset

:CALCulate:LPLot:MARKer[1|2|3|4|5|6|7|8]:X <rel\_freq>

Marker Frequency Offset

### 機能

ログプロット測定におけるマーカの周波数オフセットを設定します。

### コマンド

:CALCulate:LPLot:MARKer[n]:X <rel\_freq>

### パラメータ

<n>	マーカ番号
1	マーカ 1 を対象とする
2	マーカ 2 を対象とする
3	マーカ 3 を対象とする
4	マーカ 4 を対象とする
5	マーカ 5 を対象とする
6	マーカ 6 を対象とする
7	マーカ 7 を対象とする
8	マーカ 8 を対象とする
省略時	マーカ 1 を対象とする
<rel_freq>	周波数オフセット
範囲	Start Offset～Stop Offset
分解能	1 Hz (10 Hz～100 Hz)
	10 Hz (100 Hz～1 kHz)
	100 Hz (1～10 kHz)
	1 kHz (10～100 kHz)
	10 kHz (100 kHz～1 MHz)
	100 kHz (1 MHz～10 MHz)
初期値	10 Hz (マーカ 1)
	100 Hz (マーカ 2)
	1 kHz (マーカ 3)
	10 kHz (マーカ 4)
	100 kHz (マーカ 5)
	1 MHz (マーカ 6)
	10 MHz (マーカ 7)
	10 MHz (マーカ 8)
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。

### 使用例

マーカ 1 の周波数オフセットを 5 kHz に設定する

CALC:LPL:MARK1:X 5KHZ

:CALCulate:LPLot:MARKer[1]|2|3|4|5|6|7|8:X?

Marker Frequency Offset Query

## 機能

ログプロット測定におけるマーカの周波数オフセットを読み出します。

## クエリ

:CALCulate:LPLot:MARKer[n]:X?

## レスポンス

&lt;rel\_freq&gt;

## パラメータ

<n>	マーカ番号
1	マーカ 1 を対象とする
2	マーカ 2 を対象とする
3	マーカ 3 を対象とする
4	マーカ 4 を対象とする
5	マーカ 5 を対象とする
6	マーカ 6 を対象とする
7	マーカ 7 を対象とする
8	マーカ 8 を対象とする
省略時	マーカ 1 を対象とする
<rel_freq>	周波数オフセット
範囲	Start Offset~Stop Offset
分解能	1 Hz (10 Hz~100 Hz)
	10 Hz (100 Hz~1 kHz)
	100 Hz (1~10 kHz)
	1 kHz (10~100 kHz)
	10 kHz (100 kHz~1 MHz)
	100 kHz (1 MHz~10 MHz)
	Hz 単位の値を返します。

## 使用例

```

マーカ 1 の周波数オフセットを読み出す
CALC:LPL:MARK1:X?
> 5000

```

## 2.5.7 Marker Result

:CALCulate:LPLot:MARKer[1]|2|3|4|5|6|7|8:Y?

Marker Result Query

機能

ログプロット測定におけるマーカ点の位相雑音レベルを読み出します。

コマンド

:CALCulate:LPLot:MARKer[n]:Y?

レスポンス

<level>

パラメータ

<n>	マーカ番号
1	マーカ 1 を対象とする
2	マーカ 2 を対象とする
3	マーカ 3 を対象とする
4	マーカ 4 を対象とする
5	マーカ 5 を対象とする
6	マーカ 6 を対象とする
7	マーカ 7 を対象とする
8	マーカ 8 を対象とする
省略時	マーカ 1 を対象とする

<level>	位相雑音レベル
サフィックスコード	なし, dBc/Hz 単位の値を返します。

使用例

```
マーカ 2 の位相雑音レベルを読み出す  
CALC:LPL:MARK2:Y?  
> -150.00
```

:CALCulate:LPLot:MARKer[1]|2|3|4|5|6|7|8:VALue?

Marker Result Value Query

機能

ログプロット測定におけるマーカ点の計算値を読み出します。

コマンド

:CALCulate:LPLot:MARKer[n]:VALue?

レスポンス

<value>

パラメータ

<n>	マーカ番号
1	マーカ 1 を対象とする
2	マーカ 2 を対象とする
3	マーカ 3 を対象とする
4	マーカ 4 を対象とする
5	マーカ 5 を対象とする
6	マーカ 6 を対象とする
7	マーカ 7 を対象とする
8	マーカ 8 を対象とする
省略時	マーカ 1 を対象とする

<value>	計算値
サフィックスコード	なし, マーカモードの単位の計算値を返します。
マーカモード	単位
NORMal	dBc/Hz
INTegralnoise	dBc
RMSNoise	radian
JITTer	second
RESidualfm	Hz

使用例

マーカ 2 のマーカモードの単位の計算値を読み出す  
 CALC:LPL:MARK2:VAL?  
 > -60.00



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

範囲 0~65535

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。

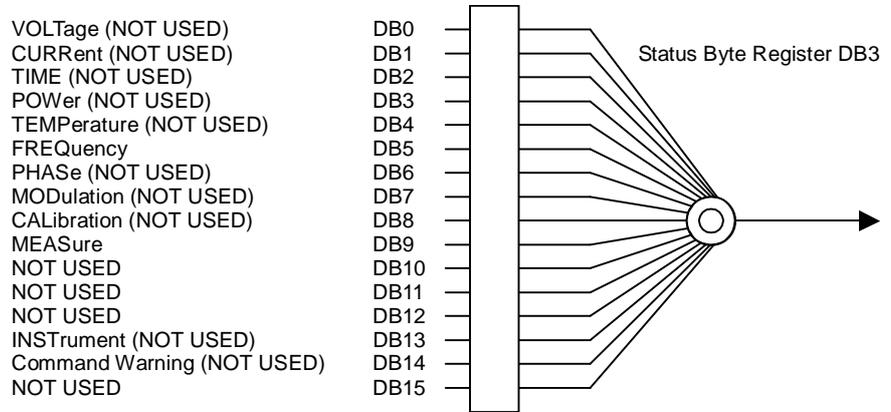


図3.2-1 QUESTIONABLE ステータスレジスタ

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

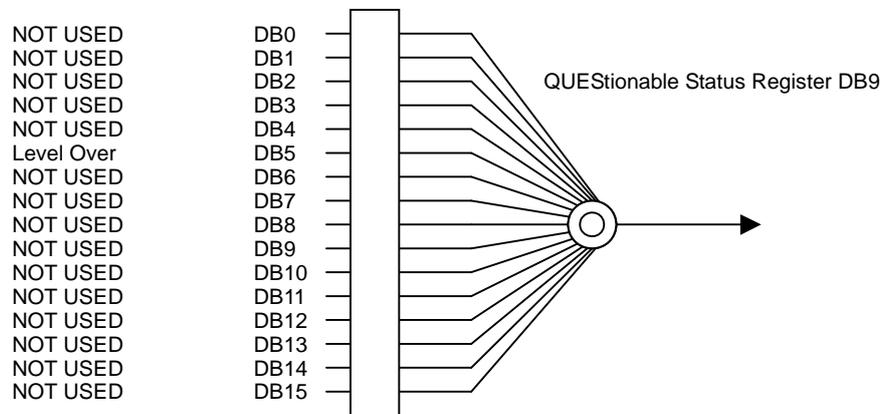


図3.2-2 QUESTIONABLE Measure レジスタ

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

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 3.2-3のとおりです。

表3.2-3 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

&lt;integer&gt;

## 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

&lt;integer&gt;

## 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

&lt;integer&gt;

## 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 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

&lt;integer&gt;

## 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 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。

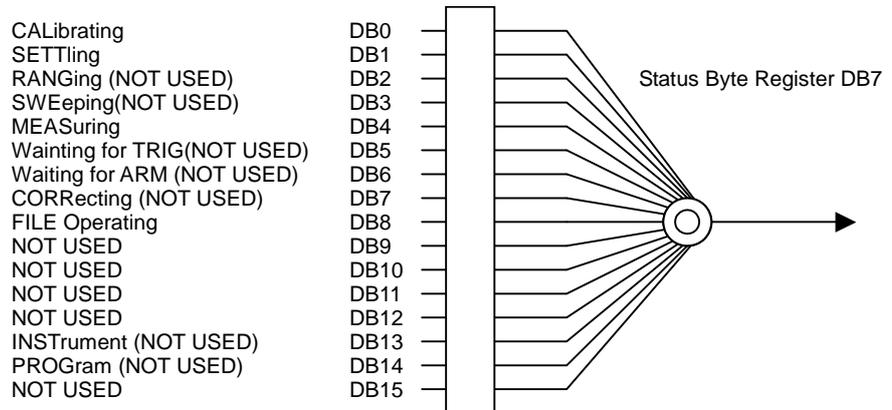


図3.3-1 OPERation ステータスレジスタ

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 (Continuous 中は常に 1 となります)
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
```

