User's Manual

Notice of Alterations

DL9140/DL9140L/DL9240/DL9240L Digital Oscilloscope Communicaton Interface

Please make the following alterations to the User's Manual IM701310-17E.

Change the Name of the User's Manual

The Communication Interface USER'S MANUAL (IM701310-17E) on the CD-ROM can be used for both the Yokogawa DL9040 Digital Oscilloscope (model 701307) and DL9040L (model 701308). Please consider the title of the manual to be "DL9040/DL9140/DL9240 Series Digital Oscilloscope Communication Interface USER'S MANUAL."

Functions that have been added to products with firmware version 2.00 or later

■ Page 5-88

:GONogo?

Function Queries all settings related to GO/NO-GO determination.

Syntax `:GONogo?`

Example
```
:GONOGO? -> :GONOGO:ACTION:BUZZER 0;
HCOPY 0;MAIL:INTERVAL OFF;MODE 0;:
GONOGO:ACTION:SAVE 0;:GONOGO:
CONDITION1 DONTCARE;
CONDITION2 DONTCARE;
CONDITION3 DONTCARE;
CONDITION4 DONTCARE;
LOGIC AND;MODE OFF;SCONDITION:
NGCOUNT 1;STOPCOUNT 1;:GONOGO:
TELECOMTEST:SELECT1:MASK:ELEMENT1:
WCOUNT 1.000E+00,2.000E+00;:GONOGO:
TELECOMTEST:SELECT2:MASK:ELEMENT1:
WCOUNT 1.000E+00,2.000E+00;:GONOGO:
TELECOMTEST:SELECT3:MASK:ELEMENT1:
WCOUNT 1.000E+00,2.000E+00;:GONOGO:
TELECOMTEST:SELECT4:MASK:ELEMENT1:
WCOUNT 2.000E+00,3.000E+00;:GONOGO:
ZPARAMETER:SELECT1:MODE PARAMETER;
PARAMETER:CATEGORY FFT;FFT1:
CALCULATION1 0.000E+00,1.000E+00;:
GONOGO:ZPARAMETER:SELECT1:PARAMETER:
MEASURE:STATISTICS MAXIMUM;:GONOGO:
ZPARAMETER:SELECT1:RECTANGLE:
HORIZONTAL 0.000E+00,1.000E+00;
VERTICAL 0.000E+00,1.000E+00;:GONOGO:
ZPARAMETER:SELECT1:TRACE 1;WAVE:
TRANGE 1.000E+00,2.000E+00;:GONOGO:
ZPARAMETER:SELECT1:WINDOW MAIN;:
GONOGO:ZPARAMETER:SELECT2:
MODE PARAMETER;PARAMETER:
CATEGORY FFT;FFT1:
CALCULATION1 0.000E+00,1.000E+00...
```

:GONogo:ACTion?

Function Queries all settings related to the action taken when the determination result is NO-GO and the criteria values.

Syntax `:GONogo:ACTion?`

Example
```
:GONOGO:ACTION? -> :GONOGO:ACTION:
BUZZER 0;HCOPY 0;MAIL:INTERVAL OFF;
MODE 0;:GONOGO:ACTION:SAVE 0
```

■ Page 5-136

:SYSTem:LANGuage

Function Sets the message language or queries the current setting.

Syntax
```
:SYSTem:LANGuage {CHINese|ENGLish|
JAPANese|KORean}
:SYSTem:LANGuage?
```

Example
```
:SYSTEM:LANGUAGE JAPANESE
:SYSTEM:LANGUAGE? ->:SYSTEM:
LANGUAGE JAPANESE
```

:SYSTem:MLANguage

Function Sets the menu language or queries the current setting.

Syntax
```
:SYSTem:MLANguage
{CHINese|ENGLish|KORean}
:SYSTem:MLANguage?
```

Example
```
:SYSTEM:MLANGUAGE ENGLISH
:SYSTEM:MLANGUAGE? ->:SYSTEM:
MLANGUAGE ENGLISH
```

■ Page 5-143

:TRIGger?

Function Queries all settings related to the trigger.

```
Syntax  :TRIGger?
Example :TRIGGER? -> :TRIGGER:ACTION:
        ACQCOUNT 1;BUZZER 0;HCOPY 1;MAIL:
        INTERVAL OFF;MODE 0;:TRIGGER:ACTION:
        MODE ACONDITION;SAVE 1;:TRIGGER:
        TYPE EICYCLE;CLOCK:SOURCE 1;
        POLARITY RISE;:TRIGGER:DELAY:
        EDGECOUNT:COUNT 1;:TRIGGER:DELAY:
        MODE 1;POLARITY FALL;SOURCE 4;
        TIME 1.000E+00;TYPE EDGECOUNT;:
        TRIGGER:EINTERVAL:EVENT1:TYPE EDGE;
        CLOCK:SOURCE 1;POLARITY FALL;:
        TRIGGER:EINTERVAL:EVENT1:ESTATE:
        SOURCE 1;POLARITY FALL;: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 "11100101011001000111100010
        10011001010100010000100011111111110
        10";SIGN UNSIGN;:TRIGGER:EINTERVAL:
        EVENT1:CANBUS:IDEXT:PATTERN "XXXXXX
        XXXXXXXXXXXXXXXXXXXXXX";: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 "0000000100100011
        01000101011001111000100111010101011100
        110111101111";SIGN UNSIGN;:TRIGGER:
        EINTERVAL:EVENT1:CANBUS:IDOR:ID1:
        FORMAT STD;IDEXT:PATTERN "1101010
        11110011011111011110000";: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 "1111111011011100
        10111010100110000111011001010100001 1
        001000010000";SIGN UNSIGN;:TRIGGER:
        EINTERVAL:EVENT1:CANBUS:IDOR:ID2:
        FORMAT STD;IDEXT:PATTERN "1001000
        11010001011001111000";:TRIGGER:
        EINTERVAL:EVENT1:CANBUS:IDOR:ID2:
        IDSTD:PATTERN "10001010110";:TRIGGER:
        EINTERVAL:EVENT1:CANBUS:IDOR:ID2:
        MODE 0.....
```

:TRIGger:ACTion?

Function Queries all settings related to action-on-trigger.

```
Syntax  :TRIGger:ACTion?
Example :TRIGGER:ACTION? -> :TRIGGER:ACTION:
        ACQCOUNT 1;BUZZER 0;
        HCOPY 1;MAIL:INTERVAL OFF;MODE 0;
        :TRIGGER:ACTION:MODE ACONDITION;
        SAVE 1
```

■ Page 5-159

: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|
        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
```

■ Page 5-175

:TRIGger:TYPE

Function Sets the trigger type or queries the current setting.

```
Syntax  :TRIGger:TYPE {CANBus|EDGE|EICYcle|
        EIDelay|EISequence|EOR|EQUalify|
        I2CBus|PQUalify|PSTAte|PULSe|
        SPATtern|SPIBus|STATe|TV}
        :TRIGger:TYPE?
Example :TRIGGER:TYPE CANBUS:TRIGGER:TYPE? ->
        :TRIGGER:TYPE CANBUS
```

■ Mail Transmission Function

:GONogo:ACTion:MAIL?

Function Queries all settings related to the mail transmission when the determination is NO-GO.

Syntax
```
:GONogo:ACTion:MAIL?
```

Example
```
:GONOGO:ACTION:MAIL? -> :GONOGO:
ACTION:MAIL:INTERVAL 10;
MODE 1
```

:GONogo:ACTion:MAIL:INTerval

Function Sets the interval at which to send mail when the determination is NO-GO or queries the current setting.

Syntax
```
:GONogo:ACTion:MAIL:INTerval
{OFF|<NRf>}
:GONogo:ACTion:MAIL:INTerval?
<NRf> = 1 to 1440 (min)
```

Example
```
:GONOGO:ACTION:MAIL:INTERVAL 10
:GONOGO:ACTION:MAIL:INTERVAL? ->
:GONOGO:ACTION:MAIL:INTERVAL 10
```

:GONogo:ACTion:MAIL:MODE

Function Sets whether to send mail when the determination is NO-GO or queries the current setting.

Syntax
```
:GONogo:ACTion:MAIL:MODE {<Boolean>}
:GONogo:ACTion:MAIL:MODE?
```

Example
```
:GONOGO:ACTION:MAIL:MODE ON
:GONOGO:ACTION:MAIL:MODE? ->
:GONOGO:ACTION:MAIL:MODE 1
```

:TRIGger:ACTion:MAIL?

Function Queries all settings related to the mail transmission when an action is activated.

Syntax
```
:TRIGger:ACTion:MAIL?
```

Example
```
:TRIGGER:ACTION:MAIL? ->
:TRIGGER:ACTION:MAIL:INTERVAL 10;
MODE 1
```

:TRIGger:ACTion:MAIL:INTerval

Function Sets the interval at which to send mail when an action is activated or queries the current setting.

Syntax
```
:TRIGger:ACTion:MAIL:INTerval
{OFF|<NRf>}
:TRIGger:ACTion:MAIL:INTerval?
<NRf> = 1 to 1440 (min)
```

Example
```
:TRIGGER:ACTION:MAIL:INTERVAL 10
:TRIGGER:ACTION:MAIL:INTERVAL? ->
:TRIGGER:ACTION:MAIL:INTERVAL 10
```

:TRIGger:ACTion:MAIL:MODE

Function Sets whether to send mail when an action is activated or queries the current setting.

Syntax
```
:TRIGger:ACTion:MAIL:MODE {<Boolean>}
:TRIGger:ACTion:MAIL:MODE?
```

Example
```
:TRIGGER:ACTION:MAIL:MODE ON:TRIGGER:
ACTION:MAIL:MODE? -> :TRIGGER:ACTION:
MAIL:MODE 1
```

■ Trigger Function of the CAN Bus Signal

: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 " 111001010
11001000111100010010011001010100010
0001000111111111111010";SIGN UNSIGN;:
TRIGGER:EINTERVAL:EVENT1:CANBUS:
IDEXT:PATTERN " XXXXXXXXXXXXXXXXXXXXX
XXXXXXXX";: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 " 00000001001000110100010101 1
00111100010011010101111001101111011
11";SIGN UNSIGN;:TRIGGER:EINTERVAL:
EVENT1:CANBUS:IDOR:ID1:FORMAT STD;
IDEXT:PATTERN " 11010101111001101111 0
11110000";: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 " 11111110110111001011101010 0
11000011101100101010000110010000100
00";SIGN UNSIGN;:TRIGGER:EINTERVAL:
EVENT1:CANBUS:IDOR:ID2:FORMAT STD;
IDEXT:PATTERN " 10010001101000101011 0
01111000";: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> = 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" 11100101011001000111100010010
0011001010100010000100011111111111010"
;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 " 110010110111000
01110111011111"
```

: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 " 110010110111000011101
1011111"
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
IDEXT:PATTERN? -> :TRIGGER:EINTERVAL:
EVENT1:CANBUS:IDEXT:PATTERN " 11001011
01110000111011101111"
```

: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 " 00000001001000110100010101100
11110001001101010111100110111101111";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:
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 " 11111110110111001011101010011
10000111011001010100001100100001000
00";SIGN UNSIGN;:TRIGGER:EINTERVAL:
EVENT1:CANBUS:IDOR:ID2:FORMAT STD;
IDEXT:PATTERN"100100011010001010110011
1111000";: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 " 00000001001000110100010101100
1111000100110101011110011011111011
11";SIGN UNSIGN;:TRIGGER:EINTERVAL:
EVENT1:CANBUS:IDOR:ID1:FORMAT STD;
IDEXT:PATTERN " 110101011100110111101
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 " 00000001001000110100010101100
1111000100110101011110011011111011111"
;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" 110010110111000011101
1011111"
```

: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" 1100101101110
0001110111011111"
:TRIGGER:EINTERVAL:EVENT1:CANBUS:
IDOR:ID1:IDEXT:PATTERN? -> :TRIGGER:
EINTERVAL:EVENT1:CANBUS:IDOR:ID1:
IDEXT:PATTERN" 110010110111000011101
1011111"
```

: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: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" 11100101011001000111100010
01001100101010001000010001111111111
010";SIGN UNSIGN;:TRIGGER:ENHANCED:
CANBUS:IDEXT:PATTERN" XXXXXXXXXXXXXX
XXXXXXXXXXXXXXX";: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" 00000001001000110100010101
10011110001001101010111100110111101
111";SIGN UNSIGN;:TRIGGER:ENHANCED:
CANBUS:IDOR:ID1:FORMAT STD;IDEXT:
PATTERN" 11010101111001101111011110000
0";: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" 11111110110111001011101010
01100001110110010101000011001000010
000";SIGN UNSIGN;:TRIGGER:ENHANCED:
CANBUS:IDOR:ID2:FORMAT STD;IDEXT:
PATTERN" 10010001101000101011001111100
0";: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> = 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" 11100101011001000111100010010
0110010101000100001000111111111111010"
;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

: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
1”
:TRIGGER:ENHANCED:CANBUS:IDEXT:
PATTERN? -> :TRIGGER:ENHANCED:CANBUS:
IDEXT:PATTERN“ 1100101101110000111011
1011111”
```

: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“ 00000001001000110100010101
10011110001001101010111100110111101
111”;SIGN UNSIGN;:TRIGGER:ENHANCED:
CANBUS:IDOR:ID1:FORMAT STD;IDEXT:
PATTERN“ 11010101111001101111011110000
0”;: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“ 11111110110111001011101010
01100001110110010101000011001000010
000”;SIGN UNSIGN;:TRIGGER:ENHANCED:
CANBUS:IDOR:ID2:FORMAT STD;IDEXT:
PATTERN“ 10010001101000101011001111100
0”;: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“ 00000001001000110100010101
10011110001001101010111100110111101
111”;SIGN UNSIGN;:TRIGGER:ENHANCED:
CANBUS:IDOR:ID1:FORMAT STD;IDEXT:
PATTERN“ 11010101111001101111011110000
0”;: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" 00000001001000110100010101100
11110001001101010111100110111101111"
;SIGN UNSIGN
```

: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 " 1100101101110
0001110111011111"
```

: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 " 1100101101110000111011
1011111"
:TRIGGER:ENHANCED:CANBUS:IDOR:ID1:
IDEXT:PATTERN? -> :TRIGGER:ENHANCED:
CANBUS:IDOR:ID1:IDEXT:PATTERN " 110010
110111000011101110111111"
```

: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
```