User's Manual

DL9040/DL9140/DL9240 Series Digital Oscilloscope Serial Bus Signal Analysis Function

(I^2C Bus Signal Analysis Function/CAN Bus Signal Analysis Function/LIN Bus Signal Analysis Function/SPI Bus Signal Analysis Function)

YOKOGAWA
Yokogawa Electric Corporation

IM 701310-51E
5th Edition

Thank you for purchasing the DL9040/DL9140/DL9240 Series Digital Oscilloscope (DL9040/DL9040L/DL9140/DL9140L/DL9240/DL9240L, hereafter referred to as the DL9000) with the Serial Bus Signal Analysis Function (option).* This user's manual describes only the serial bus signal analysis function (I^2C Bus Signal Analysis Function, CAN Bus Signal Analysis Function, LIN Bus Signal Analysis Function, and SPI Bus Signal Analysis Function).

* /F5 option: Model with the I^2C Bus Signal Analysis Function and SPI Bus Signal Analysis Function
/F7 option: Model with the CAN Bus Signal Analysis Function, LIN Bus Signal Analysis Function and SPI Bus Signal Analysis Function
/F8 option: Model with the I^2C Bus Signal Analysis Function, CAN Bus Signal Analysis Function, LIN Bus Signal Analysis Function, and SPI Bus Signal Analysis Function

For information about other functions, operating procedures, and handling precautions of the DL9000, see the following manuals.

Manual Title	Manual No.	Description
DL9040/DL9140/DL9240 Series Digital Oscilloscope User's Manual	IM 701310-01E	Explains all functions and procedures of the DL9040/DL9140/DL9240 series excluding the communication functions.
DL9040/DL9140/DL9240 Series Digital Oscilloscope Communication Interface User's Manual (in CD)	IM 701310-17E	Explains the communication interface functions of the DL9040/DL9140/DL9240 series.
DL9000 Series Digital Oscilloscope Power Supply Analysis Function User's Manual	IM 701310-61E	Explains the operatiing procedures of the optional power supply analysis function.

Notes

- The contents of this manual are subject to change without prior notice as a result of continuing improvements to the instrument's performance and functions. The figures given in this manual may differ from those that actually appear on your screen.
- Every effort has been made in the preparation of this manual to ensure the accuracy of its contents. However, should you have any questions or find any errors, please contact your nearest YOKOGAWA dealer.
- The contents of this manual may not be transcribed or reproduced, in part or in their entirety, without prior permission.

Trademarks

- Microsoft, Internet Explorer, MS-DOS, Windows, Windows NT, and Windows XP are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- Adobe, Acrobat, and PostScript are trademarks of Adobe Systems Incorporated.
- For purposes of this manual, the TM and ® symbols do not accompany their respective trademark names or registered trademark names.
- Other company and product names are trademarks or registered trademarks of their respective holders.

Revisions

- 1st Edition: September 2005
- 2nd Edition: March 2006
- 3rd Edition: December 2006
- 4th Edition: March 2007
- 5th Edition: August 2007

5th Edition : August 2007 (YK)

Conventions Used in This Manual

Safety Markings

The following markings are used in this manual.

⚠ *Improper handling or use can lead to injury to the user or damage to the instrument.*This symbol appears on the instrument to indicate that the user must refer to the user's manual for special instructions. The same symbol appears in the corresponding place in the user's manual to identify those instructions. In the manual, the symbol is used in conjunction with the word "WARNING" or "CAUTION."

WARNING Calls attention to actions or conditions that could cause serious or fatal injury to the user, and precautions that can be taken to prevent such occurrences.

CAUTION Calls attentions to actions or conditions that could cause light injury to the user or damage to the instrument or user's data, and precautions that can be taken to prevent such occurrences.

Note Calls attention to information that is important for proper operation of the instrument.

Notations Used on Pages Describing Operating Procedures

On pages that describe the operating procedures in Chapter 1 and 4, the following notations are used to distinguish the procedures from their explanations.

Procedure This subsection contains the operating procedure used to carry out the function described in the current chapter. All procedures are written with inexperienced users in mind; experienced users may not need to carry out all the steps.

Explanation This section describes the setup items and the limitations regarding the procedures.

Notations Used in the Procedures

Panel Keys and Soft keys
Bold characters used in the procedural explanations indicate characters that are marked on the panel keys or the characters of the soft keys or menus displayed on the screen.

SHIFT+Key
SHIFT+key means you will press the SHIFT key to turn ON the SHIFT key followed by the operation key. The setup menu marked in purple above the panel key that you pressed appears on the screen.

Rotary knob & SELECT
Rotary knob & SELECT key indicates selecting or setting parameters and entering values using the rotary knob, the SET key, and other keys. For details on the procedure, see section 4.1 or 4.2 in the *User's Manual (IM701310-01E).*

Unit

k	Denotes 1000. Example: 100 kS/s
K	Denotes 1024. Example: 459 KB (file data size)

Contents

1.1 Overview of the I²C Bus Signal Analysis Function

I²C Bus Signal Analysis Function

I²C Bus is an abbreviation for Integrated Circuit Bus. It is a bidirectional bus for connecting ICs. By using this function, you will be able to analyze data while displaying the I²C Bus signal waveform. The I²C Bus Signal Analysis Function consists of the following three main functions.

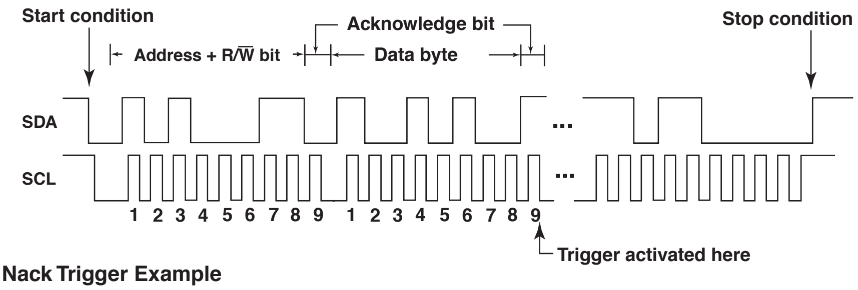
Trigger Function <See page 1-2 for the operating procedure>

A trigger can be activated under the following conditions.

- When a start condition is detected.
- When a Nack is detected.
- When the specified address pattern (7-bit address, 7-bit address + sub address, or 10-bit address) is met.
- When the data pattern is met or not met.
- When a specified general call address is detected.

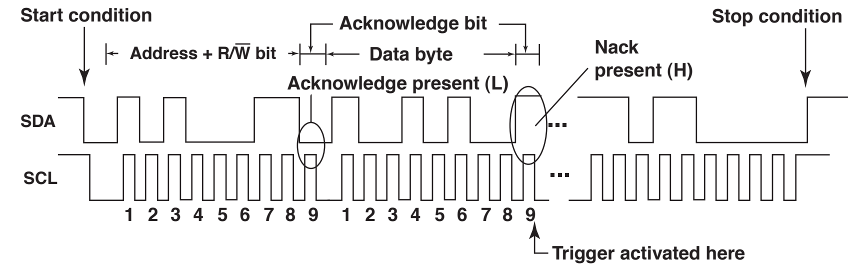
Address & Data Trigger Example

• When Activating a Trigger on the Start Condition or Address Pattern/Data Pattern



Nack Trigger Example

• When Activating a Trigger When the Acknowledge Bit Is Not Present (When the SDA Signal is high)



A trigger can be activated on the combination of the trigger conditions of the I²C bus signal and analog signal (event interval trigger). For details on the event interval trigger, see section 6.14 in the *User's Manual (IM701310-01E).*

Analysis Function <See pages 1-14 and 1-25 for the operating procedure>

This function analyzes the I²C bus signal data and shows a list of the analysis results. There are two types of analysis result lists: simple and detail. The simple list displays the analysis number, start and stop conditions, analysis data, address and data types, read/write signal, and the status of the Acknowledge bit for each byte. The detail list displays the time from the trigger position and data information in addition to the items displayed by the simple list. The data of the detail list can be saved to an arbitrary storage medium in CSV format. In addition, you can select an arbitrary byte in the analysis result list and move the zoom position (the center of the zoom box) to the head of that byte.

Search Function <See page 1-20 for the operating procedure>

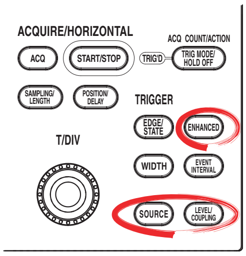
This function searches for data that matches a specific address pattern, data pattern, or Acknowledge bit status in the I²C bus signal data. When the search is executed, the zoom box (ZOOM1 or ZOOM2) moves to the data position where the conditions are met, and the data is displayed expanded in the zoom window.

1.2 Setting the I²C Bus Signal Acquisition Conditions

I²C Bus signal is acquired using a certain condition* of the I²C Bus signal as a trigger condition.

* When a start condition is detected, when a Nack is detected, when a specified address pattern is met, when a data pattern is met or not met, etc.

Procedure



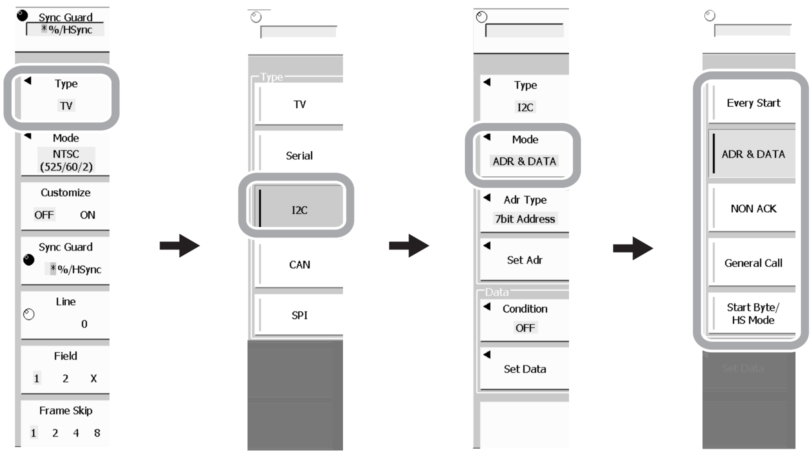
- To exit the menu during operation, press **ESC** located above the soft keys.
- For details on the operation using the rotary knob, SET, and RESET, see sections 4.1 or 4.2 in the *User's Manual (IM701310-01E).*
- For a description of the operation using a USB keyboard or a USB mouse, see section 4.3 in the *User's Manual (IM701310-01E).*

Setting the Trigger Conditions of the I²C Bus Signal

1. Press **ENHANCED** to display the ENHANCED menu.
2. Press the **Type** soft key to display the trigger type selection menu.
3. Press the **I²C** soft key.

Selecting the Trigger Mode

4. Press the **Mode** soft key to display the trigger mode of the I²C bus signal selection menu.
5. Press any of the soft keys, **Every Start**, **ADR & DATA**, **NON ACK**, **General Call**, or **Start Byte/HS Mode** to set the trigger mode.



Proceed to the steps indicated below depending on the specified trigger mode.

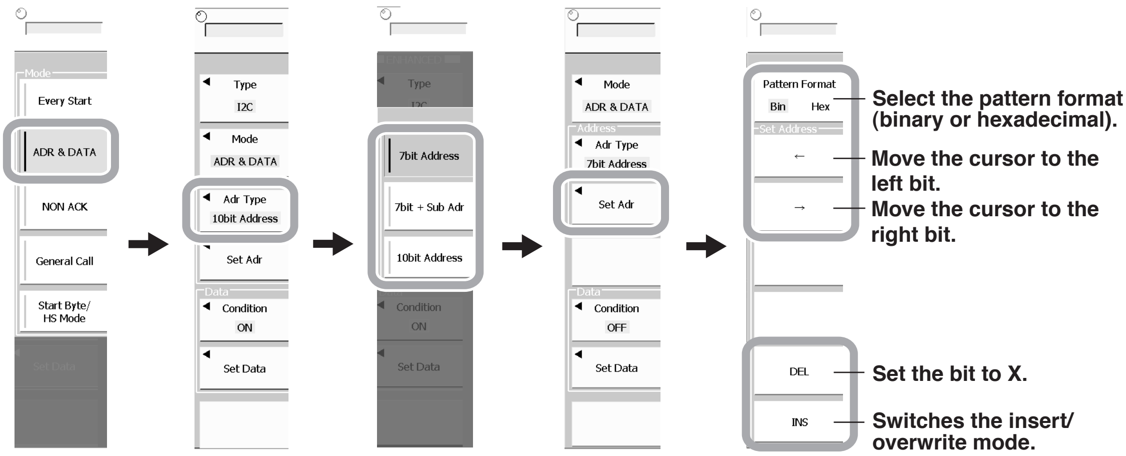
- Every Start (trigger when the start condition is detected): The setting procedure is complete.
- ADR & DATA (trigger on an address pattern or data pattern): Step 6
- NON ACK (trigger when a Nack is detected): Step 21
- General Call (trigger on the general call address): Step 23
- Start Byte/HS Mode (trigger on the start byte or HS mode start condition): Step 27

• Activating a Trigger on an Address Pattern/Data Pattern

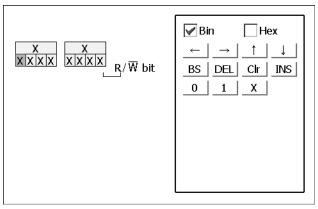
6. After step 4 of page 1-2, Press the **ADR & DATA** soft key.

Setting the Address Pattern

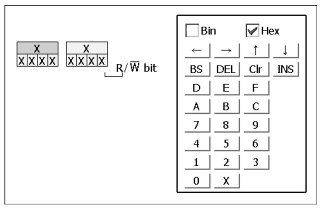
7. Press the **Adr Type** soft key to display the address type selection menu.
8. Press the soft key corresponding to the desired address type.
9. Press the **Set Adr** soft key. A screen for entering the address pattern appears. The screen varies depending on the specified address type.
10. Turn the **rotary knob**, **arrow** keys, and **SET** key to set the pattern. You can use the soft keys to change the format to binary or hexadecimal or clear all the bits (X).
11. Press **ESC**.



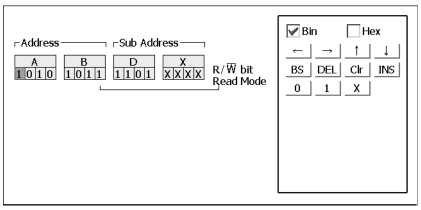
7bit Address in Binary



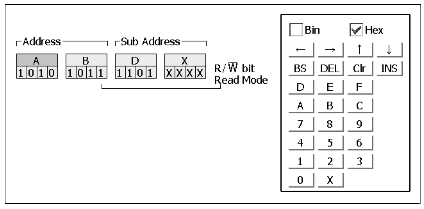
7bit Address in Hexadecimal



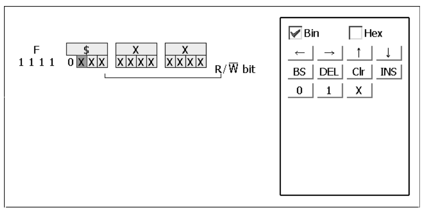
7bit+Sub Address in Binary



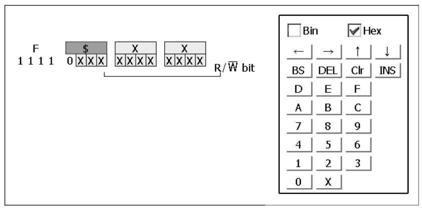
7bit+Sub Address in Hexadecimal



10bit Address in Binary

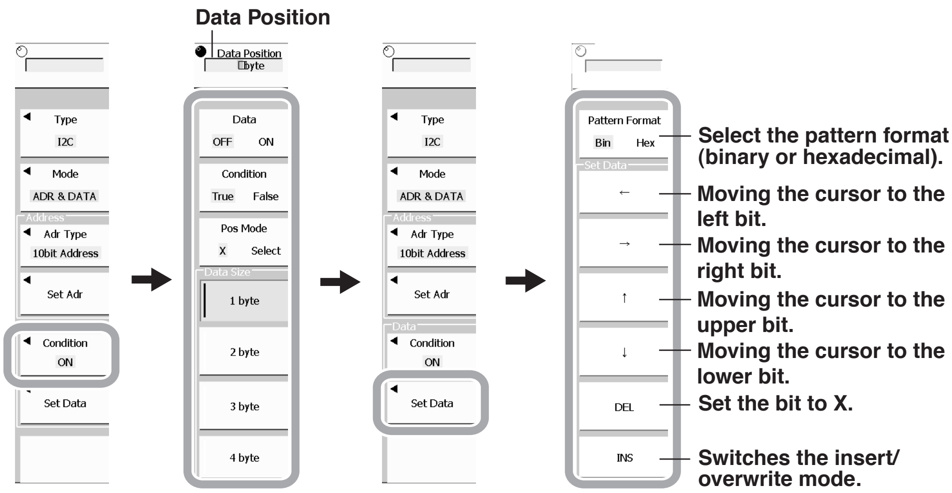


10bit Address in Hexadecimal

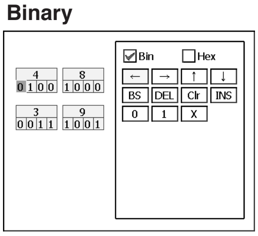


Setting the Data Pattern

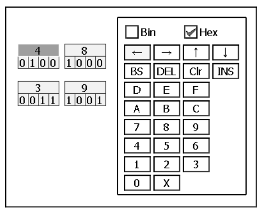
12. Press the **Condition** soft key to display a data pattern setting menu.
13. Press the **Data** soft key to select ON (use a data pattern) or OFF (not use a data pattern). If you select ON, continue with the steps below. If you select OFF, you are done.
14. Press the **Condition** soft key to select True (activate a trigger when the data matches the specified data pattern) or False (activate a trigger when the data does not match the specified data pattern).
15. Press the **Pos Mode** soft key to select X (ignore) or Select.
16. If you select Select, use the **rotary knob** to specify the data position in the pattern to be compared.
17. Press any of the soft keys, **1 byte**, **2 byte**, **3 byte**, or **4 byte** to set the data size.
18. Press **ESC**.
19. Press the **Set Data** soft key to display a data pattern setting screen.
20. Turn the **rotary knob**, **arrow** keys, and **SET** key to set the pattern.
    You can use the soft keys to change the format to binary or hexadecimal, or clear all the bits (X).



Binary

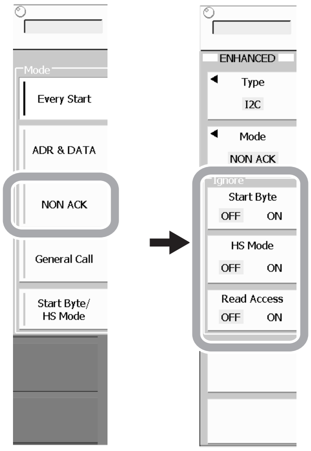


Hexadecimal



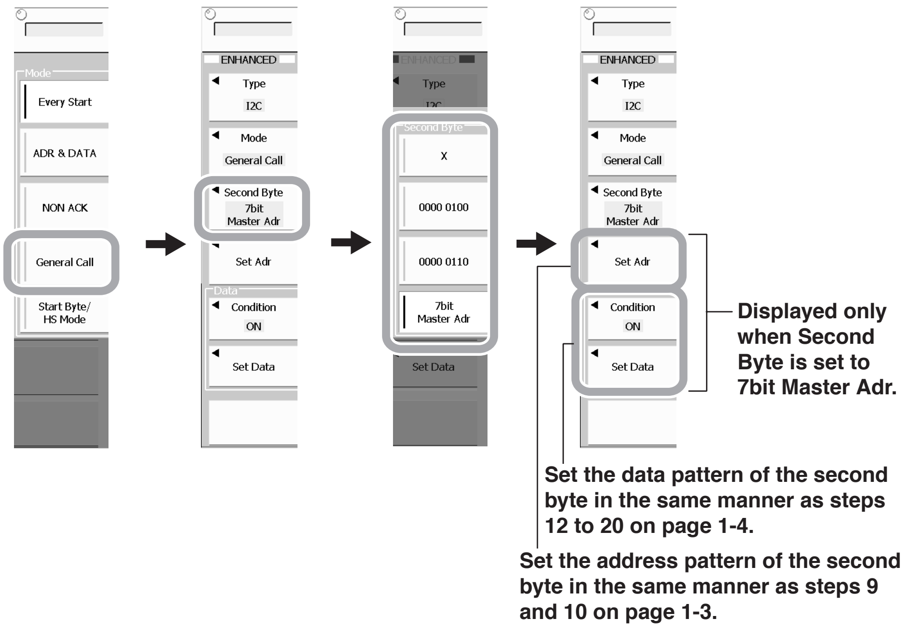
• Activating a Trigger When a Nack Bit Is Detected

21. After step 4 of page 1-2, press the **NON ACK** soft key.
22. Set whether to discard the Nack of Start Byte, HS Mode, and Read Access or include them in the trigger conditions.
    Press each of the **Start Byte**, **HS Mode**, and **Read Access** soft keys and select OFF (include in the trigger conditions) or ON (not include in the trigger conditions).



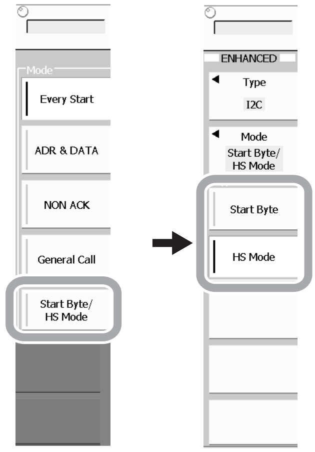
• Activating Triggers on an General Call Data Pattern

23. After step 4 of page 1-2, Press the **General Call** soft key.
24. Press the **Second Byte** soft key to display a menu used to select the format of the Second Byte.
25. Press a soft key corresponding to the desired Second Byte format.
    - When you select X, 0000 0100, or 0000 0110: The setting procedure is complete.
    - When you select 7bit Master Adr: Proceed to step 26.
26. Set the address pattern and data pattern in the same fashion as "Setting the Address Pattern" and "Setting the Data Pattern" in "Activating a Trigger on an Address Pattern or Data Pattern" of page 1-3.



• Activating a Trigger on the Start Byte or HS Mode Start Condition

27. After step 4 of page 1-2, Press the **Start Byte/HS Mode** soft key.
28. Press the **Start Byte** or **HS Mode** soft key to select whether to activate a trigger on the start byte or high speed mode start condition.



Setting the Source Channel

29. Press **SOURCE** on the front panel to display the SOURCE menu.

Setting the SDA Signal

30. Press the **SDA** soft key to display a menu used to select a channel to be the SDA signal.
31. Press any of the soft keys **CH1** to **CH4** to assign the channel to be the SDA signal.

Setting the SCL Signal

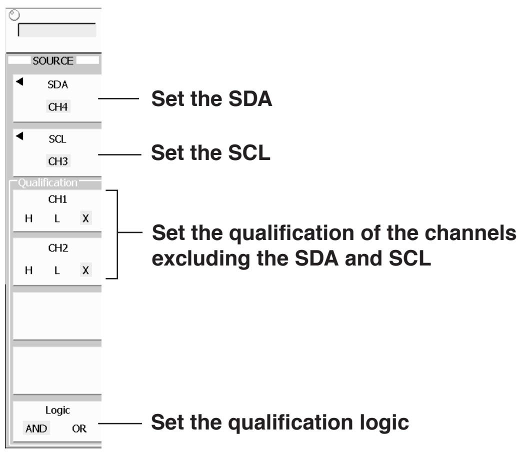
32. Press the **SCL** soft key to display a menu used to select a channel to be the SCL signal.
33. Press any of the soft keys **CH1** to **CH4** to assign the channel to the SCL signal.

Setting the Qualification of the Channels Excluding the SDA Signal and SCL Signal

34. Press a channel soft key other than the channels set to the SDA or SCL signal to select H (high), L (low), or X (don't care). If H or L is selected, a trigger is activated on the logical OR or AND of the trigger conditions of the C bus signal and the trigger conditions of the channels other than the SDA or SCL signal.

Setting the Qualification Logic

35. Press the **Logic** soft key to select AND (logical product) or OR (logical sum).



Setting the Level, Coupling, HF Rejection, and Hysteresis

36. Press **LEVEL/COUPLING**. The LEVEL/COUPLING menu appears.

Selecting the Channel to Be Configured

37. Press the **CH** soft key. The menu used to select the channel appears.
38. Press any of the soft keys, **CH1** to **CH4**.

Setting the Level

39. Turn the **rotary knob** to set the level used to determine high/low.

Selecting the Coupling

40. Press the **Coupling** soft key to select DC.

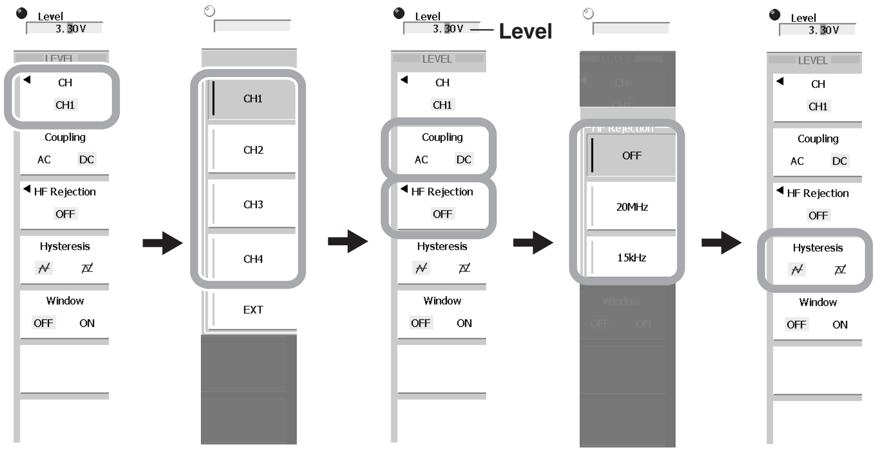
Selecting the HF Rejection

41. Press the **HF Rejection** soft key. The menu used to select the HF rejection appears.
42. Press the **OFF**, **20MHz**, or **15kHz** soft key.

Selecting the Hysteresis

43. Press the **Hysteresis** soft key to select the hysteresis.

As necessary, repeat steps 37 to 43.



Explanation

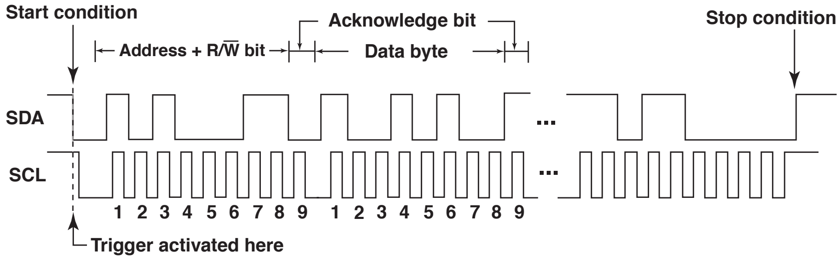
I^{2}C Bus Signal Trigger Mode

Select the trigger mode from the following list.

Every Start	Activate a trigger at the same time the start condition is detected
ADR & DATA	Activate a trigger on the address pattern or data pattern (Address & data trigger)
NON ACK	Activate a trigger when a Nack is detected (Nack trigger)
General Call	Activate a trigger on the general call address pattern
Start Byte/HS Mode	Activate a trigger on the start byte or high speed mode start condition

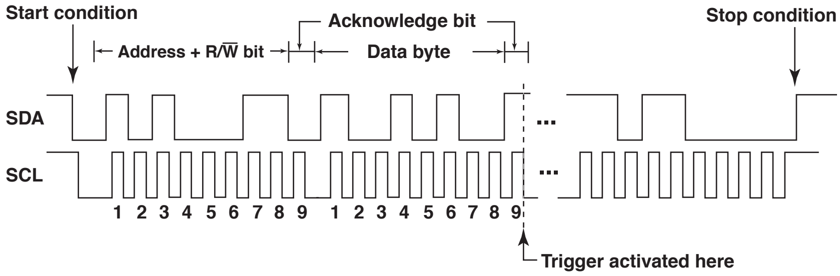
- **Every Start**

  When the start condition is detected, a trigger is activated on the falling edge of the SDA signal.



- **ADR & DATA (Address & Data Trigger)**

  When the data matches the specified address pattern or data pattern, a trigger is activated on the falling edge of the acknowledge bit.



- **Address Type (Adr Type)**

  Set the address type to 7-bit address, 7-bit + sub address, or 10-bit address.

- **Address (Set Adr)**

  Set the pattern according to the address type. The data matching the specified pattern is one of the trigger conditions.

- **Data Condition**

  To activate a trigger on a data pattern, set Data to ON in data conditions, and set the Condition, Pos Mode, Data Position, and Data Size items.

Pattern Condition

Select the pattern condition from below.

True	A trigger is activated when the data matches the data pattern.
False	A trigger is activated when the data does not match the data pattern.

Position in the Pattern to Be Compared (Pos Mode, Data Position)

Select the Pos Mode from below.

X	Does not use the position where the pattern is compared (Data Position) as a trigger condition.
Select	Set the position where the pattern is to be compared (Data Position) in terms of the number of data bytes. Set the data position in the range of 0 to 9999 bytes.

Data Size

Set the number of data bytes of the data pattern in the range of 1 to 4 bytes.

Examples of setting

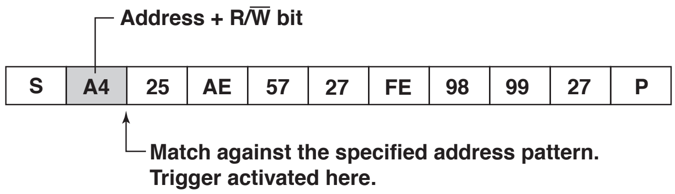
This section will indicate the data sequence in bytes (hexadecimal notation) and indicate the position where the trigger will occur. The symbols used in the figures are as follows:

S: Start condition, P: Stop condition, Shaded area: Byte pattern to be compared.

- **Activating a Trigger Only on the Address Pattern**

Trigger conditions

Mode		ADR & DATA
Address	Adr Type: Set Adr:	7bit Address A4
Data	Data	OFF

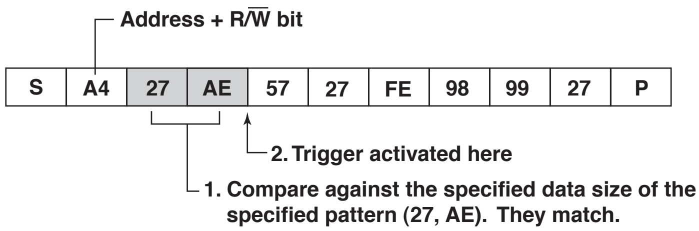


- **Activating a Trigger Only on the Data Pattern**

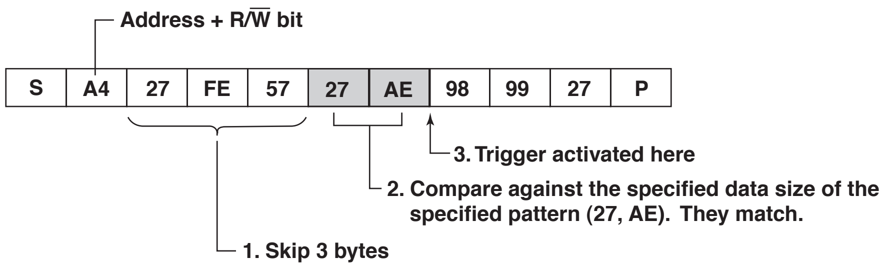
Trigger conditions

Mode		ADR & DATA
Address	Set Adr	Not applicable
Data	Data Condition Data Size Set Data	ON True 2 bytes 27, AE

<Data Position: X>



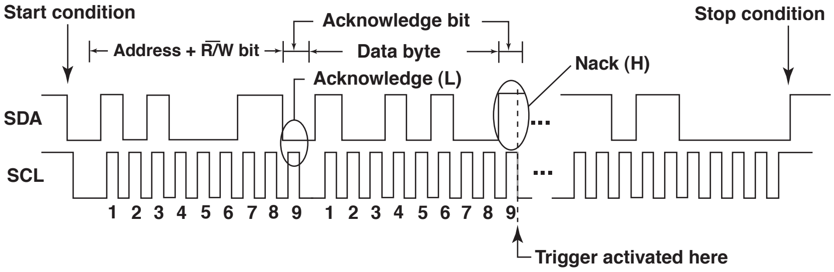
<Data Position: 3>



• NON ACK (Nack Trigger)

Activating a Trigger When the Acknowledge Bit Is High

A trigger is activated when the Acknowledge bit is Nack (when the SDA signal is set to "H").

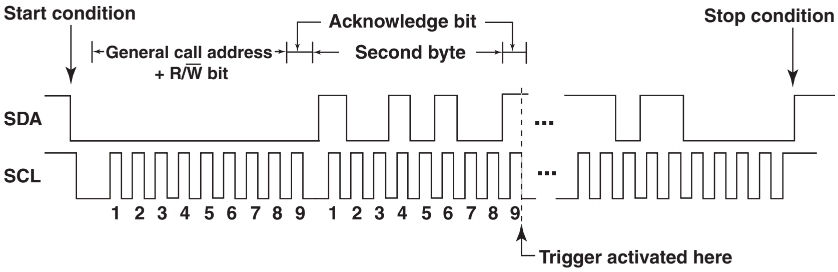


Note

You can select any of the Acknowledge bits, the status byte, HS mode master code, or read access byte, to be used for triggering.

• General Call Trigger

A trigger is activated on a general call address (Address: 0000 0000).
The second byte pattern after the general call address can also be used for triggering.



• Second Byte

Select the second byte from below.

X	Does not use the second byte pattern as a trigger condition.
0000 0100	Activates a trigger when the second byte pattern is 0000 0100.
0000 0110	Activates a trigger when the second byte pattern is 0000 0110.
7bit Master Adr	Arbitrarily set 7bit Master Adr of the second byte pattern.

Examples of setting

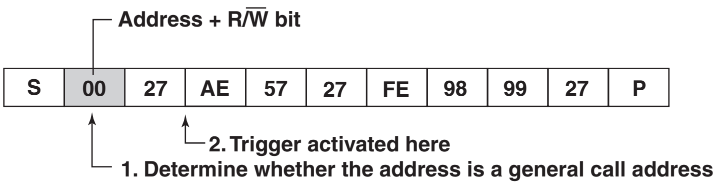
This section will indicate the data sequence in bytes (hexadecimal notation) and indicate the position where the trigger will occur. The symbols used in the figures are as follows:

S: Start condition, P: Stop condition, Shaded area: Byte pattern to be compared.

- **Activating a Trigger Only on the General Call Address**

Trigger conditions

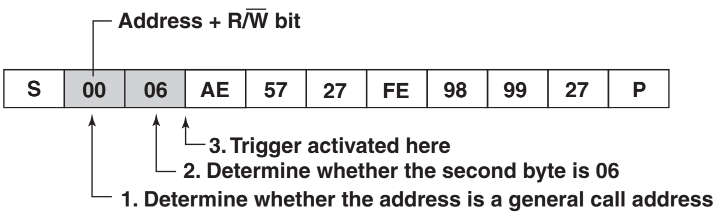
Mode	General Call
Second Byte	Not applicable



- **Activating a Trigger on the Second Byte Pattern Set to 06**

Trigger conditions

Mode	General Call
Second Byte	0000 0110

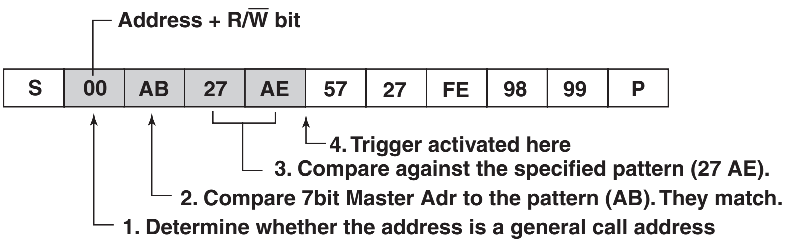


- **Activating a Trigger on an Arbitrary Pattern on the Second and Subsequent Bytes**

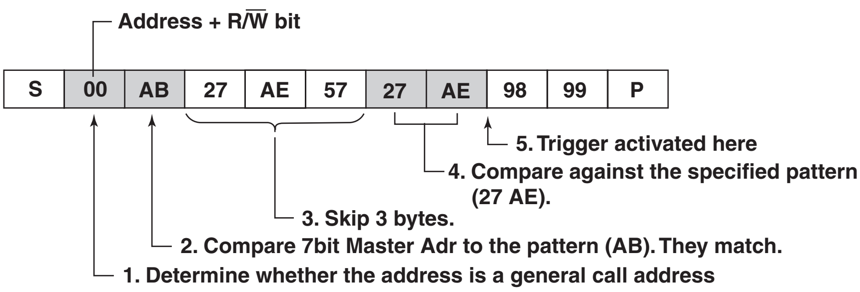
Trigger conditions

Mode		General Call
Second Byte		7bit Master Adr (1010 1011)
Data	Data	ON
	Condition	True
	Data Size	2 bytes
	Set Data	27, AE

<Data Position:X>



<Data Position:3>

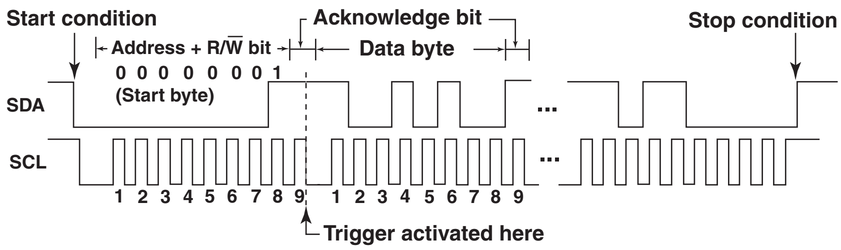


• Start Byte/HS Mode Trigger

A trigger is activated on the start byte or HS mode master code.

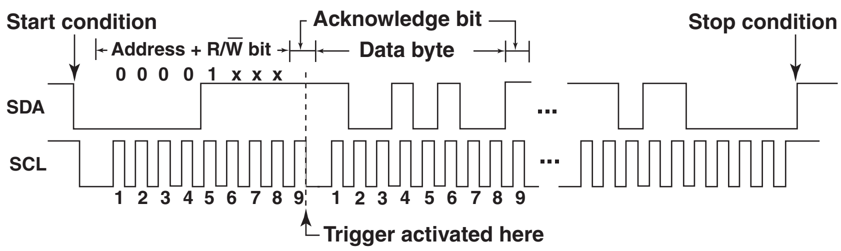
- **Start Byte (Address: 0000 0001)**

  When a start byte is detected, a trigger is activated.



- **HS Mode**

  A trigger is activated when the master code (Address: 0000 1xxx) of HS mode (high speed mode) is detected.



Examples of setting

This section will indicate the data sequence in bytes (hexadecimal notation) and indicate the position where the trigger will occur. The symbols used in the figures are as follows:

S: Start condition, P: Stop condition, Shaded area: Byte pattern to be compared.

- **Activating a Trigger Only on the Start Byte**

Trigger conditions

Mode	Start Byte/HS Mode
Type	Start Byte

S	01	Sr	AE	57	27	FE	98	99	27	P

Trigger activated here

Sr: Restart

Source Channel

The source channel of the I²C bus signal is set using the SOURCE menu that appears when you press the SOURCE key. The menu used to set the source channel of the I²C bus signal appears only when Type is set to I²C in the ENHANCED menu.

Specifying the I²C Bus Signal (SDA and SCL Signals)

The SDA and SCL signals are assigned to channels using the SOURCE menu.

Trigger Conditions of the Channels Other Than I²C Bus Signal (Qualification)

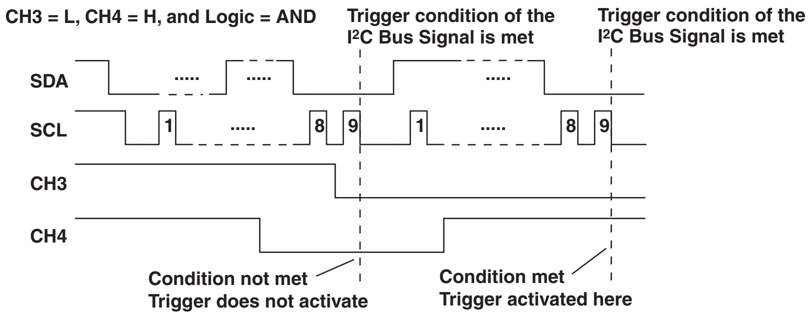
A trigger can be activated on the combination of the trigger conditions of the I²C bus signal (SDA/SCL signal) and the trigger conditions of the channels other than the I²C bus signal. Select the trigger conditions of the channels other than the I²C bus signal from the following:

H	The trigger source level is above the preset trigger level.
L	The trigger source level is below the preset trigger level.
X	Not used as a trigger source.

Logic

If the trigger condition of a channel other than the SDA or SCL signal is set to H or L, a trigger is activated on the trigger conditions of the I²C bus signal and the trigger conditions of the channels other than the SDA and SCL signals. Select the logic to used from the following:

AND	A trigger is activated when both the trigger conditions of the I²C bus signal and the trigger conditions of the channels other than the I²C bus signal are met.
OR	A trigger is activated when either the trigger conditions of the I²C bus signal or the trigger conditions of the channels other than the I²C bus signal are met.



Note

To activate a trigger only on the SCL and SDA signals (trigger condition of the I²C bus), set the status of the other channels to ignore (X), and set the logic to AND.

Trigger Level, Trigger Coupling, Etc.

Set the trigger level, trigger coupling, HF rejection, and hysteresis of each channel. For details on these items, see the *User's Manual (IM701310-01E)*.

Unification of the Trigger Setting of the I²C Bus Signal and the Setting of the I²C Bus Signal Analysis and Search

On products with software version 1.80 or higher, the trigger settings of the I²C bus signal on the menu that is entered through the ENHANCED key and the settings of the I²C bus signal analysis and search on the menu entered through the WINDOW1 and ZOOM1 key are common. The CH1 to CH4 sources are the waveforms of which the settings are common.

Trigger settings of the I²C bus signal that are applied to the settings of both the I²C bus signal analysis and search
Level (trigger level), Hysteresis (trigger hysteresis) (The settings of the I²C bus signal analysis and search are not applied to the trigger settings of the I²C bus signal.)
Item of which the trigger settings of the I²C bus signal, the settings of the I²C bus signal analysis, and the settings of the I²C bus signal search that are mutually applied
SDA Source, SCL Source

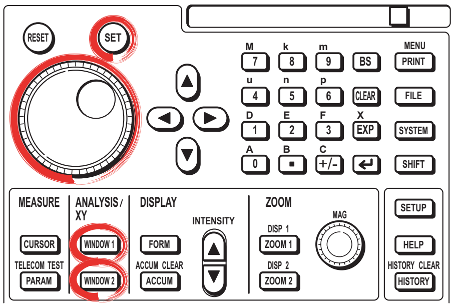
Note

- M1 to M4 are does not apply to the common settings.
- The settings of the I²C bus signal analysis and search on the menu entered through the WINDOW2 and ZOOM2 keys are not made common. They are independent settings.
- If you change common items (excluding Level and Hys) in the analysis and search menus while the waveform acquisition is in progress and the trigger type is set to I²C, the waveform acquisition is restarted.
- Even if something other than I²C bus is selected in the analysis or search menu, the level and hysteresis settings of the analysis and search are set to the same value if the trigger level or hysteresis setting is changed.
- When the trigger level or hysteresis is changed by executing auto setup, the level and hysteresis settings of the analysis and hysteresis are also set to the same new value. This also applies when the setup information is initialized.
- Trigger hysteresis and correspond to 0.6 division and 1.0 division, respectively, of the analysis or search.

1.3 Analyzing the Data

You can set any of the channels, CH1 to CH4, to SDA or SCL and analyze the data. Because the I^2C bus can be analyzed separately on Window 1 and Window 2, two waveforms can be analyzed.

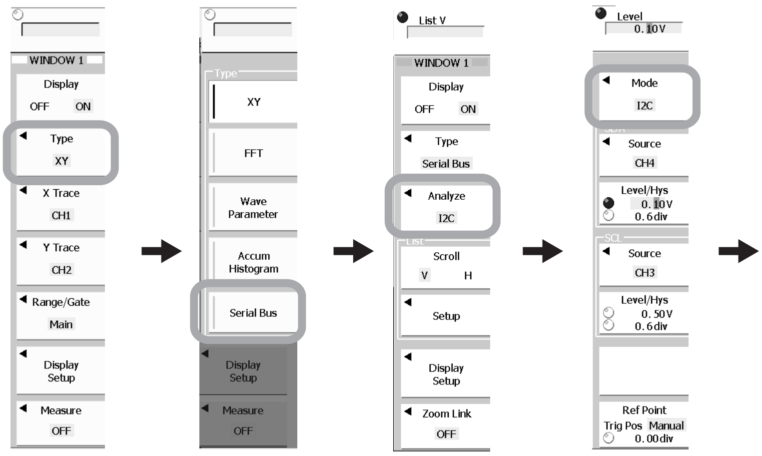
Procedure

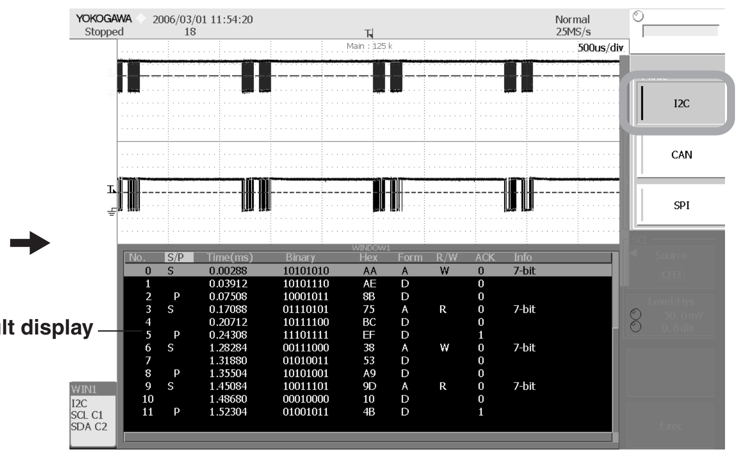


- To exit the menu during operation, press **ESC** located above the soft keys.
- For details on the operation using the rotary knob, SET, and RESET, see sections 4.1 or 4.2 in the *User's Manual ((IM701310-01E).*
- For a description of the operation using a USB keyboard or a USB mouse, see section 4.3 in the *User's Manual (IM701310-01E).*

Setting the Analysis Conditions

1. Press **WINDOW1** or **WINDOW2**.
2. Press the **Type** soft key to display the Type menu.
3. Press the **Serial Bus** soft key.
4. Press the **Analyze** soft key to display the Analyze menu.
5. Press the **Mode** soft key to display the Mode menu.
6. Press the **I^2C** soft key.





Analysis result display

Setting the Data Channel (SDA)

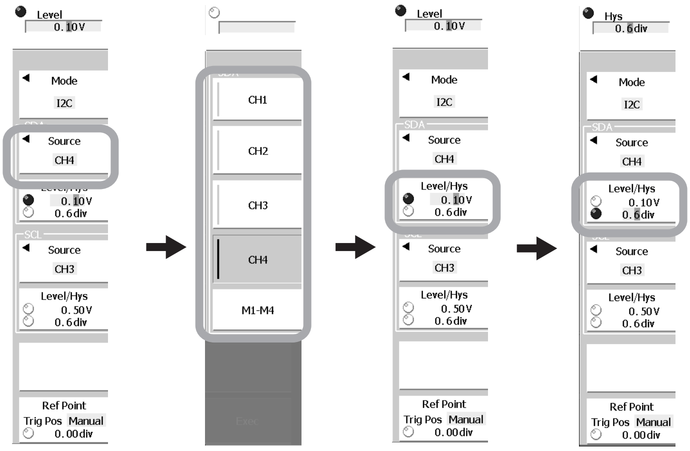
7. Press the **Source** soft key under SDA to display a data channel menu.
8. Select the data channel from CH1 to CH4 and M1 to M4 (computation channels).

- **Setting the Level**

9. Press the **Level/Hys** soft key under SDA to activate the level.
10. Turn the **rotary knob** to set the threshold level.

- **Setting the Hysteresis**

11. Press the **Level/Hys** soft key under SDA to activate the hysteresis.
12. Turn the **rotary knob** to set the hysteresis.



Setting the Clock Channel (SCL)

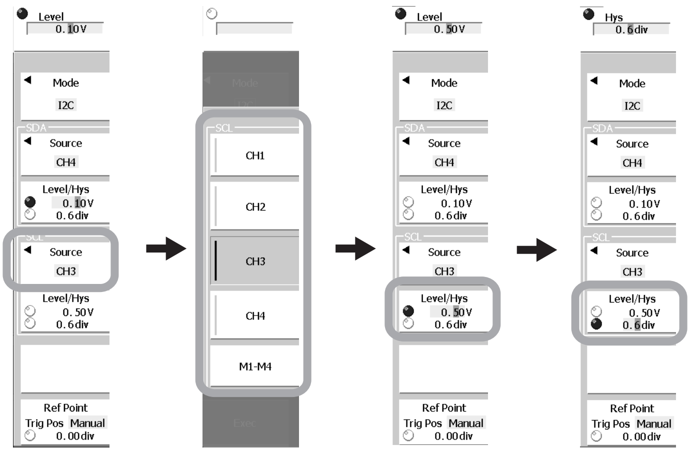
13. Press the **Source** soft key under the SCL to display a clock channel menu.
14. Select the clock channel from CH1 to CH4 and M1 to M4 (computation channels).

- **Setting the Level**

15. Press the **Level/Hys** soft key under SCL to activate the level.
16. Turn the **rotary knob** to set the threshold level.

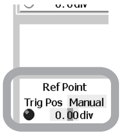
- **Setting the Hysteresis**

17. Press the **Level/Hys** soft key under SCL to activate the hysteresis.
18. Turn the **rotary knob** to set the hysteresis.



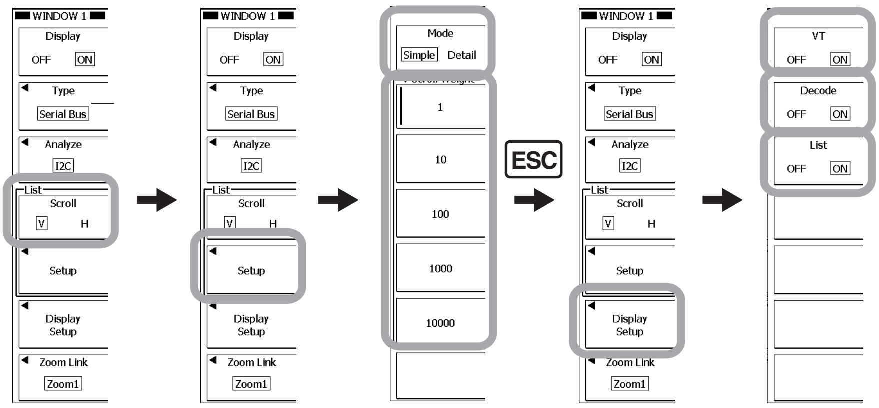
Setting the Analysis Reference Point

19. Pressing the **Ref Point** soft key to select Trig Pos or Manual.
    If you selected Manual, turn the rotary knob to set the reference point in the range of ±5.00 divisions.
20. Press **ESC**.



Display the Analysis Results

21. Press the **Scroll** soft key to select V (scroll vertically) or H (scroll horizontally).
22. Press the **Setup** soft key to display a menu used to set the analysis display.
23. Press the **Mode** soft key to select Simple (simple display) or Detail (detailed display).
24. Press any of the soft keys **1**, **10**, **100**, **1000**, and **10000** to set the step number for scrolling. The display scrolls according to the specified number.
25. Press **ESC**.
26. Press the **Display Setup** soft key to display the Display Setup menu.
27. Press the **VT** soft key to select ON or OFF. If ON is specified, the waveform is displayed on a normal voltage-time axis.
28. Likewise, press the **Decode** (firmware version 3.6 or later) or **List** soft key to select ON or OFF. If set to ON, the decoded data or a list of analysis results is displayed.
29. Press **ESC**.

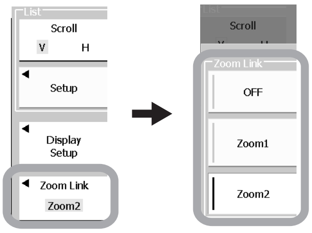


Note

The data of the simple display and detail display of the analysis results can be saved directly to a storage medium in CSV format (.csv extension). For details, see section 1.5.

Zoom Link

29. Press the **Zoom Link** soft key to display the Zoom Link menu.
30. Press any of the soft keys, **OFF**, **Zoom1**, or **Zoom2** to set the link to the zoom screen. The zoom position (the center of the zoom box) can be moved to the head of the byte selected in the analysis result list by linking Zoom1 or Zoom2.



Explanation

Setting the Analysis Conditions

You can set the following conditions.

Data Channel (SDA)

Select the data channel from CH1 to CH4 and M1 to M4. The clock channel and data channel can be set arbitrarily.

- **Level**

  Set the level used to determine the data channel signal level (0 or 1).

  Selectable range: ±10 divisions around the vertical position

  Resolution: 0.01 divisions (For example, the resolution for 2 mV/div is 0.02 mV.)

- **Hysteresis (Hys)**

  Selectable range: 0.0 division to 4.0 divisions

  Resolution: 0.1 divisions

Clock Channel (SCL)

Select the clock channel from CH1 to CH4 and M1 to M4. The clock channel and data channel can be set arbitrarily.

- **Level**

  Set the level used to determine the rising or falling edge of the clock signal.

  Selectable range: ±10 divisions around the vertical position

  Resolution: 0.01 divisions (For example, the resolution for 2 mV/div is 0.02 mV.)

- **Hysteresis (Hys)**

  Selectable range: 0.0 division to 4.0 divisions

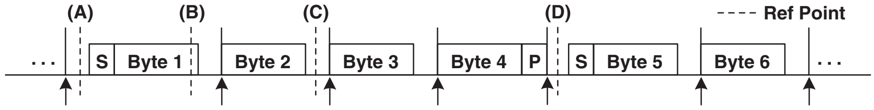
  Resolution: 0.1 divisions

Analysis Reference Point (Ref Point)

Select the analysis reference point from below.

Trig Pos	Sets the trigger position to the analysis reference point.
Manual	Manually set the analysis reference point. Selectable range: ±5.00 divisions, resolution: 0.01 divisions

Depending on the analysis reference point (Ref Point), the No. 0 byte in the analysis result list is as follows:



(A): No. 0 byte -> Byte 1 (byte 2 is No. 1, byte 3 is No. 2, ...)
(B): No. 0 byte -> Byte 1 (byte 2 is No. 1, byte 3 is No. 2, ...)
(C): No. 0 byte -> Byte 2 (byte 1 is No. -1, byte 3 is No. 1, byte 4 is No. 2...)
(D): No. 0 byte -> Byte 5 (byte 1 is No. -4, ... byte 4 is No. -1, byte 6 is No. 1...)

S: Start condition, P: Stop condition

Unification of the Trigger Setting of the I²C Bus Signal and the Setting of the I²C Bus Signal Analysis and Search

On products with software version 1.80 or higher, the trigger settings of the I²C bus signal on the menu that is entered through the ENHANCED key and the settings of the I²C bus signal analysis and search on the menu entered through the WINDOW1 and ZOOM1 key are common. For details, see page 1-13.

Analyzed Data

The following data can be analyzed:

- Waveform Data
  Analysis can be performed regardless of whether the waveform acquisition is in progress. If the waveform acquisition is in progress, the analysis results are updated in real-time in sync with the displayed waveform. The waveform data stored in the history memory (waveform data of the record number selected using HISTORY menu > Select) is also analyzed.
- Loaded Acquisition Data (ACQ data)

Analysis Range

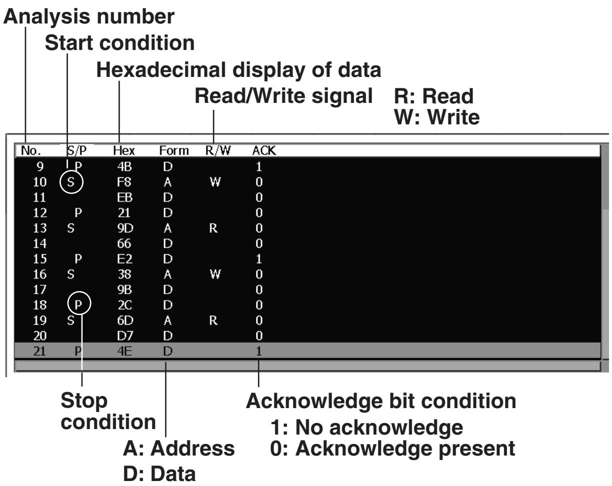
Up to 40000 bytes of analysis data can be displayed.

Analysis Data List (Analysis Result List)

The following information is displayed.

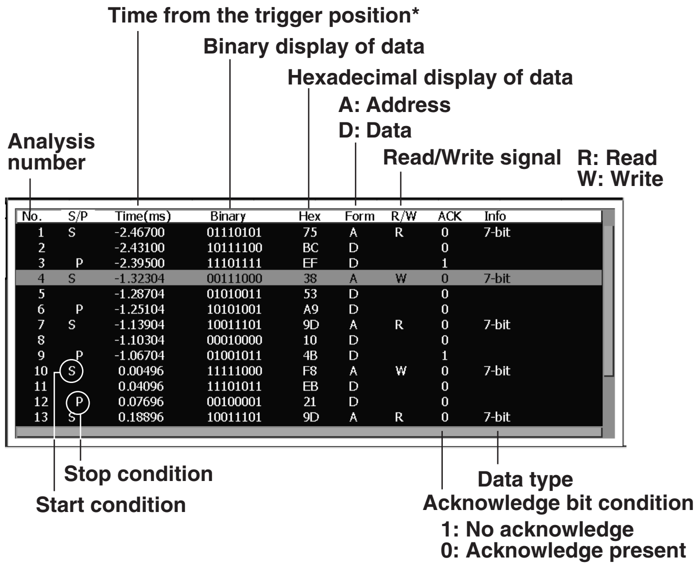
Simple Display

- Analysis number (No.): Up to 40000 points can be displayed.
- Start condition/Stop condition (S/P)
- Hexadecimal display of the data (Hex)
- Address/ Data (Form)
- Read/Write signal (R/W)
- Acknowledge bit condition (ACK)

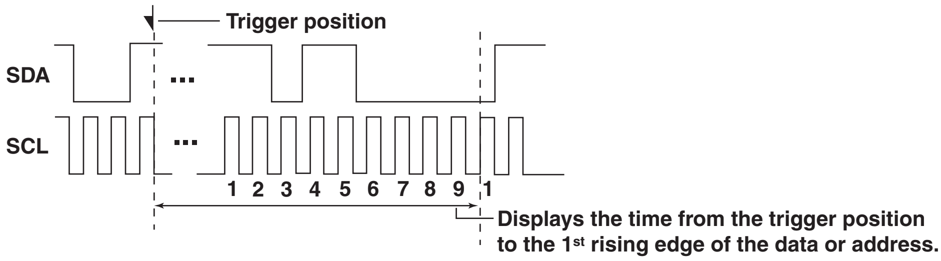


Detail Display

- Analysis number (No.): Up to 40000 points can be displayed.
- Start condition/Stop condition (S/P)
- Time from the trigger position (Time(ms))
- Binary display of the data (Binary)
- Hexadecimal display of the data (Hex)
- Address/ Data (Form)
- Read/Write signal (R/W)
- Acknowledge bit condition (ACK)
- Data information (Info)



* **About the Time (ms) Display**



Decode Display (firmware version 3.6 or later)

Decodes and displays the values of each field of the I2C bus signal.

Adr:	Displays the address in hexadecimal notation in light green.
Data:	Displays the data in hexadecimal notation in cyan.
R/W:	Displays Read/Write in pink.
Ack:	Displays Acknowledge in yellow.
General Call:	Displays the address in green for a General Call pattern.
Start Byte:	Displays the address in orange for a Start Byte pattern.
HS Mode:	Displays the address in orange for a HS Mode pattern.

Zoom Link

Select from the following:

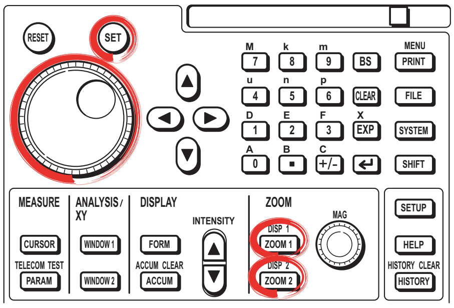
OFF	Disables the zoom link function.
Zoom1	Links to Zoom1.
Zoom2	Links to Zoom2.

By default, WINDOW1 and WINDOW2 are linked to Zoom1 and Zoom2, respectively.
If Zoom1 or Zoom2 is selected and you select (highlight) an arbitrary byte in the analysis result list, the zoom position (the center of the zoom box) of Zoom1 or Zoom2 moves to the head of that byte. Conversely, if you change the zoom position of Zoom1 or Zoom2, the highlight in the analysis result list moves to the byte that is shown in the zoom box of Zoom1 or Zoom2.

1.4 Searching the Waveform

You can search waveforms that match the specified conditions among the acquired waveforms. The setting of the search conditions are the same as the setting of the trigger conditions of section 1.2.

Procedure



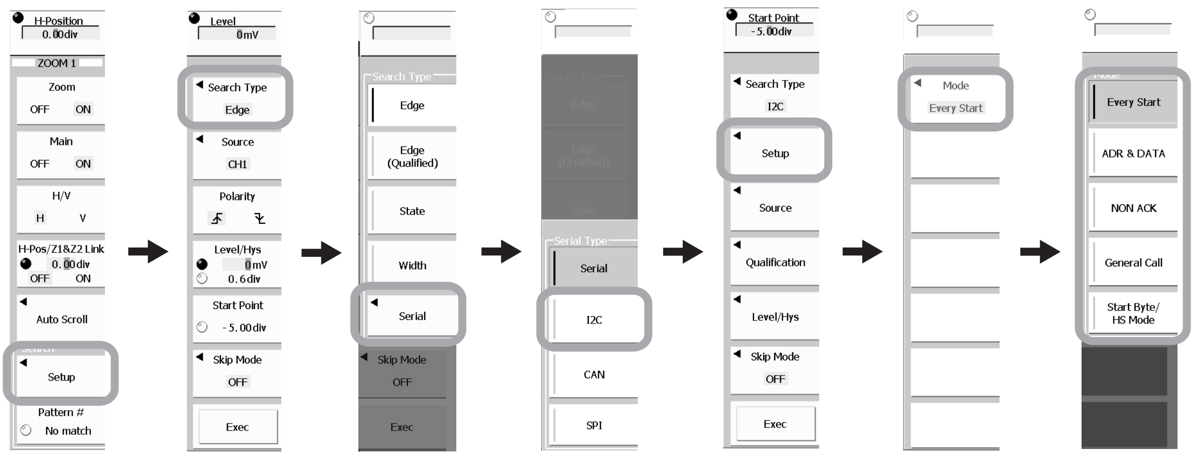
- To exit the menu during operation, press **ESC** located above the soft keys.
- For details on the operation using the rotary knob, SET, and RESET, see sections 4.1 or 4.2 in the *User's Manual (IM701310-01E).*
- For a description of the operation using a USB keyboard or a USB mouse, see section 4.3 in the *User's Manual (IM701310-01E).*

Setting the Search Conditions

1. Press **ZOOM1** or **ZOOM2** to display the ZOOM menu.
2. Press the **Setup** soft key to display the search condition menu.
3. Press the **Search Type** soft key.
4. Press the **Serial** soft key.
5. Press the **I^{2}C** soft key.
6. Press the **Setup** soft key to display the search condition menu.

Selecting the Search Mode

7. Press the **Mode** soft key to display the search mode selection menu.
8. Press the soft key corresponding to the desired search mode.



The procedure of setting the search conditions are the same as the procedure of setting the trigger conditions. See the pages indicated below according to the mode (the position of the soft keys may be different).

- Every Start (search the start condition): The setting procedure is complete.
- ADR & DATA (search an address pattern or data pattern): Page 1-3
- NON ACK (search the Nack): Page 1-5
- General Call (search the general call address): Page 1-5
- Start Byte/HS Mode (search the start byte or HS mode): Page 1-6

Setting the Source Channel

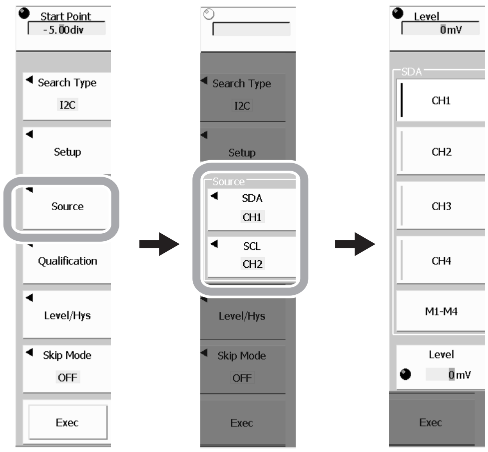
9. Press the **Source** soft key to display the source channel menu.

Setting the SDA Signal

10. Press the **SDA** soft key to display a menu used to select a channel to be the SDA signal and the level.
11. Press any of the soft keys **CH1** to **CH4** to assign the channel to the SDA signal .
12. Press the **Level** soft key. Turn the **rotary knob** to set the level used to determine high or low.

Setting the SCL Signal

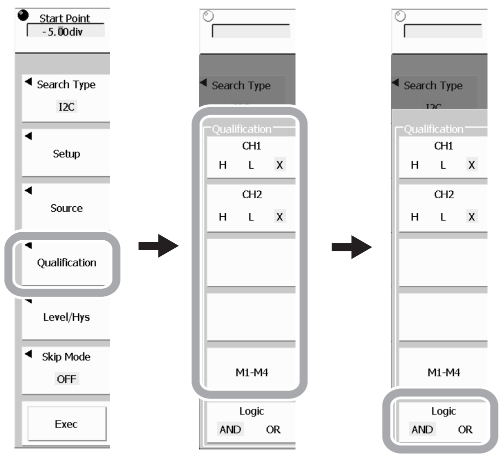
13. Press the **SCL** soft key to display a menu used to select a channel to be the SCLsignal .
14. Like the SDA, set the channel to be the SCL signal and the level.
15. Press **ESC**.



Setting the Qualification

You can search waveforms that meet the qualifications specified here and the search conditions specified in the Setup menu.

16. Press the **Qualification** soft key to display the Qualification menu.
17. Press a soft key of a channel that is not set to SDA or SCL, and set H, L, or X.
18. Press the **Logic** soft key to set whether to search the logic sum (OR) or logical product (AND) of the conditions specified in step 17.
19. Press **ESC**.

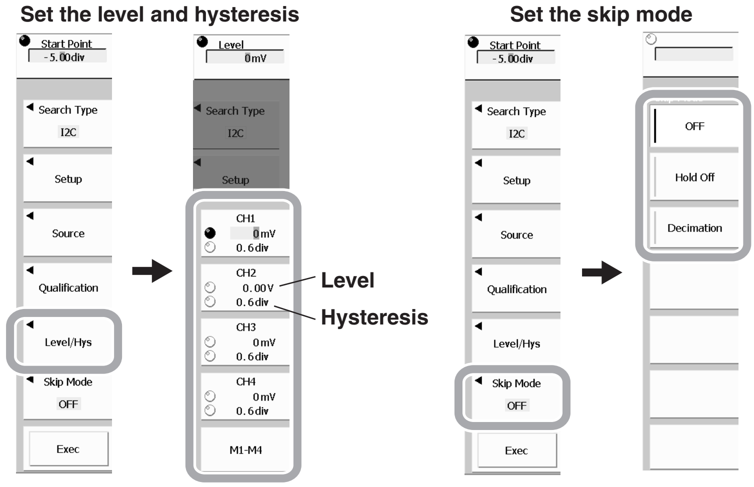


Setting the Level and Hysteresis

20. Press the **Level/Hys** soft key.
21. Press the soft key corresponding to the channel you wish to configure.
22. Turn the **rotary knob** to set the level or hysteresis.
    If you change the level of the channel set to be the source channel, the level specified in the source channel settings (steps 12 and 14) is changed.
23. Press **ESC**.

Setting the Skip Mode

24. Set the skip mode as necessary.
    Press the **Skip Mode** soft key to display the skip mode menu.
25. Press the **OFF**, **Hold Off**, or **Decimation** soft key.
26. Turn the **rotary knob** to set the time to skip or the search count.
27. Press **ESC**.



Setting the Search Start Point

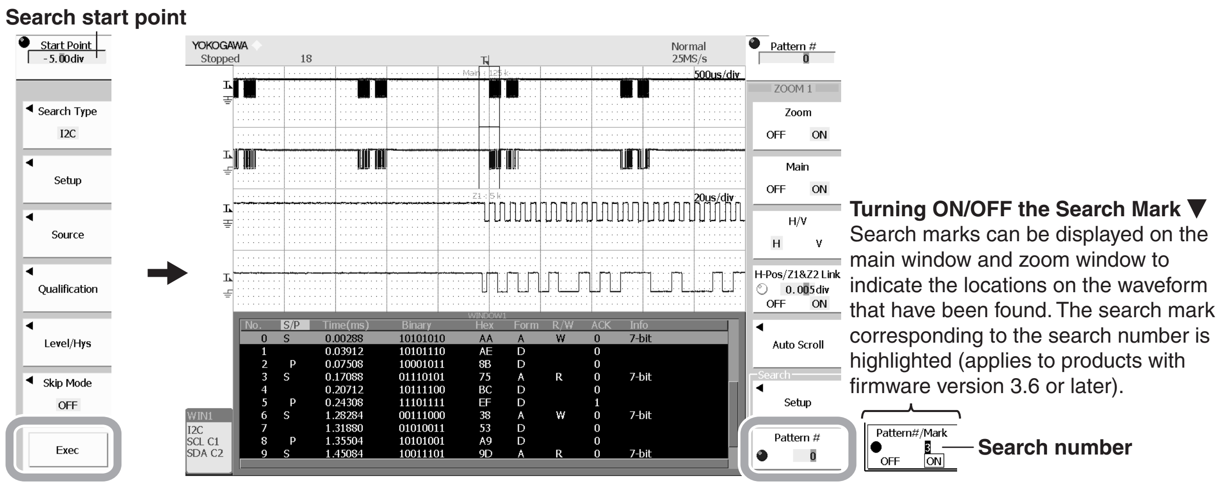
28. Turn the **rotary knob** to set the Start Point.

Executing the Search

29. Press the **Exec** soft key. The section that meets the search conditions is shown on the ZOOM display.

Viewing Search Points

30. Press **ESC**.
31. Press the **Pattern #** soft key.
32. Turn the **rotary knob** to select an arbitrary search point. The zoom position moves to the selected search point.



Explanation

Setting the Search Conditions

Search Mode

You can select the search mode.

Every Start	Search on the start condition
ADR & DATA	Search an address pattern or data pattern
NON ACK	Search Nack
General Call	Search the general call address
Start Byte/HS Mode	Search the start byte or the master code of high speed mode

The search conditions are same as the trigger conditions. For details, see section 1.2.

Source Channel, Qualification, Level, Hysteresis, Etc.

The settings for the source channel, qualification, level, hysteresis, and so on are the same as the settings of the analysis conditions. For details, see section 1.3.

Skip Mode

After searching the position that meets the search conditions, the specified amount of time or the number of positions that match the search conditions are skipped.

OFF	Searches all sections that meet the search conditions.
Hold Off	Skips the search for the specified time.
Decimation	Skips the specified number of search positions.

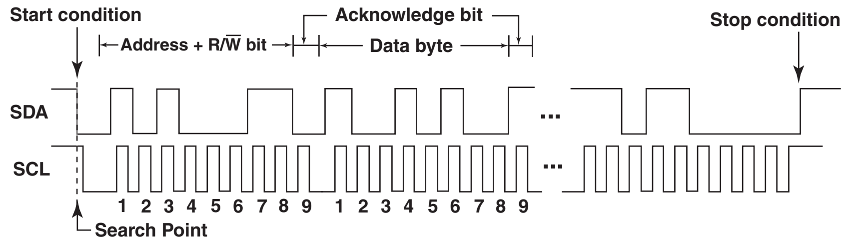
Search Start Point

Set the position to start the search in the range of ±5.00 divisions.

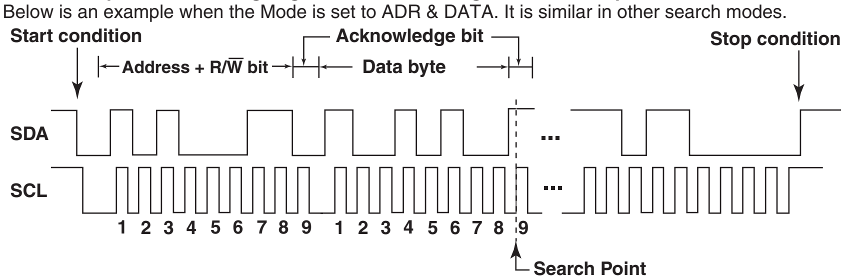
Search Point

When the search is executed, the zoom box (ZOOM1 or ZOOM2) moves to the data position where the conditions are met.. The search point is the center of this zoom box. The search point varies depending on the selected search mode as follows:

- **Other than Every Start**
  As with the trigger point, the search point is the falling edge of the SDA signal.



- **Mode: Every Start**
  The search point is the rising edge of the Acknowledge bit after the specified conditions are met.
  Below is an example when the Mode is set to ADR & DATA. It is similar in other search modes.



Viewing Search Points

If you select an arbitrary search point using Pattern #, the zoom position moves to the selected search point. If the zoom link function is enabled (see page 1-19), the highlight in the analysis result list also moves to the byte containing the selected search point.

Unification of the Trigger Setting of the I^{2}C Bus Signal and the Setting of the I^{2}C Bus Signal Analysis and Search

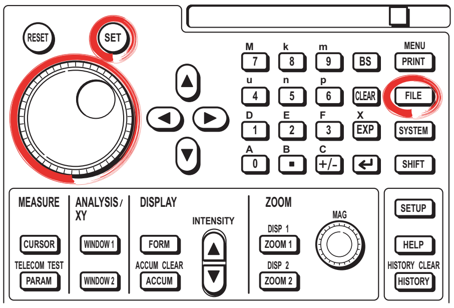
On products with software version 1.80 or higher, the trigger settings of the I^{2}C bus signal on the menu that is entered through the ENHANCED key and the settings of the I^{2}C bus signal analysis and search on the menu entered through the WINDOW1 and ZOOM1 key are common. For details, see page 1-13.

1.5 Saving the Data of the Analysis Result List

CAUTION

Do not remove the medium (disk) or turn OFF the power when the access indicator is blinking. Doing so can damage the storage medium or destroy the data on the medium.

Procedure



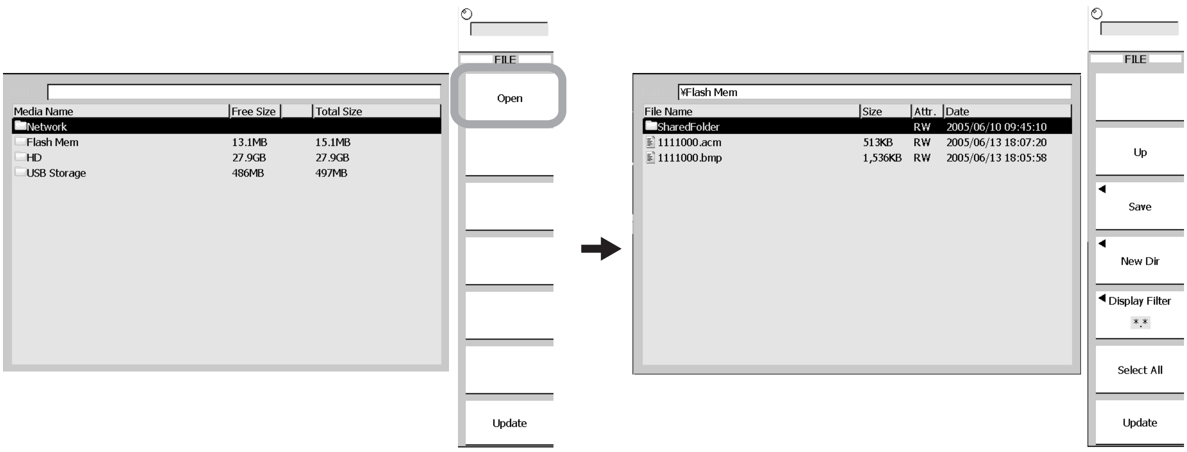
- To exit the menu during operation, press **ESC** located above the soft keys.
- For details on the operation using the rotary knob, SET, and RESET, see sections 4.1 or 4.2 in the *User's Manual (IM701310-01E).*
- For a description of the operation using a USB keyboard or a USB mouse, see section 4.3 in the *User's Manual (IM701310-01E).*

1. Press **FILE** to display the FILE menu.

Setting the Storage Conditions

Selecting the Save Destination Medium and Directory

2. Turn the **rotary knob** to select the save destination storage medium.
3. Press the **Open** soft key to confirm the medium.
   To store the data in a directory on the storage medium, select the directory in the same manner as step 2. Then, press the **Open** soft key to confirm the directory.
   The selected medium/directory is displayed in "Path=......" located in the upper-left of the File List window.
   Press the **UP** soft key to return to the next higher directory.

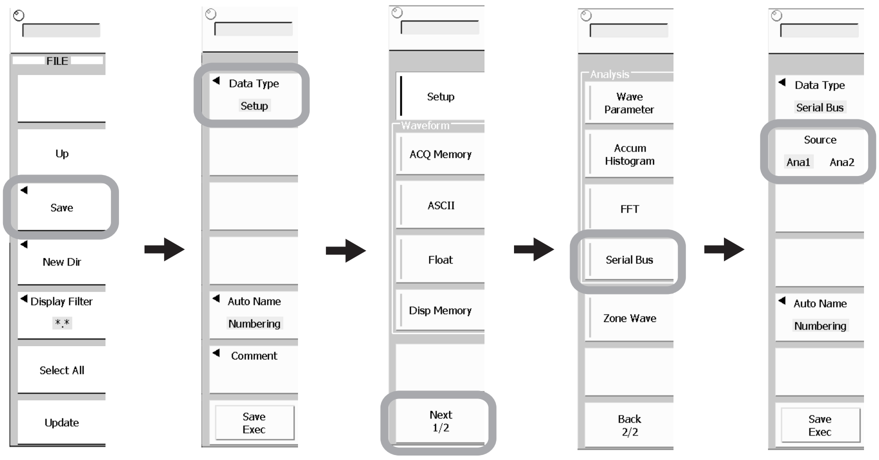


Setting the Data Type

4. Press the **Save** soft key to display the file name menu.
5. Press the **Data Type** soft key to display the data type selection menu.
6. Press the **Next 1/2**soft key.
7. Press the **Serial Bus** soft key.

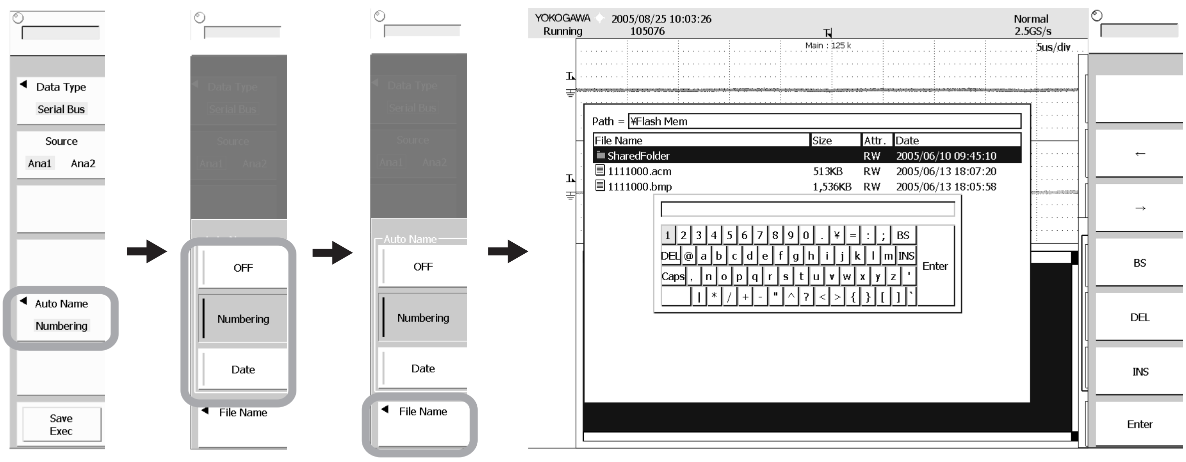
Selecting the Window to Be Saved

8. Press the **Source** soft key to select Ana1 (WINDOW1) or Ana2 (WINDOW2) to be saved.



Setting the File Name

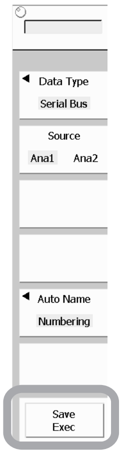
9. Press the **Auto Name** soft key to display a menu used to select the method of setting the file name.
10. Press the **OFF**, **Numbering**, or **Date** soft key.
    - If you select Numbering, a sequence number is automatically appended to the specified file name. Proceed to step 11.
    - If you select Date, the date on which the data is saved is the file name. Proceed to step 12.
11. Press the **File Name** soft key.
12. Enter the file name according to the procedure given in section 4.2 in the *User's Manual (IM701310-01E)*.
13. Press the **Enter** to confirm the entered file name.



14. Press **ESC**.

Executing the Save Operation

15. Press the **Save EXEC** soft key. The data is saved to the directory indicated by Path=...... At the same time, the Save EXEC soft key changes to the Abort soft key. A media access icon appears at the upper left corner of the screen while the save operation is in progress.
    When you abort the save operation, press the **Abort** soft key. The save operation is aborted. At the same time, the Abort soft key changes to the Save EXEC soft key.



Explanation

The data of the analysis result (simple display or detail display) can be saved to a specified storage medium in CSV format. The extension is .csv. Data in CSV format is a text file with data separated by commas. It is one of the common data formats used to convert data between spreadsheet and database applications.

Setting the Storage Conditions

Data to Be Saved (Source)

Select Ana1 or Ana2.

Ana1: Analysis result through the WINDOW1 key
Ana2: Analysis result through the WINDOW2 key

Selecting the Storage Medium and Directory (File)

The available storage medium is displayed in the File List window.

- Display examples of storage media
  [Storage Card]: PC card
  [HD]: Internal hard disk (option)
  [NetWork]: Network drive (option)
  [Flash Mem]: Internal memory
  [USB Storage]: USB storage device

Setting the File Name

The number of characters that can be used for a file or directory name is up to 64 characters from the beginning of the entered characters. However, the following conditions apply.

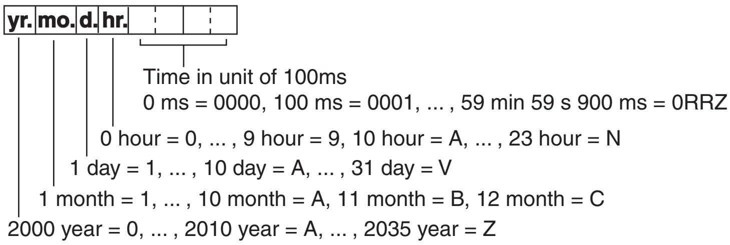
The types of characters that can be used are 0 to 9, A to Z, a to z, _, -, =, (,), {, }, [,], #, $, %, &, ~, !, ', and @ on the keyboard that appears on screen.

† Multiple @ characters cannot be entered consecutively.

- **Auto Naming**

  Select from the following three types.

  - OFF: The name specified by File Name is assigned.
  - Numbering: Files are automatically named with three digit numbers from 000 to 999. You can specify a common name (up to 5 characters, specified through File Name) that is placed before the number.
  - Date: Files are automatically named using eight characters (base 36 consisting of 0 to 9 and A to Z) based on the date and time. (The file name specified by File Name is void.)



Notes When Saving Data

The number of directories and files that can be displayed in the file list is 2500. If the number of directories and files in a directory exceeds 2500, the file list randomly displays 2500 of the directories and files.

Note

- An error occurs, if a key other than the Abort key is pressed while saving a file.
- File names are not case-sensitive. Comments are case-sensitive. In addition, the following names cannot be used due to limitations of MS-DOS.
  AUX, CON, PRN, NUL, CLOCK, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, and COM9.
- Make sure that the full path name (absolute path name from the root directory) is within 260 characters. If 260 characters is exceeded, an error occurs when you execute a file operation (save, copy, rename, create directory, etc.).
  Full path name: If you are operating a directory, specify up to the directory name. If you are operating a file, specify up to the file name.
- The length of the character string displayed in the entry box of the keyboard that is shown on the screen is 36. If you use the auto naming function of file names, the following conditions are added.
  - If you set auto naming to Numbering, eight-character file names are used consisting of the first five characters entered as the file name and a three-digit serial number.
  - If auto naming is set to Date (date/time), the characters entered as the file name are not used. The file name will only consists of the date information.

Saved Data

Up to 40000 analysis results.

Data Size

Data size* = (Number of bytes + 4) × 65 [bytes]

* The data size is a reference value. It is not strictly warranted. Use it as a guideline when saving the data.

Data Format

Binary display of the analysis result

Hexadecimal display of the analysis result

Acknowledge bit condition
1: No acknowledge
0: Acknowledge present

```
Analysis
Model Nam DL9000
Model Ver *.**
No.      S/P      Time(ms)   Binary        Hex       Form   R/W   ACK    Info
       0 S         -4.92896  11110010 Bin  F2 Hex    A      W     0      10-bit
       1           -4.92064  00001111 Bin  0F Hex    A            0
       2           -4.91264  10110000 Bin  B0 Hex    D            0
       3   P       -4.90456  01101100 Bin  6C Hex    D            0
       4 S         -4.82896  11110010 Bin  F2 Hex    A      W     0      10-bit
       5           -4.82064  00001111 Bin  0F Hex    A            0
```

Time from the trigger position

Start condition (S)/
stop condition (P)

Analysis number

Address type

Condition of the data direction bit
(W: write, R: data request)

Data format
A: address, D: data

1.6 Error Messages

A message may appear during operation. This section describes the meanings of the messages and their corrective actions. This section lists only the error messages related to the I^2C Bus signal analysis function. There are other error messages related to the DL9000 and communications. These messages are described in the *User's Manual (IM701310-01E)* and the *Communication Interface User's Manual (IM701310-17E)*.
You can set the messages to be displayed in English or Japanese. For the procedure of setting the message language, see section 17.1 in the *User's Manual IM701310-01E*.
If the corrective action requires servicing, contact your nearest YOKOGAWA dealer for repairs.

Code	Messages	Corrective Action	Page
58	Search executed is completed, but no record was found that matshed the pattern.	–	1-20
506	Save data do not exist. Check the content to be saved.	Display the analysis result, and then execute the save operation again.	1-14, 1-25

2.1 Overview of the CAN Bus Signal Analysis Function

CAN Bus Signal Analysis Function

CAN stands for Controller Area Network. It is a serial communication protocol standardized internationally by the ISO (International Organization for Standardization). By using this function, data can be analyzed while displaying the signal waveforms on the CAN bus as analog waveforms. In addition, synchronized monitoring of the data on the CAN bus and the analog waveform is possible.

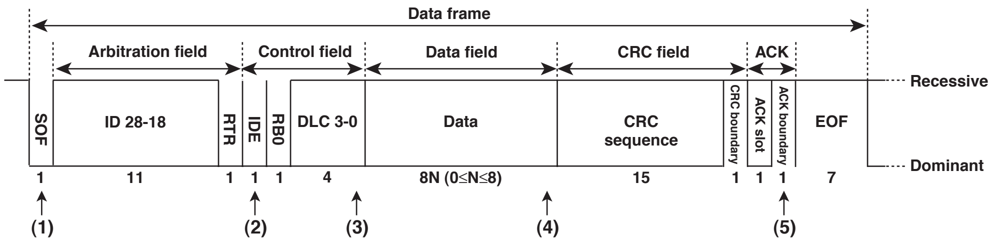
The CAN bus signal analysis function consists of the following four main functions.

Trigger Function <See page 2-5 for the operating procedure>

By setting the ID bit pattern, DLC, Data, and ACK slot status of the CAN bus, a trigger can be activated on a specific data frame or remote frame. Up to four ID/Data conditions can be specified allowing triggers to be activated on their OR conditions.

In addition, the SOF (Start of Frame) or error frame can be used as a trigger condition.

[Example] Standard Format of the Data Frame



(1) Trigger point when the trigger condition is set to SOF
(2) Trigger point when the trigger condition is set only on the ID bit pattern
(3) Trigger point when the trigger condition is set on the ID bit pattern and DLC
(4) Trigger point when the trigger condition is set on the ID bit pattern and Data bit pattern
(5) Trigger point when the trigger condition is set to the ACK slot state

A trigger can be activated on the combination of the trigger conditions of the CAN bus signal and analog signal (event interval trigger). For details on the event interval trigger, see section 6.14 in the *User's Manual (IM701310-01E)*.

Analysis Function <See pages 2-17 and 2-32 for the operating procedure>

This function analyzes the CAN bus signal data and shows a list of the analysis results. There are two types of analysis result lists: simple and detail. The simple list displays the analysis number, the type of analyzed frame, ID, Data, ACK slot status for each frame. The detail list displays the time from the trigger position, DLC, and CRC sequence in addition to the items displayed by the simple list. The data of the analysis result can be saved to an arbitrary storage medium in CSV format.

You can select an arbitrary frame in the analysis results list and automatically display the CAN bus signal for that frame (zoom link). The zoom position (the center of the zoom box) can be moved to the head of a specified field of the frame (field jump).

Search Function <See page 2-25 for the operating procedure>

A specific frame or field can be searched on the CAN bus signal data. When the search is executed, the zoom box moves to the data position where the conditions are met, and the data is displayed expanded in the zoom window (Zoom1 or Zoom2).

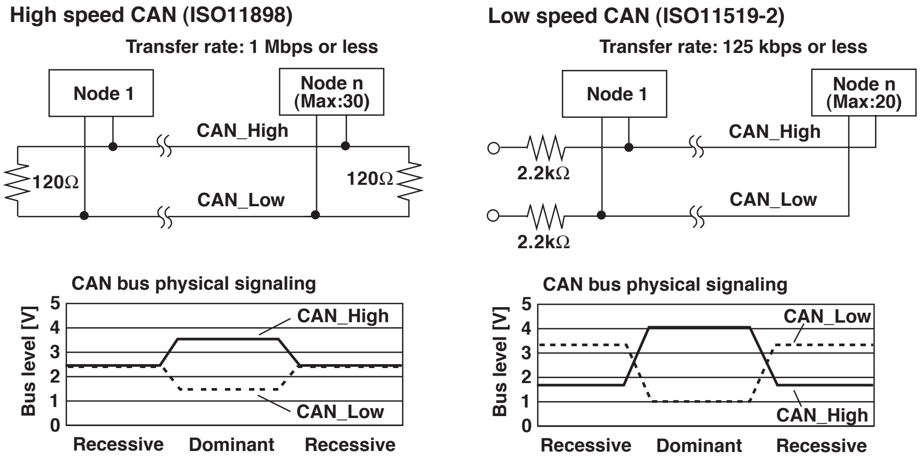
Stuff Bit Computation Function <See page 2-29 for the operating procedure>

Stuff bits within the CAN bus signals can be detected, and stuff bit waveforms can be displayed as math waveforms (stuff bit computation).

High Speed CAN (ISO11898) and Low Speed CAN (ISO11519-2)

Two typical standards of the CAN physical layer are High speed CAN (ISO11898) and Low speed CAN (ISO11519-2).

In the figure below, the bus level is determined for both High and Low speed CAN according to the difference of potential between the CAN_High and CAN_Low busses.



Connecting the Probe

Probe to Be Used

A differential probe is used when measuring CAN bus signals.

Compatible differential probes: 701920 and 701922 by Yokogawa

Probe Connection Procedure

When displaying the recessive voltage level higher than the dominant voltage level (Recessive: H)

- **For a two wire system (differential)**
  Connect the differential probe negative (–) to CAN_High, and the probe positive (+) to CAN_Low.
- **For a one wire system (single-ended)**
  Connect the differential probe negative (–) to CAN_High, and probe positive (+) to GND (ground potential).

When displaying the recessive voltage level less than the dominant voltage level (Recessive: L)

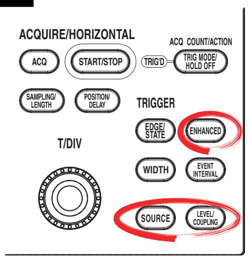
- **For a two wire system (differential)**
  Connect the differential probe negative (–) to CAN_Low, and the probe positive (+) to CAN_High.
- **For a one wire system (single-ended)***
  Connect the differential probe negative (–) to GND (ground potential), and probe positive (+) to CAN_High.

* In this case, the passive probe (model 700988) can be connected to CAN_High.

2.2 Setting the CAN Bus Signal Acquisition Conditions

With the CAN bus signal analysis function, you can acquire CAN bus signals using specific frames of the CAN bus as trigger conditions.

Procedure



- To exit the menu during operation, press **ESC** located above the soft keys.
- For details on the operation using the rotary knob, SET, and RESET, see sections 4.1 or 4.2 in the *User's Manual (IM701310-01E).*
- For a description of the operation using a USB keyboard or a USB mouse, see section 4.3 in the *User's Manual (IM701310-01E).*

Setting the Trigger Conditions of the CAN Bus Signal

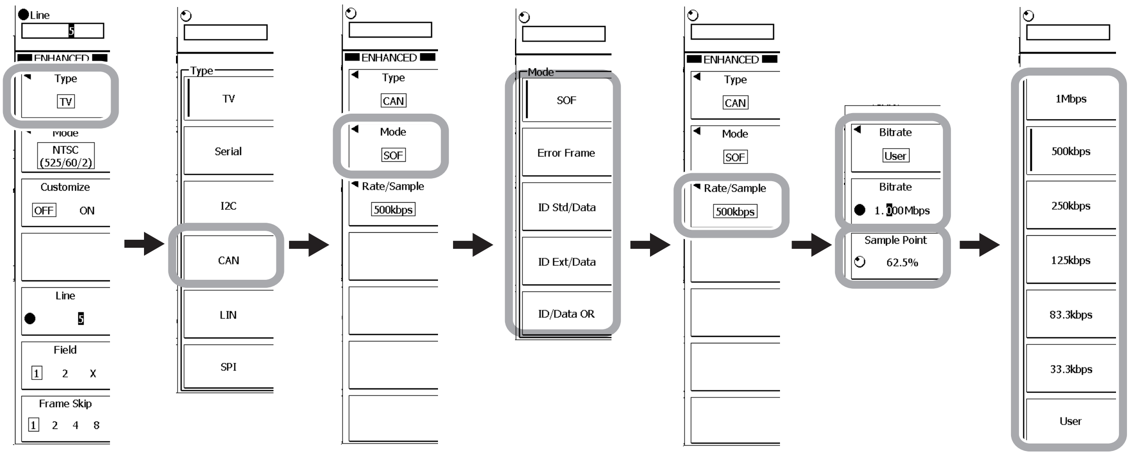
1. Press **ENHANCED**. The ENHANCED menu appears.
2. Press the **Type** soft key. The trigger type selection menu appears.
3. Press the **CAN** soft key.

Selecting the Trigger Mode

4. Press the **Mode** soft key. A menu used to select the trigger mode of the CAN bus signal appears.
5. Press the **SOF**, **Error Frame**, **ID Std/Data**, **ID Ext/Data**, or **ID/Data OR** soft key to set the trigger mode.

Setting the Bit Rate and Sample Point

6. Press the **Rate/Sample** soft key.
7. Press the **Bitrate** soft key. The bit rate selection menu appears.
8. Press the **1Mbps**, **500kbps**, **250kbps**, **125kbps**, **83.3kbps**, **33.3kbps** (33.3kbps supported by firmware version 2.40 or higher) or **User** soft key to select the bit rate. If you select User, press the **Bitrate** soft key and turn the **rotary knob** to set an arbitrary value.
9. Press the **Sample Point** soft key.
10. Turn the **rotary knob** to set the sample point.



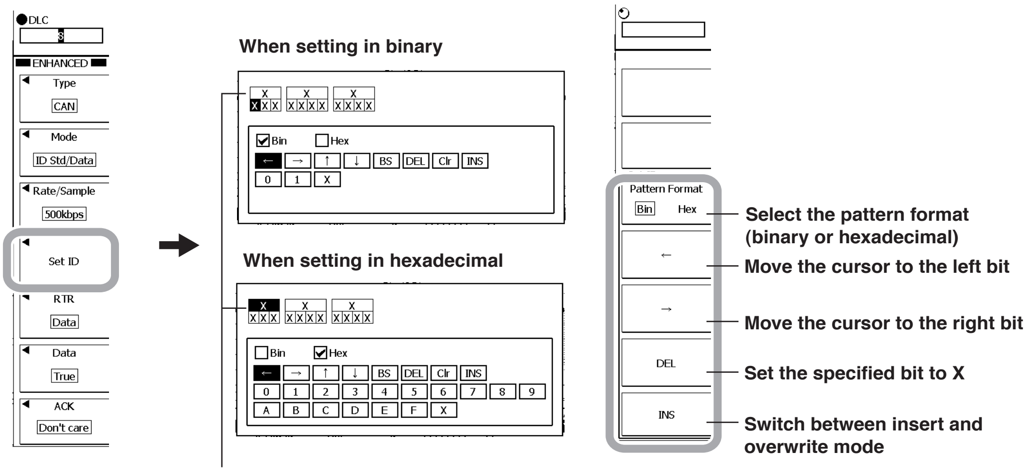
Proceed to the following pages depending on the trigger mode you selected in step 5.

• SOF (trigger on the Start of Frame):	Step 33
• Error Frame (trigger on an error frame):	Step 33
• ID Std/Data (trigger on the data/remote frame (ID: standard format)):	Step 11
• ID Ext/Data (trigger on the data/remote frame (ID: extended format)):	Step 11
• ID/Data OR (trigger on the OR condition of multiple data/remote frames):	Step 30

• If You Selected ID Std/Data or ID Ext/Data in Step 5

Setting the ID Bit Pattern

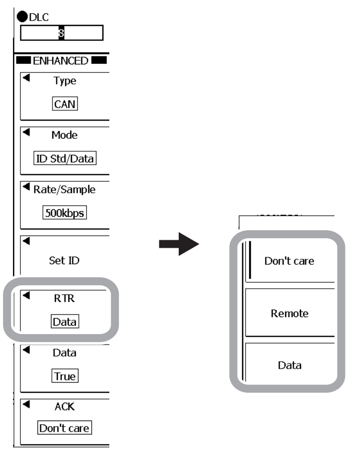
11. Press the **Set ID** soft key. The ID setup screen appears.
12. Use the **rotary knob**, **arrow** keys, and **SET** key to set the ID pattern.
    You can use the soft keys to change the format to binary or hexadecimal, move between bits, or clear all the bits (X).
13. Press **ESC**.



The figure above is the screen for ID Std/Data (ID: 11 bits).
The ID is 29 bits for ID Ext/Data.

Setting the Remote Frame or Data Frame

14. Press the **RTR** soft key.
15. Press the **Don't care**, **Remote**, or **Data** soft key.
    - If you select Don't care, a trigger is activated on a remote frame or data frame. Proceed to step 28.
    - If you select Remote, a trigger is activated on a remote frame. Proceed to step 28.
    - If you select Data, a trigger is activated on a data frame. Proceed to step 16.



- **Setting the DLC**

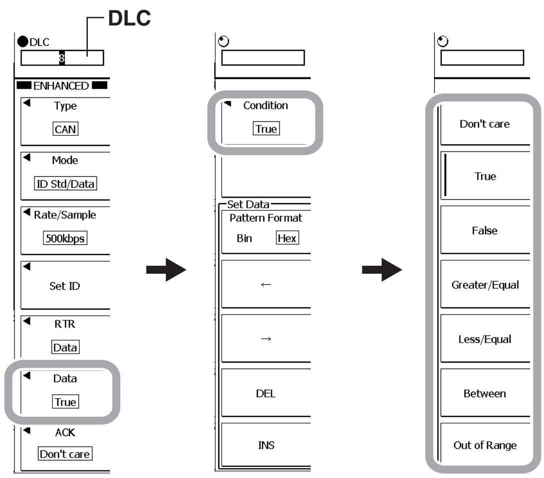
16. Turn the **rotary knob** to set the DLC (number of valid bytes).

- **Setting the Data Field Condition**

17. Press the **Data** soft key.
18. Press the **Condition** soft key.
19. Press the **Don't care**, **True**, **False**, **Greater/Equal**, **Less/Equal**, **Between**, or **Out of Range** soft key.
    - If you select Don't care, trigger is activated when the DLC matches. Proceed to step 28.
    - If you select True, a trigger is activated when Data in the data field matches the specified bit pattern. Proceed to step 20.
    - If you select False, a trigger is activated when Data in the data field does not match the specified bit pattern. Proceed to step 20.
    - If you select Greater/Equal, a trigger is activated when Data in the data field is greater than or equal to the specified comparison data. Proceed to step 22.
    - If you select Less/Equal, a trigger is activated when Data in the data field is less than or equal to the specified comparison data. Proceed to step 22.
    - If you select Between, a trigger is activated when Data in the data field is greater than or equal to the specified comparison data Data1, and less than or equal to the comparison data Data2. Proceed to step 22.
    - If you select Out of Range, a trigger is activated when Data in the data field is less than the specified comparison data Data1, or greater than the comparison data Data2. Proceed to step 22.

Note

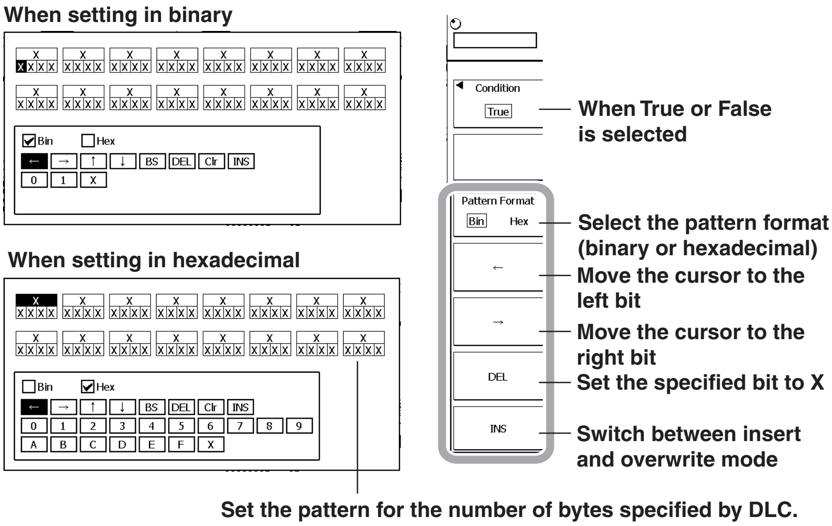
For details on the trigger point for each condition, see page 2-15.



- **Setting the Data Bit Pattern**

  If you selected True or False in step 19, a screen used to set the Data bit pattern appears.

  20. Use the **rotary knob**, **arrow** keys, and **SET** key to set the Data bit pattern. You can use the soft keys to change the format to binary or hexadecimal, move between bits, or clear all the bits (X).
  21. Press **ESC**. Proceed to step 28.

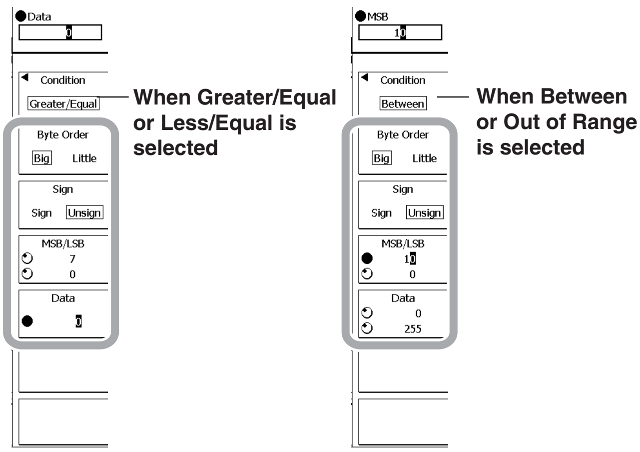


- **Setting the Comparison Data**

  If you select Greater/Equal, Less/Equal, Between, or Out of Range in step 19, a menu used to set the comparison data appears.

  22. Press the **Byte Order** soft key to set the Endian of the value to Big or Little.
  23. Press the **Sign** soft key to set the sign of the value to Sign (with a sign) or Unsign (without a sign).
  24. Press the **MSB/LSB** soft key to set the MSB and LSB positions of the comparison data.
  25. Press the **Data** soft key.
  26. Turn the **rotary knob** to set the comparison data (Data, Data1, and Data2) in decimal notation.

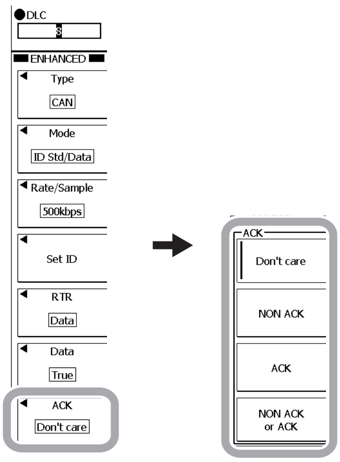
      If you select Greater/Equal or Less/Equal, set Data. If you select Between or Out of Range, set Data1 and Data2.
  27. Press **ESC**. Proceed to step 28.



Setting the ACK Slot

28. Press the **ACK** soft key. The ACK slot setup menu appears.
29. Press the **Don't care**, **NON ACK**, **ACK**, or **NON ACK or ACK** soft key.
    - If you select Don't care, the bus level of the ACK slot is not used as a trigger condition.
    - If you select NON ACK, a trigger is activated when the bus level of the ACK slot is recessive.
    - If you select ACK, a trigger is activated when the bus level of the ACK slot is dominant.
    - If you select NON ACK or ACK, a trigger is activated when the ACK slot is recessive or dominant.

Proceed to step 33.



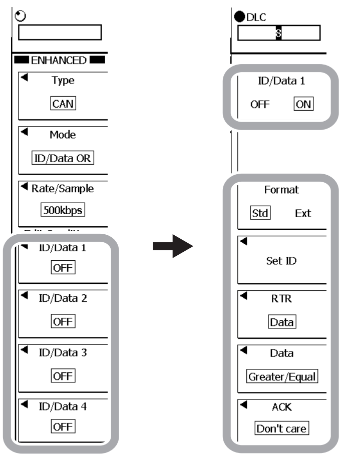
- **If You Selected ID/Data OR in Step 5**

Setting the ID/Data for the OR Condition

30. Press the **ID/Data 1**, **ID/Data 2**, **ID/Data 3**, or **ID/Data 4** soft key to select the ID/Data on which to set the OR condition.
31. Press the **ID/Data X** soft key (where X is a number between 1 and 4) to select ON (include in the OR condition) or OFF (not include in the OR condition).
32. Press the **Format** soft key to set the ID format to Std (standard format) or Ext (extended format).

The subsequent steps are the same as when you select ID Std/Data or ID Ext/Data in step 5. For the procedure, see steps 11 to 29 on page 2-4.
To activate a trigger on the OR condition of multiple ID/Data, repeat steps 30 to 32.

Proceed to step 33.



Setting the Source Channel

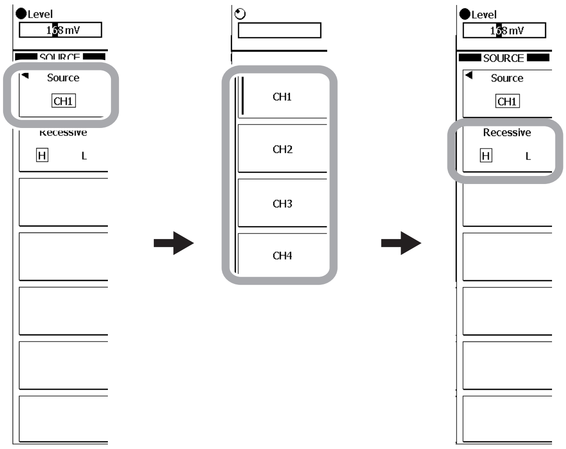
33. Press **SOURCE**. The SOURCE menu appears.

Selecting the Source Channel

34. Press the **Source** soft key. A menu used to select the source channel of the CAN bus signal appears.
35. Press any of the soft keys **CH1** to **CH4** to select the source channel.

Selecting the Bus Level

36. Press the **Recessive** soft key to select H (set the recessive electric potential higher than the dominant electric potential) or L (set the recessive electric potential less than the dominant electric potential).



Setting the Level, Coupling, HF Rejection, and Hysteresis

37. Press **LEVEL/COUPLING**. The LEVEL/COUPLING menu appears.

Selecting the Channel to Be Configured

38. Press the **CH** soft key. The menu used to select the channel appears.
39. Press any of the soft keys, **CH1** to **CH4**.

Setting the Level

40. Turn the **rotary knob** to set the level used to determine high/low.

Selecting the Coupling

41. Press the **Coupling** soft key to select DC.

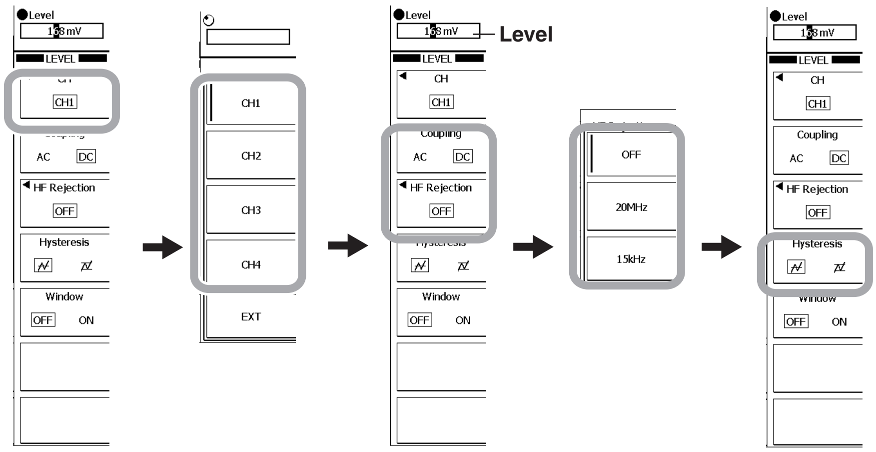
Selecting the HF Rejection

42. Press the **HF Rejection** soft key. The menu used to select the HF rejection appears.
43. Press the **OFF**, **20MHz**, or **15kHz** soft key.

Selecting the Hysteresis

44. Press the **Hysteresis** soft key to select the hysteresis.

As necessary, repeat steps 38 to 44.



Explanation

Setting the Trigger Conditions of the CAN Bus Signal

Trigger Mode

The following trigger modes are available.

SOF	Activate a trigger on the Start of Frame (SOF).
Error Frame	Activate a trigger an error frame.
ID Std/Data	Activate a trigger on a data frame or remote frame (ID: standard format) that matches the specified conditions.
ID Ext/Data	Activate a trigger on a data frame or remote frame (ID: extended format) that matches the specified conditions.
ID/Data OR	Activate a trigger on the OR condition of four types of data frames or remote frames. You can select standard or extended format for each ID.

For details on the trigger point for each trigger mode, see page 2-15.

Bitrate

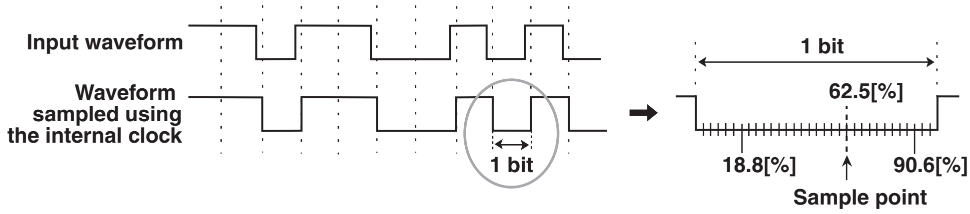
Select the transfer rate of the CAN bus signal from 1Mbps, 500kbps, 250kbps, 125kbps, 83.3kbps, 33.3kbps (33.3kbps supported by firmware version 2.40 or higher) and User. If you select User, you can set an arbitrary value in the range of 10k to 1M [bps] (0.1 kbps resolution).

Sample Point

Set the point used to judge the bus level (recessive or dominant) in the range of 18.8 to 90.6 [%].

In the trigger circuits of the instrument's CAN bus signal analysis function, the input CAN bus signal is sampled once per the internal clock, and the point of change from recessive to dominant is detected. The detected point is taken as 0%, and the point one bit time (the reciprocal of the specified bit rate) thereafter is taken as 100%, allowing expression of the sample point as a percentage.

When the sample point is set to 62.5 [%]



• When the Trigger Mode Is Set to ID Std/Data or ID Ext/Data

ID Bit Pattern (Set ID)

Set an 11-bit ID pattern if you select ID Std/Data or a 29-bit pattern if you select ID Ext/Data. Binary bit patterns are set using 0, 1, and X. Hexadecimal bit patterns are set using 0 to 9, A-F, and X. Matching of the specified ID bit pattern is a trigger condition.

Note

- If an ID bit pattern is set to X, the condition is assumed to be met regardless of the status of the corresponding bit.
  Binary: Condition met on either 0 or 1
  Hexadecimal: Condition met on any value (0 to 9 and A to F)
- If there is at least one "X" bit in a group of four bits of the ID bit pattern in the binary display, the corresponding hexadecimal display will show a "$."

Remote Frame or Data Frame (RTR)

A trigger can be activated on the combination of remote frames or data frames and the ID bit pattern.

Don't care	Remote frames and data frames are used as trigger conditions.
Remote	Remote frames are used as trigger conditions.
Data	Data frames are used as trigger conditions.

The items below are set only when RTR is set to Data.

- **DLC**

  Set the DLC (number of valid bytes) of the data field in the range of 0 to 8 bytes. The default value is 8 bytes.

- **Data Field Condition**

  Select the data condition when using the data field as a trigger condition from the following:

Don't care	A trigger is activated when the DLC matches.
True	A trigger is activated when Data in the data field matches the specified bit pattern.
False	A trigger is activated when Data in the data field does not match the specified bit pattern.
Greater/Equal	A trigger is activated when Data in the data field is greater than or equal to the specified comparison data.
Less/Equal	A trigger is activated when Data in the data field is less than or equal to the specified comparison data.
Between	A trigger is activated when Data in the data field is greater than or equal to the specified comparison data Data1, and less than or equal to the comparison data Data2.
Out of Range	A trigger is activated when Data in the data field is less than the specified comparison data Data1, or greater than the comparison data Data2.

If Data1 is set equal to Data2 for Between, a trigger is activated when the data of the data field is equal to Data1 which is equal to Data2.

Data Bit Pattern (Set Data)

Set the bit pattern only when Condition is set to True or False.

Set the Data bit pattern. Binary bit patterns are set using 0, 1, and X. Hexadecimal bit patterns are set using 0 to 9, A-F, and X. The length of the bit pattern is the number of bytes set by DLC.

Note

- If an Data bit pattern is set to X, the condition is assumed to be met regardless of the status of the corresponding bit.
  Binary: Condition met on either 0 or 1
  Hexadecimal: Condition met on any value (0 to 9 and A to F)
- If there is at least one "X" bit in a group of four bits of the Data bit pattern in the binary display, the corresponding hexadecimal display will show a "$."

Comparison Data

Set the items below if Condition is set to Greater/Equal, Less/Equal, Between, or Out of Range.

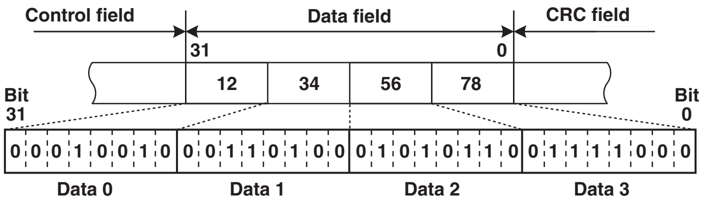
- **Data, Data1, and Data2**

  Set the value (Data, Data1, and Data2) that is compared against the Data in the data field in decimal notation. Set Data for Greater/Equal or Less/Equal. Set Data1 and Data2 for Between or Out of Range.
  For the selectable ranges of Data, Data1, and Data2, see "Sign" below.

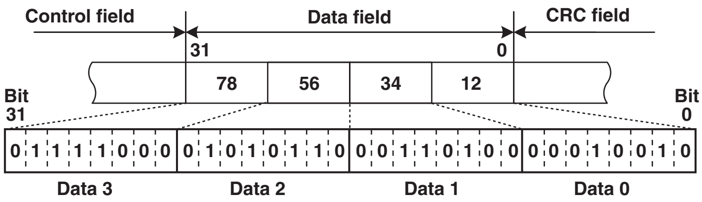
- **Byte Order**

  Set the byte read order of the Data, Data1, and Data2 to Big Endian or Little Endian.
  For example, when a node transmits four-byte data (12345678: hexadecimal notation), the image of the frame on the bus is as shown in the following figure.

• Big Endian



• Little Endian



- **Sign**

  Select whether to make Data, Data1, and Data2 signed numbers.
  The selectable range of Data, Data1, and Data2 when you select Sign or Unsign is shown below.

Sign	–9E+18 to 9E+18
Unsign	0 to 9E+18

The Data, Data1, and Data2 values are displayed using exponential notation when the values exceed seven digits (example: 1234567E+10).

- **MSB/LSB**

  Set the bit positions of Data, Data1, and Data2 to be compared using MSB and LSB. The selectable range is 0 to 63 (the selectable range may be limited depending on the DLC setting).

  For example, to compare only bits 5 to 20 of the four-byte data (12345678: hexadecimal notation), set LSB to 5 and MSB to 20. The bits of the data field that are compared for this case are as shown in the figure below for the different byte order settings (Big Endian and Little Endian).

• Big Endian

Bit 31 … **Bit 0**

Data 0	Data 1	Data 2	Data 3
0 0 0 1 0 0 1 0	0 0 1 1 0 1 0 0	0 1 0 1 0 1 1 0	0 1 1 1 1 0 0 0

• Little Endian

Bit 31 … **Bit 0**

Data 3	Data 2	Data 1	Data 0
0 1 1 1 1 0 0 0	0 1 0 1 0 1 1 0	0 0 1 1 0 1 0 0	0 0 0 1 0 0 1 0

Note

Relationship between the DLC, MSB, LSB, Data, Data1, and Data2 values

The relationship between the DLC, MSB, LSB, Data, Data1, and Data2 values is shown below.

DLC → **MSB/LSB** → **Data/Data1/Data2** ← **Sign**

DLC	MSB/LSB	Data/Data1/Data2	Sign
(Selectable range: 0 to 8)	**Selectable range limited by the DLC setting (Selectable range: 0 to 63)**	**Selectable range limited bythe MSB/LSB and Sign settings (Selectable range: ±9E+18)**	**(With or without a sign)**

[Example] When DLC is set to 2 bytes

$0 \le$ LSB $\le$ MSB ≤ 15(equivalent to 2 bytes)

• Sign (signed number): $-32768 \le$ Data1 $\le$ Data2 ≤ 32767

• Unsign (unsigned number): $0 \le$ Data1 $\le$ Data2 ≤ 65535

ACK Slot

The bus level of the ACK slot (recessive or dominant) can be used as a trigger condition.

Don’t care	The bus level of the ACK slot is not used as a trigger condition.
NON ACK	A trigger is activated when the bus level of the ACK slot is recessive.
ACK	A trigger is activated when the bus level of the ACK slot is dominant.
NON ACK or ACK	A trigger is activated when the bus level of the ACK slot is recessive or dominant.

Note

If the trigger mode is set to ID/Data OR, set conditions so that the trigger points of all data frames and remote frames (ID/Data1 to ID/Data4) are the same (conditions that correspond to (2), (3), (4), (5), (2)', (3)', or (4)' in the figure on page 2-15).

Otherwise, triggers may not be activated at the correct position.

- **When the Trigger Mode Is Set to ID/Data OR**

IDs Take the OR Logic (Edit Condition)

Set four trigger conditions (ID/Data1 to ID/Data4) of data frames or remote frames. You can enable or disable each ID/Data condition. You can select standard or extended format for each ID.

ID Format

Set the ID format to Std (standard format) or Ext (extended format).

ID Bit Pattern (Set ID)

The details are the same as when the trigger mode is set to ID Std/Data or ID Ext/Data. For details, see page 2-11.

Remote Frame or Data Frame (RTR)

The details are the same as when the trigger mode is set to ID Std/Data or ID Ext/Data. For details, see page 2-11.

ACK Slot

The details are the same as when the trigger mode is set to ID Std/Data or ID Ext/Data. For details, see page 2-13.

Source Channel

The source channel of the CAN bus signal is set using the SOURCE menu that appears when you press the SOURCE key. The menu used to set the source channel of the CAN bus signal appears only when Type is set to CAN in the ENHANCED menu.

Specifying the Source Channel

The source channel is specified using the SOURCE menu.

Bus Level (Recessive)

Select the bus level from either of the following: In either case, the logical value is: recessive = 1 and dominant = 0.

H	The recessive electric potential is set higher than the dominant electric potential.
L	The recessive electric potential is set lower than the dominant electric potential.

Trigger Level, Trigger Coupling, Etc.

Set the trigger level, trigger coupling, HF rejection, and hysteresis of each channel. For details on these items, see the *User's Manual (IM701310-01E)*.

Unification of the Trigger Setting of the CAN Bus Signal and the Setting of the CAN Bus Signal Analysis, Search and Stuff Bit Computation

The trigger settings of the CAN bus signal on the menu that is entered through the ENHANCED key and the settings of the CAN bus signal analysis, search, and stuff bit computation on the menu entered through the WINDOW1, ZOOM1, or M1 key are common. The CH1 to CH4 sources are the waveforms of which the settings are common.

Trigger settings of the CAN bus signal that are applied to the settings of analysis, search, and stuff bit computation
Level (trigger level), Hysteresis (trigger hysteresis), and Sample Point (The settings of the CAN bus signal analysis, search, and stuff bit computation are not applied to the trigger settings.)

Items for which the trigger settings, analysis settings, search settings, and stuff bit computation settings of the CAN bus signal are applied mutually
Source, Bitrate, and Recessive

Note

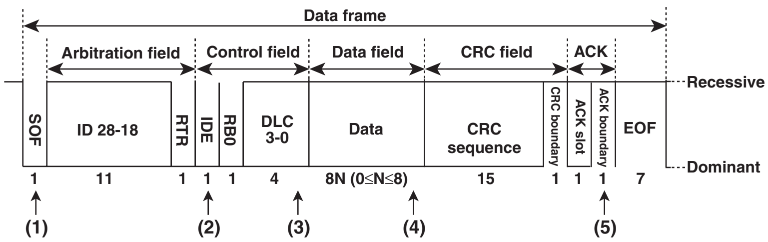
- The settings of the CAN bus signal analysis, search, and stuff bit computation on the menu entered through the WINDOW2, ZOOM2, and M2 to M4 keys are not made common. They are independent settings.
- If you change common items (excluding Level, Hys, and Sample Point) in the analysis, search, and stuff bit computation menus while the waveform acquisition is in progress and the trigger type is set to CAN, the waveform acquisition is restarted.
- Even if something other than CAN bus is selected in the analysis or search menu, the level and hysteresis settings of the analysis and search menus are set to the same value if the level or hysteresis setting is changed.
- If the trigger level or hysteresis is changed by executing auto setup, the level and hysteresis settings of the analysis and search are also set to the same new value. This also applies when the setup information is initialized.
- Trigger Hys and correspond to 0.6 division and 1.0 division, respectively, of the analysis, search, or stuff bit computation.

Frame Format and Trigger Point

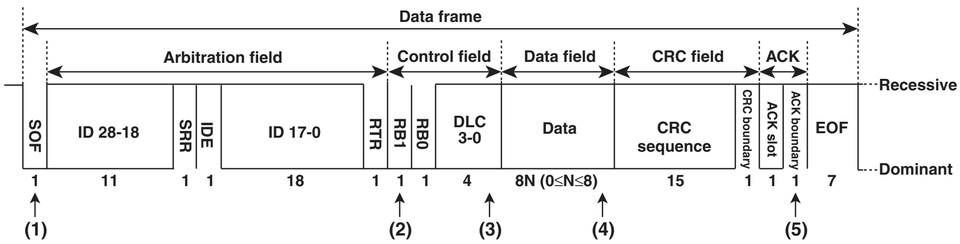
The following figure shows the format and trigger point of each frame.

Data Frame

• Standard format



• Extended format

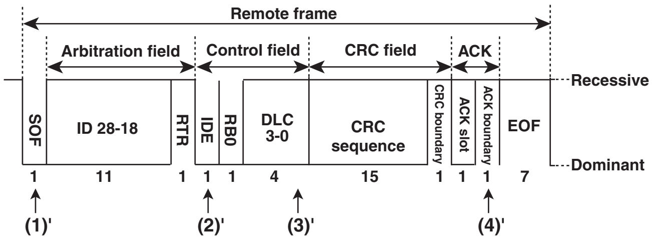


Items (1) to (5) above are the trigger points for the following conditions.
(1) Mode: SOF
(2) Mode: ID X*, RTR: Don't care, and ACK: Don't care
(3) Mode: ID X*, RTR: Data, Condition: Don't care, and ACK: Don't care
(4) Mode: ID X*, RTR: Data, Condition: Other than Don't care, ACK: Don't care
(5) ACK: Other than Don't care

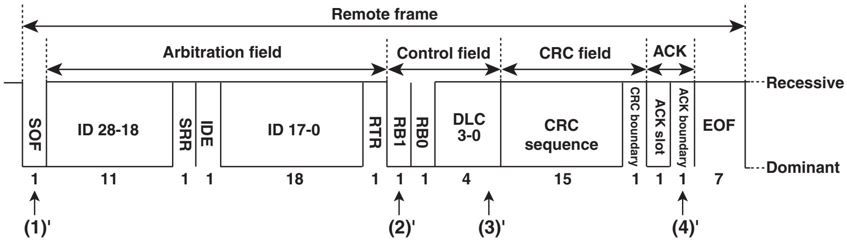
*** ID X: For ID Std/Data, ID Ext/Data, or ID/Data OR**

Remote Frame

• Standard format



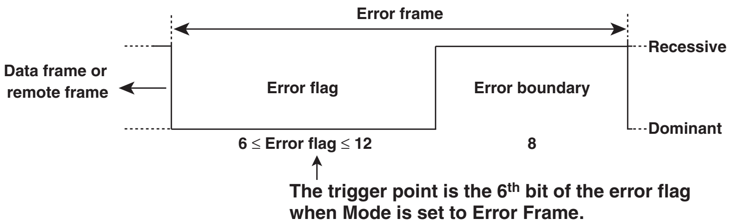
• Extended format



Items (1) to (5) above are the trigger points for the following conditions.
(1)' Mode: SOF
(2)' Mode: ID X*, RTR: Don't care, ACK: Don't care
(3)' Mode: ID X*, RTR: Remote, ACK: Don't care
(4)' ACK: Other than Don't care

*** ID X: For ID Std/Data, ID Ext/Data, or ID/Data OR**

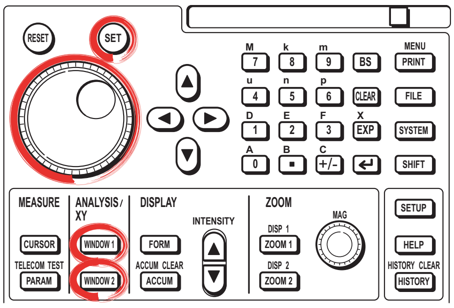
Error Frame



2.3 Analyzing the Data

By setting analysis conditions, the CAN signal data stored to the acquisition memory can be analyzed.

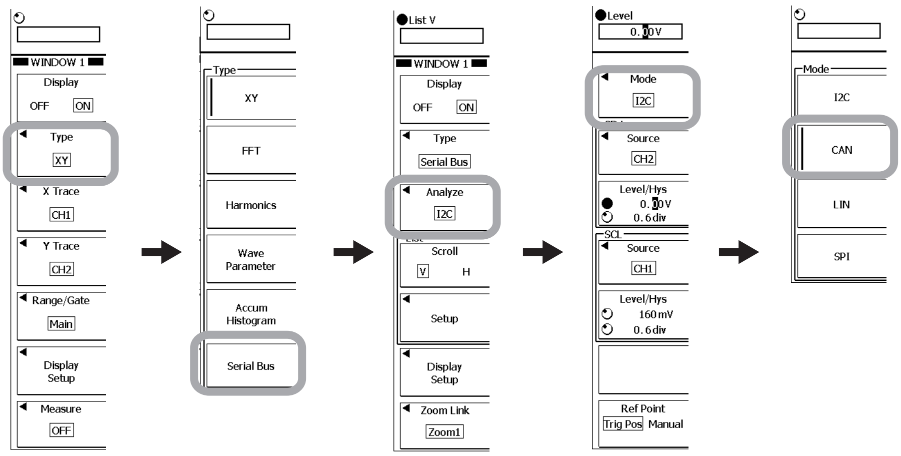
Procedure

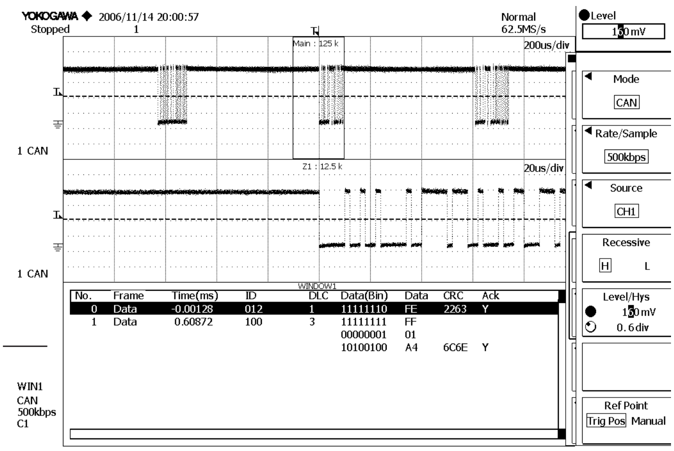


- To exit the menu during operation, press **ESC** located above the soft keys.
- For details on the operation using the rotary knob, SET, and RESET, see sections 4.1 or 4.2 in the *User's Manual (IM701310-01E).*
- For a description of the operation using a USB keyboard or a USB mouse, see section 4.3 in the *User's Manual (IM701310-01E).*

Setting the Analysis Conditions

1. Press **WINDOW1** or **WINDOW2**.
2. Press the **Type** soft key. The Type menu appears.
3. Press the **Serial Bus** soft key.
4. Press the **Analyze** soft key. The Analyze setup menu appears.
5. Press the **Mode** soft key. The Mode setup menu appears.
6. Press the **CAN** soft key.





No.	Frame	Time(ms)	ID	DLC	Data(Bin)	Data	CRC	Ack
0	Data	-0.00128	012	1	11111110	FE	2263	Y
1	Data	0.60872	100	3	11111111	FF		
					00000001	01		
					10100100	A4	6C6E	Y

Analysis result display

Setting the Bit Rate and Sample Point

7. Press the **Rate/Sample** soft key.
8. Press the **Bitrate** soft key. The bit rate selection menu appears.
9. Press the **1Mbps**, **500kbps**, **250kbps**, **125kbps**, **83.3kbps**, **33.3kbps** (33.3kbps supported by firmware version 2.40 or higher) or **User** soft key to select the bit rate. If you select User, press the **Bitrate** soft key and turn the **rotary knob** to set an arbitrary value.
10. Press the **Sample Point** soft key.
11. Turn the **rotary knob** to set the sample point.
12. Press **ESC**.

Setting the Source Channel

13. Press the **Source** soft key. The source channel setup menu appears.
14. Select the source channel from CH1 to CH4 and M1 to M4.

Selecting the Bus Level

15. Press the **Recessive** soft key to select H (set the recessive electric potential higher than the dominant electric potential) or L (set the recessive electric potential less than the dominant electric potential).

Setting the Level

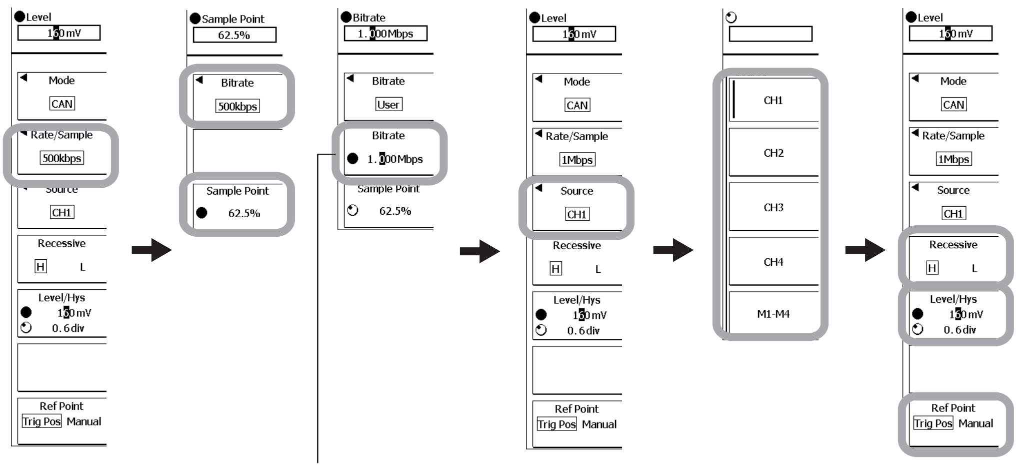
16. Press the **Level/Hys** soft key to activate Level.
17. Turn the **rotary knob** to set the level.

Setting the Hysteresis

18. Press the **Level/Hys** soft key to activate Hys (hysteresis).
19. Turn the **rotary knob** to set the hysteresis.

Setting the Analysis Reference Point

20. Press the **Ref Point** soft key to select Trig Pos (set the analysis reference point to the trigger point) or Manual (set the analysis reference point manually).
If you select Manual, turn the **rotary knob** to set the analysis reference point in the range of ±5.00 divisions.



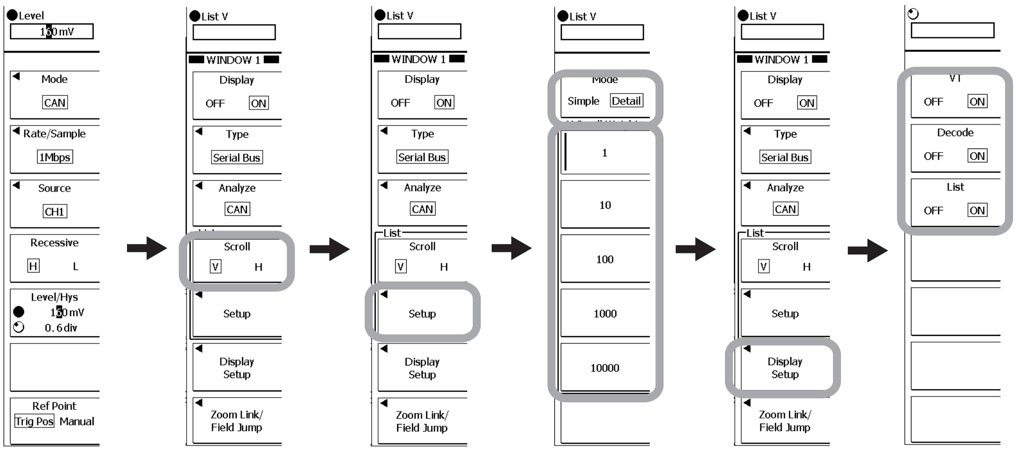
Displayed only when Bitrate is set to User.

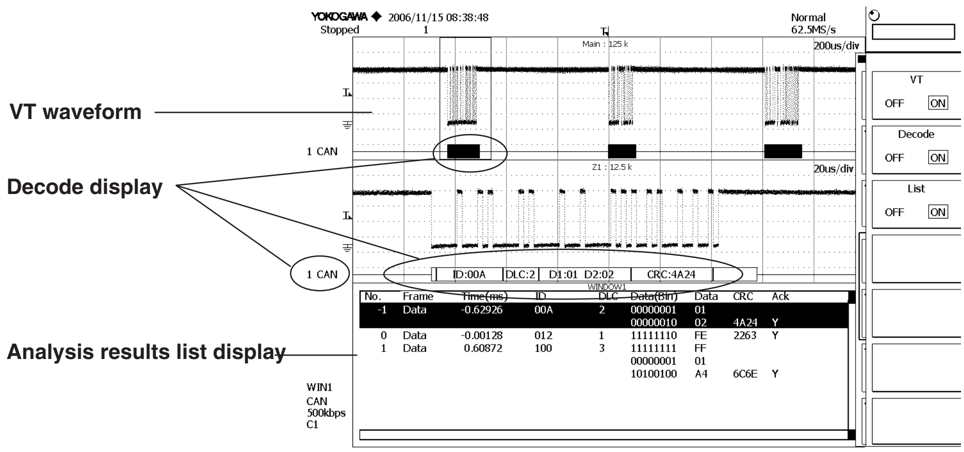
Display the Analysis Results

21. Press **ESC**.
22. Press the **Scroll** soft key to select V (scroll vertically) or H (scroll horizontally).
23. Press the **Setup** soft key. A menu used to set the analysis screen appears.
24. Press the **Mode** soft key to select Simple (simple display) or Detail (detailed display).
25. Press any of the soft keys **1**, **10**, **100**, **1000**, or **10000** to set the step number for scrolling. The display scrolls according to the specified number.
26. Press **ESC**.
27. Press the **Display Setup** soft key. The Display Setup menu appears.
28. Press the **VT** soft key to select ON or OFF. If ON is specified, the waveform is displayed on a normal voltage-time axis.
29. Likewise, press the **Decode** or **List** soft key to select ON or OFF. If set to ON, the decoded data or a list of analysis results is displayed.
30. Press **ESC**.

Note

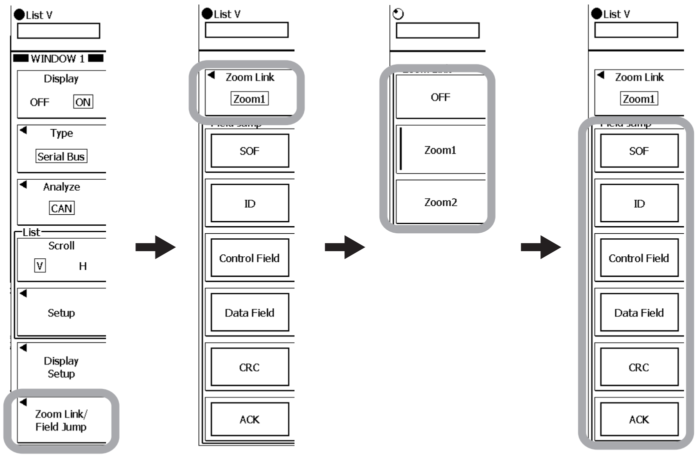
The data of the simple display and detail display of the analysis results can be saved directly to a storage medium in CSV format (.csv extension). For details, see section 2.6.





Zoom Link and Field Jump

31. Press the **Zoom Link/Field Jump** soft key. The Zoom Link/Field Jump setup menu appears.
32. Press the **Zoom Link** soft key.
33. Press the **OFF**, **Zoom1**, or **Zoom2** soft key to set the link to the zoom window. If you select Zoom1 or Zoom2, proceed to step 34.
34. Press the **SOF**, **ID**, **Control Field**, **Data Field**, **CRC**, or **ACK** soft key under Field Jump.
    If you select Zoom1 or Zoom2, set Scroll to V and turn the rotary knob to move the zoom position (the center of the zoom box) to the head of the frame highlighted in the analysis result list. If you press a soft key under Field Jump, the zoom position moves to the head of the corresponding field in the highlighted frame.



Explanation

Setting the Analysis Conditions

You can set the following conditions.

Bitrate

Select the transfer rate of the CAN bus signal from 1Mbps, 500kbps, 250kbps, 125kbps, 83.3kbps, 33.3kbps (33.3kbps supported by firmware version 2.40 or higher) and User. If you select User, you can set an arbitrary value in the range of 10k to 1M [bps] (0.1 kbps resolution).

Sample Point

The point at which the input CAN bus signal waveform changes from recessive to dominant is taken as 0%, and the point one bit time (the reciprocal of the specified bit rate) thereafter is taken as 100%, thereby allowing expression of the sample point as a percentage. The selectable range is 18.8 to 90.6 [%].
The details are the same as the trigger condition settings. See page 2-10.

Source Channel

Select the source channel from CH1 to CH4 and M1 to M4. The source channel can be set arbitrarily.

Bus Level (Recessive)

Select the bus level from either of the following: In either case, the logical value is: recessive = 1 and dominant = 0.

H	The recessive electric potential is set higher than the dominant electric potential.
L	The recessive electric potential is set lower than the dominant electric potential.

Level

Set the level used to determine the data channel signal level (0 or 1).

Selectable range: ±10 divisions around the vertical position
Resolution: 0.01 divisions (For example, the resolution for 2 mV/div is 0.02 mV.)

Hysteresis (Hys)

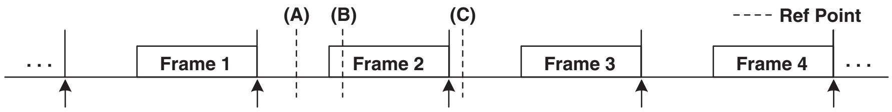
Selectable range: 0.0 divison to 4.0 divisions
Resolution: 0.1 divisions

Analysis Reference Point (Ref Point)

Select the analysis reference point from the following:

Trig Pos	Sets the trigger position to the analysis reference point.
Manual	Manually set the analysis reference point.Selectable range: !s!±5.00 divisions, resolution: 0.01 divisions

Depending on the analysis reference point (Ref Point), the No. 0 frame in the analysis result list is as follows:



(A): No. 0 frame -> Frame 2 (frame 1 is No. -1, frame 3 is No. 1, frame 4 is No. 2)
(B): No. 0 frame -> Frame 2 (frame 1 is No. -1, frame 3 is No. 1, frame 4 is No. 2)
(C): No. 0 frame -> Frame 3 (frame 1 is No. -2, frame 2 is No. -1, frame 4 is No. 1)

Unification of the Trigger Setting of the CAN Bus Signal and the Setting of the CAN Bus Signal Analysis, Search and Stuff Bit Computation

The trigger settings of the CAN bus signal on the menu that is entered through the ENHANCED key and the settings of the CAN bus signal analysis, search, and stuff bit computation on the menu entered through the WINDOW1, ZOOM1, or M1 key are common. For details, see page 2-15.

Analyzed Data

The following data can be analyzed:

- Waveform Data
  Analysis can be performed regardless of whether the waveform acquisition is in progress. If the waveform acquisition is in progress, the analysis results are updated in sync with the displayed waveform. The waveform data stored in the history memory (waveform data of the record number selected using HISTORY menu > Select) is also analyzed.
- Loaded acquisition data (ACQ data).

Analyzed Frames

The following four types of frames are analyzed.

- **Data Frame**
  Detects the ID, Data, and CRC values and the presence of ACK.
- **Remote Frame**
  Detects the ID and CRC values and the presence of ACK.
- **Error Frame**
  Detects error flags.
- **Overload Frame**
  Detects overload flags.

Analysis Range

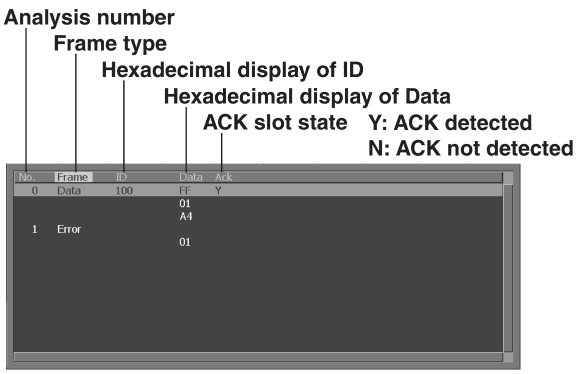
Analysis is performed on up to 3000 frames before and after the trigger source frame.

Analysis Data List (Analysis Result List)

The following information is listed.

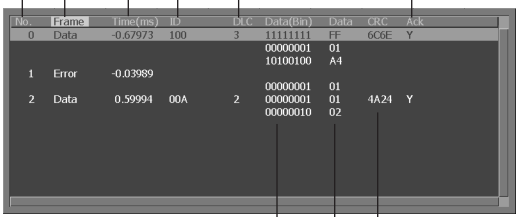
Simple Display

- Analysis Number (No.)
- Frame type
- Hexadecimal display of ID
- Hexadecimal display of Data
- ACK slot state



Detail Display

- Analysis Number (No.)
- Frame type
- Time from the trigger position (Time (ms))
- Hexadecimal display of ID
- Hexadecimal display of DLC
- Binary display of Data (Bin)
- Hexadecimal display of Data
- Hexadecimal display of the CRC sequence
- ACK slot state



- **Analysis Number (No.)**
  Points before the analysis reference point (Ref Point) are counted as No. –1, No. –2, and so on. Points before the analysis reference points are counted as No. 0, No. 1, No. 2, and so on. Up to 3000 analysis results can be displayed in the range of –2999 to 2999.
  Press the RESET key to highlight the No. 0 frame.

- **Frame Type**
  The frames that can be analyzed are data frames, remote frames, error frames, and overload frames.

- **Time from the trigger position (Time (ms))**
  Displays the time from the trigger position to the head of the frame in ms.

- **ID**
  Displays the ID value (standard format (11 bits) or extended format (29 bits)) in hexadecimal notation.

- **DLC**
  Displays the DLC (number of valid bytes) in hexadecimal notation.

- **Data**
  Displays Data in hexadecimal or binary notation (only for Detail). The frame type is displayed only for data frames. Each byte is displayed on a line.

- **CRC Sequence**
  Displays the CRC sequence in hexadecimal notation (only for Detail). The frame type is displayed only for data frames and remote frames.

- **ACK Slot State**
  Displays “Y” when ACK is detected and “N” when ACK is not detected.

Decode Display

Decodes and displays the values of each field of the CAN bus signal.

ID:	Displays the value in hexadecimal notation in light green.
DLC:	Displays the value in hexadecimal notation in pink.
Data:	Displays the data in hexadecimal notation in cyan.
CRC sequence:	Displays the value in hexadecimal notation in light blue.
Alarm Frame:	Displays in red.
Overload Frame:	Displays in green.
Other (SOF, ACK, EOF):	Displays in gray.

Zoom Link and Field Jump

Zoom Link

Select from the following:

OFF	Disables the zoom link function.
Zoom1	Links to Zoom1.
Zoom2	Links to Zoom2.

By default, WINDOW1 and WINDOW2 are linked to Zoom1 and Zoom2, respectively.
If Zoom1 or Zoom2 is selected and you select (highlight) an arbitrary frame in the analysis result list, the zoom position (the center of the zoom box) of Zoom1 or Zoom2 moves to the head of that frame. Conversely, if you change the zoom position of Zoom1 or Zoom2, the highlight in the analysis result list moves to the frame that is shown in the zoom box of Zoom1 or Zoom2.

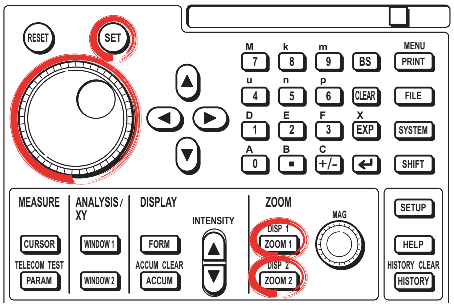
Field Jump

When the zoom link function is enabled (Zoom Link set to Zoom1 or Zoom2), the zoom position can be moved to the head of the specified field of the frame that is highlighted in the analysis result list. Select the field from SOF, ID, Control Field, Data Field, CRC, or ACK.

2.4 Searching the Waveform

You can search waveforms that match the specified conditions among the acquired waveforms. The setup procedure of search conditions is the same as the setup procedure of trigger conditions explained in section 2.2.

Procedure



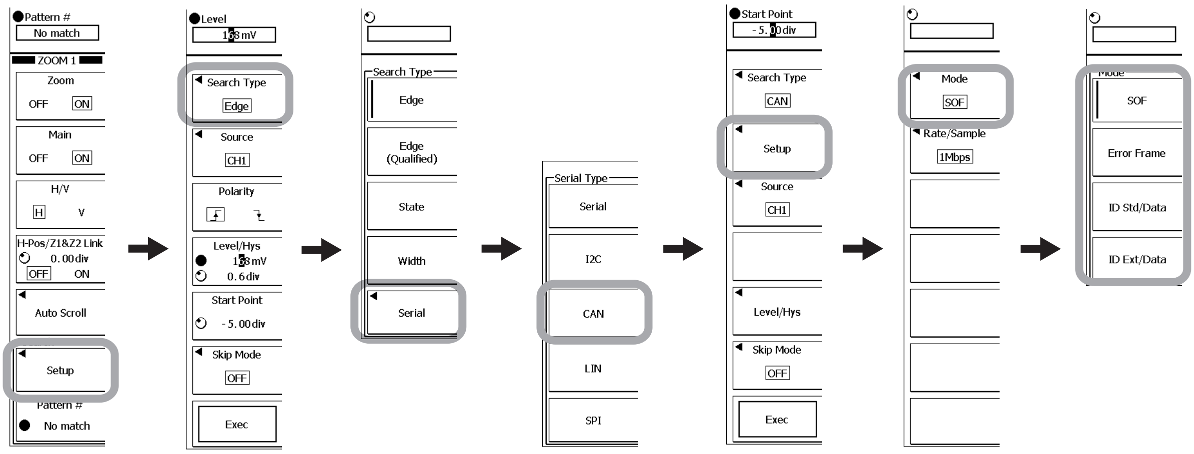
- To exit the menu during operation, press **ESC** located above the soft keys.
- For details on the operation using the rotary knob, SET, and RESET, see sections 4.1 or 4.2 in the *User's Manual (IM701310-01E).*
- For a description of the operation using a USB keyboard or a USB mouse, see section 4.3 in the *User's Manual (IM701310-01E).*

Setting the Search Conditions

1. Press **ZOOM1** or **ZOOM2**. The ZOOM menu appears.
2. Press the **Setup** soft key. The search condition setup menu appears.
3. Press the **Search Type** soft key.
4. Press the **Serial** soft key.
5. Press the **CAN** soft key.
6. Press the **Setup** soft key. The search condition setup menu appears.

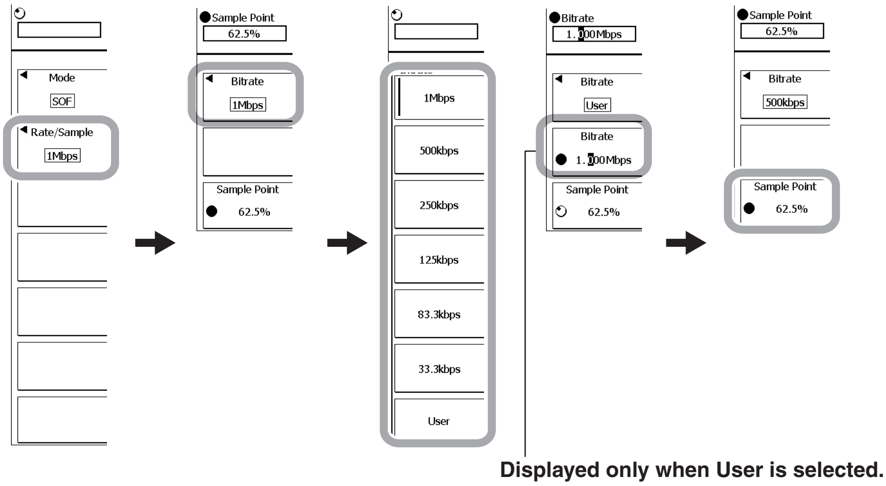
Selecting Search Mode

7. Press the **Mode** soft key. The search mode selection menu appears.
8. Press the soft key corresponding to the desired search mode.



Setting the Bit Rate and Sample Point

9. Press the **Rate/Sample** soft key.
10. Press the **Bitrate** soft key. The bit rate selection menu appears.
11. Press the **1Mbps**, **500kbps**, **250kbps**, **125kbps**, **83.8kbps**, or **User** soft key to select the bit rate. If you select User, press the **Bitrate** soft key and turn the **rotary knob** to set an arbitrary value.
12. Press the **Sample Point** soft key.
13. Turn the **rotary knob** to set the sample point.



The setup procedure of other search conditions is the same as the setup procedure of trigger conditions. See the pages listed below depending on search mode. (The soft key positions may differ.)

- SOF (search the Start of Frame): You are done with the settings.
- Error Frame (search an error frame): You are done with the settings.
- ID Std/Data (search the data/remote frame (ID: standard format)): Page 2-4
- ID Ext/Data (search the data/remote frame (ID: extended format)): Page 2-4

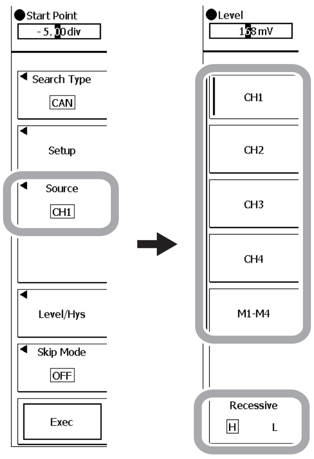
Setting the Source Channel

Selecting the Source Channel

14. Press **ESC** twice.
15. Press the **Source** soft key. The source channel setup menu appears.
16. Press any of the soft keys **CH1** to **CH4** or **M1** to **M4** to select the source channel.

Selecting the Bus Level

17. Press the **Recessive** soft key to select H (set the recessive electric potential higher than the dominant electric potential) or L (set the recessive electric potential less than the dominant electric potential).
18. Press **ESC**.



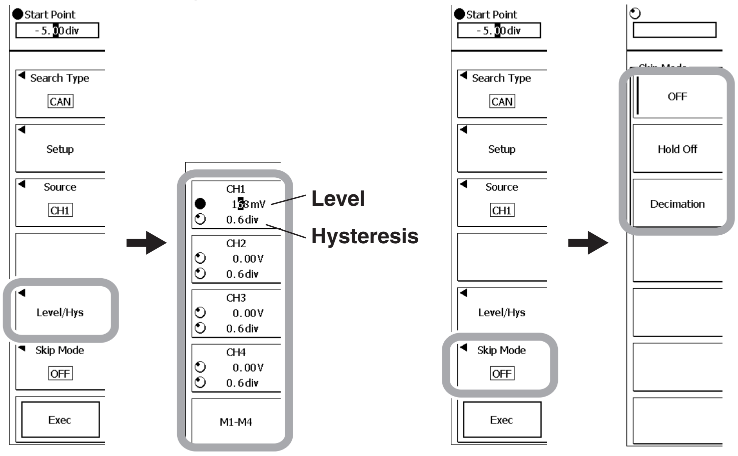
Setting the Level and Hysteresis

19. Press the **Level/Hys** soft key.
20. Press the soft key corresponding to the channel you wish to configure.
21. Turn the **rotary knob** to set the level or hysteresis.
22. Press **ESC**.

Setting the Skip Mode

23. Set the skip mode as necessary.
    Press the **Skip Mode** soft key. The skip mode setup menu appears.
24. Press the **OFF**, **Hold Off**, or **Decimation** soft key.
25. Turn the **rotary knob** to set the time to skip or the search count.
26. Press **ESC**.

Set the level and hysteresis **Set the skip mode**



Setting the Search Start Point

27. Turn the **rotary knob** to set the search start point.

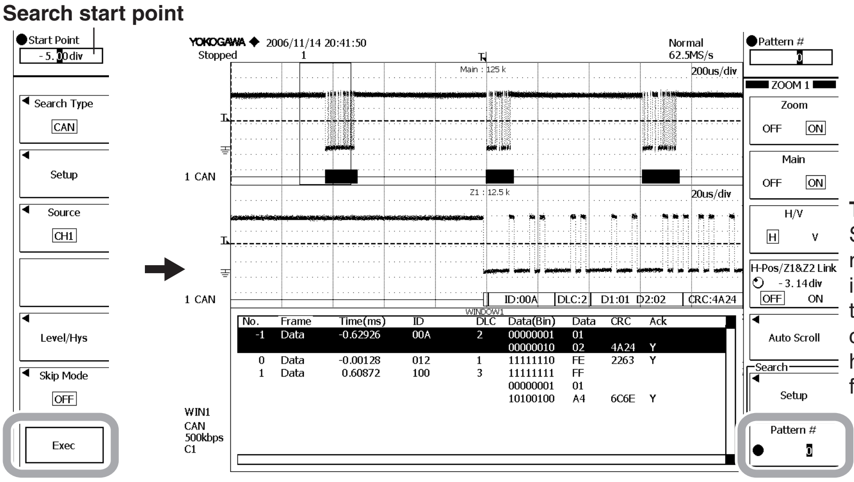
Executing the Search

28. Press the **Exec** soft key. The section that meets the search conditions is displayed expanded in the zoom window.

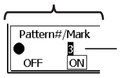
Viewing Search Points

29. Press **ESC**.
30. Press the **Pattern #** soft key.
31. Turn the **rotary knob** to select an arbitrary search point. The zoom position moves to the selected search point.

Search start point



Turning ON/OFF the Search Mark ▼
Search marks can be displayed on the main window and zoom window to indicate the locations on the waveform that have been found. The search mark corresponding to the search number is highlighted (applies to products with firmware version 3.6 or later).



Search number

Explanation

Setting Search Conditions

Search Mode

You can select the search mode from below.

SOF	Search the SOF (Start of Frame)
Error Frame	Search an error frame
ID Std/Data	Search a data frame or remote frame (ID: standard format) that matches the specified conditions.
ID Ext/Data	Search a data frame or remote frame (ID: extended format) that matches the specified conditions.

The settings of other search conditions are the same as the settings of trigger conditions. For details, see section 2.2.

Source Channel, Level, Hysteresis, Etc.

The settings for the source channel, level, hysteresis, and so on are the same as the settings of the analysis conditions. For details, see section 2.3.

Skip Mode

After searching the position that meets the search conditions, the specified amount of time or the number of positions that match the search conditions are skipped.
Select the skip mode from the following:

OFF	Searches all sections that meet the search conditions.
Hold Off	Skips the search for the specified time.
Decimation	Skips the specified number of search positions.

Search Start Point

Set the position to start the search in the range of !s!±5.00 divisions.

Search Point

When the search is executed, the zoom box moves to the data position where the conditions are met.. The search point is the center of this zoom box. The search point is set to the same position as the trigger position (see page 2-15).

Viewing Search Points

If you select an arbitrary search point using Pattern #, the zoom position moves to the selected search point. If the zoom link function is enabled (see page 2-24), the highlight in the analysis result list also moves to the frame containing the selected search point.

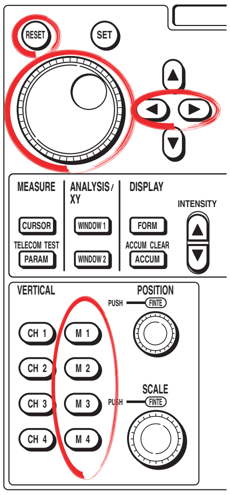
Unification of the Trigger Setting of the CAN Bus Signal and the Setting of the CAN Bus Signal Analysis, Search and Stuff Bit Computation

The trigger settings of the CAN bus signal on the menu that is entered through the ENHANCED key and the settings of the CAN bus signal analysis, search, and stuff bit computation on the menu entered through the WINDOW1, ZOOM1, or M1 key are common. For details, see page 2-15.

2.5 Performing Stuff Bit Computation

Stuff bits can be extracted from the CAN bus signal waveform and displayed as computed waveforms (MATH1 to MATH4).

Procedure



- To exit the menu during operation, press **ESC** located above the soft keys.
- For details on the operation using the rotary knob, SET, and RESET, see sections 4.1 or 4.2 in the *User's Manual (IM701310-01E).*
- For a description of the operation using a USB keyboard or a USB mouse, see section 4.3 in the *User's Manual (IM701310-01E).*

Setting the Stuff Bit Computation

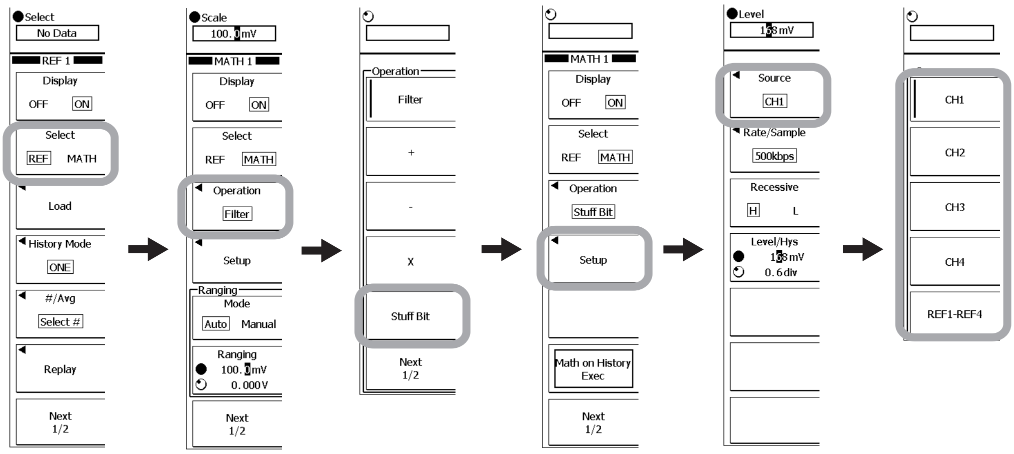
1. Press any of the keys **M1** to **M4** to select the channel on which to display the stuff bit waveform.
2. Press the **Select** soft key to select MATH.

Selecting the Operator

3. Press the **Operation** soft key. The operator selection menu appears.
4. Press the **Stuff Bit** soft key.
5. Press the **Setup** soft key. The stuff bit computation setup menu appears.

Selecting the Waveform to Be Computed

6. Press the **Source** soft key. The search condition setup menu appears.
7. Press any of the soft keys **CH1** to **CH4** and **REF1-REF4** to waveform to be computed.



Setting the Bit Rate and Sample Point

8. Press the **Rate/Sample** soft key.
9. Press the **Bitrate** soft key. The bit rate selection menu appears.
10. Press the **1Mbps**, **500kbps**, **250kbps**, **125kbps**, **83.8kbps**, or **User** soft key to select the bit rate. If you select User, press the **Bitrate** soft key and turn the **rotary knob** to set an arbitrary value.
11. Press the **Sample Point** soft key.
12. Turn the **rotary knob** to set the sample point.
13. Press **ESC**.

Selecting the Bus Level

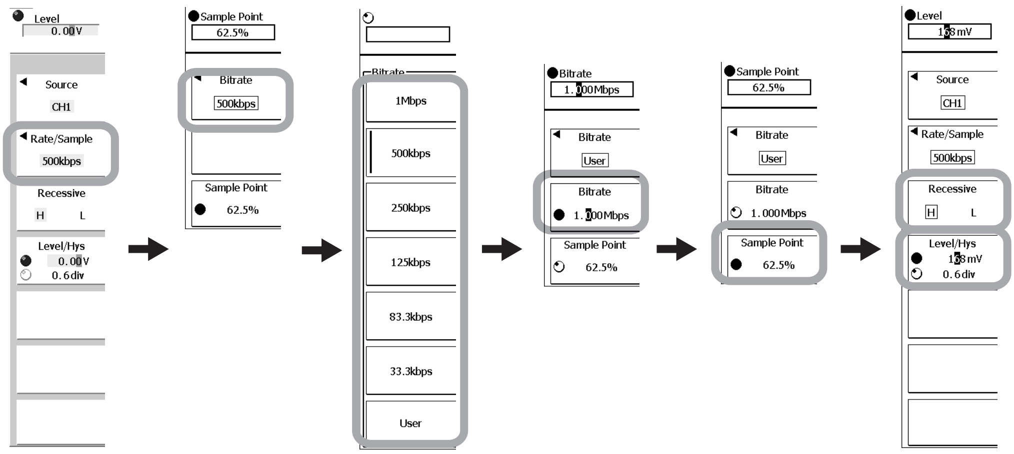
14. Press the **Recessive** soft key to select H (set the recessive electric potential higher than the dominant electric potential) or L (set the recessive electric potential less than the dominant electric potential).

Setting the Level

15. Press the **Level/Hys** soft key to activate Level.
16. Turn the **rotary knob** to set the level.

Setting the Hysteresis

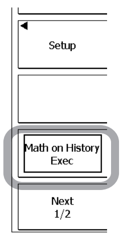
17. Press the **Level/Hys** soft key to activate Hys (hysteresis).
18. Turn the **rotary knob** to set the hysteresis.



19. Press **ESC**.

Executing the Computation on All History Waveforms

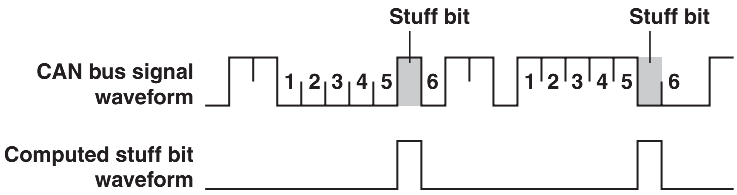
20. To perform specified computations on all history waveforms, press the **Math on History Exec** soft key. The computation is executed, and the Math on History Exec display changes to Abort.
    To cancel computation, press the Abort soft key. The computation is Aborted, and the Abort display changes to Math on History Exec.



Explanation

Stuff Bit

In CAN communication, if five or more continuous bits between the Start of Frame and CRC sequence are at the same level, CAN inserts a bit (stuff bit) having the opposite logical value of the five previous bits for the next bit (6^{th} bit).
The DL9000 lets you extract stuff bits from the CAN bus signal waveform and display them as a MATH waveform.



Setting the Stuff Bit Computation

You can set the following conditions.

Waveform to Be Computed (Source)

Select the source waveform for the stuff bit computation from CH1 to CH4 and REF1 to REF4. The default settings are MATH1: CH1, MATH2: CH2, MATH3: CH3, and MATH4: CH4.

Bitrate

Select the transfer rate of the CAN bus signal from 1Mbps, 500kbps, 250kbps, 125kbps, 83.8kbps, and User. If you select User, you can set an arbitrary value in the range of 10k to 1M [bps] (0.1 kbps resolution).

Sample Point

The point at which the input CAN bus signal waveform changes from recessive to dominant is taken as 0%, and the point one bit time (the reciprocal of the specified bit rate) thereafter is taken as 100%, thereby allowing expression of the sample point as a percentage. The selectable range is 18.8 to 90.6 [%].
The details are the same as the trigger condition settings. See page 2-10.

Bus Level (Recessive)

Select the bus level from either of the following: In either case, the logical value is: recessive = 1 and dominant = 0.

H	The recessive electric potential is set higher than the dominant electric potential.
L	The recessive electric potential is set lower than the dominant electric potential.

Level

Set the level used to determine the data channel signal level (0 or 1).
Selectable range: ±10 divisions around the vertical position
Resolution: 0.01 divisions (For example, the resolution for 2 mV/div is 0.02 mV.)

Hysteresis (Hys)

Selectable range: 0.0 division to 4.0 divisions
Resolution: 0.1 divisions

Executing the Computation on All History Waveforms

With waveform acquisition stopped, press the Math on History Exec soft key, and perform stuff bit math on all history waveforms.

Note

- Stuff bit math cannot be performed on all history waveforms while waveforms are being acquired.
- The computation-in-progress icon appears at the lower left of the screen, and a progress bar is displayed in the center of the screen while the stuff bit math on all history waveforms is in progress. All operations other than the Abort soft key are disabled.
- If you set the trigger mode to N Single and start the waveform acquisition, stuff bit math is performed only on the latest waveform after the acquisition is stopped. To perform stuff bit math on all history waveforms, carry out the procedure given in "Executing the Computation on All History Waveforms."
- If you change a setting that affects the stuff bit math result, recomputation is performed only on the selected history waveform.

Unification of the Trigger Setting of the CAN Bus Signal and the Setting of the CAN Bus Signal Analysis, Search and Stuff Bit Computation

The trigger settings of the CAN bus signal on the menu that is entered through the ENHANCED key and the settings of the CAN bus signal analysis, search, and stuff bit computation on the menu entered through the WINDOW1, ZOOM1, or M1 key are common. For details, see page 2-15.

2.6 Saving the Data of the Analysis Result List

CAUTION

Do not remove the medium (disk) or turn OFF the power when the access indicator is blinking. Doing so can damage the storage medium or destroy the data on the medium.

Procedure

The procedure to save the data of the analysis result list is the same as that of the I^2C bus signal analysis.
For the operating procedure, see section 1.5, "Saving the Data of the Detailed Analysis Result List."

Explanation

The data of the analysis result (simple display or detail display) can be saved to a specified storage medium in CSV format. The extension is .csv. Data in CSV format is a text file with data separated by commas. It is one of the common data formats used to convert data between spreadsheet and database applications.

Setting the Storage Conditions and Notes When Saving Data

The details are the same as when saving data of the analysis result list of the I^2C bus signal. For details, see section 1.5.

Saved Data

Up to 3000 analysis results can be saved.

Data Size

Data size = (Number of frames + 4) × 155 [bytes]

* The data size is a reference value. It is not strictly warranted. Use it as a guideline when saving the data.

Data Format

For the Detail display

Hexadecimal display of DLC

Hexadecimal display of Data

Binary display of Data

Hexadecimal display of the CRC sequence

```
Analysis Type SerialBus(CAN)
Model Name    DL9000
Model Version *.**
No.           Frame  Time(ms)  ID        DLC    Data(Bin)     Data(Hex)  CRC       Ack
            0 Data   -9.6091   00A(Hex)  2 Hex  00000001 Bin  01 Hex     A424 Hex  Y
                                                00000010 Bin  02 Hex
            1 Data   -8.9693   012(Hex)  1 Hex  11111110 Bin  FE Hex     2263 Hex  Y
            2 Data   -9.3291   100(Hex)  3 Hex  11111111 Bin  FF Hex     6C6E Hex  Y
                                                00000001 Bin  01 Hex
                                                10100100 Bin  A4 Hex
            3 Error  -7.6893
                                                00000001 Bin  01 Hex
```

Hexadecimal display of ID

Time from the trigger position to the head of the frame

Frame type

Analysis number

ACK slot state
Y: ACK detected
N: ACK not detected

2.7 Error Messages

A message may appear on the screen during operation. This section describes the meanings of the messages and their corrective actions. This section lists only the error messages related to the CAN bus signal analysis function. There are other error messages related to the DL9000 and communications. These messages are described in the *User's Manual (IM701310-01E)* and the *Communication Interface User's Manual (IM701310-17E)*.

You can set the messages to be displayed in English or Japanese. For the procedure of setting the message language, see section 17.1 in the *User's Manual IM701310-01E.*

If the corrective action requires servicing, contact your nearest YOKOGAWA dealer for repairs.

Code	Message	Corrective Action	Page
58	Search execution is completed, but no record was found that matched the pattern.	–	2-25
506	Save data do not exist. Check the content to be saved.	Display the analysis result, and then execute the save operation again.	2-17, 2-32
670	The corresponding field was not found.	–	2-17

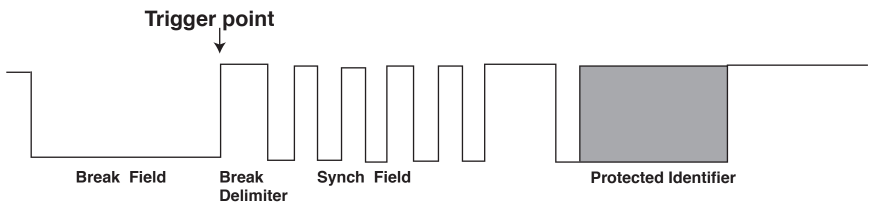
3.1 Overview of the LIN Bus Signal Analysis Function

LIN Bus Signal Analysis Function

LIN stands for Local Interconnect Network. It is a serial communication protocol used mainly for automobiles and other vehicles.
By using this function, data can be analyzed while displaying the signal waveforms on the LIN bus as analog waveforms. Synchronized monitoring of the data on the LIN bus and the analog waveform is also possible. The LIN bus signal analysis function consists of the following three main functions.

Trigger Function <See page 3-5 for the operating procedure>

The trigger activates on the rising edge of the break delimitr. One of the following can be selected for the bit rate: 19200 bps, 9600 bps, 4800 bps, 2400 bps, 1200 bps, or User.



A trigger can be activated on the combination of the trigger conditions of the LIN bus signal and CAN bus signal, or of the LIN bus signal and analog signal (Event Interval trigger). For details on the event interval trigger, see section 6.14 in the *User's Manual (IM701310-01E)*.

Analysis Function <See pages 3-17 and 3-32 for the operating procedure>

This function analyzes the LIN bus signal data and shows a list of the analysis results. There are two types of analysis result lists: simple and detail. The simple list displays the analysis number, ID, Data, and Checksum status. The detail list displays the time from the trigger position, ID field, ID parity error, and Checksum error in addition to the items displayed by the simple list. The data of the analysis result can be saved to an arbitrary storage medium in CSV format. You can select an arbitrary field in the analysis results list and automatically display the LIN bus signal for that field (zoom link).

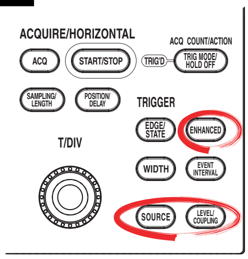
Search Function <See page 3-25 for the operating procedure>

You can search for a specific field on the LIN bus signal data. When the search is executed, the zoom box moves to the data position where the conditions are met, and the data is expanded in the zoom window (Zoom1 or Zoom2).

3.2 Setting the LIN Bus Signal Acquisition Conditions

With the LIN bus signal analysis function, you can acquire LIN bus signals using break fields of the LIN bus as trigger conditions.

Procedure



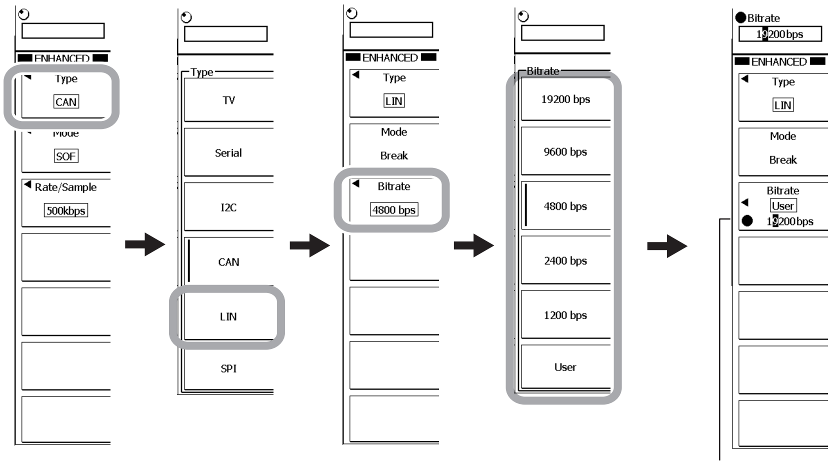
- To exit the menu during operation, press **ESC** located above the soft keys.
- For details on the operation using the rotary knob, SET, and RESET, see sections 4.1 or 4.2 in the *User's Manual (IM701310-01E).*
- For a description of the operation using a USB keyboard or a USB mouse, see section 4.3 in the *User's Manual (IM701310-01E).*

Setting the Trigger Conditions of the LIN Bus Signal

1. Press **ENHANCED**. The ENHANCED menu appears.
2. Press the **Type** soft key. The trigger type selection menu appears.
3. Press the **LIN** soft key.

Setting the Bitrate

4. Press the **Bitrate** soft key. The bit rate selection menu appears.
5. Press the **19200 bps**, **9600 bps**, **4800 bps**, **2400 bps**, **1200 bps**, or **User** soft key to select the bit rate. If you select User, turn the **rotary knob** to set an arbitrary value.



Displayed only when User is selected.

Note

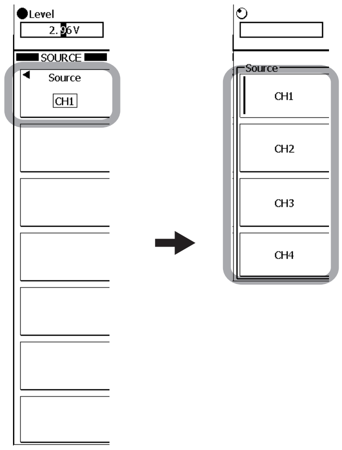
The trigger mode is fixed to Break.

Setting the Source Channel

6. Press **SOURCE**. The SOURCE menu appears.

Selecting the Source Channel

7. Press the **Source** soft key. A menu used to select the source channel of the LIN bus signal appears.
8. Press any of the soft keys from **CH1** to **CH4** to select the source channel.



Setting the Level, Coupling, HF Rejection, and Hysteresis

9. Press **LEVEL/COUPLING**. The LEVEL/COUPLING menu appears.

Selecting the Channel to Be Configured

10. Press the **CH** soft key. The menu used to select the channel appears.
11. Press any of the soft keys, **CH1** to **CH4**.

Setting the Level

12. Turn the **rotary knob** to set the level used to determine high/low.

Selecting the Coupling

13. Press the **Coupling** soft key to select DC.

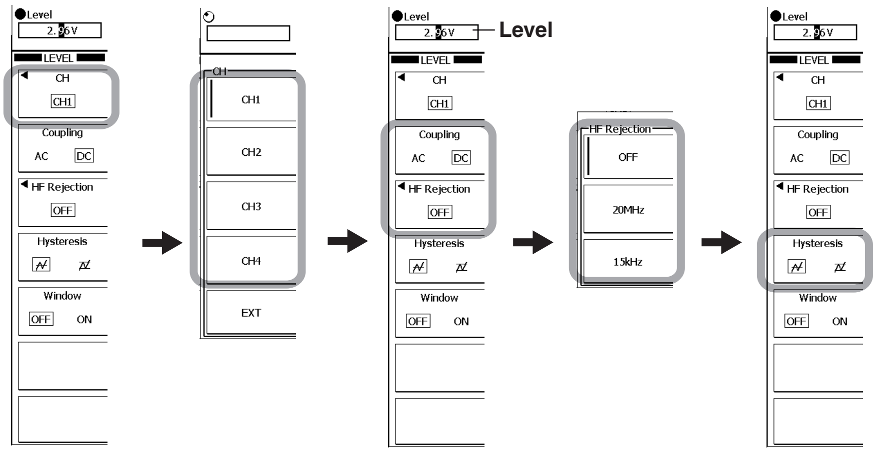
Selecting the HF Rejection

14. Press the **HF Rejection** soft key. The menu used to select the HF rejection appears.
15. Press the **OFF**, **20MHz**, or **15kHz** soft key.

Selecting the Hysteresis

16. Press the **Hysteresis** soft key to select the hysteresis.

As necessary, repeat steps 10 to 16.



Explanation

Setting the Trigger Conditions of the LIN Bus Signal

Trigger Mode

The trigger mode is fixed to Break.
The trigger activates on the rising edge of the Break delimiter of the LIN bus signal.

Bitrate

Select the transfer rate of the LIN bus signal from 19200 bps, 9600 bps, 4800 bps, 2400 bps, 1200 bps, or User. If you select User, you can set an arbitrary value in the range of 1000 bps to 20 kbps (setting resolution: 10 bps).

Source Channel

The source channel of the LIN bus signal is set using the SOURCE menu that appears when you press the SOURCE key. The menu used to set the source channel of the LIN bus signal appears only when Type is set to LIN in the ENHANCED menu.

Specifying the Source Channel

The source channel is specified using the SOURCE menu.

Trigger Level, Trigger Coupling, Etc.

Set the trigger level, trigger coupling, HF rejection, and hysteresis of each channel. For details on these items, see the User's Manual (IM701310-01E).

Unification of the Trigger Setting of the LIN Bus Signal and the Setting of the LIN Bus Signal Analysis and Search

The trigger settings of the LIN bus signal on the menu that is entered through the ENHANCED key and the settings of the LIN bus signal analysis and search on the menu entered through the WINDOW1, ZOOM1, or M1 key are common.
The CH1 to CH4 sources are the waveforms of which the settings are common.

Trigger settings of the LIN bus signal that are applied to the settings of analysis, search, and stuff bit computation

Level (Trigger Level) and Hysteresis (Trigger Hysteresis)
(The settings of the LIN bus signal analysis and search are not applied to the trigger settings.)

Items for which the trigger settings, analysis settings, and search settings of the LIN bus signal are applied mutually

Source, Bitrate

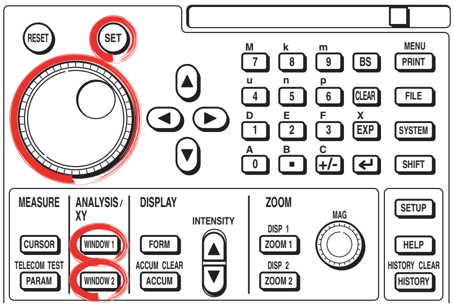
Note

- The settings of the LIN bus signal analysis and search on the menu entered through the WINDOW2 and ZOOM2 keys are not made common. They are independent settings.
- If you change common items (excluding Level and Hys) in the analysis and search menus while the waveform acquisition is in progress and the trigger type is set to LIN, the waveform acquisition is restarted.
- Even if something other than LIN bus is selected in the analysis or search menu, the level and hysteresis settings of the analysis and search menus are set to the same value if the level or hysteresis setting is changed.
- If the trigger level or hysteresis is changed by executing auto setup, the level and hysteresis settings of the analysis and search are also set to the same new value. This also applies when the setup information is initialized.
- Trigger Hysteresis and corresponds to 0.6 division and 1.0 division, respectively, of the analysis and search.

3.3 Analyzing the Data

By setting analysis conditions, the LIN signal data stored to the acquisition memory can be analyzed.

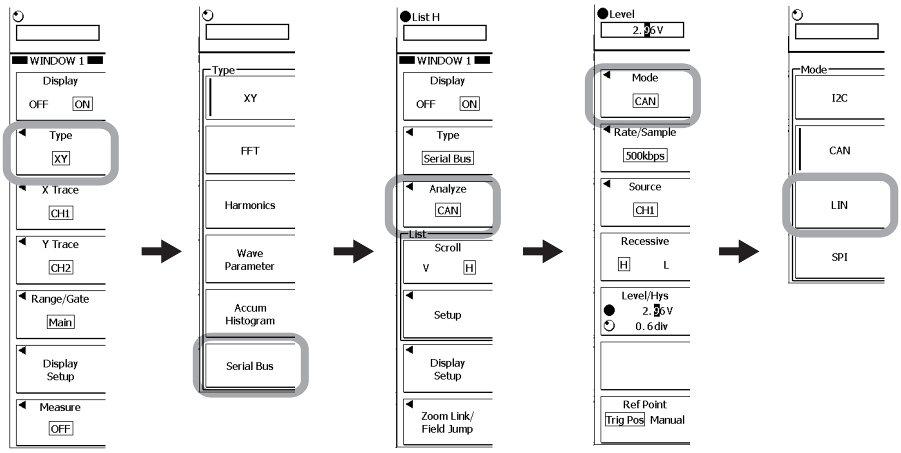
Procedure



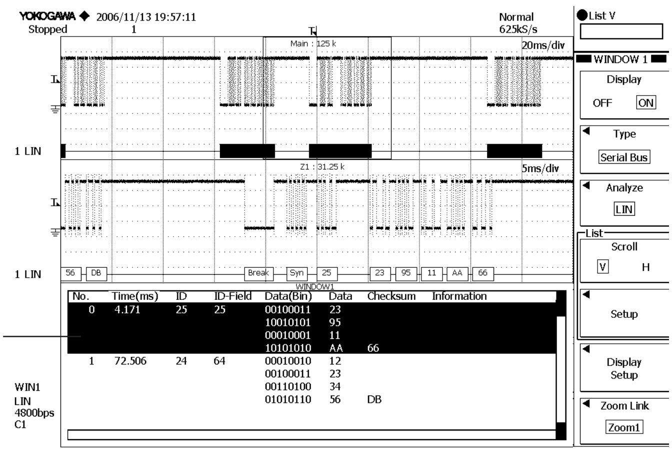
- To exit the menu during operation, press **ESC** located above the soft keys.
- For details on the operation using the rotary knob, SET, and RESET, see sections 4.1 or 4.2 in the *User's Manual (IM701310-01E).*
- For a description of the operation using a USB keyboard or a USB mouse, see section 4.3 in the *User's Manual (IM701310-01E).*

Setting the Analysis Conditions

1. Press **WINDOW1** or **WINDOW2**.
2. Press the **Type** soft key. The Type menu appears.
3. Press the **Serial** Bus soft key.
4. Press the **Analyze** soft key. The Analyze setup menu appears.
5. Press the **Mode** soft key. The Mode setup menu appears.
6. Press the **LIN** soft key.



Analysis result display



Setting the Bitrate

7. Press the **Bitrate** soft key. The bit rate selection menu appears.
8. Press the **19200 bps**, **9600 bps**, **4800 bps**, **2400 bps**, **1200 bps**, or **User** soft key to select the bit rate. If you select User, press the **Bitrate** soft key and turn the **rotary knob** to set an arbitrary value.

Setting the Source Channel

9. Press the **Source** soft key. The source channel setup menu appears.
10. Select the source channel from CH1 to CH4 and M1 to M4.

Setting the Level

11. Press the **Level/Hys** soft key to activate Level.
12. Turn the **rotary knob** to set the level.

Setting the Hysteresis

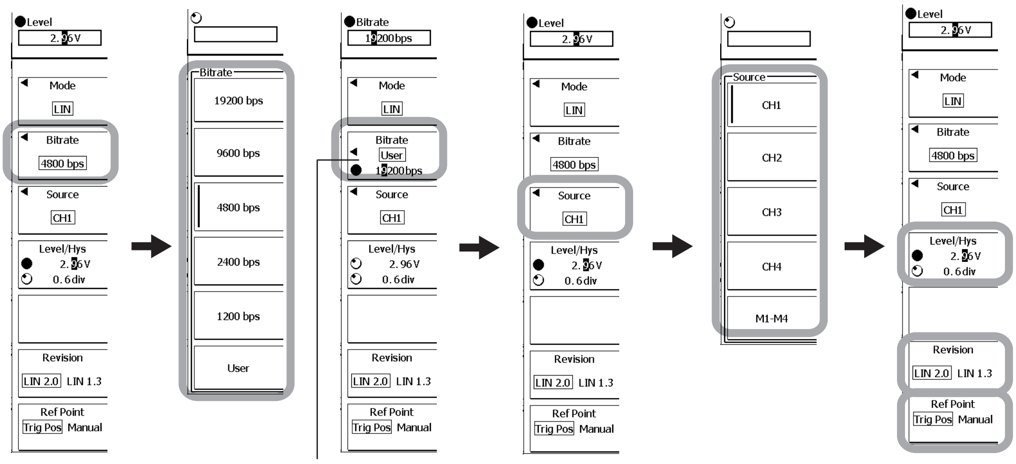
13. Press the **Level/Hys** soft key to activate Hys (hysteresis).
14. Turn the **rotary knob** to set the hysteresis.

Revision Settings

15. Press the **Revision** soft key and select LIN2.0 or LIN1.3.

Setting the Analysis Reference Point

16. Press the **Ref Point** soft key to select Trig Pos (set the analysis reference point to the trigger point) or Manual (set the analysis reference point manually).
    If you select Manual, turn the **rotary knob** to set the analysis reference point in the range of ±5.00 divisions.
17. Press **ESC**.



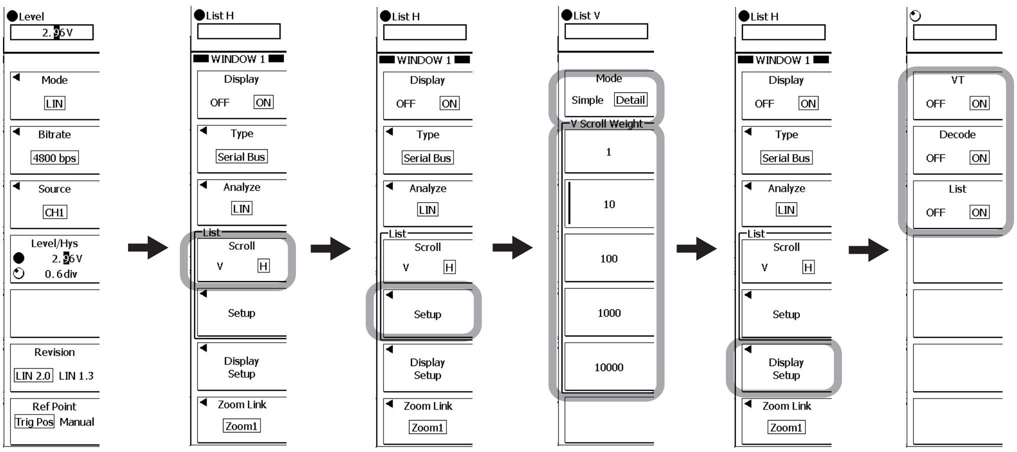
Displayed only when Bitrate is set to User.

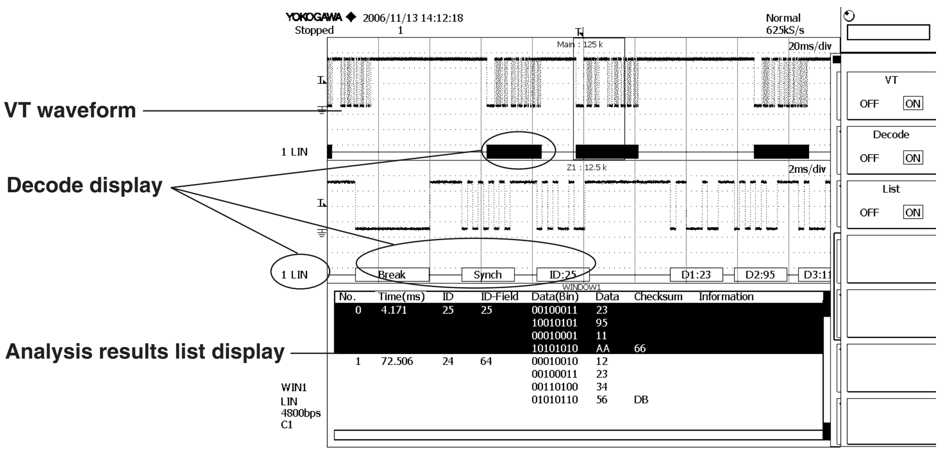
Display the Analysis Results

18. Press the **Scroll** soft key to select V (scroll vertically) or H (scroll horizontally).
19. Press the **Setup** soft key. A menu used to set the analysis screen appears.
20. Press the **Mode** soft key to select Simple (simple display) or Detail (detailed display).
21. Press any of the soft keys **1**, **10**, **100**, **1000**, or **10000** to set the step number for scrolling. The display scrolls according to the specified number.
22. Press **ESC**.
23. Press the **Display Setup** soft key. The Display Setup menu appears.
24. Press the **VT** soft key to select ON or OFF. If ON is specified, the waveform is displayed on a normal voltage-time axis.
25. In the same manner, press the **Decode** or **List** soft key to select ON or OFF. If turned ON, Decode or the analysis results list is displayed.
26. Press **ESC**.

Note

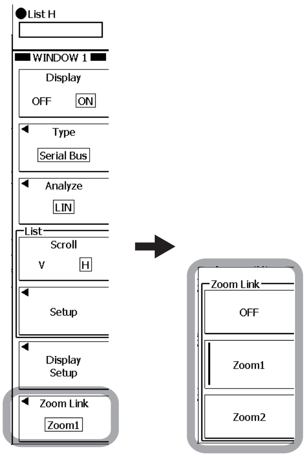
The data of the simple display and detail display of the analysis results can be saved directly to a storage medium in CSV format (.csv extension). For details, see section 3.5.





Zoom Link

27. Press the **Zoom Link** soft key. The Zoom Link setup menu appears.
28. Press the **OFF**, **Zoom1**, or **Zoom2** soft key to set the link to the zoom window. If you select Zoom1 or Zoom2, set Scroll to V and turn the rotary knob to move the zoom position (the center of the zoom box) to the start of the frame highlighted in the analysis result list.



Explanation

Setting the Analysis Conditions

You can set the following conditions.

Bitrate

Select the transfer rate of the LIN bus signal from 19200 bps, 9600 bps, 4800 bps, 2400 bps, 1200 bps, or User. If you select User, you can set an arbitrary value in the range of 1000 bps to 20 kbps (setting resolution: 10bps).

Source Channel

Select the source channel from CH1 to CH4 and M1 to M4. The source channel can be set arbitrarily.

Level

Set the level used to determine the data channel signal level (0 or 1).
Selectable range: ±10 divisions around the vertical position
Setting resolution: 0.01 div (for example, the resolution for 2 mV/div is 0.02 mV.)

Hysteresis (Hys)

Selectable range: 0.0 division to 4.0 divisions
Resolution: 0.1 divisions

Revision

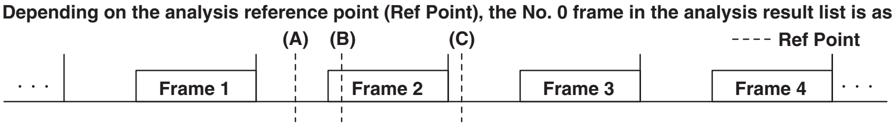
You can select a LIN revision of LIN2.0 or LIN1.3.
LIN2.0: Sets Checksum to Enhanced Checksum including a Protected ID.
LIN1.3: Performs Checksum per the Classic Checksum of the Data field only.
(Note that ID=60 (0x3c) through 63(ox3f) use Classic Checksum)

Analysis Reference Point (Ref Point)

Select the analysis reference point from the following:

Trig Pos	Sets the trigger position (on the rising edge of the Break delimiter) to the analysis reference point.
Manual	Sets the analysis reference point manually. Setting range: ±5.00 div, setting resolution: 0.01 div

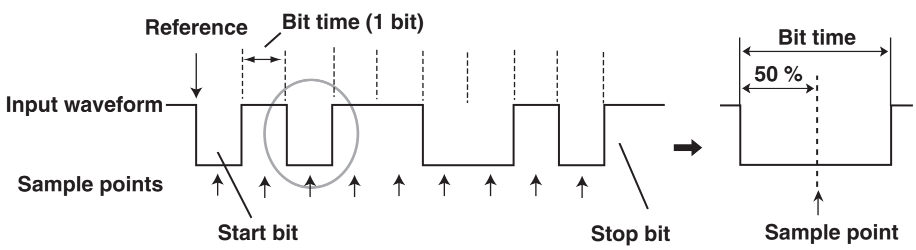
Depending on the analysis reference point (Ref Point), the No. 0 frame in the analysis result list is as follows:



(A): No. 0 frame -> Frame 2 (frame 1 is No. -1, frame 3 is No. 1, frame 4 is No. 2)
(B): No. 0 frame -> Frame 2 (frame 1 is No. -1, frame 3 is No. 1, frame 4 is No. 2)
(C): No. 0 frame -> Frame 3 (frame 1 is No. -2, frame 2 is No. -1, frame 4 is No. 1)

Sample Points

The points at which the bus level is evaluated.
The instrument detects the falling edge of the start bit of the byte field. Using the detected falling point as a reference, the input waveform is divided up into units of the bit time (the reciprocal of the set bitrate). One division is 1 bit, and the bus level is determined at the 50% position of each bit.



Unification of the Trigger Setting of the LIN Bus Signal and the Setting of the LIN Bus Signal Analysis and Search

The trigger settings of the LIN bus signal on the menu that is entered through the ENHANCED key and the settings of the LIN bus signal analysis and search on the menu entered through the WINDOW1, ZOOM1, or M1 key are common.
For details, see page 3-5.

Analyzed Data

The following data can be analyzed:

- Waveform data
  Analysis can be performed regardless of whether the waveform acquisition is in progress. If the waveform acquisition is in progress, the analysis results are updated in sync with the displayed waveform.
  The waveform data stored in the history memory (waveform data of the record number selected using HISTORY menu > Select) is also analyzed.
- Loaded acquisition data (ACQ data).

Data to be Analyzed

The following five types of data are analyzed.

Value of ID (6 bits)
Value of ID-Field (8 bits = 6 ID bits + 2 parity bits)
Value of Data (1 to 8 bytes)
Checksum byte
Additional information (ID Parity value error and Checksum error)

Analysis Range

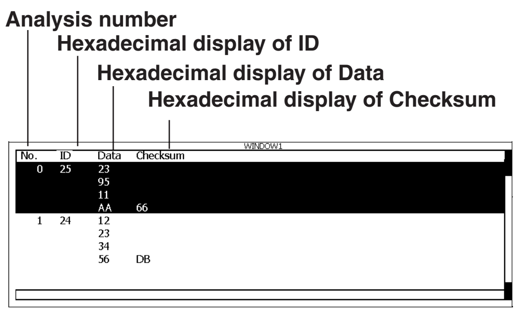
A maximum of 3000 frames on either side of the frame nearest the analysis reference point are analyzed.

Analysis Data List (Analysis Result List)

The following information is listed.

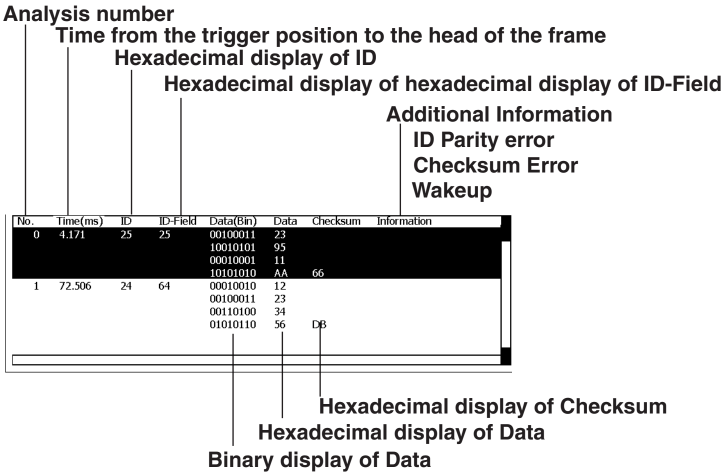
Simple Display

- Analysis number
- Hexadecimal display of ID
- Hexadecimal display of Data
- Hexadecimal display of Checksum



Detail Display

- Analysis number
- Time from the trigger position (Time (ms))
- Hexadecimal display of ID
- Hexadecimal display of ID-Field
- Binary display of Data (Bin)
- Hexadecimal display of Data (Hex)
- Hexadecimal display of Checksum
- Additional Information



- **Analysis Number**
  Points before the analysis reference point (Ref Point) are counted as No. -1, No. -2, and so on. Up to 3000 analysis results can be displayed in the range of -2999 to 2999.
  Press the RESET key to highlight the No. 0 frame.

- **Time from the Trigger Position (Time (ms))**
  Displays the time from the trigger position to the start of the frame in ms.

- **ID**
  Hexadecimal display of ID.

- **ID-Field**
  Hexadecimal display of ID value including 2 parity bits.

- **Data**
  Displays data in hexadecimal or binary notation (only for Detail). Each byte is displayed on a line.

- **Checksum**
  Displays the Checksum value in hexadecimal.

- **Information**
  If an ID parity error, Checksum error, or Wakeup signal detection occurs, “ID Parity Error,” “Checksum Error,” or “Wakeup” is displayed, respectively.

Decode Display

This function decodes each field value of the LIN bus signal.

Break:	Displays Break in orange.
Synch field:	Displays Synch in pink.
Protected identifier:	Displays the hexadecimal value in light green.
Data field:	Displays the hexadecimal value in cyan.
Checksum:	Displays the hexadecimal value in light blue.
WakeUp:	Displayed in green.

Note

If the contents extend beyond the display, the following abbreviations are made.
Break:Brk, Synch:Syn, WakeUp:WU
If even the abbreviated contents extend beyond the display, a filled background is displayed.

Zoom Link

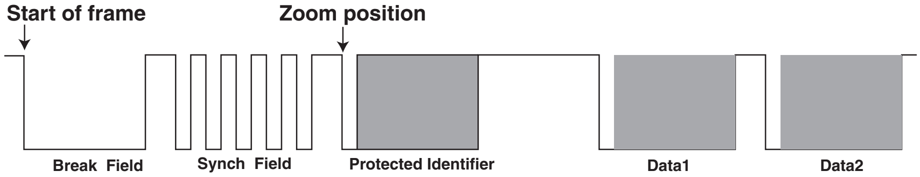
Zoom Link

Select from the following:

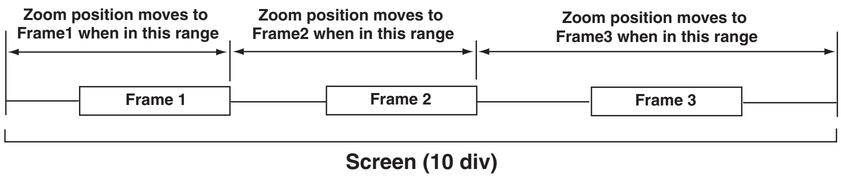
OFF	Disables the zoom link function.
Zoom1	Links to Zoom1.
Zoom2	Links to Zoom2.

By default, WINDOW1 and WINDOW2 are linked to Zoom1 and Zoom2, respectively. If Zoom1 or Zoom2 is selected and you select (highlight) an arbitrary frame in the analysis result list, the zoom position (the center of the zoom box) of Zoom1 or Zoom2 moves to the head of the ID field of that frame. Conversely, if you change the zoom position of Zoom1 or Zoom2, the highlight in the analysis result list moves to the zoom box. The frame that moves depends on the zoom position and the frame position.

Zoom Position When the List Selection (Highlighted) Position is Moved



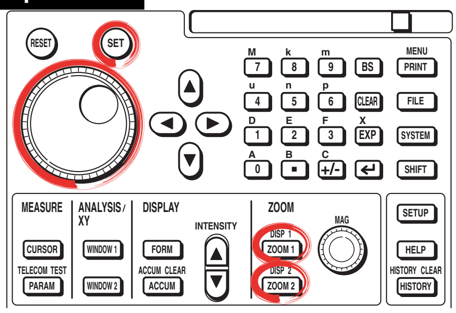
List Selection (Highlighted) Position When the Zoom Position is Moved



3.4 Searching for Waveforms

You can search the acquired waveforms for a waveform that matches the specified conditions. The setup procedure of search conditions is the same as the setup procedure of trigger conditions explained in section 3.2.

Operation



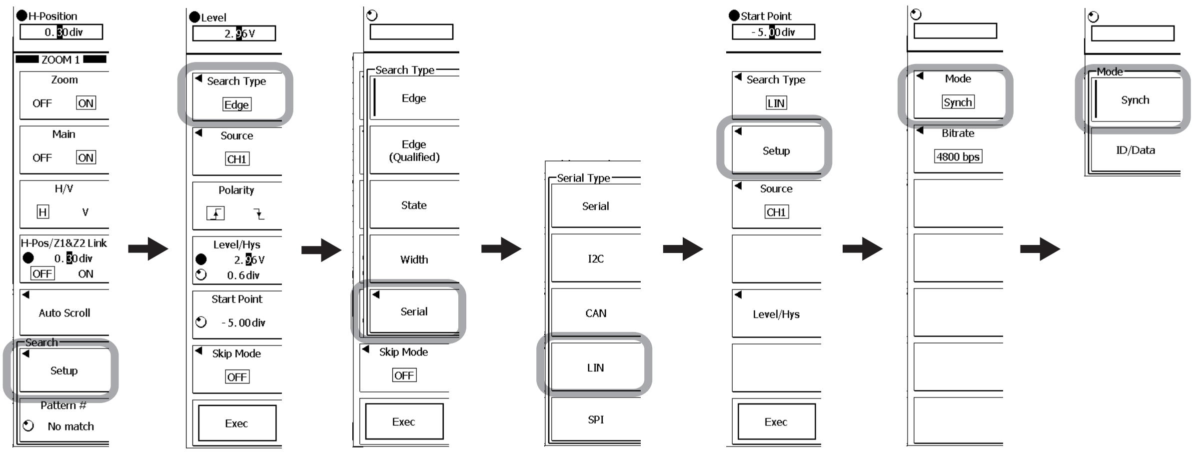
- To exit the menu during operation, press **ESC** located above the soft keys.
- For details on the operation using the rotary knob, SET, and RESET, see sections 4.1 or 4.2 in the *User's Manual (IM701310-01E).*
- For a description of the operation using a USB keyboard or a USB mouse, see section 4.3 in the *User's Manual (IM701310-01E).*

Setting Search Conditions

1. Press **ZOOM1** or **ZOOM2**. The ZOOM menu appears.
2. Press the **Setup** soft key. The search condition setup menu appears.
3. Press the **Search Type** soft key.
4. Press the **Serial** soft key.
5. Press the **LIN** soft key.
6. Press the **Setup** soft key. The search condition setup menu appears.

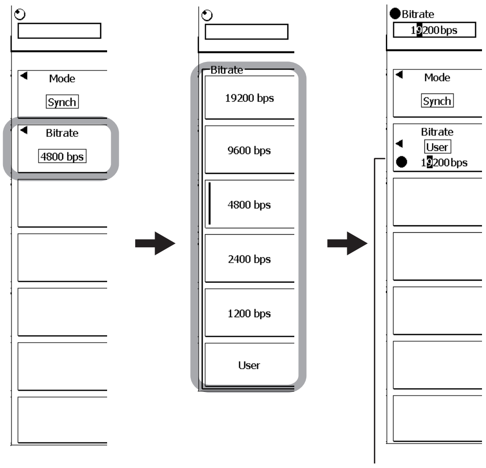
Selecting Search Mode

7. Press the **Mode** soft key. The search mode selection menu appears.
8. Press the soft key corresponding to the desired search mode.



Setting the Bitrate

9. Press the **Bitrate** soft key. The bit rate selection menu appears.
10. Press the **19200 bps**, **9600 bps**, **4800 bps**, **2400 bps**, **1200 bps**, or **User** soft key to select the bit rate. If you select User, turn the **rotary knob** to set an arbitrary value.

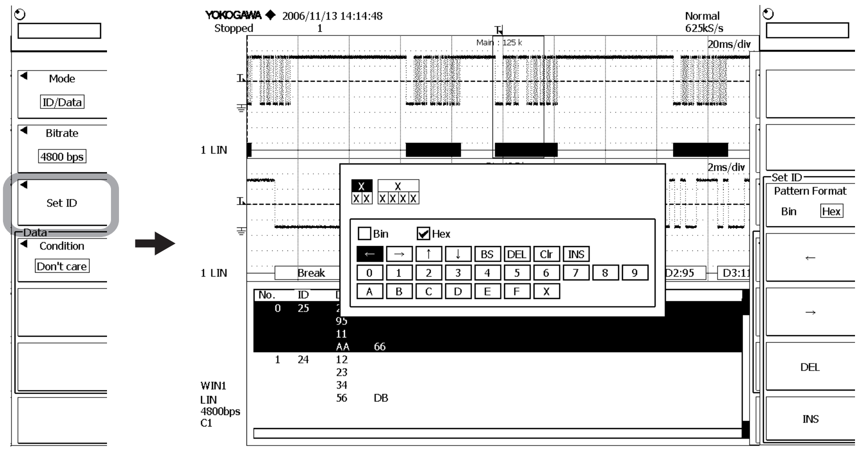


Displayed only when User is selected.

If you set the mode to Sync in step 8, skip to step 20.
If you set the mode to ID/Data, perform the settings below.

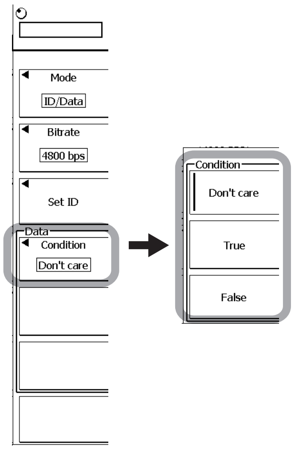
Setting the ID Bit Pattern

11. Press the **Set ID** soft key. The ID setup screen appears.
12. Use the **rotary knob**, **arrow keys**, and **SET** key to set the ID pattern.
    You can use the soft keys to change the format to binary or hexadecimal, move between bits, or clear all the bits (X).
13. Press **ESC**.



Setting the Data Field Condition

14. Press the **Condition** soft key.
15. Press the **Don't care**, **True**, or **False** soft key.
    - If you select Don't care, a trigger is activated when the ID matches. Proceed to step 20.
    - If you select True, a trigger is activated when data in the Data field matches the specified bit pattern. Proceed to step 16.
    - If you select False, a trigger is activated when data in the Data field does not match the specified bit pattern. Proceed to step 16.



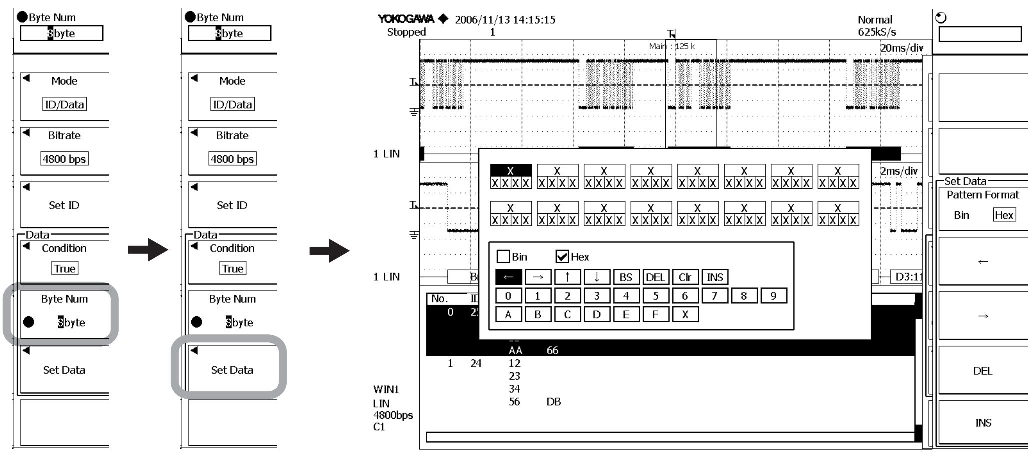
Note

For details on the search point for each condition, see page 3-19.

Setting the Data Bit Pattern

If you selected True or False in step 15, set the data bit pattern.

16. Turn the **rotary knob** to set the number of data bytes. The number of bytes in the bit pattern setting screen in the next step varies according to the number of bytes set here.
17. Press the **Set Data** soft key. A screen appears for setting the Data bit pattern.
18. Use the **rotary knob**, **arrow keys**, and **SET** key to set the Data bit pattern. You can use the soft keys to change the format to binary or hexadecimal, move between bits, or clear all the bits (X).
19. Press **ESC**. Proceed to step 20.



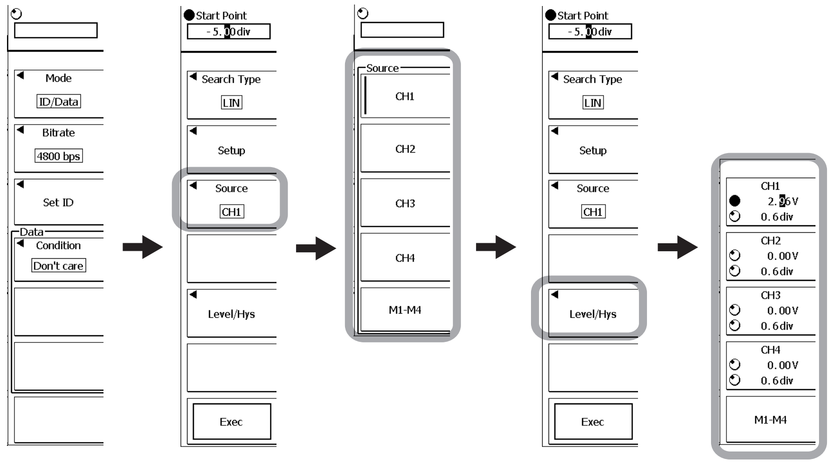
Setting the Source Channel

Selecting the Source Channel

20. Continuing on from step 19, press **ESC**.
21. Press the **Source** soft key. The source channel setup menu appears.
22. Press any of the soft keys from **CH1** to **CH4** or **M1** to **M4** to select the source channel.

Set the Level and Hysteresis

23. Press the **Level/Hys** soft key.
24. Press the soft key corresponding to the channel you wish to configure.
25. Turn the **rotary knob** to set the level or hysteresis.
26. Press **ESC**.



Setting the Search Start Point

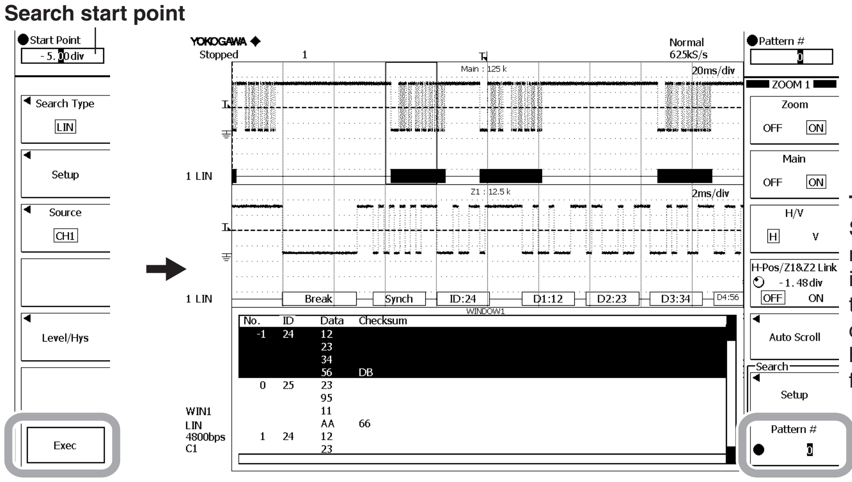
27. Turn the **rotary knob** to set the search start point.

Executing the Search

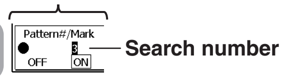
28. Press the **Exec** soft key. The section that meets the search conditions is expanded in the zoom window.

Viewing Search Points

39. Press **ESC**.
30. Press the **Pattern #** soft key.
31. Turn the **rotary knob** to select an arbitrary search point. The zoom position moves to the selected search point.



Turning ON/OFF the Search Mark ▼
Search marks can be displayed on the main window and zoom window to indicate the locations on the waveform that have been found. The search mark corresponding to the search number is highlighted (applies to products with firmware version 3.6 or later).



Explanation

Setting Search Conditions

Search Mode

You can select the search mode from below.

Synch	Search by Synch field
ID/Data	Search by data frames (ID: standard format) matching the specified conditions

Source Channel, Level, Hysteresis, Etc.

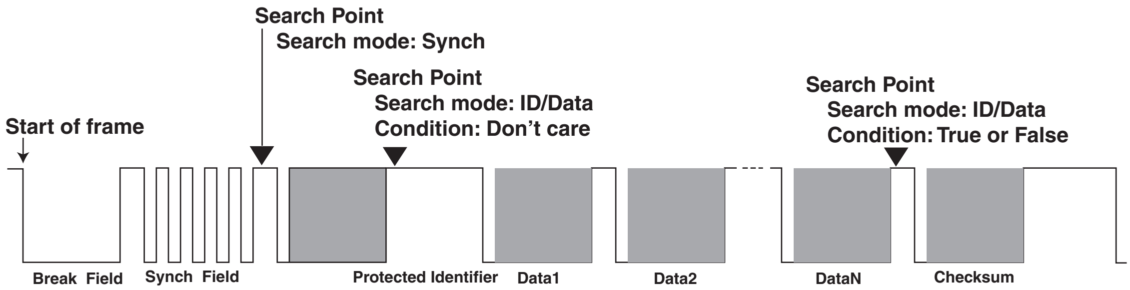
The settings for the source channel, level, hysteresis, and so on are the same as the settings of the analysis conditions. For details, see section 3.3.

Search Start Point

Set the position to start the search in the range of Å}5.00 divisions.

Search Point

When the search is executed, the zoom box moves to the data position where the conditions are met. The search point is the center of this zoom box.
The search point is as follows according to the search mode and condition settings.



Viewing Search Points

If you select an arbitrary search point using Pattern #, the zoom position moves to the selected search point. If the zoom link function is enabled (see page 3-14), the highlight in the analysis results list also moves to the frame containing the selected search point.

Unification of the Trigger Setting of the LIN Bus Signal and the Setting of the LIN Bus Signal Analysis and Search

The trigger settings of the LIN bus signal on the menu that is entered through the ENHANCED key and the settings of the LIN bus signal analysis and search on the menu entered through the WINDOW1, ZOOM1, or M1 key are common.
For details, see page 3-5.

3.5 Saving the Data of the Analysis Result List

CAUTION

Do not remove the medium (disk) or turn OFF the power when the access indicator is blinking. Doing so can damage the storage medium or destroy the data on the medium.

Procedure

The procedure to save the data of the analysis result list is the same as that of the I2C bus signal analysis.
For the operating procedure, see section 1.5, ÅgSaving the Data of the Detailed Analysis Result List.Åh

Explanation

The data of the analysis result (simple display or detail display) can be saved to a specified storage medium in CSV format. The extension is .csv. Data in CSV format is a text file with data separated by commas. It is one of the common data formats used to convert data between spreadsheet and database applications.

Setting the Storage Conditions and Notes When Saving Data

The details are the same as when saving data of the analysis result list of the I^2C bus signal. For details, see section 1.5.

Saved Data

Up to 3000 analysis results can be saved.

Data Size

Data size = (Number of frames + 4) × 170 [bytes]

* The data size is a reference value. It is not strictly warranted. Use it as a guideline when saving the data.

Data Format

For the Detail display

Binary display of Data
Hexadecimal display of Data
Hexadecimal display of Checksum

```
Analysis Type SerialBus(LIN)
Model Name DL9000
Model Version 2.3
No.   Time(ms)    ID       ID-Field    Data(Bin)     Data     Checksum    Information
-2    -99.995     Wake up
-1    -31.661     24 Hex   64 Hex      00010010 Bin  12 Hex
                                       00100011 Bin  23 Hex
                                       00110100 Bin  34 Hex
                                       01010110 Bin  56 Hex   DB Hex
0     4.171       25 Hex   25 Hex      00100011 Bin  23 Hex
                                       10010101 Bin  95 Hex
                                       00010001 Bin  11 Hex
                                       10101010 Bin  AA Hex   66 Hex
1     72.506      24Hex    64 Hex      00010010 Bin  12 Hex
                                       00100011 Bi   23 Hex
                                       00110100 Bin  34 Hex
                                       01010110 Bin  56 Hex   DB Hex
```

Hexadecimal display of ID-Field
Hexadecimal display of ID
Time from the trigger position to the start of the frame
Analysis number

Errors
ID parity error
Checksum Error

3.6 Error Messages

A message may appear on the screen during operation. This section describes the meanings of the messages and their corrective actions. This section lists only the error messages related to the LIN bus signal analysis function. There are other error messages related to the DL9000 and communications. These messages are described in the *User's Manual (IM701310-01E)* and the *Communication Interface User's Manual (IM701310-17E)*.
You can set the messages to be displayed in English or Japanese. For the procedure of setting the message language, see section 17.1 in the *User's Manual IM701310-01E.*
If the corrective action requires servicing, contact your nearest YOKOGAWA dealer for repairs.

Code	Message	Corrective Action	Page
58	Search execution is completed, but no record was found that matched the pattern.	–	3-15
506	Save data do not exist. Check the content to be saved.	Resave the analysis results after displaying them.	3-8, 3-11
670	Display the analysis result, and then execute the save operation again.	–	

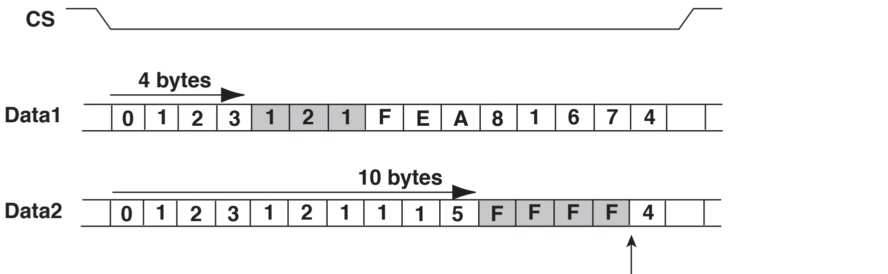
4.1 Overview of the SPI Bus Signal Analysis Function

SPI Bus Signal Analysis Function

The SPI (Serial Peripheral Interface) Bus is a synchronized serial bus that is widely used for inter-IC communications and data communications.
By using this function, you will be able to analyze data while displaying the SPI Bus signal waveform.
The SPI Bus signal analysis function consists of the following three main functions.

Trigger Function <See page 4-5 for the operating procedure>

Acquires SPI Bus signals by comparing the specified conditions with the CH1 to CH4 input signals at the byte level (8 bits).
The data position to be compared can be specified in terms of the number of bytes from the assertion of the chip select signal (CS). You can set two data patterns (Data 1 and Data 2) for the four-wire SPI and one data pattern for the three-wire SPI. For Data 1 and Data 2, a trigger is activated at the position where the latter data pattern matches.
An example is given below for the case when comparing Data 1 (3 bytes) from the 4th byte after the assertion of the CS and comparing Data 2 (4 bytes) from the 10th byte after the assertion of the CS and activating the trigger when both patterns match



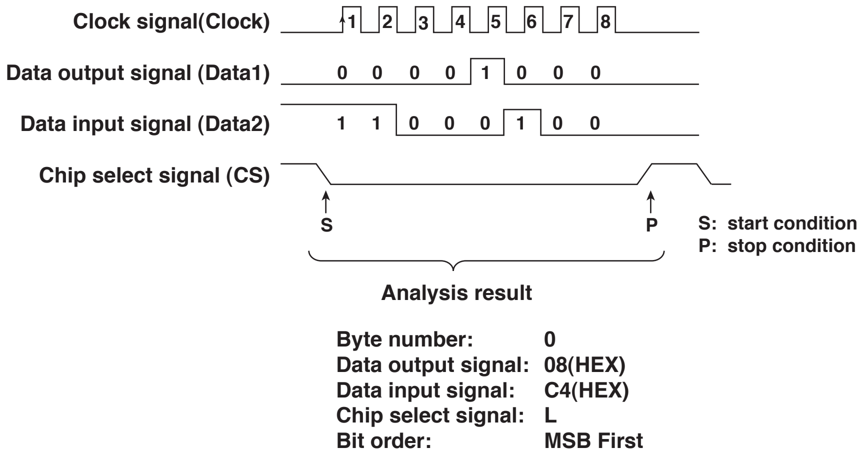
CS: Active low

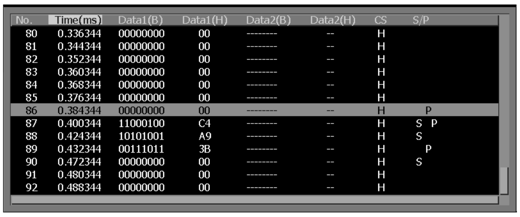
Data1	**Condition:**	**True**
	Data Size:	**3**
	Pattern:	**121**
	Data Psosition:	**4**
Data2	**Condition:**	**True**
	Data Size:	**4**
	Pattern:	**FFFF**
	Data Psosition:	**10**

A trigger can be activated on the combination of the trigger conditions of the SPI bus signal and analog signal (event interval trigger). For details on the event interval trigger, see section 6.14 in the *User's Manual (IM701310-01E)*.

Analysis Function <See pages 4-12 and 4-22 for the operating procedure>

This function analyzes the SPI bus signal data and shows a list of the analysis results. Analysis is performed at the byte level (8 bits) by synchronizing to the clock signal. There are two types of analysis result lists: simple and detail. The simple list displays the analysis number, Data 1/Data 2, and CS status for each byte. The detail list displays the time from the trigger position and the start and end positions of the active period in addition to the items displayed by the simple list. The data of the detail list can be saved to an arbitrary storage medium in CSV format.
In addition, you can select an arbitrary byte in the analysis result list and move the zoom position (the center of the zoom box) to the head of that byte.





No.	Time(ms)	Data1(B)	Data1(H)	Data2(B)	Data2(H)	CS	S/P
80	0.336344	00000000	00	--------	--	H	
81	0.344344	00000000	00	--------	--	H	
82	0.352344	00000000	00	--------	--	H	
83	0.360344	00000000	00	--------	--	H	
84	0.368344	00000000	00	--------	--	H	
85	0.376344	00000000	00	--------	--	H	
86	0.384344	00000000	00	--------	--	H	P
87	0.400344	11000100	C4	--------	--	H	S P
88	0.424344	10101001	A9	--------	--	H	S
89	0.432344	00111011	3B	--------	--	H	P
90	0.472344	00000000	00	--------	--	H	S
91	0.480344	00000000	00	--------	--	H	
92	0.488344	00000000	00	--------	--	H	

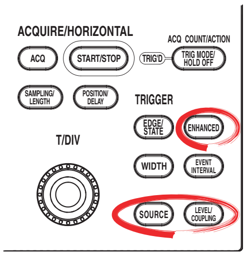
Search Function <See page 4-21 for the operating procedure>

This function searches for data that matches or does not match a specific data pattern in the SPI bus signal data. When the search is executed, the zoom box moves to the data position where the conditions are met, and the data is displayed expanded in the zoom window (Zoom1 or Zoom2).

4.2 Setting the SPI Bus Signal Acquisition Conditions

The SPI Bus signal is acquired using the conditions of the SPI bus signal as trigger conditions.

Procedure



- To exit the menu during operation, press **ESC** located above the soft keys.
- For details on the operation using the rotary knob, SET, and RESET, see sections 4.1 or 4.2 in the *User's Manual (IM701310-01E).*
- For a description of the operation using a USB keyboard or a USB mouse, see section 4.3 in the *User's Manual (IM701310-01E).*

Setting the Trigger Conditions of the SPI Bus Signal

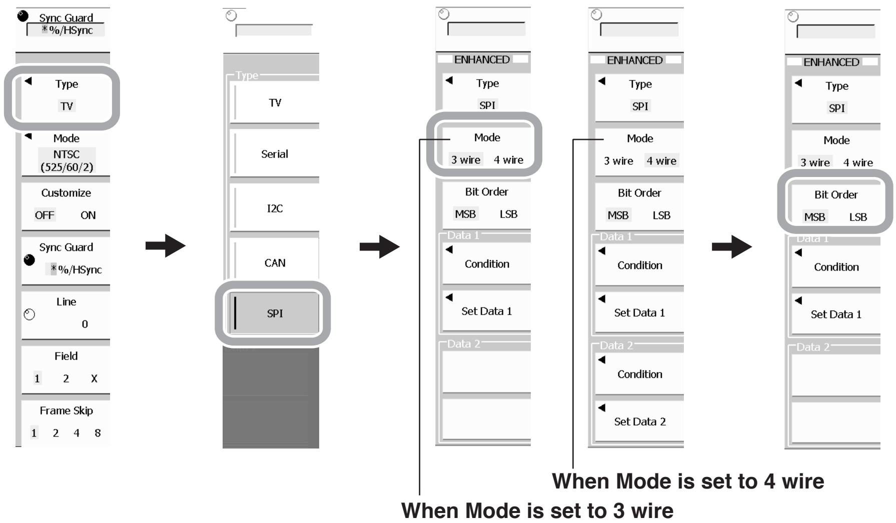
1. Press **ENHANCED** to display the ENHANCED menu.
2. Press the **Type** soft key to display the Type menu.
3. Press the **SPI** soft key.

Selecting Three-Wire or Four-Wire

4. Press the **Mode** soft key to select 3 wire or 4 wire.

Selecting the Bit Order

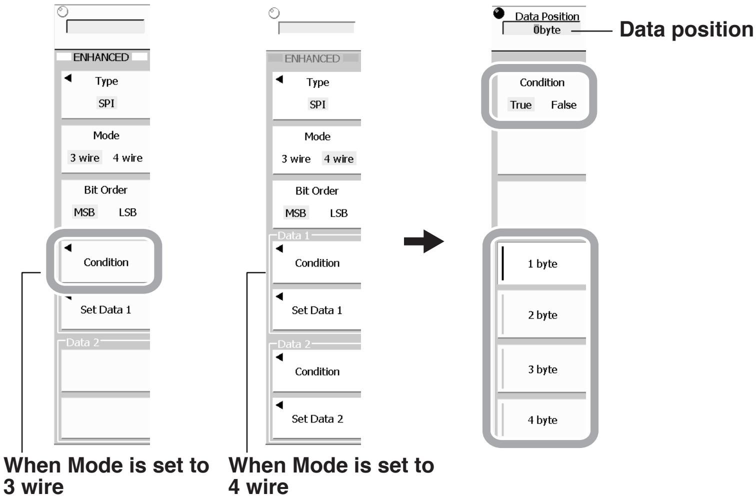
5. Press the **Bit Order** soft key to set the read direction of the I/O data bits to MSB (MSB first) or LSB (LSB first).



Setting the Data Conditions

- **Setting the Conditions and Settinge the Data Size**

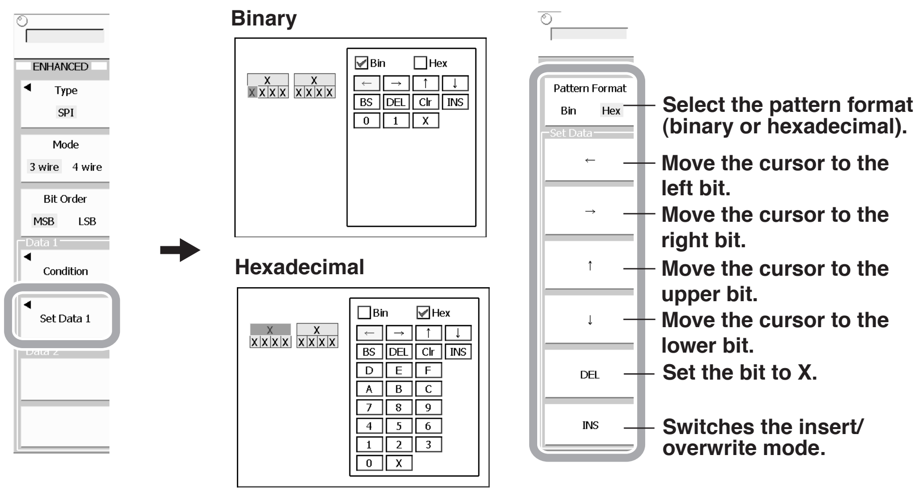
6. Press the **Condition** soft key to display a menu used to set the condition and data size.
7. Press the **Condition** soft key to select True (match the specified bit pattern) or False (not match the specified bit pattern).
8. Press a soft key corresponding to the data size of the bit pattern (a trigger condition).
9. To activate a trigger the specified number of bytes after the assertion of the CS, use the **rotary knob** to set the trigger position (data position).



10. For four-wire, set the condition, data size, and data position for Data 2 in the same manner.
11. Press **ESC**.

- **Setting the Bit Pattern**

12. Press the **Set Data 1** soft key to display the bit pattern setting screen.
13. Turn the **rotary knob**, **arrow** keys, and **SET** key to set the pattern.
    You can use the soft keys to change the format to binary or hexadecimal or clear all the bits (X).



14. Press **ESC**.
15. For four-wire, set the bit pattern for Data 2 in the same manner.

Setting the Source Channel

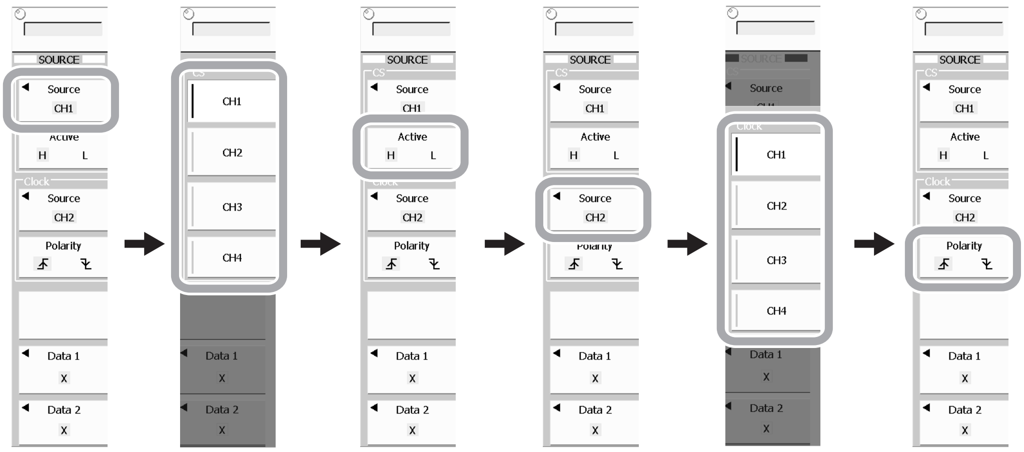
16. Press **SOURCE** to display the SOURCE menu.

Setting the CS

17. Press the **Source** soft key under CS to display the CS source channel selection menu.
18. Press any of the soft keys **CH1** to **CH4** to assign the channel to the CS.
19. Press the **Active** soft key to select H (high) or L (low).

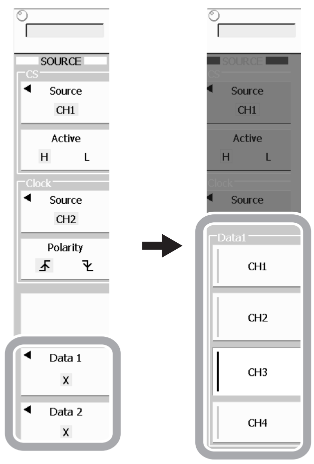
Setting the Clock

20. Press the **Source** soft key under Clock to display the clock source channel selection menu.
21. Press any of the soft keys **CH1** to **CH4** to assign the channel to the clock channel.
22. Press the **Polarity** soft key to select rising or falling.



Select the Channel for Comparing the Bit Pattern

23. Press the **Data 1** soft key to display a menu used to select the channel to compare to the bit pattern of Data 1.
24. Press any of the soft keys, **CH1** to **CH4**.
25. For four-wire, select the channel for comparing the bit pattern for Data 2 in the same manner.



Setting the Level, Coupling, HF Rejection, and Hysteresis

26. Press **LEVEL/COUPLING**. The LEVEL/COUPLING menu appears.

Selecting the Channel to Be Configured

27. Press the **CH** soft key. The menu used to select the channel appears.
28. Press any of the soft keys, **CH1** to **CH4**.

Setting the Level

29. Turn the **rotary knob** to set the level used to determine high/low.

Selecting the Coupling

30. Press the **Coupling** soft key to select DC.

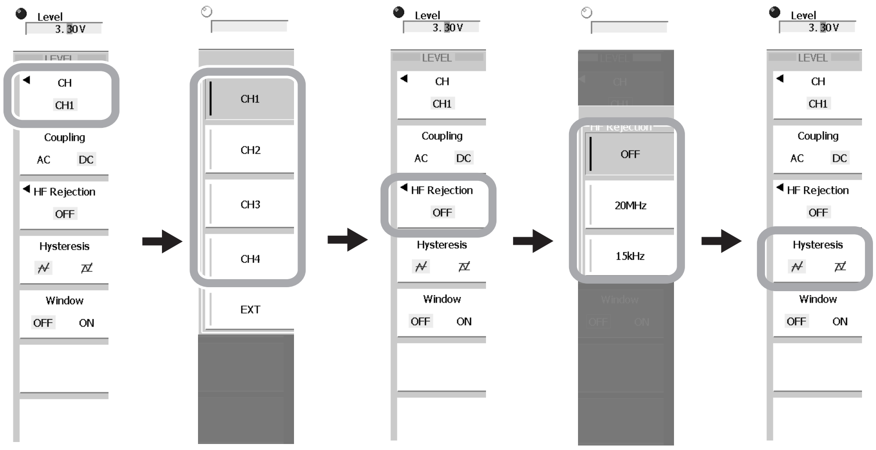
Selecting the HF Rejection

31. Press the **HF Rejection** soft key. The menu used to select the HF rejection appears.
32. Press the **OFF**, **20MHz**, or **15kHz** soft key.

Selecting the Hysteresis

33. Press the **Hysteresis** soft key to select the hysteresis.

As necessary, repeat steps 27 to 33.



Explanation

Setting the Trigger Conditions of the SPI Bus Signal

Three-Wire or Four-Wire (Mode)

For three-wire, triggers are activated on the Data 1 bit pattern.
For four-wire, triggers are activated on the Data 1 and Data 2 bit patterns. The Data 1 and Data 2 data patterns can also be used as independent trigger conditions.

Bit Order

You can select the bit order of Pattern A and Pattern B according to the signal flow of the input/output data.

MSB	Select this when the I/O data signal is flowing MSB first.
LSB	Select this when the I/O data signal is flowing LSB first.

Data Conditions

- **Condition**

  You can select either of the following:

True	A trigger is activated when the data matches the specified bit pattern.
False	A trigger is activated when the data does not match the specified bit pattern.

- **Data Size**

  Select the number of bytes (data length) of bit pattern to be specified in the range of 1 to 4 bytes.

- **Data Position**

  Compares the pattern the specified number of bytes after the assertion of the chip select (CS). The data position can be set separately for Data 1 and Data 2. The selectable range is 0 to 9999.

- **Pattern Format**

  Set the bit pattern format to either of the following:

Hex	Hexadecimal
Bin	Binary

- **Bit Pattern**

  Set the bit pattern. When Pattern Format is set to Hex (hexadecimal), you can enter X, 0 to 9, or A to F in units of 4 bits. When Pattern Format is set to Bin (binary), you can enter X, 0, or 1 for each bit.

Source Channel

The source channel of the SPI bus signal is set using the SOURCE menu that appears when you press the SOURCE key. The menu used to set the source channel of the SPI bus signal appears only when Type is set to SPI in the ENHANCED menu.

Chip Select (CS)

Select the CS from CH1 to CH4. Select whether to make the signal active when it is high or low with the Active item.

Clock Signal

Select the input signal to be the clock signal from CH1 to CH4.
The bit pattern is compared on the rising or falling edge of the signal.

Channel for Comparing the Bit Pattern (Data1/Data2)

Select the signal for comparing the bit pattern from CH1 to CH4.

Trigger Level, Trigger Coupling, Etc.

Set the trigger level, hysteresis, trigger coupling, and HF rejection of each channel.
For details on these items, see the *User's Manual (IM701310-01E)*.

Unification of the Trigger Setting of the SPI Bus Signal and the Setting of the SPI Bus Signal Analysis and Search

On products with software version 1.80 or higher, the trigger settings of the SPI bus signal on the menu that is entered through the ENHANCED key and the settings of the SPI bus signal analysis and search on the menu entered through the WINDOW1 and ZOOM1 key are common. The CH1 to CH4 sources are the waveforms of which the settings are common.

Trigger settings of the SPI bus signal that are applied to the settings of both the SPI bus signal analysis and search
Level (trigger level), Hysteresis (trigger hysteresis) (The settings of the SPI bus signal analysis and search are not applied to the trigger settings of the SPI bus signal.)

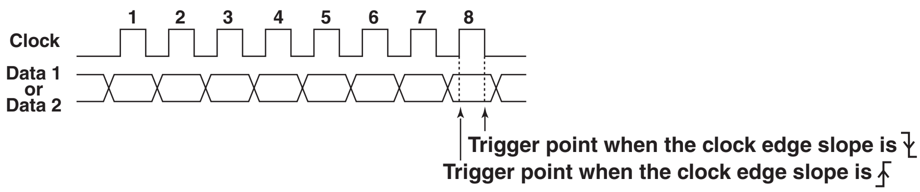
Item of which the trigger settings of the SPI bus signal, the settings of the SPI bus signal analysis, and the settings of the SPI bus signal search that are mutually applied
Mode, Bit Order, CS Source, CS Active, Clock Source, Clock Polarity, Data1, Data2

Note

- M1 to M4 are does not apply to the common settings.
- The settings of the SPI bus signal analysis and search on the menu entered through the WINDOW2 and ZOOM2 keys are not made common. They are independent settings.
- If you change common items (excluding Level and Hys) in the analysis and search menus while the waveform acquisition is in progress and the trigger type is set to SPI, the waveform acquisition is restarted.
- Even if something other than SPI bus is selected in the analysis or search menu, the level and hysteresis settings of the analysis and search are set to the same value if the trigger level or hysteresis setting is changed.
- When the trigger level or hysteresis is changed by executing auto setup, the level and hysteresis settings of the analysis and hysteresis are also set to the same new value. This also applies when the setup information is initialized.
- Trigger hysteresis and correspond to 0.6 division and 1.0 division, respectively, of the analysis or search.

Examples of SPI Bus Signal Trigger Conditions

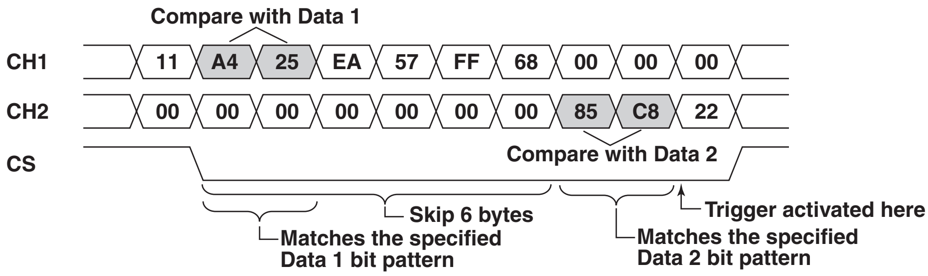
The trigger point is set to the position indicated below depending on the clock edge slope setting (see page 4-5).



This section will indicate the data sequence at the byte level in hexadecimal and indicate the position where the trigger will occur. The shaded section is the figure indicates the byte pattern (sequence) that is compared. We assume that CH1 is selected for comparing the bit pattern of Data 1 and CH2 is selected for comparing that of Data 2.

Trigger Conditions

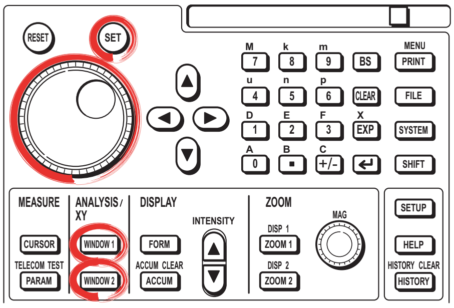
CS		L
Data 1		A4 25
	Data Byte	2
	Condition	True
	Byte Position	0
Data 2		85 C8
	Data Byte	2
	Condition	True
	Byte Position	6



4.3 Analyzing the Data

By setting analysis conditions, the SPI signal data stored to the acquisition memory can be analyzed.

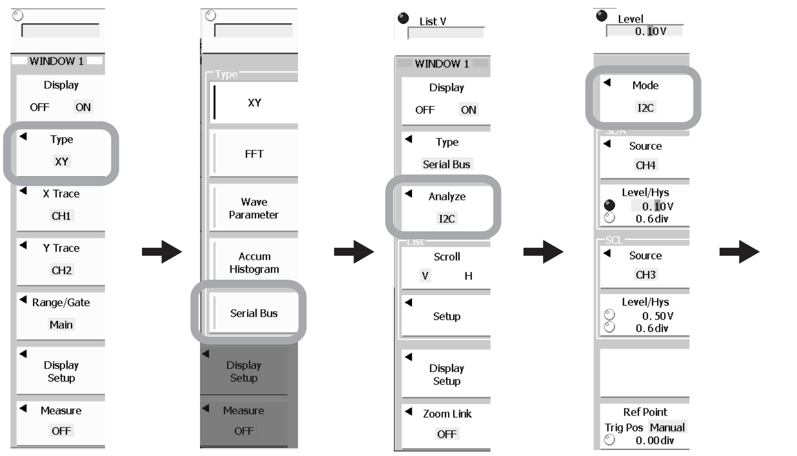
Procedure

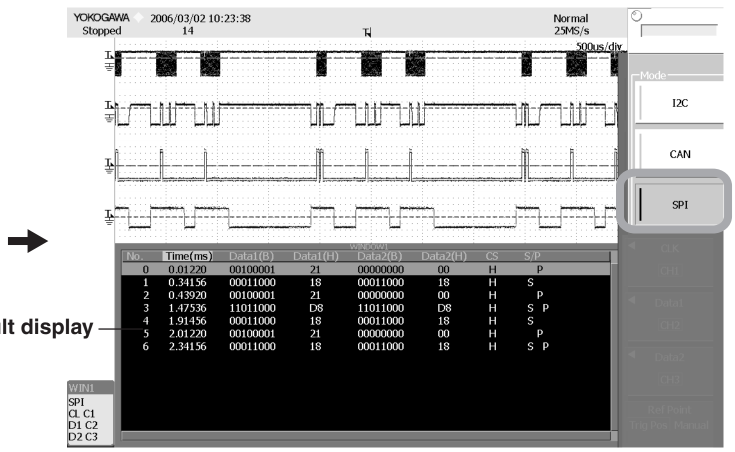


- To exit the menu during operation, press **ESC** located above the soft keys.
- For details on the operation using the rotary knob, SET, and RESET, see sections 4.1 or 4.2 in the *User's Manual (IM701310-01E).*
- For a description of the operation using a USB keyboard or a USB mouse, see section 4.3 in the *User's Manual (IM701310-01E).*

Setting the Analysis Conditions

1. Press **WINDOW1** or **WINDOW2**.
2. Press the **Type** soft key to display the Type menu.
3. Press the **Serial Bus** soft key.
4. Press the **Analyze** soft key to display the Analyze menu.
5. Press the **Mode** soft key to display the Mode menu.
6. Press the **SPI** soft key.

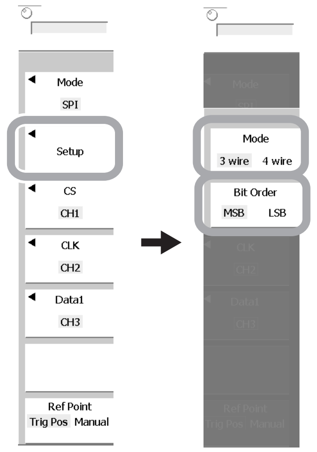




Analysis result display

Selecting Three-Wire or Four-Wire and Selecting the Bit Order

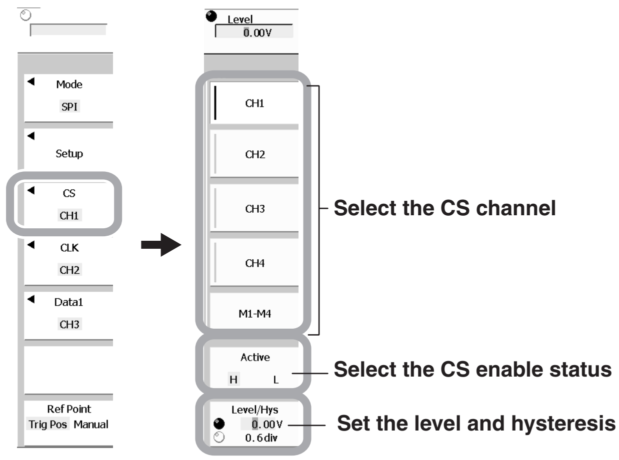
7. Press the **Setup** soft key to display a menu used to set three-wire or four-wire.
8. Press the **Mode** soft key to select 3 wire or 4 wire.
9. Press the **Bit Order** soft key to set the read direction of the I/O data bits to MSB or LSB.



10. Press **ESC**.

Setting the Chip Select Signal (CS)

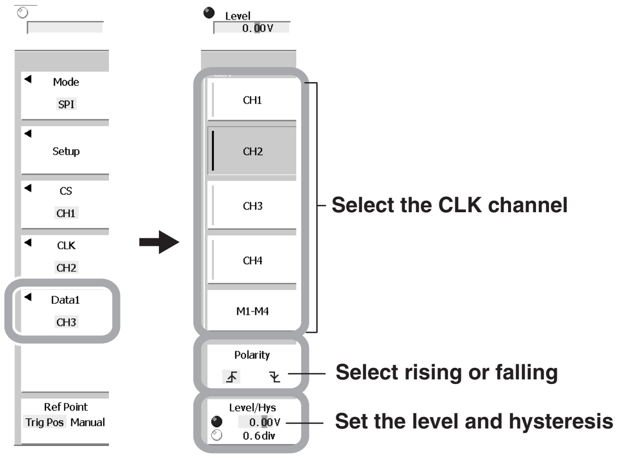
11. Press the **CS** soft key to display the CS menu.
12. Press the soft key corresponding to the channel to be assigned to CS.
13. Press the **Active** soft key to set the CS enable status to H (high) or L (low).
14. Press the **Level/Hys** soft key to move the cursor to the level value.
15. Turn the **rotary knob** to set the level for determining the status of the CS signal.
16. Press the **Level/Hys** soft key to move the cursor to the hysteresis value.
17. Turn the **rotary knob** to set the hysteresis.



18. Press **ESC**.

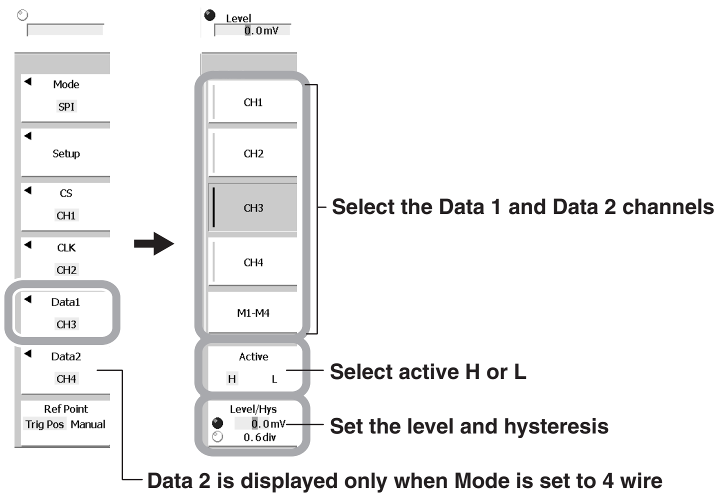
Setting the Clock Signal (CLK)

19. Press the **CLK** soft key to display the CLK menu.
20. Press the soft key corresponding to the channel to be assigned to the clock signal.
21. Press the **Polarity** soft key to select rising (⨍) or falling (⤓).
22. Press the **Level/Hys** soft key to move the cursor to the level value.
23 Turn the **rotary knob** to set the level used to determine the rising or falling edge of the clock signal.
24. Press the **Level/Hys** soft key to move the cursor to the hysteresis value.
25. Turn the **rotary knob** to set the hysteresis.
26. Press **ESC**.



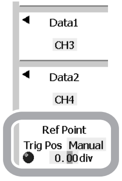
Setting Data 1 (Data 2)

27. Press the **Data 1** or **Data 2** soft key (for four-wire) to display the Data 1 or Data 2 menu.
28. Press the soft key corresponding to the channel to be assigned to Data 1 or Data 2.
29. Press the **Active** soft key to set H or L. If set to H, the signal is determined to be 1 when the level for determining the status is exceeded and 0 otherwise. If set to L, the signal is determined to be to 1 when the level for determining the status is not exceeded and 0 otherwise.
30. Press the **Level/Hys** soft key to move the cursor to the level value.
31. Turn the **rotary knob** to set the level used to determine status.
32. Press the **Level/Hys** soft key to move the cursor to the hysteresis value.
33. Turn the **rotary knob** to set the hysteresis.
34. Press **ESC**.



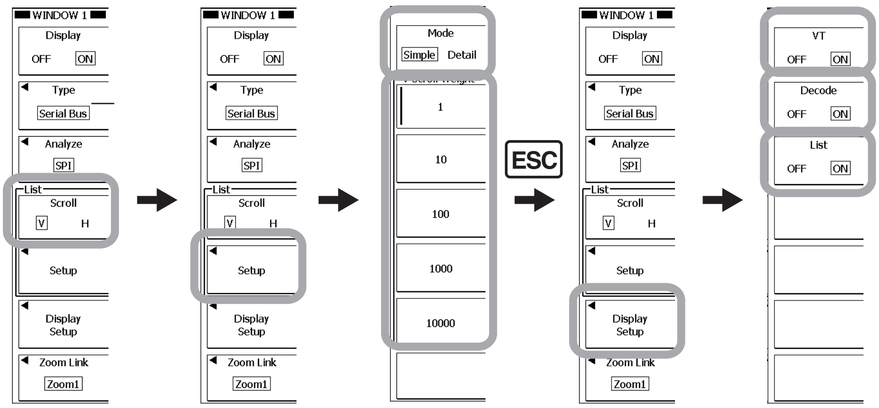
Setting the Analysis Reference Point

35. Pressing the **Ref Point** soft key to select Trig Pos or Manual.
    If you selected Manual, turn the rotary knob to set the reference point in the range of ±5.00 divisions.
36. Press **ESC**.



Display the Analysis Results

37. Press the **Scroll** soft key to select V (scroll vertically) or H (scroll horizontally).
38. Press the **Setup** soft key to display a menu used to set the analysis display.
39. Press the **Mode** soft key to select Simple (simple display) or Detail (detailed display).
40. Press any of the soft keys **1**, **10**, **100**, **1000**, and **10000** to set the step number for scrolling.
41. Press **ESC**.
42. Press the **Display Setup** soft key to display the Display Setup menu.
43. Press the **VT** soft key to select ON or OFF. If ON is specified, the waveform is displayed on a normal voltage-time axis.
44. Likewise, press the **Decode** (firmware version 3.6 or later) or **List** soft key to select ON or OFF. If set to ON, the decoded data or a list of analysis results is displayed.
45. Press **ESC**.

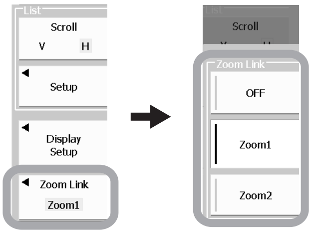


Note

The data of the simple display and detail display of the analysis results can be saved directly to a storage medium in CSV format (.csv extension). For details, see section 4.5.

Zoom Link

46. Press the **Zoom Link** soft key to display the Zoom Link menu.
47. Press any of the soft keys, **OFF**, **Zoom1**, or **Zoom2** to set the link to the zoom screen. The zoom position (the center of the zoom box) can be moved to the head of the byte selected in the analysis result list by linking Zoom1 or Zoom2.



Explanation

Setting the Analysis Conditions

You can set the following conditions.

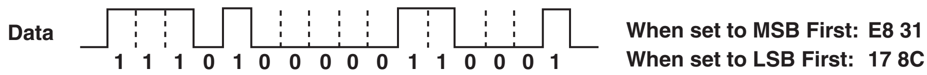
Three-Wire or Four-Wire

Select three-wire SPI or four-wire SPI. One type of data (Data 1) can be analyzed for three-wire SPI and two types of data (Data 1 and Data 2) for four-wire SPI.

Read Direction of the I/O Data Bits (Bit Order)

You can select the read direction of the bits according to the signal flow.

MSB First	Select this when the I/O data signal is flowing MSB first.
LSB First	Select this when the I/O data signal is flowing LSB first.



Chip Select Signal (CS (SS))

You can select CH1 to CH4 or M1 to M4 signals for the CS signal on the SPI bus.

- **Level**
  When the channel signal is set to be the CS signal, you can set the level for determining the high (H) or low (L) status of the CS signal for each channel.
  Selectable range: ±10 divisions around the vertical position
  Resolution: 0.01 divisions (For example, the resolution for 2 mV/div is 0.02 mV.)

- **Hysteresis**
  Selectable range: 0.0 division to 4.0 divisions
  Resolution: 0.1 divisions

- **Enable Status of the CS Signal (Active)**
  You can select the enable status of the CS signal. The setting applies to all CS signals.

H	Analyzes the I/O data when the CS signal is higher than the level for determining the status.
L	Analyzes the I/O data when the CS signal is less than the level for determining the status.

Clock Signal

Set the channel to which the clock signal on the SPI Bus is being applied. The status of the I/O data is determined by synchronizing to the clock signal. You can set the detection level, slope, and hysteresis of the clock signal.

- **Level**
  You can set the level* for detecting the synchronization clock.
  Selectable range: ±10 divisions around the vertical position
  Resolution: 0.01 divisions (For example, the resolution for 2 mV/div is 0.02 mV.)
- **Hysteresis**
  You can set the hysteresis* on the level for detecting the synchronization clock.
  Selectable range: 0.0 division to 4.0 divisions
  Resolution: 0.1 divisions

  * **Detecting the Synchronization Clock**
  - When the level of the clock signal changes from below the specified lower limit of hysteresis to above and including the upper limit of hysteresis, it is detected as a synchronization clock.
  - When the level of the clock signal changes from above the specified upper limit of hysteresis to below and including the lower limit of hysteresis, it is detected as a synchronization clock.
  - For all other cases, it is not detected as a synchronization clock.
- **Slope (Polarity)**
  You can select which slope edge, rising or falling, of the synchronization clock is to be detected.

⨍	Rising slope
⺄	Falling slope

Analyzed Data (Data 1 and Data 2)

The data below can be analyzed for the I/O data signal (Data 1 and Data 2) on the SPI bus.
- Waveform Data
  Analysis can be performed regardless of whether the waveform acquisition is in progress. If the waveform acquisition is in progress, the analysis results are updated in real-time in sync with the displayed waveform. The waveform data stored in the history memory (waveform data of the record number selected using HISTORY menu > Select) is also analyzed.
- Loaded Acquisition Data (ACQ data)
- **Condition for Determining the Status (Active)**
  Set the condition for determining 1 or 0.

H	Determines that the status is 1 when the signal is greater than or equal to the level for determining the status or 0 otherwise.
L	Determines that the status is 1 when the signal is less than or equal to the level for determining the status or 0 otherwise.

- **Level**
  You can set the level* for determining the status of the data to be analyzed.
  Selectable range: ±10 divisions around the vertical position
  Resolution: 0.01 divisions (For example, the resolution for 2 mV/div is 0.02 mV.)
- **Hysteresis**
  You can set a hysteresis* on the level for determining the status of the data to be analyzed.
  Selectable range: 0.0 division to 4.0 divisions
  Resolution: 0.1 divisions

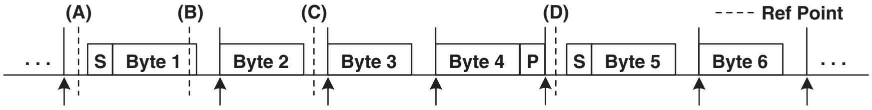
  * **Determining 1 and 0 of the Analyzed Data**
  - **When Active Is Set to H**
    When the level of the analyzed data changes from below the hysteresis lower limit to greater than or equal to the hysteresis upper limit, the data is determined to be 1. When the level of the analyzed data changes from above the hysteresis upper limit to less than or equal to the hysteresis lower limit, the data is determined to be 0.
  - **When Active Is Set to L**
    When the level of the analyzed data changes from above the hysteresis upper limit to less than or equal to the hysteresis lower limit, the data is determined to be 1. When the level of the analyzed data changes from below the hysteresis lower limit to greater than or equal to the hysteresis upper limit, the data is determined to be 0.

Analysis Reference Point (Ref Point)

Select the analysis reference point from below.

Trig Pos	Sets the trigger position to the analysis reference point.
Manual	Manually set the analysis reference point. Selectable range: ±5.00 divisions, resolution: 0.01 divisions

Depending on the analysis reference point (Ref Point), the No. 0 byte in the analysis result list is as follows:



(A): No. 0 byte -> Byte 1 (Byte 2 is No. 1, byte 3 is No. 2, ...)
(B): No. 0 byte -> Byte 1 (Byte 2 is No. 1, byte 3 is No. 2, ...)
(C): No. 0 byte -> Byte 2 (byte 1 is No. -1, byte 3 is No. 1, byte 4 is No. 2...)
(D): No. 0 byte -> Byte 5 (byte 1 is No. -4, ... byte 4 is No. -1, byte 6 is No. 1...)
S: Start condition, P: Stop condition

Unification of the Trigger Setting of the SPI Bus Signal and the Setting of the SPI Bus Signal Analysis and Search

On products with software version 1.80 or higher, the trigger settings of the SPI bus signal on the menu that is entered through the ENHANCED key and the settings of the SPI bus signal analysis and search on the menu entered through the WINDOW1 and ZOOM1 key are common. For details, see page 4-8.

Analysis Range

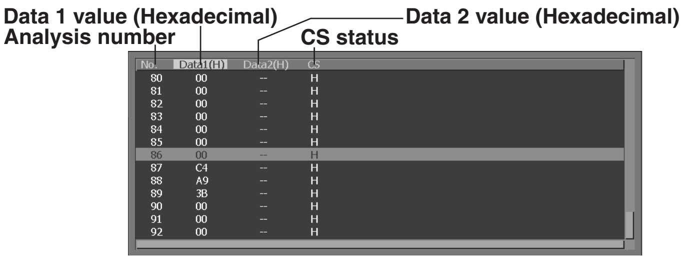
Up to 40000 bytes of analysis data can be displayed.

Analysis Data List (Analysis Result List)

The following items are displayed.

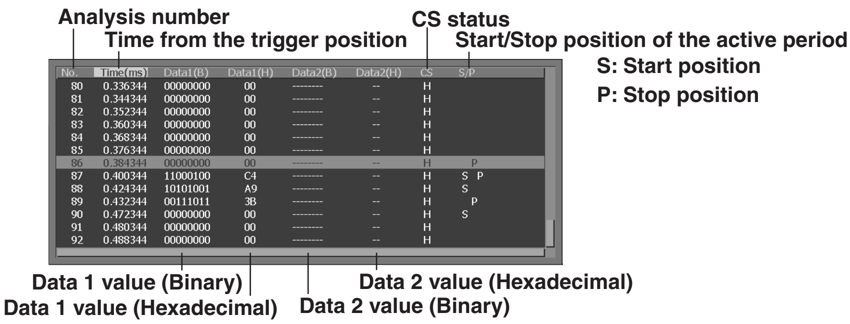
Simple Display

- Analysis number (No.): Up to 40000 points can be displayed.
- Hexadecimal display of the Data 1 (Data 1(H))
- Hexadecimal display of the Data 2 (Data 2(H))
- CS status



Detail Display

- Analysis number (No.): Up to 40000 points can be displayed.
- Time from the trigger position to the start bit of each data (Time(ms))
- Binary display of the Data 1 (Data 1(B))
- Hexadecimal display of the Data 1 (Data 1(H))
- Binary display of the Data 2 (Data 2(B))
- Hexadecimal display of the Data 2 (Data 2(H))
- CS status
- Start/Stop position of the active period (S/P)



Decode Display (firmware version 3.6 or later)

Decodes and displays the values of each field of the SPI bus signal.

Displays the data in hexadecimal notation in cyan.
Displays the group background in gray.

Zoom Link

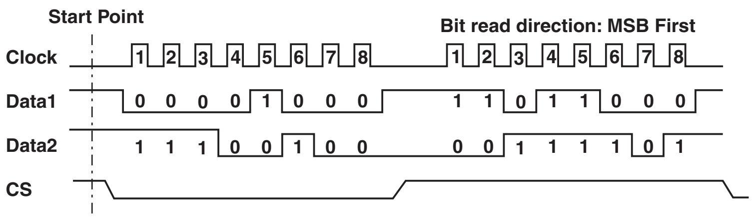
Select from the following:

OFF	Disables the zoom link function.
Zoom1	Links to Zoom1.
Zoom2	Links to Zoom2.

By default, WINDOW1 and WINDOW2 are linked to Zoom1 and Zoom2, respectively. If Zoom1 or Zoom2 is selected and you select (highlight) an arbitrary byte in the analysis result list, the zoom position (the center of the zoom box) of Zoom1 or Zoom2 moves to the head of that byte. Conversely, if you change the zoom position of Zoom1 or Zoom2, the highlight in the analysis result list moves to the byte that is shown in the zoom box of Zoom1 or Zoom2.

Display Example of Analysis Data

Examples of analysis using different analysis conditions are indicated below.



- **Analysis Condition Clock (CH1) = ↑ and CS (CH4) = L**

Display Item	Analysis Data Display
Analysis number (No.)	0
Hexadecimal display of Data1 (Dt1)	08
Hexadecimal display of Data2 (Dt2)	E4
Status of the CS signal (CS)	L

- **Analysis Condition Clock (CH1) = ↑ and CS (CH4) = H**

Display Item	Analysis Data Display
Analysis number (No.)	0
Hexadecimal display of Data1 (Dt1)	D8
Hexadecimal display of Data2 (Dt2)	3D
Status of the CS signal (CS)	H

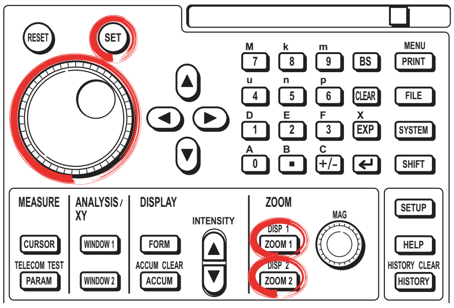
Note

If the CS signal is ON and the CS signal on the main waveform display screen does not contain a transition point from H to L or L to H, the I/O data is not analyzed.

4.4 Searching the Waveform

You can search waveforms that match the specified conditions among the acquired waveforms. The search conditions are the same as the trigger settings of section 4.2.

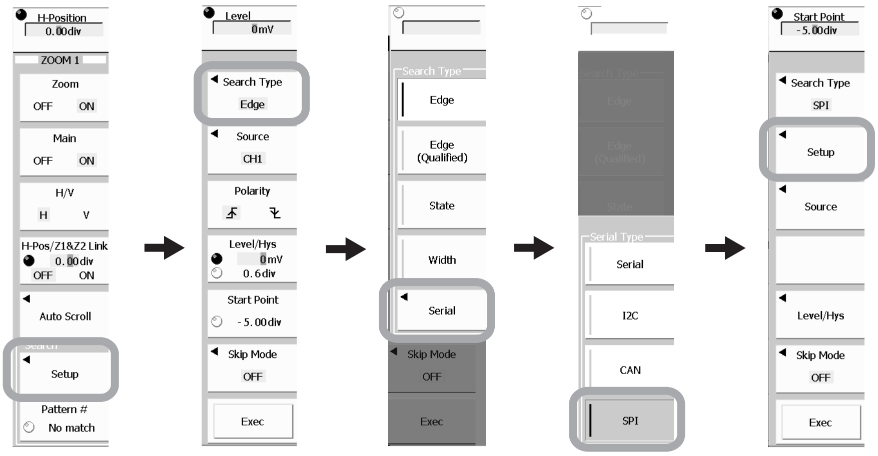
Procedure



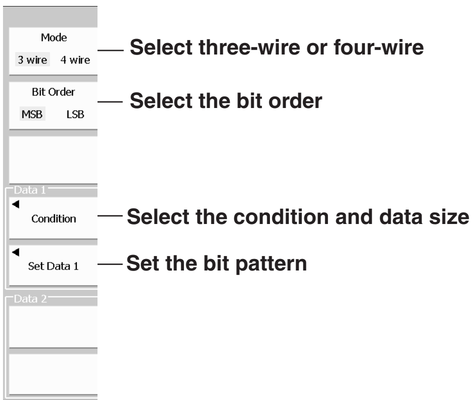
- To exit the menu during operation, press **ESC** located above the soft keys.
- For details on the operation using the rotary knob, SET, and RESET, see sections 4.1 or 4.2 in the *User's Manual (IM701310-01E).*
- For a description of the operation using a USB keyboard or a USB mouse, see section 4.3 in the *User's Manual (IM701310-01E).*

Setting the Search Conditions

1. Press **ZOOM1** or **ZOOM2** to display the ZOOM menu.
2. Press the **Setup** soft key to display the search condition menu.
3. Press the **Search Type** soft key.
4. Press the **Serial** soft key.
5. Press the **SPI** soft key.
6. Press the **Setup** soft key to display the search condition menu.



The procedure of setting the search conditions are the same as the trigger conditions. See page 2-6 (the position of the soft keys may be different).



7. Press **ESC**.

4

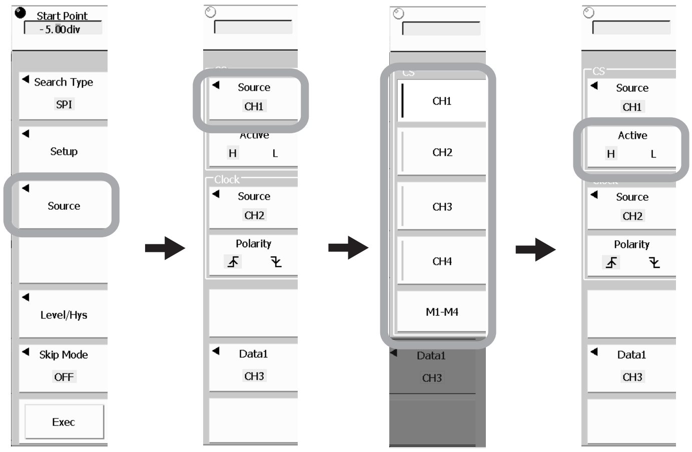
SPI Bus Signal Analysis Function

Setting the Source Channel

8. Press the **Source** soft key to display the source channel menu.

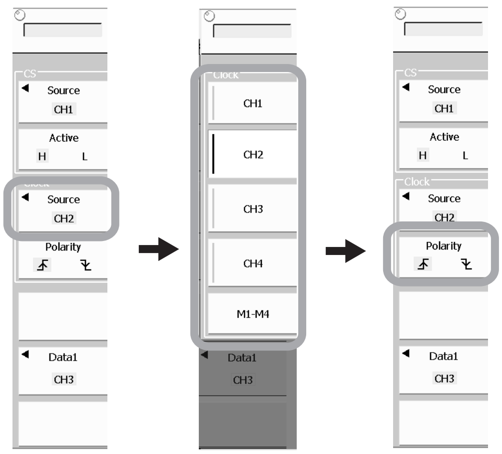
Setting the CS

9. Press the **Source** soft key under CS to display the CS source channel selection menu.
10. Press the soft key corresponding to the channel to assign to CS.
11. Press the **Active** soft key to set the CS enable status to H (high) or L (low).



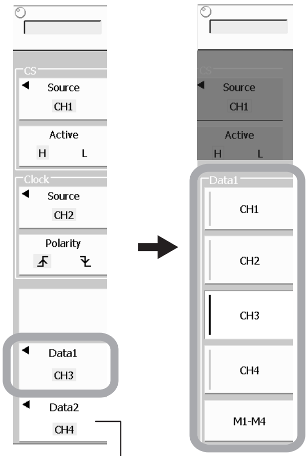
Setting the Clock

12. Press the **Source** soft key under Clock to display the Clock source channel selection menu.
13. Press the soft key corresponding to the channel to assign to the clock signal.
14. Press the **Polarity** soft key to select rising (↑) or falling (↓).



Setting Data 1 (Data 2)

14. Press the **Data1** or **Data2** soft key (for four-wire) to display a menu used to select the Data1 or Data2 channels.
16. Press the soft key corresponding to the channel to be assigned to Data1 or Data2.
17. Press **ESC**.



Data 2 is displayed only when Mode is set to 4 wire

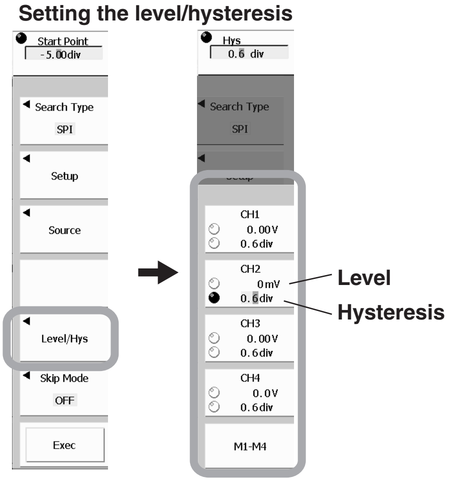
Setting the Level and Hysteresis

18. Press the **Level/Hys** soft key.
19. Press the soft key corresponding to the channel you wish to configure.
20. Turn the **rotary knob** to set the level or hysteresis.
21. Press **ESC**.

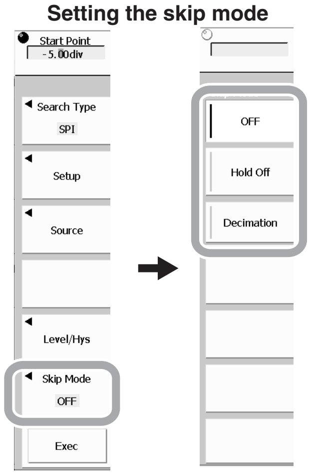
Setting the Skip Mode

22. Set the skip mode as necessary.
    Press the **Skip Mode** soft key to display the skip mode menu.
23. Press the **OFF**, **Hold Off**, or **Decimation** soft key.
24. Turn the **rotary knob** to set the time to skip or the search count.
25. Press **ESC** .

Setting the level/hysteresis



Setting the skip mode



Setting the Search Start Point

26. Turn the **rotary knob** to set the Start Point.

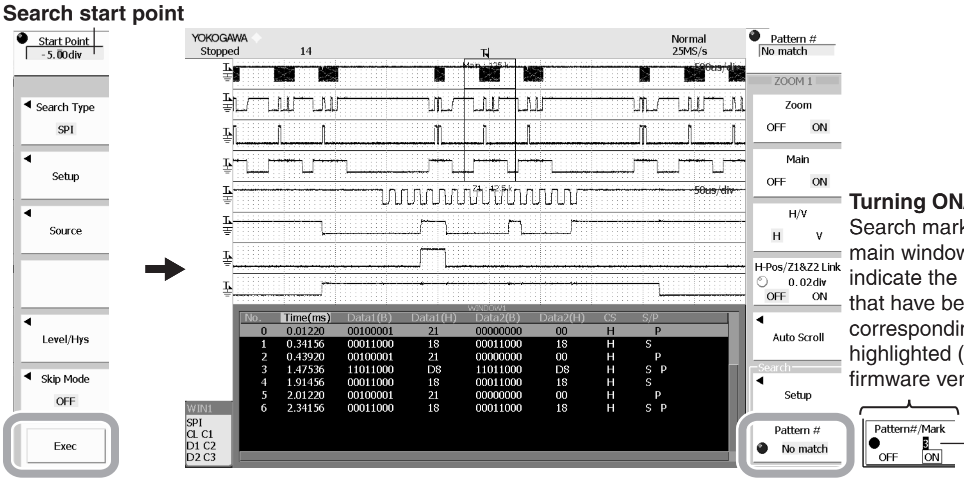
Executing the Search

27. Press the **Exec** soft key. The section that meets the search conditions is shown on the ZOOM display.

Viewing Search Points

28. Press **ESC**.
29. Press the **Pattern #** soft key.
30. Turn the **rotary knob** to select an arbitrary search point. The zoom position moves to the selected search point.

Search start point



Turning ON/OFF the Search Mark ▼
Search marks can be displayed on the main window and zoom window to indicate the locations on the waveform that have been found. The search mark corresponding to the search number is highlighted (applies to products with firmware version 3.6 or later).

Search number

Explanation

Setting the Search Conditions

You can set the following conditions.

- Three-wire or four-wire
- Bit Order
- Condition and bit pattern of Data 1 and Data 2

The search conditions are the same as the trigger conditions. For details, see section 4.2.

Source Channel, Qualification, Level, Hysteresis, Etc.

The settings for the source channel, qualification, trigger level, trigger coupling, and so on are the same as the settings of the analysis conditions. For details, see section 4.3.

Skip Mode

After searching the position that meets the search conditions, the specified amount of time or the number of positions that match the search conditions are skipped.

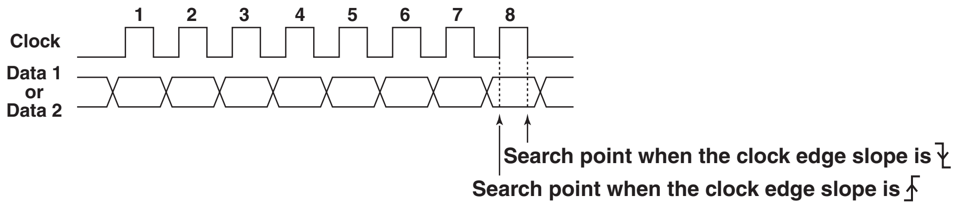
OFF	Searches all sections that meet the search conditions.
Hold Off	Skips the search for the specified time.
Decimation	Skips the specified number of search positions.

Search Start Point

Set the position to start the search in the range of ±5.00 divisions.

Search Point

When the search is executed, the zoom box (ZOOM1 or ZOOM2) moves to the data position where the conditions are met. The search point is the center of this zoom box. The search point is set to the same position as the trigger position (see page 4-8).



Viewing Search Points

If you select an arbitrary search point using Pattern #, the zoom position moves to the selected search point. If the zoom link function is enabled (see page 4-16), the highlight in the analysis result list also moves to the byte containing the selected search point.

Unification of the Trigger Setting of the SPI Bus Signal and the Setting of the SPI Bus Signal Analysis and Search

On products with software version 1.80 or higher, the trigger settings of the SPI bus signal on the menu that is entered through the ENHANCED key and the settings of the SPI bus signal analysis and search on the menu entered through the WINDOW1 and ZOOM1 key are common. For details, see page 4-8.

4.5 Saving the Data of the Analysis Result List

CAUTION

Do not remove the medium (disk) or turn OFF the power when the access indicator is blinking. Doing so can damage the storage medium or destroy the data on the medium.

Procedure

The procedure to save the results is the same as with the I^2C Bus signal analysis.
For the operating procedure, see section 1.5, “Saving the Data of the Detailed Analysis List.”

Explanation

The data of the analysis result (simple display or detail display) can be saved to a specified storage medium in CSV format. The extension is .csv. Data in CSV format is a text file with data separated by commas. It is one of the common data formats used to convert data between spreadsheet and database applications.

Setting the Storage Conditions and Notes When Saving Data

The details are the same as when saving data of the analysis result list of the I^2C bus signal. For details, see section 1.5.

Saved Data

Up to 40000 analysis results in the past from the point when the save operation was executed can be saved.

Data Size

Data size* = (Number of bytes + 4) × 79 [bytes]

* The data size is a reference value. It is not strictly warranted. Use it as a guideline when saving the data.

Data Format

Binary display of Data2
Hexadecimal display of Data2
CS status

```
Analysis  SerialBus(SPI)
Model Nam DL9000
Model Ver *.**
No.       Time(ms)  Data1(B)      Data1(H)   Data2(B)      Data2(H)   CS    S/P
       0  -0.68336  00011110 Bin    1E Hex   11110111 Bin    F7 Hex    L    S   P
       1  -0.66736  10111101 Bin    BD Hex   01111111 Bin    7F Hex    L    S
       2  -0.65936  01000010 Bin    42 Hex   11010100 Bin    D4 Hex    L        P
       3  -0.63536  10010011 Bin    93 Hex   11100111 Bin    E7 Hex    L    S
       4  -0.62736  10111101 Bin    BD Hex   11111111 Bin    FF Hex    L
       5  -0.61936  01000010 Bin    42 Hex   11111111 Bin    FF Hex    L
       6  -0.61136  10001100 Bin    8C Hex   01111111 Bin    7F Hex    L        P
```

Hexadecimal display of Data1
CS active start (S)/stop (P)
Binary display of Data1
Time from the start point
Analysis number

4.6 Error Messages

A message may appear during operation. This section describes the meanings of the messages and their corrective actions. This section lists only the error messages related to the SPI Bus signal analysis function. There are other error messages related to the DL9000 and communications. These messages are described in the *User's Manual (IM701310-01E)* and the *Communication Interface User's Manual (IM701310-17E)*.
You can set the messages to be displayed in English or Japanese. For the procedure of setting the message language, see section 17.1 in the *User's Manual IM701310-01E*.
If the corrective action requires servicing, contact your nearest YOKOGAWA dealer for repairs.

Code	Messages	Corrective Action	Page
58	Search executed is completed, but no record was found that matshed the pattern.	–	4-17
506	Save data do not exist. Check the content to be saved.	Display the analysis result, and then execute the save operation again.	4-9, 4-22

5.1 A List of Commands

ANALysis Group

Command	Function	Page
:ANALysis:SBUS<x>?	Queries all settings related to the serial bus signal analysis function.	5-17
:ANALysis:SBUS<x>:ANALyze?	Queries all settings related to the serial bus signal analysis.	5-17
:ANALysis:SBUS<x>[:ANALyze]:CANBus?	Queries all settings related to the CAN bus signal analysis.	5-17
:ANALysis:SBUS<x>[:ANALyze]:CANBus:BRATe	Sets the bit rate (data transfer rate) of the CAN bus signal analysis or queries the current setting.	5-18
:ANALysis:SBUS<x>[:ANALyze]:CANBus:FJUMp:ACK	Executes a field jump to the ACK Field in the results of the CAN bus signal analysis.	5-18
:ANALysis:SBUS<x>[:ANALyze]:CANBus:FJUMp:CONTrol	Executes a field jump to the Control Field in the results of the CAN bus signal analysis.	5-18
:ANALysis:SBUS<x>[:ANALyze]:CANBus:FJUMp:CRC	Executes a field jump to the CRC Field in the results of the CAN bus signal analysis.	5-18
:ANALysis:SBUS<x>[:ANALyze]:CANBus:FJUMp:DATA	Executes a field jump to the Data Field in the results of the CAN bus signal analysis.	5-18
:ANALysis:SBUS<x>[:ANALyze]:CANBus:FJUMp:IDENtifier	Executes a field jump to the Identifier Field in the results of the CAN bus signal analysis.	5-18
:ANALysis:SBUS<x>[:ANALyze]:CANBus:FJUMp:SOF	Executes a field jump to the SOF Field in the results of the CAN bus signal analysis.	5-18
:ANALysis:SBUS<x>[:ANALyze]:CANBus:RECessive	Sets the recessive level (bus level) of the CAN bus signal analysis or queries the current setting.	5-18
:ANALysis:SBUS<x>[:ANALyze]:CANBus:SPOint	Sets the sample point of the CAN bus signal analysis or queries the current setting.	5-19
:ANALysis:SBUS<x>[:ANALyze]:CANBus:TRACe	Sets the source channel of the CAN bus signal analysis or queries the current setting.	5-19
:ANALysis:SBUS<x>[:ANALyze]:DECode	Turns the serial bus signal analysis decoding display ON/OFF or queries the current status.	5-19
:ANALysis:SBUS<x>[:ANALyze]:I2CBus?	Queries all settings related to the I^{2}C bus signal analysis.	5-19
:ANALysis:SBUS<x>[:ANALyze]:I2CBus:CLOCk	Sets the clock channel of the I^{2}C bus signal analysis or queries the current setting.	5-19
:ANALysis:SBUS<x>[:ANALyze]:I2CBus:DTRace	Sets the data channel of the I^{2}C bus signal analysis or queries the current setting.	5-19
:ANALysis:SBUS<x>[:ANALyze]:LINBus?	Queries all settings related to the LIN bus signal analysis.	5-19
:ANALysis:SBUS<x>[:ANALyze]:LINBus:BRATe	Sets the LIN bus signal analysis bitrate (data transfer rate) or queries the current setting.	5-20
:ANALysis:SBUS<x>[:ANALyze]:LINBus:REVision	Sets the LIN bus signal analysis revision (1.3 or 2.0) or queries the current setting.	5-20
:ANALysis:SBUS<x>[:ANALyze]:LINBus:TRACe	Sets the LIN bus signal analysis trace or queries the current setting.	5-20

Command	Function	Page
:ANALysis:SBUS<x>[:ANALyze]:LIST?	Queries all settings related to the list display of the serial bus signal analysis.	5-20
:ANALysis:SBUS<x>[:ANALyze]:LIST:DISPlay	Turns the serial bus signal analysis list display ON/OFF or queries the current status.	5-20
:ANALysis:SBUS<x>[:ANALyze]:LIST:ITEM?	Queries the item in the list display of the serial bus signal analysis.	5-20
:ANALysis:SBUS<x>[:ANALyze]:LIST:MODE	Sets the mode of the list display of the serial bus signal analysis or queries the current setting.	5-20
:ANALysis:SBUS<x>[:ANALyze]:LIST:SCRoll	Sets the scroll method of the list display of the serial bus signal analysis or queries the current setting.	5-21
:ANALysis:SBUS<x>[:ANALyze]:LIST:VALue?	Queries the analyzed value of the specified list display number in the serial bus signal analysis.	5-21
:ANALysis:SBUS<x>[:ANALyze]:MODE	Sets the serial bus signal analysis mode or queries the current setting.	5-21
:ANALysis:SBUS<x>[:ANALyze]:RPOint	Sets the analysis reference point of the serial bus signal analysis or queries the current setting.	5-21
:ANALysis:SBUS<x>[:ANALyze]:SPIBus?	Queries all settings related to the SPI bus signal analysis.	5-21
:ANALysis:SBUS<x>[:ANALyze]:SPIBus:CLOCk?	Queries all settings related to the clock channel of the SPI bus signal analysis.	5-21
:ANALysis:SBUS<x>[:ANALyze]:SPIBus:CLOCk:POLarity	Sets the polarity of the clock channel of the SPI bus signal analysis or queries the current setting.	5-21
:ANALysis:SBUS<x>[:ANALyze]:SPIBus:CLOCk:SOURce	Sets the clock channel of the SPI bus signal analysis or queries the current setting.	5-22
:ANALysis:SBUS<x>[:ANALyze]:SPIBus:CS?	Queries all settings related to the chip select channel of the SPI bus signal analysis.	5-22
:ANALysis:SBUS<x>[:ANALyze]:SPIBus:CS:ACTive	Sets the active level of the chip select channel of the SPI bus signal analysis or queries the current setting.	5-22
:ANALysis:SBUS<x>[:ANALyze]:SPIBus:CS:TRACe	Sets the chip select channel of the SPI bus signal analysis or queries the current setting.	5-22
:ANALysis:SBUS<x>[:ANALyze]:SPIBus:DATA<x>?	Queries all settings related to the data of the SPI bus signal analysis.	5-22
:ANALysis:SBUS<x>[:ANALyze]:SPIBus:DATA<x>:ACTive	Sets the active level of the data of the SPI bus signal analysis or queries the current setting.	5-22
:ANALysis:SBUS<x>[:ANALyze]:SPIBus:DATA<x>:TRACe	Sets the data channel of the SPI bus signal analysis or queries the current setting.	5-23
:ANALysis:SBUS<x>[:ANALyze]:SPIBus:SETup?	Queries all settings related to the SPI bus signal analysis setup.	5-23
:ANALysis:SBUS<x>[:ANALyze]:SPIBus[:SETup]:BITorder	Sets the bit order of the SPI bus signal analysis or queries the current setting.	5-23
:ANALysis:SBUS<x>[:ANALyze]:SPIBus[:SETup]:MODE	Sets the wiring system of the SPI bus signal analysis (three-wire or four-wire) or queries the current setting.	5-23
:ANALysis:SBUS<x>[:ANALyze]:TRACe<x>?	Queries all settings related to the threshold level of the source channel of the serial bus signal analysis.	5-23
:ANALysis:SBUS<x>[:ANALyze]:TRACe<x>:HYSTeresis	Sets the hysteresis of the threshold level of the source channel of the serial bus signal analysis or queries the current setting.	5-23
:ANALysis:SBUS<x>[:ANALyze]:TRACe<x>:LEVel	Sets the level of the threshold level of the source channel of the serial bus signal analysis or queries the current setting.	5-24

Command	Function	Page
:ANALysis:SBUS<x>:ZLINkage	Sets the zoom link of the serial bus signal analysis or queries the current setting.	5-24
:ANALysis:TYPE<x>	Sets the analysis function type or queries the current setting.	5-24

MATH Group

Command	Function	Page
:MATH<x>:OPERation	Sets the operator or queries the current setting.	5-25
:MATH<x>:SBIT?	Queries all settings related to the stuff bit computation.	5-25
:MATH<x>:SBIT:BRATe	Sets the bit rate (data transfer rate) of the stuff bit computation or queries the current setting.	5-25
:MATH<x>:SBIT:HISTory:ABORt	Cancels history computation for stuff bit computation.	5-25
:MATH<x>:SBIT:HISTory:EXECute	Executes history computation for stuff bit computation.	5-25
:MATH<x>:SBIT:HYSTeresis	Sets the hysteresis of the stuff bit computation or queries the current setting.	5-25
:MATH<x>:SBIT:LEVel	Sets the threshold level of the stuff bit computation or queries the current setting.	5-25
:MATH<x>:SBIT:RECessive	Sets the recessive level (bus level) of the stuff bit computation or queries the current setting.	5-25
:MATH<x>:SBIT:SPOint	Sets the sample point of the stuff bit computation or queries the current setting.	5-25

SEARch Group

Command	Function	Page
:SEARch<x>:CANBus?	Queries all settings related to the CAN bus signal search.	5-26
:SEARch<x>:CANBus:SETup?	Queries all settings related to the CAN bus signal search setup.	5-26
:SEARch<x>:CANBus[:SETup]:ACK	Sets the ACK condition of the CAN bus signal search or queries the current setting.	5-26
:SEARch<x>:CANBus[:SETup]:BRATe	Sets the bit rate (data transfer rate) of the CAN bus signal search or queries the current setting.	5-26
:SEARch<x>:CANBus[:SETup]:DATA?	Queries all settings related to the CAN bus signal search data.	5-26
:SEARch<x>:CANBus[:SETup]:DATA:BORDer	Sets the byte order of the CAN bus signal search data or queries the current setting.	5-26
:SEARch<x>:CANBus[:SETup]:DATA:CONDition	Sets the data condition of the CAN bus signal search or queries the current setting.	5-27
:SEARch<x>:CANBus[:SETup]:DATA:DATA<x>	Sets the comparison data of the CAN bus signal search data or queries the current setting.	5-27
:SEARch<x>:CANBus[:SETup]:DATA:DLC	Sets the number of valid bytes (DLC) of the CAN bus signal search data or queries the current setting.	5-27
:SEARch<x>:CANBus[:SETup]:DATA:HEXA	Sets the CAN bus signal search data in hexadecimal notation.	5-27
:SEARch<x>:CANBus[:SETup]:DATA:MSBLsb	Sets the MSB and LSB bits of the CAN bus signal search data or queries the current setting.	5-27
:SEARch<x>:CANBus[:SETup]:DATA:PATTern	Sets the CAN bus signal search data in binary notation or queries the current setting.	5-27
:SEARch<x>:CANBus[:SETup]:DATA:SIGN	Sets the sign of the CAN bus signal search data or queries the current setting.	5-27
:SEARch<x>:CANBus[:SETup]:IDEXt?	Queries all settings related to the ID of the extended format of the CAN bus signal search.	5-28
:SEARch<x>:CANBus[:SETup]:IDEXt:HEXA	Sets the ID of the extended format of the CAN bus signal search in hexadecimal notation.	5-28
:SEARch<x>:CANBus[:SETup]:IDEXt:PATTern	Sets the ID of the extended format of the CAN bus signal search in binary notation or queries the current setting.	5-28
:SEARch<x>:CANBus[:SETup]:IDSTd?	Queries all settings related to the ID of the standard format of the CAN bus signal search.	5-28
:SEARch<x>:CANBus[:SETup]:IDSTd:HEXA	Sets the ID of the standard format of the CAN bus signal search in hexadecimal notation.	5-28
:SEARch<x>:CANBus[:SETup]:IDSTd:PATTern	Sets the ID of the standard format of the CAN bus signal search in binary notation or queries the current setting.	5-28

Command	Function	Page
:SEARch<x>:CANBus[:SETup]:MODE	Sets the CAN bus signal search mode or queries the current setting.	5-28
:SEARch<x>:CANBus[:SETup]:RECessive	Sets the recessive level (bus level) of the CAN bus signal search or queries the current setting.	5-28
:SEARch<x>:CANBus[:SETup]:RTR	Sets the RTR of the CAN bus signal search or queries the current setting.	5-28
:SEARch<x>:CANBus[:SETup]:SPOint	Sets the sample point of the CAN bus signal search or queries the current setting.	5-29
:SEARch<x>:CANBus[:SETup]:TRACe	Sets the source channel of the CAN bus signal search or queries the current setting.	5-29
:SEARch<x>:I2CBus?	Queries all settings related to the I^2C bus signal search.	5-29
:SEARch<x>:I2CBus:CLOCk?	Queries all settings related to the clock of the I^2C bus signal search.	5-29
:SEARch<x>:I2CBus:CLOCk:SOURce	Sets the clock trace of the I^2C bus signal search or queries the current setting.	5-29
:SEARch<x>:I2CBus:SETup?	Queries all settings related to the I^2C bus signal search setup.	5-29
:SEARch<x>:I2CBus[:SETup]:ADATa?	Queries all settings related to the address of the I^2C bus signal search.	5-30
:SEARch<x>:I2CBus[:SETup]:ADATa:BIT10address?	Queries all settings related to the 10-bit address of the I^2C bus signal search.	5-30
:SEARch<x>:I2CBus[:SETup]:ADATa:BIT10address:HEXA	Sets the 10-bit address of the I^2C bus signal search in hexadecimal notation.	5-30
:SEARch<x>:I2CBus[:SETup]:ADATa:BIT10address:PATTern	Sets the 10-bit address of the I^2C bus signal search in binary notation or queries the current setting.	5-30
:SEARch<x>:I2CBus[:SETup]:ADATa:BIT7ADdress?	Queries all settings related to the 7-bit address of the I^2C bus signal search.	5-30
:SEARch<x>:I2CBus[:SETup]:ADATa:BIT7ADdress:HEXA	Sets the 7-bit address of the I^2C bus signal search in hexadecimal notation.	5-30
:SEARch<x>:I2CBus[:SETup]:ADATa:BIT7ADdress:PATTern	Sets the 7-bit address of the I^2C bus signal search in binary notation or queries the current setting.	5-31
:SEARch<x>:I2CBus[:SETup]:ADATa:BIT7APsub?	Queries all settings related to the 7-bit + Sub address of the I^2C bus signal search.	5-31
:SEARch<x>:I2CBus[:SETup]:ADATa:BIT7APsub:ADDRess?	Queries all settings related to the 7-bit address of the 7-bit + Sub address of the I^2C bus signal search.	5-31
:SEARch<x>:I2CBus[:SETup]:ADATa:BIT7APsub:ADDRess:HEXA	Sets the 7-bit address of the 7-bit + Sub address of the I^2C bus signal search in hexadecimal notation.	5-31
:SEARch<x>:I2CBus[:SETup]:ADATa:BIT7APsub:ADDRess:PATTern	Sets the 7-bit address of the 7-bit + Sub address of the I^2C bus signal search in binary notation or queries the current setting.	5-31
:SEARch<x>:I2CBus[:SETup]:ADATa:BIT7APsub:SADDress?	Queries all settings related to the Sub address of the 7-bit + Sub address of the I^2C bus signal search.	5-31
:SEARch<x>:I2CBus[:SETup]:ADATa:BIT7APsub:SADDress:HEXA	Sets the Sub address of the 7-bit + Sub address of the I^2C bus signal search in hexadecimal notation.	5-31
:SEARch<x>:I2CBus[:SETup]:ADATa:BIT7APsub:SADDress:PATTern	Sets the Sub address of the 7-bit + Sub address of the I^2C bus signal search in binary notation or queries the current setting.	5-32
:SEARch<x>:I2CBus[:SETup]:ADATa:TYPE	Sets the address type of the I^2C bus signal search or queries the current setting.	5-32
:SEARch<x>:I2CBus[:SETup]:DATA?	Queries all settings related to the data of the I^2C bus signal search.	5-32
:SEARch<x>:I2CBus[:SETup]:DATA:BYTE	Sets the number of data bytes of the I^2C bus signal search or queries the current setting.	5-32
:SEARch<x>:I2CBus[:SETup]:DATA:CONDition	Sets the determination method (match or not match) of the data of the I^2C bus signal search or queries the current setting.	5-32

Command	Function	Page
:SEARch<x>:I2CBus[:SETup]:DATA:DPOSition	Sets the position for comparing the data pattern of the I^2C bus signal search or queries the current setting.	5-32
:SEARch<x>:I2CBus[:SETup]:DATA:HEXA<x>	Sets the data of the I^2C bus signal search in hexadecimal notation.	5-32
:SEARch<x>:I2CBus[:SETup]:DATA:MODE	Enables/Disables the data conditions of the I^2C bus signal search or queries the current setting.	5-32
:SEARch<x>:I2CBus[:SETup]:DATA:PATTern<x>	Sets the data of the I^2C bus signal search in binary notation or queries the current setting.	5-33
:SEARch<x>:I2CBus[:SETup]:DATA:PMODe	Sets the pattern comparison start position mode of the data of the I^2C bus signal search or queries the current setting.	5-33
:SEARch<x>:I2CBus[:SETup]:GCALl?	Queries all settings related to the general call of the I^2C bus signal search.	5-33
:SEARch<x>:I2CBus[:SETup]:GCALl:BIT7maddress?	Queries all settings related to the 7-bit master address of the general call of the I^2C bus signal search.	5-33
:SEARch<x>:I2CBus[:SETup]:GCALl:BIT7maddress:HEXA	Sets the 7-bit master address of the general call of the I^2C bus signal search in hexadecimal notation.	5-33
:SEARch<x>:I2CBus[:SETup]:GCALl:BIT7maddress:PATTern	Sets the 7-bit master address of the general call of the I^2C bus signal search in binary notation or queries the current setting.	5-33
:SEARch<x>:I2CBus[:SETup]:GCALl:SBYTe	Sets the second byte type of the general call of the I^2C bus signal search or queries the current setting.	5-33
:SEARch<x>:I2CBus[:SETup]:MODE	Sets the search mode of the I^2C bus signal search or queries the current setting.	5-34
:SEARch<x>:I2CBus[:SETup]:NAIGnore?	Queries all settings related to the NON ACK ignore mode of the I^2C bus signal search.	5-34
:SEARch<x>:I2CBus[:SETup]:NAIGnore:HSMode	Sets whether to ignore NON ACK in high speed mode of the I^2C bus signal search or queries the current setting.	5-34
:SEARch<x>:I2CBus[:SETup]:NAIGnore:RACCess	Sets whether to ignore NON ACK in read access mode of the I^2C bus signal search or queries the current setting.	5-34
:SEARch<x>:I2CBus[:SETup]:NAIGnore:SBYTe	Sets whether to ignore NON ACK in the start byte of the I^2C bus signal search or queries the current setting.	5-34
:SEARch<x>:I2CBus[:SETup]:SBHSmode?	Queries all settings related to the start byte and high speed mode of the I^2C bus signal search.	5-34
:SEARch<x>:I2CBus[:SETup]:SBHSmode:TYPE	Sets the type of the start byte or high speed mode of the I^2C bus signal search or queries the current setting.	5-34
:SEARch<x>:LINBus?	Queries all settings related to the LIN bus signal search or queries the current setting.	5-34
:SEARch<x>:LINBus:SETup?	Queries all settings related to setup of the LIN bus signal search or queries the current setting.	5-35
:SEARch<x>:LINBus[:SETup]:BRATe	Sets the LIN bus signal search bitrate (data transfer rate) or queries the current setting.	5-35
:SEARch<x>:LINBus[:SETup]:DATA?	Queries all settings related to data of the LIN bus signal search or queries the current setting.	5-35
:SEARch<x>:LINBus[:SETup]:DATA:BNUM	Sets the number of LIN bus signal search data bytes or queries the current setting.	5-35
:SEARch<x>:LINBus[:SETup]:DATA:CONDition	Sets the LIN bus signal search data or queries the current setting.	5-35
:SEARch<x>:LINBus[:SETup]:DATA:HEXA	Sets the LIN bus signal search data in hexadecimal.	5-35

Command	Function	Page
:SEARch<x>:LINBus[:SETup]:DATA:PATTern	Sets the LIN bus signal search data in binary or queries the current setting.	5-35
:SEARch<x>:LINBus[:SETup]:ID?	Queries all settings related to ID of the LIN bus signal search or queries the current setting.	5-35
:SEARch<x>:LINBus[:SETup]:ID:HEXA	Sets the LIN bus signal search ID in hexadecimal.	5-36
:SEARch<x>:LINBus[:SETup]:ID:PATTern	Sets the LIN bus signal search ID in binary or queries the current setting.	5-36
:SEARch<x>:LINBus[:SETup]:MODE	Sets the LIN bus signal search mode or queries the current setting.	5-36
:SEARch<x>:LINBus[:SETup]:TRACe	Sets the LIN bus signal search trace or queries the current setting.	5-36
:SEARch<x>:SPIBus?	Queries all settings related to the SPI bus signal search.	5-36
:SEARch<x>:SPIBus:CLOCk	Queries all settings related to the clock channel of the SPI bus signal search.	5-36
:SEARch<x>:SPIBus:CLOCk:POLarity	Sets the polarity of the clock channel of the SPI bus signal search or queries the current setting.	5-36
:SEARch<x>:SPIBus:CLOCk:SOURce	Sets the clock channel of the SPI bus signal search or queries the current setting.	5-36
:SEARch<x>:SPIBus:CS?	Queries all settings related to the chip select channel of the SPI bus signal search.	5-37
:SEARch<x>:SPIBus:CS:ACTive	Sets the active level of the chip select channel of the SPI bus signal search or queries the current setting.	5-37
:SEARch<x>:SPIBus:CS:TRACe	Sets the chip select channel of the SPI bus signal search or queries the current setting.	5-37
:SEARch<x>:SPIBus:SETup?	Queries all settings related to the SPI bus signal search setup.	5-37
:SEARch<x>:SPIBus[:SETup]:BITorder	Sets the bit order of the SPI bus signal search or queries the current setting.	5-37
:SEARch<x>:SPIBus[:SETup]:DATA<x>?	Queries all settings related to the data of the SPI bus signal search.	5-37
:SEARch<x>:SPIBus[:SETup]:DATA<x>:BYTE	Sets the number of bytes of the data of the SPI bus signal search or queries the current setting.	5-37
:SEARch<x>:SPIBus[:SETup]:DATA<x>:CONDition	Sets the determination method (match or not match) of the data of the SPI bus signal search or queries the current setting.	5-37
:SEARch<x>:SPIBus[:SETup]:DATA<x>:DPOSition	Sets the pattern comparison start position of the data of the SPI bus signal search or queries the current setting.	5-38
:SEARch<x>:SPIBus[:SETup]:DATA<x>:HEXA<x>	Sets the data of the SPI bus signal search in hexadecimal notation.	5-38
:SEARch<x>:SPIBus[:SETup]:DATA<x>:PATTern<x>	Sets the data of the SPI bus signal search in binary notation or queries the current setting.	5-38
:SEARch<x>:SPIBus[:SETup]:DATA<x>:TRACe	Sets the source channel of the data of the SPI bus signal search or queries the current setting.	5-38
:SEARch<x>:SPIBus[:SETup]:MODE	Sets the wiring system of the SPI bus signal search (three-wire or four-wire) or queries the current setting.	5-38
:SEARch<x>:STRace	Sets the search source channel or queries the current setting.	5-38
:SEARch<x>:TRACe<x>:LEVel	Sets the threshold level of the trace or queries the current setting.	5-38
:SEARch<x>:TYPE	Sets the search type or queries the current setting.	5-38

TRIGger Group

Command	Function	Page
:TRIGger:EINTerval:EVENt<x>:CANBus?	Queries all settings related to the CAN bus trigger of the event.	5-39
:TRIGger:EINTerval:EVENt<x>:CANBus:ACK	Sets the ACK condition of the CAN bus signal trigger or queries the current setting.	5-39
:TRIGger:EINTerval:EVENt<x>:CANBus:BRATe	Sets the bit rate (data transfer rate) of the CAN bus signal trigger or queries the current setting.	5-39

Command	Function	Page
:TRIGger:EINTerval:EVENt<x>:CANBus:DATA?	Queries all settings related to the CAN bus signal trigger data.	5-39
:TRIGger:EINTerval:EVENt<x>:CANBus:DATA:BORDer	Sets the byte order of the CAN bus signal trigger data or queries the current setting.	5-40
:TRIGger:EINTerval:EVENt<x>:CANBus:DATA:CONDition	Sets the data condition of the CAN bus signal trigger or queries the current setting.	5-40
:TRIGger:EINTerval:EVENt<x>:CANBus:DATA:DATA<x>	Sets the comparison data of the CAN bus signal trigger data or queries the current setting.	5-40
:TRIGger:EINTerval:EVENt<x>:CANBus:DATA:DLC	Sets the number of valid bytes (DLC) of the CAN bus signal trigger data or queries the current setting.	5-40
:TRIGger:EINTerval:EVENt<x>:CANBus:DATA:HEXA	Sets the CAN bus signal trigger data in hexadecimal notation.	5-41
:TRIGger:EINTerval:EVENt<x>:CANBus:DATA:MSBLsb	Sets the MSB and LSB bits of the CAN bus signal trigger data or queries the current setting.	5-41
:TRIGger:EINTerval:EVENt<x>:CANBus:DATA:PATTern	Sets the CAN bus signal trigger data in binary notation or queries the current setting.	5-41
:TRIGger:EINTerval:EVENt<x>:CANBus:DATA:SIGN	Sets the sign of the CAN bus signal trigger data or queries the current setting.	5-41
:TRIGger:EINTerval:EVENt<x>:CANBus:IDEXt?	Queries all settings related to the ID of the extended format of the CAN bus signal trigger.	5-41
:TRIGger:EINTerval:EVENt<x>:CANBus:IDEXt:HEXA	Sets the ID of the extended format of the CAN bus signal trigger in hexadecimal notation.	5-41
:TRIGger:EINTerval:EVENt<x>:CANBus:IDEXt:PATTern	Sets the ID of the extended format of the CAN bus signal trigger in binary notation or queries the current setting.	5-42
:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR?	Queries all settings related to the OR condition of the CAN bus signal trigger.	5-42
:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>?	Queries all settings related to each ID of the OR condition of the CAN bus signal trigger.	5-42
:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:ACK	Sets each ACK condition of the OR condition of the CAN bus signal trigger or queries the current setting.	5-42
:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:DATA?	Queries all settings related to each data of the OR condition of the CAN bus signal trigger.	5-43
:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:DATA:BORDer	Sets byte order of each data of the OR condition of the CAN bus signal trigger or queries the current setting.	5-43
:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:DATA:CONDition	Sets each data condition of the OR condition of the CAN bus signal trigger or queries the current setting.	5-43
:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:DATA:DATA<x>	Sets comparison data of each data of the OR condition of the CAN bus signal trigger or queries the current setting.	5-43
:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:DATA:DLC	Sets the number of valid bytes (DLC) of each data of the OR condition of the CAN bus signal trigger or queries the current setting.	5-43

Command	Function	Page
:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:DATA:HEXA	Sets each data of the OR condition of the CAN bus signal trigger in hexadecimal notation.	5-44
:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:DATA:MSBLsb	Sets the MSB and LSB bits of each data of the OR condition of the CAN bus signal trigger or queries the current setting.	5-44
:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:DATA:PATTern	Sets each data of the OR condition of the CAN bus signal trigger in binary notation or queries the current setting.	5-44
:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:DATA:SIGN	Sets sign of each data of the OR condition of the CAN bus signal trigger or queries the current setting.	5-44
:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:FORMat	Sets each message format (standard or extended) of the OR condition of the CAN bus signal trigger or queries the current setting.	5-44
:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:IDEXt?	Queries all settings related to the ID of each extended format of the OR condition of the CAN bus signal trigger.	5-44
:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:IDEXt:HEXA	Sets the ID of each extended format of the OR condition of the CAN bus signal trigger in hexadecimal notation.	5-45
:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:IDEXt:PATTern	Sets the ID of each extended format of the OR condition of the CAN bus signal trigger in binary notation or queries the current setting.	5-45
:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:IDSTd?	Queries all settings related to the ID of each standard format of the OR condition of the CAN bus signal trigger.	5-45
:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:IDSTd:HEXA	Sets the ID of each standard format of the OR condition of the CAN bus signal trigger in hexadecimal notation.	5-45
:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:IDSTd:PATTern	Sets the ID of each standard format of the OR condition of the CAN bus signal trigger in binary notation or queries the current setting.	5-45
:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:MODE	Enables or disables each condition of the OR condition of the CAN bus signal trigger or queries the current setting.	5-45
:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:RTR	Sets each RTR of the OR condition of the CAN bus signal trigger or queries the current setting.	5-45
:TRIGger:EINTerval:EVENt<x>:CANBus:IDSTd?	Queries all settings related to the ID of the standard format of the CAN bus signal trigger.	5-46
:TRIGger:EINTerval:EVENt<x>:CANBus:IDSTd:HEXA	Sets the ID of the standard format of the CAN bus signal trigger in hexadecimal notation.	5-46
:TRIGger:EINTerval:EVENt<x>:CANBus:IDSTd:PATTern	Sets the ID of the standard format of the CAN bus signal trigger in binary notation or queries the current setting.	5-46
:TRIGger:EINTerval:EVENt<x>:CANBus:MODE	Sets the CAN bus signal trigger mode or queries the current setting.	5-46
:TRIGger:EINTerval:EVENt<x>:CANBus:RECessive	Sets the recessive level (bus level) of the CAN bus signal trigger or queries the current setting.	5-46
:TRIGger:EINTerval:EVENt<x>:CANBus:RTR	Sets the RTR of the CAN bus signal trigger or queries the current setting.	5-46

Command	Function	Page
:TRIGger:EINTerval:EVENt<x>:CANBus:SOURce	Sets the trigger source of the CAN bus signal trigger or queries the current setting.	5-46
:TRIGger:EINTerval:EVENt<x>:CANBus:SPOint	Sets the sample point of the CAN bus signal trigger or queries the current setting.	5-47
:TRIGger:EINTerval:EVENt<x>:I2CBus?	Queries all settings related to the I^2C bus signal trigger of the event.	5-47
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa?	Queries all settings related to the address of the I^2C bus signal trigger.	5-47
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT10address?	Queries all settings related to the 10-bit address of the I^2C bus signal trigger.	5-47
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT10address:HEXA	Sets the 10-bit address of the I^2C bus signal trigger in hexadecimal notation.	5-47
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT10address:PATTern	Sets the 10-bit address of the I^2C bus signal trigger in binary notation or queries the current setting.	5-48
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7ADdress?	Queries all settings related to the 7-bit address of the I^2C bus signal trigger.	5-48
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7ADdress:HEXA	Sets the 7-bit address of the I^2C bus signal trigger in hexadecimal notation.	5-48
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7ADdress:PATTern	Sets the 7-bit address of the I^2C bus signal trigger in binary notation or queries the current setting.	5-48
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub?	Queries all settings related to the 7-bit + Sub address of the I^2C bus signal trigger.	5-48
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:ADDRess?	Queries all settings related to the7-bit address of the 7-bit + Sub address of the I^2C bus signal trigger.	5-48
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:ADDRess:HEXA	Sets the 7-bit address of the 7-bit + Sub address of the I^2C bus signal trigger in hexadecimal notation.	5-49
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:ADDRess:PATTern	Sets the 7-bit address of the 7-bit + Sub address of the I^2C bus signal trigger in binary notation or queries the current setting.	5-49
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:SADDress?	Queries all settings related to the Sub address of the 7-bit + Sub address of the I^2C bus signal trigger.	5-49
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:SADDress:HEXA	Sets the Sub address of the 7-bit + Sub address of the I^2C bus signal trigger in hexadecimal notation.	5-49
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:SADDress:PATTern	Sets the Sub address of the 7-bit + Sub address of the I^2C bus signal trigger in binary notation or queries the current setting.	5-49
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:TYPE	Sets the address type of the I^2C bus signal trigger or queries the current setting.	5-49
:TRIGger:EINTerval:EVENt<x>:I2CBus:CLOCk?	Queries all settings related to the clock channel of the I^2C bus signal trigger.	5-49
:TRIGger:EINTerval:EVENt<x>:I2CBus:CLOCk:SOURce	Sets the clock channel of the I^2C bus signal trigger or queries the current setting.	5-50
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA?	Queries all settings related to the data of the I^2C bus signal trigger.	5-50
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:BYTE	Sets the number of data bytes of the I^2C bus signal trigger or queries the current setting.	5-50

Command	Function	Page
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:CONDition	Sets the determination method (match or not matched) of the data of the I^2C bus signal trigger or queries the current setting.	5-50
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:DPOSition	Sets the position for comparing the data pattern of the I^2C bus signal trigger or queries the current setting.	5-50
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:HEXA<x>	Sets the data of the I^2C bus signal trigger in hexadecimal notation.	5-50
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:MODE	Enables/Disables the data conditions of the I^2C bus signal trigger or queries the current setting.	5-51
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:PATTern<x>	Sets the data of the I^2C bus signal trigger in binary notation or queries the current setting.	5-51
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:PMODe	Sets the pattern comparison start position mode of the data of the I^2C bus signal trigger or queries the current setting.	5-51
:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:SOURce	Sets the data channel of the I^2C bus signal trigger or queries the current setting.	5-51
:TRIGger:EINTerval:EVENt<x>:I2CBus:GCALl?	Queries all settings related to the general call of the I^2C bus signal trigger.	5-51
:TRIGger:EINTerval:EVENt<x>:I2CBus:GCALl:BIT7maddress?	Queries all settings related to the 7-bit master address of the general call of the I^2C bus signal trigger.	5-51
:TRIGger:EINTerval:EVENt<x>:I2CBus:GCALl:BIT7maddress:HEXA	Sets the 7-bit master address of the general call of the I^2C bus signal trigger in hexadecimal notation.	5-51
:TRIGger:EINTerval:EVENt<x>:I2CBus:GCALl:BIT7maddress:PATTern	Sets the 7-bit master address of the general call of the I^2C bus signal trigger in binary notation or queries the current setting.	5-52
:TRIGger:EINTerval:EVENt<x>:I2CBus:GCALl:SBYTe	Sets the second byte type of the general call of the I^2C bus signal trigger or queries the current setting.	5-52
:TRIGger:EINTerval:EVENt<x>:I2CBus:MODE	Sets the trigger mode of the I^2C bus signal trigger or queries the current setting.	5-52
:TRIGger:EINTerval:EVENt<x>:I2CBus:NAIGnore?	Queries all settings related to the NON ACK ignore mode of the I^2C bus signal trigger.	5-52
:TRIGger:EINTerval:EVENt<x>:I2CBus:NAIGnore:HSMode	Sets whether to ignore NON ACK in high speed mode of the I^2C bus trigger or queries the current setting.	5-52
:TRIGger:EINTerval:EVENt<x>:I2CBus:NAIGnore:RACCess	Sets whether to ignore NON ACK in read access mode of the I^2C bus trigger or queries the current setting.	5-52
:TRIGger:EINTerval:EVENt<x>:I2CBus:NAIGnore:SBYTe	Sets whether to ignore NON ACK in the start byte of the I^2C bus signal trigger or queries the current setting.	5-53
:TRIGger:EINTerval:EVENt<x>:I2CBus:SBHSmode?	Queries all settings related to the start byte and high speed mode of the I^2C bus signal trigger.	5-53
:TRIGger:EINTerval:EVENt<x>:I2CBus:SBHSmode:TYPE	Sets the type of the start byte or high speed mode of the I^2C bus signal trigger or queries the current setting.	5-53
:TRIGger:EINTerval:EVENt<x>:LINBus?	Queries all settings related to LIN bus signal triggers of each event.	5-53
:TRIGger:EINTerval:EVENt<x>:LINBus:BRATe	Sets the LIN bus signal trigger bitrate (data transfer rate) or queries the current setting.	5-53

Command	Function	Page
:TRIGger:EINTerval:EVENt<x>:LINBus:SOURce	Sets the LIN bus signal trigger source or queries the current setting.	5-53
:TRIGger:EINTerval:EVENt<x>:SPIBus?	Queries all settings related to the SPI bus signal trigger of the event.	5-53
:TRIGger:EINTerval:EVENt<x>:SPIBus:BITorder	Sets the bit order of the SPI bus signal trigger or queries the current setting.	5-54
:TRIGger:EINTerval:EVENt<x>:SPIBus:CLOCk?	Queries all settings related to the clock channel of the SPI bus signal trigger.	5-54
:TRIGger:EINTerval:EVENt<x>:SPIBus:CLOCk:POLarity	Sets the polarity of the clock channel of the SPI bus signal trigger or queries the current setting.	5-54
:TRIGger:EINTerval:EVENt<x>:SPIBus:CLOCk:SOURce	Sets the clock channel of the SPI bus signal trigger or queries the current setting.	5-54
:TRIGger:EINTerval:EVENt<x>:SPIBus:CS?	Queries all settings related to the chip select channel of the SPI bus signal trigger.	5-54
:TRIGger:EINTerval:EVENt<x>:SPIBus:CS:ACTive	Sets the active level of the chip select channel of the SPI bus signal trigger or queries the current setting.	5-54
:TRIGger:EINTerval:EVENt<x>:SPIBus:CS:SOURce	Sets the chip select channel of the SPI bus signal trigger or queries the current setting.	5-55
:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>?	Queries all settings related to the data of the SPI bus signal trigger.	5-55
:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:BYTE	Sets the number of bytes of the data of the SPI bus signal trigger or queries the current setting.	5-55
:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:CONDition	Sets the determination method (match or not match) of the data of the SPI bus signal trigger or queries the current setting.	5-55
:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:DPOSition	Sets the pattern comparison start position of the data of the SPI bus trigger or queries the current setting.	5-55
:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:HEXA<x>	Sets the data of the SPI bus signal trigger in hexadecimal notation.	5-55
:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:PATTern<x>	Sets the data of the SPI bus signal trigger in binary notation or queries the current setting.	5-56
:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:SOURce	Sets the trace of the data of the SPI bus signal trigger or queries the current setting.	5-56
:TRIGger:EINTerval:EVENt<x>:SPIBus:MODE	Sets the wiring system of the SPI bus signal trigger (three-wire or four-wire) orqueries the current setting.	5-56
:TRIGger:EINTerval:EVENt<x>:STATe:CHANnel<x>	Sets the condition to be satisfied of the channel or queries the current setting.	5-56
:TRIGger:EINTerval:EVENt<x>:TYPE	Sets the trigger type of the event or queries the current setting.	5-56
:TRIGger:ENHanced:CANBus?	Queries all settings related to the CAN bus signal trigger.	5-57
:TRIGger:ENHanced:CANBus:ACK	Sets the ACK condition of the CAN bus signal trigger or queries the current setting.	5-57
:TRIGger:ENHanced:CANBus:BRATe	Sets the bit rate (data transfer rate) of the CAN bus signal trigger or queries the current setting.	5-57
:TRIGger:ENHanced:CANBus:DATA?	Queries all settings related to the CAN bus signal trigger data.	5-57
:TRIGger:ENHanced:CANBus:DATA:BORDer	Sets the byte order of the CAN bus signal trigger data or queries the current setting.	5-57
:TRIGger:ENHanced:CANBus:DATA:CONDition	Sets the data condition of the CAN bus signal trigger or queries the current setting.	5-57

Command	Function	Page
:TRIGger:ENHanced:CANBus:DATA:DATA<x>	Sets the comparison data of the CAN bus signal trigger data or queries the current setting.	5-58
:TRIGger:ENHanced:CANBus:DATA:DLC	Sets the number of valid bytes (DLC) of the CAN bus signal trigger data or queries the current setting.	5-58
:TRIGger:ENHanced:CANBus:DATA:HEXA	Sets the CAN bus signal trigger data in hexadecimal notation.	5-58
:TRIGger:ENHanced:CANBus:DATA:MSBLsb	Sets the MSB and LSB bits of the CAN bus signal trigger data or queries the current setting.	5-58
:TRIGger:ENHanced:CANBus:DATA:PATTern	Sets the CAN bus signal trigger data in binary notation or queries the current setting.	5-58
:TRIGger:ENHanced:CANBus:DATA:SIGN	Sets the sign of the CAN bus signal trigger data or queries the current setting.	5-58
:TRIGger:ENHanced:CANBus:IDEXt?	Queries all settings related to the ID of the extended format of the CAN bus signal trigger.	5-58
:TRIGger:ENHanced:CANBus:IDEXt:HEXA	Sets the ID of the extended format of the CAN bus signal trigger in hexadecimal notation.	5-58
:TRIGger:ENHanced:CANBus:IDEXt:PATTern	Sets the ID of the extended format of the CAN bus signal trigger in binary notation or queries the current setting.	5-59
:TRIGger:ENHanced:CANBus:IDOR?	Queries all settings related to the OR condition of the CAN bus signal trigger.	5-59
:TRIGger:ENHanced:CANBus:IDOR:ID<x>?	Queries all settings related to each ID of the OR condition of the CAN bus signal trigger.	5-59
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:ACK	Sets each ACK condition of the OR condition of the CAN bus signal trigger or queries the current setting.	5-59
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA?	Queries all settings related to each data of the OR condition of the CAN bus signal trigger.	5-59
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA:BORDer	Sets byte order of each data of the OR condition of the CAN bus signal trigger or queries the current setting.	5-60
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA:CONDition	Sets each data condition of the OR condition of the CAN bus signal trigger or queries the current setting.	5-60
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA:DATA<x>	Sets comparison data of each data of the OR condition of the CAN bus signal trigger or queries the current setting.	5-60
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA:DLC	Sets the number of valid bytes (DLC) of each data of the OR condition of the CAN bus signal trigger or queries the current setting.	5-60
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA:HEXA	Sets each data of the OR condition of the CAN bus signal trigger in hexadecimal notation.	5-61
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA:MSBLsb	Sets the MSB and LSB bits of each data of the OR condition of the CAN bus signal trigger or queries the current setting.	5-61
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA:PATTern	Sets each data of the OR condition of the CAN bus signal trigger in binary notation or queries the current setting.	5-61
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA:SIGN	Sets sign of each data of the OR condition of the CAN bus signal trigger or queries the current setting.	5-61

Command	Function	Page
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:FORMat	Sets each message format (standard or extended) of the OR condition of the CAN bus signal trigger or queries the current setting.	5-61
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDEXt?	Queries all settings related to the ID of each extended format of the OR condition of the CAN bus signal trigger.	5-61
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDEXt:HEXA	Sets the ID of each extended format of the OR condition of the CAN bus signal trigger in hexadecimal notation.	5-62
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDEXt:PATTern	Sets the ID of each extended format of the OR condition of the CAN bus signal trigger in binary notation or queries the current setting.	5-62
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDSTd?	Queries all settings related to the ID of each standard format of the OR condition of the CAN bus signal trigger.	5-62
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDSTd:HEXA	Sets the ID of each standard format of the OR condition of the CAN bus signal trigger in hexadecimal notation.	5-62
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDSTd:PATTern	Sets the ID of each standard format of the OR condition of the CAN bus signal trigger in binary notation or queries the current setting.	5-62
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:MODE	Enables or disables each condition of the OR condition of the CAN bus signal trigger or queries the current setting.	5-62
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:RTR	Sets each RTR of the OR condition of the CAN bus signal trigger or queries the current setting.	5-62
:TRIGger:ENHanced:CANBus:IDSTd?	Queries all settings related to the ID of the standard format of the CAN bus signal trigger.	5-63
:TRIGger:ENHanced:CANBus:IDSTd:HEXA	Sets the ID of the standard format of the CAN bus signal trigger in hexadecimal notation.	5-63
:TRIGger:ENHanced:CANBus:IDSTd:PATTern	Sets the ID of the standard format of the CAN bus signal trigger in binary notation or queries the current setting.	5-63
:TRIGger:ENHanced:CANBus:MODE	Sets the CAN bus signal trigger mode or queries the current setting.	5-63
:TRIGger:ENHanced:CANBus:RECessive	Sets the recessive level (bus level) of the CAN bus signal trigger or queries the current setting.	5-63
:TRIGger:ENHanced:CANBus:RTR	Sets the RTR of the CAN bus signal trigger or queries the current setting.	5-63
:TRIGger:ENHanced:CANBus:SOURce	Sets the trigger source of the CAN bus signal trigger or queries the current setting.	5-63
:TRIGger:ENHanced:CANBus:SPOint	Sets the sample point of the CAN bus signal trigger or queries the current setting.	5-63
:TRIGger:ENHanced:I2CBus?	Queries all settings related to the I^2C bus signal trigger.	5-64
:TRIGger:ENHanced:I2CBus:ADATa?	Queries all settings related to the address of the I^2C bus signal trigger.	5-64
:TRIGger:ENHanced:I2CBus:ADATa:BIT10address?	Queries all settings related to the 10-bit address of the I^2C bus signal trigger.	5-64
:TRIGger:ENHanced:I2CBus:ADATa:BIT10address:HEXA	Sets the 10-bit address of the I^2C bus signal trigger in hexadecimal notation.	5-64
:TRIGger:ENHanced:I2CBus:ADATa:BIT10address:PATTern	Sets the 10-bit address of the I^2C bus signal trigger in binary notation or queries the current setting.	5-64
:TRIGger:ENHanced:I2CBus:ADATa:BIT7ADdress?	Queries all settings related to the 7-bit address of the I^2C bus signal trigger.	5-64
:TRIGger:ENHanced:I2CBus:ADATa:BIT7ADdress:HEXA	Sets the 7-bit address of the I^2C bus signal trigger in hexadecimal notation.	5-64

Command	Function	Page
:TRIGger:ENHanced:I2CBus:ADATa:BIT7ADdress:PATTern	Sets the 7-bit address of the I^2C bus signal trigger in binary notation or queries the current setting.	5-65
:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub?	Queries all settings related to the 7-bit + Sub address of the I^2C bus signal trigger.	5-65
:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub:ADDRess?	Queries all settings related to the7-bit address of the 7-bit + Sub address of the I^2C bus signal trigger.	5-65
:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub:ADDRess:HEXA	Sets the 7-bit address of the 7-bit + Sub address of the I^2C bus signal trigger in hexadecimal notation.	5-65
:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub:ADDRess:PATTern	Sets the 7-bit address of the 7-bit + Sub address of the I^2C bus signal trigger in binary notation or queries the current setting.	5-65
:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub:SADDress?	Queries all settings related to the Sub address of the 7-bit + Sub address of the I^2C bus signal trigger.	5-65
:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub:SADDress:HEXA	Sets the Sub address of the 7-bit + Sub address of the I^2C bus signal trigger in hexadecimal notation.	5-65
:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub:SADDress:PATTern	Sets the Sub address of the 7-bit + Sub address of the I^2C bus signal trigger in binary notation or queries the current setting.	5-65
:TRIGger:ENHanced:I2CBus:ADATa:TYPE	Sets the address type of the I^2C bus signal trigger or queries the current setting.	5-66
:TRIGger:ENHanced:I2CBus:CLOCk?	Queries all settings related to the clock channel of the I^2C bus signal trigger.	5-66
:TRIGger:ENHanced:I2CBus:CLOCk:SOURce	Sets the clock channel of the I^2C bus signal trigger or queries the current setting.	5-66
:TRIGger:ENHanced:I2CBus:DATA?	Queries all settings related to the data of the I^2C bus signal trigger.	5-66
:TRIGger:ENHanced:I2CBus:DATA:BYTE	Sets the number of data bytes of the I^2C bus signal trigger or queries the current setting.	5-66
:TRIGger:ENHanced:I2CBus:DATA:CONDition	Sets the determination method (match or not matched) of the data of the I^2C bus signal trigger or queries the current setting.	5-66
:TRIGger:ENHanced:I2CBus:DATA:DPOSition	Sets the position for comparing the data pattern of the I^2C bus signal trigger or queries the current setting.	5-66
:TRIGger:ENHanced:I2CBus:DATA:HEXA<x>	Sets the data of the I^2C bus signal trigger in hexadecimal notation.	5-66
:TRIGger:ENHanced:I2CBus:DATA:MODE	Enables/Disables the data conditions of the I^2C bus signal trigger or queries the current setting.	5-67
:TRIGger:ENHanced:I2CBus:DATA:PATTern<x>	Sets the data of the I^2C bus signal trigger in binary notation or queries the current setting.	5-67
:TRIGger:ENHanced:I2CBus:DATA:PMODe	Sets the pattern comparison start position mode of the data of the I^2C bus signal trigger or queries the current setting.	5-67
:TRIGger:ENHanced:I2CBus:DATA:SOURce	Sets the data channel of the I^2C bus signal trigger or queries the current setting.	5-67
:TRIGger:ENHanced:I2CBus:GCALl?	Queries all settings related to the general call of the I^2C bus signal trigger.	5-67
:TRIGger:ENHanced:I2CBus:GCALl:BIT7maddress?	Queries all settings related to the 7-bit master address of the general call of the I^2C bus signal trigger.	5-67
:TRIGger:ENHanced:I2CBus:GCALl:BIT7maddress:HEXA	Sets the 7-bit master address of the general call of the I^2C bus trigger in hexadecimal notation.	5-67
:TRIGger:ENHanced:I2CBus:GCALl:BIT7maddress:PATTern	Sets the 7-bit master address of the general call of the I^2C bus signal trigger in binary notation or queries the current setting.	5-67

Command	Function	Page
:TRIGger:ENHanced:I2CBus:GCALl:SBYTe	Sets the second byte type of the general call of the I^{2}C bus signal trigger or queries the current setting.	5-68
:TRIGger:ENHanced:I2CBus:MODE	Sets the trigger mode of the I^{2}C bus signal trigger or queries the current setting.	5-68
:TRIGger:ENHanced:I2CBus:NAIGnore?	Queries all settings related to the NON ACK ignore mode of the I^{2}C bus signal trigger.	5-68
:TRIGger:ENHanced:I2CBus:NAIGnore:HSMode	Sets whether to ignore NON ACK in high speed mode of the I^{2}C bus signal trigger or queries the current setting.	5-68
:TRIGger:ENHanced:I2CBus:NAIGnore:RACCess	Sets whether to ignore NON ACK in read access mode of the I^{2}C bus signal trigger or queries the current setting.	5-68
:TRIGger:ENHanced:I2CBus:NAIGnore:SBYTe	Sets whether to ignore NON ACK in the start byte of the I^{2}C bus signal trigger or queries the current setting.	5-68
:TRIGger:ENHanced:I2CBus:SBHSmode?	Queries all settings related to the start byte and high speed mode of the I^{2}C bus signal trigger.	5-68
:TRIGger:ENHanced:I2CBus:SBHSmode:TYPE	Sets the type of the start byte or high speed mode of the I^{2}C bus signal trigger or queries the current setting.	5-68
:TRIGger:ENHanced:LINBus?	Queries all settings related to the LIN bus trigger or queries the current setting.	5-69
:TRIGger:ENHanced:LINBus:BRATe	Sets the LIN bus signal trigger bitrate (data transfer rate) or queries the current setting.	5-69
:TRIGger:ENHanced:LINBus:SOURce	Sets the LIN bus signal trigger source or queries the current setting.	5-69
:TRIGger:ENHanced:SPIBus?	Queries all settings related to the SPI bus signal trigger.	5-69
:TRIGger:ENHanced:SPIBus:BITorder	Sets the bit order of the SPI bus signal trigger or queries the current setting.	5-69
:TRIGger:ENHanced:SPIBus:CLOCk?	Queries all settings related to the clock channel of the SPI bus signal trigger.	5-69
:TRIGger:ENHanced:SPIBus:CLOCk:POLarity	Sets the polarity of the clock channel of the SPI bus signal trigger or queries the current setting.	5-69
:TRIGger:ENHanced:SPIBus:CLOCk:SOURce	Sets the clock channel of the SPI bus signal trigger or queries the current setting.	5-69
:TRIGger:ENHanced:SPIBus:CS?	Queries all settings related to the chip select channel of the SPI bus signal trigger.	5-69
:TRIGger:ENHanced:SPIBus:CS:ACTive	Sets the active level of the chip select channel of the SPI bus signal trigger or queries the current setting.	5-70
:TRIGger:ENHanced:SPIBus:CS:SOURce	Sets the chip select channel of the SPI bus signal trigger or queries the current setting.	5-70
:TRIGger:ENHanced:SPIBus:DATA<x>?	Queries all settings related to the data of the SPI bus signal trigger.	5-70
:TRIGger:ENHanced:SPIBus:DATA<x>:BYTE	Sets the number of bytes of the data of the SPI bus signal trigger or queries the current setting.	5-70
:TRIGger:ENHanced:SPIBus:DATA<x>:CONDition	Sets the determination method (match or not match) of the data of the SPI bus signal trigger or queries the current setting.	5-70
:TRIGger:ENHanced:SPIBus:DATA<x>:DPOSition	Sets the pattern comparison start position of the data of the SPI bus signal trigger or queries the current setting.	5-70
:TRIGger:ENHanced:SPIBus:DATA<x>:HEXA<x>	Sets the data of the SPI bus signal trigger in hexadecimal notation.	5-70
:TRIGger:ENHanced:SPIBus:DATA<x>:PATTern<x>	Sets the data of the SPI bus signal trigger in binary notation or queries the current setting.	5-71
:TRIGger:ENHanced:SPIBus:DATA<x>:SOURce	Sets the source channel of the data of the SPI bus signal trigger or queries the current setting.	5-71

Command	Function	Page
:TRIGger:ENHanced:SPIBus::MODE	Sets the wiring system of the SPI bus signal trigger (three-wire or four-wire) or queries the current setting.	5-71
:TRIGger:SOURce:CHANnel<x>:LEVel	Sets the trigger level of the channel or queries the current setting.	5-71
:TRIGger:SOURce:CHANnel<x>:STATe	Sets the condition to be satisfied of the channel or queries the current setting.	5-71
:TRIGger:TYPE	Sets the trigger type or queries the current setting.	5-71

5.2 ANALysis Group

:ANALysis:SBUS<x>?

Function Queries all settings related to the serial bus signal analysis function.

Syntax :ANALysis:SBUS<x>?
<x> = 1 or 2

Example :ANALYSIS:SBUS1? -> :ANALYSIS:SBUS1:
ANALYZE:CANBUS:BRATE 1000000;
RECESSIVE HIGH;SPOINT 62.5E+00;
TRACE 1;:ANALYSIS:SBUS1:ANALYZE:
DECODE 1;I2CBUS:CLOCK 1;DTRACE 1;:
ANALYSIS:SBUS1:ANALYZE:LINBUS:
BRATE 19200;REVISION LIN1_3;TRACE 1;:
ANALYSIS:SBUS1:ANALYZE:LIST:DISPLAY 1;
MODE DETAIL;SCROLL HORIZONTAL;:
ANALYSIS:SBUS1:ANALYZE:MODE CANBUS;
MPOSITION 0.00000E+00;RPOINT TRIGGER;
SPIBUS:CLOCK:POLARITY FALL;SOURCE 1;:
ANALYSIS:SBUS1:ANALYZE:SPIBUS:CS:
ACTIVE HIGH;TRACE 1;:ANALYSIS:SBUS1:
ANALYZE:SPIBUS:DATA1:ACTIVE HIGH;
TRACE 1;:ANALYSIS:SBUS1:ANALYZE:SPIBUS:
DATA2:ACTIVE HIGH;TRACE 1;:ANALYSIS:
SBUS1:ANALYZE:SPIBUS:SETUP:
BITORDER LSBFIRST;MODE WIRE3;:ANALYSIS:
SBUS1:ANALYZE:TRACE1:
HYSTERESIS 1.000E+00;LEVEL 1.000E+00;:
ANALYSIS:SBUS1:ANALYZE:TRACE2:
HYSTERESIS 1.000E+00;LEVEL 1.000E+00;:
ANALYSIS:SBUS1:ANALYZE:TRACE3:
HYSTERESIS 1.000E+00;LEVEL 1.000E+00;:
ANALYSIS:SBUS1:ANALYZE:TRACE4:
HYSTERESIS 1.000E+00;LEVEL 1.000E+00;:
ANALYSIS:SBUS1:ANALYZE:TRACE5:
HYSTERESIS 1.000E+00;LEVEL 1.000E+00;:
ANALYSIS:SBUS1:ANALYZE:TRACE6:
HYSTERESIS 1.000E+00;LEVEL 1.000E+00;:
ANALYSIS:SBUS1:ANALYZE:TRACE7:
HYSTERESIS 1.000E+00;LEVEL 1.000E+00;:
ANALYSIS:SBUS1:ANALYZE:TRACE8:
HYSTERESIS 1.000E+00;LEVEL 1.000E+00;:
ANALYSIS:SBUS1:ZLINKAGE OFF

:ANALysis:SBUS<x>:ANALyze?

Function Queries all settings related to the serial bus signal analysis.

Syntax :ANALysis:SBUS<x>:ANALyze?
<x> = 1 or 2

Example :ANALYSIS:SBUS1:ANALYZE? -> :ANALYSIS:
SBUS1:ANALYZE:CANBUS:BRATE 1000000;
RECESSIVE HIGH;SPOINT 62.5E+00;
TRACE 1;:ANALYSIS:SBUS1:ANALYZE:
DECODE 1;I2CBUS:CLOCK 1;DTRACE 1;:
ANALYSIS:SBUS1:ANALYZE:LINBUS:
BRATE 19200;REVISION LIN1_3;TRACE 1;:
ANALYSIS:SBUS1:ANALYZE:LIST:DISPLAY 1;
MODE DETAIL;SCROLL HORIZONTAL;:
ANALYSIS:SBUS1:ANALYZE:MODE CANBUS;
MPOSITION 0.00000E+00;RPOINT TRIGGER;
SPIBUS:CLOCK:POLARITY FALL;SOURCE 1;:
ANALYSIS:SBUS1:ANALYZE:SPIBUS:CS:
ACTIVE HIGH;TRACE 1;:ANALYSIS:SBUS1:
ANALYZE:SPIBUS:DATA1:ACTIVE HIGH;
TRACE 1;:ANALYSIS:SBUS1:ANALYZE:SPIBUS:
DATA2:ACTIVE HIGH;TRACE 1;:ANALYSIS:
SBUS1:ANALYZE:SPIBUS:SETUP:
BITORDER LSBFIRST;MODE WIRE3;:ANALYSIS:
SBUS1:ANALYZE:TRACE1:
HYSTERESIS 1.000E+00;LEVEL 1.000E+00;:
ANALYSIS:SBUS1:ANALYZE:TRACE2:
HYSTERESIS 1.000E+00;LEVEL 1.000E+00;:
ANALYSIS:SBUS1:ANALYZE:TRACE3:
HYSTERESIS 1.000E+00;LEVEL 1.000E+00;:
ANALYSIS:SBUS1:ANALYZE:TRACE4:
HYSTERESIS 1.000E+00;LEVEL 1.000E+00;:
ANALYSIS:SBUS1:ANALYZE:TRACE5:
HYSTERESIS 1.000E+00;LEVEL 1.000E+00;:
ANALYSIS:SBUS1:ANALYZE:TRACE6:
HYSTERESIS 1.000E+00;LEVEL 1.000E+00;:
ANALYSIS:SBUS1:ANALYZE:TRACE7:
HYSTERESIS 1.000E+00;LEVEL 1.000E+00;:
ANALYSIS:SBUS1:ANALYZE:TRACE8:
HYSTERESIS 1.000E+00;LEVEL 1.000E+00

:ANALysis:SBUS<x>[:ANALyze]:CANBus?

Function Queries all settings related to the CAN bus signal analysis.

Syntax :ANALysis:SBUS<x>[:ANALyze]:CANBus?
<x> = 1, 2

Example :ANALYSIS:SBUS1:ANALYZE:CANBUS? ->
:ANALYSIS:SBUS1:ANALYZE:CANBUS:
BRATE 1000000;RECESSIVE HIGH;
SPOINT 62.5E+00;TRACE 1

:ANALysis:SBUS<x>[:ANALyze]:CANBus:BRATe

Function Sets the bit rate (data transfer rate) of the CAN bus signal analysis or queries the current setting.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:CANBus:
BRATe {<NRf>|USER,<NRf>}
:ANALysis:SBUS<x>[:ANALyze]:CANBus:
BRATe?
```
<x> = 1, 2
<NRf> = 33300, 83300, 125000, 250000, 500000, 1000000

signal <NRf> of USER = See the *User's Manual (IM701310-01E)*.

Example
```
:ANALYSIS:SBUS1:ANALYZE:CANBUS:
BRATE 83300
:ANALYSIS:SBUS1:ANALYZE:CANBUS:
BRATE? ->
:ANALYSIS:SBUS1:ANALYZE:CANBUS:
BRATE 83300
```

:ANALysis:SBUS<x>[:ANALyze]:CANBus:FJUMp:ACK

Function Executes a field jump to the ACK Field in the results of the CAN bus signal analysis.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:CANBus:
FJUMp:ACK
```
<x> = 1, 2

Example
```
:ANALYSIS:SBUS1:ANALYZE:CANBUS:FJUMP:
ACK
```

:ANALysis:SBUS<x>[:ANALyze]:CANBus:FJUMp:CONTrol

Function Executes a field jump to the Control Field in the results of the CAN bus signal analysis.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:CANBus:
FJUMp:CONTrol
```
<x> = 1, 2

Example
```
:ANALYSIS:SBUS1:ANALYZE:CANBUS:FJUMP:
CONTROL
```

:ANALysis:SBUS<x>[:ANALyze]:CANBus:FJUMp:CRC

Function Executes a field jump to the CRC Field in the results of the CAN bus signal analysis.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:CANBus:
FJUMp:CRC
```
<x> = 1, 2

Example
```
:ANALYSIS:SBUS1:ANALYZE:CANBUS:FJUMP:
CRC
```

:ANALysis:SBUS<x>[:ANALyze]:CANBus:FJUMp:DATA

Function Executes a field jump to the Data Field in the results of the CAN bus signal analysis.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:CANBus:
FJUMp:DATA
```
<x> = 1, 2

Example
```
:ANALYSIS:SBUS1:ANALYZE:CANBUS:FJUMP:
DATA
```

:ANALysis:SBUS<x>[:ANALyze]:CANBus:FJUMp:IDENtifier

Function Executes a field jump to the Identifier Field in the results of the CAN bus signal analysis.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:CANBus:
FJUMp:IDENtifier
```
<x> = 1, 2

Example
```
:ANALYSIS:SBUS1:ANALYZE:CANBUS:FJUMP:
IDENTIFIER
```

:ANALysis:SBUS<x>[:ANALyze]:CANBus:FJUMp:SOF

Function Executes a field jump to the SOF Field in the results of the CAN bus signal analysis.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:CANBus:
FJUMp:SOF
```
<x> = 1, 2

Example
```
:ANALYSIS:SBUS1:ANALYZE:CANBUS:FJUMP:
SOF
```

:ANALysis:SBUS<x>[:ANALyze]:CANBus:RECessive

Function Sets the recessive level (bus level) of the CAN bus signal analysis or queries the current setting.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:CANBus:
RECessive {HIGH|LOW}
:ANALysis:SBUS<x>[:ANALyze]:CANBus:
RECessive?
```
<x> = 1, 2

Example
```
:ANALYSIS:SBUS1:ANALYZE:CANBUS:
RECESSIVE HIGH
:ANALYSIS:SBUS1:ANALYZE:CANBUS:
RECESSIVE? ->
:ANALYSIS:SBUS1:ANALYZE:CANBUS:
RECESSIVE HIGH
```

:ANALysis:SBUS<x>[:ANALyze]:CANBus:SPOint

Function Sets the sample point of the CAN bus signal analysis or queries the current setting.

Syntax :ANALysis:SBUS<x>[:ANALyze]:CANBus:SPOint {<NRf>}
:ANALysis:SBUS<x>[:ANALyze]:CANBus:SPOint?
<x> = 1, 2
<NRf> = 18.8 to 90.6(%)

Example :ANALYSIS:SBUS1:ANALYZE:CANBUS:SPOINT 18.8
:ANALYSIS:SBUS1:ANALYZE:CANBUS:SPOINT? ->
:ANALYSIS:SBUS1:ANALYZE:CANBUS:SPOINT 18.8E+00

:ANALysis:SBUS<x>[:ANALyze]:CANBus:TRACe

Function Sets the trace of the CAN bus signal analysis or queries the current setting.

Syntax :ANALysis:SBUS<x>[:ANALyze]:CANBus:TRACe {<NRf>}
:ANALysis:SBUS<x>[:ANALyze]:CANBus:TRACe?
<x> = 1, 2
<NRf> = 1 to 8

Example :ANALYSIS:SBUS1:ANALYZE:CANBUS:TRACE 1
:ANALYSIS:SBUS1:ANALYZE:CANBUS:TRACE? ->
:ANALYSIS:SBUS1:ANALYZE:CANBUS:TRACE 1

:ANALysis:SBUS<x>[:ANALyze]:DECode

Function Turns the serial bus signal analysis decoding display ON/OFF or queries the current status.

Syntax ANALysis:SBUS<x>[:ANALyze]:DECode {<Boolean>}
:ANALysis:SBUS<x>[:ANALyze]:DECode?
<x>=1, 2

Example ANALYSIS:SBUS1:ANALYZE:DECODE ON
:ANALYSIS:SBUS1:ANALYZE:DECODE? ->
:ANALYSIS:SBUS1:ANALYZE:DECODE 1

:ANALysis:SBUS<x>[:ANALyze]:I2CBus?

Function Queries all settings related to the I^2C bus signal analysis.

Syntax :ANALysis:SBUS<x>[:ANALyze]:I2CBus?
<x> = 1 or 2

Example :ANALYSIS:SBUS1:ANALYZE:I2CBUS? ->
:ANALYSIS:SBUS1:ANALYZE:I2CBUS:CLOCK 1;DTRACE 1

:ANALysis:SBUS<x>[:ANALyze]:I2CBus:CLOCk

Function Sets the clock channel of the I^2C bus signal analysis or queries the current setting.

Syntax :ANALysis:SBUS<x>[:ANALyze]:I2CBus:CLOCk {<NRf>}
:ANALysis:SBUS<x>[:ANALyze]:I2CBus:CLOCk?
<x> = 1 or 2
<NRf> = 1 to 8

Example :ANALYSIS:SBUS1:ANALYZE:I2CBUS:CLOCK 1
:ANALYSIS:SBUS1:ANALYZE:I2CBUS:CLOCK? ->
:ANALYSIS:SBUS1:ANALYZE:I2CBUS:CLOCK 1

:ANALysis:SBUS<x>[:ANALyze]:I2CBus:DTRace

Function Sets the data channel of the I^2C bus signal analysis or queries the current setting.

Syntax :ANALysis:SBUS<x>[:ANALyze]:I2CBus:DTRace {<NRf>}
:ANALysis:SBUS<x>[:ANALyze]:I2CBus:DTRace?
<x> = 1 or 2
<NRf> = 1 to 8

Example :ANALYSIS:SBUS1:ANALYZE:I2CBUS:DTRACE 1
:ANALYSIS:SBUS1:ANALYZE:I2CBUS:DTRACE? ->
:ANALYSIS:SBUS1:ANALYZE:I2CBUS:DTRACE 1

:ANALysis:SBUS<x>[:ANALyze]:LINBus?

Function Queries all settings related to the LIN bus signal analysis.

Syntax :ANALysis:SBUS<x>[:ANALyze]:LINBus?
<x>=1, 2

Example :ANALYSIS:SBUS1:ANALYZE:LINBUS? ->
:ANALYSIS:SBUS1:ANALYZE:LINBUS:BRATE 19200;REVISION LIN1_3;TRACE 1

:ANALysis:SBUS<x>[:ANALyze]:LINBus:BRATe

Function Sets the LIN bus signal analysis bitrate (data transfer rate) or queries the current setting.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:LINBus:
BRATe {<NRf>|USER,<NRf>}
:ANALysis:SBUS<x>[:ANALyze]:LINBus:
BRATe?
```
<x>=1, 2
<NRf>=1200, 2400, 4800, 9600, 19200
USER <NRf>=See this *User's Manual*.

Example
```
:ANALYSIS:SBUS1:ANALYZE:LINBUS:
BRATE 19200
:ANALYSIS:SBUS1:ANALYZE:LINBUS:BRATE?
-> :ANALYSIS:SBUS1:ANALYZE:LINBUS:
BRATE 19200
```

:ANALysis:SBUS<x>[:ANALyze]:LINBus:REVision

Function Sets the LIN bus signal analysis revision (1.3 or 2.0) or queries the current setting.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:LINBus:
REVision {LIN1_3|LIN2_0}
:ANALysis:SBUS<x>[:ANALyze]:LINBus:
REVision?
```
<x>=1, 2

Example
```
:ANALYSIS:SBUS1:ANALYZE:LINBUS:
REVISION LIN1_3
:ANALYSIS:SBUS1:ANALYZE:LINBUS:
REVISION? -> :ANALYSIS:SBUS1:ANALYZE:
LINBUS:REVISION LIN1_3
```

:ANALysis:SBUS<x>[:ANALyze]:LINBus:TRACe

Function Sets the LIN bus signal analysis trace or queries the current setting.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:LINBus:
TRACe {<NRf>}
:ANALysis:SBUS<x>[:ANALyze]:LINBus:
TRACe?
```
<x>=1, 2
<NRf>=1–8

Example
```
:ANALYSIS:SBUS1:ANALYZE:LINBUS:TRACE 1
:ANALYSIS:SBUS1:ANALYZE:LINBUS:TRACE?
-> :ANALYSIS:SBUS1:ANALYZE:LINBUS:
TRACE 1
```

:ANALysis:SBUS<x>[:ANALyze]:LIST?

Function Queries all settings related to the list display of the serial bus signal analysis.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:LIST?
```
<x> = 1 or 2

Example
```
:ANALYSIS:SBUS1:ANALYZE:LIST? ->
:ANALYSIS:SBUS1:ANALYZE:LIST:DISPLAY 1;
MODE DETAIL;SCROLL HORIZONTAL
```

:ANALysis:SBUS<x>[:ANALyze]:LIST:DISPlay

Function Turns the serial bus signal analysis list display ON/OFF or queries the current status.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:LIST:
DISPlay {<Boolean>}
:ANALysis:SBUS<x>[:ANALyze]:LIST:
DISPlay?
```
<x>=1, 2

Example
```
:ANALYSIS:SBUS1:ANALYZE:LIST:DISPLAY ON
:ANALYSIS:SBUS1:ANALYZE:LIST:DISPLAY?
-> :ANALYSIS:SBUS1:ANALYZE:LIST:
DISPLAY 1
```

:ANALysis:SBUS<x>[:ANALyze]:LIST:ITEM?

Function Queries the item in the list display of the serial bus signal analysis.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:LIST:ITEM?
```
<x> = 1 or 2

Example
```
:ANALYSIS:SBUS1:ANALYZE:LIST:ITEM? ->
:ANALYSIS:SBUS1:ANALYZE:LIST:ITEM
"No.,S/P,Hex,Form,R/W,ACK,"
```

:ANALysis:SBUS<x>[:ANALyze]:LIST:MODE

Function Sets the mode of the list display of the serial bus signal analysis or queries the current setting.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:LIST:
MODE {DETail|SIMPle}
:ANALysis:SBUS<x>[:ANALyze]:LIST:MODE?
```
<x> = 1 or 2

Example
```
:ANALYSIS:SBUS1:ANALYZE:LIST:
MODE DETAIL
:ANALYSIS:SBUS1:ANALYZE:LIST:MODE? ->
:ANALYSIS:SBUS1:ANALYZE:LIST:
MODE DETAIL
```

:ANALysis:SBUS<x>[:ANALyze]:LIST:SCRoll

Function Sets the scroll method of the list display of the serial bus signal analysis or queries the current setting.

Syntax :ANALysis:SBUS<x>[:ANALyze]:LIST:SCRoll {HORizontal|VERTical}
:ANALysis:SBUS<x>[:ANALyze]:LIST:SCRoll?
<x> = 1 or 2

Example :ANALYSIS:SBUS1:ANALYZE:LIST:SCROLL HORIZONTAL
:ANALYSIS:SBUS1:ANALYZE:LIST:SCROLL? ->
:ANALYSIS:SBUS1:ANALYZE:LIST:SCROLL HORIZONTAL

ANALysis:SBUS<x>[:ANALyze]:LIST:VALue?

Function Queries the automated measured value of the specified analysis number in the analysis result list of the serial bus signal analysis.

Syntax :ANALysis:SBUS<x>[:ANALyze]:LIST:VALue? {<NRf>|MAXimum|MINimum}
<x> = 1, 2
<NRf> = –40000 to 40000
(<NRf> = –2999 to 2999 for :ANALysis:SBUS<x>[:ANALyze]:MODE CANBus)

Example :ANALYSIS:SBUS1:ANALYZE:LIST:VALUE? 1 ->
:ANALYSIS:SBUS1:ANALYZE:LIST:VALUE "1, P, 00, A, , 0,"

Description Set the data to MAXimum or MINimum to specify the maximum list display number or the minimum list display number.

:ANALysis:SBUS<x>[:ANALyze]:MODE

Function Sets the serial bus signal analysis mode or queries the current setting.

Syntax :ANALysis:SBUS<x>[:ANALyze]:MODE {CANBus|I2CBus|LIN|SPIBus}
:ANALysis:SBUS<x>[:ANALyze]:MODE?
<x> = 1 or 2

Example :ANALYSIS:SBUS1:ANALYZE:MODE I2CBUS
:ANALYSIS:SBUS1:ANALYZE:MODE? ->
:ANALYSIS:SBUS1:ANALYZE:MODE I2CBUS

:ANALysis:SBUS<x>[:ANALyze]:RPOint

Function Sets the analysis reference point of the serial bus signal analysis or queries the current setting.

Syntax :ANALysis:SBUS<x>[:ANALyze]:RPOint {MANual,<NRf>|TRIGger}
:ANALysis:SBUS<x>[:ANALyze]:RPOint?
<x> = 1, 2
<NRf> = –5 to 5 (div)

Example :ANALYSIS:SBUS1:ANALYZE:RPOINT MANAUL,1
:ANALYSIS:SBUS1:ANALYZE:RPOINT? ->
:ANALYSIS:SBUS1:ANALYZE:RPOINT MANUAL,1.00000E+00

:ANALysis:SBUS<x>[:ANALyze]:SPIBus?

Function Queries all settings related to the SPI bus signal analysis.

Syntax :ANALysis:SBUS<x>[:ANALyze]:SPIBus?
<x> = 1 or 2

Example :ANALYSIS:SBUS1:ANALYZE:SPIBUS? ->
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:CLOCK:POLARITY FALL;SOURCE 1;:ANALYSIS:SBUS1:ANALYZE:SPIBUS:CS:ACTIVE HIGH;TRACE 1;:ANALYSIS:SBUS1:ANALYZE:SPIBUS:DATA1:ACTIVE HIGH;TRACE 1;:ANALYSIS:SBUS1:ANALYZE:SPIBUS:DATA2:ACTIVE HIGH;TRACE 1;:ANALYSIS:SBUS1:ANALYZE:SPIBUS:SETUP:BITORDER LSBFIRST;MODE WIRE3

:ANALysis:SBUS<x>[:ANALyze]:SPIBus:CLOCk?

Function Queries all settings related to the clock channel of the SPI bus signal analysis.

Syntax :ANALysis:SBUS<x>[:ANALyze]:SPIBus:CLOCk?
<x> = 1 or 2

Example :ANALYSIS:SBUS1:ANALYZE:SPIBUS:CLOCK? -> :ANALYSIS:SBUS1:ANALYZE:SPIBUS:CLOCK:POLARITY FALL;SOURCE 1

:ANALysis:SBUS<x>[:ANALyze]:SPIBus:CLOCk:POLarity

Function Sets the polarity of the clock channel of the SPI bus signal analysis or queries the current setting.

Syntax :ANALysis:SBUS<x>[:ANALyze]:SPIBus:CLOCk:POLarity {FALL|RISE}
:ANALysis:SBUS<x>[:ANALyze]:SPIBus:CLOCk:POLarity?
<x> = 1 or 2

Example :ANALYSIS:SBUS1:ANALYZE:SPIBUS:CLOCK:POLARITY FALL
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:CLOCK:POLARITY? ->
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:CLOCK:POLARITY FALL

:ANALysis:SBUS<x>[:ANALyze]:SPIBus:CLOCk:SOURce

Function Sets the clock channel of the SPI bus signal analysis or queries the current setting.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:SPIBus:
CLOCk:SOURce {<NRf>}
:ANALysis:SBUS<x>[:ANALyze]:SPIBus:
CLOCk:SOURce?
```
<x> = 1 or 2
<NRf> = 1 to 8

Example
```
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:CLOCK:
SOURCE 1
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:CLOCK:
SOURCE? ->
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:CLOCK:
SOURCE 1
```

:ANALysis:SBUS<x>[:ANALyze]:SPIBus:CS?

Function Queries all settings related to the chip select channel of the SPI bus signal analysis.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:SPIBus:CS?
```
<x> = 1 or 2

Example
```
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:CS? ->
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:CS:
ACTIVE HIGH;TRACE 1
```

:ANALysis:SBUS<x>[:ANALyze]:SPIBus:CS:ACTive

Function Sets the active level of the chip select channel of the SPI bus signal analysis or queries the current setting.

Syntax:
```
ANALysis:SBUS<x>[:ANALyze]:SPIBus:CS:
ACTive {HIGH|LOW}
:ANALysis:SBUS<x>[:ANALyze]:SPIBus:CS:
ACTive?
```
<x> = 1 or 2

Example:
```
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:CS:
ACTIVE HIGH
:ANALYSIS:SBUS1:ANALYZE:
SPIBUS:CS:ACTIVE? ->
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:CS:
ACTIVE HIGH
```

:ANALysis:SBUS<x>[:ANALyze]:SPIBus:CS:TRACe

Function Sets the chip select channel of the SPI bus signal analysis or queries the current setting.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:SPIBus:
CS:TRACe {<NRf>}
:ANALysis:SBUS<x>[:ANALyze]:SPIBus:
CS:TRACe?
```
<x> = 1 or 2
<NRf> = 1 to 8

Example
```
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:CS:
TRACE 1
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:
CS:TRACE? ->
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:CS:
TRACE 1
```

:ANALysis:SBUS<x>[:ANALyze]:SPIBus:DATA<x>?

Function Queries all settings related to the data of the SPI bus signal analysis.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:SPIBus:
DATA<x>?
```
<x> of SBUS<x> = 1 or 2
<x> of DATA<x> = 1 or 2

Example
```
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:
DATA1? ->
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:DATA1:
ACTIVE HIGH;TRACE 1
```

:ANALysis:SBUS<x>[:ANALyze]:SPIBus:DATA<x>:ACTive

Function Sets the active level of the data of the SPI bus signal analysis or queries the current setting.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:SPIBus:
DATA<x>:ACTive {HIGH|LOW}
:ANALysis:SBUS<x>[:ANALyze]:SPIBus:
DATA<x>:ACTive?
```
<x> of SBUS<x> = 1 or 2
<x> of DATA<x> = 1 or 2

Example
```
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:DATA1:
ACTIVE HIGH
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:DATA1:
ACTIVE? ->
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:DATA1:
ACTIVE HIGH
```

:ANALysis:SBUS<x>[:ANALyze]:SPIBus:DATA<x>:TRACe

Function Sets the data channel of the SPI bus signal analysis or queries the current setting.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:SPIBus:
DATA<x>:TRACe {<NRf>}
:ANALysis:SBUS<x>[:ANALyze]:SPIBus:
DATA<x>:TRACe?
```
<x> of SBUS<x> = 1 or 2
<x> of DATA<x> = 1 or 2
<NRf> = 1 to 8

Example
```
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:DATA1:
TRACE 1
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:DATA1:
TRACE? ->
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:DATA1:
TRACE 1
```

:ANALysis:SBUS<x>[:ANALyze]:SPIBus:SETup?

Function Queries all settings related to the SPI bus signal analysis setup.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:SPIBus:
SETup?
```
<x> = 1 or 2

Example
```
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:SETUP?
-> :ANALYSIS:SBUS1:ANALYZE:SPIBUS:
SETUP:BITORDER LSBFIRST;MODE WIRE3
```

:ANALysis:SBUS<x>[:ANALyze]:SPIBus[:SETup]:BITorder

Function Sets the bit order of the SPI bus signal analysis or queries the current setting.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:
SPIBus[:SETup]:
BITorder {LSBFirst|MSBFirst}
:ANALysis:SBUS<x>[:ANALyze]:
SPIBus[:SETup]:BITorder?
```
<x> = 1 or 2

Example
```
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:SETUP:
BITORDER LSBFIRST
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:SETUP:
BITORDER? ->
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:SETUP:
BITORDER LSBFIRST
```

:ANALysis:SBUS<x>[:ANALyze]:SPIBus[:SETup]:MODE

Function Sets the wiring system of the SPI bus signal analysis (three-wire or four-wire) or queries the current setting.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:
SPIBus[:SETup]:MODE {WIRE3|WIRE4}
:ANALysis:SBUS<x>[:ANALyze]:
SPIBus[:SETup]:MODE?
```
<x> = 1 or 2

Example
```
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:
SETUP:MODE WIRE3
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:
SETUP:MODE? ->
:ANALYSIS:SBUS1:ANALYZE:SPIBUS:
SETUP:MODE WIRE3
```

:ANALysis:SBUS<x>[:ANALyze]:TRACe<x>?

Function Queries all settings related to the threshold level of the source channel of the serial bus signal analysis.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:TRACe<x>?
```
<x> of SBUS<x> = 1 or 2
<x> of TRACe<x> = 1 to 8

Example
```
:ANALYSIS:SBUS1:ANALYZE:TRACE1? ->
:ANALYSIS:SBUS1:ANALYZE:TRACE1:
HYSTERESIS 1.000E+00;LEVEL 1.000E+00
```

:ANALysis:SBUS<x>[:ANALyze]:TRACe<x>:HYSTeresis

Function Sets the hysteresis of the threshold level of the source channel of the serial bus signal analysis or queries the current setting.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:TRACe<x>:
HYSTeresis {<NRf>}
:ANALysis:SBUS<x>[:ANALyze]:TRACe<x>:
HYSTeresis?
```
<x> of SBUS<x> = 1 or 2
<x> of TRACe<x> = 1 to 8
<NRf> = 0 to 4 (div)

Example
```
:ANALYSIS:SBUS1:ANALYZE:TRACE1:
HYSTERESIS 1
:ANALYSIS:SBUS1:ANALYZE:TRACE1:
HYSTERESIS? ->
:ANALYSIS:SBUS1:ANALYZE:TRACE1:
HYSTERESIS 1.000E+00
```

:ANALysis:SBUS<x>[:ANALyze]:TRACe<x>:LEVel

Function Sets the level of the threshold level of the source channel of the serial bus signal analysis or queries the current setting.

Syntax
```
:ANALysis:SBUS<x>[:ANALyze]:TRACe<x>:
LEVel {<NRf>|<Voltage>|<current>}
:ANALysis:SBUS<x>[:ANALyze]:TRACe<x>:
LEVel?
```
<x> of SBUS<x> = 1 or 2
<x> of TRACe<x> = 1 to 8
<NRf>, <Voltage>, and <Current> = See the *User's Manual (IM701310-01E)*.

Example
```
:ANALYSIS:SBUS1:ANALYZE:TRACE1:
LEVEL 1V
:ANALYSIS:SBUS1:ANALYZE:TRACE1:LEVEL?
-> :ANALYSIS:SBUS1:ANALYZE:TRACE1:
LEVEL 1.000E+00
```

:ANALysis:SBUS<x>:ZLINkage

Function Sets the zoom link of the serial bus signal analysis or queries the current setting.

Syntax
```
:ANALysis:SBUS<x>:ZLINkage {OFF|Z1|Z2}
:ANALysis:SBUS<x>:ZLINkage?
```
<x> = 1 or 2

Example
```
:ANALYSIS:SBUS1:ZLINKAGE OFF
:ANALYSIS:SBUS1:ZLINKAGE? -
>:ANALYSIS:SBUS1:ZLINKAGE OFF
```

:ANALysis:TYPE<x>

Function Sets the analysis function type or queries the current setting.

Syntax
```
:ANALysis:TYPE<x>
{AHIStogram|FFT|SBUS|WPARameter|XY}
:ANALysis:TYPE<x>?
```
<x> = 1 or 2

Example
```
:ANALYSIS:TYPE1 AHISTOGRAM
:ANALYSIS:TYPE1? ->
:ANALYSIS:TYPE1 AHISTOGRAM
```

5.3 MATH Group

:MATH<x>:OPERation

Function Sets the operator or queries the current setting.

Syntax `:MATH<x>:OPERation {(ECOunt|FILTer|INTegral|MINus|MULTiple|PLUS|RCOunt|SBIT),<NRf>,<NRf>}:MATH<x>:OPERation?`

<x> = 1 to 8 (<x> = 1 to 4 only when you select {SBIT})

<NRf> = 1 to 4

Example `:MATH1:OPERATION FILTER,1:MATH1:OPERATION? -> :MATH1:OPERATION FILTER,1`

Description For unary operators (`ECOunt|FILTer|INTegral|SBIT`), select the target waveform using the first <NRf>.

For binary operators (`MINus|MULTiple|PLUS|RCOunt`), select the target waveform of the first term using the first <NRf> and the target waveform of the second term using the second <NRf>.

:MATH<x>:SBIT?

Function Queries all settings related to the stuff bit computation.

Syntax `:MATH<x>:SBIT?`

<x> = 1 to 4

Example `:MATH1:SBIT? -> :MATH1:SBIT:BRATE 1000000;HYSTERESIS 600.000E-03;LEVEL 0.0000000E+00;RECESSIVE HIGH`

:MATH<x>:SBIT:BRATe

Function Sets the bit rate (data transfer rate) of the stuff bit computation or queries the current setting.

Syntax `:MATH<x>:SBIT:BRATe {<NRf>|USER,<NRf>}`
`:MATH<x>:SBIT:BRATe?`

<x> = 1 to 4

<NRf> = 83300, 125000, 250000, 500000, 1000000

<NRf> of USER = See the *User's Manual (IM701310-01E)*.

Example `:MATH1:SBIT:BRATE 83300`
`:MATH1:SBIT:BRATE? -> :MATH1:SBIT:BRATE 83300`

:MATH<x>:SBIT:HISTory:ABORt

Function Cancels history computation for stuff bit computation.

Syntax `:MATH<x>:SBIT:HISTory:ABORt`

<x>=1-4

Example `:MATH1:SBIT:HISTORY:ABORT`

:MATH<x>:SBIT:HISTory:EXECute

Function Executes history computation for stuff bit computation.

Syntax `:MATH<x>:SBIT:HISTory:EXECute`

<x>=1-4

Example `:MATH1:SBIT:HISTORY:EXECUTE`

:MATH<x>:SBIT:HYSTeresis

Function Sets the hysteresis of the stuff bit computation or queries the current setting.

Syntax `:MATH<x>:SBIT:HYSTeresis {<NRf>}`
`:MATH<x>:SBIT:HYSTeresis?`

<x> = 1 to 4

<NRf> = 0 to 4(div)

Example `:MATH1:SBIT:HYSTERESIS 1`
`:MATH1:SBIT:HYSTERESIS? -> :MATH1:SBIT:HYSTERESIS 1.00000E+00`

:MATH<x>:SBIT:LEVel

Function Sets the threshold level of the stuff bit computation or queries the current setting.

Syntax `:MATH<x>:SBIT:LEVel {<NRf>|<Voltage>|<Current>}`
`:MATH<x>:SBIT:LEVel?`

<x> = 1 to 4

<NRf>, <Voltage>, and <Current> = See the *User's Manual (IM701310-01E)*.

Example `:MATH1:SBIT:LEVEL 1`
`:MATH1:SBIT:LEVEL? -> :MATH1:SBIT:LEVEL 1.0000000E+00`

:MATH<x>:SBIT:RECessive

Function Sets the recessive level (bus level) of the stuff bit computation or queries the current setting.

Syntax `:MATH<x>:SBIT:RECessive {HIGH|LOW}`
`:MATH<x>:SBIT:RECessive?`

<x> = 1 to 4

Example `:MATH1:SBIT:RECESSIVE HIGH`
`:MATH1:SBIT:RECESSIVE? -> :MATH1:SBIT:RECESSIVE HIGH`

:MATH<x>:SBIT:SPOint

Function Sets the sample point of the stuff bit computation or queries the current setting.

Syntax `:MATH<x>:SBIT:SPOint {<NRf>}`
`:MATH<x>:SBIT:SPOint?`

<x> = 1 to 4

<NRf> = 18.8 to 90.6(%)

Example `:MATH1:SBIT:SPOINT 18.8`
`:MATH1:SBIT:SPOINT? -> :MATH1:SBIT:SPOINT 18.8E+00`

5.4 SEARch Group

SEARch<x>:CANBus?

Function Queries all settings related to the CAN bus signal search.

Syntax `:SEARch<x>:CANBus?`
<x> = 1, 2

Example
```
:SEARCH1:CANBUS? ->
:SEARCH1:CANBUS:SETUP:ACK DONTCARE;
BRATE 1000000;DATA:BORDER BIG;
CONDITION DONTCARE;DATA1 0.0000000E+00;
DATA2 255.00000E+00;DLC 8;MSBLSB 7,0;
PATTERN "1001011001011000011101000101001
000010011010101111010111110111100";
SIGN UNSIGN;:SEARCH1:CANBUS:SETUP:
IDEXT:PATTERN"11001011011100001110111011
1111";:SEARCH1:CANBUS:SETUP:IDSTD:
PATTERN "00011111101";:SEARCH1:CANBUS:
SETUP:MODE SOF;RECESSIVE HIGH;RTR DATA;
SPOINT 62.5E+00;TRACE 1
```

:SEARch<x>:CANBus:SETup?

Function Queries all settings related to the CAN bus signal search setup.

Syntax `:SEARch<x>:CANBus:SETup?`
<x> = 1, 2

Example
```
:SEARCH1:CANBUS:SETUP? ->
:SEARCH1:CANBUS:SETUP:ACK DONTCARE;
BRATE 1000000;DATA:BORDER BIG;
CONDITION DONTCARE;DATA1 0.0000000E+00;
DATA2 255.00000E+00;DLC 8;MSBLSB 7,0;
PATTERN "1001011001011000011101000101001
000010011010101111010111110111100";
SIGN UNSIGN;:SEARCH1:CANBUS:SETUP:
IDEXT:PATTERN "110010110111000011101110
11111";:SEARCH1:CANBUS:SETUP:IDSTD:
PATTERN "00011111101";:SEARCH1:CANBUS:
SETUP:MODE SOF;RECESSIVE HIGH;RTR DATA;
SPOINT 62.5E+00;TRACE 1
```

:SEARch<x>:CANBus[:SETup]:ACK

Function Sets the ACK condition of the CAN bus signal search or queries the current setting.

Syntax
```
:SEARch<x>:CANBus[:SETup]:
ACK {ACK|ACKBoth|DONTcare|NONack}
:SEARch<x>:CANBus[:SETup]:ACK?
```
<x> = 1, 2

Example
```
:SEARCH1:CANBUS:SETUP:ACK ACK
:SEARCH1:CANBUS:SETUP:ACK? ->
:SEARCH1:CANBUS:SETUP:ACK ACK
```

:SEARch<x>:CANBus[:SETup]:BRATe

Function Sets the bit rate (data transfer rate) of the CAN bus signal search or queries the current setting.

Syntax
```
:SEARch<x>:CANBus[:SETup]:
BRATe {<NRf>|USER,<NRf>}
:SEARch<x>:CANBus[:SETup]:BRATe?
```
<x> = 1, 2
<NRf> = 33300, 83300, 125000, 250000, 500000, 1000000
<NRf> of USER = See the *User's Manual (IM701310-01E)*.

Example
```
:SEARCH1:CANBUS:SETUP:BRATE 83300
:SEARCH1:CANBUS:SETUP:BRATE? ->
:SEARCH1:CANBUS:SETUP:BRATE 83300
```

:SEARch<x>:CANBus[:SETup]:DATA?

Function Queries all settings related to the CAN bus signal search data.

Syntax `:SEARch<x>:CANBus[:SETup]:DATA?`
<x> = 1, 2

Example
```
:SEARCH1:CANBUS:SETUP:DATA? ->
:SEARCH1:CANBUS:SETUP:DATA:BORDER BIG;
CONDITION TRUE;DATA1 0.0000000E+00;
DATA2 255.00000E+00;DLC 8;MSBLSB 7,0;
PATTERN "1001011001011000011101000101001
000010011010101111010111110111100";
SIGN UNSIGN
```

:SEARch<x>:CANBus[:SETup]:DATA:BORDer

Function Sets the byte order of the CAN bus signal search data or queries the current setting.

Syntax
```
:SEARch<x>:CANBus[:SETup]:DATA:
BORDer {BIG|LITTle}
:SEARch<x>:CANBus[:SETup]:DATA:BORDer?
```
<x> = 1, 2

Example
```
:SEARCH1:CANBUS:SETUP:DATA:BORDER BIG
:SEARCH1:CANBUS:SETUP:DATA:BORDER? ->
:SEARCH1:CANBUS:SETUP:DATA:BORDER BIG
```

:SEARch<x>:CANBus[:SETup]:DATA:CONDition

Function Sets the data condition of the CAN bus signal search or queries the current setting.

Syntax
```
:SEARch<x>:CANBus[:SETup]:DATA:
CONDition {BETWeen|DONTcare|FALSe|
GTHan|LTHan|ORANge|TRUE}
:SEARch<x>:CANBus[:SETup]:DATA:
CONDition?
```
<x> = 1, 2

Example
```
:SEARCH1:CANBUS:SETUP:DATA:
CONDITION BETWEEN
:SEARCH1:CANBUS:SETUP:DATA:CONDITION? ->
:SEARCH1:CANBUS:SETUP:DATA:
CONDITION BETWEEN
```

:SEARch<x>:CANBus[:SETup]:DATA:DATA<x>

Function Sets the comparison data of the CAN bus signal search data or queries the current setting.

Syntax
```
:SEARch<x>:CANBus[:SETup]:DATA:
DATA<x> {<NRf>}
:SEARch<x>:CANBus[:SETup]:DATA:DATA<x>?
```
<x> of SEARch<x> = 1 or 2
<x> of DATA<x> = 1 or 2
<NRf> = See the *User's Manual (IM701310-01E).*

Example
```
:SEARCH1:CANBUS:SETUP:DATA:DATA1 1
:SEARCH1:CANBUS:SETUP:DATA:DATA1? ->
:SEARCH1:CANBUS:SETUP:DATA:
DATA1 1.0000000E+00
```

Description
- Use `:SEARch<x>:CANBus[:SETup]:DATA:DATA1` when `:SEARch<x>:CANBus[:SETup]:DATA:CONDition GTHan` is specified.
- Use `:SEARch<x>:CANBus[:SETup]:DATA:DATA2` when `:SEARch<x>:CANBus[:SETup]:DATA:CONDition LTHan` is specified.
- Use `:SEARch<x>:CANBus[:SETup]:DATA:DATA1` to set the smaller value and `:SEARch<x>:CANBus[:SETup]:DATA:DATA2` to set the larger value when `:SEARch<x>:CANBus[:SETup]:DATA:CONDition BETWeen|ORANge` is specified.

:SEARch<x>:CANBus[:SETup]:DATA:DLC

Function Sets the number of valid bytes (DLC) of the CAN bus signal search data or queries the current setting.

Syntax
```
:SEARch<x>:CANBus[:SETup]:DATA:
DLC {<NRf>}
:SEARch<x>:CANBus[:SETup]:DATA:DLC?
```
<x> = 1, 2
<NRf> = 0 to 8

Example
```
:SEARCH1:CANBUS:SETUP:DATA:DLC 0
:SEARCH1:CANBUS:SETUP:DATA:DLC? ->
:SEARCH1:CANBUS:SETUP:DATA:DLC 0
```

:SEARch<x>:CANBus[:SETup]:DATA:HEXA

Function Sets the CAN bus signal search data in hexadecimal notation.

Syntax
```
:SEARch<x>:CANBus[:SETup]:DATA:
HEXA {<String>}
```
<x> = 1, 2
<String> = Up to 16 characters by combining '0' to 'F' and 'X' (in one-byte unit)

Example
```
:SEARCH1:CANBUS:SETUP:DATA:HEXA "A9"
```

:SEARch<x>:CANBus[:SETup]:DATA:MSBLsb

Function Sets the MSB and LSB bits of the CAN bus signal search data or queries the current setting.

Syntax
```
:SEARch<x>:CANBus[:SETup]:DATA:
MSBLsb {<NRf>,<NRf>}
:SEARch<x>:CANBus[:SETup]:DATA:MSBLsb?
```
<x> = 1, 2
<NRf> = See the *User's Manual (IM701310-01E).*

Example
```
:SEARCH1:CANBUS:SETUP:DATA:MSBLSB 1,0
:SEARCH1:CANBUS:SETUP:DATA:MSBLSB? ->
:SEARCH1:CANBUS:SETUP:DATA:MSBLSB 1,0
```

:SEARch<x>:CANBus[:SETup]:DATA:PATTern

Function Sets the CAN bus signal search data in binary notation or queries the current setting.

Syntax
```
:SEARch<x>:CANBus[:SETup]:DATA:
PATTern {<String>}
:SEARch<x>:CANBus[:SETup]:DATA:PATTern?
```
<x> = 1, 2
<String> = Up to 64 characters by combining '0,' '1,' and 'X' (in one-byte unit)

Example
```
:SEARCH1:CANBUS:SETUP:DATA:
PATTERN "11011111"
:SEARCH1:CANBUS:SETUP:DATA:PATTERN? ->
:SEARCH1:CANBUS:SETUP:DATA:
PATTERN "11011111"
```

:SEARch<x>:CANBus[:SETup]:DATA:SIGN

Function Sets the sign of the CAN bus signal search data or queries the current setting.

Syntax
```
:SEARch<x>:CANBus[:SETup]:DATA:
SIGN {SIGN|UNSign}
:SEARch<x>:CANBus[:SETup]:DATA:SIGN?
```
<x> = 1, 2

Example
```
:SEARCH1:CANBUS:SETUP:DATA:SIGN SIGN
:SEARCH1:CANBUS:SETUP:DATA:SIGN? ->
:SEARCH1:CANBUS:SETUP:DATA:SIGN SIGN
```

:SEARch<x>:CANBus[:SETup]:IDEXt?

Function Queries all settings related to the ID of the extended format of the CAN bus signal search.

Syntax :SEARch<x>:CANBus[:SETup]:IDEXt?
<x> = 1, 2

Example :SEARCH1:CANBUS:SETUP:IDEXT? ->
:SEARCH1:CANBUS:SETUP:IDEXT:
PATTERN "11001011011100001110111011111"

:SEARch<x>:CANBus[:SETup]:IDEXt:HEXA

Function Sets the ID of the extended format of the CAN bus signal search in hexadecimal notation.

Syntax :SEARch<x>:CANBus[:SETup]:IDEXt:
HEXA {<String>}
<x> = 1, 2
<String> = 8 characters by combining '0' to 'F' and 'X'

Example :SEARCH1:CANBUS:SETUP:IDEXT:
HEXA "1AEF5906"

:SEARch<x>:CANBus[:SETup]:IDEXt:PATTern

Function Sets the ID of the extended format of the CAN bus signal search in binary notation or queries the current setting.

Syntax :SEARch<x>:CANBus[:SETup]:IDEXt:
PATTern {<String>}
:SEARch<x>:CANBus[:SETup]:IDEXt:
PATTern?
<x> = 1, 2
<String> = 29 characters by combining '0,' '1,' and 'X'

Example :SEARCH1:CANBUS:SETUP:IDEXT:
PATTERN "11001011011100001110111011111"
:SEARCH1:CANBUS:SETUP:IDEXT:PATTERN? ->
:SEARCH1:CANBUS:SETUP:IDEXT:
PATTERN "11001011011100001110111011111"

:SEARch<x>:CANBus[:SETup]:IDSTd?

Function Queries all settings related to the ID of the standard format of the CAN bus signal search.

Syntax :SEARch<x>:CANBus[:SETup]:IDSTd?
<x> = 1, 2

Example :SEARCH1:CANBUS:SETUP:IDSTD? ->
:SEARCH1:CANBUS:SETUP:IDSTD:
PATTERN "00011111101"

:SEARch<x>:CANBus[:SETup]:IDSTd:HEXA

Function Sets the ID of the standard format of the CAN bus signal search in hexadecimal notation.

Syntax :SEARch<x>:CANBus[:SETup]:IDSTd:
HEXA {<String>}
<x> = 1, 2
<String> = 3 characters by combining '0' to 'F' and 'X'

Example :SEARCH1:CANBUS:SETUP:IDSTD:HEXA "5DF"

:SEARch<x>:CANBus[:SETup]:IDSTd:PATTern

Function Sets the ID of the standard format of the CAN bus signal search in binary notation or queries the current setting.

Syntax :SEARch<x>:CANBus[:SETup]:IDSTd:
PATTern {<String>}
:SEARch<x>:CANBus[:SETup]:IDSTd:
PATTern?
<x> = 1, 2
<String> = 11 characters by combining '0,' '1,' and 'X'

Example :SEARCH1:CANBUS:SETUP:IDSTD:
PATTERN "10111011111"
:SEARCH1:CANBUS:SETUP:IDSTD:PATTERN? ->
:SEARCH1:CANBUS:SETUP:IDSTD:
PATTERN "10111011111"

:SEARch<x>:CANBus[:SETup]:MODE

Function Sets the CAN bus signal search mode or queries the current setting.

Syntax :SEARch<x>:CANBus[:SETup]:
MODE {EFRame|IDEXt|IDSTd|SOF}
:SEARch<x>:CANBus[:SETup]:MODE?
<x> = 1, 2

Example :SEARCH1:CANBUS:SETUP:MODE EFRAME
:SEARCH1:CANBUS:SETUP:MODE? ->
:SEARCH1:CANBUS:SETUP:MODE EFRAME

:SEARch<x>:CANBus[:SETup]:RECessive

Function Sets the recessive level (bus level) of the CAN bus signal search or queries the current setting.

Syntax :SEARch<x>:CANBus[:SETup]:
RECessive {HIGH|LOW}
:SEARch<x>:CANBus[:SETup]:RECessive?
<x> = 1, 2

Example :SEARCH1:CANBUS:SETUP:RECESSIVE HIGH
:SEARCH1:CANBUS:SETUP:RECESSIVE? ->
:SEARCH1:CANBUS:SETUP:RECESSIVE HIGH

:SEARch<x>:CANBus[:SETup]:RTR

Function Sets the RTR of the CAN bus signal search or queries the current setting.

Syntax :SEARch<x>:CANBus[:SETup]:
RTR {DATA|DONTcare|REMote}
:SEARch<x>:CANBus[:SETup]:RTR?
<x> = 1, 2

Example :SEARCH1:CANBUS:SETUP:RTR DATA
:SEARCH1:CANBUS:SETUP:RTR? ->
:SEARCH1:CANBUS:SETUP:RTR DATA

:SEARch<x>:CANBus[:SETup]:SPOint

Function Sets the sample point of the CAN bus signal search or queries the current setting.

Syntax :SEARch<x>:CANBus[:SETup]:SPOint {<NRf>}
:SEARch<x>:CANBus[:SETup]:SPOint?
<x> = 1, 2
<NRf>=18.8 to 90.6(%)

Example :SEARCH1:CANBUS:SETUP:SPOINT 18.8
:SEARCH1:CANBUS:SETUP:SPOINT? ->
:SEARCH1:CANBUS:SETUP:SPOINT 18.8E+00

:SEARch<x>:CANBus[:SETup]:TRACe

Function Sets the trace of the CAN bus signal search or queries the current setting.

Syntax :SEARch<x>:CANBus[:SETup]:TRACe {<NRf>}
:SEARch<x>:CANBus[:SETup]:TRACe?
<x> = 1, 2
<NRf> = 1 to 8

Example :SEARCH1:CANBUS:SETUP:TRACE 1
:SEARCH1:CANBUS:SETUP:TRACE? ->
:SEARCH1:CANBUS:SETUP:TRACE 1

SEARch<x>:I2CBus?

Function Queries all settings related to the I^{2}C bus signal search.

Syntax :SEARch<x>:I2CBus?
<x> = 1 or 2

Example :SEARCH1:I2CBUS? ->
:SEARCH1:I2CBUS:CLOCK:SOURCE 1;:SEARCH1:I2CBUS:SETUP:ADATA:BIT10ADDRESS:PATTERN "10111011111";:SEARCH1:I2CBUS:SETUP:ADATA:BIT7ADDRESS:PATTERN "11011110";:SEARCH1:I2CBUS:SETUP:ADATA:BIT7APSUB:ADDRESS:PATTERN "11001101";:SEARCH1:I2CBUS:SETUP:ADATA:BIT7APSUB:SADDRESS:PATTERN "11101111";:SEARCH1:I2CBUS:SETUP:ADATA:TYPE BIT7APSUB;:SEARCH1:I2CBUS:SETUP:DATA:BYTE 1;CONDITION TRUE;DPOSITION 0;MODE 0;PATTERN1 "10101011";PATTERN2 "10101011";PATTERN3 "10101011";PATTERN4 "10101011";PMODE DONTCARE;:SEARCH1:I2CBUS:SETUP:GCALL:BIT7MADDRESS:PATTERN "1010101";:SEARCH1:I2CBUS:SETUP:GCALL:SBYTE BIT7MADDRESS;:SEARCH1:I2CBUS:SETUP:MODE SBHSMODE;.....

:SEARch<x>:I2CBus:CLOCk?

Function Queries all settings related to the clock of the I^{2}C bus signal search.

Syntax :SEARch<x>:I2CBus:CLOCk?
<x> = 1 or 2

Example :SEARCH1:I2CBUS:CLOCK? ->
:SEARCH1:I2CBUS:CLOCK:SOURCE 1

:SEARch<x>:I2CBus:CLOCk:SOURce

Function Sets the clock trace of the I^{2}C bus signal search or queries the current setting.

Syntax :SEARch<x>:I2CBus:CLOCk:SOURce {<NRf>}
:SEARch<x>:I2CBus:CLOCk:SOURce?
<x> = 1 or 2
<NRf> = 1 to 8

Example :SEARCH1:I2CBUS:CLOCK:SOURCE 1
:SEARCH1:I2CBUS:CLOCK:SOURCE? ->
:SEARCH1:I2CBUS:CLOCK:SOURCE 1

:SEARch<x>:I2CBus:SETup?

Function Queries all settings related to the I^{2}C bus signal search setup.

Syntax :SEARch<x>:I2CBus:SETup?
<x> = 1 or 2

Example :SEARCH1:I2CBUS:SETUP? ->
:SEARCH1:I2CBUS:SETUP:ADATA:BIT10ADDRESS:PATTERN "10111011111";:SEARCH1:I2CBUS:SETUP:ADATA:BIT7ADDRESS:PATTERN "11011110";:SEARCH1:I2CBUS:SETUP:ADATA:BIT7APSUB:ADDRESS:PATTERN "11001101";:SEARCH1:I2CBUS:SETUP:ADATA:BIT7APSUB:SADDRESS:PATTERN "11101111";:SEARCH1:I2CBUS:SETUP:ADATA:TYPE BIT7APSUB;:SEARCH1:I2CBUS:SETUP:DATA:BYTE 1;CONDITION TRUE;DPOSITION 0;MODE 0;PATTERN1 "10101011";PATTERN2 "10101011";PATTERN3 "10101011";PATTERN4 "10101011";PMODE DONTCARE;:SEARCH1:I2CBUS:SETUP:GCALL:BIT7MADDRESS:PATTERN "1010101";:SEARCH1:I2CBUS:SETUP:GCALL:SBYTE BIT7MADDRESS;:SEARCH1:I2CBUS:SETUP:MODE SBHSMODE;NAIGNORE:HSMODE 0;RACCESS 0;SBYTE 0;:SEARCH1:I2CBUS:SETUP:SBHSMODE:TYPE SBYTE

:SEARch<x>:I2CBus[:SETup]:ADATa?

Function Queries all settings related to the address of the I^{2}C bus signal search.

Syntax :SEARch<x>:I2CBus[:SETup]:ADATa?
<x> = 1 or 2

Example :SEARCH1:I2CBUS:SETUP:ADATA? ->
:SEARCH1:I2CBUS:SETUP:ADATA:
BIT10ADDRESS:PATTERN "10111011111";:
SEARCH1:I2CBUS:SETUP:ADATA:
BIT7ADDRESS:PATTERN "11011110";:
SEARCH1:I2CBUS:SETUP:ADATA:
BIT7APSUB:ADDRESS:PATTERN "11001101";:
SEARCH1:I2CBUS:SETUP:ADATA:BIT7APSUB:
SADDRESS:PATTERN "11101111";:
SEARCH1:I2CBUS:SETUP:ADATA:
TYPE BIT7APSUB

:SEARch<x>:I2CBus[:SETup]:ADATa:BIT10address?

Function Queries all settings related to the 10-bit address of the I^{2}C bus signal search.

Syntax :SEARch<x>:I2CBus[:SETup]:ADATa:
BIT10address?
<x> = 1 or 2

Example :SEARCH1:I2CBUS:SETUP:
ADATA:BIT10ADDRESS? ->
:SEARCH1:I2CBUS:SETUP:ADATA:
BIT10ADDRESS:PATTERN "00011111101"

:SEARch<x>:I2CBus[:SETup]:ADATa:BIT10address:HEXA

Function Sets the 10-bit address of the I^{2}C bus signal search in hexadecimal notation.

Syntax :SEARch<x>:I2CBus[:SETup]:ADATa:
BIT10address:HEXA {<String>}
<x> = 1 or 2
<String> = 3 characters by combining '0' to 'F' and 'X' (bit 8 is the $R/\overline{W}$ bit)

Example :SEARCH1:I2CBUS:SETUP:ADATA:
BIT10ADDRESS:HEXA "5DF"

:SEARch<x>:I2CBus[:SETup]:ADATa:BIT10address:PATTern

Function Sets the 10-bit address of the I^{2}C bus signal search in binary notation or queries the current setting.

Syntax :SEARch<x>:I2CBus[:SETup]:ADATa:
BIT10address:PATTern {<String>}
:SEARch<x>:I2CBus[:SETup]:ADATa:
BIT10address:PATTern?
<x> = 1 or 2
<String> = 11 characters by combining '0', '1', and 'X' (bit 8 is the $R/\overline{W}$ bit)

Example :SEARCH1:I2CBUS:SETUP:ADATA:
BIT10ADDRESS:PATTERN "10111011111"
:SEARCH1:I2CBUS:SETUP:ADATA:
BIT10ADDRESS:PATTERN? ->
:SEARCH1:I2CBUS:SETUP:ADATA:
BIT10ADDRESS:PATTERN "10111011111"

:SEARch<x>:I2CBus[:SETup]:ADATa:BIT7ADdress?

Function Queries all settings related to the 7-bit address of the I^{2}C bus signal search.

Syntax :SEARch<x>:I2CBus[:SETup]:ADATa:
BIT7ADdress?
<x> = 1 or 2

Example :SEARCH1:I2CBUS:SETUP:ADATA:
BIT7ADDRESS? ->
:SEARCH1:I2CBUS:SETUP:ADATA:
BIT7ADDRESS:PATTERN "11011110"

:SEARch<x>:I2CBus[:SETup]:ADATa:BIT7ADdress:HEXA

Function Sets the 7-bit address of the I^{2}C bus signal search in hexadecimal notation.

Syntax :SEARch<x>:I2CBus[:SETup]:ADATa:
BIT7ADdress:HEXA {<String>}
<x> = 1 or 2
<String> = 2 characters by combining '0' to 'F' and 'X' (bit 0 is the $R/\overline{W}$ bit)

Example :SEARCH1:I2CBUS:SETUP:ADATA:
BIT7ADDRESS:HEXA "DE"

:SEARch<x>:I2CBus[:SETup]:ADATa:BIT7ADdress:PATTern

Function Sets the 7-bit address of the I^2C bus signal search in binary notation or queries the current setting.

Syntax :SEARch<x>:I2CBus[:SETup]:ADATa:BIT7ADdress:PATTern {<String>}
:SEARch<x>:I2CBus[:SETup]:ADATa:BIT7ADdress:PATTern?
<x> = 1 or 2
<String> = 8 characters by combining '0', '1', and 'X' (bit 0 is the R/$\overline{W}$ bit)

Example :SEARCH1:I2CBUS:SETUP:ADATA:BIT7ADDRESS:PATTERN "11011110"
:SEARCH1:I2CBUS:SETUP:ADATA:BIT7ADDRESS:PATTERN? ->
:SEARCH1:I2CBUS:SETUP:ADATA:BIT7ADDRESS:PATTERN "11011110"

:SEARch<x>:I2CBus[:SETup]:ADATa:BIT7APsub?

Function Queries all settings related to the 7-bit + Sub address of the I^2C bus signal search.

Syntax :SEARch<x>:I2CBus[:SETup]:ADATa:BIT7APsub?
<x> = 1 or 2

Example :SEARCH1:I2CBUS:SETUP:ADATA:BIT7APSUB? ->
:SEARCH1:I2CBUS:SETUP:ADATA:BIT7APSUB:ADDRESS:PATTERN "11001101";:SEARCH1:I2CBUS:SETUP:ADATA:BIT7APSUB:SADDRESS:PATTERN "11101111"

:SEARch<x>:I2CBus[:SETup]:ADATa:BIT7APsub:ADDRess?

Function Queries all settings related to the 7-bit address of the 7-bit + Sub address of the I^2C bus signal search.

Syntax :SEARch<x>:I2CBus[:SETup]:ADATa:BIT7APsub:ADDRess?
<x> = 1 or 2

Example :SEARCH1:I2CBUS:SETUP:ADATA:BIT7APSUB:ADDRESS? ->
:SEARCH1:I2CBUS:SETUP:ADATA:BIT7APSUB:ADDRESS:PATTERN "11001101"

:SEARch<x>:I2CBus[:SETup]:ADATa:BIT7APsub:ADDRess:HEXA

Function Sets the 7-bit address of the 7-bit + Sub address of the I^2C bus signal search in hexadecimal notation.

Syntax :SEARch<x>:I2CBus[:SETup]:ADATa:BIT7APsub:ADDRess:HEXA {<String>}
<x> = 1 or 2
<String> = 2 characters by combining '0' to 'F' and 'X' (bit 0 is the R/$\overline{W}$ bit)

Example :SEARCH1:I2CBUS:SETUP:ADATA:BIT7APSUB:ADDRESS:HEXA "CD"

:SEARch<x>:I2CBus[:SETup]:ADATa:BIT7APsub:ADDRess:PATTern

Function Sets the 7-bit address of the 7-bit + Sub address of the I^2C bus signal search in binary notation or queries the current setting.

Syntax :SEARch<x>:I2CBus[:SETup]:ADATa:BIT7APsub:ADDRess:PATTern {<String>}
:SEARch<x>:I2CBus[:SETup]:ADATa:BIT7APsub:ADDRess:PATTern?
<x> = 1 or 2
<String> = 8 characters by combining '0', '1', and 'X' (bit 0 is the R/$\overline{W}$ bit)

Example :SEARCH1:I2CBUS:SETUP:ADATA:BIT7APSUB:ADDRESS:PATTERN "11001101"
:SEARCH1:I2CBUS:SETUP:ADATA:BIT7APSUB:ADDRESS:PATTERN? ->
:SEARCH1:I2CBUS:SETUP:ADATA:BIT7APSUB:ADDRESS:PATTERN "11001101"

:SEARch<x>:I2CBus[:SETup]:ADATa:BIT7APsub:SADDress?

Function Queries all settings related to the Sub address of the 7-bit + Sub address of the I^2C bus signal search.

Syntax :SEARch<x>:I2CBus[:SETup]:ADATa:BIT7APsub:SADDress?
<x> = 1 or 2

Example :SEARCH1:I2CBUS:SETUP:ADATA:BIT7APSUB:SADDRESS? ->
:SEARCH1:I2CBUS:SETUP:ADATA:BIT7APSUB:SADDRESS:PATTERN "11101111"

:SEARch<x>:I2CBus[:SETup]:ADATa:BIT7APsub:SADDress:HEXA

Function Sets the Sub address of the 7-bit + Sub address of the I^2C bus signal search in hexadecimal notation.

Syntax :SEARch<x>:I2CBus[:SETup]:ADATa:BIT7APsub:SADDress:HEXA {<String>}
<x> = 1 or 2
<String> = 2 characters by combining '0' to 'F' and 'X'

Example :SEARCH1:I2CBUS:SETUP:ADATA:BIT7APSUB:SADDRESS:HEXA "EF"

:SEARch<x>:I2CBus[:SETup]:ADATa:BIT7APsub:SADDress:PATTern

Function Sets the Sub address of the 7-bit + Sub address of the I^{2}C bus signal search in binary notation or queries the current setting.

Syntax :SEARch<x>:I2CBus[:SETup]:ADATa:BIT7APsub:SADDress:PATTern {<String>}
:SEARch<x>:I2CBus[:SETup]:ADATa:BIT7APsub:SADDress:PATTern?
<x> = 1 or 2
<String> = 8 characters by combining '0,' '1,' and 'X'

Example :SEARCH1:I2CBUS:SETUP:ADATA:BIT7APSUB:SADDRESS:PATTERN "11101111"
:SEARCH1:I2CBUS:SETUP:ADATA:BIT7APSUB:SADDRESS:PATTERN? ->
:SEARCH1:I2CBUS:SETUP:ADATA:BIT7APSUB:SADDRESS:PATTERN "11101111"

:SEARch<x>:I2CBus[:SETup]:ADATa:TYPE

Function Sets the address type of the I^{2}C bus signal search or queries the current setting.

Syntax :SEARch<x>:I2CBus[:SETup]:ADATa:TYPE {BIT10address|BIT7ADdress|BIT7APsub}
:SEARch<x>:I2CBus[:SETup]:ADATa:TYPE?
<x> = 1 or 2

Example :SEARCH1:I2CBUS:SETUP:ADATA:TYPE BIT10ADDRESS
:SEARCH1:I2CBUS:SETUP:ADATA:TYPE? ->
:SEARCH1:I2CBUS:SETUP:ADATA:TYPE BIT10ADDRESS

:SEARch<x>:I2CBus[:SETup]:DATA?

Function Queries all settings related to the data of the I^{2}C bus signal search.

Syntax :SEARch<x>:I2CBus[:SETup]:DATA?
<x> = 1 or 2

Example :SEARCH1:I2CBUS:SETUP:DATA? ->
:SEARCH1:I2CBUS:SETUP:DATA:BYTE 1;CONDITION TRUE;DPOSITION 0;MODE 0;PATTERN1 "10101011";PATTERN2 "10101011";PATTERN3 "10101011";PATTERN4 "10101011";PMODE DONTCARE

:SEARch<x>:I2CBus[:SETup]:DATA:BYTE

Function Sets the number of data bytes of the I^{2}C bus signal search or queries the current setting.

Syntax :SEARch<x>:I2CBus[:SETup]:DATA:BYTE {<NRf>}
:SEARch<x>:I2CBus[:SETup]:DATA:BYTE?
<x> = 1 or 2
<NRf> = 1 to 4

Example :SEARCH1:I2CBUS:SETUP:DATA:BYTE 1
:SEARCH1:I2CBUS:SETUP:DATA:BYTE? ->
:SEARCH1:I2CBUS:SETUP:DATA:BYTE 1

:SEARch<x>:I2CBus[:SETup]:DATA:CONDition

Function Sets the determination method (match or not match) of the data of the I^{2}C bus signal search or queries the current setting.

Syntax :SEARch<x>:I2CBus[:SETup]:DATA:CONDition {FALSe|TRUE}
:SEARch<x>:I2CBus[:SETup]:DATA:CONDition?
<x> = 1 or 2

Example :SEARCH1:I2CBUS:SETUP:DATA:CONDITION TRUE
:SEARCH1:I2CBUS:SETUP:DATA:CONDITION? ->
:SEARCH1:I2CBUS:SETUP:DATA:CONDITION TRUE

:SEARch<x>:I2CBus[:SETup]:DATA:DPOSition

Function Sets the position for comparing the data pattern of the I^{2}C bus signal search or queries the current setting.

Syntax :SEARch<x>:I2CBus[:SETup]:DATA:DPOSition {<NRf>}
:SEARch<x>:I2CBus[:SETup]:DATA:DPOSition?
<x> = 1 or 2
<NRf> = 0 to 9999

Example :SEARCH1:I2CBUS:SETUP:DATA:DPOSITION 1
:SEARCH1:I2CBUS:SETUP:DATA:DPOSITION? ->
:SEARCH1:I2CBUS:SETUP:DATA:DPOSITION 1

:SEARch<x>:I2CBus[:SETup]:DATA:HEXA<x>

Function Sets the data of the I^{2}C bus signal search in hexadecimal notation.

Syntax :SEARch<x>:I2CBus[:SETup]:DATA:HEXA<x> {<String>}
<x> of SEARch<x> = 1 or 2
<x> of HEXA<x> = 1 to 4
<String> = 2 characters by combining '0' to 'F' and 'X'

Example :SEARCH1:I2CBUS:SETUP:DATA:HEXA1 "AB"

:SEARch<x>:I2CBus[:SETup]:DATA:MODE

Function Enables/Disables the data conditions of the I^{2}C bus signal search or queries the current setting.

Syntax :SEARch<x>:I2CBus[:SETup]:DATA:MODE {<Boolean>}
:SEARch<x>:I2CBus[:SETup]:DATA:MODE?
<x> = 1 or 2

Example :SEARCH1:I2CBUS:SETUP:DATA:MODE ON
:SEARCH1:I2CBUS:SETUP:DATA:MODE? ->
:SEARCH1:I2CBUS:SETUP:DATA:MODE 1

:SEARch<x>:I2CBus[:SETup]:DATA:PATTern<x>

Function Sets the data of the I²C bus signal search in binary notation or queries the current setting.

Syntax
```
:SEARch<x>:I2CBus[:SETup]:DATA:
PATTern<x> {<String>}
:SEARch<x>:I2CBus[:SETup]:DATA:
PATTern<x>?
```
<x> of SEARch<x> = 1 or 2
<x> of <PATTern x> = 1 to 4
<String> = 8 characters by combining ‘0,’ ‘1,’ and ‘X’

Example
```
:SEARCH1:I2CBUS:SETUP:DATA:
PATTERN1 "10101011"
:SEARCH1:I2CBUS:SETUP:DATA:PATTERN1?
-> :SEARCH1:I2CBUS:SETUP:DATA:
PATTERN1 "10101011"
```

:SEARch<x>:I2CBus[:SETup]:DATA:PMODe

Function Sets the pattern comparison start position mode of the data of the I²C bus signal search or queries the current setting.

Syntax
```
:SEARch<x>:I2CBus[:SETup]:DATA:
PMODe {DONTcare|SELect}
:SEARch<x>:I2CBus[:SETup]:DATA:PMODe?
```
<x> = 1 or 2

Example
```
:SEARCH1:I2CBUS:SETUP:DATA:
PMODE DONTCARE
:SEARCH1:I2CBUS:SETUP:DATA:PMODE? ->
:SEARCH1:I2CBUS:SETUP:DATA:
PMODE DONTCARE
```

:SEARch<x>:I2CBus[:SETup]:GCALl?

Function Queries all settings related to the general call of the I²C bus signal search.

Syntax
```
:SEARch<x>:I2CBus[:SETup]:GCALl?
```
<x> = 1 or 2

Example
```
:SEARCH1:I2CBUS:SETUP:GCALL? ->
:SEARCH1:I2CBUS:SETUP:GCALL:
BIT7MADDRESS:PATTERN "1010101";:
SEARCH1:I2CBUS:SETUP:GCALL:
SBYTE BIT7MADDRESS
```

:SEARch<x>:I2CBus[:SETup]:GCALl:BIT7maddress?

Function Queries all settings related to the 7-bit master address of the general call of the I²C bus signal search.

Syntax
```
:SEARch<x>:I2CBus[:SETup]:GCALl:
BIT7maddress?
```
<x> = 1 or 2

Example
```
:SEARCH1:I2CBUS:SETUP:GCALL:
BIT7MADDRESS? ->
:SEARCH1:I2CBUS:SETUP:GCALL:
BIT7MADDRESS:PATTERN "1010101"
```

:SEARch<x>:I2CBus[:SETup]:GCALl:BIT7maddress:HEXA

Function Sets the 7-bit master address of the general call of the I²C bus signal search in hexadecimal notation.

Syntax
```
:SEARch<x>:I2CBus[:SETup]:GCALl:
BIT7maddress:HEXA {<String>}
```
<x> = 1 or 2
<String> = 2 characters by combining ‘0’ to ‘F’ and ‘X’ (bit 0 is fixed 1)

Example
```
:SEARCH1:I2CBUS:SETUP:GCALL:
BIT7MADDRESS:HEXA "BA"
```

:SEARch<x>:I2CBus[:SETup]:GCALl:BIT7maddress:PATTern

Function Sets the 7-bit master address of the general call of the I²C bus signal search in binary notation or queries the current setting.

Syntax
```
:SEARch<x>:I2CBus[:SETup]:GCALl:
BIT7maddress:PATTern {<String>}
:SEARch<x>:I2CBus[:SETup]:GCALl:
BIT7maddress:PATTern?
```
<x> = 1 or 2
<String> = 7 characters by combining ‘0,’ ‘1,’ and ‘X’

Example
```
:SEARCH1:I2CBUS:SETUP:GCALL:
BIT7MADDRESS:PATTERN "1010101"
:SEARCH1:I2CBUS:SETUP:GCALL:
BIT7MADDRESS:PATTERN? ->
:SEARCH1:I2CBUS:SETUP:GCALL:
BIT7MADDRESS:PATTERN "1010101"
```

:SEARch<x>:I2CBus[:SETup]:GCALl:SBYTe (Second Byte)

Function Sets the second byte type of the general call of the I²C bus signal search or queries the current setting.

Syntax
```
:SEARch<x>:I2CBus[:SETup]:GCALl:
SBYTe {BIT7maddress|DONTcare|H04|H06}
:SEARch<x>:I2CBus[:SETup]:GCALl:SBYTe?
```
<x> = 1 or 2

Example
```
:SEARCH1:I2CBUS:SETUP:GCALL:
SBYTE BIT7MADDRESS
:SEARCH1:I2CBUS:SETUP:GCALL:SBYTE? ->
:SEARCH1:I2CBUS:SETUP:GCALL:
SBYTE BIT7MADDRESS
```

:SEARch<x>:I2CBus[:SETup]:MODE

Function Sets the search mode of the I^{2}C bus signal search or queries the current setting.

Syntax :SEARch<x>:I2CBus[:SETup]:
MODE {ADATa|EStart|GCALl|NAIGnore|
SBHSmode}
:SEARch<x>:I2CBus[:SETup]:MODE?
<x> = 1 or 2

Example :SEARCH1:I2CBUS:SETUP:MODE ADATA
:SEARCH1:I2CBUS:SETUP:MODE? ->
:SEARCH1:I2CBUS:SETUP:MODE ADATA

:SEARch<x>:I2CBus[:SETup]:NAIGnore?

Function Queries all settings related to the NON ACK ignore mode of the I^{2}C bus signal search.

Syntax :SEARch<x>:I2CBus[:SETup]:NAIGnore?
<x> = 1 or 2

Example :SEARCH1:I2CBUS:SETUP:NAIGNORE? ->
:SEARCH1:I2CBUS:SETUP:NAIGNORE:
HSMODE 1;RACCESS 1;SBYTE 1

:SEARch<x>:I2CBus[:SETup]:NAIGnore:HSMode

Function Sets whether to ignore NON ACK in high speed mode of the I^{2}C bus signal search or queries the current setting.

Syntax :SEARch<x>:I2CBus[:SETup]:NAIGnore:
HSMode {<Boolean>}
:SEARch<x>:I2CBus[:SETup]:NAIGnore:
HSMode?
<x> = 1 or 2

Example :SEARCH1:I2CBUS:SETUP:NAIGNORE:
HSMODE ON
:SEARCH1:I2CBUS:SETUP:NAIGNORE:HSMODE? ->
:SEARCH1:I2CBUS:SETUP:NAIGNORE:HSMODE 1

:SEARch<x>:I2CBus[:SETup]:NAIGnore:RACCess

Function Sets whether to ignore NON ACK in read access mode of the I^{2}C bus signal search or queries the current setting.

Syntax :SEARch<x>:I2CBus[:SETup]:NAIGnore:
RACCess {<Boolean>}
:SEARch<x>:I2CBus[:SETup]:NAIGnore:
RACCess?
<x> = 1 or 2

Example :SEARCH1:I2CBUS:SETUP:NAIGNORE:
RACCESS ON
:SEARCH1:I2CBUS:SETUP:NAIGNORE:
RACCESS? ->
:SEARCH1:I2CBUS:SETUP:NAIGNORE:
RACCESS 1

:SEARch<x>:I2CBus[:SETup]:NAIGnore:SBYTe(Start Byte)

Function Sets whether to ignore NON ACK in the start byte of the I^{2}C bus signal search or queries the current setting.

Syntax :SEARch<x>:I2CBus[:SETup]:NAIGnore:
SBYTe {<Boolean>}
:SEARch<x>:I2CBus[:SETup]:NAIGnore:
SBYTe?
<x> = 1 or 2

Example :SEARCH1:I2CBUS:SETUP:NAIGNORE:SBYTE ON
:SEARCH1:I2CBUS:SETUP:NAIGNORE:
SBYTE? ->
:SEARCH1:I2CBUS:SETUP:NAIGNORE:SBYTE 1

:SEARch<x>:I2CBus[:SETup]:SBHSmode?

Function Queries all settings related to the start byte and high speed mode of the I^{2}C bus signal search.

Syntax :SEARch<x>:I2CBus[:SETup]:SBHSmode?
<x> = 1 or 2

Example :SEARCH1:I2CBUS:SETUP:SBHSMODE? ->
:SEARCH1:I2CBUS:SETUP:SBHSMODE:
TYPE HSMODE

:SEARch<x>:I2CBus[:SETup]:SBHSmode:TYPE

Function Sets the type of the start byte or high speed mode of the I^{2}C bus signal search or queries the current setting.

Syntax :SEARch<x>:I2CBus[:SETup]:SBHSmode:
TYPE {HSMode|SBYTe}
:SEARch<x>:I2CBus[:SETup]:SBHSmode:
TYPE?
<x> = 1 or 2

Example :SEARCH1:I2CBUS:SETUP:SBHSMODE:
TYPE HSMODE
:SEARCH1:I2CBUS:SETUP:SBHSMODE:TYPE? ->
:SEARCH1:I2CBUS:SETUP:SBHSMODE:
TYPE HSMODE

:SEARch<x>:LINBus?

Function Queries all settings related to the LIN bus signal search or queries the current setting.

Syntax SEARch<x>:LINBus?
<x>=1, 2

Example SEARCH1:LINBUS? -> :SEARCH1:LINBUS:
SETUP:BRATE 19200;DATA:BNUM 8;
CONDITION DONTCARE;PATTERN "11011111";:
SEARCH1:LINBUS:SETUP:ID:
PATTERN "101111";:SEARCH1:LINBUS:SETUP:
MODE SYNCH;TRACE 1

:SEARch<x>:LINBus:SETup?

Function Queries all settings related to setup of the LIN bus signal search or queries the current setting.

Syntax `SEARch<x>:LINBus[:SETup]?`
<x>=1, 2

Example `:SEARCH1:LINBUS:SETUP? -> :SEARCH1:LINBUS:SETUP:BRATE 19200;DATA:BNUM 8;CONDITION DONTCARE;PATTERN "11011111";:SEARCH1:LINBUS:SETUP:ID:PATTERN "101111";:SEARCH1:LINBUS:SETUP:MODE SYNCH;TRACE 1`

:SEARch<x>:LINBus[:SETup]:BRATe

Function Sets the LIN bus signal search bitrate (data transfer rate) or queries the current setting.

Syntax `:SEARch<x>:LINBus[:SETup]:BRATe {<NRf>|USER,<NRf>}`
`:SEARch<x>:LINBus[:SETup]:BRATe?`
<x>=1, 2
<NRf>=1200, 2400, 4800, 9600, 19200
USER <NRf>=See this *User's Manual*.

Example `:SEARCH1:LINBUS:SETUP:BRATE 19200`
`:SEARCH1:LINBUS:SETUP:BRATE? -> :SEARCH1:LINBUS:SETUP:BRATE 19200`

:SEARch<x>:LINBus[:SETup]:DATA?

Function Queries all settings related to data of the LIN bus signal search or queries the current setting.

Syntax `:SEARch<x>:LINBus[:SETup]:DATA?`
<x>=1, 2

Example `:SEARCH1:LINBUS:SETUP:DATA? -> :SEARCH1:LINBUS:SETUP:DATA:BNUM 8;CONDITION DONTCARE;PATTERN "11011111"`

:SEARch<x>:LINBus[:SETup]:DATA:BNUM

Function Sets the number of LIN bus signal search data bytes or queries the current setting.

Syntax `:SEARch<x>:LINBus[:SETup]:DATA:BNUM {<NRf>}`
`:SEARch<x>:LINBus[:SETup]:DATA:BNUM?`
<x>=1, 2
<NRf>=1–8

Example `:SEARCH1:LINBUS:SETUP:DATA:BNUM 1`
`:SEARCH1:LINBUS:SETUP:DATA:BNUM? -> :SEARCH1:LINBUS:SETUP:DATA:BNUM 1`

:SEARch<x>:LINBus[:SETup]:DATA:CONDition

Function Sets the LIN bus signal search data or queries the current setting.

Syntax `:SEARch<x>:LINBus[:SETup]:DATA:CONDition {DONTcare|FALSe|TRUE}`
`:SEARch<x>:LINBus[:SETup]:DATA:CONDition?`
<x>=1, 2

Example `:SEARCH1:LINBUS:SETUP:DATA:CONDITION DONTCARE`
`:SEARCH1:LINBUS:SETUP:DATA:CONDITION? -> :SEARCH1:LINBUS:SETUP:DATA:CONDITION DONTCARE`

:SEARch<x>:LINBus[:SETup]:DATA:HEXA

Function Sets the LIN bus signal search data in hexadecimal.

Syntax `:SEARch<x>:LINBus[:SETup]:DATA:HEXA {<string>}`
<x>=1, 2
<string>=Combination of up to 16 hex characters ('0' – 'F' and 'X') (changed with the BNUM setting)

Example `:SEARCH1:LINBUS:SETUP:DATA:HEXA "3B"`

:SEARch<x>:LINBus[:SETup]:DATA:PATTern

Function Sets the LIN bus signal search data in binary or queries the current setting.

Syntax `:SEARch<x>:LINBus[:SETup]:DATA:PATTern {<string>}`
`:SEARch<x>:LINBus[:SETup]:DATA:PATTern?`
<x>=1, 2
<string>=Combination of up to 64 characters ('0,' '1,' and 'X') (changed with the BNUM setting)

Example `:SEARCH1:LINBUS:SETUP:DATA:PATTERN "11011111"`
`:SEARCH1:LINBUS:SETUP:DATA:PATTERN? -> :SEARCH1:LINBUS:SETUP:DATA:PATTERN "11011111"`

:SEARch<x>:LINBus[:SETup]:ID?

Function Queries all settings related to ID of the LIN bus signal search or queries the current setting.

Syntax `:SEARch<x>:LINBus[:SETup]:ID?`
<x>=1, 2

Example `:SEARCH1:LINBUS:SETUP:ID? -> :SEARCH1:LINBUS:SETUP:ID:PATTERN "101111"`

:SEARch<x>:LINBus[:SETup]:ID:HEXA

Function Sets the LIN bus signal search ID in hexadecimal.

Syntax `:SEARch<x>:LINBus[:SETup]:ID:HEXA {<string>}`
<x>=1, 2
<string>=Combination of up to 2 characters ('0' –'F' and 'X')

Example `:SEARCH1:LINBUS:SETUP:ID:HEXA "2A"`

:SEARch<x>:LINBus[:SETup]:ID:PATTern

Function Sets the LIN bus signal search ID in binary or queries the current setting.

Syntax `:SEARch<x>:LINBus[:SETup]:ID:PATTern {<string>}`
`:SEARch<x>:LINBus[:SETup]:ID:PATTern?`
<x>=1, 2
<string>=Combination of up to 6 characters ('0,' '1,' and 'X')

Example `:SEARCH1:LINBUS:SETUP:ID:PATTERN "101111"`
`:SEARCH1:LINBUS:SETUP:ID:PATTERN? -> :SEARCH1:LINBUS:SETUP:ID:PATTERN "101111"`

:SEARch<x>:LINBus[:SETup]:MODE

Function Sets the LIN bus signal search mode or queries the current setting.

Syntax `SEARch<x>:LINBus[:SETup]:MODE {IDData|SYNCh}`
`:SEARch<x>:LINBus[:SETup]:MODE?`
<x>=1, 2

Example `:SEARCH1:LINBUS:SETUP:MODE IDDATA`
`:SEARCH1:LINBUS:SETUP:MODE? -> :SEARCH1:LINBUS:SETUP:MODE IDDATA`

:SEARch<x>:LINBus[:SETup]:TRACe

Function Sets the LIN bus signal search trace or queries the current setting.

Syntax `:SEARch<x>:LINBus[:SETup]:TRACe {<NRf>}`
`:SEARch<x>:LINBus[:SETup]:TRACe?`
<x>=1, 2
<NRf>=1–8

Example `:SEARCH1:LINBUS:SETUP:TRACE 1`
`:SEARCH1:LINBUS:SETUP:TRACE? -> :SEARCH1:LINBUS:SETUP:TRACE 1`

:SEARch<x>:SPIBus?

Function Queries all settings related to the SPI bus signal search.

Syntax `:SEARch<x>:SPIBus?`
<x> = 1 or 2

Example
```
:SEARCH1:SPIBUS? ->
:SEARCH1:SPIBUS:CLOCK:POLARITY FALL;
SOURCE 1;:SEARCH1:SPIBUS:CS:
ACTIVE HIGH;TRACE 1;:SEARCH1:SPIBUS:
SETUP:BITORDER LSBFIRST;DATA1:BYTE 1;
CONDITION TRUE;DPOSITION 1;
PATTERN1 "00010010";
PATTERN2 "00110100";
PATTERN3 "01010110";
PATTERN4 "01111000";TRACE 1;:SEARCH1:
SPIBUS:SETUP:DATA2:BYTE 1;
CONDITION TRUE;DPOSITION 1;
PATTERN1 "00010010";
PATTERN2 "00110100";
PATTERN3 "01010110";
PATTERN4 "01111000";TRACE 1;:SEARCH1:
SPIBUS:SETUP:MODE WIRE3
```

:SEARch<x>:SPIBus:CLOCk

Function Queries all settings related to the clock channel of the SPI bus signal search.

Syntax `:SEARch<x>:SPIBus:CLOCk?`
<x> = 1 or 2

Example `:SEARCH1:SPIBUS:CLOCK? -> :SEARCH1:SPIBUS:CLOCK:POLARITY FALL; SOURCE 1`

:SEARch<x>:SPIBus:CLOCk:POLarity

Function Sets the polarity of the clock channel of the SPI bus signal search or queries the current setting.

Syntax `:SEARch<x>:SPIBus:CLOCk:POLarity {FALL|RISE}`
`:SEARch<x>:SPIBus:CLOCk:POLarity?`
<x> = 1 or 2

Example `:SEARCH1:SPIBUS:CLOCK:POLARITY FALL`
`:SEARCH1:SPIBUS:CLOCK:POLARITY? -> :SEARCH1:SPIBUS:CLOCK:POLARITY FALL`

:SEARch<x>:SPIBus:CLOCk:SOURce

Function Sets the clock channel of the SPI bus signal search or queries the current setting.

Syntax `:SEARch<x>:SPIBus:CLOCk:SOURce {<NRf>}`
`:SEARch<x>:SPIBus:CLOCk:SOURce?`
<x> = 1 or 2
<NRf> = 1 to 8

Example `:SEARCH1:SPIBUS:CLOCK:SOURCE 1`
`:SEARCH1:SPIBUS:CLOCK:SOURCE? -> :SEARCH1:SPIBUS:CLOCK:SOURCE 1`

:SEARch<x>:SPIBus:CS?

Function Queries all settings related to the chip select channel of the SPI bus signal search.

Syntax :SEARch<x>:SPIBus:CS?
<x> = 1 or 2

Example :SEARCH1:SPIBUS:CS? ->
:SEARCH1:SPIBUS:CS:ACTIVE HIGH;TRACE 1

:SEARch<x>:SPIBus:CS:ACTive

Function Sets the active level of the chip select channel of the SPI bus signal search or queries the current setting.

Syntax :SEARch<x>:SPIBus:CS:ACTive {HIGH|LOW}
:SEARch<x>:SPIBus:CS:ACTive?
<x> = 1 or 2

Example :SEARCH1:SPIBUS:CS:ACTIVE HIGH
:SEARCH1:SPIBUS:CS:ACTIVE? ->
:SEARCH1:SPIBUS:CS:ACTIVE HIGH

:SEARch<x>:SPIBus:CS:TRACe

Function Sets the chip select channel of the SPI bus signal search or queries the current setting.

Syntax :SEARch<x>:SPIBus:CS:TRACe {<NRf>}
:SEARch<x>:SPIBus:CS:TRACe?
<x> = 1 or 2
<NRf> = 1 to 8

Example :SEARCH1:SPIBUS:CS:TRACE 1
:SEARCH1:SPIBUS:CS:TRACE? ->
:SEARCH1:SPIBUS:CS:TRACE 1

:SEARch<x>:SPIBus:SETup?

Function Queries all settings related to the SPI bus signal search setup.

Syntax :SEARch<x>:SPIBus:SETup?
<x> = 1 or 2

Example :SEARCH1:SPIBUS:SETUP? ->
:SEARCH1:SPIBUS:SETUP:
BITORDER LSBFIRST;DATA1:BYTE 1;
CONDITION TRUE;DPOSITION 1;
PATTERN1 "00010010";
PATTERN2 "00110100";
PATTERN3 "01010110";
PATTERN4 "01111000";TRACE 1;:SEARCH1:
SPIBUS:SETUP:DATA2:BYTE 1;
CONDITION TRUE;DPOSITION 1;
PATTERN1 "00010010";
PATTERN2 "00110100";
PATTERN3 "01010110";
PATTERN4 "01111000";TRACE 1;:SEARCH1:
SPIBUS:SETUP:MODE WIRE3

:SEARch<x>:SPIBus[:SETup]:BITorder

Function Sets the bit order of the SPI bus signal search or queries the current setting.

Syntax :SEARch<x>:SPIBus[:SETup]:
BITorder {LSBFirst|MSBFirst}
:SEARch<x>:SPIBus[:SETup]:BITorder?
<x> = 1 or 2

Example :SEARCH1:SPIBUS:SETUP:BITORDER LSBFIRST
:SEARCH1:SPIBUS:SETUP:BITORDER? ->
:SEARCH1:SPIBUS:SETUP:BITORDER LSBFIRST

:SEARch<x>:SPIBus[:SETup]:DATA<x>?

Function Queries all settings related to the data of the SPI bus signal search.

Syntax :SEARch<x>:SPIBus[:SETup]:DATA<x>?
<x> of SEARch<x> = 1 or 2
<x> of DATA<x> = 1 or 2

Example :SEARCH1:SPIBUS:SETUP:DATA1? ->
:SEARCH1:SPIBUS:SETUP:DATA1:BYTE 1;
CONDITION TRUE;DPOSITION 1;
PATTERN1 "00010010";
PATTERN2 "00110100";
PATTERN3 "01010110";
PATTERN4 "01111000";TRACE 1

Description DATA2 is valid when :SEARch<x>:
SPIBus[:SETup]:MODE WIRE4 is specified.

:SEARch<x>:SPIBus[:SETup]:DATA<x>:BYTE

Function Sets the number of bytes of the data of the SPI bus signal search or queries the current setting.

Syntax :SEARch<x>:SPIBus[:SETup]:DATA<x>:
BYTE {<NRf>}
:SEARch<x>:SPIBus[:SETup]:DATA<x>:BYTE?
<x> of SEARch<x> = 1 or 2
<x> of DATA<x> = 1 or 2
<NRf> = 1 to 4

Example :SEARCH1:SPIBUS:SETUP:DATA1:BYTE 1
:SEARCH1:SPIBUS:SETUP:DATA1:BYTE? ->
:SEARCH1:SPIBUS:SETUP:DATA1:BYTE 1

:SEARch<x>:SPIBus[:SETup]:DATA<x>:CONDition

Function Sets the determination method (match or not match) of the data of the SPI bus signal search or queries the current setting.

Syntax :SEARch<x>:SPIBus[:SETup]:DATA<x>:
CONDition {FALSe|TRUE}
:SEARch<x>:SPIBus[:SETup]:DATA<x>:
CONDition?
<x> of SEARch<x> = 1 or 2
<x> of DATA<x> = 1 or 2

Example :SEARCH1:SPIBUS:SETUP:DATA1:
CONDITION TRUE
:SEARCH1:SPIBUS:SETUP:DATA1:
CONDITION? ->
:SEARCH1:SPIBUS:SETUP:DATA1:
CONDITION TRUE

:SEARch<x>:SPIBus[:SETup]:DATA<x>:DPOSition

Function Sets the pattern comparison start position of the data of the SPI bus signal search or queries the current setting.

Syntax
```
:SEARch<x>:SPIBus[:SETup]:DATA<x>:
DPOSition {<NRf>}
:SEARch<x>:SPIBus[:SETup]:DATA<x>:
DPOSition?
```
<x> of SEARch<x> = 1 or 2
<x> of DATA<x> = 1 or 2
<NRf> = 0 to 9999

Example
```
:SEARCH1:SPIBUS:SETUP:DATA1:DPOSITION 1
:SEARCH1:SPIBUS:SETUP:DATA1:
DPOSITION? ->
:SEARCH1:SPIBUS:SETUP:DATA1:DPOSITION 1
```

:SEARch<x>:SPIBus[:SETup]:DATA<x>:HEXA<x>

Function Sets the data of the SPI bus signal search in hexadecimal notation.

Syntax
```
:SEARch<x>:SPIBus[:SETup]:DATA<x>:
HEXA<x> {<String>}
```
<x> of SEARch<x> = 1 or 2
<x> of DATA<x> = 1 or 2
<x> of HEXA<x> = 1 to 4
<String> = 2 characters by combining '0' to 'F' and 'X'

Example
```
:SEARCH1:SPIBUS:SETUP:DATA1:HEXA1 "EF"
```

:SEARch<x>:SPIBus[:SETup]:DATA<x>:PATTern<x>

Function Sets the data of the SPI bus signal search in binary notation or queries the current setting.

Syntax
```
:SEARch<x>:SPIBus[:SETup]:DATA<x>:
PATTern<x> {<String>}
:SEARch<x>:SPIBus[:SETup]:DATA<x>:
PATTern<x>?
```
<x> of SEARch<x> = 1 or 2
<x> of DATA<x> = 1 or 2
<x> of <PATTern x> = 1 to 4
<String> = 8 characters by combining '0,' '1,' and 'X'

Example
```
:SEARCH1:SPIBUS:SETUP:DATA1:
PATTERN1 "11101111"
:SEARCH1:SPIBUS:SETUP:DATA1:
PATTERN1? ->
:SEARCH1:SPIBUS:SETUP:DATA1:
PATTERN1 "11101111"
```

:SEARch<x>:SPIBus[:SETup]:DATA<x>:TRACe

Function Sets the source channel of the data of the SPI bus signal search or queries the current setting.

Syntax
```
:SEARch<x>:SPIBus[:SETup]:DATA<x>:
TRACe {<NRf>}
:SEARch<x>:SPIBus[:SETup]:DATA<x>:
TRACe?
```
<x> of SEARch<x> = 1 or 2
<x> of DATA<x> = 1 or 2
<NRf> = 1 to 8

Example
```
:SEARCH1:SPIBUS:SETUP:DATA1:TRACE 1
:SEARCH1:SPIBUS:SETUP:DATA1:TRACE? ->
:SEARCH1:SPIBUS:SETUP:DATA1:TRACE 1
```

:SEARch<x>:SPIBus[:SETup]:MODE

Function Sets the wiring system of the SPI bus signal search (three-wire or four-wire) or queries the current setting.

Syntax
```
:SEARch<x>:SPIBus[:SETup]:
MODE {WIRE3|WIRE4}
:SEARch<x>:SPIBus[:SETup]:MODE?
```
<x> = 1 or 2

Example
```
:SEARCH1:SPIBUS:SETUP:MODE WIRE3
:SEARCH1:SPIBUS:SETUP:MODE? ->
:SEARCH1:SPIBUS:SETUP:MODE WIRE3
```

:SEARch<x>:TRACe<x>:LEVel

Function Sets the threshold level of the trace or queries the current setting.

Syntax
```
:SEARch<x>:TRACe<x>:LEVel
{<NRf>|<Voltage>|<Current>}
:SEARch<x>:TRACe<x>:LEVel?
```
<x> of SEARch<x> = 1 or 2
<x> of TRACe<x> = 1 to 8
<NRf>, <Voltage>, and <Current> = See the *User's Manual (IM701310-01E).*

Example
```
:SEARCH1:TRACE1:LEVEL 0
:SEARCH1:TRACE1:LEVEL? ->
:SEARCH1:TRACE1:LEVEL 0.000E+00
```

Description This command applies to the channel corresponding to the source specified by the following commands.
- :SEARch<x>:I2CBus:CLOCk:SOURce
- :SEARch<x>:STRace
- :SEARch<x>:SPIBus:CLOCk:SOURce
- :SEARch<x>:SPIBus:CS:TRACe
- :SEARch<x>:SPIBus:DATA[1-2]:TRACe

:SEARch<x>:TYPE

Function Sets the search type or queries the current setting.

Syntax
```
:SEARch<x>:TYPE {CANBus|EDGE|EQUalify|
I2CBus|LINBus|SPATtern|SPIBus|STATe|
WIDTh}
:SEARch<x>:TYPE?
```
<x> = 1 or 2

Example
```
:SEARCH1:TYPE CANBus
:SEARCH1:TYPE? -> :SEARCH1:TYPE CANBus
```

5.5 TRIGger Group

:TRIGger:EINTerval:EVENt<x>:CANBus?

Function Queries all settings related to the CAN bus trigger of the event.

Syntax :TRIGger:EINTerval:EVENt<x>:CANBus?
<x> = 1, 2

Example :TRIGGER:EINTERVAL:EVENT1:CANBUS? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
ACK DONTCARE;BRATE 1000000;DATA:
BORDER BIG;CONDITION DONTCARE;
DATA1 0.0000000E+00;
DATA2 255.00000E+00;DLC 8;MSBLSB 7,0;
PATTERN "1110010101100100011110001001001
1001010100010000100011111111111010";
SIGNUNSIGN;:TRIGGER:EINTERVAL:EVENT1:
CANBUS:IDEXT:PATTERN "XXXXXXXXXXXXXXXXX
XXXXXXXXXXXX";:TRIGGER:EINTERVAL:
EVENT1:CANBUS:IDOR:ID1:ACK DONTCARE;
DATA:BORDER BIG;CONDITION DONTCARE;
DATA1 0.0000000E+00;
DATA2 255.00000E+00;DLC 8;MSBLSB 7,0;
PATTERN "0000000100100011010001010110001
1110001001101010111100110111101111";
SIGN UNSIGN;:TRIGGER:EINTERVAL:EVENT1:
CANBUS:IDOR:ID1:FORMAT STD;IDEXT:
PATTERN "11010101111001101111011110000";:
TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:IDSTD:PATTERN "00100100011";:
TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:MODE 0;RTR DATA;:TRIGGER:EINTERVAL:
EVENT1:CANBUS:IDOR:ID2:ACK DONTCARE;
DATA:BORDER BIG;CONDITION DONTCARE;
DATA1 0.0000000E+00;
DATA2 255.00000E+00;DLC 8;MSBLSB 7,0;
PATTERN "1111111011011100101110101001100
0001110110010101000011001000010000";
SIGN UNSIGN;:TRIGGER:EINTERVAL:EVENT1:
CANBUS:IDOR:ID2:FORMAT STD;IDEXT:
PATTERN "10010001101000101011001111000";:
TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID2:IDSTD:PATTERN "10001010110";:
TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID2:MODE 0.....

:TRIGger:EINTerval:EVENt<x>:CANBus:ACK

Function Sets the ACK condition of the CAN bus signal trigger or queries the current setting.

Syntax :TRIGger:EINTerval:EVENt<x>:CANBus:
ACK {ACK|ACKBoth|DONTcare|NONack}
:TRIGger:EINTerval:EVENt<x>:CANBus:ACK?
<x> = 1, 2

Example :TRIGGER:EINTERVAL:EVENT1:CANBUS:
ACK ACK
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
ACK? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
ACK ACK

:TRIGger:EINTerval:EVENt<x>:CANBus:BRATe

Function Sets the bit rate (data transfer rate) of the CAN bus signal trigger or queries the current setting.

Syntax :TRIGger:EINTerval:EVENt<x>:CANBus:
BRATe {<NRf>|USER,<NRf>}
:TRIGger:EINTerval:EVENt<x>:CANBus:
BRATe?
<x> = 1, 2
<NRf> = 33300, 83300, 125000, 250000, 500000, 1000000
<NRf> of USER = See the *User's Manual (IM701310-01E)*.

Example :TRIGGER:EINTERVAL:EVENT1:CANBUS:
BRATE 83300
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
BRATE? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
BRATE 83300

:TRIGger:EINTerval:EVENt<x>:CANBus:DATA?

Function Queries all settings related to the CAN bus signal trigger data.

Syntax :TRIGger:EINTerval:EVENt<x>:CANBus:
DATA?
<x> = 1, 2

Example :TRIGGER:EINTERVAL:EVENT1:CANBUS:
DATA? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:
BORDER BIG;CONDITION DONTCARE;
DATA1 0.0000000E+00;
DATA2 255.00000E+00;DLC 8;MSBLSB 7,0;
PATTERN "1110010101100100011110001001001
1001010100010000100011111111111010";
SIGN UNSIGN

:TRIGger:EINTerval:EVENt<x>:CANBus:DATA:BORDer

Function Sets the byte order of the CAN bus signal trigger data or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
DATA:BORDer {BIG|LITTle}
:TRIGger:EINTerval:EVENt<x>:CANBus:
DATA:BORDer?
```
<x> = 1, 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:
BORDER BIG
:TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:
BORDER? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:
BORDER BIG
```

:TRIGger:EINTerval:EVENt<x>:CANBus:DATA:CONDition

Function Sets the data condition of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
DATA:CONDition {BETWeen|DONTcare|FALSe|
GTHan|LTHan|ORANge|TRUE}
:TRIGger:EINTerval:EVENt<x>:CANBus:
DATA:CONDition?
```
<x> = 1, 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:
CONDITION BETWEEN
:TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:
CONDITION? ->
TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:
CONDITION BETWEEN
```

:TRIGger:EINTerval:EVENt<x>:CANBus:DATA:DATA<x>

Function Sets the comparison data of the CAN bus signal trigger data or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
DATA:DATA<x> {<NRf>}
:TRIGger:EINTerval:EVENt<x>:CANBus:
DATA:DATA<x>?
```
<x> of EVENt<x> = 1 or 2
<x> of DATA<x> = 1 or 2
<NRf> = See the *User's Manual (IM701310-01E).*

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
DATA:DATA1 1
:TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:
DATA1? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:
DATA1 1.0000000E+00
```

Description
- Use `:TRIGger:EINTerval:EVENt<x>:CANBus:DATA:DATA1` when `:TRIGger:EINTerval:EVENt<x>:CANBus:DATA:CONDition GTHan` is specified.
  Use `:TRIGger:EINTerval:EVENt<x>:CANBus:DATA:DATA2` when `:TRIGger:EINTerval:EVENt<x>:CANBus:DATA:CONDition LTHan` is specified.
- Use `:TRIGger:EINTerval:EVENt<x>:CANBus:DATA:DATA1` to set the smaller value and `:TRIGger:EINTerval:EVENt<x>:CANBus:DATA:DATA2` to set the larger value when `:TRIGger:EINTerval:EVENt<x>:CANBus:DATA:CONDition BETWeen|ORANge` is specified.

:TRIGger:EINTerval:EVENt<x>:CANBus:DATA:DLC

Function Sets the number of valid bytes (DLC) of the CAN bus signal trigger data or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
DATA:DLC {<NRf>}
:TRIGger:EINTerval:EVENt<x>:CANBus:
DATA:DLC?
```
<x> = 1, 2
<NRf> = 0 to 8

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:
DLC 0
:TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:
DLC? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:
DLC 0
```

:TRIGger:EINTerval:EVENt<x>:CANBus:DATA:HEXA

Function Sets the CAN bus signal trigger data in hexadecimal notation.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
DATA:HEXA {<String>}
```
<x> = 1, 2
<String> = Up to 16 characters by combining '0' to 'F' and 'X' (in one-byte unit)

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
DATA:HEXA "A9"
```

:TRIGger:EINTerval:EVENt<x>:CANBus:DATA:MSBLsb

Function Sets the MSB and LSB bits of the CAN bus signal trigger data or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
DATA:MSBLsb {<NRf>,<NRf>}
:TRIGger:EINTerval:EVENt<x>:CANBus:
DATA:MSBLsb?
```
<x> = 1, 2
<NRf> = See the *User's Manual (IM701310-01E).*

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:
MSBLSB 1,0
:TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:
MSBLSB? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:
MSBLSB 1,0
```

:TRIGger:EINTerval:EVENt<x>:CANBus:DATA:PATTern

Function Sets the CAN bus signal trigger data in binary notation or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
DATA:PATTern {<String>}
:TRIGger:EINTerval:EVENt<x>:CANBus:
DATA:PATTern?
```
<x> = 1, 2
<String> = Up to 64 characters by combining '0,' '1,' and 'X' (in one-byte unit)

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:
PATTERN "11011111"
:TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:
PATTERN? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:
PATTERN "11011111"
```

:TRIGger:EINTerval:EVENt<x>:CANBus:DATA:SIGN

Function Sets the sign of the CAN bus signal trigger data or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
DATA:SIGN {SIGN|UNSign}
:TRIGger:EINTerval:EVENt<x>:CANBus:
DATA:SIGN?
```
<x> = 1, 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:
SIGN SIGN
:TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:
SIGN? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:DATA:
SIGN SIGN
```

:TRIGger:EINTerval:EVENt<x>:CANBus:IDEXt?

Function Queries all settings related to the ID of the extended format of the CAN bus signal trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDEXt?
```
<x> = 1, 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
IDEXT? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDEXT:
PATTERN "11001011011100001110111011111"
```

:TRIGger:EINTerval:EVENt<x>:CANBus:IDEXt:HEXA

Function Sets the ID of the extended format of the CAN bus signal trigger in hexadecimal notation.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDEXt:
HEXA {<String>}
```
<x> = 1, 2
<String> = 8 characters by combining '0' to 'F' and 'X'

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDEXT:
HEXA "1AEF5906"
```

:TRIGger:EINTerval:EVENt<x>:CANBus:IDEXt:PATTern

Function Sets the ID of the extended format of the CAN bus signal trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDEXt:PATTern {<String>}
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDEXt:PATTern?
```
<x> = 1, 2
<String> = 29 characters by combining '0,' '1,' and 'X'

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDEXT:
PATTERN "11001011011100001110111011111"
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDEXT:
PATTERN? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDEXT:
PATTERN "11001011011100001110111011111"
```

:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR?

Function Queries all settings related to the OR condition of the CAN bus signal trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR?
```
<x> = 1, 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
IDOR? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:ACK DONTCARE;DATA:BORDER BIG;
CONDITION DONTCARE;DATA1 0.0000000E+00;
DATA2 255.00000E+00;DLC 8;MSBLSB 7,0;
PATTERN "0000000100100011010001010110011
1100010011010101111001101111101111";
SIGN UNSIGN;:TRIGGER:EINTERVAL:EVENT1:
CANBUS:IDOR:ID1:FORMAT STD;IDEXT:
PATTERN "11010101111001101111011110000";:
TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:IDSTD:PATTERN "00100100011";:
TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:MODE 0;RTR DATA;:TRIGGER:EINTERVAL:
EVENT1:CANBUS:IDOR:ID2:ACK DONTCARE;
DATA:BORDER BIG;CONDITION DONTCARE;
DATA1 0.0000000E+00;
DATA2 255.00000E+00;DLC 8;MSBLSB 7,0;
PATTERN "1111111011011100101110101001110
0001110110010101000011001000010000";
SIGN UNSIGN;:TRIGGER:EINTERVAL:EVENT1:
CANBUS:IDOR:ID2:FORMAT STD;IDEXT:
PATTERN"10010001101000101011001111000";:
TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID2:IDSTD:PATTERN "10001010110";:
TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID2:MODE 0;RTR DATA;:TRIGGER:EINTERVAL:
EVENT1:CANBUS:IDOR:ID3:ACK DONTCARE;
DATA:BORDER BIG;CONDITION DONTCARE;
DATA1 0.0000000E+00;
DATA2 255.00000E+00;DLC 8.....
```

:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>?

Function Queries all settings related to each ID of the OR condition of the CAN bus signal trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>?
```
<x> of EVENt<x> = 1 or 2
<x> of ID<x> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
IDOR:ID1? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
IDOR:ID1:ACK DONTCARE;DATA:
BORDER BIG;CONDITION DONTCARE;
DATA1 0.0000000E+00;
DATA2 255.00000E+00;DLC 8;MSBLSB 7,0;
PATTERN "000000010010001101000101011O
0111100010011010101111001101111101111";
SIGN UNSIGN;:TRIGGER:EINTERVAL:
EVENT1:CANBUS:IDOR:ID1:FORMAT STD;
IDEXT:PATTERN "110101011110011011110
11110000";:TRIGGER:EINTERVAL:EVENT1:
CANBUS:IDOR:ID1:IDSTD:
PATTERN"00100100011";:TRIGGER:
EINTERVAL:EVENT1:CANBUS:IDOR:ID1:
MODE 0;RTR DATA
```

:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:ACK

Function Sets each ACK condition of the OR condition of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:ACK {ACK|ACKBoth|
DONTcare|NONack}
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:ACK?
```
<x> of EVENt<x> = 1 or 2
<x> of ID<x> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
IDOR:ID1:ACK ACK
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
IDOR:ID1:ACK? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
IDOR:ID1:ACK ACK
```

:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:DATA?

Function Queries all settings related to each data of the OR condition of the CAN bus signal trigger.

Syntax `:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:DATA?`
<x> of EVENt<x> = 1 or 2
<x> of ID<x> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:DATA? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:DATA:BORDER BIG;CONDITION DONTCARE;
DATA1 0.0000000E+00;
DATA2 255.00000E+00;DLC 8;MSBLSB 7,0;
PATTERN "0000000100100011010001010110011
1100010011010101111001101111101111";
SIGN UNSIGN
```

:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:DATA:BORDer

Function Sets byte order of each data of the OR condition of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:DATA:BORDer {BIG|LITTle}
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:DATA:BORDer?
```
<x> of EVENt<x> = 1 or 2
<x> of ID<x> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:DATA:BORDER BIG
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:DATA:BORDER? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:DATA:BORDER BIG
```

:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:DATA:CONDition

Function Sets each data condition of the OR condition of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:DATA:CONDition {BETWeen|
DONTcare|FALSe|GTHan|LTHan|ORANge|TRUE}
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:DATA:CONDition?
```
<x> of EVENt<x> = 1 or 2
<x> of ID<x> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:DATA:CONDITION BETWEEN
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:DATA:CONDITION? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:DATA:CONDITION BETWEEN
```

:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:DATA:DATA<x>

Function Sets comparison data of each data of the OR condition of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:DATA:DATA<x> {<NRf>}
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:DATA:DATA<x>?
```
<x> of EVENt<x> = 1 or 2
<x> of ID<x> = 1 to 4
<x> of DATA<x> = 1 or 2
<NRf> = See the *User's Manual (IM701310-01E)*.

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:DATA:DATA1 1
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:DATA:DATA1? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:DATA:DATA1 1.0000000E+00
```

Description
- Use `:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:DATA:DATA1` when `:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:DATA:CONDition GTHan` is specified.
- Use `:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:DATA:DATA2` when `:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:DATA:CONDition LTHan` is specified.
- Use `:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:DATA:DATA1` to set the smaller value and `:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:DATA:DATA2` to set the larger value when `:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:DATA:CONDition BETWeen|ORANge` is specified.

:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:DATA:DLC

Function Sets the number of valid bytes (DLC) of each data of the OR condition of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:DATA:DLC {<NRf>}
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:DATA:DLC?
```
<x> of EVENt<x> = 1 or 2
<x> of ID<x> = 1 to 4
<NRf> = 0 to 8

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:DATA:DLC 0
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:DATA:DLC? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:DATA:DLC 0
```

:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:DATA:HEXA

Function Sets each data of the OR condition of the CAN bus signal trigger in hexadecimal notation.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:DATA:HEXA {<String>}
```
<x> of EVENt<x> = 1 or 2
<x> of ID<x> = 1 to 4
<String> = Up to 16 characters by combining '0' to 'F' and 'X' (in one-byte unit)

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:DATA:HEXA "A9"
```

:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:DATA:MSBLsb

Function Sets the MSB and LSB bits of each data of the OR condition of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:DATA:MSBLsb {<NRf>,<NRf>}
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:DATA:MSBLsb?
```
<x> of EVENt<x> = 1 or 2
<x> of ID<x> = 1 to 4
<NRf> = See the *User's Manual (IM701310-01E)*.

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:DATA:MSBLSB 1,0
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:DATA:MSBLSB? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:DATA:MSBLSB 1,0
```

:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:DATA:PATTern

Function Sets each data of the OR condition of the CAN bus signal trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:DATA:PATTern {<String>}
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:DATA:PATTern?
```
<x> of EVENt<x> = 1 or 2
<x> of ID<x> = 1 to 4
<String> = Up to 64 characters by combining '0,' '1,' and 'X' (in one-byte unit)

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:DATA:PATTERN "11011111"
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:DATA:PATTERN? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:DATA:PATTERN "11011111"
```

:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:DATA:SIGN

Function Sets sign of each data of the OR condition of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:DATA:SIGN {SIGN|UNSign}
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:DATA:SIGN?
```
<x> of EVENt<x> = 1 or 2
<x> of ID<x> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:DATA:SIGN SIGN
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:DATA:SIGN? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:DATA:SIGN SIGN
```

:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:FORMat

Function Sets each message format (standard or extended) of the OR condition of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:FORMat {STD|EXT}
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:FORMat?
```
<x> of EVENt<x> = 1 or 2
<x> of ID<x> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:FORMAT STD
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:FORMAT? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:FORMAT STD
```

:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:IDEXt?

Function Queries all settings related to the ID of each extended format of the OR condition of the CAN bus signal trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:IDEXt?
```
<x> of EVENt<x> = 1 or 2
<x> of ID<x> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:IDEXT? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:IDEXT:PATTERN "1100101101110000111
0111011111"
```

:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:IDEXt:HEXA

Function Sets the ID of each extended format of the OR condition of the CAN bus signal trigger in hexadecimal notation.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:IDEXt:HEXA {<String>}
```
<x> of EVENt<x> = 1 or 2
<x> of ID<x> = 1 to 4
<String> = 8 characters by combining '0' to 'F' and 'X'

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:IDEXT:HEXA "1AEF5906"
```

:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:IDEXt:PATTern

Function Sets the ID of each extended format of the OR condition of the CAN bus signal trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:IDEXt:PATTern {<String>}
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:IDEXt:PATTern?
```
<x> of EVENt<x> = 1 or 2
<x> of ID<x> = 1 to 4
<String> = 29 characters by combining '0,' '1,' and 'X'

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:IDEXT:PATTERN "11001011011100001110
111011111"
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:IDEXT:PATTERN? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:IDEXT:PATTERN "11001011011100001110
111011111"
```

:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:IDSTd?

Function Queries all settings related to the ID of each standard format of the OR condition of the CAN bus signal trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:IDSTd?
```
<x> of EVENt<x> = 1 or 2
<x> of ID<x> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
IDOR:ID1:IDSTD? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:IDSTD:PATTERN "00011111101"
```

:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:IDSTd:HEXA

Function Sets the ID of each standard format of the OR condition of the CAN bus signal trigger in hexadecimal notation.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:IDSTd:HEXA {<String>}
```
<x> of EVENt<x> = 1 or 2
<x> of ID<x> = 1 to 4
<String> = 3 characters by combining '0' to 'F' and 'X'

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:IDSTD:HEXA "5DF"
```

:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:IDSTd:PATTern

Function Sets the ID of each standard format of the OR condition of the CAN bus signal trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:IDSTd:PATTern {<String>}
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:IDSTd:PATTern?
```
<x> of EVENt<x> = 1 or 2
<x> of ID<x> = 1 to 4
<String> = 11 characters by combining '0,' '1,' and 'X'

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:IDSTD:PATTERN "10111011111"
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:IDSTD:PATTERN? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:IDSTD:PATTERN "10111011111"
```

:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:MODE

Function Enables or disables each condition of the OR condition of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:MODE {<Boolean>}
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:MODE?
```
<x> of EVENt<x> = 1 or 2
<x> of ID<x> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:MODE ON
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:MODE? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:MODE 1
```

:TRIGger:EINTerval:EVENt<x>:CANBus:IDOR:ID<x>:RTR

Function Sets each RTR of the OR condition of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:RTR {DATA|DONTcare|REMote}
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDOR:ID<x>:RTR?
```
<x> of EVENt<x> = 1 or 2
<x> of ID<x> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:RTR DATA
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:RTR? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDOR:
ID1:RTR DATA
```

:TRIGger:EINTerval:EVENt<x>:CANBus:IDSTd?

Function Queries all settings related to the ID of the standard format of the CAN bus signal trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDSTd?
```
<x> = 1, 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
IDSTD? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDSTD:
PATTERN "00011111101"
```

:TRIGger:EINTerval:EVENt<x>:CANBus:IDSTd:HEXA

Function Sets the ID of the standard format of the CAN bus signal trigger in hexadecimal notation.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDSTd:HEXA {<String>}
```
<x> = 1, 2
<String> = 8 characters by combining '0' to 'F' and 'X'

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDSTD:
HEXA "5DF"
```

:TRIGger:EINTerval:EVENt<x>:CANBus:IDSTd:PATTern

Function Sets the ID of the standard format of the CAN bus signal trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDSTd:PATTern {<String>}
:TRIGger:EINTerval:EVENt<x>:CANBus:
IDSTd:PATTern?
```
<x> = 1, 2
<String> = 11 characters by combining '0,' '1,' and 'X'

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDSTD:
PATTERN "10111011111"
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDSTD:
PATTERN? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:IDSTD:
PATTERN "10111011111"
```

:TRIGger:EINTerval:EVENt<x>:CANBus:MODE

Function Sets the CAN bus signal trigger mode or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
MODE {EFRame|IDEXt|IDOR|IDSTd|SOF}
:TRIGger:EINTerval:EVENt<x>:CANBus:MODE?
```
<x> = 1, 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
MODE EFRAME
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
MODE? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
MODE EFRAME
```

:TRIGger:EINTerval:EVENt<x>:CANBus:RECessive

Function Sets the recessive level (bus level) of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
RECessive {HIGH|LOW}
:TRIGger:EINTerval:EVENt<x>:CANBus:
RECessive?
```
<x> = 1, 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
RECESSIVE HIGH
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
RECESSIVE? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
RECESSIVE HIGH
```

:TRIGger:EINTerval:EVENt<x>:CANBus:RTR

Function Sets the RTR of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
RTR {DATA|DONTcare|REMote}
:TRIGger:EINTerval:EVENt<x>:CANBus:RTR?
```
<x> = 1, 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
RTR DATA
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
RTR? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
RTR DATA
```

:TRIGger:EINTerval:EVENt<x>:CANBus:SOURce

Function Sets the trigger source of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
SOURce {<NRf>}
:TRIGger:EINTerval:EVENt<x>:CANBus:
SOURce?
```
<x> = 1, 2
<NRf> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
SOURCE 1
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
SOURCE? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
SOURCE 1
```

:TRIGger:EINTerval:EVENt<x>:CANBus:SPOint

Function Sets the sample point of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:CANBus:
SPOint {<NRf>}
:TRIGger:EINTerval:EVENt<x>:CANBus:
SPOint?
```
<x> = 1, 2
<NRf> = 18.8 to 90.6(%)

Example
```
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
SPOINT 18.8
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
SPOINT? ->
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
SPOINT 18.8E+00
```

:TRIGger:EINTerval:EVENt<x>:I2CBus?

Function Queries all settings related to the I^2C bus signal trigger of the event.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT10ADDRESS:PATTERN "10111011111";:
TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA
:BIT7ADDRESS:PATTERN "11011110";:
TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7APSUB:ADDRESS:PATTERN "10101011";:
TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7APSUB:SADDRESS:PATTERN "10101011";:
TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
TYPE BIT10ADDRESS;:TRIGGER:EINTERVAL:
EVENT1:I2CBUS:CLOCK:SOURCE 1;:TRIGGER:
EINTERVAL:EVENT1:I2CBUS:DATA:BYTE 1;
CONDITION TRUE;DPOSITION 1;MODE 1;
PATTERN1 "10101011";
PATTERN2 "10101010";
PATTERN3 "10101111";
PATTERN4 "10101011";PMODE DONTCARE;
SOURCE 1;:TRIGGER:EINTERVAL:EVENT1:
I2CBUS:GCALL:BIT7MADDRESS:
PATTERN "1010101";:TRIGGER:EINTERVAL:
EVENT1:I2CBUS:GCALL:
SBYTE BIT7MADDRESS;:TRIGGER:EINTERVAL:
EVENT1:I2CBUS:MODE ADATA;NAIGNORE:
HSMODE 1;RACCESS 1;SBYTE 1;:TRIGGER:
EINTERVAL:EVENT1:I2CBUS:SBHSMODE:
TYPE HSMODE
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa?

Function Queries all settings related to the address of the I^2C bus signal trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
ADATa?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA?
-> :TRIGGER:EINTERVAL:EVENT1:I2CBUS:
ADATA:BIT10ADDRESS:
PATTERN "10111011111";:TRIGGER:
EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT7ADDRESS:
PATTERN "11011110";:TRIGGER:EINTERVAL:
EVENT1:I2CBUS:ADATA:BIT7APSUB:ADDRESS:
PATTERN "10101011";:TRIGGER:EINTERVAL:
EVENT1:I2CBUS:ADATA:BIT7APSUB:SADDRESS:
PATTERN "10101011";:TRIGGER:EINTERVAL:
EVENT1:I2CBUS:ADATA:TYPE BIT10ADDRESS
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT10address?

Function Queries all settings related to the 10-bit address of the I^2C bus signal trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
ADATa:BIT10address?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT10ADDRESS? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT10ADDRESS:PATTERN "10111011111"
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT10address:HEXA

Function Sets the 10-bit address of the I^2C bus signal trigger in hexadecimal notation.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
ADATa:BIT10address:HEXA {<String>}
```
<x> = 1 or 2
<String> = 3 characters by combining '0' to 'F' and 'X' (bit 8 is the R/$\overline{W}$ bit)

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:
BIT10ADDRESS:HEXA "7AB"
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT10address:PATTern

Function Sets the 10-bit address of the I^2C bus signal trigger in binary notation or queries the current setting.

Syntax :TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT10address:PATTern {<String>}
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT10address:PATTern?
<x> = 1 or 2
<String> = 11 characters by combining '0', '1', and 'X' (bit 8 is the $R/\overline{W}$ bit)

Example :TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:BIT10ADDRESS:PATTERN "10111011111"
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:BIT10ADDRESS:PATTERN? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:BIT10ADDRESS:PATTERN "10111011111"

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7ADdress?

Function Queries all settings related to the 7-bit address of the I^2C bus signal trigger.

Syntax :TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7ADdress?
<x> = 1 or 2

Example :TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:BIT7ADDRESS? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:BIT7ADDRESS:PATTERN "11011110"

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7ADdress:HEXA

Function Sets the 7-bit address of the I^2C bus signal trigger in hexadecimal notation.

Syntax :TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7ADdress:HEXA {<String>}
<x> = 1 or 2
<String> = 2 characters by combining '0' to 'F' and 'X' (bit 0 is the $R/\overline{W}$ bit)

Example :TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:BIT7ADDRESS:HEXA "DE"

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7ADdress:PATTern

Function Sets the 7-bit address of the I^2C bus signal trigger in binary notation or queries the current setting.

Syntax :TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7ADdress:PATTern {<String>}
:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7ADdress:PATTern?
<x> = 1 or 2
<String> = 8 characters by combining '0', '1', and 'X' (bit 0 is the $R/\overline{W}$ bit)

Example :TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:BIT7ADDRESS:PATTERN "11011110"
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:BIT7ADDRESS:PATTERN? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:BIT7ADDRESS:PATTERN "11011110"

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub?

Function Queries all settings related to the 7-bit + Sub address of the I^2C bus signal trigger.

Syntax :TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub?
<x> = 1 or 2

Example :TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB:ADDRESS:PATTERN "10101011";:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB:SADDRESS:PATTERN "10101011"

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:ADDRess?

Function Queries all settings related to the 7-bit address of the 7-bit + Sub address of the I^2C bus signal trigger.

Syntax :TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:ADDRess?
<x> = 1 or 2

Example :TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB:ADDRESS? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB:ADDRESS:PATTERN "10101011"

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:ADDRess:HEXA

Function Sets the 7-bit address of the 7-bit + Sub address of the I²C bus signal trigger in hexadecimal notation.

Syntax `:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:ADDRess:HEXA {<String>}`
<x> = 1 or 2
<String> = 2 characters by combining '0' to 'F' and 'X' (bit 0 is the R/$\overline{W}$ bit)

Example `:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB:ADDRESS:HEXA "AB"`

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:ADDRess:PATTern

Function Sets the 7-bit address of the 7-bit + Sub address of the I²C bus signal trigger in binary notation or queries the current setting.

Syntax `:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:ADDRess:PATTern {<String>}`
`:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:ADDRess:PATTern?`
<x> = 1 or 2
<String> = 8 characters by combining '0', '1', and 'X' (bit 0 is the R/$\overline{W}$ bit)

Example `:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB:ADDRESS:PATTERN "10101011"`
`:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB:ADDRESS:PATTERN? -> :TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB:ADDRESS:PATTERN "10101011"`

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:SADDress?

Function Queries all settings related to the Sub address of the 7-bit + Sub address of the I²C bus signal trigger.

Syntax `:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:SADDress?`
<x> = 1 or 2

Example `:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB:SADDRESS? -> :TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB:SADDRESS:PATTERN "10101011"`

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:SADDress:HEXA

Function Sets the Sub address of the 7-bit + Sub address of the I²C bus signal trigger in hexadecimal notation.

Syntax `:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:SADDress:HEXA {<String>}`
<x> = 1 or 2
<String> = 2 characters by combining '0' to 'F' and 'X'

Example `:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB:SADDRESS:HEXA "EF"`

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:SADDress:PATTern

Function Sets the Sub address of the 7-bit + Sub address of the I²C bus signal trigger in binary notation or queries the current setting.

Syntax `:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:SADDress:PATTern {<String>}`
`:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:BIT7APsub:SADDress:PATTern?`
<x> = 1 or 2
<String> = 8 characters by combining '0', '1', and 'X'

Example `:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB:SADDRESS:PATTERN "10101011"`
`:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB:SADDRESS:PATTERN? -> :TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:BIT7APSUB:SADDRESS:PATTERN "10101011"`

:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:TYPE

Function Sets the address type of the I²C bus signal trigger or queries the current setting.

Syntax `:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:TYPE {BIT10address|BIT7ADdress|BIT7APsub}`
`:TRIGger:EINTerval:EVENt<x>:I2CBus:ADATa:TYPE?`
<x> = 1 or 2

Example `:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:TYPE BIT10ADDRESS`
`:TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:TYPE? -> :TRIGGER:EINTERVAL:EVENT1:I2CBUS:ADATA:TYPE BIT10ADDRESS`

:TRIGger:EINTerval:EVENt<x>:I2CBus:CLOCk?

Function Queries all settings related to the clock channel of the I²C bus signal trigger.

Syntax `:TRIGger:EINTerval:EVENt<x>:I2CBus:CLOCk?`
<x> = 1 or 2

Example `:TRIGGER:EINTERVAL:EVENT1:I2CBUS:CLOCK? -> :TRIGGER:EINTERVAL:EVENT1:I2CBUS:CLOCK:SOURCE 1`

:TRIGger:EINTerval:EVENt<x>:I2CBus:CLOCk:SOURce

Function Sets the clock channel of the I^2C bus signal trigger or queries the current setting.

Syntax:
```
TRIGger:EINTerval:EVENt<x>:I2CBus:
CLOCk:SOURce {<NRf>}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
CLOCk:SOURce?
```
<x> = 1 or 2
<NRf> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:CLOCK:
SOURCE 1
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:CLOCK:
SOURCE? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:CLOCK:
SOURCE 1
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA?

Function Queries all settings related to the data of the I^2C bus signal trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
DATA?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
DATA? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
BYTE 1;CONDITION TRUE;DPOSITION 1;
MODE 1;PATTERN1 "10101011";
PATTERN2 "10101010";
PATTERN3 "10101111";
PATTERN4 "10101011";PMODE DONTCARE;
SOURCE 1
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:BYTE

Function Sets the number of data bytes of the I^2C bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
DATA:BYTE {<NRf>}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
DATA:BYTE?
```
<x> = 1 or 2
<NRf> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
BYTE 1
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
BYTE? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
BYTE 1
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:CONDition

Function Sets the determination method (match or not match) of the data of the I^2C bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
DATA:CONDition {FALSe|TRUE}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
DATA:CONDition?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
CONDITION TRUE
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
CONDITION? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
CONDITION TRUE
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:DPOSition

Function Sets the position for comparing the data pattern of the I^2C bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
DATA:DPOSition {<NRf>}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
DATA:DPOSition?
```
<x> = 1 or 2
<NRf> = 0 to 9999

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
DPOSITION 1
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
DPOSITION? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
DPOSITION 1
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:HEXA<x>

Function Sets the data of the I^2C bus signal trigger in hexadecimal notation.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
DATA:HEXA<x> {<String>}
```
<x> of EVENt<x> = 1 or 2
<x> of HEXA<x> = 1 to 4
<String> = 2 characters by combining '0' to 'F' and 'X'

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
HEXA1 "AB"
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:MODE

Function Enables/Disables the data conditions of the I^2C bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
DATA:MODE {<Boolean>}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
DATA:MODE?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
MODE ON
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
MODE? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
MODE 1
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:PATTern<x>

Function Sets the data of the I^2C bus signal trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
DATA:PATTern<x> {<String>}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
DATA:PATTern<x>?
```
<x> of EVENt<x> = 1 or 2
<x> of <PATTern x> = 1 to 4
<String> = 8 characters by combining ‘0’, ‘1’, and ‘X’

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
PATTERN1 "10101011"
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
PATTERN1? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
PATTERN1 "10101011"
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:PMODe

Function Sets the pattern comparison start position mode of the data of the I^2C bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
DATA:PMODe {DONTcare|SELect}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
DATA:PMODe?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
PMODE SELECT
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
PMODE? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
PMODE SELECT
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:DATA:SOURce

Function Sets the data channel of the I^2C bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
DATA:SOURce {<NRf>}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
DATA:SOURce?
```
<x> = 1 or 2
<NRf> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
SOURCE 1
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
SOURCE? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:DATA:
SOURCE 1
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:GCALl?

Function Queries all settings related to the general call of the I^2C bus signal trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
GCALl?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
GCALL? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:GCALL:
BIT7MADDRESS:PATTERN "1010101";:
TRIGGER:EINTERVAL:EVENT1:I2CBUS:GCALL:
SBYTE BIT7MADDRESS
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:GCALl:BIT7maddress?

Function Queries all settings related to the 7-bit master address of the general call of the I^2C bus signal trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
GCALl:BIT7maddress?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:GCALL:
BIT7MADDRESS? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:GCALL:
BIT7MADDRESS:PATTERN "1010101"
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:GCALl:BIT7maddress:HEXA

Function Sets the 7-bit master address of the general call of the I^2C bus signal trigger in hexadecimal notation.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
GCALl:BIT7maddress:HEXA {<String>}
```
<x> = 1 or 2
<String> = 2 characters by combining ‘0’ to ‘F’ and ‘X’ (bit 0 is fixed to 1)

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:GCALL:
BIT7MADDRESS:HEXA "AB"
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:GCALl:BIT7maddress:PATTern

Function Sets the 7-bit master address of the general call of the I^2C bus signal trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
GCALl:BIT7maddress:PATTern {<String>}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
GCALl:BIT7maddress:PATTern?
```
<x> = 1 or 2
<String> = 7 characters by combining ‘0’, ‘1’, and ‘X’

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
GCALL:BIT7MADDRESS:PATTERN “1010101”
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:GCALL:
BIT7MADDRESS:PATTERN? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:GCALL:
BIT7MADDRESS:PATTERN “1010101”
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:GCALl:SBYTe (Second Byte)

Function Sets the second byte type of the general call of the I^2C bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
GCALl:SBYTe {BIT7maddress|DONTcare|H04|
H06}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
GCALl:SBYTe?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:GCALL:
SBYTE BIT7MADDRESS
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:GCALL:
SBYTE? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:GCALL:
SBYTE BIT7MADDRESS
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:MODE

Function Sets the trigger mode of the I^2C bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
MODE {ADATa|ESTart|GCALl|NAIGnore|
SBHSmode}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
MODE?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
MODE ADATA
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
MODE? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
MODE ADATA
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:NAIGnore?

Function Queries all settings related to the NON ACK ignore mode of the I^2C bus signal trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
NAIGnore?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
NAIGNORE? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
NAIGNORE:HSMODE 1;RACCESS 1;SBYTE 1
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:NAIGnore:HSMode

Function Sets whether to ignore NON ACK in high speed mode of the I^2C bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
NAIGnore:HSMode {<Boolean>}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
NAIGnore:HSMode?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
NAIGNORE:HSMODE ON
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
NAIGNORE:HSMODE? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
NAIGNORE:HSMODE 1
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:NAIGnore:RACCess

Function Sets whether to ignore NON ACK in read access mode of the I^2C bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
NAIGnore:RACCess {<Boolean>}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
NAIGnore:RACCess?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
NAIGNORE:RACCESS ON
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
NAIGNORE:RACCESS? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
NAIGNORE:RACCESS 1
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:NAIGnore:SBYTe (Start Byte)

Function Sets whether to ignore NON ACK in the start byte of the I²C bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
NAIGnore:SBYTe {<Boolean>}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
NAIGnore:SBYTe?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
NAIGNORE:SBYTE ON
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
NAIGNORE:SBYTE? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
NAIGNORE:SBYTE 1
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:SBHSmode?

Function Queries all settings related to the start byte and high speed mode of the I²C bus signal trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
SBHSmode?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
SBHSMODE? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
SBHSMODE:TYPE HSMODE
```

:TRIGger:EINTerval:EVENt<x>:I2CBus:SBHSmode:TYPE

Function Sets the type of the start byte or high speed mode of the I²C bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:I2CBus:
SBHSmode:TYPE {HSMode|SBYTe}
:TRIGger:EINTerval:EVENt<x>:I2CBus:
SBHSmode:TYPE?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
SBHSMODE:TYPE HSMODE
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
SBHSMODE:TYPE? ->
:TRIGGER:EINTERVAL:EVENT1:I2CBUS:
SBHSMODE:TYPE HSMODE
```

:TRIGger:EINTerval:EVENt<x>:LINBus?

Function Queries all settings related to LIN bus signal triggers of each event.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:LINBus?
```
<x>=1, 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:LINBUS? ->
:TRIGGER:EINTERVAL:EVENT1:LINBUS:
BRATE 19200;SOURCE 1
```

:TRIGger:EINTerval:EVENt<x>:LINBus:BRATe

Function Sets the LIN bus signal trigger bitrate (data transfer rate) or queries the current setting.

Syntax
```
TRIGger:EINTerval:EVENt<x>:LINBus:
BRATe {<NRf>|USER,<NRf>}
:TRIGger:EINTerval:EVENt<x>:LINBus:
BRATe?
```
<x>=1, 2
<NRf>=1200, 2400, 4800, 9600, 19200
USER <NRf>=See this *User's Manual* .

Example
```
:TRIGGER:EINTERVAL:EVENT1:LINBUS:
BRATE 19200
:TRIGGER:EINTERVAL:EVENT1:LINBUS:BRATE?
-> :TRIGGER:EINTERVAL:EVENT1:LINBUS:
BRATE 19200
```

:TRIGger:EINTerval:EVENt<x>:LINBus:SOURce

Function Sets the LIN bus signal trigger source or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:LINBus:
SOURce {<NRf>}
:TRIGger:EINTerval:EVENt<x>:LINBus:
SOURce?
```
<x>=1, 2
<NRf>=1–4

Example
```
:TRIGGER:EINTERVAL:EVENT1:LINBUS:
SOURCE 1
:TRIGGER:EINTERVAL:EVENT1:LINBUS:
SOURCE? -> :TRIGGER:EINTERVAL:EVENT1:
LINBUS:SOURCE 1
```

:TRIGger:EINTerval:EVENt<x>:SPIBus?

Function Queries all settings related to the SPI bus signal trigger of the event.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPIBus?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPIBUS? ->
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:
BITORDER LSBFIRST;CLOCK:POLARITY FALL;
SOURCE 1;:TRIGGER:EINTERVAL:EVENT1:
SPIBUS:CS:ACTIVE HIGH;SOURCE 1;:
TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:
BYTE 1;CONDITION TRUE;DPOSITION 1;
PATTERN1 "00010010";
PATTERN2 "00110100";
PATTERN3 "01010110";
PATTERN4 "00010010";SOURCE 3;:TRIGGER:
EINTERVAL:EVENT1:SPIBUS:DATA2:BYTE 4;
CONDITION TRUE;DPOSITION 1;
PATTERN1 "00010010";
PATTERN2 "00110100";
PATTERN3 "01010110";
PATTERN4 "00010010";SOURCE 3;:TRIGGER:
EINTERVAL:EVENT1:SPIBUS:MODE WIRE3
```

:TRIGger:EINTerval:EVENt<x>:SPIBus: BITorder

Function Sets the bit order of the SPI bus signal trigger or queries the current setting.

Syntax :TRIGger:EINTerval:EVENt<x>:SPIBus: BITorder {LSBFirst|MSBFirst}
:TRIGger:EINTerval:EVENt<x>:SPIBus: BITorder?
<x> = 1 or 2

Example :TRIGGER:EINTERVAL:EVENT1:SPIBUS: BITORDER LSBFIRST
:TRIGGER:EINTERVAL:EVENT1:SPIBUS: BITORDER? ->
:TRIGGER:EINTERVAL:EVENT1:SPIBUS: BITORDER LSBFIRST

:TRIGger:EINTerval:EVENt<x>:SPIBus: CLOCk?

Function Queries all settings related to the clock channel of the SPI bus signal trigger.

Syntax :TRIGger:EINTerval:EVENt<x>:SPIBus: CLOCk?
<x> = 1 or 2

Example :TRIGGER:EINTERVAL:EVENT1:SPIBUS: CLOCK? ->
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:CLOCK: POLARITY FALL;SOURCE 1

:TRIGger:EINTerval:EVENt<x>:SPIBus: CLOCk:POLarity

Function Sets the polarity of the clock channel of the SPI bus signal trigger or queries the current setting.

Syntax :TRIGger:EINTerval:EVENt<x>:SPIBus: CLOCk:POLarity {FALL|RISE}
:TRIGger:EINTerval:EVENt<x>:SPIBus: CLOCk:POLarity?
<x> = 1 or 2

Example :TRIGGER:EINTERVAL:EVENT1:SPIBUS:CLOCK: POLARITY FALL
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:CLOCK: POLARITY? ->
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:CLOCK: POLARITY FALL

:TRIGger:EINTerval:EVENt<x>:SPIBus: CLOCk:SOURce

Function Sets the clock channel of the SPI bus signal trigger or queries the current setting.

Syntax :TRIGger:EINTerval:EVENt<x>:SPIBus: CLOCk:SOURce {<NRf>}
:TRIGger:EINTerval:EVENt<x>:SPIBus: CLOCk:SOURce?
<x> = 1 or 2
<NRf> = 1 to 4

Example :TRIGGER:EINTERVAL:EVENT1:SPIBUS:CLOCK: SOURCE 1
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:CLOCK: SOURCE? ->
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:CLOCK: SOURCE 1

:TRIGger:EINTerval:EVENt<x>: SPIBus:CS?

Function Queries all settings related to the chip select channel of the SPI bus signal trigger.

Syntax :TRIGger:EINTerval:EVENt<x>:SPIBus:CS?
<x> = 1 or 2

Example :TRIGGER:EINTERVAL:EVENT1:SPIBUS:CS? ->
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:CS: ACTIVE HIGH;SOURCE 1

:TRIGger:EINTerval:EVENt<x>:SPIBus: CS:ACTive

Function Sets the active level of the chip select channel of the SPI bus signal trigger or queries the current setting.

Syntax :TRIGger:EINTerval:EVENt<x>:SPIBus:CS: ACTive {HIGH|LOW}
:TRIGger:EINTerval:EVENt<x>:SPIBus:CS: ACTive?
<x> = 1 or 2

Example :TRIGGER:EINTERVAL:EVENT1:SPIBUS:CS: ACTIVE HIGH
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:CS: ACTIVE? ->
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:CS: ACTIVE HIGH

:TRIGger:EINTerval:EVENt<x>:SPIBus:CS:SOURce

Function Sets the chip select channel of the SPI bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPIBus:CS:SOURce {<NRf>}
:TRIGger:EINTerval:EVENt<x>:SPIBus:CS:SOURce?
```
<x> = 1 or 2
<NRf> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:CS:SOURCE 1
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:CS:SOURCE? ->
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:CS:SOURCE 1
```

:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>?

Function Queries all settings related to the data of the SPI bus signal trigger.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>?
```
<x> of EVENt<x> = 1 or 2
<x> of DATA<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1? ->
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:BYTE 1;CONDITION TRUE;DPOSITION 1;
PATTERN1 “00010010”;
PATTERN2 “00110100”;
PATTERN3 “01010110”;
PATTERN4 “00010010”;SOURCE 3
```

Description DATA2 is valid when `:TRIGger:EINTerval:EVENt<x>:SPIBus:MODE WIRE4` is specified.

:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:BYTE

Function Sets the number of bytes of the data of the SPI bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:BYTE {<NRf>}
:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:BYTE?
```
<x> of EVENt<x> = 1 or 2
<x> of DATA<x> = 1 or 2
<NRf> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:BYTE 1
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:BYTE? ->
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:BYTE 1
```

:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:CONDition

Function Sets the determination method (match or not match) of the data of the SPI bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:CONDition {FALSe|TRUE}
:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:CONDition?
```
<x> of EVENt<x> = 1 or 2
<x> of DATA<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:CONDITION TRUE
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:CONDITION? ->
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:CONDITION TRUE
```

:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:DPOSition

Function Sets the pattern comparison start position of the data of the SPI bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:DPOSition {<NRf>}
:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:DPOSition?
```
<x> of EVENt<x> = 1 or 2
<x> of DATA<x> = 1 or 2
<NRf> = 0 to 9999

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:DPOSITION 1
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:DPOSITION? ->
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:DPOSITION 1
```

:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:HEXA<x>

Function Sets the data of the SPI bus signal trigger in hexadecimal notation.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:HEXA<x> {<String>}
```
<x> of EVENt<x> = 1 or 2
<x> of DATA<x> = 1 or 2
<x> of HEXA<x> = 1 to 4
<String> = 2 characters by combining ‘0’ to ‘F’ and ‘X’

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:HEXA1 “AB”
```

:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:PATTern<x>

Function Sets the data of the SPI bus signal trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPIBus:
DATA<x>:PATTern<x> {<String>}
:TRIGger:EINTerval:EVENt<x>:SPIBus:
DATA<x>:PATTern<x>?
```
<x> of EVENt<x> = 1 or 2
<x> of DATA<x> = 1 or 2
<x> of <PATTern x> = 1 to 4
<String> = 8 characters by combining '0', '1', and 'X'

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:
PATTERN1 "10101011"
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:
PATTERN1? ->
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:
PATTERN1 "10101011"
```

:TRIGger:EINTerval:EVENt<x>:SPIBus:DATA<x>:SOURce

Function Sets the trace of the data of the SPI bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPIBus:
DATA<x>:SOURce {<NRf>}
:TRIGger:EINTerval:EVENt<x>:SPIBus:
DATA<x>:SOURce?
```
<x> of EVENt<x> = 1 or 2
<x> of DATA<x> = 1 or 2
<NRf> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:
SOURCE 1
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:
SOURCE? ->
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:DATA1:
SOURCE 1
```

:TRIGger:EINTerval:EVENt<x>:SPIBus:MODE

Function Sets the wiring system of the SPI bus signal trigger (three-wire or four-wire) or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:SPIBus:
MODE {WIRE3|WIRE4}
:TRIGger:EINTerval:EVENt<x>:SPIBus:
MODE?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:
MODE WIRE3
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:
MODE? ->
:TRIGGER:EINTERVAL:EVENT1:SPIBUS:
MODE WIRE3
```

:TRIGger:EINTerval:EVENt<x>:STATe:CHANnel<x>

Function Sets the condition to be satisfied of the channel or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:STATe:
CHANnel<x> {DONTcare|HIGH|LOW}
:TRIGger:EINTerval:EVENt<x>:STATe:
CHANnel<x>?
```
<x> of EVENt<x> = 1 or 2
<x> of CHANnel<x> = 1 to 4

Example
```
:TRIGGER:EINTERVAL:EVENT1:STATE:
CHANNEL1 HIGH
:TRIGGER:EINTERVAL:EVENT1:STATE:
CHANNEL1? ->
:TRIGGER:EINTERVAL:EVENT1:STATE:
CHANNEL1 HIGH
```

Description This command is valid when `:TRIGger:EINTerval:EVENt<x>:TYPE I2CBus`.

:TRIGger:EINTerval:EVENt<x>:TYPE

Function Sets the trigger type of the event or queries the current setting.

Syntax
```
:TRIGger:EINTerval:EVENt<x>:
TYPE {CANBus|EDGE|EQUalify|I2CBus|
LINBus|PQUalify|PSTAte|PULSe|SPATtern|
SPIBus|STATe}
:TRIGger:EINTerval:EVENt<x>:TYPE?
```
<x> = 1 or 2

Example
```
:TRIGGER:EINTERVAL:EVENT1:TYPE CANBUS
:TRIGGER:EINTERVAL:EVENT1:TYPE? ->
:TRIGGER:EINTERVAL:EVENT1:TYPE CANBUS
```

:TRIGger:ENHanced:CANBus?

Function Queries all settings related to the CAN bus signal trigger.

Syntax `:TRIGger:ENHanced:CANBus?`

Example
```
:TRIGGER:ENHANCED:CANBUS? ->
:TRIGGER:ENHANCED:CANBUS:ACK DONTCARE;
BRATE 1000000;DATA:BORDER BIG;
CONDITION DONTCARE;DATA1 0.0000000E+00;
DATA2 255.00000E+00;DLC 8;MSBLSB 7,0;
PATTERN "1110010101100100011110001001001
1001010100010000100011111111111010";
SIGN UNSIGN;:TRIGGER:ENHANCED:CANBUS:
IDEXT:PATTERN "XXXXXXXXXXXXXXXXXXXXXXXX
XXXXX";:TRIGGER:ENHANCED:CANBUS:IDOR:
ID1:ACK DONTCARE;DATA:BORDER BIG;
CONDITION DONTCARE;DATA1 0.0000000E+00;
DATA2 255.00000E+00;DLC 8;MSBLSB 7,0;
PATTERN "00000001001000110100010101100 1
1110001001101010111100110111101111";
SIGN UNSIGN;:TRIGGER:ENHANCED:CANBUS:
IDOR:ID1:FORMAT STD;IDEXT:
PATTERN "11010101111001101111011110000";:
TRIGGER:ENHANCED:CANBUS:IDOR:ID1:IDSTD:
PATTERN "00100100011";:TRIGGER:
ENHANCED:CANBUS:IDOR:ID1:MODE 0;
RTR DATA;:TRIGGER:ENHANCED:CANBUS:IDOR:
ID2:ACK DONTCARE;DATA:BORDER BIG;
CONDITION DONTCARE;DATA1 0.0000000E+00;
DATA2 255.00000E+00;DLC 8;MSBLSB 7,0;
PATTERN "1111111011011100101110101001 10
0001110110010101000011001000010000";
SIGN UNSIGN;:TRIGGER:ENHANCED:CANBUS:
IDOR:ID2:FORMAT STD;IDEXT:
PATTERN "10010001101000101011001111000";:
TRIGGER:ENHANCED:CANBUS:IDOR:ID2:IDSTD:
PATTERN "10001010110";:TRIGGER:
ENHANCED:CANBUS:IDOR:ID2:MODE 0.....
```

:TRIGger:ENHanced:CANBus:ACK

Function Sets the ACK condition of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:CANBus:
ACK {ACK|ACKBoth|DONTcare|NONack}
:TRIGger:ENHanced:CANBus:ACK?
```

Example
```
:TRIGGER:ENHANCED:CANBUS:ACK ACK
:TRIGGER:ENHANCED:CANBUS:ACK? ->
:TRIGGER:ENHANCED:CANBUS:ACK ACK
```

:TRIGger:ENHanced:CANBus:BRATe

Function Sets the bit rate (data transfer rate) of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:CANBus:
BRATe {<NRf>|USER,<NRf>}
:TRIGger:ENHanced:CANBus:BRATe?
```
<NRf> = 33300, 83300, 125000, 250000, 500000, 1000000

<NRf> of USER = See the *User's Manual (IM701310-01E)*.

Example
```
:TRIGGER:ENHANCED:CANBUS:BRATE 83300
:TRIGGER:ENHANCED:CANBUS:BRATE? ->
:TRIGGER:ENHANCED:CANBUS:BRATE 83300
```

:TRIGger:ENHanced:CANBus:DATA?

Function Queries all settings related to the CAN bus signal trigger data.

Syntax `:TRIGger:ENHanced:CANBus:DATA?`

Example
```
:TRIGGER:ENHANCED:CANBUS:DATA? ->
:TRIGGER:ENHANCED:CANBUS:DATA:
BORDER BIG;CONDITION DONTCARE;
DATA1 0.0000000E+00;
DATA2 255.00000E+00;DLC 8;MSBLSB 7,0;
PATTERN "1110010101100100011110001001001
1001010100010000100011111111111010";
SIGN UNSIGN
```

:TRIGger:ENHanced:CANBus:DATA:BORDer

Function Sets the byte order of the CAN bus signal trigger data or queries the current setting.

Syntax
```
:TRIGger:ENHanced:CANBus:DATA:
BORDer {BIG|LITTle}
:TRIGger:ENHanced:CANBus:DATA:BORDer?
```

Example
```
:TRIGGER:ENHANCED:CANBUS:DATA:
BORDER BIG
:TRIGGER:ENHANCED:CANBUS:DATA:
BORDER? ->
:TRIGGER:ENHANCED:CANBUS:DATA:
BORDER BIG
```

:TRIGger:ENHanced:CANBus:DATA:CONDition

Function Sets the data condition of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:CANBus:DATA:
CONDition {BETWeen|DONTcare|FALSe|
GTHan|LTHan|ORANge|TRUE}
:TRIGger:ENHanced:CANBus:DATA:
CONDition?
```

Example
```
:TRIGGER:ENHANCED:CANBUS:DATA:
CONDITION BETWEEN
:TRIGGER:ENHANCED:CANBUS:DATA:
CONDITION? ->
:TRIGGER:ENHANCED:CANBUS:DATA:
CONDITION BETWEEN
```

5

Communication Commands

:TRIGger:ENHanced:CANBus:DATA:DATA<x>

Function Sets the comparison data of the CAN bus signal trigger data or queries the current setting.

Syntax :TRIGger:ENHanced:CANBus:DATA:DATA<x> {<NRf>}
:TRIGger:ENHanced:CANBus:DATA:DATA<x>?
<x> = 1, 2
<NRf> = See the *User's Manual (IM701310-01E)*.

Example :TRIGGER:ENHANCED:CANBUS:DATA:DATA1 1
:TRIGGER:ENHANCED:CANBUS:DATA:DATA1? -> :TRIGGER:ENHANCED:CANBUS:DATA:DATA1 1.0000000E+00

Description
- Use :TRIGger:ENHANCED:CANBus:DATA:DATA1 when :TRIGger:ENHANCED:CANBus:DATA:CONDition GTHan is specified.
- Use :TRIGger:ENHANCED:CANBus:DATA:DATA2 when :TRIGger:ENHANCED:CANBus:DATA:CONDition LTHan is specified.
- Use :TRIGger:ENHANCED:CANBus:DATA:DATA1 to set the smaller value and :TRIGger:ENHANCED:CANBus:DATA:DATA2 to set the larger value when :TRIGger:ENHANCED:CANBus:DATA:CONDition BETWeen|ORANge is specified.

:TRIGger:ENHanced:CANBus:DATA:DLC

Function Sets the number of valid bytes (DLC) of the CAN bus signal trigger data or queries the current setting.

Syntax :TRIGger:ENHanced:CANBus:DATA:DLC {<NRf>}
:TRIGger:ENHanced:CANBus:DATA:DLC?
<NRf> = 0 to 8

Example :TRIGGER:ENHANCED:CANBUS:DATA:DLC 0
:TRIGGER:ENHANCED:CANBUS:DATA:DLC? -> :TRIGGER:ENHANCED:CANBUS:DATA:DLC 0

:TRIGger:ENHanced:CANBus:DATA:HEXA

Function Sets the CAN bus signal trigger data in hexadecimal notation.

Syntax :TRIGger:ENHanced:CANBus:DATA:HEXA {<String>}
<String> = Up to 16 characters by combining '0' to 'F' and 'X' (in one-byte unit)

Example :TRIGGER:ENHANCED:CANBUS:DATA:HEXA "A9"

:TRIGger:ENHanced:CANBus:DATA:MSBLsb

Function Sets the MSB and LSB bits of the CAN bus signal trigger data or queries the current setting.

Syntax :TRIGger:ENHanced:CANBus:DATA:MSBLsb {<NRf>,<NRf>}
:TRIGger:ENHanced:CANBus:DATA:MSBLsb?
<NRf> = See the *User's Manual (IM701310-01E)*.

Example :TRIGGER:ENHANCED:CANBUS:DATA:MSBLSB 1,0
:TRIGGER:ENHANCED:CANBUS:DATA:MSBLSB? -> :TRIGGER:ENHANCED:CANBUS:DATA:MSBLSB 1,0

:TRIGger:ENHanced:CANBus:DATA:PATTern

Function Sets the CAN bus signal trigger data in binary notation or queries the current setting.

Syntax :TRIGger:ENHanced:CANBus:DATA:PATTern {<String>}
:TRIGger:ENHanced:CANBus:DATA:PATTern?
<String> = Up to 64 characters by combining '0,' '1,' and 'X' (in one-byte unit)

Example :TRIGGER:ENHANCED:CANBUS:DATA:PATTERN "11011111"
:TRIGGER:ENHANCED:CANBUS:DATA:PATTERN? -> :TRIGGER:ENHANCED:CANBUS:DATA:PATTERN "11011111"

:TRIGger:ENHanced:CANBus:DATA:SIGN

Function Sets the sign of the CAN bus signal trigger data or queries the current setting.

Syntax :TRIGger:ENHanced:CANBus:DATA:SIGN {SIGN|UNSign}
:TRIGger:ENHanced:CANBus:DATA:SIGN?

Example :TRIGGER:ENHANCED:CANBUS:DATA:SIGN SIGN
:TRIGGER:ENHANCED:CANBUS:DATA:SIGN? -> :TRIGGER:ENHANCED:CANBUS:DATA:SIGN SIGN

:TRIGger:ENHanced:CANBus:IDEXt?

Function Queries all settings related to the ID of the extended format of the CAN bus signal trigger.

Syntax :TRIGger:ENHanced:CANBus:IDEXt?

Example :TRIGGER:ENHANCED:CANBUS:IDEXT? -> :TRIGGER:ENHANCED:CANBUS:IDEXT:PATTERN "11001011011100001110111011111"

:TRIGger:ENHanced:CANBus:IDEXt:HEXA

Function Sets the ID of the extended format of the CAN bus signal trigger in hexadecimal notation.

Syntax :TRIGger:ENHanced:CANBus:IDEXt:HEXA {<String>}
<String> = 8 characters by combining '0' to 'F' and 'X'

Example :TRIGGER:ENHANCED:CANBUS:IDEXT:HEXA "1AEF5906"

:TRIGger:ENHanced:CANBus:IDEXt:PATTern

Function: Sets the ID of the extended format of the CAN bus signal trigger in binary notation or queries the current setting.

Syntax:
```
:TRIGger:ENHanced:CANBus:IDEXt:
PATTern {<String>}
:TRIGger:ENHanced:CANBus:IDEXt:PATTern?
```
<String> = 29 characters by combining '0,' '1,' and 'X'

Example:
```
:TRIGGER:ENHANCED:CANBUS:IDEXT:
PATTERN "11001011011100001110111011111"
:TRIGGER:ENHANCED:CANBUS:IDEXT:
PATTERN? ->
:TRIGGER:ENHANCED:CANBUS:IDEXT:
PATTERN "11001011011100001110111011111"
```

:TRIGger:ENHanced:CANBus:IDOR?

Function: Queries all settings related to the OR condition of the CAN bus signal trigger.

Syntax:
```
:TRIGger:ENHanced:CANBus:IDOR?
```

Example:
```
:TRIGGER:ENHANCED:CANBUS:IDOR? ->
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:
ACK DONTCARE;DATA:BORDER BIG;
CONDITION DONTCARE;DATA1 0.0000000E+00;
DATA2 255.00000E+00;DLC 8;MSBLSB 7,0;
PATTERN "0000000100100011010001010110011
11100010011010101111001101111101111";
SIGN UNSIGN;:TRIGGER:ENHANCED:CANBUS:
IDOR:ID1:FORMAT STD;IDEXT:
PATTERN "11010101111001101111011110000";:
TRIGGER:ENHANCED:CANBUS:IDOR:ID1:IDSTD:
PATTERN "00100100011";:TRIGGER:
ENHANCED:CANBUS:IDOR:ID1:MODE 0;
RTR DATA;:TRIGGER:ENHANCED:CANBUS:IDOR:
ID2:ACK DONTCARE;DATA:BORDER BIG;
CONDITION DONTCARE;DATA1 0.0000000E+00;
DATA2 255.00000E+00;DLC 8;MSBLSB 7,0;
PATTERN "1111111011011100101110101001100
00011101100101010000110010000010000";
SIGN UNSIGN;:TRIGGER:ENHANCED:CANBUS:
IDOR:ID2:FORMAT STD;IDEXT:
PATTERN "10010001101000101011001111000";:
TRIGGER:ENHANCED:CANBUS:IDOR:ID2:IDSTD:
PATTERN "10001010110";:TRIGGER:
ENHANCED:CANBUS:IDOR:ID2:MODE 0;
RTR DATA;:TRIGGER:ENHANCED:CANBUS:IDOR:
ID3:ACK DONTCARE;DATA:BORDER BIG;
CONDITION DONTCARE;DATA1 0.0000000E+00;
DATA2 255.00000E+00;DLC 8.....
```

:TRIGger:ENHanced:CANBus:IDOR:ID<x>?

Function: Queries all settings related to each ID of the OR condition of the CAN bus signal trigger.

Syntax:
```
:TRIGger:ENHanced:CANBus:IDOR:ID<x>?
```
<x> = 1 to 4

Example:
```
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1? ->
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:
ACK DONTCARE;DATA:BORDER BIG;
CONDITION DONTCARE;DATA1 0.0000000E+00;
DATA2 255.00000E+00;DLC 8;MSBLSB
7,0;PATTERN "0000000100100011010001010l
10011110001001101010111100110111101111";
SIGN UNSIGN;:TRIGGER:ENHANCED:CANBUS:
IDOR:ID1:FORMAT STD;IDEXT:
PATTERN "11010101111001101111011110000";:
TRIGGER:ENHANCED:CANBUS:IDOR:ID1:IDSTD:
PATTERN "00100100011";:TRIGGER:
ENHANCED:CANBUS:IDOR:ID1:MODE 0;
RTR DATA
```

:TRIGger:ENHanced:CANBus:IDOR:ID<x>:ACK

Function: Sets each ACK condition of the OR condition of the CAN bus signal trigger or queries the current setting.

Syntax:
```
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:
ACK {ACK|ACKBoth|DONTcare|NONack}
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:
ACK?
```
<x> = 1 to 4

Example:
```
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:
ACK ACK
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:
ACK? ->
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:
ACK ACK
```

:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA?

Function: Queries all settings related to each data of the OR condition of the CAN bus signal trigger.

Syntax:
```
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:
DATA?
```
<x> = 1 to 4

Example:
```
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:
DATA? ->
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:DATA:
BORDER BIG;CONDITION DONTCARE;
DATA1 0.0000000E+00;
DATA2 255.00000E+00;DLC 8;MSBLSB 7,0;
PATTERN "0000000100100011010001010110011
11100010011010101111001101111101111";
SIGN UNSIGN
```

5 Communication Commands

:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA:BORDer

Function Sets byte order of each data of the OR condition of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:
DATA:BORDer {BIG|LITTle}
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:
DATA:BORDer?
```
<x> = 1 to 4

Example
```
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:DATA:
BORDER BIG
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:DATA:
BORDER? ->
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:DATA:
BORDER BIG
```

:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA:CONDition

Function Sets each data condition of the OR condition of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:
DATA:CONDition {BETWeen|DONTcare|FALSe|
GTHan|LTHan|ORANge|TRUE}
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:
DATA:CONDition?
```
<x> = 1 to 4

Example
```
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:DATA:
CONDITION BETWEEN
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:DATA:
CONDITION? ->
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:DATA:
CONDITION BETWEEN
```

:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA:DATA<x>

Function Sets comparison data of each data of the OR condition of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:
DATA:DATA<x> {<NRf>}
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:
DATA:DATA<x>?
```
<x> of ID<x> = 1 to 4
<x> of DATA<x> = 1 or 2
<NRf> = See the *User's Manual (IM701310-01E)*.

Example
```
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:DATA:
DATA1 1
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:DATA:
DATA1? ->
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:DATA:
DATA1 1.0000000E+00
```

Description
- Use `:TRIGger:ENHANCED:CANBus:IDOR:ID<x>:DATA:DATA1` when `:TRIGger:ENHANCED:CANBus:IDOR:ID<x>:DATA:CONDition GTHan` is specified.
- Use `:TRIGger:ENHANCED:CANBus:IDOR:ID<x>:DATA:DATA2` when `:TRIGger:ENHANCED:CANBus:IDOR:ID<x>:DATA:CONDition LTHan` is specified.
- Use `:TRIGger:ENHANCED:CANBus:IDOR:ID<x>:DATA:DATA1` to set the smaller value and `:TRIGger:ENHANCED:CANBus:IDOR:ID<x>:DATA:DATA2` to set the larger value when `:TRIGger:ENHANCED:CANBus:IDOR:ID<x>:DATA:CONDition BETWeen|ORANge` is specified.

:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA:DLC

Function Sets the number of valid bytes (DLC) of each data of the OR condition of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:
DATA:DLC {<NRf>}
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:
DATA:DLC?
```
<x> = 1 to 4
<NRf> = 0 to 8

Example
```
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:DATA:
DLC 0
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:DATA:
DLC? ->
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:DATA:
DLC 0
```

:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA:HEXA

Function Sets each data of the OR condition of the CAN bus signal trigger in hexadecimal notation.

Syntax
```
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:
DATA:HEXA {<String>}
```
<x> = 1 to 4
<String> = Up to 16 characters by combining '0' to 'F' and 'X' (in one-byte unit)

Example
```
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:DATA:
HEXA "A9"
```

:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA:MSBLsb

Function Sets the MSB and LSB bits of each data of the OR condition of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:
DATA:MSBLsb {<NRf>,<NRf>}
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:
DATA:MSBLsb?
```
<x> = 1 to 4
<NRf> = See the *User's Manual (IM701310-01E)*.

Example
```
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:DATA:
MSBLSB 1,0
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:DATA:
MSBLSB? ->
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:DATA:
MSBLSB 1,0
```

:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA:PATTern

Function Sets each data of the OR condition of the CAN bus signal trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:
DATA:PATTern {<String>}
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:
DATA:PATTern?
```
<x> = 1 to 4
<String> = Up to 64 characters by combining '0,' '1,' and 'X' (in one-byte unit)

Example
```
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:DATA:
PATTERN "11011111"
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:DATA:
PATTERN? ->
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:DATA:
PATTERN "11011111"
```

:TRIGger:ENHanced:CANBus:IDOR:ID<x>:DATA:SIGN

Function Sets sign of each data of the OR condition of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:
DATA:SIGN {SIGN|UNSign}
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:
DATA:SIGN?
```
<x> = 1 to 4

Example
```
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:DATA:
SIGN SIGN
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:DATA:
SIGN? ->
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:DATA:
SIGN SIGN
```

:TRIGger:ENHanced:CANBus:IDOR:ID<x>:FORMat

Function Sets each message format (standard or extended) of the OR condition of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:
FORMat {STD|EXT}
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:
FORMat?
```
<x> = 1 to 4

Example
```
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:
FORMAT STD
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:
FORMAT? ->
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:
FORMAT STD
```

:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDEXt?

Function Queries all settings related to the ID of each extended format of the OR condition of the CAN bus signal trigger.

Syntax
```
:TRIGger:ENHanced:CANBus:IDOR:ID<x>:
IDEXt?
```
<x> = 1 to 4

Example
```
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:
IDEXT? ->
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:
IDEXT:PATTERN "110010110111000011101110
11111"
```

:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDEXt:HEXA

Function Sets the ID of each extended format of the OR condition of the CAN bus signal trigger in hexadecimal notation.

Syntax `:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDEXt:HEXA {<String>}`

<x> = 1 to 4

<String> = 8 characters by combining '0' to 'F' and 'X'

Example `:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:IDEXT:HEXA "1AEF5906"`

:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDEXt:PATTern

Function Sets the ID of each extended format of the OR condition of the CAN bus signal trigger in binary notation or queries the current setting.

Syntax `:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDEXt:PATTern {<String>}`

`:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDEXt:PATTern?`

<x> = 1 to 4

<String> = 29 characters by combining '0,' '1,' and 'X'

Example `:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:IDEXT:PATTERN "11001011011100001110111011111"`

`:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:IDEXT:PATTERN? -> :TRIGGER:ENHANCED:CANBUS:IDOR:ID1:IDEXT:PATTERN "11001011011100001110111011111"`

:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDSTd?

Function Queries all settings related to the ID of each standard format of the OR condition of the CAN bus signal trigger.

Syntax `:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDSTd?`

<x> = 1 to 4

Example `:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:IDSTD? -> :TRIGGER:ENHANCED:CANBUS:IDOR:ID1:IDSTD:PATTERN "00011111101"`

:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDSTd:HEXA

Function Sets the ID of each standard format of the OR condition of the CAN bus signal trigger in hexadecimal notation.

Syntax `:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDSTd:HEXA {<String>}`

<x> = 1 to 4

<String> = 3 characters by combining '0' to 'F' and 'X'

Example `:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:IDSTD:HEXA "5DF"`

:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDSTd:PATTern

Function Sets the ID of each standard format of the OR condition of the CAN bus signal trigger in binary notation or queries the current setting.

Syntax `:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDSTd:PATTern {<String>}`

`:TRIGger:ENHanced:CANBus:IDOR:ID<x>:IDSTd:PATTern?`

<x> = 1 to 4

<String> = 11 characters by combining '0,' '1,' and 'X'

Example `:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:IDSTD:PATTERN "10111011111"`

`:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:IDSTD:PATTERN? -> :TRIGGER:ENHANCED:CANBUS:IDOR:ID1:IDSTD:PATTERN "10111011111"`

:TRIGger:ENHanced:CANBus:IDOR:ID<x>:MODE

Function Enables or disables each condition of the OR condition of the CAN bus signal trigger or queries the current setting.

Syntax `:TRIGger:ENHanced:CANBus:IDOR:ID<x>:MODE {<Boolean>}`

`:TRIGger:ENHanced:CANBus:IDOR:ID<x>:MODE?`

<x> = 1 to 4

Example `:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:MODE ON`

`:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:MODE? -> :TRIGGER:ENHANCED:CANBUS:IDOR:ID1:MODE 1`

:TRIGger:ENHanced:CANBus:IDOR:ID<x>:RTR

Function Sets each RTR of the OR condition of the CAN bus signal trigger or queries the current setting.

Syntax `:TRIGger:ENHanced:CANBus:IDOR:ID<x>:RTR {DATA|DONTcare|REMote}`

`:TRIGger:ENHanced:CANBus:IDOR:ID<x>:RTR?`

<x> = 1 to 4

Example `:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:RTR DATA`

`:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:RTR? -> :TRIGGER:ENHANCED:CANBUS:IDOR:ID1:RTR DATA`

:TRIGger:ENHanced:CANBus:IDSTd?

Function Queries all settings related to the ID of the standard format of the CAN bus signal trigger.

Syntax `:TRIGger:ENHanced:CANBus:IDSTd?`

Example
```
:TRIGGER:ENHANCED:CANBUS:IDSTD? ->
:TRIGGER:ENHANCED:CANBUS:IDSTD:
PATTERN "00011111101"
```

:TRIGger:ENHanced:CANBus:IDSTd:HEXA

Function Sets the ID of the standard format of the CAN bus signal trigger in hexadecimal notation.

Syntax
```
:TRIGger:ENHanced:CANBus:IDSTd:
HEXA {<String>}
```
<String> = 3 characters by combining '0' to 'F' and 'X'

Example
```
:TRIGGER:ENHANCED:CANBUS:IDSTD:
HEXA "5DF"
```

:TRIGger:ENHanced:CANBus:IDSTd:PATTern

Function Sets the ID of the standard format of the CAN bus signal trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:ENHanced:CANBus:IDSTd:
PATTern {<String>}
:TRIGger:ENHanced:CANBus:IDSTd:PATTern?
```
<String> = 11 characters by combining '0,' '1,' and 'X'

Example
```
:TRIGGER:ENHANCED:CANBUS:IDSTD:
PATTERN "10111011111"
:TRIGGER:ENHANCED:CANBUS:IDSTD:
PATTERN? ->
:TRIGGER:ENHANCED:CANBUS:IDSTD:
PATTERN "10111011111"
```

:TRIGger:ENHanced:CANBus:MODE

Function Sets the CAN bus signal trigger mode or queries the current setting.

Syntax
```
:TRIGger:ENHanced:CANBus:
MODE {EFRame|IDEXt|IDOR|IDSTd|SOF}
:TRIGger:ENHanced:CANBus:MODE?
```

Example
```
:TRIGGER:ENHANCED:CANBUS:MODE EFRAME
:TRIGGER:ENHANCED:CANBUS:MODE? ->
:TRIGGER:ENHANCED:CANBUS:MODE EFRAME
```

:TRIGger:ENHanced:CANBus:RECessive

Function Sets the recessive level (bus level) of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:CANBus:
RECessive {HIGH|LOW}
:TRIGger:ENHanced:CANBus:RECessive?
```

Example
```
:TRIGGER:ENHANCED:CANBUS:RECESSIVE HIGH
:TRIGGER:ENHANCED:CANBUS:RECESSIVE? ->
:TRIGGER:ENHANCED:CANBUS:RECESSIVE HIGH
```

:TRIGger:ENHanced:CANBus:RTR

Function Sets the RTR of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:CANBus:
RTR {DATA|DONTcare|REMote}
:TRIGger:ENHanced:CANBus:RTR?
```

Example
```
:TRIGGER:ENHANCED:CANBUS:RTR DATA
:TRIGGER:ENHANCED:CANBUS:RTR? ->
:TRIGGER:ENHANCED:CANBUS:RTR DATA
```

:TRIGger:ENHanced:CANBus:SOURce

Function Sets the trigger source of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:CANBus:SOURce {<NRf>}
:TRIGger:ENHanced:CANBus:SOURce?
```
<NRf> = 1 to 4

Example
```
:TRIGGER:ENHANCED:CANBUS:SOURCE 1
:TRIGGER:ENHANCED:CANBUS:SOURCE? ->
:TRIGGER:ENHANCED:CANBUS:SOURCE 1
```

:TRIGger:ENHanced:CANBus:SPOint

Function Sets the sample point of the CAN bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:CANBus:SPOint {<NRf>}
:TRIGger:ENHanced:CANBus:SPOint?
```
<NRf> = 18.8 to 90.6(%)

Example
```
:TRIGGER:ENHANCED:CANBUS:SPOINT 18.8
:TRIGGER:ENHANCED:CANBUS:SPOINT? ->
:TRIGGER:ENHANCED:CANBUS:
SPOINT 18.8E+00
```

:TRIGger:ENHanced:I2CBus?

Function Queries all settings related to the I^2C bus signal trigger.

Syntax :TRIGger:ENHanced:I2CBus?

Example :TRIGGER:ENHANCED:I2CBUS? -> :TRIGGER:ENHANCED:I2CBUS:ADATA: BIT10ADDRESS:PATTERN "10111011111";: TRIGGER:ENHANCED:I2CBUS:ADATA: BIT7ADDRESS:PATTERN "11011110";: TRIGGER:ENHANCED:I2CBUS:ADATA: BIT7APSUB:ADDRESS:PATTERN "10101011";: TRIGGER:ENHANCED:I2CBUS:ADATA: BIT7APSUB:SADDRESS:PATTERN "10101011";: TRIGGER:ENHANCED:I2CBUS:ADATA: TYPE BIT10ADDRESS;:TRIGGER:ENHANCED: I2CBUS:CLOCK:SOURCE 1;:TRIGGER: ENHANCED:I2CBUS:DATA:BYTE 1; CONDITION TRUE;DPOSITION 1;MODE 1; PATTERN1 "10101011"; PATTERN2 "10101010"; PATTERN3 "10101111"; PATTERN4 "10101011";PMODE DONTCARE; SOURCE 1;:TRIGGER:ENHANCED:I2CBUS: GCALL:BIT7MADDRESS:PATTERN "1010101";: TRIGGER:ENHANCED:I2CBUS:GCALL: SBYTE BIT7MADDRESS;:TRIGGER:ENHANCED: I2CBUS:MODE ADATA;NAIGNORE:HSMODE 1; RACCESS 1;SBYTE 1;:TRIGGER:ENHANCED: I2CBUS:SBHSMODE:TYPE HSMODE

:TRIGger:ENHanced:I2CBus:ADATa?

Function Queries all settings related to the address of the I^2C bus signal trigger.

Syntax :TRIGger:ENHanced:I2CBus:ADATa?

Example :TRIGGER:ENHANCED:I2CBUS:ADATA? -> :TRIGGER:ENHANCED:I2CBUS:ADATA: BIT10ADDRESS:PATTERN "10111011111";: TRIGGER:ENHANCED:I2CBUS:ADATA: BIT7ADDRESS:PATTERN "11011110";: TRIGGER:ENHANCED:I2CBUS:ADATA: BIT7APSUB:ADDRESS:PATTERN "10101011";: TRIGGER:ENHANCED:I2CBUS:ADATA: BIT7APSUB:SADDRESS:PATTERN "10101011";: TRIGGER:ENHANCED:I2CBUS:ADATA: TYPE BIT10ADDRESS

:TRIGger:ENHanced:I2CBus:ADATa: BIT10address?

Function Queries all settings related to the 10-bit address of the I^2C bus signal trigger.

Syntax :TRIGger:ENHanced:I2CBus:ADATa: BIT10address?

Example :TRIGGER:ENHANCED:I2CBUS:ADATA: BIT10ADDRESS? -> :TRIGGER:ENHANCED:I2CBUS:ADATA: BIT10ADDRESS:PATTERN "10111011111"

:TRIGger:ENHanced:I2CBus:ADATa: BIT10address:HEXA

Function Sets the 10-bit address of the I^2C bus signal trigger in hexadecimal notation.

Syntax :TRIGger:ENHanced:I2CBus:ADATa: BIT10address:HEXA {<String>}
<String> = 3 characters by combining '0' to 'F' and 'X' (bit 8 is the R/$\overline{W}$ bit)

Example :TRIGGER:ENHANCED:I2CBUS:ADATA: BIT10ADDRESS:HEXA "7AB"

:TRIGger:ENHanced:I2CBus:ADATa: BIT10address:PATTern

Function Sets the 10-bit address of the I^2C bus signal trigger in binary notation or queries the current setting.

Syntax :TRIGger:ENHanced:I2CBus:ADATa: BIT10address:PATTern {<String>}
:TRIGger:ENHanced:I2CBus:ADATa: BIT10address:PATTern?
<String> = 11 characters by combining '0', '1', and 'X' (bit 8 is the R/$\overline{W}$ bit)

Example :TRIGGER:ENHANCED:I2CBUS:ADATA: BIT10ADDRESS:PATTERN "10111011111"
:TRIGGER:ENHANCED:I2CBUS:ADATA: BIT10ADDRESS:PATTERN? -> :TRIGGER:ENHANCED:I2CBUS:ADATA: BIT10ADDRESS:PATTERN "10111011111"

:TRIGger:ENHanced:I2CBus:ADATa: BIT7ADdress?

Function Queries all settings related to the 7-bit address of the I^2C bus signal trigger.

Syntax :TRIGger:ENHanced:I2CBus:ADATa: BIT7ADdress?

Example :TRIGGER:ENHANCED:I2CBUS:ADATA: BIT7ADDRESS? -> :TRIGGER:ENHANCED:I2CBUS:ADATA: BIT7ADDRESS:PATTERN "11011110"

:TRIGger:ENHanced:I2CBus:ADATa: BIT7ADdress:HEXA

Function Sets the 7-bit address of the I^2C bus signal trigger in hexadecimal notation.

Syntax :TRIGger:ENHanced:I2CBus:ADATa: BIT7ADdress:HEXA {<String>}
<String> = 2 characters by combining '0' to 'F' and 'X' (bit 0 is the R/$\overline{W}$ bit)

Example :TRIGGER:ENHANCED:I2CBUS:ADATA: BIT7ADDRESS:HEXA "DE"

:TRIGger:ENHanced:I2CBus:ADATa:BIT7ADdress:PATTern

Function Sets the 7-bit address of the I^2C bus signal trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:ADATa:
BIT7ADdress:PATTern {<String>}
:TRIGger:ENHanced:I2CBus:ADATa:
BIT7ADdress:PATTern?
```
<String> = 8 characters by combining '0', '1', and 'X' (bit 0 is the R/$\overline{W}$ bit)

Example
```
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7ADDRESS:PATTERN "11011110"
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7ADDRESS:PATTERN? ->
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7ADDRESS:PATTERN "11011110"
```

:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub?

Function Queries all settings related to the 7-bit + Sub address of the I^2C bus signal trigger.

Syntax
```
:TRIGger:ENHanced:I2CBus:ADATa:
BIT7APsub?
```

Example
```
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB? ->
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:ADDRESS:PATTERN "10101011";:
TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:SADDRESS:PATTERN "10101011"
```

:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub:ADDRess?

Function Queries all settings related to the 7-bit address of the 7-bit + Sub address of the I^2C bus signal trigger.

Syntax
```
:TRIGger:ENHanced:I2CBus:ADATa:
BIT7APsub:ADDRess?
```

Example
```
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:ADDRESS? ->
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:ADDRESS:PATTERN "10101011"
```

:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub:ADDRess:HEXA

Function Sets the 7-bit address of the 7-bit + Sub address of the I^2C bus signal trigger in hexadecimal notation.

Syntax
```
:TRIGger:ENHanced:I2CBus:ADATa:
BIT7APsub:ADDRess:HEXA {<String>}
```
<String> = 2 characters by combining '0' to 'F' and 'X' (bit 0 is the R/$\overline{W}$ bit)

Example
```
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:ADDRESS:HEXA "AB"
```

:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub:ADDRess:PATTern

Function Sets the 7-bit address of the 7-bit + Sub address of the I^2C bus signal trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:ADATa:
BIT7APsub:ADDRess:PATTern {<String>}
:TRIGger:ENHanced:I2CBus:ADATa:
BIT7APsub:ADDRess:PATTern?
```
<String> = 8 characters by combining '0', '1', and 'X' (bit 0 is the R/$\overline{W}$ bit)

Example
```
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:ADDRESS:PATTERN "10101011"
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:ADDRESS:PATTERN? ->
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:ADDRESS:PATTERN "10101011"
```

:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub:SADDress?

Function Queries all settings related to the Sub address of the 7-bit + Sub address of the I^2C bus signal trigger.

Syntax
```
:TRIGger:ENHanced:I2CBus:ADATa:
BIT7APsub:SADDress?
```

Example
```
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:SADDRESS? ->
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:SADDRESS:PATTERN "10101011"
```

:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub:SADDress:HEXA

Function Sets the Sub address of the 7-bit + Sub address of the I^2C bus signal trigger in hexadecimal notation.

Syntax
```
:TRIGger:ENHanced:I2CBus:ADATa:
BIT7APsub:SADDress:HEXA {<String>}
```
<String> = 2 characters by combining '0' to 'F' and 'X'

Example
```
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:SADDRESS:HEXA "EF"
```

:TRIGger:ENHanced:I2CBus:ADATa:BIT7APsub:SADDress:PATTern

Function Sets the Sub address of the 7-bit + Sub address of the I^2C bus signal trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:ADATa:
BIT7APsub:SADDress:PATTern {<String>}
:TRIGger:ENHanced:I2CBus:ADATa:
BIT7APsub:SADDress:PATTern?
```
<String> = 8 characters by combining '0', '1', and 'X'

Example
```
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:SADDRESS:PATTERN "10101011"
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:SADDRESS:PATTERN? ->
:TRIGGER:ENHANCED:I2CBUS:ADATA:
BIT7APSUB:SADDRESS:PATTERN "10101011"
```

:TRIGger:ENHanced:I2CBus:ADATa:TYPE

Function Sets the address type of the I²C bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:ADATa:
TYPE {BIT10address|BIT7ADdress|
BIT7APsub}
:TRIGger:ENHanced:I2CBus:ADATa:TYPE?
```

Example
```
:TRIGGER:ENHANCED:I2CBUS:ADATA:
TYPE BIT10ADDRESS
:TRIGGER:ENHANCED:I2CBUS:ADATA:
TYPE? ->
:TRIGGER:ENHANCED:I2CBUS:ADATA:
TYPE BIT10ADDRESS
```

:TRIGger:ENHanced:I2CBus:CLOCk?

Function Queries all settings related to the clock channel of the I²C bus signal trigger.

Syntax
```
:TRIGger:ENHanced:I2CBus:CLOCk?
```

Example
```
:TRIGGER:ENHANCED:I2CBUS:CLOCK? ->
:TRIGGER:ENHANCED:I2CBUS:CLOCK:SOURCE 1
```

:TRIGger:ENHanced:I2CBus:CLOCk:SOURce

Function Sets the clock channel of the I²C bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:CLOCk:
SOURce {<NRf>}
:TRIGger:ENHanced:I2CBus:CLOCk:SOURce?
```
<NRf> = 1 to 4

Example
```
:TRIGGER:ENHANCED:I2CBUS:CLOCK:SOURCE 1
:TRIGGER:ENHANCED:I2CBUS:CLOCK:
SOURCE? ->
:TRIGGER:ENHANCED:I2CBUS:CLOCK:
SOURCE 1
```

:TRIGger:ENHanced:I2CBus:DATA?

Function Queries all settings related to the data of the I²C bus signal trigger.

Syntax
```
:TRIGger:ENHanced:I2CBus:DATA?
```
<x> = 1 or 2

Example
```
:TRIGGER:ENHANCED:I2CBUS:DATA? ->
:TRIGGER:ENHANCED:I2CBUS:DATA:BYTE 1;
CONDITION TRUE;DPOSITION 1;MODE 1;
PATTERN1 "10101011";
PATTERN2 "10101010";
PATTERN3 "10101111";
PATTERN4 "10101011";PMODE DONTCARE;
SOURCE 1
```

:TRIGger:ENHanced:I2CBus:DATA:BYTE

Function Sets the number of data bytes of the I²C bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:DATA:
BYTE {<NRf>}
:TRIGger:ENHanced:I2CBus:DATA:BYTE?
```
<NRf> = 1 to 4

Example
```
:TRIGGER:ENHANCED:I2CBUS:DATA:BYTE 1
:TRIGGER:ENHANCED:I2CBUS:DATA:BYTE? ->
:TRIGGER:ENHANCED:I2CBUS:DATA:BYTE 1
```

:TRIGger:ENHanced:I2CBus:DATA:CONDition

Function Sets the determination method (match or not match) of the data of the I²C bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:DATA:
CONDition {FALSe|TRUE}
:TRIGger:ENHanced:I2CBus:DATA:
CONDition?
```

Example
```
:TRIGGER:ENHANCED:I2CBUS:DATA:
CONDITION TRUE
:TRIGGER:ENHANCED:I2CBUS:DATA:
CONDITION? ->
:TRIGGER:ENHANCED:I2CBUS:DATA:
CONDITION TRUE
```

:TRIGger:ENHanced:I2CBus:DATA:DPOSition

Function Sets the position for comparing the data pattern of the I²C bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:DATA:
DPOSition {<NRf>}
:TRIGger:ENHanced:I2CBus:DATA:
DPOSition?
```
<NRf> = 0 to 9999

Example
```
:TRIGGER:ENHANCED:I2CBUS:DATA:
DPOSITION 1
:TRIGGER:ENHANCED:I2CBUS:DATA:
DPOSITION? ->
:TRIGGER:ENHANCED:I2CBUS:DATA:
DPOSITION 1
```

:TRIGger:ENHanced:I2CBus:DATA:HEXA<x>

Function Sets the data of the I²C bus signal trigger in hexadecimal notation.

Syntax
```
:TRIGger:ENHanced:I2CBus:DATA:
HEXA<x> {<String>}
```
<x> = 1 to 4
<String> = 2 characters by combining '0' to 'F' and 'X'

Example
```
:TRIGGER:ENHANCED:I2CBUS:DATA:
HEXA1 "AB"
```

:TRIGger:ENHanced:I2CBus:DATA:MODE

Function Enables/Disables the data conditions of the I^{2}C bus signal trigger or queries the current setting.

Syntax :TRIGger:ENHanced:I2CBus:DATA:MODE {<Boolean>}
:TRIGger:ENHanced:I2CBus:DATA:MODE?

Example :TRIGGER:ENHANCED:I2CBUS:DATA:MODE ON
:TRIGGER:ENHANCED:I2CBUS:DATA:MODE? ->
:TRIGGER:ENHANCED:I2CBUS:DATA:MODE 1

:TRIGger:ENHanced:I2CBus:DATA:PATTern<x>

Function Sets the data of the I^{2}C bus signal trigger in binary notation or queries the current setting.

Syntax :TRIGger:ENHanced:I2CBus:DATA:PATTern<x> {<String>}
:TRIGger:ENHanced:I2CBus:DATA:PATTern<x>?
<x> = 1 to 4
<String> = 8 characters by combining ‘0’, ‘1’, and ‘X’

Example :TRIGGER:ENHANCED:I2CBUS:DATA:PATTERN1 “10101011”
:TRIGGER:ENHANCED:I2CBUS:DATA:PATTERN1? ->
:TRIGGER:ENHANCED:I2CBUS:DATA:PATTERN1 “10101011”

:TRIGger:ENHanced:I2CBus:DATA:PMODe

Function Sets the pattern comparison start position mode of the data of the I^{2}C bus signal trigger or queries the current setting.

Syntax :TRIGger:ENHanced:I2CBus:DATA:PMODe {DONTcare|SELect}
:TRIGger:ENHanced:I2CBus:DATA:PMODe?

Example :TRIGGER:ENHANCED:I2CBUS:DATA:PMODE SELECT
:TRIGGER:ENHANCED:I2CBUS:DATA:PMODE? ->
:TRIGGER:ENHANCED:I2CBUS:DATA:PMODE SELECT

:TRIGger:ENHanced:I2CBus:DATA:SOURce

Function Sets the data trace of the I^{2}C bus signal trigger or queries the current setting.

Syntax :TRIGger:ENHanced:I2CBus:DATA:SOURce {<NRf>}
:TRIGger:ENHanced:I2CBus:DATA:SOURce?
<NRf> = 1 to 4

Example :TRIGGER:ENHANCED:I2CBUS:DATA:SOURCE 1
:TRIGGER:ENHANCED:I2CBUS:DATA:SOURCE? ->
:TRIGGER:ENHANCED:I2CBUS:DATA:SOURCE 1

:TRIGger:ENHanced:I2CBus:GCALl?

Function Queries all settings related to the general call of the I^{2}C bus signal trigger.

Syntax :TRIGger:ENHanced:I2CBus:GCALl?
<x> = 1 or 2

Example :TRIGGER:ENHANCED:I2CBUS:GCALL? ->
:TRIGGER:ENHANCED:I2CBUS:GCALL:BIT7MADDRESS:PATTERN “1010101”;:TRIGGER:ENHANCED:I2CBUS:GCALL:SBYTE BIT7MADDRESS

:TRIGger:ENHanced:I2CBus:GCALl:BIT7maddress?

Function Queries all settings related to the 7-bit master address of the general call of the I^{2}C bus signal trigger.

Syntax :TRIGger:ENHanced:I2CBus:GCALl:BIT7maddress?
<x> = 1 or 2

Example :TRIGGER:ENHANCED:I2CBUS:GCALL:BIT7MADDRESS? ->
:TRIGGER:ENHANCED:I2CBUS:GCALL:BIT7MADDRESS:PATTERN “1010101”

:TRIGger:ENHanced:I2CBus:GCALl:BIT7maddress:HEXA

Function Sets the 7-bit master address of the general call of the I^{2}C bus signal trigger in hexadecimal notation.

Syntax :TRIGger:ENHanced:I2CBus:GCALl:BIT7maddress:HEXA {<String>}
<x> = 1 or 2
<String> = 2 characters by combining ‘0’ to ‘F’ and ‘X’ (bit 0 is fixed to ‘1’)

Example :TRIGGER:ENHANCED:I2CBUS:GCALL:BIT7MADDRESS:HEXA “AB”

:TRIGger:ENHanced:I2CBus:GCALl:BIT7maddress:PATTern

Function Sets the 7-bit master address of the general call of the I^{2}C bus signal trigger in binary notation or queries the current setting.

Syntax :TRIGger:ENHanced:I2CBus:GCALl:BIT7maddress:PATTern {<String>}
:TRIGger:ENHanced:I2CBus:GCALl:BIT7maddress:PATTern?
<x> = 1 or 2
<String> = 7 characters by combining ‘0’, ‘1’, and ‘X’

Example :TRIGGER:ENHANCED:I2CBUS:GCALL:BIT7MADDRESS:PATTERN “1010101”
:TRIGGER:ENHANCED:I2CBUS:GCALL:BIT7MADDRESS:PATTERN? ->
:TRIGGER:ENHANCED:I2CBUS:GCALL:BIT7MADDRESS:PATTERN “1010101”

:TRIGger:ENHanced:I2CBus:GCALl:SBYTe (Second Byte)

Function Sets the second byte type of the general call of the I^2C bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:GCALl:
SBYTe {BIT7maddress|DONTcare|H04|H06}
:TRIGger:ENHanced:I2CBus:GCALl:SBYTe?
```
Example
```
:TRIGGER:ENHANCED:I2CBUS:GCALL:
SBYTE BIT7MADDRESS
:TRIGGER:ENHANCED:I2CBUS:GCALL:
SBYTE? ->
:TRIGGER:ENHANCED:I2CBUS:GCALL:
SBYTE BIT7MADDRESS
```

:TRIGger:ENHanced:I2CBus:MODE

Function Sets the trigger mode of the I^2C bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:
MODE {ADATa|EStart|GCALl|NAIGnore|
SBHSmode}
:TRIGger:ENHanced:I2CBus:MODE?
```
Example
```
:TRIGGER:ENHANCED:I2CBUS:MODE ADATA
:TRIGGER:ENHANCED:I2CBUS:MODE? ->
:TRIGGER:ENHANCED:I2CBUS:MODE ADATA
```

:TRIGger:ENHanced:I2CBus:NAIGnore?

Function Queries all settings related to the NON ACK ignore mode of the I^2C bus signal trigger.

Syntax
```
:TRIGger:ENHanced:I2CBus:NAIGnore?
```
Example
```
:TRIGGER:ENHANCED:I2CBUS:NAIGNORE? ->
:TRIGGER:ENHANCED:I2CBUS:NAIGNORE:
HSMODE 1;RACCESS 1;SBYTE 1
```

:TRIGger:ENHanced:I2CBus:NAIGnore:HSMode

Function Sets whether to ignore NON ACK in high speed mode of the I^2C bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:NAIGnore:
HSMode {<Boolean>}
:TRIGger:ENHanced:I2CBus:NAIGnore:
HSMode?
```
Example
```
:TRIGGER:ENHANCED:I2CBUS:NAIGNORE:
HSMODE ON
:TRIGGER:ENHANCED:I2CBUS:NAIGNORE:
HSMODE? ->
:TRIGGER:ENHANCED:I2CBUS:NAIGNORE:
HSMODE 1
```

:TRIGger:ENHanced:I2CBus:NAIGnore:RACCess

Function Sets whether to ignore NON ACK in read access mode of the I^2C bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:NAIGnore:
RACCess {<Boolean>}
:TRIGger:ENHanced:I2CBus:NAIGnore:
RACCess?
```
Example
```
:TRIGGER:ENHANCED:I2CBUS:NAIGNORE:
RACCESS ON
:TRIGGER:ENHANCED:I2CBUS:NAIGNORE:
RACCESS? ->
:TRIGGER:ENHANCED:I2CBUS:NAIGNORE:
RACCESS 1
```

:TRIGger:ENHanced:I2CBus:NAIGnore:SBYTe (Start Byte)

Function Sets whether to ignore NON ACK in the start byte of the I^2C bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:NAIGnore:
SBYTe {<Boolean>}
:TRIGger:ENHanced:I2CBus:NAIGnore:
SBYTe?
```
Example
```
:TRIGGER:ENHANCED:I2CBUS:NAIGNORE:
SBYTE ON
:TRIGGER:ENHANCED:I2CBUS:NAIGNORE:
SBYTE? ->
:TRIGGER:ENHANCED:I2CBUS:NAIGNORE:
SBYTE 1
```

:TRIGger:ENHanced:I2CBus:SBHSmode?

Function Queries all settings related to the start byte and high speed mode of the I^2C bus signal trigger.

Syntax
```
:TRIGger:ENHanced:I2CBus:SBHSmode?
```
Example
```
:TRIGGER:ENHANCED:I2CBUS:SBHSMODE? ->
:TRIGGER:ENHANCED:I2CBUS:SBHSMODE:
TYPE HSMODE
```

:TRIGger:ENHanced:I2CBus:SBHSmode:TYPE

Function Sets the type of the start byte or high speed mode of the I^2C bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:I2CBus:SBHSmode:
TYPE {HSMode|SBYTe}
:TRIGger:ENHanced:I2CBus:SBHSmode:TYPE?
```
Example
```
:TRIGGER:ENHANCED:I2CBUS:SBHSMODE:
TYPE HSMODE
:TRIGGER:ENHANCED:I2CBUS:SBHSMODE:
TYPE? ->
:TRIGGER:ENHANCED:I2CBUS:SBHSMODE:
TYPE HSMODE
```

:TRIGger:ENHanced:LINBus?

Function Queries all settings related to the LIN bus trigger or queries the current setting.

Syntax `:TRIGger:ENHanced:LINBus?`

Example `:TRIGGER:ENHANCED:LINBUS? ->`
`:TRIGGER:ENHANCED:LINBUS:BRATE 19200;`
`SOURCE 1`

:TRIGger:ENHanced:LINBus:BRATe

Function Sets the LIN bus signal trigger bitrate (data transfer rate) or queries the current setting.

Syntax `:TRIGger:ENHanced:LINBus:BRATe {<NRf>|`
`USER,<NRf>}`
`:TRIGger:ENHanced:LINBus:BRATe?`
<NRf>=1200, 2400, 4800, 9600, 19200
USER <NRf>=See this *User's Manual*.

Example `:TRIGGER:ENHANCED:LINBUS:BRATE 19200`
`:TRIGGER:ENHANCED:LINBUS:BRATE? ->`
`:TRIGGER:ENHANCED:LINBUS:BRATE 19200`

:TRIGger:ENHanced:LINBus:SOURce

Function Sets the LIN bus signal trigger source or queries the current setting.

Syntax `:TRIGger:ENHanced:LINBus:SOURce {<NRf>}`
`:TRIGger:ENHanced:LINBus:SOURce?`
<NRf>=1–4

Example `:TRIGGER:ENHANCED:LINBUS:SOURCE 1`
`:TRIGGER:ENHANCED:LINBUS:SOURCE? ->`
`:TRIGGER:ENHANCED:LINBUS:SOURCE 1`

:TRIGger:ENHanced:SPIBus?

Function Queries all settings related to the SPI bus signal trigger.

Syntax `:TRIGger:ENHanced:SPIBus?`

Example
```
:TRIGGER:ENHANCED:SPIBUS? ->
:TRIGGER:ENHANCED:SPIBUS:
BITORDER LSBFIRST;CLOCK:POLARITY FALL;
SOURCE 1;:TRIGGER:ENHANCED:SPIBUS:CS:
ACTIVE HIGH;SOURCE 1;:TRIGGER:ENHANCED:
SPIBUS:DATA1:BYTE 1;CONDITION TRUE;
DPOSITION 1;PATTERN1 "00010010";
PATTERN2 "00110100";
PATTERN3 "01010110";
PATTERN4 "00010010";SOURCE 3;:TRIGGER:
ENHANCED:SPIBUS:DATA2:BYTE 4;
CONDITION TRUE;DPOSITION 1;
PATTERN1 "00010010";
PATTERN2 "00110100";
PATTERN3 "01010110";
PATTERN4 "00010010";SOURCE 3;:TRIGGER:
ENHANCED:SPIBUS:MODE WIRE3
```

:TRIGger:ENHanced:SPIBus:BITorder

Function Sets the bit order of the SPI bus signal trigger or queries the current setting.

Syntax `:TRIGger:ENHanced:SPIBus:`
`BITorder {LSBFirst|MSBFirst}`
`:TRIGger:ENHanced:SPIBus:BITorder?`

Example `:TRIGGER:ENHANCED:SPIBUS:`
`BITORDER LSBFIRST`
`:TRIGGER:ENHANCED:SPIBUS:BITORDER? ->`
`:TRIGGER:ENHANCED:SPIBUS:`
`BITORDER LSBFIRST`

:TRIGger:ENHanced:SPIBus:CLOCk?

Function Queries all settings related to the clock channel of the SPI bus signal trigger.

Syntax `:TRIGger:ENHanced:SPIBus:CLOCk?`

Example `:TRIGGER:ENHANCED:SPIBUS:CLOCK? ->`
`:TRIGGER:ENHANCED:SPIBUS:CLOCK:`
`POLARITY FALL;SOURCE 1`

:TRIGger:ENHanced:SPIBus:CLOCk:POLarity

Function Sets the polarity of the clock channel of the SPI bus signal trigger or queries the current setting.

Syntax `:TRIGger:ENHanced:SPIBus:CLOCk:`
`POLarity {FALL|RISE}`
`:TRIGger:ENHanced:SPIBus:CLOCk:`
`POLarity?`

Example `:TRIGGER:ENHANCED:SPIBUS:CLOCK:`
`POLARITY FALL`
`:TRIGGER:ENHANCED:SPIBUS:CLOCK:`
`POLARITY? ->`
`:TRIGGER:ENHANCED:SPIBUS:CLOCK:`
`POLARITY FALL`

:TRIGger:ENHanced:SPIBus:CLOCk:SOURce

Function Sets the clock channel of the SPI bus signal trigger or queries the current setting.

Syntax `:TRIGger:ENHanced:SPIBus:CLOCk:`
`SOURce {<NRf>}`
`:TRIGger:ENHanced:SPIBus:CLOCk:SOURce?`
<NRf> = 1 to 4

Example `:TRIGGER:ENHANCED:SPIBUS:CLOCK:SOURCE 1`
`:TRIGGER:ENHANCED:SPIBUS:CLOCK:`
`SOURCE? ->`
`:TRIGGER:ENHANCED:SPIBUS:CLOCK:SOURCE 1`

:TRIGger:ENHanced:SPIBus:CS?

Function Queries all settings related to the chip select channel of the SPI bus signal trigger.

Syntax `:TRIGger:ENHanced:SPIBus:CS?`

Example `:TRIGGER:ENHANCED:SPIBUS:CS? ->`
`:TRIGGER:ENHANCED:SPIBUS:CS:ACTIVE`
`HIGH;SOURCE 1`

:TRIGger:ENHanced:SPIBus:CS:ACTive

Function Sets the active level of the chip select channel of the SPI bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPIBus:CS:
ACTive {HIGH|LOW}
:TRIGger:ENHanced:SPIBus:CS:ACTive?
```

Example
```
:TRIGGER:ENHANCED:SPIBUS:CS:ACTIVE HIGH
:TRIGGER:ENHANCED:SPIBUS:CS:ACTIVE? ->
:TRIGGER:ENHANCED:SPIBUS:CS:ACTIVE HIGH
```

:TRIGger:ENHanced:SPIBus:CS:SOURce

Function Sets the chip select channel of the SPI bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPIBus:CS:
SOURce {<NRf>}
:TRIGger:ENHanced:SPIBus:CS:SOURce?
```
<NRf> = 1 to 4

Example
```
:TRIGGER:ENHANCED:SPIBUS:CS:SOURCE 1
:TRIGGER:ENHANCED:SPIBUS:CS:SOURCE? ->
:TRIGGER:ENHANCED:SPIBUS:CS:SOURCE 1
```

:TRIGger:ENHanced:SPIBus:DATA<x>?

Function Queries all settings related to the data of the SPI bus signal trigger.

Syntax
```
:TRIGger:ENHanced:SPIBus:DATA<x>?
```
<x> = 1 or 2

Example
```
:TRIGGER:ENHANCED:SPIBUS:DATA1? ->
:TRIGGER:ENHANCED:SPIBUS:DATA1:BYTE 1;
CONDITION TRUE;DPOSITION 1;
PATTERN1 "00010010";
PATTERN2 "00110100";
PATTERN3 "01010110";
PATTERN4 "00010010";SOURCE 3
```

Description DATA2 is valid when `:TRIGger:ENHanced:SPIBus:MODE WIRE4` is specified.

:TRIGger:ENHanced:SPIBus:DATA<x>:BYTE

Function Sets the number of bytes of the data of the SPI bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPIBus:DATA<x>:
BYTE {<NRf>}
:TRIGger:ENHanced:SPIBus:DATA<x>:BYTE?
```
<x> = 1 or 2
<NRf> = 1 to 4

Example
```
:TRIGGER:ENHANCED:SPIBUS:DATA1:BYTE 1
:TRIGGER:ENHANCED:SPIBUS:DATA1:BYTE? ->
:TRIGGER:ENHANCED:SPIBUS:DATA1:BYTE 1
```

:TRIGger:ENHanced:SPIBus:DATA<x>:CONDition

Function Sets the determination method (match or not match) of the data of the SPI bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPIBus:DATA<x>:
CONDition {FALSe|TRUE}
:TRIGger:ENHanced:SPIBus:DATA<x>:
CONDition?
```
<x> = 1 or 2

Example
```
:TRIGGER:ENHANCED:SPIBUS:DATA1:
CONDITION TRUE
:TRIGGER:ENHANCED:SPIBUS:DATA1:
CONDITION? ->
:TRIGGER:ENHANCED:SPIBUS:DATA1:
CONDITION TRUE
```

:TRIGger:ENHanced:SPIBus:DATA<x>:DPOSition

Function Sets the pattern comparison start position of the data of the SPI bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPIBus:DATA<x>:
DPOSition {<NRf>}
:TRIGger:ENHanced:SPIBus:DATA<x>:
DPOSition?
```
<x> = 1 or 2
<NRf> = 0 to 9999

Example
```
:TRIGGER:ENHANCED:SPIBUS:DATA1:
DPOSITION 1
:TRIGGER:ENHANCED:SPIBUS:DATA1:
DPOSITION? ->
:TRIGGER:ENHANCED:SPIBUS:DATA1:
DPOSITION 1
```

:TRIGger:ENHanced:SPIBus:DATA<x>:HEXA<x>

Function Sets the data of the SPI bus signal trigger in hexadecimal notation.

Syntax
```
:TRIGger:ENHanced:SPIBus:DATA<x>:
HEXA<x> {<String>}
```
<x> of DATA<x> = 1 or 2
<x> of HEXA<x> = 1 to 4
<String> = 2 characters by combining '0' to 'F' and 'X'

Example
```
:TRIGGER:ENHANCED:SPIBUS:DATA1:
HEXA1 "AB"
```

:TRIGger:ENHanced:SPIBus:DATA<x>:PATTern<x>

Function Sets the data of the SPI bus signal trigger in binary notation or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPIBus:DATA<x>:
PATTern<x> {<String>}
:TRIGger:ENHanced:SPIBus:DATA<x>:
PATTern<x>?
```
<x> of DATA<x> = 1 or 2
<x> of <PATTern x> = 1 to 4
<String> = 8 characters by combining ‘0’, ‘1’, and ‘X’

Example
```
:TRIGGER:ENHANCED:SPIBUS:DATA1:
PATTERN1 "10101011"
:TRIGGER:ENHANCED:SPIBUS:DATA1:
PATTERN1? ->
:TRIGGER:ENHANCED:SPIBUS:DATA1:
PATTERN1 "10101011"
```

:TRIGger:ENHanced:SPIBus:DATA<x>:SOURce

Function Sets the source channel of the data of the SPI bus signal trigger or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPIBus:DATA<x>:
SOURce {<NRf>}
:TRIGger:ENHanced:SPIBus:DATA<x>:
SOURce?
```
<x> = 1 or 2
<NRf> = 1 to 4

Example
```
:TRIGGER:ENHANCED:SPIBUS:DATA1:SOURCE 1
:TRIGGER:ENHANCED:SPIBUS:DATA1:
SOURCE? ->
:TRIGGER:ENHANCED:SPIBUS:DATA1:SOURCE 1
```

:TRIGger:ENHanced:SPIBus::MODE

Function Sets the wiring system of the SPI bus signal trigger (three-wire or four-wire) or queries the current setting.

Syntax
```
:TRIGger:ENHanced:SPIBus:
MODE {WIRE3|WIRE4}
:TRIGger:ENHanced:SPIBus:MODE?
```

Example
```
:TRIGGER:ENHANCED:SPIBUS:MODE WIRE3
:TRIGGER:ENHANCED:SPIBUS:MODE? ->
:TRIGGER:ENHANCED:SPIBUS:MODE WIRE3
```

:TRIGger:SOURce:CHANnel<x>:LEVel

Function Sets the trigger level of the channel or queries the current setting.

Syntax
```
:TRIGger:SOURce:CHANnel<x>:
LEVel {<Voltage>|<Current>}
:TRIGger:SOURce:CHANnel<x>:LEVel?
```
<x> = 1 to 4
<Voltage> and <Current> = See the *User's Manual (IM701310-01E)*.

Example
```
:TRIGGER:SOURCE:CHANNEL1:LEVEL 1V
:TRIGGER:SOURCE:CHANNEL1:LEVEL? ->
:TRIGGER:SOURCE:CHANNEL1:
LEVEL 1.000E+00
```

Description This command applies to the channel corresponding to the source specified by the following commands.
- :TRIGger:ENHanced:I2CBus:CLOCk:SOURce
- :TRIGger:ENHanced:I2CBus:DATA:SOURce
- :TRIGger:ENHanced:SPIBus:CLOCk:SOURce
- :TRIGger:ENHanced:SPIBus:CS:SOURce
- :TRIGger:ENHanced:SPIBus:DATA[1-2]:SOURce

:TRIGger:SOURce:CHANnel<x>:STATe

Function Sets the condition to be satisfied of the channel or queries the current setting.

Syntax
```
:TRIGger:SOURce:CHANnel<x>:
STATe {DONTcare|HIGH|LOW}
:TRIGger:SOURce:CHANnel<x>:STATe?
```
<x> = 1 to 4

Example
```
:TRIGGER:SOURCE:CHANNEL1:STATE HIGH
:TRIGGER:SOURCE:CHANNEL1:STATE? ->
:TRIGGER:SOURCE:CHANNEL1:STATE HIGH
```

Description • This command is valid when :TRIGger:TYPE I2CBus.

:TRIGger:TYPE

Function Sets the trigger type or queries the current setting.

Syntax
```
:TRIGger:TYPE {CANBus|EDGE|EICYcle|
EIDelay|EISequence|EOR|EQUalify|I2CBus|
LINBus|PQUalify|PSTAte|PULSe|SPATtern|
SPIBus|STATe|TV}
:TRIGger:TYPE?
```

Example
```
:TRIGGER:TYPE CANBus
:TRIGGER:TYPE? ->
:TRIGGER:TYPE CANBus
```

6.1 I²C Bus Signal Analysis Function

Applicable Bus

Item	Specifications
I²C bus	Bus transfer rate: Up to 3.4 Mbits/s Address mode: 7 bits/10 bits

Trigger Function

Item	Specifications
Trigger source	CH1 to CH4
I²C bus signal trigger	Select from the following five trigger types. •Every Start: Activate a trigger when a start condition is detected •Address&Data: Activate a trigger based on the comparison against the specified address and data •Non-ACK: Activate a trigger when Nack is detected •General Call: Activate a trigger based on the comparison against the second byte pattern of the general call address •Start Byte/HS Mode: Activate a trigger on the start byte or the master address of HS mode The address type of the Address&Data trigger can be selected from the following three types. •7bit address •7bit + Sub Adr •10bit Address

Analysis Function

Item	Specifications
Signal	Select from CH1 to CH4 or M1 to M4.
Number of data bytes that can be analyzed	40000 bytes max.
Display of the analysis results	Displays No. (analysis number and start/stop conditions), Time (time from the analysis reference), Binary (binary display of the analysis data), Hex (hexadecimal display of analysis data), and Ack (Acknowledge bit condition).
Zoom link	Moves the zoom position (the center of the zoom box) to the head of the byte highlighted in the analysis result list. Changing the zoom position causes the highlighting in the analysis result list to also move.
Saving of the data of the analysis results list	Saves the list of detailed analysis results to a file in CSV format (.csv extension).

Search Function

Item	Specifications
Data search	Set the address pattern, data pattern, and Acknowledge bit condition and search the waveform. If a waveform that matches the condition is found, the zoom box moves to that point and displays the specified waveform.

6.2 CAN Bus Signal Analysis Function

Applicable CAN Bus

Item	Specifications
CAN bus	CAN Version 2.0B
Baud rate	Set any of the following bit rates: 1 M, 500 k, 250 k, 125 k, 83.3 k, 33.3 k[bps](33.3 kbps supported bu firmware version 2.40 or later), or an arbitrary bit rate from 10 k to 1 M[bps] (0.1 kbps resolution). Supports High speed CAN (ISO11898) and Low speed CAN (ISO11519-2).

Trigger Function

Item	Specifications
Trigger source	Select CH1 to CH4.
CAN bus signal trigger	Select from the following five trigger modes. • SOF: Activates a trigger on the SOF (Start of Frame). • Error Frame: Activates a trigger on an error frame. • ID Std/Data: Activates a trigger on a data frame or remote frame (ID: standard format). • ID Ext/Data: Activates a trigger on a data frame or remote frame (ID: extended format). • ID/Data OR: Activates a trigger on the OR conditions of four types of data frames or remote frames. Select standard or extended format for each ID.

Analysis Function

Item	Specifications
Signal	Select a channel from CH1 to CH4 and M1 to M4.
Number of frames that can be analyzed	Up to 3000 frames
Analyzed frames	Data frame, remote frame, error frame, and overload frame.
Display of the analysis results	• Simple display Displays the analysis number (No.), frame type (Frame), ID in hexadecimal notation, Data in hexadecimal notation, and ACK slot state. • Detail display Displays the analysis number (No.), frame type (Frame), time from the trigger position (Time (ms)), ID in hexadecimal notation, DLC in hexadecimal notation, Data in binary notation (Data (Bin)), Data in hexadecimal notation (Data), CRC sequence in hexadecimal notation, and ACK slot state.
Zoom link	• Zoom link Moves the zoom position (the center of the zoom box) to the head of the frame highlighted in the analysis result list. Changing the zoom position causes the highlighting in the analysis result list to also move. • Field jump When the zoom link function is enabled, the zoom position can be moved to the head of the specified field of the frame that is highlighted in the analysis result list. Select the field from SOF, ID, Control Field, Data Field, CRC, or ACK.
Saving of the data of the analysis results list	Saves the data of the simple display and detail display of the analysis result list in CSV format (.csv extension).

Search Function

Item	Specifications
Data search	Search the waveform by specifying a field or frame condition. If a waveform that matches the condition is found, the zoom box moves to that point and displays the specified waveform in the zoom window.

Stuff Bit Computation Function

Item	Specifications
Stuff bit computation	Extracts stuff bits from the CAN bus waveform and displays them as a MATH waveform (MATH1 to MATH4).

6.3 LIN Bus Signal Analysis Function

Applicable LIN Bus

Item	Specifications
LIN bus	LIN1.3 or LIN2.0
Baud rate	Set any of the following bit rates: 19200, 9600, 4800, 2400, 1200 [bps], or an arbitrary bit rate from 1000 to 20k[bps] (10 bps resolution).

Trigger Function

Item	Specifications
Trigger source	Select CH1 to CH4.
LIN bus signal trigger	Trigged at the rise of Break delimiter

Analysis Function

Item	Specifications
Signal	Select a channel from CH1 to CH4 and M1 to M4.
Number of frames that can be analyzed	Up to 3000 frames
Analyzed frames	ID, ID-field, Data, Checksum, Information(ID parity error, Checksum error, Wakeup signal)
Display of the analysis results	• Simple display Displays the analysis number (No.), ID in hexadecimal notation, Data in hexadecimal notation, and Checksum in hexadecimal notation. • Detail display Displays the analysis number (No.), time from the trigger position (Time (ms)), ID in hexadecimal notation, ID-field in hexadecimal notation, Data in binary notation (Data (Bin)), Data in hexadecimal notation (Data), Checksum in hexadecimal notation, and information.
Zoom link	• Zoom link Moves the zoom position (the center of the zoom box) to the head of the frame highlighted in the analysis result list. Changing the zoom position causes the highlighting in the analysis result list to also move.
Saving of the data of the analysis results list	Saves the data of the simple display and detail display of the analysis result list in CSV format (.csv extension).

Search Function

Item	Specifications
Data search	Search the waveform by specifying Synch field or ID/Data condition. If a waveform that matches the condition is found, the zoom box moves to that point and displays the specified waveform in the zoom window.

6.4 SPI Bus Signal Analysis Function

Trigger Function

Item	Specifications
Trigger source	CH1 to CH4
SPI bus signal trigger	Activate a trigger by comparing data from an arbitrary byte counts after the assertion of the CS. The length of data that is compared can be set to 1 to 4 bytes.

Analysis Function

Item	Specifications
Number of data bytes that can be analyzed	40000 bytes max.
Display of the analysis results	Detailed analysis result list display. Displays No. (analysis number), Time (time from the analysis reference point), Dt1 (hexadecimal or binary display of the Data 1 value), Dt2 (hexadecimal or binary display of the Data 2 value), CS (CS signal status or the CS signal name with high precedence).
Zoom Link	Moves the zoom position (the center of the zoom box) to the head of the byte highlighted in the analysis result list. Changing the zoom position causes the highlighting in the analysis result list to also move.
Saving of the data of the analysis results list	Saves the list of detailed analysis results to a file in CSV format (.csv extension).

Search Function

Item	Specifications
Data search	Search the waveform by specifying a data pattern. If a waveform that matches the pattern is found, the zoom box moves to that point and displays the specified waveform.

Index

I Page

L Page

M Page

N Page

O Page

R Page

S Page

T Page

V **Page**

Z **Page**

Index