

# Multi-phase Programmable AC/DC Power Source

ASR-6000 Series

---

**DeviceNet MANUAL**

Rev. B



ISO-9001 CERTIFIED MANUFACTURER

**GW INSTEK**

This manual contains proprietary information, which is protected by copyright. All rights are reserved. No part of this manual may be photocopied, reproduced or translated to another language without prior written consent of Good Will company.

The information in this manual was correct at the time of printing. However, Good Will continues to improve products and reserves the rights to change specification, equipment, and maintenance procedures at any time without notice.

**Good Will Instrument Co., Ltd.**  
**No. 7-1, Jhongsing Rd., Tucheng Dist., New Taipei City 236, Taiwan.**

---

# Table of Contents

<b>SAFETY INSTRUCTIONS .....</b>	<b>4</b>
<b>GETTING STARTED.....</b>	<b>8</b>
ASR-6000 Series Overview .....	9
Appearance.....	13
<b>DEVICENET .....</b>	<b>21</b>
Communication Introduction.....	22
Object Model.....	26
Duplicate MAC ID Check Message Protocol.....	29
Connection Establishment Protocol.....	32
Explicit Message Protocol.....	36
<b>REMOTE CONTROL.....</b>	<b>49</b>
DeviceNet Configuration .....	50
Object Class Specification .....	54
Manufacturer-specific Command List .....	82
Manufacturer-specific Command Example.....	311
DeviceNet Errors Code .....	320
<b>APPENDIX.....</b>	<b>321</b>
Factory Default Settings.....	321
<b>INDEX.....</b>	<b>332</b>

# S SAFETY INSTRUCTIONS

This chapter contains important safety instructions that you must follow during operation and storage. Read the following before any operation to ensure your safety and to keep the instrument in the best possible condition.

## Safety Symbols

These safety symbols may appear in this manual or on the instrument.

---



Warning: Identifies conditions or practices that could result in injury or loss of life.



Caution: Identifies conditions or practices that could result in damage to the ASR-6000 or to other properties.



DANGER High Voltage



Attention Refer to the Manual



Protective Conductor Terminal



Earth (ground) Terminal



Do not dispose electronic equipment as unsorted municipal waste. Please use a separate collection facility or contact the supplier from which this instrument was purchased.

## Safety Guidelines

---

### General Guideline



### CAUTION

- Do not place any heavy object on the ASR-6000.
- Avoid severe impact or rough handling that leads to damaging the ASR-6000.
- Do not discharge static electricity to the ASR-6000.
- Use only mating connectors, not bare wires, for the terminals.
- Do not block the cooling fan opening.
- Do not disassemble the ASR-6000 unless you are qualified.
- If the equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.
- Due to the fact that ASR-6000 unit weights greater than 18kg, please resort to the standard kit GRA-451-E for transport or remove the unit by at least two persons in case of danger occurred.

## Power Supply



## WARNING

- AC Input voltage range:  
200 Vac to 240 Vac  $\pm 10\%$   
phase voltage (Delta: L-L, Y: L-N)
  - Frequency: 47 ~ 63 Hz
  - To avoid electrical shock connect the protective grounding conductor of the AC power cord to an earth ground.
  - The power switch that is included in the instrument is not considered a disconnecting device.
  - The permanently connected power input is used as the disconnecting device and shall remain readily operable.
    - a. A switch or circuit-breaker must be included in the installation
    - b. It must be suitably located and easily reached
    - c. It must be marked as the disconnecting device for the equipment.
    - d. It shall be located near the equipment
  - Do not position the equipment so that it is difficult to operate the disconnecting device.
  - Ask for professional technician for installation.
  - It requires 200Vac input condition and the maximum input current [30A (ASR-6450), 40A (ASR-6600)], which conforms to cord diameter by local regulations.
  - Breaker, of which the specification is required to larger than 30A (ASR-6450), 40A (ASR-6600) individually, should be in the near proximity of unit.
-

- Cleaning the ASR-6000
- Disconnect the circuit-breaker or permanently connected power input before cleaning.
  - Use a soft cloth dampened in a solution of mild detergent and water. Do not spray any liquid.
  - Do not use chemicals containing harsh material such as benzene, toluene, xylene, and acetone.

- Operation Environment
- Location: Indoor, no direct sunlight, dust free, almost non-conductive pollution (Note below)
  - Relative Humidity: 20%~ 80%, no condensation
  - Altitude: < 2000m
  - Temperature: 0°C to 40°C
- (Pollution Degree) EN 61010-1:2010 specifies the pollution degrees and their requirements as follows. The ASR-6000 falls under degree 2. Pollution refers to “addition of foreign matter, solid, liquid, or gaseous (ionized gases), that may produce a reduction of dielectric strength or surface resistivity”.
- Pollution degree 1: No pollution or only dry, non-conductive pollution occurs. The pollution has no influence.
  - Pollution degree 2: Normally only non-conductive pollution occurs. Occasionally, however, a temporary conductivity caused by condensation must be expected.
  - Pollution degree 3: Conductive pollution occurs, or dry, non-conductive pollution occurs which becomes conductive due to condensation which is expected. In such conditions, equipment is normally protected against exposure to direct sunlight, precipitation, and full wind pressure, but neither temperature nor humidity is controlled.

- Storage environment
- Location: Indoor
  - Temperature: -10°C to 70°C
  - Relative Humidity: ≤90%, no condensation

Disposal



Do not dispose this instrument as unsorted municipal waste. Please use a separate collection facility or contact the supplier from which this instrument was purchased. Please make sure discarded electrical waste is properly recycled to reduce environmental impact.

# G **ETTING STARTED**

This chapter describes the ASR-6000 power supply in a nutshell, including its main features and front / rear panel introduction.

## ASR-6000 series



<b>ASR-6000 Series Overview</b> .....	<b>9</b>
Series lineup .....	9
Main Features .....	10
Accessories .....	11
<b>Appearance</b> .....	<b>13</b>
Front Panel .....	13
Rear Panel .....	18



## ASR-6000 Series Overview

### Series lineup

The ASR-6000 series consists of 2 models, the ASR-6450 and ASR-6600, differing in capacity. Note that throughout the user manual, the term “ASR-6000” refers to any of the models, unless stated otherwise.

### 1P Output Condition

Model Name	Power Rating	Max. Output Current	Max. Output Voltage
ASR-6450	4500 VA	45 / 22.5 A	350 Vrms / 500 Vdc
ASR-6600	6000 VA	60 / 30 A	350 Vrms / 500 Vdc

### 1P3W Output Condition

Model Name	Power Rating	Max. Output Current	Max. Output Voltage
ASR-6450	3000 VA	15 / 7.5 A	700 Vrms / 1000 Vdc
ASR-6600	4000 VA	20 / 10 A	700 Vrms / 1000 Vdc

### 3P Output Condition (Pre phase)

Model Name	Power Rating	Max. Output Current	Max. Output Voltage
ASR-6450	1500 VA	15 / 7.5 A	350 Vrms / 500 Vdc
ASR-6600	2000 VA	20 / 10 A	350 Vrms / 500 Vdc

## Main Features

---

Performance	<ul style="list-style-type: none"><li>• Maximum phase voltage is 350 Vrms, line voltage is 700 Vrms</li><li>• Maximum DC output voltage is 1000 Vdc</li><li>• Maximum output frequency is 2000 Hz</li><li>• Adjustable Voltage rising time</li><li>• DC full capacity output ability</li><li>• Output voltage total harmonic distortion is less than 0.3% at 50 and 60 Hz</li><li>• Maximum crest factor reached 4 times</li></ul>
Features	<ul style="list-style-type: none"><li>• Include sine, square, triangle, arbitrary and DC output waveforms</li><li>• Variable voltage, frequency and current limiter</li><li>• 100 steps Harmonic voltage and current analysis ability</li><li>• Supported three phase unbalanced output mode</li><li>• Sequence, simulate and preset memory functions</li><li>• AC line frequency synchronized output</li><li>• USB memory save and recall</li><li>• Remote sense compensator</li><li>• Supported 1P, 1P3W and 3P output phase</li><li>• External control I/O and signal input applications</li><li>• Voltage and current monitor output</li><li>• Voltage control amplifier output</li><li>• PC software, web control and data log functions</li></ul>
Interface	<ul style="list-style-type: none"><li>• Built-in LAN, USB host, USB device and RS232 interface</li><li>• Optional GPIB, DeviceNet and CAN BUS interface</li></ul>

## Accessories

Before using the ASR-6000 power source unit, check the package contents to make sure all the standard accessories are included.

Standard Accessories	Part number	Description
		Quick Start Guide
	82GW1SAFE0M*1	Safety guide
	62SR-6K0SC401	Input terminal cover
	62SR-6K0SC301	Output terminal cover
	62SR-6K0CP101	Copper plate for delta connection input (Mark 1)
	62SR-6K0CP201	Copper plate for single phase and Y connection input (Mark 2)
	62SR-6K0CP301	Copper plate for delta connection input (Mark 3)
	62SR-6K0CP401	Copper plate for 1P output (Mark 4)
	GRA-451-E	Rack mount adapter (EIA)
	GTL-246	USB cable (USB 2.0 Type A - Type B cable, approx. 1.2M)
Optional Accessories	Part number	Description
	GRA-451-J	Rack mount adapter (JIS)
	GPW-008	Power Cord SJT 10AWG/3C, 3m Max Length, 105oC, RV5-5*3P, RV5-5*3P UL TYPE
	GPW-009	Power Cord H05VV-F 2.5mm <sup>2</sup> /3C, 3m Max Length, 105oC, RVS3-5*3P, RVS3-5*3P VDE TYPE

GPW-010	Power Cord VCTF 2.0mm2/3C, 3m Max Length, 105oC, RVS2-5*3P, RVS2-5*3P PSE TYPE
GPW-011	Power Cord SJT 10AWG/5C, 3m Max Length, 105oC, RV5-5*5P, RV5-5*5P UL TYPE
GPW-012	Power Cord H05VV-F 2.5mm2/5C, 3m Max Length, 105oC, RVS3-5*5P, RVS3-5*5P VDE TYPE
GPW-013	Power Cord VCTF 2.0mm2/5C, 3m Max Length, 105oC, , RVS2-5*5P, RVS2-5*5P PSE TYPE
GPW-014	Power Cord SJT 10AWG/4C, 3m Max Length, 105oC, RV5-5*4P, RV5-5*4P UL TYPE
GPW-015	Power Cord H05VV-F 2.5mm2/4C, 3m Max Length, 105oC, RVS3-5*4P, RVS3-5*4P VDE TYPE
GPW-016	Power Cord VCTF 2.0mm2/4C, 3m Max Length, 105oC, , RVS2-5*4P, RVS2-5*4P PSE TYPE
GTL-232	RS232C cable, approx. 2M
GTL-248	GPIB cable, approx. 2M
ASR-003	GPIB interface card
ASR-004	DeviceNet interface card
ASR-005	CAN BUS interface card
ASR-006	External parallel cable

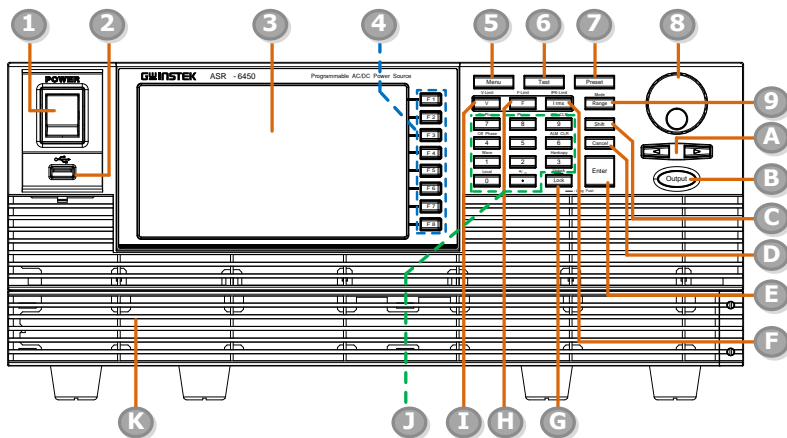


Note

- GPW-008, 009, 010 are for single phase input only.
- GPW-011, 012, 013 are for Y connection input only.
- GPW-014, 015, 016 are for Delta connection input only.

# Appearance

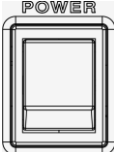
## Front Panel





Item Index	Description
1	Power switch button
2	USB interface connector (A Type)
3	LCD screen
4	Function keys (blue zone)
5	Menu key
6	Test key
7	Preset key
8	Scroll wheel
9	Range key/Output mode key
A	Arrow keys
B	Output key
C	Shift key


D	Cancel key
E	Enter key
F	Irms/IPK-Limit button
G	Lock/Unlock button
H	F/F-Limit button
I	V/V-Limit button
J	Numerical Keypad with additional “Shift + key” shortcut functions (green zone)
K	Air inlet

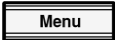


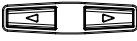
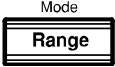

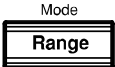

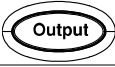



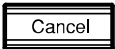
Item	Description
------	-------------









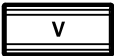

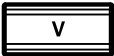


Power Switch		Turn on the mains power
--------------	---	-------------------------

USB A Port		<p>The USB port is used for data transfers and upgrading software. Also, it is available for screenshot hardcopy.</p> <p> It supports FAT32 format with maximum 32G storage.</p>
------------	---	---


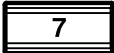



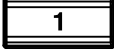

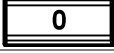

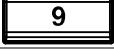
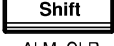
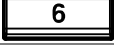



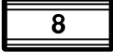
LCD Screen	Displays the setting and measured values or menu system
------------	---

Function Keys		Assigned to the functions displayed on the right side of the screen.
---------------	---	--

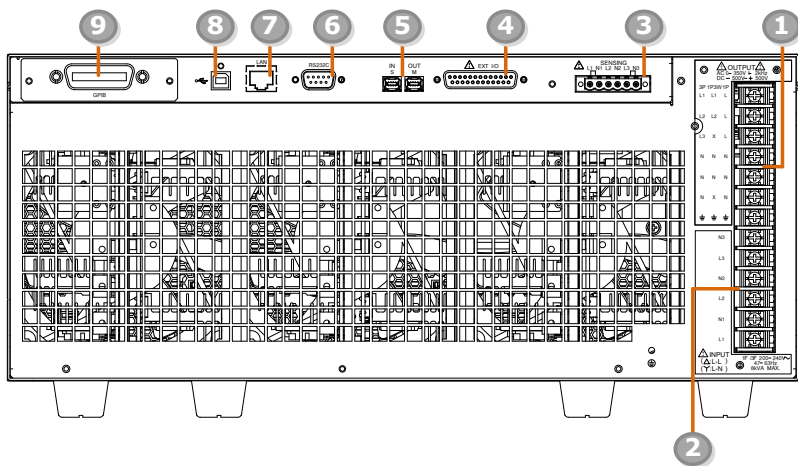
Menu Key		Enters the Main menu or goes back to one of the display modes.
Test Key		Puts the instrument into the Sequence and Simulation control mode.
Preset Key		Puts the instrument into Preset mode.
Arrow Keys		The arrow keys are used to select the digit power of a value that is being edited.
Range Key		Switches between the 100V, 200V and AUTO ranges
Output Mode	 + 	Selects between the AC+DC-INT, AC-INT, DC-INT, AC+DC-EXT, AC-EXT, AC+DC-ADD, AC-ADD, AC+DC-Sync, AC-Sync and AC-VCA modes.
Scroll Wheel		Used to navigate menu items or for increment/decrement values one step at a time.
Output Key		Turns the output on or off.
Shift Key		Turns on the shift state, which enables shortcut operations with an icon  indicated on the top status bar. The shift state, which allows continuous shortcut operations, is kept until another press on shift key again.
		When performing shortcut operations, press shift key followed by another shortcut function key. Do Not press both shift key and shortcut function key simultaneously.
Cancel Key		Used to cancel function setting menus or dialogs.

Enter Key		Confirms selections and settings.
Irms	<p style="text-align: center;">IPK-Limit</p> 	Used for setting the maximum output current.
IPK-Limit	<p style="text-align: center;">Shift</p> <p style="text-align: center;">+ IPK-Limit</p>  <p style="text-align: center;">Irms</p> 	Used to set the peak output current limit value.
Lock/Unlock Key	<p style="text-align: center;">Unlock</p>  <p style="text-align: center;">Lock</p> <p style="text-align: center;">— : Long Push</p>	Used to lock or unlock the front panel keys except output key. Simply press to lock, whilst long press to unlock.
F	<p style="text-align: center;">F-Limit</p> 	Used for setting the output frequency (DC mode N/A).
F-Limit	<p style="text-align: center;">Shift</p> <p style="text-align: center;">+ F-Limit</p>  <p style="text-align: center;">F</p> 	Used for setting the output frequency limit value (DC mode N/A).
V	<p style="text-align: center;">V-Limit</p> 	Used for setting the output voltage.
V-Limit	<p style="text-align: center;">Shift</p> <p style="text-align: center;">+ V-Limit</p>  <p style="text-align: center;">V</p> 	Used for setting the output voltage limit value.
Keypad		Used to input power of a value directly. The  key is used to input decimal / plus or minus.

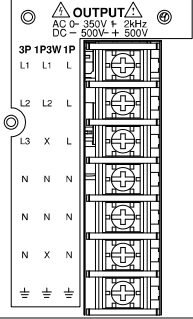
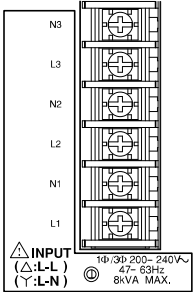
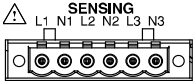
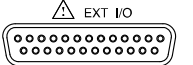
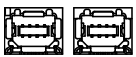


On Phase	 <small>On Phase</small>	+ Sets the on phase for the output voltage.
		
<hr/>		
Off Phase	 <small>Off Phase</small>	+ Sets the off phase for the output voltage.
		
<hr/>		
Output Waveform	 <small>Wave</small>	+ Selects between the Sine, Square, Triangle and ARB 1~253 waveforms (not available for DC-INT, AC+DC-EXT and AC-EXT).
		
<hr/>		
Local Mode	 <small>Local</small>	+ Switches operation back to local mode from remote mode.
		
<hr/>		
IPK CLR	 <small>IPK CLR</small>	+ Used to clear peak output current value.
		
<hr/>		
ALM CLR	 <small>ALM CLR</small>	+ Clears alarms.
		
<hr/>		
Hardcopy Key	 <small>Hardcopy</small>	+ Used to take a screenshot. Make sure an USB flash disk is well inserted before the action.
		
<hr/>		
Output Phase	 <small>Phase</small>	+ Used to switch output phase sequentially between 1P2W, 1P3W and 3P4W in a cyclic manner.
		

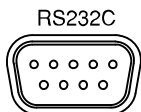
Rear Panel



Item Index	Description
1	Output terminal
2	AC power input terminal
3	Remote sensing input terminal
4	External I/O connector
5	External IN/OUT connection in parallel function
6	RS232 connector
7	Ethernet (LAN) connector
8	USB interface connector (B Type)
9	Optional interface Slot <ul style="list-style-type: none"> <li>▪ GPIB card (ASR-003)</li> <li>▪ DeviceNet card (ASR-004)</li> <li>▪ CAN BUS card (ASR-005)</li> </ul>

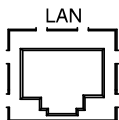
Item	Description
Output Terminal	<div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  <p style="font-size: small; margin-top: 5px;">                     ⚠ <b>OUTPUT</b> ⚠                      AC 0-350V ± 20Hz                      DC 0-500V ± 500V                 </p> </div> <div style="flex: 2; padding-left: 10px;"> <p>Output terminal (M4 screw type, 8 ~ 18 AWG) (Screw torque value:18kgf-cm)</p> </div> </div>
AC Power Input Terminal	<div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  <p style="font-size: small; margin-top: 5px;">                     ⚠ <b>INPUT</b> ⚠                      (Δ:L-L)                      (Y:L-N)                 </p> <p style="font-size: x-small; margin-top: 5px;">                     1Φ/3Φ 200-240V~                      47-63Hz                      8kVA MAX.                 </p> </div> <div style="flex: 2; padding-left: 10px;"> <p>AC inlet (M4 screw type, 8 ~ 18 AWG) (Screw torque value:18kgf-cm)</p> </div> </div>
Remote Sensing Input Terminal	<div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  <p style="font-size: small; margin-top: 5px;">                     ⚠ <b>SENSING</b> ⚠                      L1 N1 L2 N2 L3 N3                 </p> </div> <div style="flex: 2; padding-left: 10px;"> <p>Remote sensing input terminal is for compensation of load wire voltage drop. (M2.5 screw type, 12 ~ 30 AWG) (Screw torque value: 0.5N*m) (Strip length: 7 ~ 8mm)</p> </div> </div>
External Control I/O Connector	<div style="display: flex; align-items: center;"> <div style="flex: 1;">  <p style="font-size: small; margin-top: 5px;">                     ⚠ EXT I/O ⚠                 </p> </div> <div style="flex: 2; padding-left: 10px;"> <p>Used to control ASR-6000 externally by using the logic signal and monitor Sequence function status.</p> </div> </div>
External IN/OUT Connection in Parallel Function	<div style="display: flex; align-items: center;"> <div style="flex: 1;"> <div style="display: flex; justify-content: space-around; font-size: small; margin-bottom: 5px;"> <span>IN</span> <span>OUT</span> </div> <div style="display: flex; justify-content: space-around; font-size: x-small; margin-bottom: 5px;"> <span>S</span> <span>M</span> </div>  </div> <div style="flex: 2; padding-left: 10px;"> <p>The IN (Slave) and OUT (Master) ports are used for connection with external unit in parallel function.</p> </div> </div>

RS232C  
Connector



The RS232C connector for controlling the ASR-6000 remotely.

Ethernet LAN  
Port



The Ethernet port is used for remote control.

USB B-type Port



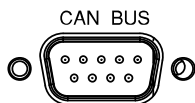
USB port for controlling the ASR-6000 remotely.

Optional GPIB  
Connector



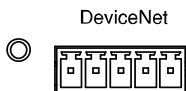
The optional GPIB connector for controlling the ASR-6000 remotely.

Optional CAN  
BUS Connector



The optional CAN BUS connector for controlling the ASR-6000 remotely.

Optional  
DeviceNet  
Connector



The optional DeviceNet connector for controlling the ASR-6000 remotely.

# D DEVICENET

This chapter describes DeviceNet Communication.

---

<b>Communication Introduction .....</b>	<b>22</b>
<b>Object Model.....</b>	<b>26</b>
Object Address Code .....	27
<b>Duplicate MAC ID Check Message Protocol .....</b>	<b>29</b>
<b>Connection Establishment Protocol .....</b>	<b>32</b>
<b>Explicit Message Protocol .....</b>	<b>36</b>
Error Response Explicit Message.....	42
Explicit Fragmentation Message .....	44

## Communication Introduction

DeviceNet is one of the kind of the network protocol base on the CAN bus. It allows direct peer to peer data exchange between nodes in organized and deterministic manner. The network managment function specified in DeviceNet simplifies project design, implementation and dianosis by providing standard mechanism for network start-up and error management.

DeviceNet commnuciation protocol is base on the concept of connection. Must establish a connection with a device in orer to exclude information with device. Each getway implement predefined master or slave connection setting through the DeviceNet network.

Identifier Bits											Hex Range	Identity Usage
10	9	8	7	6	5	4	3	2	1	0		
0		Group 1 Message ID				Source MAC ID				000 - 3FF		Message Group 1
1	0	MAC ID				Group 2 Message ID				400 - 5FF		Message Group 2
1	1	Group 3 Message ID				Source MAC ID				600 - 7BF		Message Group 3
1	1	1	1	1	Group 4 Message ID				7C0 - 7EF		Message Group 4	
1	1	1	1	1	1	1	X	X	X	X	7F0 - 7FF	Invalid CAN ID

DeviceNet define four separate groups of 11-bits CAN identifiers. Message Group1, 2 and 3 consist of two fields, 6-bits field for Media Access Code Identifier(MAC ID) and the other for Message ID. Message Group 4 is used offline communication. Message ID is respected to connect base on message. Because of the arbitration scheme defined by CAN, group 1 message has a higher priority than group 2 message, group 2 message has a higher priority than group 3 message and so on.

The four Message Groups are used as follows:

Identifier Bits										Identity Usage	
10	9	8	7	6	5	4	3	2	1	0	
0	Group 1 Message ID			Source MAC ID						Message Group 1(0x000 - 0x3FF)	
0	0	0	0	0							
0	0	0	0	0						1	
0	0	0	0	1						0	
0	0	0	0	1						1	
0	0	0	1	0						0	
0	0	0	1	0						1	
0	0	0	1	1						0	
0	0	0	1	1						1	
0	1	0	0	0	Source MAC ID						
0	1	0	0	0						1	
0	1	0	0	1						0	
0	1	0	0	1						1	
0	1	1	0	0						0	Slave I/O Multicast Poll Response Message
0	1	1	0	1						1	Slave I/O Change of State / Cycle Message
0	1	1	1	0						0	Slave I/O Bit-Strobe Response Message
0	1	1	1	1						1	Slave I/O Poll Response Message / Change of State / Cycle Acknowledge Message

Message Group 1 is assigned 1024 CAN Identifiers (0x000 - 0x3FF) . Up to 16 different Message IDs are available per device within this group. Usually used to I/O message application data communicated.

Identifier Bits										Identity Usage	
1	9	8	7	6	5	4	3	2	1	0	
0											
1	0	MAC ID					Group 2 Message ID			Message Group 2(0x400 - 0x5FF)	
1	0	Source MAC ID					0	0	0	Master I/O Bit-Strobe Command Message	
1	0	Source MAC ID					0	0	1	Master I/O Multicast Poll command Message	
1	0	Destination MAC ID					0	1	0	Master I/O Change of State / Cycle Acknowledge Message	
1	0	Source MAC ID					0	1	1	Slave Explicit Response Message/ Unconnected Response Message / Device Shutdown Message	
1	0	Destination MAC ID					1	0	0	Master Explicit Request Message	
1	0	Destination MAC ID					1	0	1	Master I/O Poll Command Message / Change of State / Cycle Message	
1	0	Destination MAC ID					1	1	0	Group 2 Only Unconnected Explicit Request Message	
1	0	Destination MAC ID					1	1	1	Duplicate MAC ID Check Message	

Message Group 2 is assigned 512 CAN identifiers (0x400 – 0x5FF). Up to 8 different Message IDs are available per device within this group. Most of the message IDs in this group are optionally defined for what is commonly referred to as the predefined master/slave connection set. One message ID is reserved for the Duplicate Node ID Check.

Identifier Bits										Identity Usage	
10	9	8	7	6	5	4	3	2	1	0	
1	1	Group 3 Message ID		Source MAC ID							Message Group 3(0x600 - 0x7BF)
1	1	0	0	0							
1	1	0	0	1							



1	1	0	1	0		
1	1	0	1	1		
1	1	1	0	0		
1	1	1	0	1		UCMM Response Message
1	1	1	1	0		UCMM Request Message
1	1	1	1	1	XXXXXXX	Not Supported

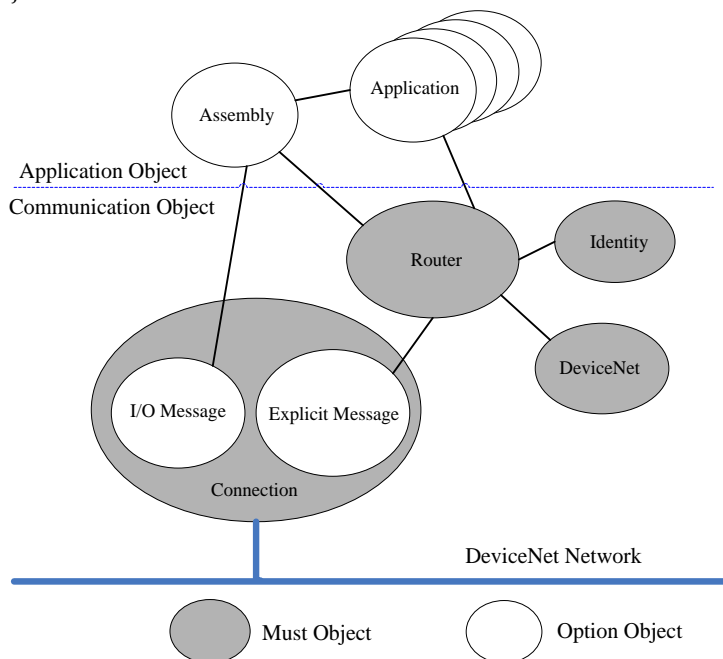
Message Group 3 is assigned 448 CAN identifiers (0x600 - 0x7BF). Up to 7 different Message IDs are available per device within this group. The Main use of this group is setting up dynamic explicit connection. Two message IDs are reserved for the Unconnected Message Manager(UCMM) port.

Identifier Bits											Identity Usage	
10	9	8	7	6	5	4	3	2	1	0		
1	1	1	1	1	Group 4 Message ID							Message Group 4(0x7C0 - 0x7EF)
1	1	1	1	1	1	0	1	1	0	0	Communication Faulted Response Message	
1	1	1	1	1	1	0	1	1	0	1	Communication Faulted Request Message	
1	1	1	1	1	1	0	1	1	1	0	Offline Ownership Response Message	
1	1	1	1	1	1	0	1	1	1	1	Offline Ownership Request Message	

Message Group 4 is assigned 48 CAN Identifiers (0x07C0 - 0x07EF). It does not include any MAC IDs, only Message IDs. The messages in this group are only used for network management. Four Message IDs are currently assigned for services of the Offline Connection Set.

## Object Model

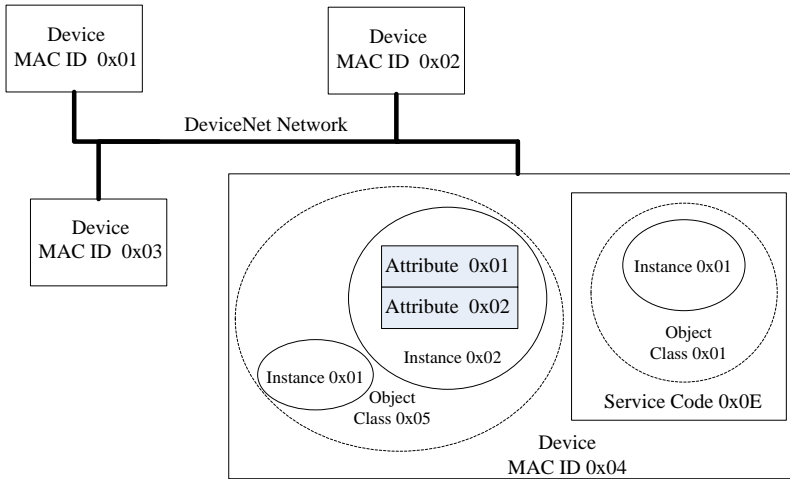
A device within DeviceNet network is represented by the below object model.



DeviceNet is used the object model to describe the network how to establish and manage device communication. Each device consists of communication object and application object. A DeviceNet can define the object and all of typical device objects contains Identity, Message Router, DeviceNet, Assembly, Connection and Application object. Device communication used Class, Instance, Attribute, Service to describe the structure encoding.

## Object Address Code

The object address described Media Access Code Identifier(MAC ID), Class ID, Instance ID, Attribute ID and Service Code from DeviceNet portocol specification for each device.



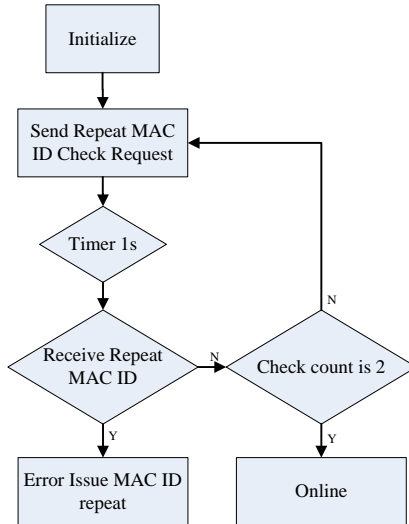
- **MAC ID:** integer identification value assigned to each node on the DeviceNet network.
- **Class ID:** integer identification value assigned to each object class accessible from the network.
- **Instance ID:** integer identification value assigned to object instance than identifies it among all instance of the same class.
- **Attribute ID:** integer identification value assigned to class or instance attribute value.
- **Service Code:** integer identification value which particular object instance or object class function.

DeviceNet protocol define the range for address in the specification.

<b>Object</b>	<b>Range</b>	<b>Identity Usage</b>
MAC ID	0X00 - 0x3F	If MAC ID is assigned other value then default value is 0x3F
	0x00 - 0x63	Open specification by DeviceNet define
Class	0x64 - 0xC7	Vendor specification
	0xC8 - 0xFF	Resvered for future use
	0x00 - 0x31	Open specification by DeviceNet define
	0x32 - 0x4A	Vendor specification
Service Code	0x4B - 0x63	Object class specification by DeviceNet define
	0x64 - 0x7F	Resvered for future use
	0x80 - 0xFF	None used

## Duplicate MAC ID Check Message Protocol

Each physical attachment to DeviceNet must be assigned a MAC ID. It is inevitable that all of modules on the same link network will be assigned the same MAC ID, therefore all DeviceNet modules are required to participate in a duplicate MAC ID check algorithm.



When each node connects to the network, it need to send the MAC ID request check twice at interval of 1 second and wait for 1 second after send a request message. If the node does not receive a duplicate response message within the period, then going online successful and become normal sending and receiving message.

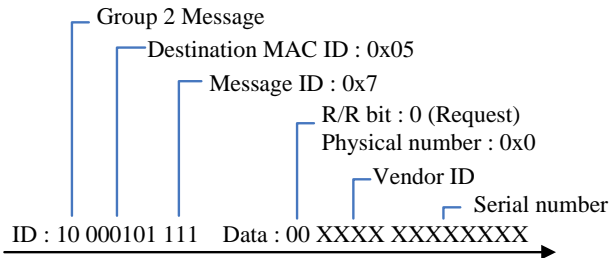
It is found that the assigned MAC ID has been used on the network, sending duplicate response message and device enter a communications fault state.

Identifier Bits										Identity Usage		
10	9	8	7	6	5	4	3	2	1	0		
1	0	MAC ID					Group 2 Message ID			Message Group 2(0x400 - 0x5FF)		
1	0	Destination MAC ID					1	1	1	Duplicate MAC ID Check Message		

		Data Bits							
Byte Offset		7	6	5	4	3	2	1	0
0	R/R	Physical Number							
1		Vendor ID(LSB)							
2		Vendor ID(MSB)							
3		Serial Number(Low)							
4		Serial Number(Mid)							
5		Serial Number(High)							
6		Serial Number(Highest)							

- Description
- R/R: Request(0)/Response(1) message flag.
  - Physical Number: DeviceNet assigned to each physical identification value. Multiple device physical connector must assign each attachment a value within the range of 0-127 decimal. Implement a single connector should set zero within the field.
  - Vendor ID: Assigned to the vendor of the device that is transmitting the message.
  - Serial Number: Assigned to the device by the vendor.

CAN bus  
frame  
message  
example



## Connection Establishment Protocol

DeviceNet device connection is established. Used to the Group 2 Only Unconnected Explicit Request port which is defined by the predefined master/slave connection set. Every device can set up a connection with every other device, and the source MAC ID of the devices is contained in the connection ID.

The predefined master/slave connection set defines a way to establish connections called the Group 2 Only Unconnected Explicit Request Port. This method allows a device to limit the messages received to only those in Group 2 with its own MAC ID. Devices that operate are referred to as Group 2 Only Servers, they are only required to receive messages in Group 2.

There are two type of messages to predefined connection objects, Explicit Message Connection and I/O Message Connection(not supported the connection establishment). Explicit message is defined in Message Group 2.

Identifier Bits										Identity Usage			
1	9	8	7	6	5	4	3	2	1	0			
1	0	MAC ID						Group 2 Message ID		Message Group 2(0x400 - 0x5FF)			
1	0	Source MAC ID						0	1	1	Slave Explicit Response Message		
1	0	Destination MAC ID						1	1	0	Group 2 Only Unconnected Explicit Request Message		



Connection request data	Data Bits							
	Byte Offset	7	6	5	4	3	2	1
0	Frag	XID	Source MAC ID					
1	R/R	Service Code(0x4B)						
2	Class ID(0x03)							
3	Instance ID(0x01)							
4	Allocation Choice							
5	x	x	Source MAC ID					

- Description
- Frag: Non-fragment(0)/Fragment(1) flag. Indicating whether or not this transmission is a fragment piece of an Explicit Message.
  - XID: Used an application to match a response with the associated request.
  - Source MAC ID: Device MAC ID.
  - R/R: Request(0)/Response(1) message flag.
  - Service Code: By the DeviceNet protocol define.
  - Allocation Choice: Predefined connection is to be allocated.(Only supported Explicit Message connection)

Allocation Choice Data Bits							
7	6	5	4	3	2	1	0
Reserved	Ack Suppression	Cyclic	COS	Multicast Polling	Bit-Strobe	Poll	Explicit Message

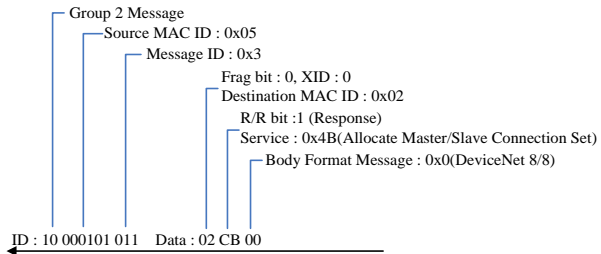
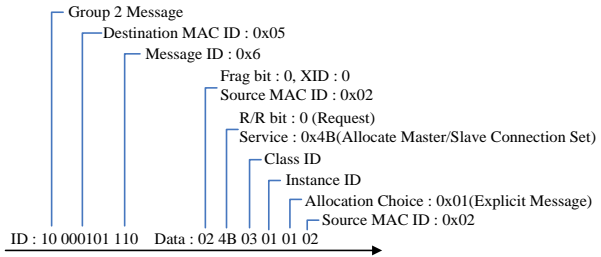
Connection response data	Byte Offset	Data Bits							
		7	6	5	4	3	2	1	0
	0	Frag	XID	Destination MAC ID					
	1	R/R	Service Code(0x4B)						
	2	Body Format Message(0x00)							

- Description
- Frag: Non-fragment(0)/Fragment(1) flag. Indicating whether or not this transmission is a fragment piece of an Explicit Message.
  - XID: Used an application to match a response with the associated request.
  - Destination MAC ID: Device MAC ID.
  - R/R: Request(0)/Response(1) message flag.
  - Service Code: By the DeviceNet protocol define.
  - Body Format Message: Request a particular message for subsequent Explicit Message transmitted.(Only Supported DeviceNet(8 / 8))

Body Fromat Message Data	
Value	Identity
0x00	DeviceNet(8 / 8). Class is 8-bit integer. Instance ID is 8-bit integer.
0x01	DeviceNet(8 / 16). Class is 8-bit integer. Instance ID is 16-bit integer.
0x02	DeviceNet(16 / 16). Class is 16-bit integer. Instance ID is 16-bit integer.
0x03	DeviceNet(16 / 8). Class is 16-bit integer. Instance ID is 8-bit integer.
0x04 - 0x0F	Reserved by DeviceNet

CAN bus  
frame  
message  
example

Master(MAC ID = 0x02),  
Slave(MAC ID = 0x05)



## Explicit Message Protocol

---

All of explicit messages are done via connection and the associated connection object instance. This can use the predefined master/slave connection set to activate a connection object in the device. The explicit message by the Group 2 Only Unconnected Port is connected object.

Explicit messages on the DeviceNet have a compact structure to make them fit into the 8-byte frame in most cases. The example of request message using the 8/8 Message Body Format.(Connection response DeviceNet Message Body Format is 8/8. Meas 1 byte for Class ID, 1 byte for Instance ID.)

Explicit message request data no including subindex	Data Bits								
	Byte Offset	7	6	5	4	3	2	1	0
	0	Frag	XID	Source MAC ID					
	1	R/R	Service Code						
	2	Class ID							
	3	Instance ID							
	4	Attribute ID							
	5								
	6	Service Data(Optional)							
	7								

- Description
- Frag: Non-fragment(0)/Fragment(1) flag. Indicating whether or not this transmission is a fragment piece of an Explicit Message.
  - XID: Used an application to match a response with the associated request.
  - Source MAC ID: Device MAC ID.
  - R/R: Request(0)/Response(1) message flag.
  - Service Code: By the DeviceNet protocol define.
  - Class ID: Defines the object class towards which this request is directed.
  - Instance ID: Defines the particular instance within the object class towards which this request is directed.
  - Attribute ID: Defines the particular attribute with the object class and instance towards which this request is directed.
  - Service Data: Request data by specific object define.

Explicit message request data including subindex	Data Bits							
	Byte Offset	7	6	5	4	3	2	1
0	Frag	XID	Source MAC ID					
1	R/R	Service Code						
2	Class ID							
3	Instance ID							
4	Attribute ID							
5	SubIndex ID							
6	Service Data(Optional)							
7								

- Description
- **Frag:** Non-fragment(0)/Fragment(1) flag. Indicating whether or not this transmission is a fragment piece of an Explicit Message.
  - **XID:** Used an application to match a response with the associated request.
  - **Source MAC ID:** Device MAC ID.
  - **R/R:** Request(0)/Response(1) message flag.
  - **Service Code:** By the DeviceNet protocol define.
  - **Class ID:** Defines the object class towards which this request is directed.
  - **Instance ID:** Defines the particular instance within the object class towards which this request is directed.
  - **Attribute ID:** Defines the particular attribute whith the object class and instance towards which this request is directed.
  - **SubIndex ID:** Defines the particular subindex whith the object class, instance and attribute towards which this request is directed.
  - **Service Data:** Request data by specific object define.

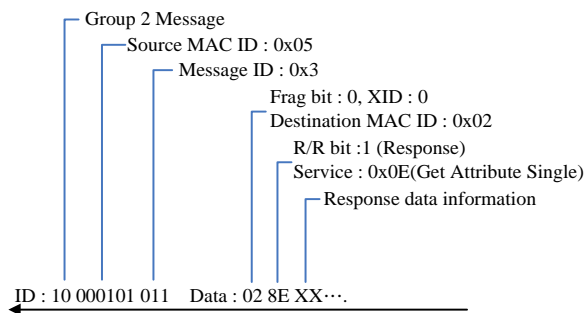
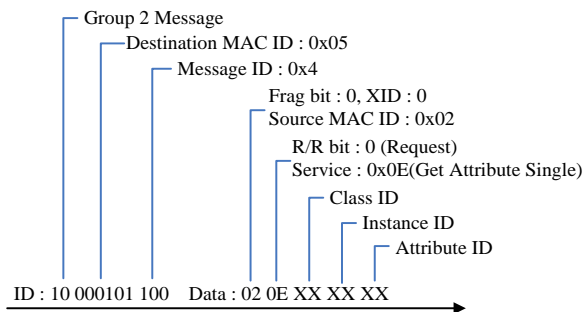
Explicit message response data	Data Bits								
	Byte Offset	7	6	5	4	3	2	1	0
	0	Frag	XID	Source MAC ID					
	1	R/R	Service Code						
	2								
	3								
	4	Service Data(Optional)							
	5								
	6								
	7								

- Description
- Frag: Non-fragment(0)/Fragment(1) flag. Indicating whether or not this transmission is a fragment piece of an Explicit Message.
  - XID: Used an application to match a response with the associated request.
  - Source MAC ID: Device MAC ID.
  - R/R: Request(0)/Response(1) message flag.
  - Service Code: By the DeviceNet protocol define.
  - Service Data: Response data by specific object define.which this request is directed.
  - Service Data: Request data by specific object define.

CAN bus frame message example

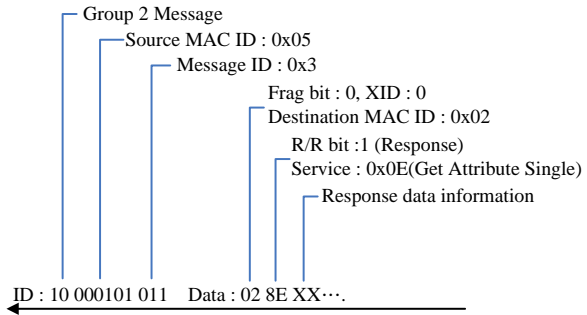
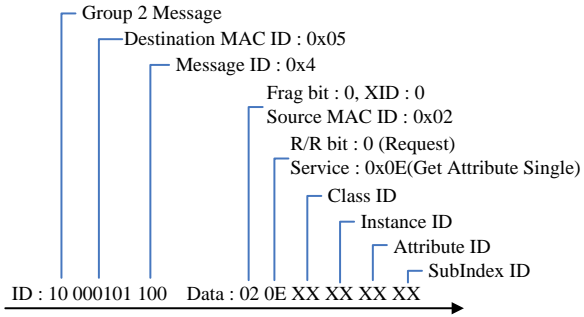
Identifier Bits										Identity Usage			
1	0	9	8	7	6	5	4	3	2	1	0		
1	0	MAC ID					Group 2 Message ID			Message Group 2(0x400 - 0x5FF)			
1	0	Source MAC ID					0	1	1	Slave Explicit Response Message			
1	0	Destination MAC ID					1	0	0	Master Explicit Request Message			

Master(MAC ID = 0x02 and no subindex), Slave(MAC ID = 0x05)





Master(MAC ID = 0x02 and include subindex), Slave(MAC ID = 0x05)



## Error Response Explicit Message

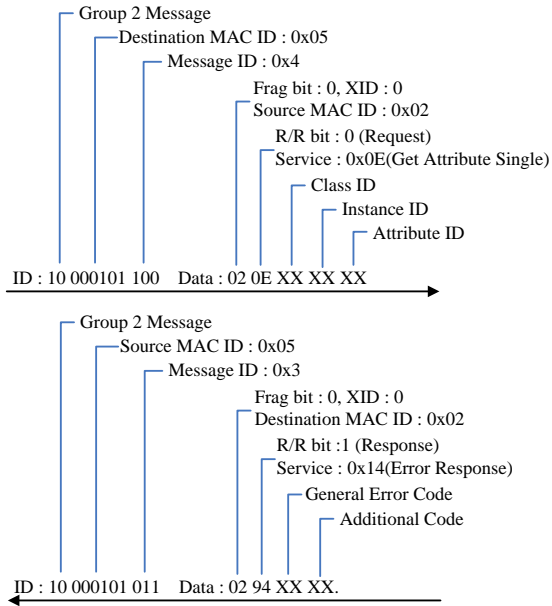
The Error Response Explicit Message is returned when an error is encountered while attempting to service a previously received Explicit Request Message. The Error Response can be sent as either a Connection Based or Unconnected response message. The Service Code for an Error Response message is 0x14, and two bytes of error code are included in the service data field to convey more information about the nature of the error.

Error response explicit message data	Byte Offset	Data Bits							
		7	6	5	4	3	2	1	0
	0	Frag	XID	Source MAC ID					
	1	R/R	Service Code (0x14)						
	2	General Error Code							
	3	Additional Code							

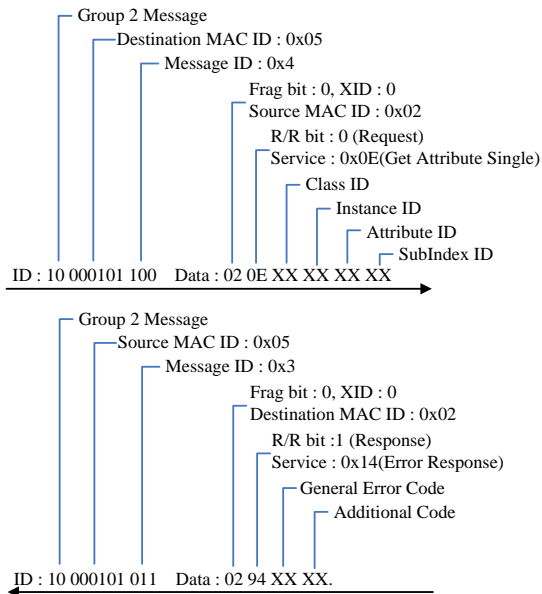
- Description
- Frag: Non-fragment(0)/Fragment(1) flag. Indicating whether or not this transmission is a fragment piece of an Explicit Message.
  - XID: Used an application to match a response with the associated request.
  - Source MAC ID: Device MAC ID.
  - R/R: Request(0)/Response(1) message flag.
  - Service Code: By the DeviceNet protocol define.
  - General Error Code: Identifies the encountered error.
  - Additional Code: Further describes the error condition. If responded object has no additional information to define, then the value default is 0xFF within this field.define.

CAN bus  
frame  
message  
example

Master(MA  
C ID = 0x02  
and no  
include  
subindex),  
Slave(MAC  
ID = 0x05)



Master(MA  
C ID = 0x02  
and include  
subindex),  
Slave(MAC  
ID = 0x05)



## Explicit Fragmentation Message

Fragmentation of an Explicit Message is performed in an Acknowledged fashion. Acknowledged fragmentation consists of the transmission fragmentation from the transmitting module followed by the transmission of an acknowledgment by the receiving module. The receiving module acknowledges the reception of each fragment.

Explicit Fragmentation on message data	Byte Offset	Data Bits							
		7	6	5	4	3	2	1	0
	0	Frag	XID	Source MAC ID					
	1	Fragment Type		Fragment Count					
	2								
	3								
	4	Explicit Fragment Message Body							
	5								
	6								
	7								

- Description
- Frag: Non-fragment(0)/Fragment(1) flag. Indicating whether or not this transmission is a fragment piece of an Explicit Message.
  - XID: Used an application to match a response with the associated request.
  - Source MAC ID: Device MAC ID.
  - Fragment Type: Indicate that is the first, middle, or last transmission.
  - Fragment Count: Marks each separate fragment. Count is increased by one for each successful fragment in the series.

Acknowledged Fragmentation data	Byte Offset	Data Bits							
		7	6	5	4	3	2	1	0
	0	Frag	XID	Source MAC ID					
	1	Fragment Type		Fragment Count					
	2	Ack Status							

- Description
- Frag: Non-fragment(0)/Fragment(1) flag. Indicating whether or not this transmission is a fragment piece of an Explicit Message.
  - XID: Used an application to match a response with the associated request.
  - Source MAC ID: Device MAC ID.
  - Fragment Type: Indicate that is the Fragment Acknowledge.
  - Fragment Count: Echo the last count value received.
  - Ack Status: Indicates whether or not received error the fragment message.

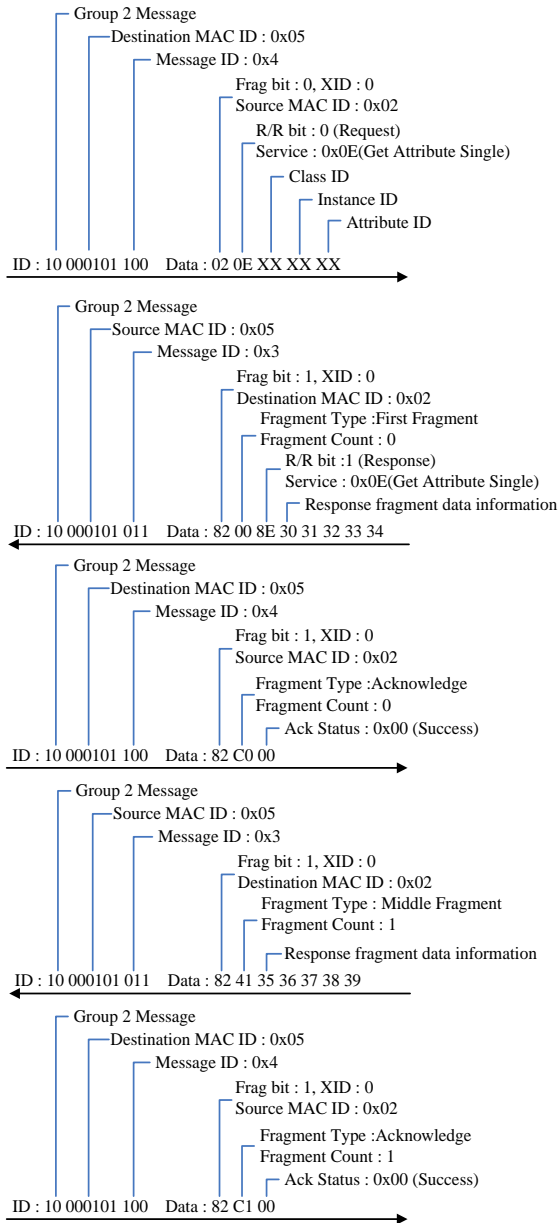
<b>Fragment Type</b>	
Value	Identity
0x0	First Fragment
0x1	Middle Fragment
0x2	Last Fragment
0x3	Fragment Acknowledge

<b>Fragmentation Ack Status</b>	
Value	Identity
0x00	Successful. No error has been detected and fragment transmission should continue.
0x01	Fail. The data is received over the maximum amount.
0x02 - 0xFF	Reserved by DeviceNet

CAN bus  
frame  
message  
example

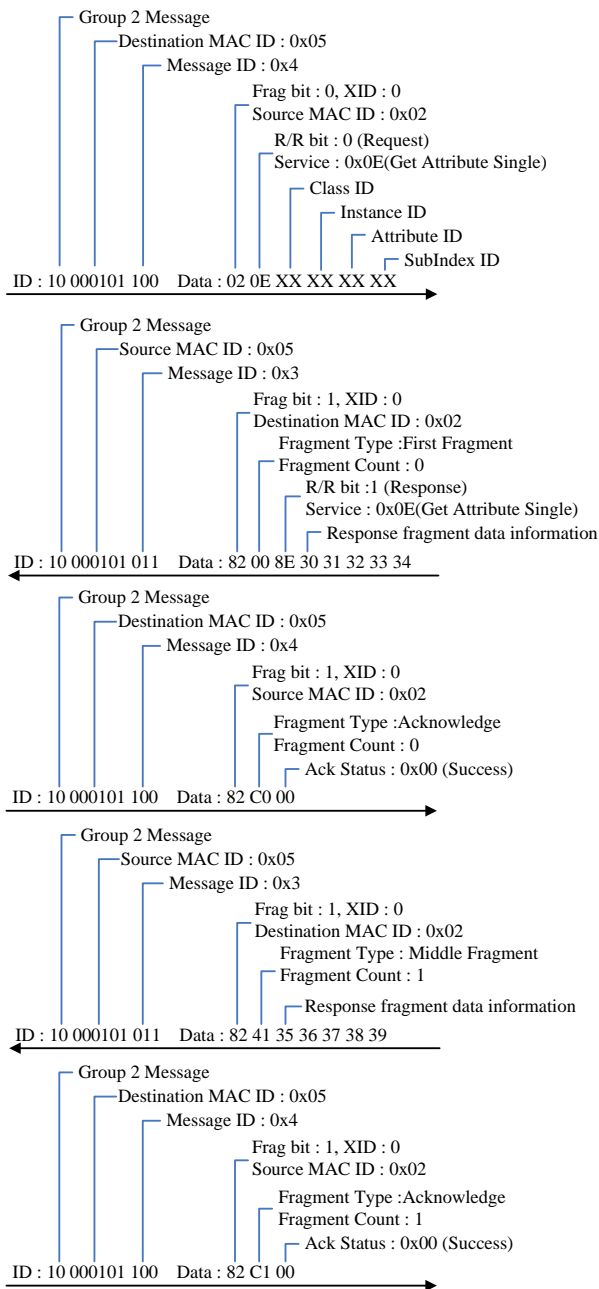
Master(MA  
C ID = 0x02  
and no  
include  
subindex),  
Slave(MAC  
ID = 0x05)

Get Slave  
Device Data  
0x30, 0x31,  
0x32, 0x33,  
0x34, 0x35,  
0x36, 0x37,  
0x38, 0x39



Master(MA  
C ID = 0x02  
and include  
subindex),  
Slave(MAC  
ID = 0x05)

Get Slave  
Device Data  
0x30, 0x31,  
0x32, 0x33,  
0x34, 0x35,  
0x36, 0x37,  
0x38, 0x39





# REMOTE CONTROL

This chapter describes basic configuration of DeviceNet-based remote control. The ASR-6000 can be programmed by a controller over DeviceNet.

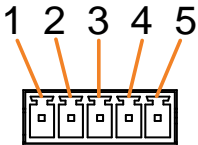
---

<b>DeviceNet Configuration .....</b>	<b>50</b>
DeviceNet Interface .....	50
Setting of the DeviceNet baud rate .....	51
Auto baud rate.....	51
CAN Bus termination resistance.....	52
Pin assignment.....	53
<b>Object Class Specification .....</b>	<b>54</b>
Identify Object (Class 0x01) .....	54
DeviceNet Object (Class 0x03) .....	55
Connection Object (Class 0x05) .....	56
Communication Command Object (Class 0x64) ..	58
<b>Manufacturer-specific Command List .....</b>	<b>82</b>
Request Read Data Range Commands Example	310
<b>Manufacturer-specific Command Example .....</b>	<b>311</b>
<b>DeviceNet Errors Code .....</b>	<b>320</b>

# DeviceNet Configuration

## DeviceNet Interface

DeviceNet Configuration	Connector Parameters	Block terminal, 5 pins, male Baudrate, MAC ID.
-------------------------	----------------------	---

Pin Assignment		1: 24V 2: CAN-H 3, 5: GND 4: CAN-L
----------------	---	---

Steps 1. Connect a DeviceNet cable from the PC to the DeviceNet port on the rear panel.

2. Press the *Menu* key. The Menu setting will appear on the display.



3. Use the scroll wheel to go to item 6, *Option Interface* and press *Enter*.

4. Set the DeviceNet relative settings.

Baudrate 125K(default), 250K, 500K, Auto

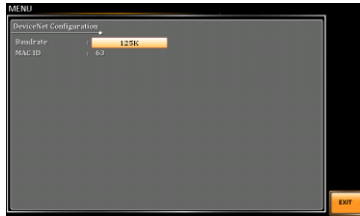
MAC ID 0 ~ 63 (63 is default value)

Exit 5. Press *Exit[F8]* to exit from the DeviceNet settings.



Example

### DeviceNet Configuration



## Setting of the DeviceNet baud rate

---

Depending on the set bit rate the length of the bus line is restricted. The specified max. bus length is an approximate value and also depends on the other devices on the bus.

Baud rate	Max. bus length
125K	500 m
250K	250 m
500K	100 m

## Auto baud rate

---

The CAN BUS of ASR6000 supports auto baud rate mode. When the Baud rate is switched to "Auto", the system will start to detect the baud rate mode. At this time, the master in the CAN BUS network must send more than 1 Byte of data. When ASR6000 receives after receiving the packet, the baud rate detection is completed and the system will automatically set the same baud rate as the master.

## CAN Bus termination resistance

---

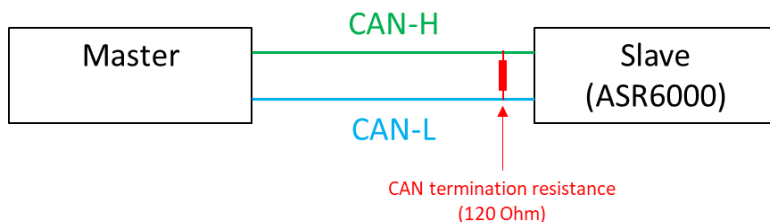
CAN Bus networks require a specific characteristic impedance of typically 120 Ohm to operate correctly. This characteristic impedance is maintained by placing termination resistors at both ends of the network, which effectively terminate the CAN signal and prevent signal reflections that can cause errors and signal corruption.

To add terminal resistance, you need to connect a 120 Ohm resistor between the CAN\_H and CAN\_L lines at each end of the network. The termination resistors are typically placed at the two ends of the bus network, which could be at the physical ends of the network or at the last two nodes in the network.

If the network has only two nodes, both nodes must have termination resistors. However, if there are more than two nodes, only the two nodes at the physical ends of the network require termination resistors.

To add a termination resistor to a CAN connector on ASR6000, you should connect the CAN connector on the ASR6000 through a CAN cable with the second last device in the network. Then, cover the CAN connector with a CAN terminal connector (with a 120 Ohm resistor between CAN-L and CAN-H) to provide proper termination for the network.

It's important to note that failure to add termination resistors can result in poor network performance and communication issues. So, it's essential to ensure proper termination of the network to maintain its integrity and reliable performance.



## Pin assignment

---

The CAN BUS port have the following pin assignments:

Pin	Signal
1	V+(+24V)
2	CAN-H
3	Not assigned
4	CAN-L
5	V-( CAN GND)

## Object Class Specification

The following table shows all objects of the class specification profile which are supported by the device:

### Identify Object (Class 0x01)

The object provides identification and general information.

#### Class Attributes

Attribute ID	Access Rule	Name	Data type
0x01	Get	Revision	UINT8
0x02	Get	Max Instance	UINT8

#### Instance Attributes

Attribute ID	Access Rule	Name	Data type
0x01	Get	Vendor Number	UINT8
0x02	Get	Device Type	UINT8
0x03	Get	Product Code Number	UINT16
0x04	Get	Product Major Revision Product Minor Revision	UINT16
0x05	Get	Status	UINT16
0x06	Get	Serial Number	UINT32
0x07	Get	Product Name	UINT16

#### Common Services

Service Code	Implemented for		Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single

## DeviceNet Object (Class 0x03)

The object provides for the configuration and status of DeviceNet port.

### Class Attributes

Attribute ID	Access Rule	Name	Data type
0x01	Get	Revision	UINT8
0x02	Get	Max Instance	UINT8

### Instance Attributes

Attribute ID	Access Rule	Name	Data type
0x01	Get	MAC ID	UINT8
0x02	Get	Baud Rate	UINT8
0x05	Get	Allocation Choice Byte	UINT8
		Master Node MAC ID	UINT8

### Common Services

Service Code	Implemented for		Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single

## Connection Object (Class 0x05)

The object provides explicit message connection to allocate and manage internal resources.

### Class Attributes

Attribute ID	Access Rule	Name	Data type
0x01	Get	Revision	UINT8
0x02	Get	Max Instance	UINT8

### Instance Attributes

Instance 0x01: Master / Slave explicit message connection

Attribute ID	Access Rule	Name	Data type
0x01	Get	State	UINT8
0x02	Get	Instance Type	UINT8
0x03	Get	TransportClass Trigger	UINT16
0x04	Get	Produced Connection ID	UINT16
0x05	Get	Consumed Connection ID	UINT16
0x06	Get	Initial Comm Characteristics	UINT8
0x07	Get	Produced Connection Size	UINT16
0x08	Get	Consumed Connection Size	UINT16
0x09	Get/Set	Expected Packed Rate	UINT16
0x0C	Get	Watch Dog Timeout Action	UINT8
0x0D	Get	Produced Connection Path Length	UINT8
0x0E	Get	Produced Connection Path	UINT8
0x0F	Get	Consumed Connection Path Length	UINT8
0x10	Get	Consumed Connection Path	UINT8



**Common Services**

<b>Service Code</b>	<b>Implemented for</b>		<b>Name</b>
	<b>Class</b>	<b>Instance</b>	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	No	Yes	Set_Attribute_Single

## Communication Command Object (Class 0x64)

---

The object provides application SCPI Command for ASR-6000 serials.

### Class Attributes

Attribute ID	Access Rule	Name	Data type
0x01	Get	Revision	UINT8

### Instance Attributes

#### Instance 0x01: IEEE Common Commands

Attribute ID	Access Rule	Name	Data type
0x02	Set	*CLS	UINT8
0x05	Get	*IDN?	Char
0x07	Set	*RCL	UINT8
0x08	Set	*RST	UINT8
0x09	Set	*SAV	UINT8
0x0E	Set	*WAI	UINT8

**Instance 0x03: Data Commands**

<b>Attribute ID</b>	<b>Sub Index</b>	<b>Access Rule</b>	<b>Name</b>	<b>Data type</b>
0x02		Set	DATA:SEQuence:CLEar	UINT8
0x04		Set	DATA:SEQuence:RECall	UINT8
0x05		Set	DATA:SEQuence:STORe	UINT8
0x06		Set	DATA:SIMulation:CLEa	UINT8
0x08		Set	DATA:SIMulation:RECall	UINT8
0x09		Set	DATA:SIMulation:STORe	UINT8
0x13		Set	DATA:WAVE:CLEar	UINT8
	0x0	Set	DATA:WAVE[:DATA]	UINT32
	0x1	Set	DATA:WAVE[:DATA]	UINT32
	0x2	Set	DATA:WAVE[:DATA]	UINT32
	0x3	Set	DATA:WAVE[:DATA]	UINT32
			.	
0x15			.	
			.	
	0xF9	Set	DATA:WAVE[:DATA]	UINT32
	0xFA	Set	DATA:WAVE[:DATA]	UINT32
	0xFB	Set	DATA:WAVE[:DATA]	UINT32
	0xFC	Set	DATA:WAVE[:DATA]	UINT32

**Instance 0x04: Display Commands**

Attribute ID	Sub Index	Access Rule	Name	Data type
0x01		Set	DISPlay[:WINDow]:DESIgn:MODE	UINT8
0x02	0x0	Get / Set	DISPlay[:WINDow]:MEASure:SOURce	UINT8
	0x1	Get / Set	DISPlay[:WINDow]:MEASure:SOURce	UINT8
	0x2	Get / Set	DISPlay[:WINDow]:MEASure:SOURce	UINT8
	0x3	Get / Set	DISPlay[:WINDow]:MEASure:SOURce	UINT8
0x10		Set	DISPLAY_ADDRESS	UINT8

**Instance 0x06: FetCh Commands**

Attribute ID	Sub Index	Access Rule	Name	Data type
0x01		Get	FETCh[:SCALar]:CURRent:CFACtor	INT32
0x02		Get	FETCh[:SCALar]:CURRent:HIGH	INT32
0x03		Get	FETCh[:SCALar]:CURRent:LOW	INT32
0x04		Get	FETCh[:SCALar]:CURRent:PEAK:HOLD	INT32
0x05		Get	FETCh[:SCALar]:CURRent[:RMS]	INT32
0x06		Get	FETCh[:SCALar]:CURRent[:RMS]:TOTal	INT32
0x07		Get	FETCh[:SCALar]:CURRent:AC	INT32
0x08		Get	FETCh[:SCALar]:CURRent:AVERAge	INT32
0x09		Get	FETCh[:SCALar]:FREQuency	INT32
0x0A		Get	FETCh[:SCALar]:POWER[:AC]:APParent	INT32

Attribute ID	Sub Index	Access Rule	Name	Data type
0x0B		Get	FETCh[:SCALar]:POWer[:AC]:APParent:TOTal	INT32
0x0C		Get	FETCh[:SCALar]:POWer[:AC]:PFACTOR	INT32
0x0D		Get	FETCh[:SCALar]:POWer[:AC]:PFACTOR:TOTal	INT32
0x0E		Get	FETCh[:SCALar]:POWer[:AC]:REACTIVE	INT32
0x0F		Get	FETCh[:SCALar]:POWer[:AC]:REACTIVE:TOTal	INT32
0x10		Get	FETCh[:SCALar]:POWer[:AC][:REAL]	INT32
0x11		Get	FETCh[:SCALar]:POWer[:AC][:REAL]:TOTal	INT32
0x12		Get	FETCh[:SCALar]:VOLTag e[:RMS]	INT32
0x13		Get	FETCh[:SCALar]:VOLTag e[:RMS]:TOTal	INT32
0x14		Get	FETCh[:SCALar]:VOLTag e:AC	INT32
0x15		Get	FETCh[:SCALar]:VOLTag e:AVERAge	INT32
0x16		Get	FETCh[:SCALar]:VOLTag e:HIGH	INT32
0x17		Get	FETCh[:SCALar]:VOLTag e:LOW	INT32
0x18	0x0	Get	FETCh[:SCALar]:CURRent:HARMonic[:RMS]	INT32
	0x1	Get	FETCh[:SCALar]:CURRent:HARMonic[:RMS]	INT32
	0x2	Get	FETCh[:SCALar]:CURRent:HARMonic[:RMS]	INT32
	0x3	Get	FETCh[:SCALar]:CURRent:HARMonic[:RMS]	INT32
	0x4	Get	FETCh[:SCALar]:CURRent:HARMonic[:RMS]	INT32
	0x5	Get	FETCh[:SCALar]:CURRent:HARMonic[:RMS]	INT32
0x19	0x0	Get	FETCh[:SCALar]:CURRent:HARMonic:RATio	INT32

Attribute ID	Sub Index	Access Rule	Name	Data type
	0x1	Get	FETCh[:SCALAr]:CURRe nt:HARMonic:RATio	INT32
	0x2	Get	FETCh[:SCALAr]:CURRe nt:HARMonic:RATio	INT32
	0x3	Get	FETCh[:SCALAr]:CURRe nt:HARMonic:RATio	INT32
	0x4	Get	FETCh[:SCALAr]:CURRe nt:HARMonic:RATio	INT32
	0x5	Get	FETCh[:SCALAr]:CURRe nt:HARMonic:RATio	INT32
0x1A	0x0	Get	FETCh[:SCALAr]:VOLTag e:HARMonic[:RMS]	INT32
	0x1	Get	FETCh[:SCALAr]:VOLTag e:HARMonic[:RMS]	INT32
	0x2	Get	FETCh[:SCALAr]:VOLTag e:HARMonic[:RMS]	INT32
	0x3	Get	FETCh[:SCALAr]:VOLTag e:HARMonic[:RMS]	INT32
	0x4	Get	FETCh[:SCALAr]:VOLTag e:HARMonic[:RMS]	INT32
	0x5	Get	FETCh[:SCALAr]:VOLTag e:HARMonic[:RMS]	INT32
0x1B	0x0	Get	FETCh[:SCALAr]:VOLTag e:HARMonic:RATio	INT32
	0x1	Get	FETCh[:SCALAr]:VOLTag e:HARMonic:RATio	INT32
	0x2	Get	FETCh[:SCALAr]:VOLTag e:HARMonic:RATio	INT32
	0x3	Get	FETCh[:SCALAr]:VOLTag e:HARMonic:RATio	INT32
	0x4	Get	FETCh[:SCALAr]:VOLTag e:HARMonic:RATio	INT32
	0x5	Get	FETCh[:SCALAr]:VOLTag e:HARMonic:RATio	INT32
0x1C		Get	FETCh[:SCALAr]:LINE:V OLTag e[:RMS]	INT32
0x1D		Get	FETCh[:SCALAr]:LINE:V OLTag e:AVERAge	INT32
0x1E		Get	FETCh[:SCALAr]:LINE:V OLTag e:HIGH	INT32

Attribute ID	Sub Index	Access Rule	Name	Data type
0x1F		Get	FETCh[:SCALar]:LINE:V OLTage:LOW	INT32

## Instance 0x07: Input Commands

Attribute ID	Access Rule	Name	Data type
0x03	Get / Set	INPut:GAIN	UINT16
0x04	Get / Set	INPut:SOURce	UINT8
0x05	Get / Set	INPut:SYNC:SOURce	UINT8

## Instance 0x08: InsTrument Commands

Attribute ID	Access Rule	Name	Data type
0x01	Get / Set	INSTRument:EDIT	UINT8
0x02	Get / Set	INSTRument:SElect	UINT8

## Instance 0x09: Measure Commands

Attribute ID	Sub Index	Access Rule	Name	Data type
0x01		Get / Set	MEASure:AVERAge:COU Nt	UINT8
0x02		Get / Set	MEASure:CONFigure:SE NSing	UINT8
0x03		Get / Set	MEASure:UPDate:RATE	UINT8
0x04		Get	MEASure[:SCALar]:CUR Rent:CFACtor	INT32
0x05		Get	MEASure[:SCALar]:CUR Rent:HIGH	INT32
0x06		Get	MEASure[:SCALar]:CUR Rent:LOW	INT32
0x07		Get	MEASure[:SCALar]:CUR Rent:PEAK:HOLD	INT32

Attribute ID	Sub Index	Access Rule	Name	Data type
0x08		Get	MEASure[:SCALar]:CURRent[:RMS]	INT32
0x09		Get	MEASure[:SCALar]:CURRent[:RMS]:TOTal	INT32
0x0A		Get	MEASure[:SCALar]:CURRent:AC	INT32
0x0B		Get	MEASure[:SCALar]:CURRent:AVERAge	INT32
0x0C		Get	MEASure[:SCALar]:FREQuency	INT32
0x0D		Set	MEASure[:SCALar]:PEAK:CLEar	UINT8
0x0E		Get	MEASure[:SCALar]:POWER[:AC]:APParent	INT32
0x0F		Get	MEASure[:SCALar]:POWER[:AC]:APParent:TOTal	INT32
0x10		Get	MEASure[:SCALar]:POWER[:AC]:PFACTOR	INT32
0x11		Get	MEASure[:SCALar]:POWER[:AC]:PFACTOR:TOTal	INT32
0x12		Get	MEASure[:SCALar]:POWER[:AC]:REACTIVE	INT32
0x13		Get	MEASure[:SCALar]:POWER[:AC]:REACTIVE:TOTal	INT32
0x14		Get	MEASure[:SCALar]:POWER[:AC][:REAL]	INT32
0x15		Get	MEASure[:SCALar]:POWER[:AC][:REAL]:TOTal	INT32
0x16		Get	MEASure[:SCALar]:VOLTage[:RMS]	INT32
0x17		Get	MEASure[:SCALar]:VOLTage[:RMS]:TOTal	INT32
0x18		Get	MEASure[:SCALar]:VOLTage:AC	INT32
0x19		Get	MEASure[:SCALar]:VOLTage:AVERAge	INT32
0x1A		Get	MEASure[:SCALar]:VOLTage:HIGH	INT32
0x1B		Get	MEASure[:SCALar]:VOLTage:LOW	INT32



Attribute ID	Sub Index	Access Rule	Name	Data type
0x1D	0x0	Get	MEASure[:SCALar]:CUR Rent:HARMonic[:RMS]	INT32
	0x1	Get	MEASure[:SCALar]:CUR Rent:HARMonic[:RMS]	INT32
	0x2	Get	MEASure[:SCALar]:CUR Rent:HARMonic[:RMS]	INT32
	0x3	Get	MEASure[:SCALar]:CUR Rent:HARMonic[:RMS]	INT32
	0x4	Get	MEASure[:SCALar]:CUR Rent:HARMonic[:RMS]	INT32
	0x5	Get	MEASure[:SCALar]:CUR Rent:HARMonic[:RMS]	INT32
0x1E	0x0	Get	MEASure[:SCALar]:CUR Rent:HARMonic:RATio	INT32
	0x1	Get	MEASure[:SCALar]:CUR Rent:HARMonic:RATio	INT32
	0x2	Get	MEASure[:SCALar]:CUR Rent:HARMonic:RATio	INT32
	0x3	Get	MEASure[:SCALar]:CUR Rent:HARMonic:RATio	INT32
	0x4	Get	MEASure[:SCALar]:CUR Rent:HARMonic:RATio	INT32
	0x5	Get	MEASure[:SCALar]:CUR Rent:HARMonic:RATio	INT32
0x1F	0x0	Get	MEASure[:SCALar]:VOL Tage:HARMonic[:RMS]	INT32
	0x1	Get	MEASure[:SCALar]:VOL Tage:HARMonic[:RMS]	INT32
	0x2	Get	MEASure[:SCALar]:VOL Tage:HARMonic[:RMS]	INT32
	0x3	Get	MEASure[:SCALar]:VOL Tage:HARMonic[:RMS]	INT32
	0x4	Get	MEASure[:SCALar]:VOL Tage:HARMonic[:RMS]	INT32
	0x5	Get	MEASure[:SCALar]:VOL Tage:HARMonic[:RMS]	INT32
0x20	0x0	Get	MEASure[:SCALar]:VOL Tage:HARMonic:RATio	INT32
	0x1	Get	MEASure[:SCALar]:VOL Tage:HARMonic:RATio	INT32

Attribute ID	Sub Index	Access Rule	Name	Data type
	0x2	Get	MEASure[:SCALar]:VOL Tag:e:HARMonic:RATio	INT32
	0x3	Get	MEASure[:SCALar]:VOL Tag:e:HARMonic:RATio	INT32
	0x4	Get	MEASure[:SCALar]:VOL Tag:e:HARMonic:RATio	INT32
	0x5	Get	MEASure[:SCALar]:VOL Tag:e:HARMonic:RATio	INT32
0x21		Get	MEASure[:SCALar]:LINE :VOLTag:e[:RMS]	INT32
0x22		Get	MEASure[:SCALar]:LINE :VOLTag:e:AVERAge	INT32
0x23		Get	MEASure[:SCALar]:LINE :VOLTag:e:HIGH	INT32
0x24		Get	MEASure[:SCALar]:LINE :VOLTag:e:LOW	INT32

**Instance 0x0A: Memory Commands**

Attribute ID	Access Rule	Name	Data type
0x04	Set	MEMory:SAV	UINT8
0x05	Set	MEMory:RCL	UINT8

**Instance 0x0B: Output Commands**

Attribute ID	Sub Index	Access Rule	Name	Data type
0x01		Get / Set	OUTPut:IMPedance	UINT8
0x02	0x0	Get / Set	OUTPut:IMPedance:INDuctance	UINT16
	0x1	Get / Set	OUTPut:IMPedance:INDuctance	UINT16
	0x2	Get / Set	OUTPut:IMPedance:INDuctance	UINT16
0x03	0x0	Get / Set	OUTPut:IMPedance:RESistance	UINT16
	0x1	Get / Set	OUTPut:IMPedance:RESistance	UINT16
	0x2	Get / Set	OUTPut:IMPedance:RESistance	UINT16
0x04		Get / Set	OUTPut:MONitor:AMPLitude	UINT8
0x05	0x0	Get / Set	OUTPut:MONitor:SOURce	UINT8
	0x1	Get / Set	OUTPut:MONitor:SOURce	UINT8
0x06		Get / Set	OUTPut:PON	UINT8
0x07		Set	OUTPut:PROTEction:CLEar	UINT8
0x08		Get / Set	OUTPut:RELay	UINT8
0x0A		Get / Set	OUTPut[::STATe]	UINT8

**Instance 0x0D: System Commands**

Attribute ID	Sub Index	Access Rule	Name	Data type
0x02		Get / Set	SYSTem:ARBITrary:EDIT:BUILtin	UINT8
0x03	0x0	Get / Set	SYSTem:ARBITrary:EDIT:CFACTOR	UINT16
	0x1	Get / Set	SYSTem:ARBITrary:EDIT:CFACTOR	UINT8

Attribute ID	Sub Index	Access Rule	Name	Data type
0x04		Get / Set	SYSTem:ARBITrary:EDIT: CLIP	UINT16
0x05	0x0	Get / Set	SYSTem:ARBITrary:EDIT: DIP	UINT32
	0x1	Get / Set	SYSTem:ARBITrary:EDIT: DIP	UINT32
	0x2	Get / Set	SYSTem:ARBITrary:EDIT: DIP	UINT32
	0x0	Get / Set	SYSTem:ARBITrary:EDIT: LFRing	UINT32
0x06	0x1	Get / Set	SYSTem:ARBITrary:EDIT: LFRing	UINT32
	0x2	Get / Set	SYSTem:ARBITrary:EDIT: LFRing	UINT32
	0x3	Get / Set	SYSTem:ARBITrary:EDIT: LFRing	UINT32
	0x4	Get / Set	SYSTem:ARBITrary:EDIT: LFRing	UINT32
	0x5	Get / Set	SYSTem:ARBITrary:EDIT: LFRing	UINT32
	0x6	Get / Set	SYSTem:ARBITrary:EDIT: LFRing	UINT32
	0x7	Get / Set	SYSTem:ARBITrary:EDIT: LFRing	UINT32
	0x07	0x0	Get / Set	SYSTem:ARBITrary:EDIT: RIPple
0x1		Get / Set	SYSTem:ARBITrary:EDIT: RIPple	UINT16
0x2		Get / Set	SYSTem:ARBITrary:EDIT: RIPple	UINT16
0x08		Get / Set	SYSTem:ARBITrary:EDIT: STAir	UINT8
0x09		Set	SYSTem:ARBITrary:EDIT: STORe	UINT8
0x0A	0x0	Get / Set	SYSTem:ARBITrary:EDIT: STORe:APPLy	UINT8
	0x1	Get / Set	SYSTem:ARBITrary:EDIT: STORe:APPLy	UINT8
	0x2	Get / Set	SYSTem:ARBITrary:EDIT: STORe:APPLy	UINT8

Attribute ID	Sub Index	Access Rule	Name	Data type
0x0B	0x0	Get / Set	SYSTem:ARBitrary:EDIT: SURGe	UINT8
	0x1	Get / Set	SYSTem:ARBitrary:EDIT: SURGe	UINT16
	0x2	Get / Set	SYSTem:ARBitrary:EDIT: SURGe	UINT16
0x0C		Get / Set	SYSTem:ARBitrary:EDIT: TRlAngle	UINT16
0x0D		Get / Set	SYSTem:VUNit	UINT8
0x0E		Get / Set	SYSTem:BEEPer:STATe	UINT8
0x11		Get / Set	SYSTem:COMMunicate:INTerface:ADDReSS	UINT8
0x12		Get / Set	SYSTem:COMMunicate:INTerface:BAUD	UINT8
0x13		Get / Set	SYSTem:COMMunicate:LAN:DHCP	UINT8
0x14		Get / Set	SYSTem:COMMunicate:LAN:DNS	UINT8
0x15		Get / Set	SYSTem:COMMunicate:LAN:GATeway	UINT8
0x17		Get / Set	SYSTem:COMMunicate:LAN:IPADdress	UINT8
0x18		Get	SYSTem:COMMunicate:LAN:MAC	Char
0x19		Get / Set	SYSTem:COMMunicate:LAN:SMASK	UINT8
0x1C		Get / Set	SYSTem:COMMunicate:RLSTate	UINT8
0x1D		Get / Set	SYSTem:COMMunicate:SERial[:RECeive]:TRANSm it:BAUD	UINT8
0x1E		Get / Set	SYSTem:COMMunicate:SERial[:RECeive]:TRANSm it:BITS	UINT8
0x1F		Get / Set	SYSTem:COMMunicate:SERial[:RECeive]:TRANSm it:PARity	UINT8
0x20		Get / Set	SYSTem:COMMunicate:SERial[:RECeive]:TRANSm	UINT8

Attribute ID	Sub Index	Access Rule	Name	Data type
it:SBITs				
0x21		Get	SYSTem:COMMunicate:TCPIP:CONTRol	UINT16
0x22		Get	SYSTem:COMMunicate:USB:FRONT:STATe	UINT8
0x24		Get	SYSTem:COMMunicate:USB:REAR:STATe	UINT8
0x25		Get / Set	SYSTem:CONFigure:EXTio[:STATe]	UINT8
0x26		Get / Set	SYSTem:CONFigure:PHASe	UINT8
0x27		Get / Set	SYSTem:CONFigure[:MODE]	UINT8
0x28		Get / Set	SYSTem:CONFigure:TRIGGER:OUTPut:SOURce	UINT8
0x29		Get / Set	SYSTem:CONFigure:TRIGGER:OUTPut:WIDTh	UINT16
0x2C		Get / Set	SYSTem:KLOCK	UINT8
0x2E		Get / Set	SYSTem:PKHold:TIME	UINT16
0x2F		Set	SYSTem:REBoot	UINT8
0x32		Get / Set	SYSTem:HOLD:STATe	UINT8

**Instance 0x0E: Trace Commands**

Attribute ID	Sub Index	Access Rule	Name	Data type
0x03		Set	TRACe:SEQuence:CLEar	UINT8
0x07		Set	TRACe:SEQuence:RECall	UINT8
0x08		Set	TRACe:SEQuence:STORe	UINT8
0x09		Set	TRACe:SIMulation:CLEar	UINT8
0x0D		Set	TRACe:SIMulation:RECal 1	UINT8
0x0E		Set	TRACe:SIMulation:STOR e	UINT8

Attribute ID	Sub Index	Access Rule	Name	Data type
0x12		Set	TRACe:WAVE:CLEar	UINT8
	0x0	Set	TRACe:WAVE[:DATA]	UINT32
	0x1	Set	TRACe:WAVE[:DATA]	UINT32
	0x2	Set	TRACe:WAVE[:DATA]	UINT32
	0x3	Set	TRACe:WAVE[:DATA]	UINT32
			.	
0x14			.	
			.	
	0xF9	Set	TRACe:WAVE[:DATA]	UINT32
	0xFA	Set	TRACe:WAVE[:DATA]	UINT32
	0xFB	Set	TRACe:WAVE[:DATA]	UINT32
	0xFC	Set	TRACe:WAVE[:DATA]	UINT32

#### Instance 0x0F: Trigger Commands

Attribute ID	Access Rule	Name	Data type
0x05	Set	TRIGger:SEQuence:SElected:EX ECute	UINT8
0x06	Set	TRIGger:SIMulation:SElected:E XECute	UINT8

**Instance 0x11: Source 1 Commands**

<b>Attribute ID</b>	<b>Sub Index</b>	<b>Access Rule</b>	<b>Name</b>	<b>Data type</b>
0x01		Get / Set	[:SOURce]:CURRENT:LIMit:PEAK:HIGH	UINT32
0x02		Get / Set	[:SOURce]:CURRENT:LIMit:PEAK:LOW	INT32
0x03		Get / Set	[:SOURce]:CURRENT:LIMit:PEAK:MODE	UINT8
0x04		Get / Set	[:SOURce]:CURRENT:LIMit:RMS[:AMPLitude]	UINT32
0x05		Get / Set	[:SOURce]:CURRENT:LIMit:RMS:MODE	UINT8
0x06		Get / Set	[:SOURce]:FREQUency:LIMit:HIGH	UINT32
0x07		Get / Set	[:SOURce]:FREQUency:LIMit:LOW	UINT32
0x08		Get / Set	[:SOURce]:FREQUency[:IMMediate]	UINT32
0x09		Get / Set	[:SOURce]:FUNCTION[:SHAPE][:IMMediate]	UINT8
0x0A		Get / Set	[:SOURce]:FUNCTION:THD:FORMat	UINT8
0x0B		Get / Set	[:SOURce]:LINE:VOLTag e[:LEVel][:IMMediate][:AMPLitude]	UINT32
0x0C		Get / Set	[:SOURce]:LINE:VOLTag e[:LEVel][:IMMediate]:OFFSet	INT32
0x0D		Get / Set	[:SOURce]:PHASe:BALance	UINT8
0x0E		Get / Set	[:SOURce]:PHASe:MODE	UINT8
0x0F	0x0	Get / Set	[:SOURce]:PHASe:PHASe	UINT16
	0x1	Get / Set	[:SOURce]:PHASe:PHASe	UINT16
0x10		Get / Set	[:SOURce]:PHASe:RELoc k	UINT8
0x11		Get / Set	[:SOURce]:PHASe:SETChange:STATe	UINT8



Attribute ID	Sub Index	Access Rule	Name	Data type
0x12		Get / Set	[:SOURce]:PHASe:START[:IMMediate]	UINT16
0x13		Get / Set	[:SOURce]:PHASe:START:STATe	UINT8
0x14		Get / Set	[:SOURce]:PHASe:STOP[:IMMediate]	UINT16
0x15		Get / Set	[:SOURce]:PHASe:STOP:STATe	UINT8
0x16		Get / Set	[:SOURce]:PHASe:SYNC[:IMMediate]	UINT16
0x17	0x00	Get	[:SOURce]:READ?	INT32
	0x01	Get	[:SOURce]:READ?	INT32
	0x02	Get	[:SOURce]:READ?	INT32
	0x03	Get	[:SOURce]:READ?	INT32
	0x04	Get	[:SOURce]:READ?	INT32
	0x05	Get	[:SOURce]:READ?	INT32
	0x06	Get	[:SOURce]:READ?	INT32
	0x07	Get	[:SOURce]:READ?	INT32
	0x08	Get	[:SOURce]:READ?	INT32
	0x09	Get	[:SOURce]:READ?	INT32
	0x0A	Get	[:SOURce]:READ?	INT32
	0x0B	Get	[:SOURce]:READ?	INT32
	0x0C	Get	[:SOURce]:READ?	INT32
0x0D	Get	[:SOURce]:READ?	INT32	
0x0E	Get	[:SOURce]:READ?	INT32	
0x0F	Get	[:SOURce]:READ?	INT32	
0x10	Get	[:SOURce]:READ?	INT32	
0x11	Get	[:SOURce]:READ?	INT32	
0x12	Get	[:SOURce]:READ?	INT32	

Attribute ID	Sub Index	Access Rule	Name	Data type
	0x13	Get	[:SOURCE]:READ?	INT32
	0x14	Get	[:SOURCE]:READ?	INT32
	0x15	Get	[:SOURCE]:READ?	INT32
	0x16	Get	[:SOURCE]:READ?	INT32
	0x17	Get	[:SOURCE]:READ?	INT32
	0x18	Get	[:SOURCE]:READ?	INT32
	0x19	Get	[:SOURCE]:READ?	INT32
	0x1A	Get	[:SOURCE]:READ?	INT32
	0x1B	Get	[:SOURCE]:READ?	INT32
	0x1C	Get	[:SOURCE]:READ?	INT32
	0x1D	Get	[:SOURCE]:READ?	INT32
	0x1E	Get	[:SOURCE]:READ?	INT32
	0x1F	Get	[:SOURCE]:READ?	INT32
	0x20	Get	[:SOURCE]:READ?	INT32
	0x21	Get	[:SOURCE]:READ?	INT32
	0x22	Get	[:SOURCE]:READ?	INT32
	0x23	Get	[:SOURCE]:READ?	INT32
	0x24	Get	[:SOURCE]:READ?	INT32
	0x25	Get	[:SOURCE]:READ?	INT32
	0x26	Get	[:SOURCE]:READ?	INT32
	0x27	Get	[:SOURCE]:READ?	INT32
	0x28	Get	[:SOURCE]:READ?	INT32
	0x29	Get	[:SOURCE]:READ?	INT32
	0x2A	Get	[:SOURCE]:READ?	INT32
	0x2B	Get	[:SOURCE]:READ?	INT32

Attribute ID	Sub Index	Access Rule	Name	Data type
	0x2C	Get	[:SOURce]:READ?	INT32
	0x2D	Get	[:SOURce]:READ?	INT32
	0x2E	Get	[:SOURce]:READ?	INT32
	0x2F	Get	[:SOURce]:READ?	INT32
	0x30	Get	[:SOURce]:READ?	INT32
	0x31	Get	[:SOURce]:READ?	INT32
	0x32	Get	[:SOURce]:READ?	INT32
0x30		Get	[:SOURce]:SEQuence:CO NDition	UINT8
	0x00	Get / Set	[:SOURce]:SEQuence:CP ARameter	UINT32
	0x01	Get / Set	[:SOURce]:SEQuence:CP ARameter	UINT16
	0x02	Get / Set	[:SOURce]:SEQuence:CP ARameter	UINT8
	0x03	Get / Set	[:SOURce]:SEQuence:CP ARameter	UINT16
	0x04	Get / Set	[:SOURce]:SEQuence:CP ARameter	UINT8
	0x05	Get / Set	[:SOURce]:SEQuence:CP ARameter	UINT8
0x31	0x06	Get / Set	[:SOURce]:SEQuence:CP ARameter	UINT16
	0x07	Get / Set	[:SOURce]:SEQuence:CP ARameter	UINT8
	0x08	Get / Set	[:SOURce]:SEQuence:CP ARameter	UINT16
	0x09	Get / Set	[:SOURce]:SEQuence:CP ARameter	UINT8
	0x0A	Get / Set	[:SOURce]:SEQuence:CP ARameter	UINT16
	0x0B	Get / Set	[:SOURce]:SEQuence:CP ARameter	UINT8
	0x0C	Get / Set	[:SOURce]:SEQuence:CP ARameter	UINT16

Attribute ID	Sub Index	Access Rule	Name	Data type
	0x0D	Get / Set	[:SOURce]:SEQuence:CP ARameter	UINT8
	0x0E	Get / Set	[:SOURce]:SEQuence:CP ARameter	UINT8
0x32		Get	[:SOURce]:SEQuence:CST ep	UINT16
0x33		Get / Set	[:SOURce]:SEQuence:INS Trument:SElect	UINT8
	0x00	Get / Set	[:SOURce]:SEQuence:SPA Rameter	UINT32
	0x01	Get / Set	[:SOURce]:SEQuence:SPA Rameter	UINT8
	0x02	Get / Set	[:SOURce]:SEQuence:SPA Rameter	UINT32
0x34	0x03	Get / Set	[:SOURce]:SEQuence:SPA Rameter	UINT8
	0x04	Get / Set	[:SOURce]:SEQuence:SPA Rameter	UINT32
	0x05	Get / Set	[:SOURce]:SEQuence:SPA Rameter	UINT8
	0x06	Get / Set	[:SOURce]:SEQuence:SPA Rameter	UINT8
	0x07	Get / Set	[:SOURce]:SEQuence:SPA Rameter	UINT16
0x35		Get / Set	[:SOURce]:SEQuence:STE P	UINT16
0x36		Get	[:SOURce]:SIMulation:CO NDition	UINT8
0x37		Get / Set	[:SOURce]:SIMulation:AB Normal:CODE	UINT8
0x38		Get / Set	[:SOURce]:SIMulation:AB Normal:FREQuency	UINT32
0x39		Get / Set	[:SOURce]:SIMulation:AB Normal:PHASe:STARt:EN ABle	UINT8
0x3A		Get / Set	[:SOURce]:SIMulation:AB Normal:PHASe:STARt[:I MMediate]	UINT16
0x3B		Get / Set	[:SOURce]:SIMulation:AB Normal:PHASe:STOP:EN	UINT8

Attribute ID	Sub Index	Access Rule	Name	Data type
ABle				
0x3C		Get / Set	[:SOURce]:SIMulation:AB Normal:PHASe:STOP[:IM Mediate]	UINT16
0x3D		Get / Set	[:SOURce]:SIMulation:AB Normal:TIME	UINT32
0x3E		Get / Set	[:SOURce]:SIMulation:AB Normal:VOLTag	UINT32
0x3F		Get	[:SOURce]:SIMulation:CS Tep	UINT8
0x40		Get / Set	[:SOURce]:SIMulation:INI Tial:CODE	UINT8
0x41		Get / Set	[:SOURce]:SIMulation:INI Tial:FREQuency	UINT32
0x42		Get / Set	[:SOURce]:SIMulation:INI Tial:PHASe:STARt:ENAB le	UINT8
0x43		Get / Set	[:SOURce]:SIMulation:INI Tial:PHASe:STARt[:IMMe diate]	UINT16
0x44		Get / Set	[:SOURce]:SIMulation:INI Tial:PHASe:STOP:ENAB le	UINT8
0x45		Get / Set	[:SOURce]:SIMulation:INI Tial:PHASe:STOP[:IMMe diate]	UINT16
0x46		Get / Set	[:SOURce]:SIMulation:INI Tial:VOLTag	UINT32
0x48	0x00	Get / Set	[:SOURce]:SIMulation:N ORMal:CODE	UINT8
	0x01	Get / Set	[:SOURce]:SIMulation:N ORMal:CODE	UINT8
0x49	0x00	Get / Set	[:SOURce]:SIMulation:N ORMal:FREQuency	UINT32
	0x01	Get / Set	[:SOURce]:SIMulation:N ORMal:FREQuency	UINT32
0x4A	0x00	Get / Set	[:SOURce]:SIMulation:N ORMal:PHASe:STARt:EN ABle	UINT8

Attribute ID	Sub Index	Access Rule	Name	Data type
	0x01	Get / Set	[:SOURce]:SIMulation:NORMal:PHASe:START:ENABLE	UINT8
0x4B	0x00	Get / Set	[:SOURce]:SIMulation:NORMal:PHASe:START[:IMMmediate]	UINT16
	0x01	Get / Set	[:SOURce]:SIMulation:NORMal:PHASe:START[:IMMmediate]	UINT16
0x4C	0x00	Get / Set	[:SOURce]:SIMulation:NORMal:PHASe:STOP:ENABLE	UINT8
	0x01	Get / Set	[:SOURce]:SIMulation:NORMal:PHASe:STOP:ENABLE	UINT8
0x4D	0x00	Get / Set	[:SOURce]:SIMulation:NORMal:PHASe:STOP[:IMMmediate]	UINT16
	0x01	Get / Set	[:SOURce]:SIMulation:NORMal:PHASe:STOP[:IMMmediate]	UINT16
0x4E	0x00	Get / Set	[:SOURce]:SIMulation:NORMal:TIME	UINT32
	0x01	Get / Set	[:SOURce]:SIMulation:NORMal:TIME	UINT32
0x4F	0x00	Get / Set	[:SOURce]:SIMulation:NORMal:VOLTage	UINT32
	0x01	Get / Set	[:SOURce]:SIMulation:NORMal:VOLTage	UINT32
0x50		Get / Set	[:SOURce]:SIMulation:REPeat:COUNT	UINT16
0x51		Get / Set	[:SOURce]:SIMulation:REPeat:ENABLE	UINT8
0x52	0x00	Get / Set	[:SOURce]:SIMulation:TRANSition:CODE	UINT8
	0x01	Get / Set	[:SOURce]:SIMulation:TRANSition:CODE	UINT8
0x53	0x00	Get / Set	[:SOURce]:SIMulation:TRANSition:TIME	UINT32
	0x01	Get / Set	[:SOURce]:SIMulation:TRANSition:TIME	UINT32

Attribute ID	Sub Index	Access Rule	Name	Data type
0x60	0x00	Get / Set	[:SOURce]:SEQuence:NSP arameter	UINT32
	0x01	Get / Set	[:SOURce]:SEQuence:NSP arameter	UINT8
	0x02	Get / Set	[:SOURce]:SEQuence:NSP arameter	UINT32
	0x03	Get / Set	[:SOURce]:SEQuence:NSP arameter	UINT8
	0x04	Get / Set	[:SOURce]:SEQuence:NSP arameter	UINT32
	0x05	Get / Set	[:SOURce]:SEQuence:NSP arameter	UINT8
	0x06	Get / Set	[:SOURce]:SEQuence:NSP arameter	UINT8
	0x07	Get / Set	[:SOURce]:SEQuence:NSP arameter	UINT16
	0x08	Get / Set	[:SOURce]:SEQuence:NSP arameter	UINT16

## Instance 0x12: Source 2 Commands

Attribute ID	Access Rule	Name	Data type
0x01	Get / Set	[:SOURce]:SQUare:DCYCLE	UINT16
0x02	Get / Set	[:SOURce]:VOLTage:LIMit:PEAK	UINT32
0x03	Get / Set	[:SOURce]:VOLTage:LIMit:RMS	UINT32
0x04	Get / Set	[:SOURce]:VOLTage:LIMit:HIGH	UINT32
0x05	Get / Set	[:SOURce]:VOLTage:LIMit:LOW	INT32
0x06	Get / Set	[:SOURce]:VOLTage:RANGE	UINT8
0x07	Get / Set	[:SOURce]:VOLTage:RESPonse	UINT8

Attribute ID	Access Rule	Name	Data type
0x08	Get / Set	[[:SOURce]:VOLTage[:LEVel][:IMMediate]][:AMPLitude]	UINT32
0x09	Get / Set	[[:SOURce]:VOLTage[:LEVel][:IMMediate]]:OFFSet	INT32
0x0A	Get / Set	[[:SOURce]:MODE	UINT8

**Instance 0x17: Read Data Range Commands**

Attribute ID	Sub Index	Access Rule	Name	Data type
0x01	0x00	Get	DATA_RANGE_MAX	UINT32
	0x01	Get	DATA_RANGE_MAX	INT32
	0x02	Get	DATA_RANGE_MAX	UINT32
	0x03	Get	DATA_RANGE_MAX	UINT32
	0x04	Get	DATA_RANGE_MAX	UINT16
	0x05	Get	DATA_RANGE_MAX	UINT16
	0x06	Get	DATA_RANGE_MAX	UINT16
	0x07	Get	DATA_RANGE_MAX	UINT16
	0x08	Get	DATA_RANGE_MAX	UINT16
	0x09	Get	DATA_RANGE_MAX	UINT16
	0x0A	Get	DATA_RANGE_MAX	UINT32
	0x0B	Get	DATA_RANGE_MAX	INT32
	0x0C	Get	DATA_RANGE_MAX	UINT32
	0x0D	Get	DATA_RANGE_MAX	INT32
	0x0E	Get	DATA_RANGE_MAX	UINT32
	0x0F	Get	DATA_RANGE_MAX	UINT32
0x02	0x00	Get	DATA_RANGE_MIN	UINT32
	0x01	Get	DATA_RANGE_MIN	INT32



Attribute ID	Sub Index	Access Rule	Name	Data type
	0x02	Get	DATA_RANGE_MIN	UINT32
	0x03	Get	DATA_RANGE_MIN	UINT32
	0x04	Get	DATA_RANGE_MIN	UINT16
	0x05	Get	DATA_RANGE_MIN	UINT16
	0x06	Get	DATA_RANGE_MIN	UINT16
	0x07	Get	DATA_RANGE_MIN	UINT16
	0x08	Get	DATA_RANGE_MIN	UINT16
	0x09	Get	DATA_RANGE_MIN	UINT16
	0x0A	Get	DATA_RANGE_MIN	UINT32
	0x0B	Get	DATA_RANGE_MIN	INT32
	0x0C	Get	DATA_RANGE_MIN	UINT32
	0x0D	Get	DATA_RANGE_MIN	INT32
	0x0E	Get	DATA_RANGE_MIN	UINT32
	0x0F	Get	DATA_RANGE_MIN	UINT32

**Common Services**

Service Code	Implemented for		Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	No	Yes	Set_Attribute_Single

## Manufacturer-specific Command List

Object Class	Instance ID	Name
0x64	0x01	Common Commands
	0x03/0x0E	Data/Trace Commands
	0x04	Display Commands
	0x06	Fetch Commands
	0x07	Input Commands
	0x08	InsTrument Commands
	0x09	Measure Commands
	0x0A	Memory Commands
	0x0B	Output Commands
	0x0D	System Function Commands
	0x0F	Trigger Commands
	0x11	Source commands
	0x12	Source 1 Commands
	0x17	Read Data Range Commands

Service Code	Name
0x0E	Get_Attribute_Single
0x10	Set_Attribute_Single

## Common Commands

---

Attribute 0x02: *CLS.....	91
Attribute 0x05: *IDN .....	91
Attribute 0x07: *RCL .....	94
Attribute 0x08: *RST .....	94
Attribute 0x09: *SAV .....	95
Attribute 0x0E: *WAI.....	95

## Data / Trace Commands (Instance 0x03/0x0E)

---

Attribute 0x02/0x03: DATA/TRACe:SEQuence:CLear .....	96
Attribute 0x04/0x07: DATA/TRACe:SEQuence:RECall .....	97
Attribute 0x05/0x08: DATA/TRACe:SEQuence:STORe .....	97
Attribute 0x06/0x09: DATA/TRACe:SIMulation:CLear .....	98
Attribute 0x08/0x0D: DATA/TRACe:SIMulation:RECall .....	98
Attribute 0x09/0x0E: DATA/TRACe:SIMulation:STORe.....	99
Attribute 0x13/0x12: DATA/TRACe:WAVe:CLear .....	99
Attribute 0x15/0x14: DATA/TRACe:WAVe[;DATA] .....	100
Request Data / Trace Commands Example .....	101

## Display Commands (Instance 0x04)

---

Attribute 0x01: DISPlay[:WINDow]:DESIgn:MODE .....	102
Attribute 0x02:DISPlay[:WINDow]:MEASure:SOURce<1   2   3   4> ..	103
Attribute 0x10: DISPLAY_ADDRESS .....	105
Request Display Commands Example .....	106

## Fetch Commands (Instance 0x06)

---

Attribute 0x01: FETCh[:SCALar]:CURRent:CFACtor.....	108
Attribute 0x02: FETCh[:SCALar]:CURRent:HIGH.....	108
Attribute 0x03: FETCh[:SCALar]:CURRent:LOW .....	109
Attribute 0x04: FETCh[:SCALar]:CURRent:PEAK:HOLD .....	110
Attribute 0x05: FETCh[:SCALar]:CURRent[:RMS].....	110
Attribute 0x06: FETCh[:SCALar]:CURRent[:RMS]:TOTal .....	111
Attribute 0x07: FETCh[:SCALar]:CURRent:AC .....	111
Attribute 0x08: FETCh[:SCALar]:CURRent:AVERage .....	112

Attribute 0x09: FETCh[:SCALar]:FREQuency .....	112
Attribute 0x0A: FETCh[:SCALar]:POWer[:AC]:APParent .....	113
Attribute 0x0B: FETCh[:SCALar]:POWer[:AC]:APParent:TOTal .....	114
Attribute 0x0C: FETCh[:SCALar]:POWer[:AC]:PFACtor .....	114
Attribute 0x0D: FETCh[:SCALar]:POWer[:AC]:PFACtor:TOTal .....	115
Attribute 0x0E: FETCh[:SCALar]:POWer[:AC]:REActive .....	115
Attribute 0x0F: FETCh[:SCALar]:POWer[:AC]:REActive:TOTal .....	116
Attribute 0x10: FETCh[:SCALar]:POWer[:AC][:REAL] .....	116
Attribute 0x11: FETCh[:SCALar]:POWer[:AC][:REAL]:TOTal .....	117
Attribute 0x12: FETCh[:SCALar]:VOLTag[:RMS] .....	118
Attribute 0x13: FETCh[:SCALar]:VOLTag[:RMS]:TOTal .....	118
Attribute 0x14: FETCh[:SCALar]:VOLTag:AC .....	119
Attribute 0x15: FETCh[:SCALar]:VOLTag:AVERAge .....	119
Attribute 0x16: FETCh[:SCALar]:VOLTag:HIGH .....	120
Attribute 0x17: FETCh[:SCALar]:VOLTag:LOW .....	120
Attribute 0x18: FETCh[:SCALar]:CURRent:HARMonic[:RMS] .....	121
Attribute 0x19: FETCh[:SCALar]:CURRent:HARMonic:RATio .....	126
Attribute 0x1A: FETCh[:SCALar]:VOLTag:HARMonic[:RMS] .....	131
Attribute 0x1B: FETCh[:SCALar]:VOLTag:HARMonic:RATio .....	136
Attribute 0x1C: FETCh[:SCALar]:LINE:VOLTag[:RMS] .....	141
Attribute 0x1D: FETCh[:SCALar]:LINE:VOLTag:AVERAge .....	141
Attribute 0x1E: FETCh[:SCALar]:LINE:VOLTag:HIGH .....	142
Attribute 0x1F: FETCh[:SCALar]:LINE:VOLTag:LOW .....	143
Request Fetch Commands Example .....	144

### Input Command (Instance 0x07)

Attribute 0x03: INPut:GAIN .....	145
Attribute 0x04: INPut:SOURce .....	146
Attribute 0x05: INPut:SYNC:SOURce .....	146
Request Input Commands Example .....	147

### Instrument Commands (Instance 0x08)

Attribute 0x01: INSTrument:EDIT .....	148
Attribute 0x02: INSTrument:SElect .....	149
Request Instrument Commands Example .....	149

## Measure Commands (Instance 0x09)

Attribute 0x01: MEASure:AVERage:COUNt.....	151
Attribute 0x02: MEASure:CONFigure:SENSing.....	152
Attribute 0x03: MEASure:UPDate:RATE.....	152
Attribute 0x04: MEASure[:SCALar]:CURRent:CFACTOR .....	153
Attribute 0x05: MEASure[:SCALar]:CURRent:HIGH .....	154
Attribute 0x06: MEASure[:SCALar]:CURRent:LOW .....	155
Attribute 0x07: MEASure[:SCALar]:CURRent:PEAK:HOLD .....	155
Attribute 0x08: MEASure[:SCALar]:CURRent[:RMS] .....	156
Attribute 0x09: MEASure[:SCALar]:CURRent[:RMS]:TOTal .....	156
Attribute 0x0A: MEASure[:SCALar]:CURRent:AC.....	157
Attribute 0x0B: MEASure[:SCALar]:CURRent:AVERage .....	157
Attribute 0x0C: MEASure[:SCALar]:FREQuency .....	158
Attribute 0x0D: MEASure[:SCALar]:PEAK:CLear .....	159
Attribute 0x0E: MEASure[:SCALar]:POWER[:AC]:APParent .....	160
Attribute 0x0F:MEASure[:SCALar]:POWER[:AC]:APParent:TOTal... 160	
Attribute 0x10: MEASure[:SCALar]:POWER[:AC]:PFACTOR .....	161
Attribute 0x11:MEASure[:SCALar]:POWER[:AC]:PFACTOR:TOTal.... 161	
Attribute 0x12: MEASure[:SCALar]:POWER[:AC]:REACTive .....	162
Attribute 0x13: MEASure[:SCALar]:POWER[:AC]:REACTive:TOTal . 163	
Attribute 0x14: MEASure[:SCALar]:POWER[:AC][:REAL] .....	163
Attribute 0x15: MEASure[:SCALar]:POWER[:AC][:REAL]:TOTal .....	164
Attribute 0x16: MEASure[:SCALar]:VOLTage[:RMS] .....	164
Attribute 0x17: MEASure[:SCALar]:VOLTage[:RMS]:TOTal .....	165
Attribute 0x18: MEASure[:SCALar]:VOLTage:AC.....	166
Attribute 0x19: MEASure[:SCALar]:VOLTage:AVERage.....	166
Attribute 0x1A: MEASure[:SCALar]:VOLTage:HIGH.....	167
Attribute 0x1B: MEASure[:SCALar]:VOLTage:LOW .....	167
Attribute 0x1D: MEASure[:SCALar]:CURRent:HARMonic[:RMS] ... 168	
Attribute 0x1E: MEASure[:SCALar]:CURRent:HARMonic:RATio .... 173	
Attribute 0x1F: MEASure[:SCALar]:VOLTage:HARMonic[:RMS] .... 178	
Attribute 0x20: MEASure[:SCALar]:VOLTage:HARMonic:RATio .... 183	
Attribute 0x21: MEASure[:SCALar]:LINE:VOLTage[:RMS] .....	187
Attribute 0x22: MEASure[:SCALar]:LINE:VOLTage:AVERage .....	188
Attribute 0x23: MEASure[:SCALar]:LINE:VOLTage:HIGH.....	189
Attribute 0x24: MEASure[:SCALar]:LINE:VOLTage:LOW .....	189
Request Measure Commands Example .....	190

## Memory Commands (Instance 0x0A)

---

Attribute 0x04: MEMory:SAV .....	191
Attribute 0x05: MEMory:RCL .....	192
Request Memory Commands Example .....	192

## Output Commands (Instance 0x0B)

---

Attribute 0x01: OUTPut:IMPedance .....	193
Attribute 0x02: OUTPut:IMPedance:INDuctance .....	194
Attribute 0x03: OUTPut:IMPedance:RESistance.....	194
Attribute 0x04: OUTPut:MONitor:AMPLitude .....	195
Attribute 0x05: OUTPut:MONitor:SOURce<1   2> .....	196
Attribute 0x06: OUTPut:PON .....	196
Attribute 0x07: OUTPut:PROTection:CLEar .....	197
Attribute 0x08: OUTPut:RELAy .....	197
Attribute 0x0A: OUTPut[.STATE] .....	198
Request Output Commands Example .....	199

## System Function Commands (Instance 0x0D)

---

Attribute 0x02: SYSTem:ARBitrary:EDIT:BUILtin .....	201
Attribute 0x0B: SYSTem:ARBitrary:EDIT:SURGe .....	202
Attribute 0x08: SYSTem:ARBitrary:EDIT:STAir .....	204
Attribute 0x03: SYSTem:ARBitrary:EDIT:CFACtor<1   2> .....	204
Attribute 0x04: SYSTem:ARBitrary:EDIT:CLIP.....	205
Attribute 0x09: SYSTem:ARBitrary:EDIT:STORe .....	206
Attribute 0x0C: SYSTem:ARBitrary:EDIT:TRlangle .....	206
Attribute 0x05: SYSTem:ARBitrary:EDIT:DIP .....	207
Attribute 0x06: SYSTem:ARBitrary:EDIT:LFRing .....	209
Attribute 0x07: SYSTem:ARBitrary:EDIT:RIPple.....	212
Attribute 0x0A: SYSTem:ARBitrary:EDIT:STORe:APPLy<1   3>.....	214
Attribute 0x0E: SYSTem:BEEPer:STATE .....	215
Attribute 0x11: SYSTem:COMMunicate:INTerface:ADDRes .....	215
Attribute 0x12: SYSTem:COMMunicate:INTerface:BAUD.....	216
Attribute 0x13: SYSTem:COMMunicate:LAN:DHCP .....	217
Attribute 0x14: SYSTem:COMMunicate:LAN:DNS .....	217
Attribute 0x15: SYSTem:COMMunicate:LAN:GATeway .....	218

Attribute 0x17: SYSTem:COMMunicate:LAN:IPADdress .....	219
Attribute 0x18: SYSTem:COMMunicate:LAN:MAC .....	220
Attribute 0x19: SYSTem:COMMunicate:LAN:SMASK.....	221
Attribute 0x1C: SYSTem:COMMunicate:RLState.....	222
Attribute 0x1D:	
SYSTem:COMMunicate:SERial[:RECeive]:TRANsmit:BAUD .....	223
Attribute 0x1E:	
SYSTem:COMMunicate:SERial[:RECeive]:TRANsmit:BITS.....	224
Attribute 0x1F:	
SYSTem:COMMunicate:SERial[:RECeive]:TRANsmit:PARity .....	225
Attribute 0x20:	
SYSTem:COMMunicate:SERial[:RECeive]:TRANsmit:SBITs .....	226
Attribute 0x21: SYSTem:COMMunicate:TCPip:CONTRol.....	226
Attribute 0x22: SYSTem:COMMunicate:USB:FRONT:STATe .....	227
Attribute 0x24: SYSTem:COMMunicate:USB:REAR:STATe.....	227
Attribute 0x27: SYSTem:CONFigure[:MODE] .....	228
Attribute 0x25: SYSTem:CONFigure:EXTio[:STATe] .....	228
Attribute 0x26: SYSTem:CONFigure:PHASe.....	228
Attribute 0x28: SYSTem:CONFigure:TRIGger:OUTPut:SOURce.....	229
Attribute 0x29: SYSTem:CONFigure:TRIGger:OUTPut:WIDTh .....	230
Attribute 0x32: SYSTem:HOLD:STATe.....	231
Attribute 0x2E: SYSTem:PKHold:TIME .....	231
Attribute 0x2C: SYSTem:KLOCK .....	232
Attribute 0x2F: SYSTem:REBoot .....	232
Attribute 0x0D: SYSTem:VUNit.....	232
Request System Function Commands Example .....	233

### Trigger Commands (Instance 0x0F)

Attribute 0x05: TRIGger:SEQuence:SElected:EXECute.....	234
Attribute 0x06: TRIGger:SIMulation:SElected:EXECute .....	235
Request Trigger Commands Example.....	235

### Source Commands (Instance 0x11)

Attribute 0x01: [:SOURce]:CURRent:LIMit:PEAK:HIGH.....	238
Attribute 0x02: [:SOURce]:CURRent:LIMit:PEAK:LOW .....	238
Attribute 0x03: [:SOURce]:CURRent:LIMit:PEAK:MODE .....	239
Attribute 0x04: [:SOURce]:CURRent:LIMit:RMS[:AMPLitude] .....	240

---

Attribute 0x05: [:SOURCE]:CURRent:LIMit:RMS:MODE .....	240
Attribute 0x06: [:SOURCE]:FREQuency:LIMit:HIGH.....	241
Attribute 0x07: [:SOURCE]:FREQuency:LIMit:LOW .....	242
Attribute 0x08: [:SOURCE]:FREQuency[:IMMEDIATE] .....	242
Attribute 0x09: [:SOURCE]:FUNCTion[:SHAPE][:IMMEDIATE].....	243
Attribute 0x0A: [:SOURCE]:FUNCTion:THD:FORMat .....	244
Attribute 0x0B: [:SOURCE]:LINE:VOLTage[:LEVel][:IMMEDIATE][:AMPLitude] .....	246
Attribute 0x0C: [:SOURCE]:LINE:VOLTage[:LEVel][:IMMEDIATE]:OFFSet .....	246
Attribute 0x0D: [:SOURCE]:PHASe:BALance.....	247
Attribute 0x0E: [:SOURCE]:PHASe:MODE .....	248
Attribute 0x0F: [:SOURCE]:PHASe:PHASe.....	249
Attribute 0x10: [:SOURCE]:PHASe:RELock .....	250
Attribute 0x11: [:SOURCE]:PHASe:SETChange:STATe .....	250
Attribute 0x12: [:SOURCE]:PHASe:START[:IMMEDIATE].....	251
Attribute 0x13: [:SOURCE]:PHASe:START:STATe.....	251
Attribute 0x14: [:SOURCE]:PHASe:STOP[:IMMEDIATE] .....	252
Attribute 0x15: [:SOURCE]:PHASe:STOP:STATe .....	252
Attribute 0x16: [:SOURCE]:PHASe:SYNC[:IMMEDIATE].....	253
Attribute 0x17: [:SOURCE]:READ .....	254
Attribute 0x30: [:SOURCE]:SEQuence:CONDition.....	256
Attribute 0x31: [:SOURCE]:SEQuence:CPARAmeter.....	256
Attribute 0x32: [:SOURCE]:SEQuence:CSTep .....	260
Attribute 0x33: [:SOURCE]:SEQuence:INSTrument:SElect .....	261
Attribute 0x34: [:SOURCE]:SEQuence:SPARAmeter .....	261
Attribute 0x35: [:SOURCE]:SEQuence:STEP .....	264
Attribute 0x36: [:SOURCE]:SIMulation:CONDition.....	265
Attribute 0x37: [:SOURCE]:SIMulation:ABNormal:CODE.....	265
Attribute 0x38: [:SOURCE]:SIMulation:ABNormal:FREQuency .....	266
Attribute 0x39: [:SOURCE]:SIMulation:ABNormal:PHASe:START:ENABLE .....	267
Attribute 0x3A: [:SOURCE]:SIMulation:ABNormal:PHASe:START[:IMMEDIATE] .....	267
Attribute 0x3B [:SOURCE]:SIMulation:ABNormal:PHASe:STOP:ENABLE .....	269
Attribute 0x3C: [:SOURCE]:SIMulation:ABNormal:PHASe:STOP[:IMMEDIATE].....	269
Attribute 0x3D: [:SOURCE]:SIMulation:ABNormal:TIME .....	270
Attribute 0x3E: [:SOURCE]:SIMulation:ABNormal:VOLTage .....	271



Attribute 0x3F: [:SOURce]:SIMulation:CSTep .....	271
Attribute 0x40: [:SOURce]:SIMulation:INITial:CODE .....	272
Attribute 0x41: [:SOURce]:SIMulation:INITial:FREQUency .....	273
Attribute 0x42: [:SOURce]:SIMulation:INITial:PHASe:STARt:ENABLE .....	273
Attribute 0x43: [:SOURce]:SIMulation:INITial:PHASe:STARt[:IMMediate].....	274
Attribute 0x44: [:SOURce]:SIMulation:INITial:PHASe:STOP:ENABLE .....	275
Attribute 0x45: [:SOURce]:SIMulation:INITial:PHASe:STOP[:IMMediate] .....	275
Attribute 0x46: [:SOURce]:SIMulation:INITial:VOLTagE .....	276
Attribute 0x48: [:SOURce]:SIMulation:NORMal<1   2>:CODE.....	277
Attribute 0x49: [:SOURce]:SIMulation:NORMal 1:FREQUency .....	278
Attribute 0x4A: [:SOURce]:SIMulation:NORMal<1   2>:PHASe:STARt:ENABLE .....	279
Attribute 0x4B: [:SOURce]:SIMulation:NORMal<1   2>:PHASe:STARt[:IMMediate].....	280
Attribute 0x4C: [:SOURce]:SIMulation:NORMal<1   2>:PHASe:STOP:ENABLE.....	281
Attribute 0x4D: [:SOURce]:SIMulation:NORMal<1   2>:PHASe:STOP[:IMMediate] .....	282
Attribute 0x4E: [:SOURce]:SIMulation:NORMal<1   2>:TIME.....	283
Attribute 0x4F: [:SOURce]:SIMulation:NORMal 1:VOLTagE .....	284
Attribute 0x50: [:SOURce]:SIMulation:REPeat:COUNt.....	285
Attribute 0x51: [:SOURce]:SIMulation:REPeat:ENABLE .....	285
Attribute 0x53: [:SOURce]:SIMulation:TRANSition<1   2>:TIME.....	286
Attribute 0x52: [:SOURce]:SIMulation:TRANSition<1   2>:CODE .....	287
Attribute 0x60: [:SOURce]:SEQUence:NSParameter.....	288
Request Source Commands Example .....	291

## Source1 Commands (Instance 0x12)

Attribute 0x01: [:SOURce]:SQUare:DCYCLE.....	293
Attribute 0x02: [:SOURce]:VOLTagE:LIMit:PEAK .....	293
Attribute 0x03: [:SOURce]:VOLTagE:LIMit:RMS .....	294
Attribute 0x04: [:SOURce]:VOLTagE:LIMit:HIGH .....	295
Attribute 0x05: [:SOURce]:VOLTagE:LIMit:LOW .....	295
Attribute 0x06: [:SOURce]:VOLTagE:RANGe.....	296

Attribute 0x07: [:SOURce]:VOLTage:RESPonse.....	297
Attribute 0x08: [:SOURce]:VOLTage[:LEVel][:IMMEdiate][:AMPLitude] .....	298
Attribute 0x09: [:SOURce]:VOLTage[:LEVel][:IMMEdiate]:OFFSet ...	299
Attribute 0x0A: [:SOURce]:MODE.....	300
Request Source1 Commands Example .....	301

### Read Data Range Commands (Instance 0x17)

---

Attribute 0x01: DATA_RANGE_MAX .....	302
Attribute 0x02: DATA_RANGE_MIN .....	306
Request Read Data Range Commands Example .....	310

## Common Commands

Attribute 0x02: *CLS.....	91
Attribute 0x05: *IDN .....	91
Attribute 0x07: *RCL .....	94
Attribute 0x08: *RST .....	94
Attribute 0x09: *SAV .....	95
Attribute 0x0E: *WAI .....	95

### Attribute 0x02: \*CLS

Description	The *CLS command clears all the event registers, including the status byte, event status and error queue.
Sub-Index	-
Name	*CLS
Value	-
Type	UINT8
Data size	1 Byte
Access	Set
Example (Data)	XX 10 64 01 02 (*CLS)  Sets the *CLS command.

### Attribute 0x05: \*IDN

Description	Queries the manufacturer, model name, serial number, and firmware version of the ASR.
Sub-Index	-
Name	*IDN
Value	Returns the instrument identification as a

string in the following format:

GW-INSTEK,ASR-  
 XXXX,GXXXXXXXX,XX.XX  
 Manufacturer: GW-INSTEK  
 Model number : ASR-XXXX  
 Serial number : GXXXXXXXX  
 Firmware version : XX.XX

Type	Char
Data Size	36 Bytes
Access	Get
Example (Data)	<p>Command :</p> <p>XX 0E 64 01 05 (*IDN?)</p> <p>Fragment Return 1:</p> <p>8X 00 8E 47 57 2D 49 4E</p> <p>(Char data is "GW-IN")</p> <p>Fragment Command Ack:</p> <p>8X C0 00</p> <p>Fragment Return 2:</p> <p>8X 41 53 54 45 4B 2C 41</p> <p>(Char data is "STEK,A")</p> <p>Fragment Command Ack:</p> <p>8X C1 00</p> <p>Fragment Return 3:</p> <p>8X 42 53 52 5D 36 36 30</p> <p>(Char data is "SR-660")</p> <p>Fragment Command Ack:</p> <p>8X C2 00</p>

Fragment Return 3:

8X 43 30 2C 53 4E 30 30

(Char data is "00,SN00")

Fragment Command Ack:

8X C3 00

Fragment Return 3:

8X 44 30 30 30 31 2C 31

(Char data is "0001,1")

Fragment Command Ack:

8X C4 00

Fragment Return 3:

8X 45 2E 32 36 2E 30 30

(Char data is ".26.00")

Fragment Command Ack:

8X C5 00

Fragment Return 3:

8X 86 30

(Char data is "0")

Fragment Command Ack:

8X C6 00

Return data is "GW-INSTEK,ASR-  
6600,SN000001,1.26.000"

---

## Attribute 0x07: \*RCL

Description	Recalls the contents stored in memory slot M0 ~ M9. These memory slots are mapped to the preset settings.
Sub-Index	-
Name	*RCL
Value	0 ~ 9 (as memory M0 ~ M9)
Type	UINT8
Data size	1 Byte
Access	Set
Example (Data)	XX 10 64 01 07 (*RCL)
	Sets the *RCL command.

## Attribute 0x08: \*RST

Description	Performs a device reset. Configures the unit to a known configuration (default settings). This known configuration is independent of the usage history.
Sub-Index	-
Name	*RST
Value	-
Type	UINT8
Data size	1 Byte
Access	Set
Example (Data)	XX 10 64 01 08 (*RST)
	Sets the *RST command.

## Attribute 0x09: \*SAV

---

Description	Saves the settings into memory slot M0 ~ M9. These memory slots are mapped to the preset settings.
Sub-Index	-
Name	*SAV
Value	0 ~ 9 (as memory M0 ~ M9)
Type	UINT8
Data size	1 Byte
Access	Set

---

## Attribute 0x0E: \*WAI

---

Description	Prevents any other commands or queries from being executed until all outstanding commands have completed.
Sub-Index	-
Name	*WAI
Value	-
Type	UINT8
Data size	1 Byte
Access	Set

---

## Data / Trace Commands (Instance 0x03/0x0E)



Note

The TRACE and DATA node for the following commands are functionally equivalent.

Attribute 0x02/0x03: DATA/TRACe:SEQuence:CLEar.....	96
Attribute 0x04/0x07: DATA/TRACe:SEQuence:RECall.....	97
Attribute 0x05/0x08: DATA/TRACe:SEQuence:STORe.....	97
Attribute 0x06/0x09: DATA/TRACe:SIMulation:CLEar.....	98
Attribute 0x08/0x0D: DATA/TRACe:SIMulation:RECall.....	98
Attribute 0x09/0x0E: DATA/TRACe:SIMulation:STORe.....	99
Attribute 0x13/0x12: DATA/TRACe:WAVe:CLEar.....	99
Attribute 0x15/0x14: DATA/TRACe:WAVe[:DATA].....	100
Request Data / Trace Commands Example.....	101

### Attribute 0x02/0x03: DATA/TRACe:SEQuence:CLEar

Description	Clears the sequence data for the selected save memory (Seq0 ~ Seq9).
Sub-Index	-
Name	DATA   TRACe:SEQuence:CLEar
Value	0 ~ 9 (Seq0 ~ Seq9).
Type	UINT8
Data size	4 Bytes
Access	Set
Example	XX 10 64 03 02 01(:DATA:SEQ:CLE 1)
(Data)	XX 10 64 0E 03 01(:TRACe:SEQ:CLE 1)
	Clears the sequence data from Seq1.



## Attribute 0x04/0x07: DATA/TRACe:SEQuence:RECall

Description	Loads the sequence data. This command is the equivalent to recalling a sequence memory in the Sequence mode.
Sub-Index	-
Name	DATA   TRACe:SEQuence:RECall
Value	0 ~ 9 (Seq0 ~ Seq9).
Type	UINT8
Data size	1 Byte
Access	Set
Example (Data)	XX 10 64 03 04 01(:DATA:SEQ:REC 1) XX 10 64 0E 07 01(:TRACe:SEQ:REC 1)
	Loads the data from Seq1.

## Attribute 0x05/0x08: DATA/TRACe:SEQuence:STORe

Description	Saves the sequence data. This command is the equivalent to saving a sequence memory in Sequence mode.
Sub-Index	-
Name	DATA   TRACe:SEQuence:STORe
Value	0 ~ 9 (Seq0 ~ Seq9).
Type	UINT8
Data size	1 Byte
Access	Set
Example (Data)	XX 10 64 03 05 01(:DATA:SEQ:STOR 1) XX 10 64 0E 08 01(:TRACe:SEQ:STOR 1)

Saves the data from Seq1.

**Attribute 0x06/0x09: DATA/TRACe:SIMulation:CLEar**

Description	Clears the simulation data for the selected save memory (SIM0 ~ SIM9).
Sub-Index	-
Name	DATA   TRACe:SIMulation:CLEar
Value	0 ~ 9 (SIM0 ~ SIM9).
Type	UINT8
Data size	1 Byte
Access	Set
Example (Data)	XX 10 64 03 06 01(:DATA:SIM:CLE 1) XX 10 64 0E 09 01(:TRACe:SIM:CLE 1)

Clears the simulation data from SIM1.

**Attribute 0x08/0x0D: DATA/TRACe:SIMulation:RECall**

Description	Loads the simulation data. This command is the equivalent to recalling a simulation memory in the Simulation mode (SIM0~SIM9).
Sub-Index	-
Name	DATA   TRACe:SIMulation:RECall
Value	0 ~ 9 (SIM0 ~ SIM9).
Type	UINT8
Data size	1 Byte
Access	Set
Example	XX 10 64 03 08 01(:DATA:SIM:REC 1)

(Data)	XX 10 64 0E 0D 01(:TRACe:SIM:REC 1)
--------	-------------------------------------

Loads the data from SIM1.

### Attribute 0x09/0x0E: DATA/TRACe:SIMulation:STORe

Description	Saves the simulation data. This command is the equivalent saving a simulation memory in Simulation mode (SIM0 ~ SIM9).
Sub-Index	-
Name	DATA   TRACe:SIMulation:STORe
Value	0 ~ 9 (SIM0 ~ SIM9).
Type	UINT8
Data size	1 Byte
Access	Set
Example	XX 10 64 03 09 01(:DATA:SIM:STOR 1)
(Data)	XX 10 64 0E 0E 01(:TRACe:SIM:STOR 1)

Saves the data from SIM1.

### Attribute 0x13/0x12: DATA/TRACe:WAVE:CLEAr

Description	Clears the ARB 1-253 data for the selected wave group.
Sub-Index	-
Name	DATA   TRACe:WAVE:CLEAr
Value	0~252 (ARB1 ~ ARB253).
Type	UINT8
Data size	1 Byte
Access	Set

Example	XX 10 64 03 13 0D(:DATA:WAV:CLE 13)
(Data)	XX 10 64 0E 12 0D(:TRACe:WAV:CLE 13)

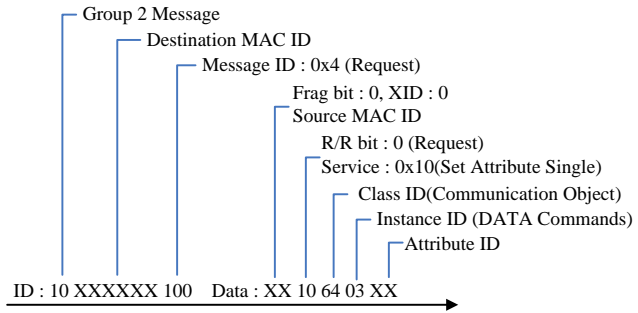
Clears the wave data from ARB14.

**Attribute 0x15/0x14: DATA/TRACe:WAVE[:DATA]**

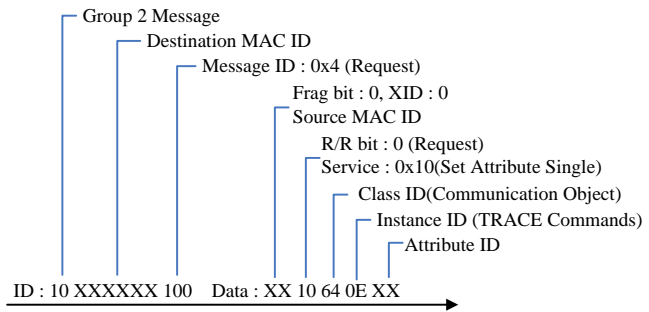
Description	Sets the arbitrary wave.
Sub-Index	0x00 ~ 0xFC (ARB 1 ~ 253)
Name	DATA/TRACe:WAVE[:DATA]
Vange	Indicates 16-bit with 4096 words waveform data. Plus, the data format of wave is the big endian in the form of two's complement.
Type	UINT32
Data size	8192 Bytes
Access	Set

## Request Data / Trace Commands Example

Data  
Request  
Example



Trace  
Request  
Example



## Display Commands (Instance 0x04)

Attribute 0x01: DISPlay[:WINDow]:DESIgn:MODE .....	102
Attribute 0x02:DISPlay[:WINDow]:MEASure:SOURce<1   2   3   4> ..	103
Attribute 0x10: DISPLAY_ADDRESS.....	105
Request Display Commands Example.....	106

### Attribute 0x01: DISPlay[:WINDow]:DESIgn:MODE

Description	Sets three display mode.	
Sub-Index	-	
Name	DISPlay[:WINDow]:DESIgn:MODE	
Value	0 = NORMAl	Configure setup and Measurement.
	1 = TOTAl	Configure setup and Measurement include total information.
	2 = SIMPlE	All measurement times.
Type	UINT8	
Data size	1 Byte	
Access	Set	
Example (Data)	XX 10 64 04 01 00 (:DISP:DES:MODE NORM)	
	Sets standard normal display.	

## Attribute

0x02:DISPlay[:WINDow]:MEASure:SOURce&lt;1|2|3|4&gt;

Description	Sets standard normal display to measurement items 1 - 4.
Sub-Index	0x00
Name	Item 1
Value	0 = VRMS 1 = VAVG 2 = VMAX 3 = VMIN 4 = VPKH 10 = RPOWer 11 = SPOWer (Not available for DC-INT) 12 = QPOWer (Not available for DC-INT) 16 = THDV (Available for AC-INT only) 18 = LRMS 19 = LAVG 20 = LMAX 21 = LMIN
Type	UINT8
Data size	1 Byte
Access	Get/Set
Sub-Index	0x01
Name	Item 2
Value	5 = IRMS 6 = IAVG 7 = IMAX 8 = IMIN 9 = IPKH 14 = PFACTOR (Not available for DC-INT) 15 = CFACTOR (Not available for DC-INT) 17 = THDI (Available for AC-I T only)
Type	UINT8

Data size	1 Byte
Access	Get/Set
Sub-Index	0x02
Name	Item 3
Value	9 = IPKH, 10 = RPOWer 11 = SPOWer 12 = QPOWer (Not available for DC-INT) 13 = FREQuency (Available for AC+DC-Sync & AC-Synonly) 14 = PFACtor (Not available for DC-INT) 15 = CFACtor (Not available for DC-INT)

Type	UINT8
Data size	1 Byte

Access	Get/Set
--------	---------

Sub-Index	0x03
-----------	------

Name	Item 4
------	--------

Value	Available for DC-INT only: 5 = IRMS 6 = IAVG 7 = IMAX 8 = IMIN 9 = IPKH
-------	--

Not available for DC-INT:

- 11 = SPOWer
- 12 = QPOWer
- 14 = PFACtor
- 15 = CFACtor
- 18 = LRMS
- 19 = LAVG
- 20 = LMAX
- 21 = LMIN

Common:

- 10 = RPOWer



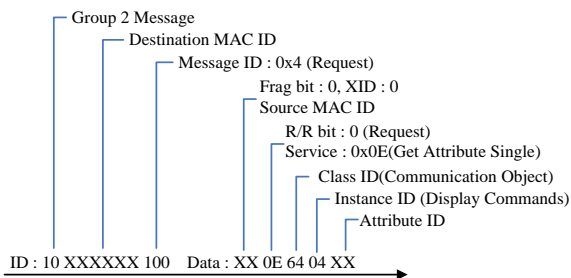
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 10 64 04 02 00 00 (:DISP:MEAS:SOUR1 VRMS)
Sets measurement source 1 VRMS display.	

### Attribute 0x10: DISPLAY\_ADDRESS

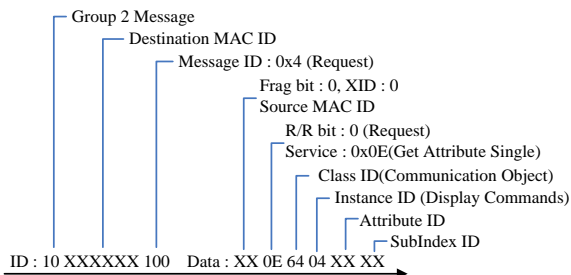
Description	Display the CAN Bus/DeviceNet address on screen.
Sub-Index	-
Name	DISPLAY_ADDRESS
Value	-
Type	UINT8
Data size	1 Byte
Access	Set
Example (Data)	XX 10 64 04 10 (DISPLAY_ADDRESS)
Display the CAN Bus/DeviceNet address on screen.	

## Request Dispalay Commands Example

Request Example no including SubIndex



Request Example including SubIndex



## Fetch Commands (Instance 0x06)

Attribute 0x01: FETCh[:SCALar]:CURRent:CFActor.....	108
Attribute 0x02: FETCh[:SCALar]:CURRent:HIGH.....	108
Attribute 0x03: FETCh[:SCALar]:CURRent:LOW .....	109
Attribute 0x04: FETCh[:SCALar]:CURRent:PEAK:HOLD .....	110
Attribute 0x05: FETCh[:SCALar]:CURRent[:RMS].....	110
Attribute 0x06: FETCh[:SCALar]:CURRent[:RMS]:TOTal .....	111
Attribute 0x07: FETCh[:SCALar]:CURRent:AC .....	111
Attribute 0x08: FETCh[:SCALar]:CURRent:AVERAge .....	112
Attribute 0x09: FETCh[:SCALar]:FREQuency .....	112
Attribute 0x0A: FETCh[:SCALar]:POWER[:AC]:APParent .....	113
Attribute 0x0B: FETCh[:SCALar]:POWER[:AC]:APParent:TOTal .....	114
Attribute 0x0C: FETCh[:SCALar]:POWER[:AC]:PFACTOR .....	114
Attribute 0x0D: FETCh[:SCALar]:POWER[:AC]:PFACTOR:TOTal.....	115
Attribute 0x0E: FETCh[:SCALar]:POWER[:AC]:REACTive .....	115
Attribute 0x0F: FETCh[:SCALar]:POWER[:AC]:REACTive:TOTal.....	116
Attribute 0x10: FETCh[:SCALar]:POWER[:AC][:REAL] .....	116
Attribute 0x11: FETCh[:SCALar]:POWER[:AC][:REAL]:TOTal .....	117
Attribute 0x12: FETCh[:SCALar]:VOLTage[:RMS].....	118
Attribute 0x13: FETCh[:SCALar]:VOLTage[:RMS]:TOTal.....	118
Attribute 0x14: FETCh[:SCALar]:VOLTage:AC .....	119
Attribute 0x15: FETCh[:SCALar]:VOLTage:AVERAge .....	119
Attribute 0x16: FETCh[:SCALar]:VOLTage:HIGH.....	120
Attribute 0x17: FETCh[:SCALar]:VOLTage:LOW .....	120
Attribute 0x18: FETCh[:SCALar]:CURRent:HARMonic[:RMS] .....	121
Attribute 0x19: FETCh[:SCALar]:CURRent:HARMonic:RATio.....	126
Attribute 0x1A: FETCh[:SCALar]: VOLTage:HARMonic[:RMS].....	131
Attribute 0x1B: FETCh[:SCALar]:VOLTage:HARMonic:RATio.....	136
Attribute 0x1C: FETCh[:SCALar]:LINE:VOLTage[:RMS] .....	141
Attribute 0x1D: FETCh[:SCALar]:LINE:VOLTage:AVERAge.....	141
Attribute 0x1E: FETCh[:SCALar]:LINE:VOLTage:HIGH.....	142
Attribute 0x1F: FETCh[:SCALar]:LINE:VOLTage:LOW .....	143
Request Fetch Commands Example .....	144

## Attribute 0x01: FETCh[:SCALAr]:CURRent:CFACtor

Description	Returns the output fetch current crest factor (CF).
Sub-Index	-
Name	FETCh[:SCALAr]:CURRent:CFACtor
Value	Returns the crest factor.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 06 01(:FETCh:CURR:CFACtor?)  Return data is 1520. XX 8E F0 05 00 00 The crest factor is 1.52, Magnification is 1000.

## Attribute 0x02: FETCh[:SCALAr]:CURRent:HIGH

Description	Returns the output fetch current maximum peak value (Imax).
Sub-Index	-
Name	FETCh[:SCALAr]:CURRent:HIGH
Value	Returns the Imax value in amps.
Type	INT32
Data size	4 Bytes
Access	Get
Note	Current maximum peak value is defined as the highest peak value in the complete period.

Example  
(Data)

XX 0E 64 06 02(:FETCh:CURR:HIGH?)

Return data is 20050.

XX 8E 52 4E 00 00

The current maximum peak value is  
20.05 A, Magnification is 1000..

#### Attribute 0x03: FETCh[:SCALAr]:CURRent:LOW

Description	Returns the output fetch current minimum value (Imin).
Sub-Index	-
Name	FETCh[:SCALAr]:CURRent:LOW
Value	Returns the Imin value in amps.
Type	INT32
Data size	4 Bytes
Access	Get
Note	Current minimum value is defined as the lowest value in the complete period.
Example (Data)	XX 0E 64 06 03 (:FETCh:CURR:LOW?)  Return data is 1050. XX 8E 1A 04 00 00 The current minimum value is 1.05 A, Magnification is 1000.

**Attribute 0x04: FETCh[:SCALar]:CURRent:PEAK:HOLD**

Description	Returns the fetch current peak hold value in amps (IPK Hold).
Sub-Index	-
Name	FETCh[:SCALar]:CURRent:PEAK:HOLD
Value	Returns the peak hold value in amps.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 06 04 (:FETCh:CURR:PEAK:HOLD?)  Return data is 20050. XX 8E 52 4E 00 00 The ipeak hold is 20.05 A, Magnification is 1000.

**Attribute 0x05: FETCh[:SCALar]:CURRent[:RMS]**

Description	Returns the output fetch current (Irms).
Sub-Index	-
Name	FETCh[:SCALar]:CURRent[:RMS]
Value	Returns the current value in Irms.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 06 05 (:FETCh:CURR?)  Return data is 10050.

---

XX 8E 42 27 00 00

The current is 10.05 A, Magnification is 1000.

---

### Attribute 0x06: FETCh[:SCALAr]:CURRent[:RMS]:TOTAl

---

Description	Returns the total of output fetch current (Irms).
Sub-Index	-
Name	FETCh[:SCALAr]:CURRent[:RMS]:TOTAl
Value	Returns the total of current value in Irms.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 06 06 (:FETCh:CURR:TOTAl?)  Return data is 10050. XX 8E 42 27 00 00 The total of current is 10.05 A, Magnification is 1000.

---

### Attribute 0x07: FETCh[:SCALAr]:CURRent:AC

---

Description	Returns the output fetch AC current (Irms).
Sub-Index	-
Name	FETCh[:SCALAr]:CURRent:AC
Value	Returns the AC current value in Irms.
Type	INT32
Data size	4 Bytes
Access	Get

---

Example (Data)	XX 0E 64 06 07 (:FETCh:CURR:AC?)  Return data is 10050. XX 8E 42 27 00 00 The AC current is 10.05 A, Magnification is 1000.
-------------------	---

**Attribute 0x08: FETCh[:SCALAr]:CURRent:AVERAge**

Description	Returns the fetch current average value (Iavg).
Sub-Index	-
Name	FETCh[:SCALAr]:CURRent:AVERAge
Value	Returns the current average value in amps.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 06 08 (:FETCh:CURR:AVERAge?)  Return data is 10050. XX 8E 42 27 00 00 The current average value is 10.05 A, Magnification is 1000.

**Attribute 0x09: FETCh[:SCALAr]:FREQUency**

Description	Returns the fetch SYNC signal source frequency in Hz. The external sync signal frequency measurement range is 10.0 Hz to 2100.0 Hz. (Only AC+DC-sync or AC-sync Active)
Sub-Index	-



Name	FETCh[:SCALar]:FREQuency
Value	Returns the SYNC frequency in Hz
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 06 09 (:FETCh:FREQuency?)  Return data is 50000. XX 8E 50 C3 00 00 The SYNC frequency is 50 Hz, Magnification is 1000.

#### Attribute 0x0A: FETCh[:SCALar]:POWer[:AC]:APParent

Description	Returns the fetch apparent power (S).
Sub-Index	-
Name	FETCh[:SCALar]:POWer[:AC]:APParent
Value	Returns the apparent power in VA.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 06 0A (:FETCh:POWer:APParent?)  Return data is 2500. XX 8E C4 09 00 00 The apparent power is 2.5, Magnification is 1000.

**Attribute 0x0B: FETCh[:SCALar]:POWer[:AC]:APParent:TOTal**

Description	Returns the fetch total of apparent power (S).
Sub-Index	-
Name	FETCh[:SCALar]:POWer[:AC]:APParent:TOTal
Value	Returns the total of apparent power in VA.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 06 0B (:FETCh:POWer:APParent:TOTal?)  Return data is 2500. XX 8E C4 09 00 00 The total of apparent power is 2.5, Magnification is 1000.

**Attribute 0x0C: FETCh[:SCALar]:POWer[:AC]:PFACTOR**

Description	Returns the fetch power factor (PF).
Sub-Index	-
Name	FETCh[:SCALar]:POWer[:AC]:PFACTOR
Value	Returns the power factor.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 06 0C (:FETCh:POWer:PFACTOR?)  Return data is 2500.

XX 8E C4 09 00 00

The power factor is 2.5, Magnification is 1000.

### Attribute 0x0D: FETCh[:SCALar]:POWer[:AC]:PFACtor:TOTal

Description	Returns the total of fetch power factor (PF).
Sub-Index	-
Name	FETCh[:SCALar]:POWer[:AC]:PFACtor:TOTal
Value	Returns the total of power factor.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 06 0D (:FETCh:POWer:PFACtor:TOTal?)  Return data is 2500. XX 8E C4 09 00 00 The total of power factor is 2.5, Magnification is 1000.

### Attribute 0x0E: FETCh[:SCALar]:POWer[:AC]:REACTive

Description	Returns the fetch reactive power (Q).
Sub-Index	-
Name	FETCh[:SCALar]:POWer[:AC]:REACTive
Value	Returns the reactive power in VAR.
Type	INT32
Data size	4 Bytes
Access	Get
Example	XX 0E 64 06 0E

(Data) (:FETCh:POWer:REACtive?)

Return data is 2500.

XX 8E C4 09 00 00

The reactive power is 2.5, Magnification is 1000.

**Attribute 0x0F: FETCh[:SCALar]:POWer[:AC]:REACtive:TOTal**

Description Returns the total of fetch reactive power (Q).

Sub-Index -

Name FETCh[:SCALar]:POWer[:AC]:REACtive:TO  
Tal

Value Returns the total of reactive power in VAR

Type INT32

Data size 4 Bytes

Access Get

Example XX 0E 64 06 0F

(Data) (:FETCh:POWer:REACtive:TOTal?)

Return data is 2500.

XX 8E C4 09 00 00

The reactive power is 2.5, Magnification is 1000.

**Attribute 0x10: FETCh[:SCALar]:POWer[:AC][:REAL]**

Description Returns the fetch active power in Watts (P).

Sub-Index -

Name FETCh[:SCALar]:POWer[:AC][:REAL]

Value Returns the power in Watts.

Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 06 10 (:FETCh:POWer?)  Return data is 100500. XX 8E 94 88 01 00 The power is 100.5 W, Magnification is 1000.

#### Attribute 0x11: FETCh[:SCALar]:POWer[:AC][:REAL]:TOTal

Description	Returns the total of fetch active power in Watts (P).
Sub-Index	-
Name	FETCh[:SCALar]:POWer[:AC][:REAL]:TOTal
Value	Returns the total power in Watts.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 06 11 (:FETCh:POWer:TOTal?)  Return data is 100500. XX 8E 94 88 01 00 The total of power is 100.5 W, Magnification is 1000.

**Attribute 0x12: FETCh[:SCALAr]:VOLTage[:RMS]**

Description	Returns the fetch voltage (Vrms).
Sub-Index	-
Name	FETCh[:SCALAr]:VOLTage[:RMS]
Value	Returns the voltage value in Vrms.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 06 12 (:FETCh:VOLTage?)  Return data is 100500. XX 8E 94 88 01 00 The voltage is 100.5 V, Magnification is 1000.

**Attribute 0x13: FETCh[:SCALAr]:VOLTage[:RMS]:TOTal**

Description	Returns the total of fetch voltage (Vrms).
Sub-Index	-
Name	FETCh[:SCALAr]:VOLTage[:RMS]:TOTal
Value	Returns the total of voltage value in Vrms.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 06 13 (:FETCh:VOLTage:TOTal?)  Return data is 100500. XX 8E 94 88 01 00

The total of voltage is 100.5 V,  
Magnification is 1000.

**Attribute 0x14: FETCh[:SCALAr]:VOLTage:AC**

Description	Returns the fetch AC voltage (Vrms).
Sub-Index	-
Name	FETCh[:SCALAr]:VOLTage:AC
Value	Returns the AC voltage value in Vrms.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 06 14 (:FETCh:VOLTage:AC?)  Return data is 100500. XX 8E 94 88 01 00 The AC voltage is 100.5 V, Magnification is 1000.

**Attribute 0x15: FETCh[:SCALAr]:VOLTage:AVERage**

Description	Returns the fetch voltage average value (Vavg).
Sub-Index	-
Name	FETCh[:SCALAr]:VOLTage:AVERage
Value	Returns the voltage average value in volts.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 06 15 (:FETCh:VOLTage:AC?)

Return data is -50750.

XX 8E C2 39 FF FF

The voltage average is -50.75 V,  
Magnification is 1000.

**Attribute 0x16: FETCh[:SCALAr]:VOLTAge:HIGH**

Description	Returns the output fetch voltage maximum peak value (Vmax).
Sub-Index	-
Name	FETCh[:SCALAr]:VOLTAge:HIGH
Value	Returns the Vmax value in volts.
Type	INT32
Data size	4 Bytes
Access	Get
Note	Voltage maximum peak value is defined as the highest peak value in the complete period.
Example (Data)	XX 0E 64 06 16 (:FETCh:VOLTAge:HIGH?)  Return data is 100500. XX 8E 94 88 01 00 The voltage maximum peak is 100.5 V, Magnification is 1000.

**Attribute 0x17: FETCh[:SCALAr]:VOLTAge:LOW**

Description	Returns the output fetch current minimum value (Vmin).
Sub-Index	-
Name	FETCh[:SCALAr]:VOLTAge:LOW



Value	Returns the Vmin value in volts
Type	INT32
Data size	4 Bytes
Access	Get
Note	Voltage minimum value is defined as the lowest value in the complete period.
Example (Data)	XX 0E 64 06 17 (:FETCh:VOLTage:LOW?)  Return data is -50750. XX 8E C2 39 FF FF The voltage minimum is -50.75 V, Magnification is 1000.

#### Attribute 0x18: FETCh[:SCALar]:CURRent:HARMonic[:RMS]

Description	Returns 101 values covering Total and order 1 to 100 fetch current (Irms) in harmonic. (Only AC-INT and 50/60 Hz Active)
Sub-Index	0x00
Name	Total Fetch current (Irms) in harmonic.
Value	Returns the entire 20 values containing Total and order 1 to 20 current (Irms) in harmonic.
Type	INT32
Data size	4 Bytes
Access	Get
Sub-Index	0x01
Name	Page 1
Value	Returns the entire 20 values containing order 21 to 40 current (Irms) in

	harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x02
Name	Page 2
Value	Returns the entire 20 values containing order 41 to 60 current (Irms) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x03
Name	Page 3
Value	Returns the entire 20 values containing order 61 to 80 current (Irms) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x04
Name	Page 4
Value	Returns the entire 20 values containing order 81 to 100 current (Irms) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x05

Name	Page 5
Value	Returns the entire 20 values containing order 81 to 100 current (Irms) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Example (Data)	<p>Command 1: XX 0E 64 06 18 05 (:FETCh:CURR:HARMonic? 5)</p> <p>Fragment Return 1: 8X 00 8E 00 00 00 00 00 (Byte 3 ~ Byte 7 is data)</p> <p>Fragment Command Ack: 8X C0 00</p> <p>Fragment Return 2: 8X 41 00 00 00 00 00 00 (Byte 2 ~ Byte 7 is data)</p> <p>Fragment Command Ack: 8X C1 00</p> <p>Fragment Return 3: 8X 42 00 00 00 00 00 00 (Byte 2 ~ Byte 7 is data)</p> <p>Fragment Command Ack: 8X C2 00</p> <p>Fragment Return 4:</p>

8X 43 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C3 00

Fragment Return 5:

8X 44 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C4 00

Fragment Return 6:

8X 45 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C5 00

Fragment Return 7:

8X 46 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C6 00

Fragment Return 8:

8X 47 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C7 00

Fragment Return 9:

---

8X 48 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C8 00

Fragment Return 10:

8X 49 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C9 00

Fragment Return 11:

8X 4A 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X CA 00

Fragment Return 12:

8X 4B 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X CB 00

Fragment Return 13:

8X 4C 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X CC 00

Fragment Return 14:

---

8X 8D 00 00 00  
 (Byte 2 ~ Byte 4 is data)

Fragment Command Ack:  
 8X CD 00

In a total of 80 bytes of data.

**Attribute 0x19: FETCh[:SCALAr]:CURRent:HARMonic:RATio**

Description	Returns 101 values covering Total and order 1 to 100 fetch current (Ratio) in harmonic. (Only AC-INT and 50/60 Hz Active)
Sub-Index	0x00
Name	Total Fetch current (Ratio) in harmonic.
Value	Returns the Total Fetch current (Ratio) in harmonic.
Type	INT32
Data size	4 Bytes
Access	Get
Sub-Index	0x01
Name	Page 1
Value	Returns the entire 20 values containing order 1 to 20 current (Ratio) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x02

Name	Page 2
Value	Returns the entire 20 values containing order 21 to 40 current (Ratio) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x03
Name	Page 3
Value	Returns the entire 20 values containing order 41 to 60 current (Ratio) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x04
Name	Page 4
Value	Returns the entire 20 values containing order 61 to 80 current (Ratio) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x05
Name	Page 5
Value	Returns the entire 20 values containing order 81 to 100 current (Ratio) in harmonic.
Type	INT32
Data size	80 Bytes

Example

(Data)

Command 1:

XX 0E 64 06 19 05

(:FETCh:CURR:HARMonic:RATio? 5)

Fragment Return 1:

8X 00 8E 00 00 00 00

(Byte 3 ~ Byte 7 is data)

Fragment Command Ack:

8X C0 00

Fragment Return 2:

8X 41 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C1 00

Fragment Return 3:

8X 42 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C2 00

Fragment Return 4:

8X 43 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C3 00

Fragment Return 5:

8X 44 00 00 00 00 00

---



(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C4 00

Fragment Return 6:

8X 45 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C5 00

Fragment Return 7:

8X 46 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C6 00

Fragment Return 8:

8X 47 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C7 00

Fragment Return 9:

8X 48 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C8 00

Fragment Return 10:

8X 49 00 00 00 00 00 00

---

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C9 00

Fragment Return 11:

8X 4A 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X CA 00

Fragment Return 12:

8X 4B 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X CB 00

Fragment Return 13:

8X 4C 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X CC 00

Fragment Return 14:

8X 8D 00 00 00

(Byte 2 ~ Byte 4 is data)

Fragment Command Ack:

8X CD 00

---

In a total of 80 bytes of data.

## Attribute 0x1A: FETCh[:SCALar]: VOLTage:HARMonic[:RMS]

Description	Returns 101 values covering Total and order 1 to 100 fetch voltage (Vrms) in harmonic. (Only AC-INT and 50/60 Hz Active)
Sub-Index	0x00
Name	Total fetch voltage (Vrms) in harmonic.
Value	Returns the Total fetch voltage (Vrms) in harmonic.
Type	INT32
Data size	4 Bytes
Access	Get
Sub-Index	0x01
Name	Page 1
Value	Returns the entire 20 values containing order 1 to 20 fetch voltage (Vrms) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x02
Name	Page 2
Value	Returns the entire 20 values containing order 21 to 40 fetch voltage (Vrms) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x03

Name	Page 3
Value	Returns the entire 20 values containing order 41 to 60 fetch voltage (Vrms) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x04
Name	Page 4
Value	Returns the entire 20 values containing order 61 to 80 fetch voltage (Vrms) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x05
Name	Page 5
Value	Returns the entire 20 values containing order 81 to 100 fetch voltage (Vrms) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Example (Data)	Command 1: XX 0E 64 06 1A 05 (:FETCh: VOLTage:HARMonic? 5)
	Fragment Return 1: 8X 00 8E 00 00 00 00 (Byte 3 ~ Byte 7 is data)

Fragment Command Ack:

8X C0 00

Fragment Return 2:

8X 41 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C1 00

Fragment Return 3:

8X 42 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C2 00

Fragment Return 4:

8X 43 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C3 00

Fragment Return 5:

8X 44 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C4 00

Fragment Return 6:

8X 45 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

---

Fragment Command Ack:

8X C5 00

Fragment Return 7:

8X 46 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C6 00

Fragment Return 8:

8X 47 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C7 00

Fragment Return 9:

8X 48 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C8 00

Fragment Return 10:

8X 49 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C9 00

Fragment Return 11:

8X 4A 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

---

Fragment Command Ack:

8X CA 00

Fragment Return 12:

8X 4B 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X CB 00

Fragment Return 13:

8X 4C 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X CC 00

Fragment Return 14:

8X 8D 00 00 00

(Byte 2 ~ Byte 4 is data)

Fragment Command Ack:

8X CD 00

---

In a total of 80 bytes of data.

---

**Attribute 0x1B: FETCh[:SCALAr]:VOLTage:HARMonic:RATio**

Description	Returns 101 values covering Total and order 1 to 100 fetch Voltage (Ratio) in harmonic. (Only AC-INT and 50/60 Hz Active)
Sub-Index	0x00
Name	Total fetch Voltage (Ratio) in harmonic.
Value	Returns the Total fetch Voltage (Ratio) in harmonic.
Type	INT32
Data size	4 Bytes
Access	Get
Sub-Index	0x01
Name	Page 1
Value	Returns the entire 20 values containing order 1 to 20 fetch Voltage (Ratio) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x02
Name	Page 2
Value	Returns the entire 20 values containing order 21 to 40 fetch Voltage (Ratio) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x03



Name	Page 3
Value	Returns the entire 20 values containing order 41 to 60 fetch Voltage (Ratio) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x04
Name	Page 4
Value	Returns the entire 20 values containing order 61 to 80 fetch Voltage (Ratio) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x05
Name	Page 5
Value	Returns the entire 20 values containing order 81 to 10 fetch Voltage (Ratio) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Example (Data)	<p>Command 1:</p> <p>XX 0E 64 06 1B 05</p> <p>(:FETCh: VOLTage:HARMonic:RATio? 5)</p> <p>Fragment Return 1:</p> <p>8X 00 8E 00 00 00 00 00</p>

(Byte 3 ~ Byte 7 is data)

Fragment Command Ack:

8X C0 00

Fragment Return 2:

8X 41 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C1 00

Fragment Return 3:

8X 42 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C2 00

Fragment Return 4:

8X 43 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C3 00

Fragment Return 5:

8X 44 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C4 00

Fragment Return 6:

8X 45 00 00 00 00 00 00

---

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C5 00

Fragment Return 7:

8X 46 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C6 00

Fragment Return 8:

8X 47 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C7 00

Fragment Return 9:

8X 48 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C8 00

Fragment Return 10:

8X 49 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C9 00

Fragment Return 11:

8X 4A 00 00 00 00 00 00

---

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X CA 00

Fragment Return 12:

8X 4B 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X CB 00

Fragment Return 13:

8X 4C 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X CC 00

Fragment Return 14:

8X 8D 00 00 00

(Byte 2 ~ Byte 4 is data)

Fragment Command Ack:

8X CD 00

---

In a total of 80 bytes of data.

---

## Attribute 0x1C: FETCh[:SCALar]:LINE:VOLTage[:RMS]

Description	Returns the fetch line voltage (Vrms).
Sub-Index	-
Name	FETCh[:SCALar]:LINE:VOLTage[:RMS]
Value	Returns the fetch line voltage value in Vrms.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 06 1C (:FETCh:LINE:VOLTage?)  Return data is 100500. XX 8E 94 88 01 00  The line voltage is 100.5 V, Magnification is 1000.

## Attribute 0x1D: FETCh[:SCALar]:LINE:VOLTage:AVERage

Description	Returns the fetch line voltage average value (Vavg).
Sub-Index	-
Name	FETCh[:SCALar]:LINE:VOLTage:AVERage
Value	Returns the fetch line voltage average value in volts.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 06 1D (:FETCh:LINE:VOLTage:AVERage?)

Return data is 100500.

XX 8E 94 88 01 00

The line voltage average is 100.5 V,  
Magnification is 1000.

**Attribute 0x1E: FETCh[:SCALar]:LINE:VOLTage:HIGH**

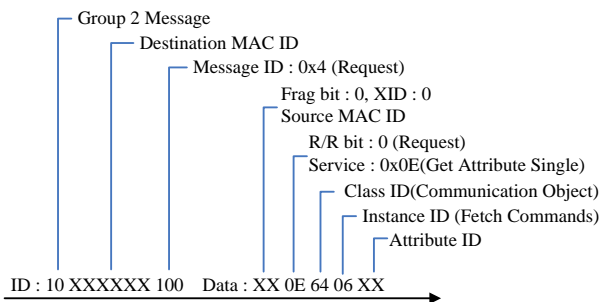
Description	Returns the output fetch voltage maximum peak value (Vmax).
Sub-Index	-
Name	FETCh[:SCALar]:LINE:VOLTage:AVERage
Value	Returns the fetch Vmax value in line volts.
Type	INT32
Data size	4 Bytes
Access	Get
Note	Line voltage maximum peak value is defined as the highest peak value in the complete period.
Example (Data)	XX 0E 64 06 1E (:FETCh:LINE:VOLTage:AVERage:HIGH?)
	Return data is 100500.
	XX 8E 94 88 01 00
	The line voltage maximum peak is 100.5 V, Magnification is 1000.

## Attribute 0x1F: FETCh[:SCALar]:LINE:VOLTage:LOW

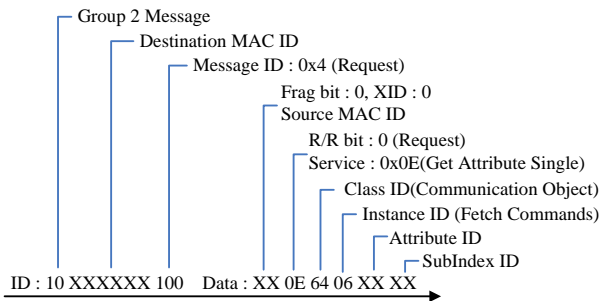
Description	Returns the output fetch current minimum value (Vmin).
Sub-Index	-
Name	FETCh[:SCALar]:LINE:VOLTage:AVERage
Value	Returns the fetch Vmin value in line volts.
Type	INT32
Data size	4 Bytes
Access	Get
Note	Line voltage minimum value is defined as the lowest value in the complete period.
Example (Data)	XX 0E 64 06 1F (:FETCh:LINE:VOLTage:LOW?)  Return data is -50750. XX 8E C2 39 FF FF The line voltage minimum is -50.75 V, Magnification is 1000.

## Request Fetch Commands Example

Request Example no including SubIndex



Request Example including SubIndex





### Input Command (Instance 0x07)

Attribute 0x03: INPut:GAIN .....	145
Attribute 0x04: INPut:SOURce .....	146
Attribute 0x05: INPut:SYNC:SOURce .....	146
Request Input Commands Example .....	147

### Attribute 0x03: INPut:GAIN

Description	Sets or queries the input gain value. (Only AC+DC-EXT or AC-EXT or AC+DC-ADD or AC-ADD or AC-VCA Active)
Sub-Index	-
Name	INPut:GAIN
Value	Input gain value (Set Value = Value * 10)
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Example (Data)	XX 0E 64 07 03 (:INP:GAIN?)  Return data is 1500 (150.0) XX 8E DC 05 00 00  Returns the input gain value as 150.0.

**Attribute 0x04: INPut:SOURce**

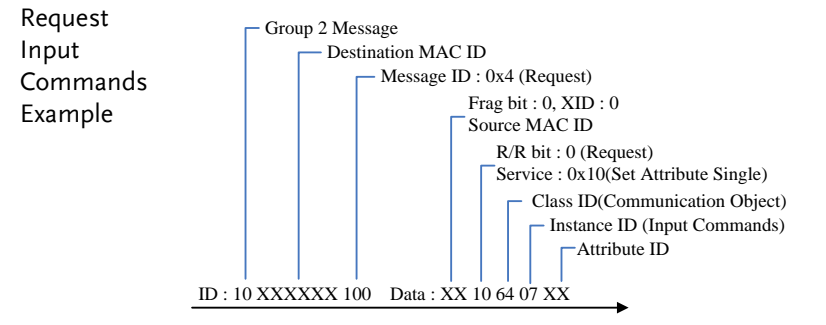
Description	Sets or queries state of source. (Only AC+DC-EXT or AC-EXT or AC+DC-ADD or AC-ADD or AC-VAC Active)
Sub-Index	-
Name	INPut:SOURce
Value	0 = L1EXT L1 EXT source 1 = L2EXT L2 EXT source 2 = L3EXT L3 EXT source
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 0E 64 07 04 (:INP:SOUR?)  Return data is 0 (L1EXT) XX 8E 00 00 00 00  Returns the input gain value as 150.0.

**Attribute 0x05: INPut:SYNC:SOURce**

Description	Sets or queries state of sync source. (Only AC+DC- sync or AC-sync Active)
Sub-Index	-
Name	INPut:SYNC:SOURce
Value	0 = L1 LINE L1 LINE sync source 1 = L2 LINE L2 LINE sync source 2 = L3 LINE L3 LINE sync source 3 = EXT EXT sync source

Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 0E 64 07 05 (:INP:SYNC:SOUR?)  Return data is 3 (EXT) XX 8E 03 00 00 00  Returns the state of sync source as EXT.

### Request Input Commands Example



## Instrumnet Commands (Instance 0x08)

Attribute 0x01: INSTRument:EDIT .....	148
Attribute 0x02: INSTRument:SElect.....	149
Request Instrument Commands Example .....	149

### Attribute 0x01: INSTRument:EDIT

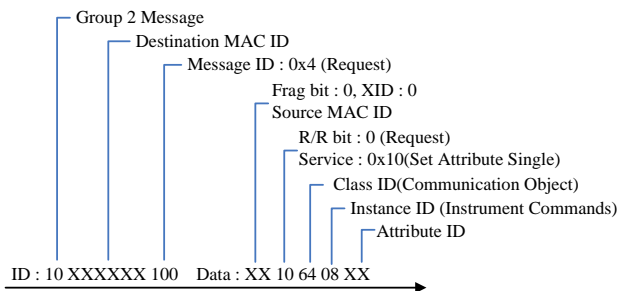
Description	Sets or queries instrument edit. It is convenient to use a programmed command to set all phases at the same time. If INSTR:EDIT ALL has been programmed, it will sent all phases. (Only three-phase four-wire anf signal-phase three-wire Active)
Sub-Index	-
Name	INSTRument:EDIT
Value	0 = EACH    Each phase 1 = ALL     All phase
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 10 64 08 01 01 (:INSTR:EDIT ALL)  Sets instrument edit all phase.

### Attribute 0x02: INSTRument:SElect

Description	Sets or queries the phase to set continuous mode. (Only continuous Mode Active. For single-phase, only L1 can be set. For single-phase three-wire, L1and L2 can be set. )						
Sub-Index	-						
Name	INSTRument:SElect						
Value	From 0 ~ 2, which represent different phase to set sequence, respectively.						
	<table> <tr> <td>0 = L1</td> <td>L1 phase</td> </tr> <tr> <td>1 = L2</td> <td>L2 phase</td> </tr> <tr> <td>2 = L3</td> <td>L3 phase</td> </tr> </table>	0 = L1	L1 phase	1 = L2	L2 phase	2 = L3	L3 phase
0 = L1	L1 phase						
1 = L2	L2 phase						
2 = L3	L3 phase						
Type	UINT8						
Data size	1 Byte						
Access	Get/Set						
Example	XX 10 64 08 02 01 (:INST:SEL L2)						
(Data)	Sets the L2 phase to set continuous mode.						

### Request Instrument Commands Example

Request  
Instrument  
Commands  
Example



## Measure Commands (Instance 0x09)

Attribute 0x01: MEASure:AVERage:COUNT .....	151
Attribute 0x02: MEASure:CONFigure:SENSing.....	152
Attribute 0x03: MEASure:UPDate:RATE .....	152
Attribute 0x04: MEASure[:SCALar]:CURRent:CFACtor .....	153
Attribute 0x05: MEASure[:SCALar]:CURRent:HIGH .....	154
Attribute 0x06: MEASure[:SCALar]:CURRent:LOW .....	155
Attribute 0x07: MEASure[:SCALar]:CURRent:PEAK:HOLD .....	155
Attribute 0x08: MEASure[:SCALar]:CURRent[:RMS] .....	156
Attribute 0x09: MEASure[:SCALar]:CURRent[:RMS]:TOTal .....	156
Attribute 0x0A: MEASure[:SCALar]:CURRent:AC .....	157
Attribute 0x0B: MEASure[:SCALar]:CURRent:AVERage .....	157
Attribute 0x0C: MEASure[:SCALar]:FREQuency .....	158
Attribute 0x0D: MEASure[:SCALar]:PEAK:CLEar .....	159
Attribute 0x0E: MEASure[:SCALar]:POWEr[:AC]:APParent .....	160
Attribute 0x0F:MEASure[:SCALar]:POWEr[:AC]:APParent:TOTal...	160
Attribute 0x10: MEASure[:SCALar]:POWEr[:AC]:PFACtor .....	161
Attribute 0x11:MEASure[:SCALar]:POWEr[:AC]:PFACtor:TOTal ....	161
Attribute 0x12: MEASure[:SCALar]:POWEr[:AC]:REACtive .....	162
Attribute 0x13: MEASure[:SCALar]:POWEr[:AC]:REACtive:TOTal ..	163
Attribute 0x14: MEASure[:SCALar]:POWEr[:AC][:REAL].....	163
Attribute 0x15: MEASure[:SCALar]:POWEr[:AC][:REAL]:TOTal.....	164
Attribute 0x16: MEASure[:SCALar]:VOLTage[:RMS].....	164
Attribute 0x17: MEASure[:SCALar]:VOLTage[:RMS]:TOTal .....	165
Attribute 0x18: MEASure[:SCALar]:VOLTage:AC .....	166
Attribute 0x19: MEASure[:SCALar]:VOLTage:AVERage .....	166
Attribute 0x1A: MEASure[:SCALar]:VOLTage:HIGH.....	167
Attribute 0x1B: MEASure[:SCALar]:VOLTage:LOW .....	167
Attribute 0x1D: MEASure[:SCALar]:CURRent:HARMonic[:RMS] ...	168
Attribute 0x1E: MEASure[:SCALar]:CURRent:HARMonic:RATio ...	173
Attribute 0x1F: MEASure[:SCALar]:VOLTage:HARMonic[:RMS] ....	178
Attribute 0x20: MEASure[:SCALar]:VOLTage:HARMonic:RATio....	183
Attribute 0x21: MEASure[:SCALar]:LINE:VOLTage[:RMS].....	187
Attribute 0x22: MEASure[:SCALar]:LINE:VOLTage:AVERage .....	188
Attribute 0x23: MEASure[:SCALar]:LINE:VOLTage:HIGH.....	189
Attribute 0x24: MEASure[:SCALar]:LINE:VOLTage:LOW .....	189
Request Measure Commands Example .....	190

## Attribute 0x01: MEASure:AVERage:COUNT

Description	Sets or queries the averaging count for Measure Function.
Sub-Index	-
Name	MEASure:AVERage:COUNT
Value	1 ~ 128
Type	UINT8
Data size	1 Byte
Access	Get/Set
Averaged Parameter	Vrms & Vmax & Vmin & Irms & Imax & Imin & P & S & Q & PF & CF & Vavg & Ivag & Ipkh
Not Averaged Parameter	Freq & THDv & THDi
Example (Data)	XX 0E 64 09 01 (:MEASure:AVERage:COUNT?)
	Return data is 1 XX 8E 01 00 00 00
	Returns the averaging count for Measure Function.

## Attribute 0x02: MEASure:CONFigure:SENSing

Description	Sets or queries the remote sense configuration. (Only AC-INT, DC-INT, AC-SYNC Mode and 100V, 200V Range and SIN Wave Shape and Output Impedance is Truned off)	
Sub-Index	-	
Name	MEASure:CONFigure:SENSing	
Value	0 = OFF	Turns the remote sense off.
	1 = ON	Turns the remote sense on.
Type	UINT8	
Data size	1 Byte	
Access	Get/Set	
Example (Data)	XX 10 64 09 02 00 (:MEAS:CONF:SENS 0)	
	Sets the remote sense off.	

## Attribute 0x03: MEASure:UPDate:RATE

Description	Sets or queries the data update interval for Measure Function.	
Sub-Index	-	
Name	MEASure:UPDate:RATE	
Value	0 = FAST	
	1 = 0.1 Sec	
	2 = 0.25 Sec	
	3 = 0.5 Sec	
	4 = 1 Sec	



- 5 = 2 Sec
- 6 = 5 Sec
- 7 = 10 Sec
- 8 = 20 Sec

Type	UINT8
Data size	1 Byte
Access	Get/Set
Use Update Rate Parameter	Vrms & Vmax & Vmin & Irms & Imax & Imin & P & S & Q & PF & CF & Vavg & Ivag & Ipkh
Not Use Update Rate Parameter	Freq & THDv & THDi(update as soon as possible)
Example (Data)	XX 0E 64 09 03 (:MEASure:UPDate:RATE?)

Return data is 0 FAST.  
XX 8E 00 00 00 00

Returns the data update interval for Measure Function.

**Attribute 0x04: MEASure[:SCALar]:CURRent:CFACtor**

Description	Returns the output current crest factor (CF).
Sub-Index	-
Name	MEASure[:SCALar]:CURRent:CFACtor
Value	Returns the crest factor.
Type	INT32
Data size	4 Bytes
Access	Get



## Attribute 0x06: MEASure[:SCALar]:CURRent:LOW

Description	Returns the output current minimum value (Imin).
Sub-Index	-
Name	MEASure[:SCALar]:CURRent:LOW
Value	Returns the Imin value in amps.
Type	INT32
Data size	4 Bytes
Access	Get
Note	Current minimum value is defined as the lowest value in the complete period.
Example (Data)	XX 0E 64 09 06 (:MEAS:CURR:LOW?)  Return data is 1050. XX 8E 1A 04 00 00  The current minimum value is 1.05 A, Magnification is 1000.

## Attribute 0x07: MEASure[:SCALar]:CURRent:PEAK:HOLD

Description	Returns the current peak hold value in amps (IPK Hold).
Sub-Index	-
Name	MEASure[:SCALar]:CURRent:PEAK:HOLD
Value	Returns the peak hold value in amps.
Type	INT32
Data size	4 Bytes
Access	Get
Example	XX 0E 64 09 07

(Data) (:MEAS:CURR:PEAK:HOLD?)

Return data is 20050.

XX 8E 52 4E 00 00

The izeak hold is 20.05 A, Magnification is 1000.

**Attribute 0x08: MEASure[:SCALar]:CURRent[:RMS]**

Description Returns the output current (Irms).

Sub-Index -

Name MEASure[:SCALar]:CURRent[:RMS]

Value Returns the current value in Irms.

Type INT32

Data size 4 Bytes

Access Get

Example XX 0E 64 09 08 (:MEAS:CURR?)

(SDO)

Return data is 10050.

XX 8E 42 27 00 00

The current is 10.05 A, Magnification is 1000.

**Attribute 0x09: MEASure[:SCALar]:CURRent[:RMS]:TOTAL**

Description Returns the total of output current (Irms).

Sub-Index -

Name MEASure[:SCALar]:CURRent[:RMS]:TOTAL

Value Returns the total of current value in Irms.

Type INT32

Data size 4 Bytes

Access	Get
Example (Data)	XX 0E 64 09 09 (:MEAS:CURR:TOTAL?)  Return data is 10050. XX 8E 42 27 00 00  The total of current is 10.05 A, Magnification is 1000.

**Attribute 0x0A: MEASure[:SCALar]:CURRent:AC**

Description	Returns the output AC current (Irms).
Sub-Index	-
Name	MEASure[:SCALar]:CURRent:AC
Value	Returns the AC current value in Irms.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 09 0A (:MEAS:CURR:AC?)  Return data is 10050. XX 8E 42 27 00 00  The AC current is 10.05 A, Magnification is 1000.

**Attribute 0x0B: MEASure[:SCALar]:CURRent:AVERage**

Description	Returns the current average value (Iavg).
Sub-Index	-
Name	MEASure[:SCALar]:CURRent:AVERage
Value	Returns the current average value in amps.

Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 09 0B (:MEAS:CURR:AVERAge?)  Return data is 10050. XX 8E 42 27 00 00 The current average value is 10.05 A, Magnification is 1000.

**Attribute 0x0C: MEASure[:SCALar]:FREQuency**

Description	Returns the SYNC signal source frequency in Hz. (Only AC+DC-sync or AC-sync Active)
Sub-Index	-
Name	MEASure[:SCALar]:FREQuency
Value	Returns the SYNC frequency in Hz
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 09 0C (:MEASure:FREQuency?)  Return data is 50000. XX 8E 50 C3 00 00 The SYNC frequency is 50 Hz, Magnification is 1000.

## Attribute 0x0D: MEASure[:SCALar]:PEAK:CLEar

Description	Clears the current peak-hold value.
Sub-Index	-
Name	MEASure[:SCALar]:PEAK:CLEar
Value	0 = ALL Clear the All phase peak hold value.
	1 = L1 Clear L1 phase peak hold value.
	2 = L2 Clear L2 phase peak hold value.
	3 = L3 Clear L3 phase peak hold value.
Type	UINT8
Data size	1 Byte
Access	Set
Example (Data)	XX 10 64 09 0D 00 (:MEASure[:SCALar]:PEAK:CLEar ALL)  Clear the three-phase and total peak hold value.

**Attribute 0x0E: MEASure[:SCALar]:POWER[:AC]:APParent**

Description	Returns the apparent power (S).
Sub-Index	-
Name	MEASure[:SCALar]:POWER[:AC]:APParent
Value	Returns the apparent power in VA.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 09 0E (:MEASure:POWER:APParent?)  Return data is 2500. XX 8E C4 09 00 00 The apparent power is 2.5, Magnification is 1000.

**Attribute**

**0x0F:MEASure[:SCALar]:POWER[:AC]:APParent:TOTal**

Description	Returns the total of apparent power (S).
Sub-Index	-
Name	MEASure[:SCALar]:POWER[:AC]:APParent:TO Tal
Value	Returns the total of apparent power in VA.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 09 0F (:MEASure:POWER:APParent:TOTal?)



Return data is 2500.

XX 8E C4 09 00 00

The total of apparent power is 2.5,  
Magnification is 1000.

### Attribute 0x10: MEASure[:SCALar]:POWer[:AC]:PFACTOR

Description	Returns the power factor (PF).
Sub-Index	-
Name	MEASure[:SCALar]:POWer[:AC]:PFACTOR
Value	Returns the power factor.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 09 10 (:MEASure:POWer:PFACTOR?)

Return data is 2500.

XX 8E C4 09 00 00

The power factor is 2.5, Magnification is  
1000.

### Attribute

### 0x11:MEASure[:SCALar]:POWer[:AC]:PFACTOR:TOTAL

Description	Returns the total of power factor (PF).
Sub-Index	-
Name	MEASure[:SCALar]:POWer[:AC]:PFACTOR:TOTAL
Value	Returns the total of power factor.

Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 09 11 (:MEASure:POWer:PFACtor:TOTal?)
	Return data is 2500. XX 8E C4 09 00 00 The total of power factor is 2.5, Magnification is 1000.

**Attribute 0x12: MEASure[:SCALar]:POWer[:AC]:REACTive**

Description	Returns the reactive power (Q).
Sub-Index	-
Name	MEASure[:SCALar]:POWer[:AC]:APParent:T OTal
Value	Returns the reactive power in VAR.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 09 12 (:MEASure:POWer:REACTive?)
	Return data is 2500. XX 8E C4 09 00 00 The reactive power is 2.5, Magnification is 1000.

Attribute 0x13:

### MEASure[:SCALar]:POWer[:AC]:REACTiVe:TOTal

Description	Returns the total of reactive power (Q).
Sub-Index	-
Name	MEASure[:SCALar]:POWer[:AC]:REACTiVe:TOTal
Value	Returns the total of reactive power in VAR.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 09 13 (:MEASure:POWer:REACTiVe:TOTal?)  Return data is 2500. XX 8E C4 09 00 00  The reactive power is 2.5, Magnification is 1000.

### Attribute 0x14: MEASure[:SCALar]:POWer[:AC][:REAL]

Description	Returns the active power in Watts (P).
Sub-Index	-
Name	MEASure[:SCALar]:POWer[:AC][:REAL]
Value	Returns the power in Watts.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 09 14 (:MEASure:POWer?)

Return data is 100500.

XX 8E 94 88 01 00

The power is 100.5 W, Magnification is 1000.

**Attribute 0x15: MEASure[:SCALar]:POWER[:AC][:REAL]:TOTal**

Description	Returns the total of active power in Watts (P).
Sub-Index	-
Name	MEASure[:SCALar]:POWER[:AC][:REAL]:TOTal
Value	Returns the total of power in Watts.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 09 15 (:MEASure:POWER:TOTal?)  Return data is 100500. XX 8E 94 88 01 00 The total of power is 100.5 W, Magnification is 1000.

**Attribute 0x16: MEASure[:SCALar]:VOLTage[:RMS]**

Description	Returns the voltage (Vrms).
Sub-Index	-
Name	MEASure[:SCALar]:VOLTage[:RMS]
Value	Returns the voltage value in Vrms.
Type	INT32
Data size	4 Bytes

Access	Get
Example (Data)	XX 0E 64 09 16 (:MEASure:VOLTage?)  Return data is 100500. XX 8E 94 88 01 00 The voltage is 100.5 V, Magnification is 1000.

**Attribute 0x17: MEASure[:SCALar]:VOLTage[:RMS]:TOTal**

Description	Returns the total of voltage (Vrms).
Sub-Index	-
Name	MEASure[:SCALar]:VOLTage[:RMS]:TOTal
Value	Returns the total of voltage value in Vrms.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 09 17 (:MEASure:VOLTage:TOTal?)  Return data is 100500. XX 8E 94 88 01 00 The total of voltage is 100.5 V, Magnification is 1000.

**Attribute 0x18: MEASure[:SCALar]:VOLTage:AC**

Description	Returns the AC voltage (Vrms).
Sub-Index	-
Name	MEASure[:SCALar]:VOLTage:AC
Value	Returns the AC voltage value in Vrms.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 09 18(:MEASure:VOLTage:AC?)  Return data is 100500. XX 8E 94 88 01 00 The AC voltage is 100.5 V, Magnification is 1000.

**Attribute 0x19: MEASure[:SCALar]:VOLTage:AVERage**

Description	Returns the voltage average value (Vavg).
Sub-Index	-
Name	MEASure[:SCALar]:VOLTage:AVERage
Value	Returns the voltage average value in volts.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 09 19 (:MEASure:VOLTage:AC?)  Return data is -50750. XX 8E C2 39 FF FF

---

The voltage average is -50.75 V,  
Magnification is 1000.

---

### Attribute 0x1A: MEASure[:SCALar]:VOLTage:HIGH

---

Description	Returns the output voltage maximum peak value (Vmax).
Sub-Index	-
Name	MEASure[:SCALar]:VOLTage:HIGH
Value	Returns the Vmax value in volts.
Type	INT32
Data size	4 Bytes
Access	Get
Note	Voltage maximum peak value is defined as the highest peak value in the complete period.
Example (Data)	XX 0E 64 09 1A (:MEASure:VOLTage:HIGH?)  Return data is 100500. XX 8E 94 88 01 00 The voltage maximum peak is 100.5 V, Magnification is 1000.

---

### Attribute 0x1B: MEASure[:SCALar]:VOLTage:LOW

---

Description	Returns the output current minimum value (Vmin).
Sub-Index	-
Name	MEASure[:SCALar]:VOLTage:LOW
Value	Returns the Vmin value in volts
Type	INT32

Data size	4 Bytes
Access	Get
Note	Voltage minimum value is defined as the lowest value in the complete period.
Example (Data)	XX 0E 64 09 1B (:MEASure:VOLTage:LOW?)  Return data is -50750. XX 8E C2 39 FF FF The voltage minimum is -50.75 V, Magnification is 1000.

**Attribute 0x1D:**

**MEASure[:SCALar]:CURRent:HARMonic[:RMS]**

Description	Returns 101 values covering Total and order 1 to 100 current (Irms) in harmonic. (Only AC-INT and 50/60 Hz Active)
Sub-Index	0x00
Name	Total current (Irms) in harmonic.
Value	Returns the Total current (Irms) in harmonic.
Type	INT32
Data size	4 Bytes
Access	Get
Sub-Index	0x01
Name	Page 1
Value	Returns the entire 20 values containing order 1 to 20 current (Irms) in harmonic.
Type	INT32



Data size	80 Bytes
Access	Get
Sub-Index	0x02
Name	Page 2
Value	Returns the entire 20 values containing order 21 to 40 current (Irms) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x03
Name	Page 3
Value	Returns the entire 20 values containing order 41 to 60 current (Irms) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x04
Name	Page 4
Value	Returns the entire 20 values containing order 61 to 80 current (Irms) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x05
Name	Page 5
Value	Returns the entire 20 values containing order 81 to 100 current

	(Irms) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get

Example Command 1:  
 (Data) XX 0E 64 09 1D 05  
 (:MEAS:CURR:HARMonic? 5)

Fragment Return 1:  
 8X 00 8E 00 00 00 00  
 (Byte 3 ~ Byte 7 is data)

Fragment Command Ack:  
 8X C0 00

Fragment Return 2:  
 8X 41 00 00 00 00 00  
 (Byte 2 ~ Byte 7 is data)

Fragment Command Ack:  
 8X C1 00

Fragment Return 3:  
 8X 42 00 00 00 00 00  
 (Byte 2 ~ Byte 7 is data)

Fragment Command Ack:  
 8X C2 00

Fragment Return 4:  
 8X 43 00 00 00 00 00  
 (Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C3 00

Fragment Return 5:

8X 44 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C4 00

Fragment Return 6:

8X 45 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C5 00

Fragment Return 7:

8X 46 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C6 00

Fragment Return 8:

8X 47 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C7 00

Fragment Return 9:

8X 48 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C8 00

Fragment Return 10:

8X 49 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C9 00

Fragment Return 11:

8X 4A 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X CA 00

Fragment Return 12:

8X 4B 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X CB 00

Fragment Return 13:

8X 4C 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X CC 00

Fragment Return 14:

8X 8D 00 00 00

(Byte 2 ~ Byte 4 is data)

---

Fragment Command Ack:  
8X CD 00

In a total of 80 bytes of data.

Attribute 0x1E:

MEASure[:SCALar]:CURRent:HARMonic:RATio

Description	Returns 101 values covering Total and order 1 to 100 current (Ratio) in harmonic. (Only AC-INT and 50/60 Hz Active)
Sub-Index	0x00
Name	Total current (Ratio) in harmonic.
Value	Returns the Total current (Ratio) in harmonic.
Type	INT32
Data size	4 Bytes
Access	Get
Sub-Index	0x01
Name	Page 1
Value	Returns the entire 20 values containing order 1 to 20 current (Ratio) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x02
Name	Page 2
Value	Returns the entire 20 values containing order 21 to 40 current (Ratio) in harmonic.

Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x03
Name	Page 3
Value	Returns the entire 20 values containing order 41 to 60 current (Ratio) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x04
Name	Page 4
Value	Returns the entire 20 values containing order 61 to 80 current (Ratio) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x05
Name	Page 5
Value	Returns the entire 20 values containing order 81 to 100 current (Ratio) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Example (Data)	Command 1: XX 0E 64 09 1E 05 (:MEAS:CURR:HARMonic:RATio? 5)
	Fragment Return 1:

8X 00 8E 00 00 00 00 00

(Byte 3 ~ Byte 7 is data)

Fragment Command Ack:

8X C0 00

Fragment Return 2:

8X 41 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C1 00

Fragment Return 3:

8X 42 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C2 00

Fragment Return 4:

8X 43 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C3 00

Fragment Return 5:

8X 44 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C4 00

Fragment Return 6:

---

8X 45 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C5 00

Fragment Return 7:

8X 46 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C6 00

Fragment Return 8:

8X 47 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C7 00

Fragment Return 9:

8X 48 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C8 00

Fragment Return 10:

8X 49 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C9 00

Fragment Return 11:

---



8X 4A 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X CA 00

Fragment Return 12:

8X 4B 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X CB 00

Fragment Return 13:

8X 4C 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X CC 00

Fragment Return 14:

8X 8D 00 00 00

(Byte 2 ~ Byte 4 is data)

Fragment Command Ack:

8X CD 00

---

In a total of 80 bytes of data.

Attribute 0x1F:

## MEASure[:SCALar]:VOLTage:HARMonic[:RMS]

Description	Returns 101 values covering Total and order 1 to 100 voltage (Vrms) in harmonic. (Only AC-INT and 50/60 Hz Active)
Sub-Index	0x00
Name	Total voltage (Vrms) in harmonic.
Value	Returns the Total voltage (Vrms) in harmonic.
Type	INT32
Data size	4 Bytes
Access	Get
Sub-Index	0x01
Name	Page 1
Value	Returns the entire 20 values containing order 1 to 20 voltage (Vrms) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x02
Name	Page 2
Value	Returns the entire 20 values containing order 21 to 40 voltage (Vrms) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get

Sub-Index	0x03
Name	Page 3
Value	Returns the entire 20 values containing order 41 to 60 voltage (Vrms) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x04
Name	Page 4
Value	Returns the entire 20 values containing order 61 to 80 voltage (Vrms) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x05
Name	Page 5
Value	Returns the entire 20 values containing order 81 to 100 voltage (Vrms) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Example (Data)	Command 1: XX 0E 64 09 1F 05 (:MEAS: VOLTage:HARMonic? 5)
	Fragment Return 1: 8X 00 8E 00 00 00 00 00 (Byte 3 ~ Byte 7 is data)

Fragment Command Ack:

8X C0 00

Fragment Return 2:

8X 41 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C1 00

Fragment Return 3:

8X 42 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C2 00

Fragment Return 4:

8X 43 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C3 00

Fragment Return 5:

8X 44 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C4 00

Fragment Return 6:

8X 45 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

---

Fragment Command Ack:

8X C5 00

Fragment Return 7:

8X 46 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C6 00

Fragment Return 8:

8X 47 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C7 00

Fragment Return 9:

8X 48 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C8 00

Fragment Return 10:

8X 49 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C9 00

Fragment Return 11:

8X 4A 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

---

Fragment Command Ack:

8X CA 00

Fragment Return 12:

8X 4B 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X CB 00

Fragment Return 13:

8X 4C 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X CC 00

Fragment Return 14:

8X 8D 00 00 00

(Byte 2 ~ Byte 4 is data)

Fragment Command Ack:

8X CD 00

---

In a total of 80 bytes of data.

---

Attribute 0x20:

MEASure[:SCALar]:VOLTage:HARMonic:RATio

Description	Returns 101 values covering Total and order 1 to 100 Voltage (Ratio) in harmonic. (Only AC-INT and 50/60 Hz Active)
Sub-Index	0x00
Name	Total Voltage (Ratio) in harmonic.
Value	Returns the Total Voltage (Ratio) in harmonic.
Type	INT32
Data size	4 Bytes
Access	Get
Sub-Index	0x01
Name	Page 1
Value	Returns the entire 20 values containing order 1 to 20 Voltage (Ratio) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x02
Name	Page 2
Value	Returns the entire 20 values containing order 21 to 40 Voltage (Ratio) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x03

Name	Page 3
Value	Returns the entire 20 values containing order 41 to 60 Voltage (Ratio) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get

Sub-Index	0x04
Name	Page 4
Value	Returns the entire 20 values containing order 61 to 80 Voltage (Ratio) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get
Sub-Index	0x05

Name	Page 5
Value	Returns the entire 20 values containing order 81 to 100 Voltage (Ratio) in harmonic.
Type	INT32
Data size	80 Bytes
Access	Get

Example  
(Data) Command 1:  
XX 0E 64 09 20 05 (:MEAS:  
VOLTage:HARMonic:RATio? 5)

Fragment Return 1:  
8X 00 8E 00 00 00 00 00  
(Byte 3 ~ Byte 7 is data)



Fragment Command Ack:

8X C0 00

Fragment Return 2:

8X 41 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C1 00

Fragment Return 3:

8X 42 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C2 00

Fragment Return 4:

8X 43 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C3 00

Fragment Return 5:

8X 44 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C4 00

Fragment Return 6:

8X 45 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

---

Fragment Command Ack:

8X C5 00

Fragment Return 7:

8X 46 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C6 00

Fragment Return 8:

8X 47 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C7 00

Fragment Return 9:

8X 48 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C8 00

Fragment Return 10:

8X 49 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X C9 00

Fragment Return 11:

8X 4A 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X CA 00

Fragment Return 12:

8X 4B 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X CB 00

Fragment Return 13:

8X 4C 00 00 00 00 00 00

(Byte 2 ~ Byte 7 is data)

Fragment Command Ack:

8X CC 00

Fragment Return 14:

8X 8D 00 00 00

(Byte 2 ~ Byte 4 is data)

Fragment Command Ack:

8X CD 00

In a total of 80 bytes of data.

### Attribute 0x21: MEASure[:SCALar]:LINE:VOLTage[:RMS]

Description	Returns the line voltage (Vrms).
Sub-Index	-
Name	MEASure[:SCALar]:LINE:VOLTage[:RMS]
Value	Returns the line voltage value in Vrms.

Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 09 21 (:MEASure:LINE:VOLTage?)  Return data is 100500. XX 8E 94 88 01 00 The line voltage is 100.5 V, Magnification is 1000.

**Attribute 0x22: MEASure[:SCALar]:LINE:VOLTage:AVERage**

Description	Returns the line voltage average value (Vavg).
Sub-Index	-
Name	MEASure[:SCALar]:LINE:VOLTage:AVERage
Value	Returns the line voltage average value in volts.
Type	INT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 09 22 (:MEASure:LINE:VOLTage:AVERage?)  Return data is 100500. XX 8E 94 88 01 00 The line voltage average is 100.5 V, Magnification is 1000.

## Attribute 0x23: MEASure[:SCALar]:LINE:VOLTage:HIGH

Description	Returns the output line voltage maximum peak value (Vmax).
Sub-Index	-
Name	MEASure[:SCALar]:LINE:VOLTage:HIGH
Value	Returns the line Vmax value in line volts.
Type	INT32
Data size	4 Bytes
Access	Get
Note	Line voltage maximum peak value is defined as the highest peak value in the complete period.
Example (Data)	XX 0E 64 09 23 (:MEASure:LINE:VOLTage:AVERage:HIGH?)  Return data is 100500. XX 8E 94 88 01 00  The line voltage maximum peak is 100.5 V, Magnification is 1000.

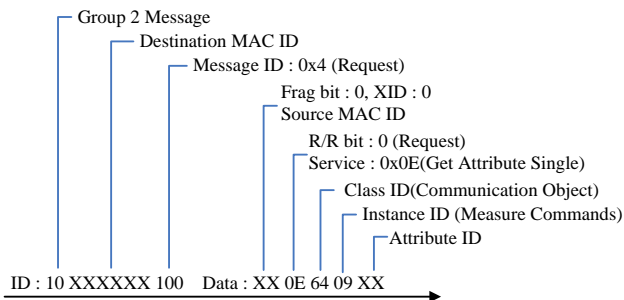
## Attribute 0x24: MEASure[:SCALar]:LINE:VOLTage:LOW

Description	Returns the output line current minimum value (Vmin).
Sub-Index	-
Name	MEASure[:SCALar]:LINE:VOLTage:AVERage
Value	Returns the line Vmin value in line volts.
Type	INT32
Data size	4 Bytes

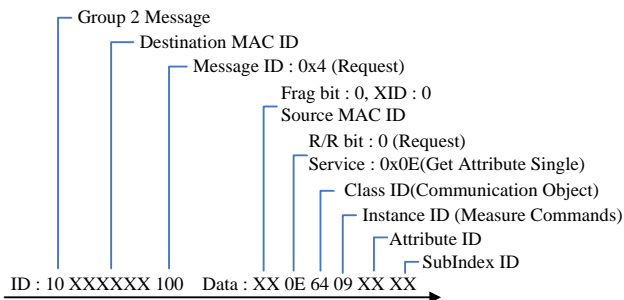
Access	Get
Note	Line voltage minimum value is defined as the lowest value in the complete period.
Example (Data)	XX 0E 64 09 24 (:MEASure:LINE:VOLTage:LOW?)  Return data is -50750. XX 8E C2 39 FF FF The line voltage minimum is -50.75 V, Magnification is 1000.

### Request Measure Commands Example

Request Example no including SubIndex



Request Example including SubIndex



## Memory Commands (Instance 0x0A)

Attribute 0x04: MEMory:SAV .....	191
Attribute 0x05: MEMory:RCL.....	192
Request Memory Commands Example .....	192

### Attribute 0x04: MEMory:SAV

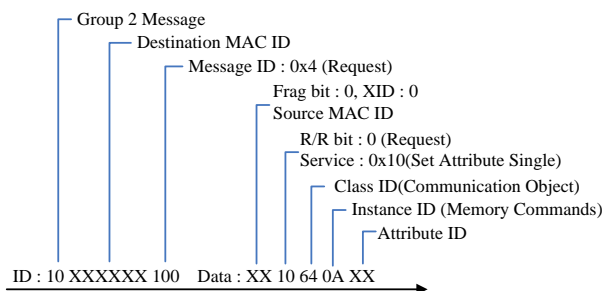
Description	Saves the settings into memory slot M0 ~ M9. These memory slots are mapped to the preset settings. Equivalent to the *SAV command.
Sub-Index	-
Name	MEMory:SAV
Value	0 ~ 9
Type	UINT8
Data size	1 Byte
Access	Set
Example (Data)	XX 10 64 0A 04 01 (:MEMory:SAV 1)  Save the settings to M1.

### Attribute 0x05: MEMory:RCL

Description	Recalls the settings from memory slot M0~M9. These memory slots are mapped to the preset settings. Equivalent to the *RCL command.
Sub-Index	-
Name	MEMory:RCL
Value	0 ~ 9
Type	UINT8
Data size	1 Byte
Access	Set
Example (Data)	XX 10 64 0A 05 01 (:MEMory:RCL 1)  Recall the settings from M1.

### Request Memory Commands Example

Request  
Memory  
Commands  
Example





## Output Commands (Instance 0x0B)

Attribute 0x01: OUTPut:IMPedance .....	193
Attribute 0x02: OUTPut:IMPedance:INDuctance .....	194
Attribute 0x03: OUTPut:IMPedance:RESistance .....	194
Attribute 0x04: OUTPut:MONitor:AMPLitude .....	195
Attribute 0x05: OUTPut:MONitor:SOURce<1   2> .....	196
Attribute 0x06: OUTPut:PON .....	196
Attribute 0x07: OUTPut:PROTection:CLEar .....	197
Attribute 0x08: OUTPut:RELAy .....	197
Attribute 0x0A: OUTPut[:STATe] .....	198
Request Output Commands Example .....	199

## Attribute 0x01: OUTPut:IMPedance

Description	Sets or queries the output impedance state of power source.
Sub-Index	-
Name	OUTPut:IMPedance
Value	0 = OFF Turns the output impedance off. 1 = ON Turns the output impedance on.
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 10 64 0B 01 00 (:OUTP:IMP 0)  Sets power output impedance off.

## Attribute 0x02: OUTPut:IMPedance:INDuctance

Description	Sets or queries the phase and inductance value parameter for output impedance inductance.
Sub-Index	0x00 ~ 0x02 (L1 ~ L3 phase)
Name	OUTPut:IMPedance:INDuctance
Value	inductance : 0.0 ~ 2000 $\mu$ H (Set Value = Value * 10)
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Example (Data)	XX 0E 64 0B 02 00 (OUTP:IMP:IND? L1)  Return data is 1 (0.1 $\mu$ H) XX 8E 01 00 00 00  Returns the L1 phase and inductance value parameter for output impedance inductance.

## Attribute 0x03: OUTPut:IMPedance:RESistance

Description	Sets or queries the phase and inductance value parameter for output impedance resistance.
Sub-Index	0x00 ~ 0x02 (L1 ~ L3 phase)
Name	OUTPut:IMPedance:RESistance
Value	resistance : 0.0 ~ 1 $\Omega$ (Set Value = Value * 10)
Type	UINT16

Data size	2 Bytes
Access	Get/Set
Example (Data)	XX 0E 64 0B 03 00 (:OUTP:IMP:RES? L1)  Return data is 1 (0.1 $\mu$ H) XX 8E 01 00 00 00  Returns the L1 phase and resistance value parameter for output impedance resistance.

**Attribute 0x04: OUTPut:MONitor:AMPLitude**

Description	Sets or queries the range for output monitor amplitude.
Sub-Index	-
Name	OUTPut:MONitor:AMPLitude
Value	0 = $\pm 2.5$ LOW 1 = $\pm 10$ HIGH
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 10 64 0B 04 01 (:OUTP:MON:AMPL HIGH)  Sets the amplitude range to high.

## Attribute 0x05: OUTPut:MONitor:SOURce&lt;1|2&gt;

Description	Sets or queries the source for monitor output1 or monitor output2.(For single-phase, only L1 can be set. For single-phase three-wire, L1 and L2,can be set.)	
Sub-Index	0x00 ~ 0x01 (Output 1 ~ 2)	
Name	OUTPut:MONitor:SOURce<1   2>	
Value	0 = L1Voltage	L1 phase voltage
	1 = L2Voltage	L2 phase voltage
	2 = L3Voltage	L3phase voltage
	3 = L1Current	L1 phase current
	4 = L2Current	L2 phase current
	5 = L3Current	L3 phase current
Type	UINT8	
Data size	1 Byte	
Access	Get/Set	
Example (Data)	XX 10 64 0B 05 00 01 (:OUTP:MON:SOUR1 L2Voltage)	
	Sets the monitor source1 L2 phase voltage.	

## Attribute 0x06: OUTPut:PON

Description	Sets the output state at power-on.	
Sub-Index	-	
Name	OUTPut:PON	
Value	0 = OFF	Disabled
	1 = ON	Enabled
	2 = SEQ	Sequence function

	3 = SIM function	Simulate
Type	UINT8	
Data size	1 Byte	
Access	Get/Set	
Example	XX 10 64 0B 06 02 (:OUTPut:PON 2)	
	Sets sequence function on at power-on.	

**Attribute 0x07: OUTPut:PROTEction:CLEAr**

Description	The Command will clear alarms like Over Current,Over Peak Current, Output Over-Power, Output Short, Output Overvoltage, Sensing Voltage Error.	
Sub-Index	-	
Name	OUTPut:PROTEction:CLEAr	
Value	-	
Type	UINT8	
Data size	1 Byte	
Access	Set	

**Attribute 0x08: OUTPut:RELAy**

Description	Sets or queries the output relay of power source.	
Sub-Index	-	
Name	OUTPut:RELAy	
Value	0 = OFF Disable.	Turns the output relay Disable.
	1 = ON Enable.	Turns the output relay Enable.

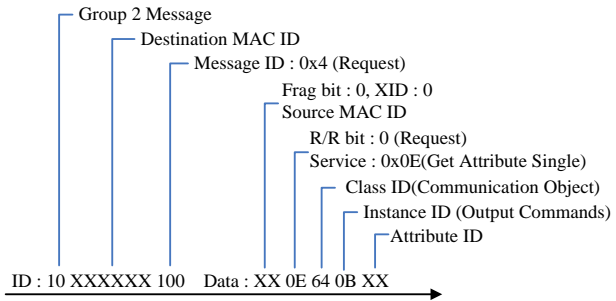
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example	XX 10 64 0B 08 01 (:OUTP:REL 1)
	Sets output relay Enable.

**Attribute 0x0A: OUTPut[:STATe]**

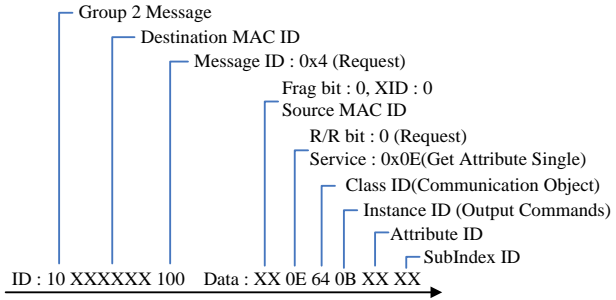
Description	Sets or queries the output state of power source.	
Sub-Index	-	
Name	OUTPut[:STATe]	
Value	0 = OFF	Turns the output off.
	1 = ON	Turns the output on.
Type	UINT8	
Data size	1 Byte	
Access	Get/Set	
Example	XX 10 64 0B 0A 00 (:OUTP 0)	
	Sets power output off.	

## Request Output Commands Example

Request Example no including SubIndex



Request Example including SubIndex



## System Function Commands (Instance 0x0D)

Attribute 0x02: SYSTem:ARBitrary:EDIT:BUILtin .....	201
Attribute 0x0B: SYSTem:ARBitrary:EDIT:SURGe .....	202
Attribute 0x08: SYSTem:ARBitrary:EDIT:STAir .....	204
Attribute 0x03: SYSTem:ARBitrary:EDIT:CFACtor<1   2> .....	204
Attribute 0x04: SYSTem:ARBitrary:EDIT:CLIP.....	205
Attribute 0x09: SYSTem:ARBitrary:EDIT:STORe .....	206
Attribute 0x0C: SYSTem:ARBitrary:EDIT:TRlangle .....	206
Attribute 0x05: SYSTem:ARBitrary:EDIT:DIP .....	207
Attribute 0x06: SYSTem:ARBitrary:EDIT:LFRing .....	209
Attribute 0x07: SYSTem:ARBitrary:EDIT:RIPple.....	212
Attribute 0x0A: SYSTem:ARBitrary:EDIT:STORe:APPLy<1   3>.....	214
Attribute 0x0E: SYSTem:BEEPer:STATe .....	215
Attribute 0x11: SYSTem:COMMunicate:INTerface:ADDRes .....	215
Attribute 0x12: SYSTem:COMMunicate:INTerface:BAUD .....	216
Attribute 0x13: SYSTem:COMMunicate:LAN:DHCP .....	217
Attribute 0x14: SYSTem:COMMunicate:LAN:DNS .....	217
Attribute 0x15: SYSTem:COMMunicate:LAN:GATeway .....	218
Attribute 0x17: SYSTem:COMMunicate:LAN:IPADdress .....	219
Attribute 0x18: SYSTem:COMMunicate:LAN:MAC .....	220
Attribute 0x19: SYSTem:COMMunicate:LAN:SMASK .....	221
Attribute 0x1C: SYSTem:COMMunicate:RLSState .....	222
Attribute 0x1D:	
SYSTem:COMMunicate:SERial[:RECeive]:TRANsmi:t:BAUD.....	223
Attribute 0x1E:	
SYSTem:COMMunicate:SERial[:RECeive]:TRANsmi:t:BITS .....	224
Attribute 0x1F:	
SYSTem:COMMunicate:SERial[:RECeive]:TRANsmi:t:PARity.....	225
Attribute 0x20:	
SYSTem:COMMunicate:SERial[:RECeive]:TRANsmi:t:SBITs.....	226
Attribute 0x21: SYSTem:COMMunicate:TCPIp:CONTRol .....	226
Attribute 0x22: SYSTem:COMMunicate:USB:FRONT:STATe.....	227
Attribute 0x24: SYSTem:COMMunicate:USB:REAR:STATe.....	227
Attribute 0x27: SYSTem:CONFIgure[:MODE] .....	228
Attribute 0x25: SYSTem:CONFIgure:EXTio[:STATe] .....	228
Attribute 0x26: SYSTem:CONFIgure:PHASe .....	228
Attribute 0x28: SYSTem:CONFIgure:TRIGger:OUTPut:SOURce.....	229



Attribute 0x29: SYSTem:CONFigure:TRIGger:OUTPut:WIDTh ..... 230  
 Attribute 0x32: SYSTem:HOLD:STATe..... 231  
 Attribute 0x2E: SYSTem:PKHold:TIME..... 231  
 Attribute 0x2C: SYSTem:KLOCK..... 232  
 Attribute 0x2F: SYSTem:REBoot ..... 232  
 Attribute 0x0D: SYSTem:VUNit..... 232  
 Request System Function Commands Example ..... 233

**Attribute 0x02: SYSTem:ARBitrary:EDIT:BUILtin**

Description	Sets or queries the built in function of arbitrary edit.
Sub-Index	-
Name	SYSTem:ARBitrary:EDIT:BUILtin
Value	0 = TRIangle Built In Triangle Wave Function 1 = STAir Built In Stair Wave Function 2 = CLIP Built In Clip Wave Function 3 = CFACtor1 Built In CF-1 Wave Function 4 = CFACtor2 Built In CF-2 Wave Function 5 = SURGe Built In Surge Wave Function 6~27 = DST01 ~ DST22 Built In DST01 ~ DST22 Wave 28 = RIPple Built In DC Ripple Wave Function 29 = DIP Built In DIP Wave Function.

	30 = LFRing Built In LFRing Wave Function.
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 0E 64 0D 02 (:SYST:ARB:EDIT:BUIL?)  Return data is 0 (TRI) XX 8E 00 00 00 00  Returns the built in function of arbitrary edit.

**Attribute 0x0B: SYSTem:ARBitrary:EDIT:SURGe**

Description	Sets or queries the type and ACV and site parameter for built in Surge wave function.	
Sub-Index	0x00	
Name	Type	
Value	0 = SQU	Square waveform type
	1 = SIN	Sine waveform type
Type	UINT8	
Data size	1 Byte	
Access	Get/Set	
Sub-Index	0x01	
Name	ACV	
Value	ACV Ratio : 0 ~100(0 ~ 100%)	
Type	UINT16	
Data size	2 Bytes	

Access	Get/Set
Sub-Index	0x02
Name	Site
Value	Site Ratio : 0 ~100(0 ~ 100%)
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Example	XX 0E 64 0D 0B 00
(Data)	XX 0E 64 0D 0B 01
	XX 0E 64 0D 0B 02
	(:SYST:ARB:EDIT:SURG?)
	Sub-Index 0 return data is 0 (SIN)
	XX 8E 00 00 00 00
	Sub-Index 1 return data is 50 (ACV: 50%)
	XX 8E 32 00 00 00
	Sub-Index 2 return data is 25 (Site: 25%)
	XX 8E 19 00 00 00
	Returns the type and ACV and site parameter for built in Surge wave function.

## Attribute 0x08: SYSTem:ARBitrary:EDIT:STAir

Description	Sets or queries the stair parameter for built in stair wave function.
Sub-Index	-
Name	SYSTem:ARBitrary:EDIT:STAir
Value	stair : 1 ~ 100
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 0E 64 0D 08 (:SYST:ARB:EDIT:STA?)  Return data is 5 (Stairs: 5) XX 8E 05 00 00 00  Returns the stair parameter for built in stair wave Function.

## Attribute 0x03: SYSTem:ARBitrary:EDIT:CFACTOR&lt;1|2&gt;

Description	Sets or queries the crest factor parameter for built in CF-1, CF-2 wave function.
Sub-Index	0x00
Name	CF-1
Value	crest factor : 1.1 ~ 10.0 (Set Value = Value * 10)
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Example	XX 0E 64 0D 03 00

(Data)	(:SYST:ARB:EDIT:CFAC1?)
	Return data is 20 (CF: 2.0) XX 8E 02 00 00 00
	Returns the crest factor parameter for built in CF-1 wave function
Sub-Index	0x01
Name	CF-2
Value	crest factor : 1.5 ~ 2.0 (Set Value = Value * 10)
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 0E 64 0D 03 01 (:SYST:ARB:EDIT:CFAC2?)
	Return data is 15 (CF: 1.5) XX 8E 14 00 00 00
	Returns the crest factor parameter for built in CF-2 wave function

#### Attribute 0x04: SYSTem:ARBItrary:EDIT:CLIP

Description	Sets or queries the ratio parameter for built in clip wave function.
Sub-Index	-
Name	SYSTem:ARBItrary:EDIT:CLIP
Value	clip ratio : 0.00 ~ 1.00 (Set Value = Value * 100)

Type	UINT16
Data size	2 Bytes
Access	Get/Set
Example (Data)	XX 0E 64 0D 04 (:SYST:ARB:EDIT:CLIP?)  Return data is 50 (CF: 0.50) XX 8E 32 00 00 00  Returns the ratio parameter for built in clip wave function.

#### Attribute 0x09: SYSTem:ARBitrary:EDIT:STORe

Description	Saves the waveform data of built in into ARB1 ~ ARB253.
Sub-Index	-
Name	SYSTem:ARBitrary:EDIT:STORe
Value	0 ~ 252 (ARB1 ~ ARB253)
Type	UINT8
Data size	1 Byte
Access	Set
Example (Data)	XX 10 64 0D 09 00 (:SYST:ARB:EDIT:STOR ARB1)  Saves the waveform data of built in into ARB1

#### Attribute 0x0C: SYSTem:ARBitrary:EDIT:TRIngle

Description	Sets or queries the symmetry parameter for built in triangle wave function.
-------------	---

Sub-Index	-
Name	SYSTem:ARBItrary:EDIT:TRlangle
Value	Symmetry : 0 ~ 100 (0 ~ 100%)
Type	UINT2
Data size	2 Bytes
Access	Get/Set
Example (Data)	XX 0E 64 0D 0C (:SYST:ARB:EDIT:TRI?)  Return data is 50 (Sym: 0.50) XX 8E 32 00 00 00  Returns the symmetry parameter for built in triangle wave function.

#### Attribute 0x05: SYSTem:ARBItrary:EDIT:DIP

Description	Sets or queries the ST Phs and SP Phs and End Phs parameter for built in DIP wave function.
Sub-Index	0x00
Name	STPhs
Value	0.1 ~ (SP Phs - 0.1) (Set Value = Value * 10)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Sub-Index	0x01
Name	SPPhs
Value	(ST Phs+ 0.1) ~ (End Phs - 0.1) (Set Value = Value * 10)

Type	UINT32
Data size	4 Bytes
Access	Get/Set
Sub-Index	0x02
Name	EndPhs
Value	(SP Phs+ 0.1) ~ 359.9 (Set Value = Value * 10)

Type	UINT32
Data size	4 Bytes
Access	Get/Set
Example (Data)	XX 0E 64 0D 05 00 XX 0E 64 0D 05 01 XX 0E 64 0D 05 02 (:SYSTEM:ARbitrary:EDIT:DIP?)

Sub-Index 0 return data is 450 (ST Phs: 45.0)

XX 8E C2 01 00 00

Sub-Index 1 return data is 540 (SP Phs: 54.0)

XX 8E 1C 02 00 00

Sub-Index 2 return data is 1720 (End Phs: 172.0)

XX 8E B8 06 00 00

Returns the ST Phs and SP Phs and End Phs parameter for built in DIP wave function.



## Attribute 0x06: SYSTem:ARBitrary:EDIT:LFRing

Description	Sets or queries the ACV and Amp and Base_F and Ring_F and Decay and ST Phs and End Phs and Ring Phs parameter for built in LFRing wave function.
Sub-Index	0x00
Name	ACV
Value	4.1 ~ 400.0 (Set Value = Value * 100)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Sub-Index	0x01
Name	Amp
Value	140 ~ 200
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Sub-Index	0x02
Name	Base_F
Value	50.0 ~ 200.0 (Set Value = Value * 10)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Sub-Index	0x03
Name	Ring_F

Value	200.0 ~ 5000.0 (Set Value = Value * 10)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Sub-Index	0x04
Name	Decay
Value	-0.100 ~ 0.100 (Set Value = Value * 1000)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Sub-Index	0x05
Name	STPhs
Value	4.1 ~ (End Phs - 0.1) (Set Value = Value * 10)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Sub-Index	0x06
Name	EndPhs
Value	(ST Phs+ 0.1) ~ 359.9 (Set Value = Value * 10)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Sub-Index	0x07

Name	RingPhs
Value	4.1 ~ 359.9 (Set Value = Value * 10)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Example (Data)	XX 0E 64 0D 06 00 XX 0E 64 0D 06 01 XX 0E 64 0D 06 02 XX 0E 64 0D 06 03 XX 0E 64 0D 06 04 XX 0E 64 0D 06 05 XX 0E 64 0D 06 06 XX 0E 64 0D 06 07 (:SYSTem:ARBitrary:EDIT:LFRing?)  Sub-Index 0 return data is 0 (ACV: 45.0) XX 8E 00 00 00 00  Sub-Index 1 return data is 140 (Amp: 140%) XX 8E 8C 00 00 00  Sub-Index 2 return data is 500 (Base_F: 50.0) XX 8E F4 01 00 00  Sub-Index 3 return data is 2000 (Ring_F: 200.0) XX 8E D0 07 00 00

Sub-Index 4 return data is 5 (Decay:  
0.005)

XX 8E 05 00 00 00

Sub-Index 5 return data is 600 (ST Phs:  
60)

XX 8E 58 02 00 00

Sub-Index 6 return data is 1200 (End Phs:  
120.0)

XX 8E B0 04 00 00

Sub-Index 7 return data is 300 (Ring Phs:  
30.0)

XX 8E 2C 01 00 00

Returns the ACV and Amp and Base\_F  
and Ring\_F and Decay and ST Phs and  
End Phs and Ring Phs parameter for built  
in LFRing wave function.

### Attribute 0x07: SYSTem:ARBitrary:EDIT:RIPple

Description	Sets or queries the Times and VDC and Level parameter for built in DC Ripple wave function.
Sub-Index	0x00
Name	Times
Value	Times : 1   2   3   6
Type	UINT16
Data size	2 Bytes

Access	Get/Set
Sub-Index	0x01
Name	VDC
Value	VDC Value : 1 ~ 100
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Sub-Index	0x02
Name	Level
Value	Level Ratio : 1 ~ 30(1 ~ 30%)
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Example	XX 0E 64 0D 07 00
(Data)	XX 0E 64 0D 07 01
	XX 0E 64 0D 07 02
	(:SYST:ARB:EDIT:RIPP?)
	Sub-Index 0 return data is 1 (Times: 1)
	XX 8E 01 00 00 00
	Sub-Index 1 return data is 48 (VDC: 48)
	XX 8E 30 00 00 00
	Sub-Index 2 return data is 15 (Level: 15%)
	XX 8E 0F 00 00 00
	Returns the Times and VDC and Level parameter for built in DC Ripple wave function.

**Attribute 0x0A: SYSTem:ARBItrary:EDIT:STORe:APPLy<1|3>**

**Description** Saves the waveform to L1 or L2 or L3 phase(into ARB1 ~ ARB253)/Output Mode/ACV/DCV/VPK+ Limit/VPK-Limit/V Unit(TRL, ARB) data(for Built in is RIPPLE)

Saves the waveform to L1 or L2 or L3 phase(into ARB1 ~ ARB253)/Output Mode/ACV/DCV/VPK+ Limit/VPK-Limit/V Unit(TRL, ARB)/Freq/Freq Hi Limit/Freq Lo Limit data(for Built in is LFRing)

For single-phase, only L1 can be set. For single-phase three-wire, L1 and L2,can be set.

If instrument edit setting all,apply all phase.

Sub-Index	0x00 ~ 0x02 (SAVE & APPLY L1 ~ L3)
Name	SYSTem:ARBItrary:EDIT:STORe:APPLy<1   3>
Value	0 ~ 252 (ARB1 ~ ARB253)
Type	UINT8
Data size	1 Byte
Access	Set
Example (Data)	XX 10 64 0D 0A 00 01 (:SYST:ARB:EDIT:STOR:APPL1 ARB2)

Saves the waveform(into ARB2) and Output Mode(AC+DC-INT) / ACV / DCV / VPK+ Limit(max) / VPK- Limit(min) / /V Unit(TRL, ARB)(p-p) data(for Built in is RIPPLE)

Saves the waveform(into ARB2) and Output Mode(AC+DC-INT) / ACV / DCV / VPK+ Limit(max) / VPK- Limit(min) / /V Unit(TRI, ARB)(p-p)/Freq/Freq Hi Limit(max)/Freq Lo Limit(min) data(for Built in is LFRing).

#### Attribute 0x0E: SYSTem:BEEPer:STATe

Description	Sets or queries the buzzer state on/off.
Sub-Index	-
Name	SYSTem:BEEPer:STATe
Value	0 = OFF Turns the buzzer off. 1 = ON Turns the buzzer on.
Type	UINT8
Data size	1 Byte
Access	Get/Set

#### Attribute 0x11: SYSTem:COMMunicate:INTerface:ADDRes

Description	Sets or queries the GPIB address or CAN Node ID or DeviceNet MAC ID.
Sub-Index	-
Name	SYSTem:COMMunicate:INTerface:ADDRes
Value	0~30 (GPIB address) 1~127 (CAN Bus Node ID) 0~63 (DeviceNet MAC ID)
Type	UINT8
Data size	1 Byte
Access	Get/Set
Note	Depends on Option interface device.The setting will only be valid after the power has

	been cycled.
Example (Data)	XX 10 64 0D 11 0F(SYST:COMM:INT:ADDR 15)
	Sets the GPIB address or CAN Node ID or DeviceNet MAC ID to 15.

**Attribute 0x12: SYSTem:COMMunicate:INTerface:BAUD**

Description	Description Sets or queries the CAN Baudrate or DeviceNet Baudrate.
Sub-Index	-
Name	SYSTem:COMMunicate:INTerface:BAUD
Value	0 = 125000 1 = 250000 2 = 500000 3 = 1000000 4 = Auto (CAN Bus Baudrate)
	0 = 125000 1 = 250000 2 = 500000 3 = Auto (DeviceNet Baudrate)
Type	UINT8
Data size	1 Byte
Access	Get/Set
Note	Depends on Option interface device.The setting will only be valid after the power has



---

been cycled.

---

Example                   XX 0E 64 0D 12  
(Data)                   (:SYSTem:COMMunicate:INTerface:BAUD?)

Return data is 1 (250000)

XX 8E 01 00 00 00

---

Returns the baud rate settings.

---

### Attribute 0x13: SYSTem:COMMunicate:LAN:DHCP

---

Description	Turns DHCP on/off. Queries the DHCP status.	
Sub-Index	-	
Name	SYSTem:COMMunicate:LAN:DHCP	
Value	0 = OFF	DHCP off
	1 = ON	DHCP on
Type	UINT8	
Data size	1 Byte	
Access	Get/Set	
Note	The setting will only be valid after the power has been cycled.	

### Attribute 0x14: SYSTem:COMMunicate:LAN:DNS

---

Description	Sets or queries the DNS address.	
Note	The setting will only be valid after the power has been cycled.	
Sub-Index	-	
Name	SYSTem:COMMunicate:LAN:DNS	
Value	0 ~ 255	

Type	UINT8
Data size	4 Bytes
Access	Get/Set
Example (Data)	<p>Command 1: 8X 00 10 64 0D 14 FC 01 (Byte 6 ~ Byte 7 is data) Return Fragment Command Ack: 8X C0 00</p> <p>Command 2: 8X 81 10 AC (Byte 2 ~ Byte 3 is data) Return Fragment Command Ack: 8X C1 00 (SYST:COMM:LAN:DNS "172.16.1.252")</p> <p>Sets the DNS to 172.16.1.252.</p>

**Attribute 0x15: SYSTem:COMMunicate:LAN:GATeway**

Description	Sets or queries the Gateway address.
Note	The setting will only be valid after the power has been cycled.
Sub-Index	-
Name	SYSTem:COMMunicate:LAN:GATeway
Value	0 ~ 255
Type	UINT8
Data size	4 Bytes
Access	Get/Set
Example	Command 1:

(Data)	8X 00 10 64 0D 15 FE 00 (Byte 6 ~ Byte 7 is data) Return Fragment Command Ack: 8X C0 00
	Command 2: 8X 81 10 AC (Byte 2 ~ Byte 3 is data) Return Fragment Command Ack: 8X C1 00 (SYST:COMM:LAN:GAT "172.16.0.254")
	Sets the LAN gateway to 172.16.0.254.

#### Attribute 0x17: SYSTem:COMMunicate:LAN:IPADdress

Description	Sets or queries LAN IP address.
Note	The setting will only be valid after the power has been cycled.
Sub-Index	-
Name	SYSTem:COMMunicate:LAN:IPADdress
Value	0 ~ 255
Type	UINT8
Data size	4 Bytes
Access	Get/Set
Example	Command 1: (Data) 8X 00 10 64 0D 17 6F 05 (Byte 6 ~ Byte 7 is data) Return Fragment Command Ack: 8X C0 00

Command 2:

8X 81 10 AC

(Byte 2 ~ Byte 3 is data)

Return Fragment Command Ack:

8X C1 00

(SYST:COMM:LAN:IPAD "172.16.5.111")

Sets the IP address to 172.16.5.111.

**Attribute 0x18: SYSTem:COMMunicate:LAN:MAC**

Description	Returns the unit MAC address as a string. The MAC address cannot be changed.
Sub-Index	-
Name	SYSTem:COMMunicate:LAN:MAC
Value	Returns the MAC address in the following format "FF-FF-FF-FF-FF-FF"
Type	char
Data size	17 Bytes
Access	Get
Example (Data)	Send XX 0E 64 0D 18 (SYST:COMM:LAN:MAC?)

Fragment Return 1:

8X 00 8E 30 32 3A 38 30

(Char data is "02:80")

Fragment Command Ack:

8X C0 00

Fragment Return 2:

8X 41 3A 41 44 3A 32 30

(Char data is “:AD:20”)

Fragment Command Ack:

8X C1 00

Fragment Return 3:

8X 82 3A 33 31 3A 42 31

(Char data is “:31:B1”)

Fragment Command Ack:

8X C2 00

Return (MAC: 02:80:AD:20:31:B1)

Returns the MAC address.

### Attribute 0x19: SYSTem:COMMunicate:LAN:SMASK

Description	Sets or queries the LAN subnet mask.
Note	The setting will only be valid after the power has been cycled.
Sub-Index	-
Name	SYSTem:COMMunicate:LAN:SMASK
Value	0 ~ 255
Type	UINT8
Data size	4 Bytes
Access	Get/Set
Example	Command 1:
(Data)	8X 00 10 64 0D 19 00 00

(Byte 6 ~ Byte 7 is data)

Return Fragment Command Ack:

8X C0 00

Command 2:

8X 81 FF FF

(Byte 2 ~ Byte 3 is data)

Return Fragment Command Ack:

8X C1 00

(SYST:COMM:LAN:SMASK  
"255.255.0.0")

Sets the LAN mask to 255.255.0.0.

**Attribute 0x1C: SYSTem:COMMunicate:RLState**

Description	Enables or disables local/remote state of the instrument.
Sub-Index	-
Name	SYSTem:COMMunicate:RLState
Value	<p>0 = LOCaI</p> <p>All keys are valid. This instrument is controlled by the front panel controls.</p> <p>1 = REMote</p> <p>All keys are invalid, except for the [local] key and the ability to turn the output off.</p> <p>2 = GET/SETLock</p> <p>All keys are invalid. The instrument can only be controlled remotely.</p> <p>3 = LREMote</p> <p>All keys are valid. This instrument is controlled by the front panel controls and</p>

	remotely.
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 10 64 0D 1C 00 (:SYST:COMM:RLST LOCAL)
	Sets the operating mode to local.

**Attribute 0x1D:**

**SYSTEM:COMMunicate:SERial[:RECEive]:TRANsmit:BAUD**

Description	Sets or queries the UART baud rate.
Sub-Index	-
Name	SYSTEM:COMMunicate:SERial[:RECEive]:TRANsmit:BAUD
Value	0 = 1200 1 = 2400 2 = 4800 3 = 9600 4 = 19200 5 = 38400 6 = 57600 7 = 115200
Type	UINT8
Data size	1 Byte
Access	Get/Set
Note	The setting will only be valid after the power has been cycled.
Example	XX 0E 64 0D 1D (SYST:COMM:SER:TRAN:BAUD?)

(Data)

Return data is 3 (Baud rate: 9600)

XX 8E 03 00 00 00

Returns the baud rate settings.

Attribute 0x1E:

**SYSTEM:COMMunicate:SERial[:RECEive]:TRANsmit:BITS**

Description	Sets or queries the UART number of data bits.
Sub-Index	-
Name	SYSTEM:COMMunicate:SERial[:RECEive]:TRANsmit:BITS
Value	0 = 7 bits 1 = 8 bits
Type	UINT8
Data size	1 Byte
Access	Get/Set
Note	The setting will only be valid after the power has been cycled.
Example (Data)	XX 0E 64 0D 1E (SYST:COMM:SER:TRAN:BITS?)  Return data is 1 XX 8E 01 00 00 00  Indicates that 8 data bits are used for the UART connection.



Attribute 0x1F:

SYSTem:COMMunicate:SERial[:RECeive]:TRANsmit:PARity

Description	Sets or queries the parity of the UART connection.	
Sub-Index	-	
Name	SYSTem:COMMunicate:SERial[:RECeive]:TRANsmit:PARity	
Value	0 = NONE	No parity
	1 = ODD	Odd parity
	2 = EVEN	Even parity
Type	UINT8	
Data size	1 Byte	
Access	Get/Set	
Note	The setting will only be valid after the power has been cycled.	
Example (Data)	XX 0E 64 0D 1F (SYST:COMM:SER:TRAN:PARity?)	
	Return data is 0 XX 8E 00 00 00 00	
	Indicates that no parity is used for the UART connection.	

## Attribute 0x20:

SYSTem:COMMunicate:SERial[:RECEive]:TRANsmit:SBITs

Description	Sets or queries the number of stop bits used for the UART connection.
Sub-Index	-
Name	SYSTem:COMMunicate:SERial[:RECEive]:TRANsmit:SBITs
Value	0 = 1 stop bit 1 = 2 stop bits
Type	UINT8
Data size	1 Byte
Access	Get/Set
Note	The setting will only be valid after the power has been cycled.
Example (Data)	XX 0E 64 0D 20 (SYST:COMM:SER:TRAN:SBITs?)  Return data is 1 XX 8E 01 00 00 00  Indicates that one stop bit is used for the UART connection.

Attribute 0x21: SYSTem:COMMunicate:TCPIp:CONTrol

Description	Queries the socket port number.
Sub-Index	-
Name	SYSTem:COMMunicate:TCPIp:CONTrol
Value	0000 ~ 9999
Type	UINT16

Data size	2 Bytes
Access	Get
Example (Data)	XX 0E 64 0D 21 (SYST:COMM:TCP:CONT?)  Return data is 2268 XX 8E DC 08 00 00
	Returns the socket port number.

#### Attribute 0x22: SYSTem:COMMunicate:USB:FRONT:STATe

Description	Queries the front panel USB-A port state.
Sub-Index	-
Name	SYSTem:COMMunicate:USB:FRONT:STATe
Value	0 = Absent 1 = Mass Storage
Type	UINT8
Data size	1 Byte
Access	Get

#### Attribute 0x24: SYSTem:COMMunicate:USB:REAR:STATe

Description	Queries the rear panel USB-B port state.
Sub-Index	-
Name	SYSTem:COMMunicate:USB:REAR:STATe
Value	0 = Absent 1 = Connected to the PC
Type	UINT8
Data size	1 Byte
Access	Get

**Attribute 0x27: SYSTem:CONFigure[:MODE]**

Description	Sets or queries the test mode for the power supply.
Sub-Index	-
Name	SYSTem:CONFigure[:MODE]
Value	0 = CONTinuous Continuous mode (normal operating mode) 1 = SEQuence                      Sequence mode 2 = SIMulation                    Simulation mode
Type	UINT8
Data size	1 Byte
Access	Get/Set

**Attribute 0x25: SYSTem:CONFigure:EXTio[:STATe]**

Description	Sets or queries the external control state on/off.
Sub-Index	-
Name	SYSTem:CONFigure:EXTio[:STATe]
Value	0 = OFF      Turns the external control off. 1 = ON      Turns the external control on.
Type	UINT8
Data size	1 Byte
Access	Get/Set

**Attribute 0x26: SYSTem:CONFigure:PHASe**

Description	Sets or queries the phase configuration . (Only Continuous Mode Active)
Sub-Index	-
Name	SYSTem:CONFigure:PHASe

Value	From 0 - 2 which represent different configure phase, respectively.
	0 = 3P4W
	1 = 1P2W
	2 = 1P3W
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 0E 64 0D 26 (:SYST:CONF:PHAS?)
	Return data is 0 (3P4W) XX 8E 00 00 00 00
	Returns the system configure phase as 3P4W.

## Attribute 0x28:

## SYSTem:CONFIgure:TRIGger:OUTPut:SOURce

Description	Sets or queries the trigger output source. (For single-phase, only L1 can be set. For single-phase three-wire, L1 and L2 can be set. )
Sub-Index	-
Name	SYSTem:CONFIgure:TRIGger:OUTPut:SOURce
Value	From 0 ~ 2, which represent different phase select, respectively.
	0 = L1 L1 phase
	1 = L2 L2 phase
	2 = L3 L3 phase

Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 10 64 0D 28 01 (SYST:CONF:TRIG:OUTP:SOUR L2)
Sets the trigger output source to L2.	

**Attribute 0x29:**

**SYSTEM:CONFigure:TRIGger:OUTPut:WIDTh**

Description	Sets or queries the type of trigger output. The trigger output can be set as a user-defined pulse width or as a trigger output level.
Sub-Index	-
Name	SYSTEM:CONFigure:TRIGger:OUTPut:WIDTh
Value	0.1 ~ 60 mSec (Set Value = Value * 10)
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Example (Data)	XX 10 64 0D 29 32 (:SYST:CONF:TRIG:OUTP:WIDT 0.5)
Sets the trigger output width to 5ms.	

## Attribute 0x32: SYSTem:HOLD:STATe

Description	Sets or queries the freeze hold state on/off.	
Sub-Index	-	
Name	SYSTem:HOLD:STATe	
Value	0 = OFF	Turns the freeze hold off.
	1 = ON	Turns the freeze hold on.
Type	UINT8	
Data size	1 Byte	
Access	Get/Set	

## Attribute 0x2E: SYSTem:PKHold:TIME

Description	Sets or queries the Ipeak hold time for peak current measurement when output on.	
Sub-Index	-	
Name	SYSTem:PKHold:TIME	
Value	1 ~ 60,000	
Type	UINT16	
Data size	2 Bytes	
Access	Get/Set	
Example (Data)	XX 10 64 0D 2E 0A 00 (:SYST:PKH:TIME 10)	
	Sets the Ipeak hold time 10ms to measure when output on.	

## Attribute 0x2C: SYSTem:KLOCK

Description	Enables or disables the front panel key lock.
Sub-Index	-
Name	SYSTem:KLOCK
Value	0 = OFF    Panel keys unlocked 1 = ON    Panel keys locked
Type	UINT8
Data size	1 Byte
Access	Get/Set

## Attribute 0x2F: SYSTem:REBoot

Description	Reboots the ASR system.
Sub-Index	-
Name	SYSTem:REBoot
Value	-
Type	UINT8
Data size	1 Byte
Access	Set

## Attribute 0x0D: SYSTem:VUNit

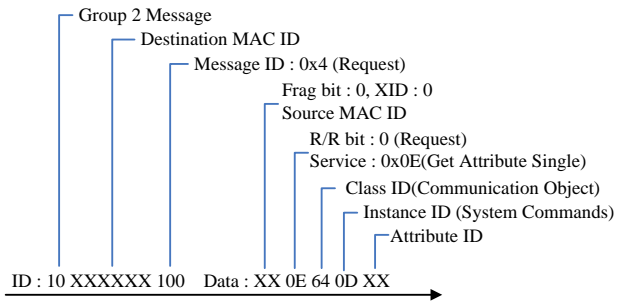
Description	Sets or Queries the Unit of Voltage Setting in Specific Wave Shape(TRI or ARB)
Sub-Index	-
Name	SYSTem:VUNit
Value	0 = RMS    Sets V Unit (TRI, ARB) as rms 1 = P-P    Sets V Unit (TRI, ARB) as p-p
Type	UINT8



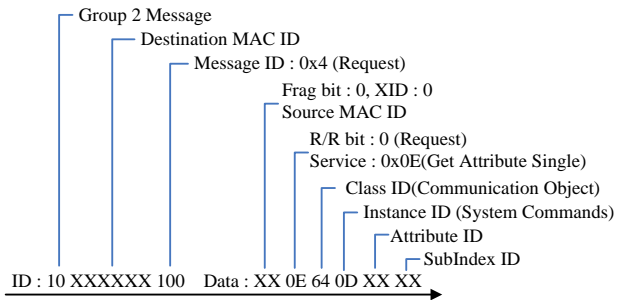
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 0E 64 0D 0D (:SYSTem:VUNit?)
	Return data is 0 (RMS ) XX 8E 00 00 00 00
	Returns the V Unit(TRI, ARB) setting.

### Request System Function Commands Example

Request Example no including SubIndex



Request Example including SubIndex



## Trigger Commands (Instance 0x0F)

Attribute 0x05: TRIGger:SEQuence:SELEcted:EXECute .....	234
Attribute 0x06: TRIGger:SIMulation:SELEcted:EXECute.....	235
Request Trigger Commands Example .....	235

### Attribute 0x05: TRIGger:SEQuence:SELEcted:EXECute

Description	Sets to execute actions for sequence mode. (Only Sequence Mode Active)	
Sub-Index	-	
Name	TRIGger:SEQuence:SELEcted:EXECute	
Value	0 = STOP	Stops sequence execution
	1 = STARt	Starts sequence execution
	2 = HOLD	Holds sequence execution
	3 = BRAN1 execution	Jumps to Branch 1
	4 = BRAN2 execution	Jumps to Branch 2
Type	UINT8	
Data size	1 Byte	
Access	Get/Set	
Example (Data)	XX 10 64 0F 05 01 (TRIG:SEQ:SEL:EXEC STAR)	
	Starts sequence execution.	

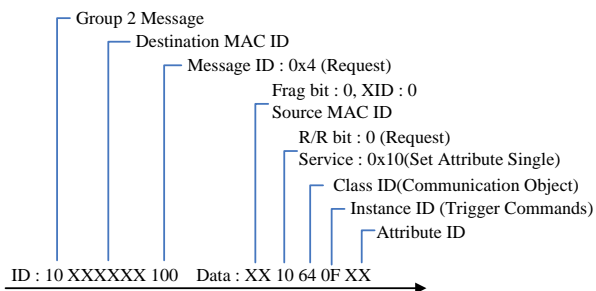
### Attribute 0x06: TRIGger:SIMulation:SElected:EXECute

Description	Sets to execute actions for simulate mode. (Only Simulation Mode Active)	
Sub-Index	-	
Name	TRIGger:SIMulation:SElected:EXECute	
Value	0 = STOP	Stops simulate execution
	1 = START	Starts simulate execution
	2 = HOLD	Holds simulate execution
Type	UINT8	
Data size	1 Byte	
Access	Get/Set	
Example (Data)	XX 10 64 0F 06 01 (:TRIG:SIM:SEL:EXEC STAR)	

Starts simulate execution.

### Request Trigger Commands Example

Request  
Trigger  
Commands  
Example



## Source Commands (Instance 0x11)

Attribute 0x01: [:SOURCE]:CURRENT:LIMIT:PEAK:HIGH.....	238
Attribute 0x02: [:SOURCE]:CURRENT:LIMIT:PEAK:LOW .....	238
Attribute 0x03: [:SOURCE]:CURRENT:LIMIT:PEAK:MODE .....	239
Attribute 0x04: [:SOURCE]:CURRENT:LIMIT:RMS[:AMPLITUDE].....	240
Attribute 0x05: [:SOURCE]:CURRENT:LIMIT:RMS:MODE .....	240
Attribute 0x06: [:SOURCE]:FREQUENCY:LIMIT:HIGH.....	241
Attribute 0x07: [:SOURCE]:FREQUENCY:LIMIT:LOW .....	242
Attribute 0x08: [:SOURCE]:FREQUENCY[:IMMEDIATE] .....	242
Attribute 0x09: [:SOURCE]:FUNCTION[:SHAPE][:IMMEDIATE].....	243
Attribute 0x0A: [:SOURCE]:FUNCTION:THD:FORMAT .....	244
Attribute 0x0B: [:SOURCE]:LINE:VOLTAGE[:LEVEL][:IMMEDIATE][:AMPLITUDE] .....	246
Attribute 0x0C: [:SOURCE]:LINE:VOLTAGE[:LEVEL][:IMMEDIATE]:OFFSET .....	246
Attribute 0x0D: [:SOURCE]:PHASE:BALANCE.....	247
Attribute 0x0E: [:SOURCE]:PHASE:MODE .....	248
Attribute 0x0F: [:SOURCE]:PHASE:PHASE.....	249
Attribute 0x10: [:SOURCE]:PHASE:RELOCK .....	250
Attribute 0x11: [:SOURCE]:PHASE:SETCHANGE:STATE .....	250
Attribute 0x12: [:SOURCE]:PHASE:START[:IMMEDIATE].....	251
Attribute 0x13: [:SOURCE]:PHASE:START:STATE.....	251
Attribute 0x14: [:SOURCE]:PHASE:STOP[:IMMEDIATE] .....	252
Attribute 0x15: [:SOURCE]:PHASE:STOP:STATE .....	252
Attribute 0x16: [:SOURCE]:PHASE:SYNC[:IMMEDIATE].....	253
Attribute 0x17: [:SOURCE]:READ .....	254
Attribute 0x30: [:SOURCE]:SEQUENCE:CONDITION.....	256
Attribute 0x31: [:SOURCE]:SEQUENCE:CPARAMETER.....	256
Attribute 0x32: [:SOURCE]:SEQUENCE:CSTEP .....	260
Attribute 0x33: [:SOURCE]:SEQUENCE:INSTRUMENT:SELECT .....	261
Attribute 0x34: [:SOURCE]:SEQUENCE:SPARAMETER .....	261
Attribute 0x35: [:SOURCE]:SEQUENCE:STEP .....	264
Attribute 0x36: [:SOURCE]:SIMULATION:CONDITION.....	265
Attribute 0x37: [:SOURCE]:SIMULATION:ABNORMAL:CODE.....	265
Attribute 0x38: [:SOURCE]:SIMULATION:ABNORMAL:FREQUENCY.....	266
Attribute 0x39: [:SOURCE]:SIMULATION:ABNORMAL:PHASE:START:ENABLE .....	267

Attribute 0x3A:  
 [:SOURce]:SIMulation:ABNormal:PHASe:STARt[:IMMediate] ..... 267  
 Attribute 0x3B  
 [:SOURce]:SIMulation:ABNormal:PHASe:STOP:ENABle ..... 269  
 Attribute 0x3C:  
 [:SOURce]:SIMulation:ABNormal:PHASe:STOP[:IMMediate] ..... 269  
 Attribute 0x3D: [:SOURce]:SIMulation:ABNormal:TIME..... 270  
 Attribute 0x3E: [:SOURce]:SIMulation:ABNormal:VOLTage..... 271  
 Attribute 0x3F: [:SOURce]:SIMulation:CSTep ..... 271  
 Attribute 0x40: [:SOURce]:SIMulation:INITial:CODE ..... 272  
 Attribute 0x41: [:SOURce]:SIMulation:INITial:FREQuency ..... 273  
 Attribute 0x42: [:SOURce]:SIMulation:INITial:PHASe:STARt:ENABle  
 ..... 273  
 Attribute 0x43:  
 [:SOURce]:SIMulation:INITial:PHASe:STARt[:IMMediate]..... 274  
 Attribute 0x44: [:SOURce]:SIMulation:INITial:PHASe:STOP:ENABle  
 ..... 275  
 Attribute 0x45:  
 [:SOURce]:SIMulation:INITial:PHASe:STOP[:IMMediate] ..... 275  
 Attribute 0x46: [:SOURce]:SIMulation:INITial:VOLTage ..... 276  
 Attribute 0x48: [:SOURce]:SIMulation:NORMal<1 | 2>:CODE..... 277  
 Attribute 0x49: [:SOURce]:SIMulation:NORMal 1:FREQuency ..... 278  
 Attribute 0x4A:  
 [:SOURce]:SIMulation:NORMal<1 | 2>:PHASe:STARt:ENABle ..... 279  
 Attribute 0x4B:  
 [:SOURce]:SIMulation:NORMal<1 | 2>:PHASe:STARt[:IMMediate]..... 280  
 Attribute 0x4C:  
 [:SOURce]:SIMulation:NORMal<1 | 2>:PHASe:STOP:ENABle..... 281  
 Attribute 0x4D:  
 [:SOURce]:SIMulation:NORMal<1 | 2>:PHASe:STOP[:IMMediate] ..... 282  
 Attribute 0x4E: [:SOURce]:SIMulation:NORMal<1 | 2>:TIME..... 283  
 Attribute 0x4F: [:SOURce]:SIMulation:NORMal 1:VOLTage ..... 284  
 Attribute 0x50: [:SOURce]:SIMulation:REPeat:COUNt..... 285  
 Attribute 0x51: [:SOURce]:SIMulation:REPeat:ENABle ..... 285  
 Attribute 0x53: [:SOURce]:SIMulation:TRANsition<1 | 2>:TIME..... 286  
 Attribute 0x52: [:SOURce]:SIMulation:TRANsition<1 | 2>:CODE ..... 287  
 Attribute 0x60: [:SOURce]:SEQuence:NSParameter..... 288  
 Request Source Commands Example ..... 291

**Attribute 0x01: [:SOURce]:CURRent:LIMit:PEAK:HIGH**

Description	Sets or queries the Ipk-High Limit parameter for the continuous operation mode.
Sub-Index	-
Name	[:SOURce]:CURRent:LIMit:PEAK:HIGH
Value	Ipk-High Limit in Arms. (Set Value = Value * 100)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Example (Data)	XX 0E 64 11 01 (CURR:LIM:PEAK:HIGH?)  Return data is 4200 (42.00) XX 8E 68 10 00 00  Returns the peak current high limit as +42.0 A.

**Attribute 0x02: [:SOURce]:CURRent:LIMit:PEAK:LOW**

Description	Sets or queries the Ipk-Low Limit parameter for the continuous operation mode.
Sub-Index	-
Name	[:SOURce]:CURRent:LIMit:PEAK:LOW
Value	Ipk-Low Limit in Arms. (Set Value = Value * 100)
Type	INT32

Data size	4 Bytes
Access	Get/Set
Example (Data)	XX 0E 64 11 02 (:CURR:LIM:PEAK:LOW?)  Return data is -4200 (-42.00) XX 8E 98 EF FF FF  Returns the peak current low limit as -42.0 A.

### Attribute 0x03: [:SOURce]:CURRent:LIMit:PEAK:MODE

Description	Sets or queries Ipk limit enabled or disabled.
Sub-Index	-
Name	[:SOURce]:CURRent:LIMit:PEAK:MODE
Value	0 = OFF      Ipk limit off 1 = ON        Ipk limit on
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 10 64 11 03 01 (:CURR:LIM:PEAK:MODE ON)  Sets Ipk limit enabled.

## Attribute 0x04: [:SOURce]:CURRent:LIMit:RMS[:AMPLitude]

Description	Sets or queries the Irms parameter for the continuous operation mode.
Sub-Index	-
Name	[:SOURce]:CURRent:LIMit:RMS[:AMPLitude]
Value	Irms in A. (Set Value = Value * 100)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Example (Data)	XX 0E 64 11 04 (:CURR:LIM:RMS?)  Return data is 1050 (10.50) XX 8E 1A 04 00 00  Returns the Irms setting.

## Attribute 0x05: [:SOURce]:CURRent:LIMit:RMS:MODE

Description	Sets or queries IRMS limit status.
Sub-Index	-
Name	[:SOURce]:CURRent:LIMit:RMS:MODE
Value	0 = OFF IRMS limit off 1 = ON IRMS limit on
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example	XX 10 64 11 05 01



---

(Data) (:CURR:LIM:RMS:MODE ON)

---

Sets IRMS limit enabled.

---

### Attribute 0x06: [:SOURce]:FREQuency:LIMit:HIGH

---

Description	Sets or queries the frequency upper limit range. (Only AC+DC-INT or AC-INT or AC+DC-ADD or AC-ADD or AC-VCA Active)
Sub-Index	-
Name	[:SOURce]:FREQuency:LIMit:HIGH
Value	Frequency in Hz. (Set Value = Value * 100)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Example (Data)	XX 0E 64 11 06 (FREQ:LIM:HIGH?)  Return data is 99990 (999.90) XX 8E 96 86 01 00
	Returns the frequency upper limit.

---

**Attribute 0x07: [:SOURce]:FREQuency:LIMit:LOW**

Description	Sets or queries the frequency lower limit range. (Only AC+DC-INT or AC-INT or AC+DC-ADD or AC-ADD or AC-VCA Active)
Sub-Index	-
Name	[:SOURce]:FREQuency:LIMit:LOW
Value	Frequency in Hz. (Set Value = Value * 100)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Example (Data)	XX 0E 64 11 07 (FREQ:LIM:LOW?)  Return data is 100 (1.00) XX 8E 64 00 00 00  Returns the frequency lower limit.

**Attribute 0x08: [:SOURce]:FREQuency[:IMMEDIATE]**

Description	Sets or queries the frequency for the immediate trigger. (Only AC+DC-INT or AC-INT or AC+DC- ADD or AC-ADD or AC-VCA Active)
Sub-Index	-
Name	[:SOURce]:FREQuency[:IMMEDIATE]
Value	Frequency setting in Hz. (Set Value = Value * 100)
Type	UINT32

Data size	4 Bytes
Access	Get/Set
Example (Data)	XX 10 64 11 08 70 17 (:FREQ 60)
	Sets the frequency of 60Hz.

**Attribute 0x09: [:SOURce]:FUNctIon[:SHAPE][:IMMediate]**

Description	Sets or queries the waveforms of power supply. (Not available for DC-INT, AC+DC-EXT and AC- EXT)	
Sub-Index	-	
Name	[:SOURce]:FUNctIon[:SHAPE][:IMMediate]	
Value	0 = ARB1	Arbitrary wave 1
	1 = ARB2	Arbitrary wave 2
	2 = ARB3	Arbitrary wave 3
	3 = ARB4	Arbitrary wave 4
		.
		.
		.
	12 = ARB13	Arbitrary wave 13
	13 = ARB14	Arbitrary wave 14
	14 = ARB15	Arbitrary wave 15
	15 = ARB16	Arbitrary wave 16
	16 = SIN	Sin wave
	17 = SQU	Square wave
	18 = TRI	Triangle wave
	19 = ARB17	Arbitrary wave 17
	20 = ARB18	Arbitrary wave 18
	21 = ARB19	Arbitrary wave 19
	22 = ARB20	Arbitrary wave 20

- 
- 
- 

252 = ARB250 Arbitrary wave 250  
 253 = ARB251 Arbitrary wave 251  
 254 = ARB252 Arbitrary wave 252  
 255 = ARB253 Arbitrary wave 253

Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 0E 64 11 09 (:SOUR:FUNC:SHAP:IMM?)  Return data is 18 (TRI) XX 8E 12 00 00 00  Returns the waveform as Triangle wave.

**Attribute 0x0A: [:SOURce]:FUNCtion:THD:FORMat**

Description	Sets or queries the THD format.
Sub-Index	-
Name	[:SOURce]:FUNCtion:THD:FORMat
Value	0 = IEC IEC THD format 1 = CSA CSA THD format
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 0E 64 11 0A (:SOUR:FUNC:THD:FORM?)

Return data is 0 (IEC)  
XX 8E 00 00 00 00

Returns the THD format as IEC.

---

**Attribute 0x0B:**

**[:SOURce]:LINE:VOLTage[:LEVel][:IMMEDIATE][:AMPLitude]**

Description	Sets or queries the RMS line voltage for the continuous operation mode. (Not available in phase mode independent and DC-INT, AC+DC-EXT, AC-EXT, AC+DC-ADD, AC-ADD and AC-VCA)
Sub-Index	-
Name	[:SOURce]:LINE:VOLTage[:LEVel][:IMMEDIATE][:AMPLitude]
Value	Vrms. (Set Value = Value * 100)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Example (Data)	XX 10 64 11 0B 98 3A (:LINE:VOLT 150.0)  Sets the line voltage to 150.0 ACV.

**Attribute 0x0C:**

**[:SOURce]:LINE:VOLTage[:LEVel][:IMMEDIATE]:OFFSet**

Description	Sets or queries the line voltage offset value. (Not available in phase mode independent and only AC+DC-INT or AC+DC-Sync Active)
Sub-Index	-
Name	[:SOURce]:LINE:VOLTage[:LEVel][:IMMEDIATE]:OFFSet
Value	Voltage offset value (Set Value = Value * 100)

Type	INT32
Data size	4 Bytes
Access	Get/Set
Example (Data)	XX 10 64 11 0C 98 3A (:LINE:VOLT:OFFS 150.0)
	Sets the line voltage offset value to 150.0 DCV.

**Attribute 0x0D: [:SOURce]:PHASe:BALance**

Description	Sets or queries the balance setting phase or line. (Only phase mode is balance Active)
Sub-Index	-
Name	[:SOURce]:PHASe:BALance
Value	0 = PHASe      Phase setting 1 = LINE        Line setting
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 0E 64 11 0D (:PHAS:BAL?)  Return data is 1 (LINE) XX 8E 01 00 00 00
	Returns the balance setting as LINE.

## Attribute 0x0E: [:SOURce]:PHASe:MODE

Description	Sets or queries the balance mode. (Only three-phase four-wire or single-phase three-wire Active)
Sub-Index	-
Name	[:SOURce]:PHASe:MODE
Value	0 = Independ 1 = Balance
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 0E 64 11 0E (:PHAS:MODE?)  Return data is 1 (Balance) XX 8E 01 00 00 00  Returns the phase mode as Balance.



## Attribute 0x0F: [:SOURce]:PHASe:PHASe

Description	Sets or queries the target and phase angle parameter. (For three-phase four-wire, L12 and L13 can be set. For single-phase three-wire, L12 can be set. )
Sub-Index	0x00
Name	Phase angle between L1-L2
Value	0° ~ 359.9° (Set Value = Value * 10)
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Sub-Index	0x01
Name	Phase angle between L1-L3
Value	0° ~ 359.9° (Set Value = Value * 10)
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Example (Data)	XX 0E 64 11 0F 00 (:PHAS:PHAS? L12)  Return data is 1200 (120.0) XX 8E B0 04 00 00  Returns the Phase angle 120.0 between L1-L2.

**Attribute 0x10: [:SOURce]:PHASe:RELock**

Description	Sets or queries the relock function in three-phase mode.	
Sub-Index	-	
Name	[:SOURce]:PHASe:RELock	
Value	0 = OFF	Phase relock off
	1 = ON	Phase relock on
Type	UINT8	
Data size	1 Byte	
Access	Get/Set	
Example (Data)	XX 10 64 11 10 01 (:PHAS:REL ON)	
	Sets Phase relock enabled.	

**Attribute 0x11: [:SOURce]:PHASe:SETChange:STATe**

Description	Sets or queries the set change phase state.	
Sub-Index	-	
Name	[:SOURce]:PHASe:SETChange:STATe	
Value	0 = OFF	Set change phase off
	1 = ON	Set change phase on
Type	UINT8	
Data size	1 Byte	
Access	Get/Set	
Example (Data)	XX 10 64 11 11 01 (:PHAS:SETC:STAT ON)	
	Sets change phase enabled.	

## Attribute 0x12: [:SOURce]:PHASe:STARt[:IMMediate]

Description	Sets or queries the start phase. (Not available for DC-INT, AC+DC-EXT and AC-EXT)
Sub-Index	-
Name	[:SOURce]:PHASe:STARt[:IMMediate]
Value	Start phase value $0^{\circ} \sim 359.9^{\circ}$ (Set Value = Value * 10)
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Example (Data)	XX 10 64 11 12 00 00 (:PHAS:STAR 0)  Sets the starting phase to 0.

## Attribute 0x13: [:SOURce]:PHASe:STARt:STATe

Description	Sets or queries state of start phase. (Not available for DC-INT, AC+DC-EXT and AC-EXT)
Sub-Index	-
Name	[:SOURce]:PHASe:STARt:STATe
Value	0 = FREE      Start phase Free 1 = FIXED      Start phase Fixed
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 0E 64 11 13 (:PHAS:STAR:STAT?)  Return data is 0 (FREE)

---

 XX 8E 00 00 00 00
 

---

 Returns the state of start phase as Free.
 

---

### Attribute 0x14: [:SOURce]:PHASe:STOP[:IMMEDIATE]

---

Description	Sets or queries the off phase of the waveform. (Not available for DC-INT, AC+DC-EXT and AC-EXT)
Sub-Index	-
Name	[:SOURce]:PHASe:STOP[:IMMEDIATE]
Value	Stop phase value $0^{\circ} \sim 359.9^{\circ}$ (Set Value = Value * 10)
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Note	Sets the off phase of the waveform after the output has been turned off.
Example (Data)	XX 10 64 11 14 58 02 (:PHAS:STOP 60)  Sets the stop phase to 60.

---

### Attribute 0x15: [:SOURce]:PHASe:STOP:STATe

---

Description	Sets or queries state of stop phase. (Not available for DC-INT, AC+DC-EXT and AC-EXT)
Sub-Index	-
Name	[:SOURce]:PHASe:STOP:STATe
Value	0 = FREE      Stop phase Free 1 = FIXED      Stop phase Fixed

Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 0E 64 11 15 (:PHAS:STOP:STAT?)  Return data is 0 (FIXED) XX 8E 01 00 00 00  Returns the state of stop phase as Fixed.

### Attribute 0x16: [:SOURce]:PHASe:SYNC[:IMMEDIATE]

Description	Sets or queries the sync delay phase. (Only AC+DC-sync or AC-sync Active)
Sub-Index	-
Name	[:SOURce]:PHASe:SYNC[:IMMEDIATE]
Value	Sync delay phase value $0^{\circ} \sim 359.9^{\circ}$ (Set Value = Value * 10)
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Example (Data)	XX 10 64 11 16 00 00 (:PHAS:SYNC 0)  Sets the sync delay phase to 0.

## Attribute 0x17: [:SOURce]:READ

Description	Returns the measurement readouts.
Sub-Index	0x00 = L1 Vrms
	0x01 = L1 Vavg
	0x02 = L1 Vmax
	0x03 = L1 Vmin
	0x04 = L1 Vpkh
	0x05 = L1 Irms
	0x06 = L1 Iavg
	0x07 = L1 Imax
	0x08 = L1 Imin
	0x09 = L1 Ipkh
	0x0A = L1 P
	0x0B = L1 S (Invalid in DC-INT mode.)
	0x0C = L1 Q (Invalid in DC-INT mode.)
	0x0D = L1 PF (Invalid in DC-INT mode.)
	0x0E = L1 CF (Invalid in DC-INT mode.)
	0x0F = L1 THD <sub>v</sub> (AC-INT mode only) / L1 Freq (AC+DC-Sync and AC-Sync modes only)
	0x10 = L1 THD <sub>i</sub> (AC-INT mode only)
	0x11 = L2 Vrms
	0x12 = L2 Vavg
	0x13 = L2 Vmax
	0x14 = L2 Vmin
	0x15 = L2 Vpkh
	0x16 = L2 Irms
	0x17 = L2 Iavg
	0x18 = L2 Imax
	0x19 = L2 Imin

0x1A = L2 I<sub>pkh</sub>  
 0x1B = L2 P  
 0x1C = L2 S (Invalid in DC-INT mode.)  
 0x1D = L2 Q (Invalid in DC-INT mode.)  
 0x1E = L2 PF (Invalid in DC-INT mode.)  
 0x1F = L2 CF (Invalid in DC-INT mode.)  
 0x20 = L2 THD<sub>v</sub> (AC-INT mode only) / L2 Freq  
 (AC+DC-Sync and AC-Sync modes only)  
 0x21 = L2 THD<sub>i</sub> (AC-INT mode only)  
 0x22 = L3 V<sub>rms</sub>  
 0x23 = L3 V<sub>avg</sub>  
 0x24 = L3 V<sub>max</sub>  
 0x25 = L3 V<sub>min</sub>  
 0x26 = L3 V<sub>pkh</sub>  
 0x27 = L3 I<sub>rms</sub>  
 0x28 = L3 I<sub>avg</sub>  
 0x29 = L3 I<sub>max</sub>  
 0x2A = L3 I<sub>min</sub>  
 0x2B = L3 I<sub>pkh</sub>  
 0x2C = L3 P  
 0x2D = L3 S (Invalid in DC-INT mode.)  
 0x2E = L3 Q (Invalid in DC-INT mode.)  
 0x2F = L3 PF (Invalid in DC-INT mode.)  
 0x30 = L3 CF (Invalid in DC-INT mode.)  
 0x31 = L3 THD<sub>v</sub> (AC-INT mode only) / L3 Freq  
 (AC+DC-Sync and AC-Sync modes only)  
 0x32 = L3 THD<sub>i</sub> (AC-INT mode only)

Type	INT32
Data size	4 Bytes
Access	Get

## Attribute 0x30: [:SOURce]:SEQuence:CONDition

Description	Returns the sequence status.(Only Sequence Mode Active)
Sub-Index	-
Name	[:SOURce]:SEQuence:CONDition
Value	Current sequence status  0 = Idle mode 1 = Run mode 2 = Hold mode
Type	UINT8
Data size	1 Byte
Access	Get
Example (Data)	XX 0E 64 11 30 (:SEQ:COND?)  Return data is 1 (Run mode) XX 8E 01 00 00 00  Returns the Current sequence status as 1.

## Attribute 0x31: [:SOURce]:SEQuence:CPARAmeter

Description	Sets the common parameters for the Sequence mode. Please see the user manual for a full description of each parameter. (Only Sequence Mode Active)
Sub-Index	0x00
Name	Step Time
Value	0.0001 ~ 999.9999



	(Set Value = Value * 10000)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Sub-Index	0x01
Name	On phase
Value	0.0 ~ 359.9 (Set Value = Value * 10)
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Sub-Index	0x02
Name	On Phase fixed/ free
Value	1 = on (fixed) 0 = off (free)
Type	UINT8
Data size	1 Byte
Access	Get/Set
Sub-Index	0x03
Name	Off phase
Value	0.0 ~ 359.9 (Set Value = Value * 10)
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Sub-Index	0x04
Name	Off Phase fixed/ free
Value	1 = on (fixed)

	0 = off (free)
Type	UINT8
Data size	1 Byte
Access	Get/Set
Sub-Index	0x05
Name	Term
Value	0 = Continue 1 = End 2 = Hold
Type	UINT8
Data size	1 Byte
Access	Get/Set
Sub-Index	0x06
Name	Jump step number
Value	0 ~ 999
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Sub-Index	0x07
Name	Jump To
Value	1 = on 0 = off
Type	UINT8
Data size	1 Byte
Access	Get/Set
Sub-Index	0x08
Name	Jump Cnt
Value	0~ 9999

Type	UINT16
Data size	2 Bytes
Access	Get/Set
Sub-Index	0x09
Name	Sync Code
Value	0 = LL 1 = LH 2 = HL 3 = HH
Type	UINT8
Data size	1 Byte
Access	Get/Set
Sub-Index	0x0A
Name	Branch1
Value	0 ~ 999
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Sub-Index	0x0B
Name	Branch1 on/off
Value	1 = on 0 = off
Type	UINT8
Data size	1 Byte
Access	Get/Set
Sub-Index	0x0C
Name	Branch2
Value	0 ~ 999

Type	UINT16
Data size	2 Bytes
Access	Get/Set
Sub-Index	0x0D
Name	Branch2 on/off
Value	1 = on 0 = off
Type	UINT8
Data size	1 Byte
Access	Get/Set
Sub-Index	0x0E
Name	Reserved
Value	-
Type	UINT8
Data size	1 Byte
Access	Get/Set

**Attribute 0x32: [:SOURce]:SEQuence:CSTep**

Description	Returns the currently running step number. (Only Sequence Mode Active)
Sub-Index	-
Name	[:SOURce]:SEQuence:CSTep
Value	Current step number
Type	UINT16
Data size	2 Bytes
Access	Get
Example (Data)	XX 0E 64 11 32 (:SEQ:CSTep?)

Return data is 1  
XX 8E 01 00 00 00

Returns the Current step number as 1.

### Attribute 0x33: [:SOURce]:SEQuence:INSTrument:SElect

Description	Sets or queries the phase to set sequence. (Only Sequence Mode Active. For single-phase, only L1 can be set. For single-phase three-wire, L1 and L2 can be set. )
Sub-Index	-
Name	[:SOURce]:SEQuence:INSTrument:SElect
Value	FGetm 0 ~ 2, which represent different phase to set sequence, respectively.
	0 = L1      L1 phase 1 = L2      L2 phase 2 = L3      L3 phase
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 10 64 11 33 01 (:SEQ:INST:SEL L2)  Sets the L2 phase to set sequence.

### Attribute 0x34: [:SOURce]:SEQuence:SPARAmeter

Description	Sets or queries the parameters for a specified step. (Only Sequence Mode Active)
Sub-Index	0x00
Name	ACV setting

Value	ACV (Set Value = Value * 100)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Sub-Index	0x01
Name	ACV mode
Value	0 = Constant 1 = Keep 2 = Sweep
Type	UINT8
Data size	1 Byte
Access	Get/Set
Sub-Index	0x02
Name	DCV. Not applicable. This parameter will be ignored.
Value	DCV (Set Value = Value * 100)
Type	INT32
Data size	4 Bytes
Access	Get/Set
Sub-Index	0x03
Name	DCV mode
Value	0 = Constant 1 = Keep 2 = Sweep
Type	UINT8
Data size	1 Byte
Access	Get/Set
Sub-Index	0x04
Name	Frequency

Value	1.00 ~ 2000.0 (Set Value = Value * 100)	
Type	UINT32	
Data size	4 Bytes	
Access	Get/Set	
Sub-Index	0x05	
Name	Frequency mode	
Value	0 = Constant 1 = Keep 2 = Sweep	
Type	UINT8	
Data size	1 Byte	
Access	Get/Set	
Sub-Index	0x06	
Name	Waveform	
Value	0 = ARB1	Arbitrary wave 1
	1 = ARB2	Arbitrary wave 2
	2 = ARB3	Arbitrary wave 3
	3 = ARB4	Arbitrary wave 4
		•
		•
		•
	12 = ARB13	Arbitrary wave 13
	13 = ARB14	Arbitrary wave 14
	14 = ARB15	Arbitrary wave 15
	15 = ARB16	Arbitrary wave 16
	16 = SIN	Sin wave
	17 = SQU	Square wave
	18 = TRI	Triangle wave

19 = ARB17	Arbitrary wave 17
20 = ARB18	Arbitrary wave 18
21 = ARB19	Arbitrary wave 19
22 = ARB20	Arbitrary wave 20
	.
	.
	.
252 = ARB250	Arbitrary wave 250
253 = ARB251	Arbitrary wave 251
254 = ARB252	Arbitrary wave 252
255 = ARB253	Arbitrary wave 253

Type	UINT8
Data size	1 Byte
Access	Get/Set
Sub-Index	0x07
Name	Phase angle
Value	Phase angle. (L1 phase Fixed to 0.) 0 ~ 359.9 (Set Value = Value * 10)
Type	UINT16
Data size	2 Bytes
Access	Get/Set

**Attribute 0x35: [:SOURce]:SEQuence:STEP**

Description	Sets or queries the current step number. (Only Sequence Mode Active)
Sub-Index	-
Name	[:SOURce]:SEQuence:STEP
Value	Step number
Type	UINT16



Data size	2 Bytes
Access	Get/Set
Example (Data)	XX 10 64 11 35 01 00 (:SEQ:STEP 1)
	Sets the step number to 1.

### Attribute 0x36: [:SOURce]:SIMulation:CONDition

Description	Returns the simulation status. (Only Simulation Mode Active)
Sub-Index	-
Name	[:SOURce]:SIMulation:CONDition
Value	Current simulation status  0 = Idle mode 1 = Run mode 2 = Hold mode
Type	UINT8
Data size	1 Byte
Access	Get
Example (Data)	XX 0E 64 11 36 (:SIM:COND?)  Return data is 1 (Run mode) XX 8E 01 00 00 00
	Returns the Current simulation status as 1.

### Attribute 0x37: [:SOURce]:SIMulation:ABNormal:CODE

Description	Sets the external trigger output for the abnormal step parameter. This option is only
-------------	---

	applicable when in the Simulation mode. (Only Simulation Mode Active)
Sub-Index	-
Name	[[:SOURce]:SIMulation:ABNormal:CODE
Value	External trigger output
	0 = LL
	1 = LH
	2 = HL
	3 = HH
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 10 64 11 37 01 (SIM:ABN:CODE 1)
	Set the external trigger output to LH

**Attribute 0x38:**

**[[:SOURce]:SIMulation:ABNormal:FREQUENCY**

Description	Sets or queries the frequency of the abnormal step of the simulation mode. (Only Simulation Mode Active)
Sub-Index	-
Name	[[:SOURce]:SIMulation:ABNormal:CODE
Value	Frequency of abnormal step 1.00 ~ 2000.0 (Set Value = Value * 100)
Type	UINT32
Data size	4 Bytes
Access	Get/Set

Example (Data)	XX 10 64 11 38 7C 15 (:SIM:ABN:FREQ 55)
	Sets the frequency to 55Hz.

**Attribute 0x39:**

**[[:SOURce]:SIMulation:ABNormal:PHASe:STARt:ENABle]**

Description	Enables/Disables (Fixed/Free) the ON Phs parameter of the abnormal step for the Simulation mode. (Only Simulation Mode Active)
Sub-Index	-
Name	[[:SOURce]:SIMulation:ABNormal:PHASe:STARt:ENABle]
Value	0 = Disabled 1 = Enabled
Type	UINT8
Data size	1Byte
Access	Get/Set
Example (Data)	XX 10 64 11 39 01 (:SIM:ABN:PHAS:STAR:ENAB 1)
	Enable the ON Phs.

**Attribute 0x3A:**

**[[:SOURce]:SIMulation:ABNormal:PHASe:STARt[:IMMEDIATE]]**

Description	Sets or queries the ON Phs parameter of the abnormal step for the Simulation mode. (Only Simulation Mode Active)
Sub-Index	-
Name	[[:SOURce]:SIMulation:ABNormal:PHASe:STARt[:IMMEDIATE]]

	ART[:IMMEDIATE]
Value	ON Phs (start phase) 0° ~ 359.9° (Set Value = Value * 10)
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Example (Data)	XX 10 64 11 3A 00 00 (:SIM:ABN:PHAS:STAR 0)
	Sets ON Phs to 0.

## Attribute 0x3B

## [:SOURce]:SIMulation:ABNormal:PHASe:STOP:ENABle

Description	Enables/Disables (Fixed/Free) the OFF Phs parameter of the abnormal step for the Simulation mode. (Only Simulation Mode Active)
Sub-Index	-
Name	[:SOURce]:SIMulation:ABNormal:PHASe:STOP:ENABle
Value	0 = Disabled 1 = Enabled
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 10 64 11 3B 01 (:SIM:ABN:PHAS:STOP:ENAB 1)
	Enable the OFF Phs.

## Attribute 0x3C:

## [:SOURce]:SIMulation:ABNormal:PHASe:STOP[:IMMEDIATE]

Description	Sets or queries the OFF Phs parameter of the abnormal step for the Simulation mode. (Only Simulation Mode Active)
Sub-Index	-
Name	[:SOURce]:SIMulation:ABNormal:PHASe:STOP[:IMMEDIATE]
Value	ON Phs (Stop phase) 0° ~ 359.9°

(Set Value = Value \* 10)

Type	UINT16
Data size	2 Bytes
Access	Get/Set
Note	Sets the off phase of the waveform after the output has been turned off.
Example (Data)	XX 10 64 11 3C 00 00 (:SIM:ABN:PHAS:STOP 0)
	Sets OFF Phs to 0.

**Attribute 0x3D: [:SOURce]:SIMulation:ABNormal:TIME**

Description	Sets or queries the Time parameter of the abnormal step for the Simulation mode. (Only Simulation Mode Active)
Sub-Index	-
Name	[:SOURce]:SIMulation:ABNormal:TIME
Value	Time of the abnormal step in seconds 0.0001 ~ 999.9999 s (Set Value = Value * 10000)
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Example (Data)	XX 10 64 11 3D 10 27 (:SIM:ABN:TIME 1)
	Sets the abnormal step time to 1 second.

## Attribute 0x3E: [:SOURce]:SIMulation:ABNormal:VOLTage

Description	Sets or queries the Vset parameter of the abnormal step for the Simulation mode. (Only Simulation Mode Active)
Sub-Index	-
Name	[:SOURce]:SIMulation:ABNormal:VOLTage
Value	Voltage of the abnormal step. (Set Value = Value * 100)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Example (Data)	XX 10 64 11 3E 10 27 (:SIM:ABN:VOLT 100)  Sets the abnormal step voltage to the 100.

## Attribute 0x3F: [:SOURce]:SIMulation:CSTep

Description	Returns the currently running step. (Only Simulation Mode Active)
Sub-Index	-
Name	[:SOURce]:SIMulation:CSTep
Value	Current step  0 = Initial step 1 = Normal1 step 2 = Transition1 step 3 = Abnormal step 4 = Transition2 step 5 = Normal2 step

Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 0E 64 11 3F (:SIM:CStep?)  Return data is 1 (Normal1 step) XX 8E 01 00 00 00  Returns the Current step as Normal1.

#### Attribute 0x40: [:SOURce]:SIMulation:INITial:CODE

Description	Sets the external trigger output for the initial step parameter. This option is only applicable when in the Simulation mode. (Only Simulation Mode Active)
Sub-Index	-
Name	[:SOURce]:SIMulation:INITial:CODE
Value	0 = LL 1 = LH 2 = HL 3 = HH
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 10 64 11 40 01 (:SIM:INIT:CODE 1)  Sets the initial step code to the LH.



## Attribute 0x41: [:SOURce]:SIMulation:INITial:FREQuency

Description	Sets the external trigger output for the initial step parameter. This option is only applicable when in the Simulation mode. (Only Simulation Mode Active)
Sub-Index	-
Name	[:SOURce]:SIMulation:INITial:FREQuency
Value	Frequency of initial step 1.00 ~ 2000.0 (Set Value = Value * 100)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Example (Data)	XX 10 64 11 41 70 10 (:SIM:INIT:FREQ 60)  Sets the frequency of the initial step to 60Hz.

## Attribute 0x42:

## [:SOURce]:SIMulation:INITial:PHASe:STARt:ENABle

Description	Enables/Disables (Fixed/Free) the ON Phs parameter of the initial step for the Simulation mode. (Only Simulation Mode Active)
Sub-Index	-
Name	[:SOURce]:SIMulation:INITial:PHASe:STARt:ENABle
Value	0 = Disabled 1 = Enabled
Type	UINT8

Data size	1 Byte
Access	Get/Set
Example (Data)	XX 10 64 11 42 01 (:SIM:INIT:PHAS:STAR:ENAB 1)
Enable the ON Phs.	

**Attribute 0x43:**

**[[:SOURce]:SIMulation:INITial:PHASe:STARt[:IMMEDIATE]]**

Description	Sets or queries the ON Phs parameter of the initial step for the Simulation mode. (Only Simulation Mode Active)
Sub-Index	-
Name	[[:SOURce]:SIMulation:INITial:PHASe:STARt[:IMMEDIATE]]
Value	ON Phs (start phase) 0° ~ 359.9° (Set Value = Value * 10)
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Note	Sets the off phase of the waveform after the output has been turned off.
Example (Data)	XX 10 64 11 43 00 00 (:SIM:INIT:PHAS:STAR 0)
Sets ON Phs to 0.	

## Attribute 0x44:

## [:SOURce]:SIMulation:INITial:PHASe:STOP:ENABLE

Description	Enables/Disables (Fixed/Free) the OFF Phs parameter of the initial step for the Simulation mode. (Only Simulation Mode Active)
Sub-Index	-
Name	[:SOURce]:SIMulation:INITial:PHASe:STOP:ENABLE
Value	0 = Disabled 1 = Enabled
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 10 64 11 44 01 (:SIM:INIT:PHAS:STOP:ENAB 1)
	Enable the OFF Phs.

## Attribute 0x45:

## [:SOURce]:SIMulation:INITial:PHASe:STOP[:IMMEDIATE]

Description	Sets or queries the OFF Phs parameter of the initial step for the Simulation mode. (Only Simulation Mode Active)
Sub-Index	-
Name	[:SOURce]:SIMulation:INITial:PHASe:START[:IMMEDIATE]
Value	OFF Phs (Stop phase) 0° ~ 359.9°

	(Set Value = Value * 10)
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Note	Sets the off phase of the waveform after the output has been turned off.
Example (Data)	XX 10 64 11 45 00 00 (:SIM:INIT:PHAS:STOP 0)
	Sets OFF Phs to 0.

**Attribute 0x46: [:SOURce]:SIMulation:INITial:VOLTage**

Description	Sets or queries the Vset parameter of the initial step for the Simulation mode. (Only Simulation Mode Active)
Sub-Index	-
Name	[:SOURce]:SIMulation:INITial:VOLTage
Value	Voltage of the initial step. (Set Value = Value * 100)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Example (Data)	XX 10 64 11 46 10 27 (:SIM:INIT:VOLT 100)
	Sets the initial step voltage to the 100.

## Attribute 0x48: [:SOURce]:SIMulation:NORMal&lt;1|2&gt;:CODE

Description	Sets the external trigger output for the normal 1 or normal 2 step parameter. This option is only applicable when in the Simulation mode. (Only Simulation Mode Active)
Sub-Index	0x00
Name	Normal 1 Code
Value	0 = LL 1 = LH 2 = HL 3 = HH
Type	UINT8
Data size	1 Byte
Access	Get/Set
Sub-Index	0x01
Name	Normal 2 Code
Value	0 = LL 1 = LH 2 = HL 3 = HH
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 10 64 11 48 00 01 (:SIM:NORM1:CODE 1)  Sets the Normal 1 Code to the LH.

## Attribute 0x49: [:SOURce]:SIMulation:NORMAL

## 1:FREQUENCY

Description	Sets or queries the frequency of the normal1 step of the simulation mode. (Only Simulation Mode Active)
Sub-Index	0x00
Name	Normal 1 Frequency
Value	Frequency of abnormal step 1.00 ~ 2000.0 (Set Value = Value * 100)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Sub-Index	0x01
Name	Normal 2 Frequency
Value	Frequency of abnormal step 1.00 ~ 2000.0 (Set Value = Value * 100)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Example (Data)	XX 10 64 11 49 00 70 17 (:SIM:NORM1:FREQ 60)
	Sets the Normal 1 Frequency to 60Hz.

## Attribute 0x4A:

[:SOURce]:SIMulation:NORMal&lt;1|2&gt;:PHASe:STARt:ENABle

Description	Enables/Disables (Fixed/Free) the ON Phs parameter of the normal1 or normal2 step for the Simulation mode. (Only Simulation Mode Active)
Sub-Index	0x00
Name	Normal 1
Value	0 = Disabled 1 = Enabled
Type	UINT8
Data size	1 Byte
Access	Get/Set
Sub-Index	0x01
Name	Normal 2
Value	0 = Disabled 1 = Enabled
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 10 64 11 4A 00 01 (:SIM:NORM1:PHAS:STAR:ENAB 1)
	Enable the ON Phs.

Attribute 0x4B:

[:SOURce]:SIMulation:NORMal<1|2>:PHASe:STARt[:IMMediate]

Description	Sets or queries the ON Phs parameter of the normal1 or normal2 step for the Simulation mode. (Only Simulation Mode Active)
Sub-Index	0x00
Name	Normal 1 ON Phs (start phase)
Value	0° ~ 359.9° (Set Value = Value * 10)
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Sub-Index	0x01
Name	Normal 2 ON Phs (start phase)
Value	0° ~ 359.9° (Set Value = Value * 10)
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Example (Data)	XX 10 64 11 4B 00 00 00 (:SIM:NORM1:PHAS:STAR 0)
	Sets ON Phs to 0.



## Attribute 0x4C:

[:SOURce]:SIMulation:NORMal&lt;1|2&gt;:PHASe:STOP:ENABLE

Description	Enables/Disables (Fixed/Free) the OFF Phs parameter of the normal1 or normal2 step for the Simulation mode. (Only Simulation Mode Active)
Sub-Index	0x00
Name	Normal 1
Value	0 = Disabled 1 = Enabled
Type	UINT8
Data size	1 Byte
Access	Get/Set
Sub-Index	0x01
Name	Normal 2
Value	0 = Disabled 1 = Enabled
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 10 64 11 4C 00 01 (:SIM:NORM1:PHAS:STOP:ENAB 1)
	Enable the OFF Phs.

Attribute 0x4D:

[[:SOURce]:SIMulation:NORMal<1|2>:PHASe:STOP[:IMMediate]

Description	Sets or queries the OFF Phs parameter of the normal1 or normal2 step for the Simulation mode. (Only Simulation Mode Active)
Note	Sets the off phase of the waveform after the output has been turned off.
Sub-Index	0x00
Name	Normal 1 OFF Phs (Stop phase)
Value	0° ~ 359.9° (Set Value = Value * 10)
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Sub-Index	0x01
Name	Normal 2 OFF Phs (Stop phase)
Value	0° ~ 359.9° (Set Value = Value * 10)
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Example (Data)	XX 10 64 11 4D 00 00 00 (:SIM:NORM1:PHAS:STOP 0)
	Sets OFF Phs to 0.

## Attribute 0x4E: [:SOURce]:SIMulation:NORMal&lt;1|2&gt;:TIME

Description	Sets or queries the Time parameter of the normal1 or normal2 step for the Simulation mode. (Only Simulation Mode Active)
Sub-Index	0x00
Name	Normal 1 Time of the step in seconds
Value	0.0001 ~ 999.9999 s (Set Value = Value * 10000)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Sub-Index	0x01
Name	Normal 2 Time of the step in seconds
Value	0.0001 ~ 999.9999 s (Set Value = Value * 10000)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Example (Data)	XX 10 64 11 4E 00 10 27 (:SIM:NORM1:TIME 1)
	Sets the step time to 1 second.

## Attribute 0x4F: [:SOURce]:SIMulation:NORMal 1:VOLTage

Description	Sets or queries the Vset parameter of the normal1 step for the Simulation mode. (Only Simulation Mode Active)
Sub-Index	0x00
Name	Normal 1
Value	Voltage of the abnormal step. (Set Value = Value * 100)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Sub-Index	0x01
Name	Normal 2
Value	Voltage of the abnormal step. (Set Value = Value * 100)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Example (Data)	XX 10 64 11 4F 00 10 27 (:SIM:NORM1:VOLT 100)
	Sets the normal1step voltage to the 100.

## Attribute 0x50: [:SOURce]:SIMulation:REPeat:COUNT

Description	Sets or queries the repeat count for the Simulation mode. (Only Simulation Mode Active)
Sub-Index	-
Name	[:SOURce]:SIMulation:REPeat:COUNT
Value	0 ~ 9999 (0 = infinite loop)
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Example (Data)	XX 10 64 11 50 01 (:SIM:REP:COUN 1)  Sets the repeat count to 1.

## Attribute 0x51: [:SOURce]:SIMulation:REPeat:ENABLE

Description	Turns the repeat function on or off for the Simulation mode. (Only Simulation Mode Active)
Sub-Index	-
Name	[:SOURce]:SIMulation:REPeat:ENABLE
Value	0 = Disabled 1 = Enabled
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 10 64 11 51 01 (:SIM:REP:ENAB 1)  Enables the repeat function.

Attribute 0x53:

[:SOURce]:SIMulation:TRANSition&lt;1|2&gt;:TIME

Description	Sets or queries the Time parameter of the transition step for the Simulation mode. (Only Simulation Mode Active)
Sub-Index	0x00
Name	Transition 1 Time of the step in seconds
Value	0.0001 ~ 999.9999 s (Set Value = Value * 10000)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Sub-Index	0x01
Name	Transition 2 Time of the step in seconds
Value	0.0001 ~ 999.9999 s (Set Value = Value * 10000)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Example (Data)	XX 10 64 11 53 00 10 27 (:SIM:TRAN1:TIME 1)
	Sets the step time to 1 second.

Attribute 0x52:

[[:SOURce]:SIMulation:TRANSition<1|2>:CODE

Description	Sets the external trigger output for the transition step parameter. This option is only applicable when in the Simulation mode. (Only Simulation Mode Active)
Sub-Index	0x00
Name	Transition 1 Code
Value	0 = LL 1 = LH 2 = HL 3 = HH
Type	UINT8
Data size	1 Byte
Access	Get/Set
Sub-Index	0x01
Name	Transition 2 Code
Value	0 = LL 1 = LH 2 = HL 3 = HH
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 10 64 11 52 00 01 (:SIM:TRAN1:CODE 1)  Set the transition 1 code to LH.

**Attribute 0x60: [:SOURce]:SEQuence:NSParameter**

Description	Sets or queries the parameters for a specified step.(Only Sequence Mode Active)
Sub-Index	0x00 (ACV setting)
Value	ACV (Set Value = Value * 100)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Sub-Index	0x01 (ACV mode)
Value	0 = Constant 1 = Keep 2 = Sweep
Type	UINT8
Data size	1 Byte
Access	Get/Set
Sub-Index	0x02 (DCV. Not applicable. This parameter will be ignored.)
Value	DCV (Set Value = Value * 100)
Type	INT32
Data size	4 Bytes
Access	Get/Set
Sub-Index	0x03 (DCV mode)
Value	0 = Constant 1 = Keep 2 = Sweep



Type	UINT8	
Data size	1 Byte	
Access	Get/Set	
Sub-Index	0x04 (Frequency)	
Value	1.00 ~ 2000.0 (Set Value = Value * 100)	
Type	UINT32	
Data size	4 Bytes	
Access	Get/Set	
Sub-Index	0x05 (Frequency mode)	
Value	0 = Constant 1 = Keep 2 = Sweep	
Type	UINT8	
Data size	1 Byte	
Access	Get/Set	
Sub-Index	0x06 (Waveform)	
Value	0 = ARB1	Arbitrary wave 1
	1 = ARB2	Arbitrary wave 2
	2 = ARB3	Arbitrary wave 3
	3 = ARB4	Arbitrary wave 4
		•
		•
		•
	12 = ARB13	Arbitrary wave 13
	13 = ARB14	Arbitrary wave 14
	14 = ARB15	Arbitrary wave 15
	15 = ARB16	Arbitrary wave 16

16 = SIN	Sin wave
17 = SQU	Square wave
18 = TRI	Triangle wave
19 = ARB17	Arbitrary wave 17
20 = ARB18	Arbitrary wave 18
21 = ARB19	Arbitrary wave 19
22 = ARB20	Arbitrary wave 20
	.
	.
	.
252 = ARB250	Arbitrary wave 250
253 = ARB251	Arbitrary wave 251
254 = ARB252	Arbitrary wave 252
255 = ARB253	Arbitrary wave 253

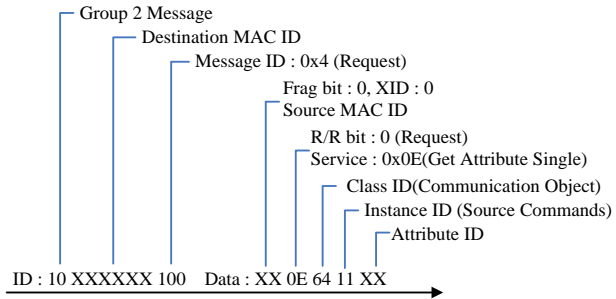
Type	UINT8
Data size	1 Byte
Access	Get/Set
Sub-Index	0x07 (Duty)
Value	Square wave signal duty cycle. (Set Value = Value * 10)
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Sub-Index	0x08 (Phase angle)
Value	Phase angle. (L1 phase Fixed to 0.) 0 ~ 359.9 (Set Value = Value * 10)
Type	UINT16
Data size	2 Bytes

Access

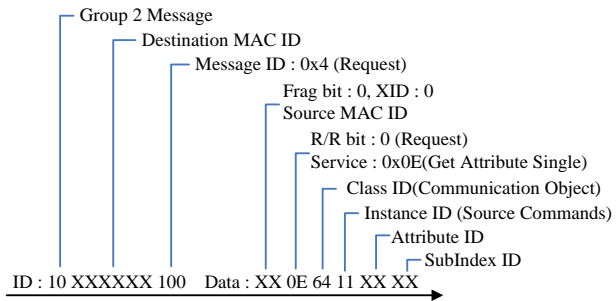
Get/Set

### Request Source Commands Example

Request Example no including SubIndex



Request Example including SubIndex



## Source1 Commands (Instance 0x12)

---

Attribute 0x01: [:SOURce]:SQUare:DCYCLE .....	293
Attribute 0x02: [:SOURce]:VOLTage:LIMit:PEAK .....	293
Attribute 0x03: [:SOURce]:VOLTage:LIMit:RMS.....	294
Attribute 0x04: [:SOURce]:VOLTage:LIMit:HIGH.....	295
Attribute 0x05: [:SOURce]:VOLTage:LIMit:LOW .....	295
Attribute 0x06: [:SOURce]:VOLTage:RANGe .....	296
Attribute 0x07: [:SOURce]:VOLTage:RESPonse.....	297
Attribute 0x08: [:SOURce]:VOLTage[:LEVel][:IMMEDIATE][:AMPLitude] .....	298
Attribute 0x09: [:SOURce]:VOLTage[:LEVel][:IMMEDIATE]:OFFSet ...	299
Attribute 0x0A: [:SOURce]:MODE.....	300
Request Source1 Commands Example .....	301

## Attribute 0x01: [:SOURce]:SQUare:DCYCLE

Description	Sets or queries the square wave signal duty cycle. The settable range depends on the frequency. Not available for DC-INT, AC+DC-EXT and ACEXT)
Sub-Index	-
Name	[:SOURce]:SQUare:DCYCLE
Value	Square wave signal duty cycle. (Set Value = Value * 10)
Type	UINT16
Data size	2 Bytes
Access	Get/Set
Example (Data)	XX 0E 64 12 01 (:SQU:DCYC?)  Return data is 500 (50.0) XX 8E F4 01 00 00  Returns the square wave signal duty cycle as 50.0%.

## Attribute 0x02: [:SOURce]:VOLTage:LIMit:PEAK

Description	Sets or Queries the Value of Vpp in Specific Mode(AC-INT or AC-ADD or AC-Sync) and Specific Wave Shape(TRI or ARB) and Specific V Unit(p-p)
Sub-Index	-
Name	[:SOURce]:VOLTage:LIMit:PEAK
Value	Vpp (Set Value = Value * 100)

Type	UINT32
Data size	4 Bytes
Access	Get/Set
Example (Data)	XX 0E 64 12 02 (VOLT:LIM:PEAK?)  Return data is 50000 (500.00) XX 8E 50 C3 00 00  Returns the Vpp limit.

**Attribute 0x03: [:SOURce]:VOLTage:LIMit:RMS**

Description	Sets or queries the voltage limit for the continuous operation mode. (Only AC-INT or AC-ADD or AC-Sync Active)
Sub-Index	-
Name	[:SOURce]:VOLTage:LIMit:RMS
Value	Vrms. (Set Value = Value * 100)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Example (Data)	XX 0E 64 12 03 (VOLT:LIM:RMS?)  Return data is 35000 (350.00) XX 8E B8 88 00 00  Returns the Vrms limit.

## Attribute 0x04: [:SOURce]:VOLTage:LIMit:HIGH

Description	Sets or queries the voltage high limit. (Only AC+DC-INT or DC-INT or AC+DC-ADD or AC+DC-Sync Active)
Sub-Index	-
Name	[:SOURce]:VOLTage:LIMit:HIGH
Value	Voltage high limit (Set Value = Value * 100)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Example (Data)	XX 0E 64 12 04 (VOLT:LIM:HIGH?)  Return data is 50000 (500.00) XX 8E 50 C3 00 00  Returns the voltage high limit.

## Attribute 0x05: [:SOURce]:VOLTage:LIMit:LOW

Description	Sets or queries the voltage low limit. (Only AC+DC-INT or DC-INT or AC+DC-ADD or AC+DC-Sync Active)
Sub-Index	-
Name	[:SOURce]:VOLTage:LIMit:LOW
Value	Voltage low limit (Set Value = Value * 100)
Type	INT32
Data size	4 Bytes

Access	Get/Set
Example (Data)	XX 0E 64 12 05 (VOLT:LIM:LOW?)  Return data is -50000 (-500.00) XX 8E B0 3C FF FF  Returns the voltage low limit.

**Attribute 0x06: [:SOURce]:VOLTage:RANGe**

Description	Sets or queries the voltage range.
Sub-Index	-
Name	[:SOURce]:VOLTage:RANGe
Value	From 0 - 2, which represent different voltage ranges, respectively.  0 = 100V 1 = 200V 2 = AUTO (Only AC+DC-INT or AC-INT or DC-INT or AC+DC-sync or AC-sync Active)
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 0E 64 12 06 (:SOUR:VOLT:RANG?)  Return data is 1 (200V) XX 8E 01  Returns the voltage range as 200V.



## Attribute 0x07: [:SOURce]:VOLTage:RESPonse

Description	Sets or queries the voltage response. (Fast not available for single-phase or output impedance set to on.)
Sub-Index	-
Name	[:SOURce]:VOLTage:RESPonse
Value	From 0 – 2, which represent different voltage response ,respectively.  0 = SLOW      Voltage response slow. 1 = MEDIUM    Voltage response medium. 2 = FAST        Voltage response fast.
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 0E 64 12 07 (:VOLT:RESP?)  Return data is 1 (Medium) XX 8E 01  Returns the voltage response as medium.

Attribute 0x08:

`[:SOURce]:VOLTage[:LEVel][:IMMediate][:AMPLitude]`

Description	Sets or queries the RMS voltage for the continuous operation mode. (Not available for DC-INT, AC+DC-EXT, AC-EXT and AC-VCA)
Sub-Index	-
Name	<code>[:SOURce]:VOLTage[:LEVel][:IMMediate][:AMPLitude]</code>
Value	Vrms. (Set Value = Value * 100)
Type	UINT32
Data size	4 Bytes
Access	Get/Set
Example (Data)	XX 10 64 12 08 98 3A (:VOLT 150.0)  Sets the voltage to 150.0 ACV.

Attribute 0x09:

[[:SOURce]:VOLTage[:LEVel]][:IMMediate]:OFFSet

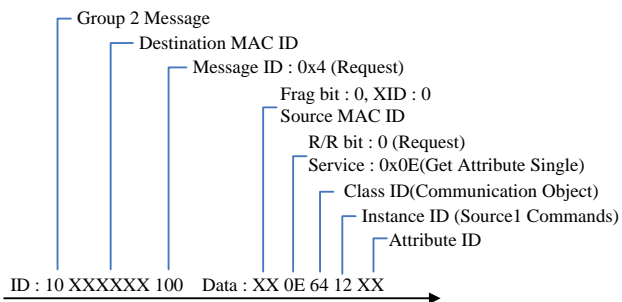
Description	Sets or queries the voltage offset value. (Only AC+DC-INT or DC-INT or AC+DC-ADD or AC+DC-Sync Active)
Sub-Index	-
Name	[[:SOURce]:VOLTage[:LEVel]][:IMMediate]:OFFSet
Value	Voltage offset value (Set Value = Value * 100)
Type	INT32
Data size	4 Bytes
Access	Get/Set
Example (Data)	XX 0E 64 12 09 (:VOLT:OFFS?)  Return data is 15000 (150.00) XX 8E 98 3A 00 00  Returns the voltage offset value as 150.0.

**Attribute 0x0A: [:SOURce]:MODE**

Description	Sets or queries the output mode of power supply.
Sub-Index	-
Name	[:SOURce]:MODE
Value	From 0 - 9, which represent different output modes, respectively.  0 = AC+DC-INT 1 = AC-INT 2 = DC-INT 3 = AC+DC-EXT 4 = AC-EXT 5 = AC+DC-ADD 6 = AC-ADD 7 = AC+DC-SYNC 8 = AC-SYNC 9 = AC-VCA
Type	UINT8
Data size	1 Byte
Access	Get/Set
Example (Data)	XX 0E 64 12 0A (MODE?)  Return data is 0 (ACDC-INT) XX 8E 00  Returns the output mode as AC+DC-INT.

## Request Source1 Commands Example

Request  
Source 1  
Commands  
Example



## Read Data Range Commands (Instance 0x17)

Attribute 0x01: DATA_RANGE_MAX .....	302
Attribute 0x02: DATA_RANGE_MIN .....	306
Request Read Data Range Commands Example .....	310

### Attribute 0x01: DATA\_RANGE\_MAX

Description	Read the maximum settable value in normal mode.
Sub-Index	0x00
Name	ACV
Value	Set Value = Value * 100
Type	UINT32
Data size	4 Bytes
Access	Get
Sub-Index	0x01
Name	DCV
Value	Set Value = Value * 100
Type	INT32
Data size	4 Bytes
Access	Get
Sub-Index	0x02
Name	Freq
Value	Set Value = Value * 100
Type	UINT32
Data size	4 Bytes
Access	Get
Sub-Index	0x03

Name	Irms
Value	Set Value = Value * 100
Type	UINT32
Data size	4 Bytes
Access	Get
Sub-Index	0x04
Name	On phase
Value	Set Value = Value * 10
Type	UINT16
Data size	2 Bytes
Access	Get
Sub-Index	0x05
Name	Off phase
Value	Set Value = Value * 10
Type	UINT32
Data size	2 Bytes
Access	Get
Sub-Index	0x06
Name	Balance L12
Value	Set Value = Value * 10
Type	UINT32
Data size	2 Bytes
Access	Get
Sub-Index	0x07
Name	Balance L13
Value	Set Value = Value * 10
Type	UINT16

Data size	2 Bytes
Access	Get
Sub-Index	0x08
Name	GAIN
Value	Set Value = Value * 10
Type	UINT16
Data size	2 Bytes
Access	Get
Sub-Index	0x09
Name	Sync Phase
Value	Set Value = Value * 10
Type	UINT32
Data size	2 Bytes
Access	Get
Sub-Index	0x0A
Name	Vpk limit High
Value	Set Value = Value * 100
Type	UINT32
Data size	4 Bytes
Access	Get
Sub-Index	0x0B
Name	Vpk limit Low
Value	Set Value = Value * 100
Type	INT32
Data size	4 Bytes
Access	Get
Sub-Index	0x0C



Name	Ipk limit High
Value	Set Value = Value * 100
Type	UINT32
Data size	4 Bytes
Access	Get
Sub-Index	0x0D
Name	Ipk limit Low
Value	Set Value = Value * 100
Type	INT32
Data size	4 Bytes
Access	Get
Sub-Index	0x0E
Name	Frequency limit High
Value	Set Value = Value * 100
Type	UINT32
Data size	4 Bytes
Access	Get
Sub-Index	0x0F
Name	Frequency limit Low
Value	Set Value = Value * 100
Type	UINT32
Data size	4 Bytes
Access	Get
Example (Data)	XX 0E 64 13 01 00 Return data is 17500 (175.00) XX 8E 5C 44 00 00 Returns the ACV maximum settable value as 175.00.

Attribute 0x02: DATA\_RANGE\_MIN

Description	Read the minimum settable value in normal mode.
Sub-Index	0x00
Name	ACV
Value	Set Value = Value * 100
Type	UINT32
Data size	4 Bytes
Access	Get
Sub-Index	0x01
Name	DCV
Value	Set Value = Value * 100
Type	INT32
Data size	4 Bytes
Access	Get
Sub-Index	0x02
Name	Freq
Value	Set Value = Value * 100
Type	UINT32
Data size	4 Bytes
Access	Get
Sub-Index	0x03
Name	Irms
Value	Set Value = Value * 100
Type	UINT32
Data size	4 Bytes

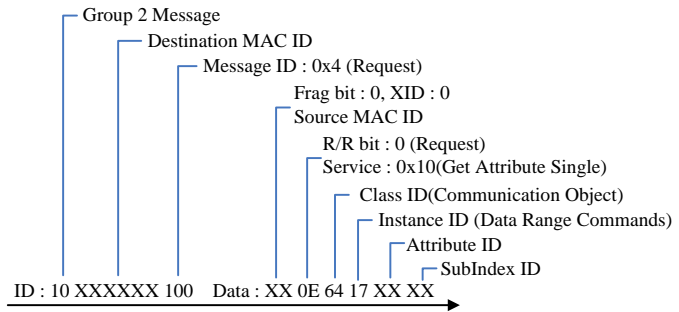
Access	Get
Sub-Index	0x04
Name	On phase
Value	Set Value = Value * 10
Type	UINT16
Data size	2 Bytes
Access	Get
Sub-Index	0x05
Name	Off phase
Value	Set Value = Value * 10
Type	UINT32
Data size	2 Bytes
Access	Get
Sub-Index	0x06
Name	Balance L12
Value	Set Value = Value * 10
Type	UINT32
Data size	2 Bytes
Access	Get
Sub-Index	0x07
Name	Balance L13
Value	Set Value = Value * 10
Type	UINT16
Data size	2 Bytes
Access	Get
Sub-Index	0x08
Name	GAIN

Value	Set Value = Value * 10
Type	UINT16
Data size	2 Bytes
Access	Get
Sub-Index	0x09
Name	Sync Phase
Value	Set Value = Value * 10
Type	UINT32
Data size	2 Bytes
Access	Get
Sub-Index	0x0A
Name	Vpk limit High
Value	Set Value = Value * 100
Type	UINT32
Data size	4 Bytes
Access	Get
Sub-Index	0x0B
Name	Vpk limit Low
Value	Set Value = Value * 100
Type	INT32
Data size	4 Bytes
Access	Get
Sub-Index	0x0C
Name	Ipk limit High
Value	Set Value = Value * 100
Type	UINT32
Data size	4 Bytes

Access	Get
Sub-Index	0x0D
Name	Ipk limit Low
Value	Set Value = Value * 100
Type	INT32
Data size	4 Bytes
Access	Get
Sub-Index	0x0E
Name	Frequency limit High
Value	Set Value = Value * 100
Type	UINT32
Data size	4 Bytes
Access	Get
Sub-Index	0x0F
Name	Frequency limit Low
Value	Set Value = Value * 100
Type	UINT32
Data size	4 Bytes
Access	Get
Example	XX 0E 64 13 01 00
(Data)	Return data is 0 (0)
	XX 8E 00 00 00 00
	Returns the ACV minimum settable value as 0.

## Request Read Data Range Commands Example

Request  
Read Data  
Range  
Commands  
Example

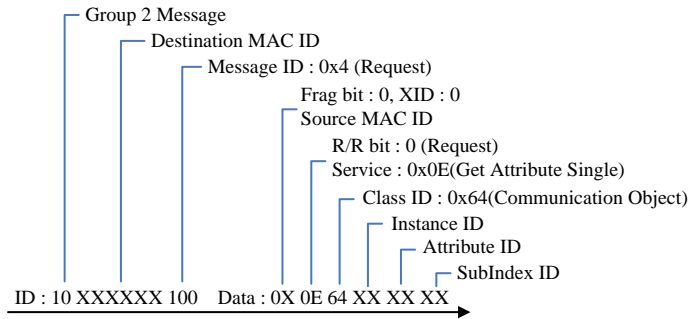


## Manufacturer-specific Command Example

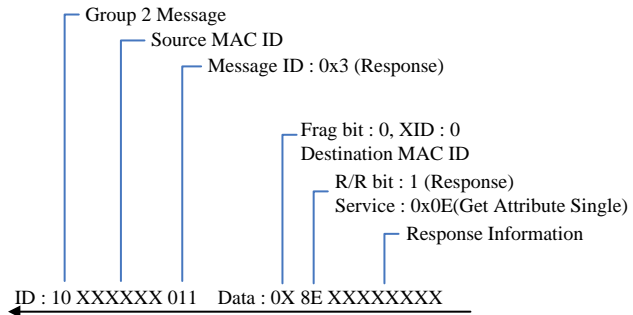
CAN Bus command example for SCPI application object request and response.

### For Get information include subindex

#### Request

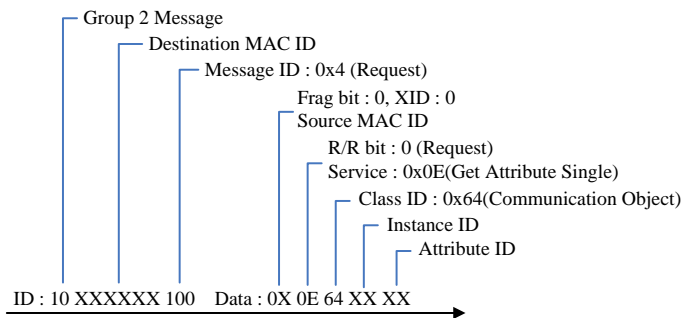


#### Response

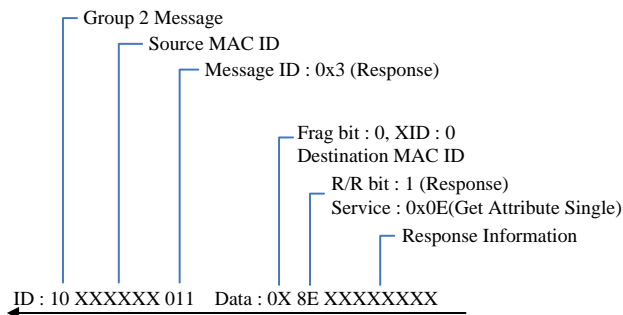


**For Get information non include subindex**

**Request**



**Response**

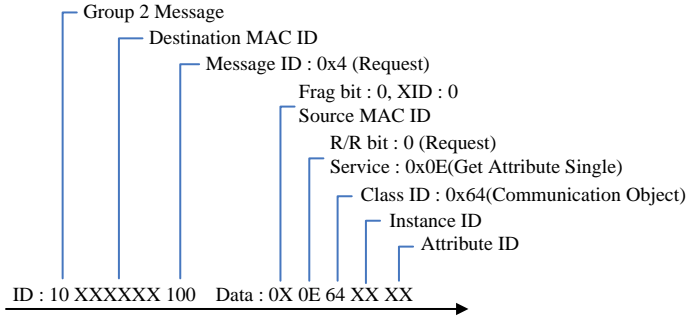




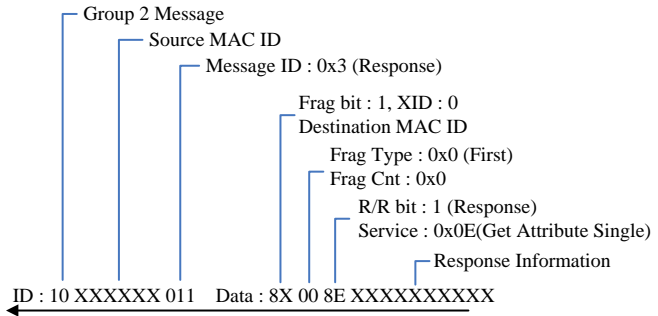
**For Get information and response information data more than 6-Bytes (For Example : Response 16-Bytes)**

**First Fragment response 5-Bytes**

Request

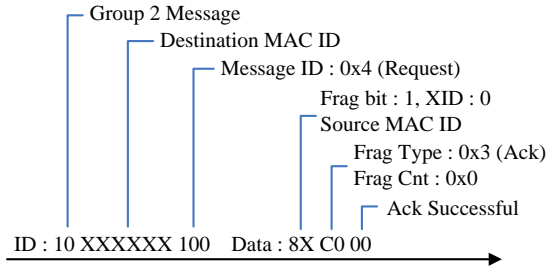


Response

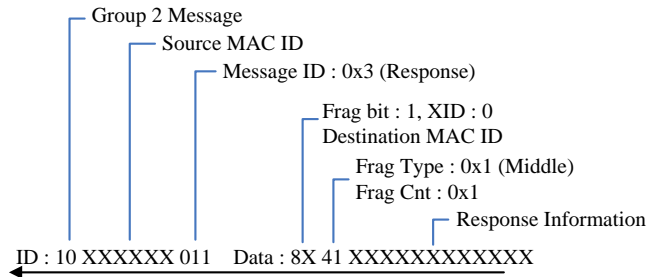


**Middle Fragment response 6-Bytes**

Request

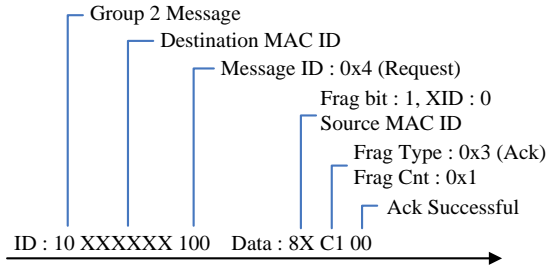


Response

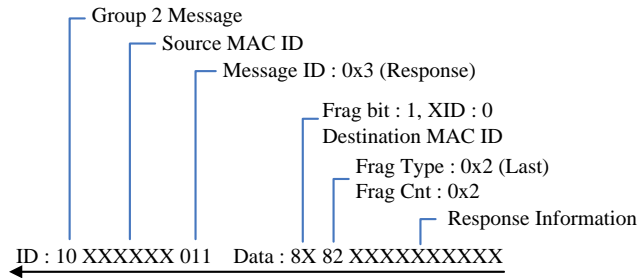


**Last Fragment response 5-Bytes and successful acknowledgement**

Request

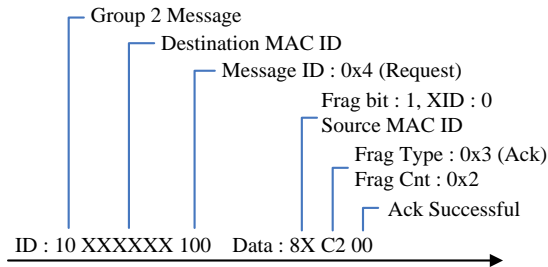


Response



Response

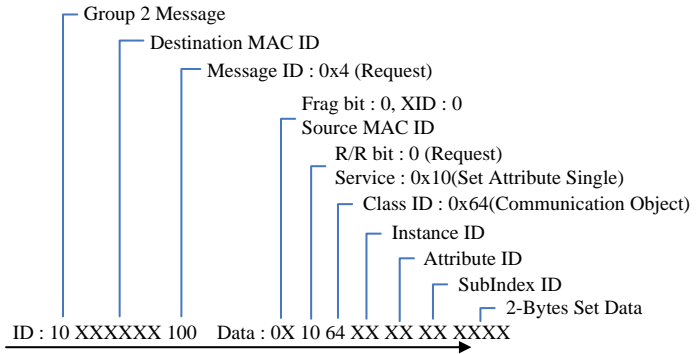
\*1



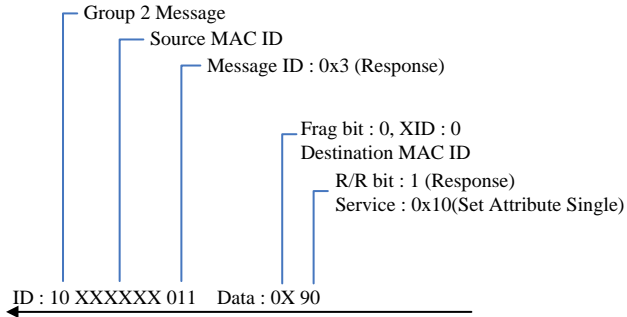
\*1 When master got response last fragmentation data then sent ack successful to slave.

**For Set information include subindex  
(For Example : Request 2-Bytes)**

Request

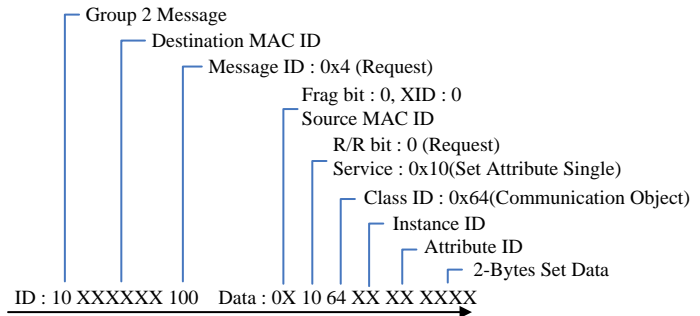


Response

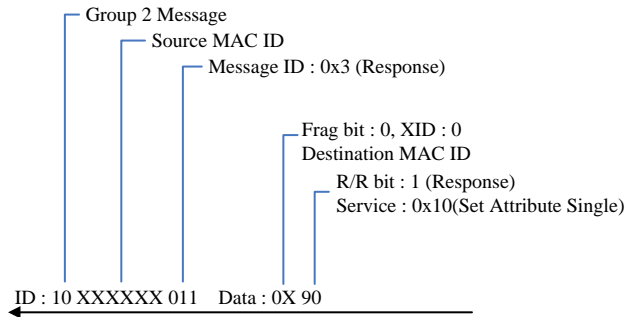


**For Set information non include subindex  
(For Example : Request 2-Bytes)**

Request



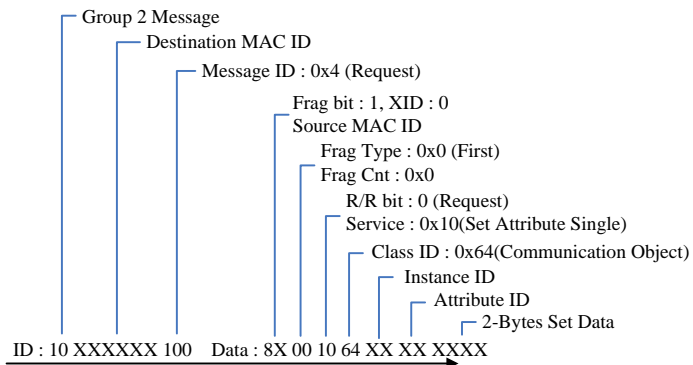
Response



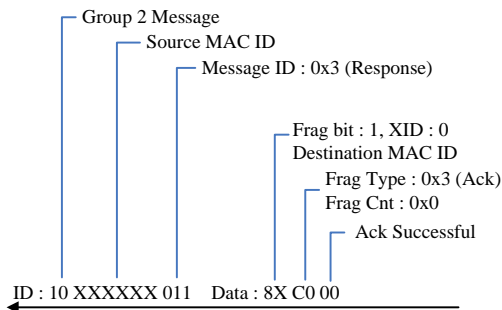
**For Set information include subindex more than 2-Bytes or non include subindex more than 3-Bytes (For Example : Set 4-Bytes data)**

**First Fragment request 2-Bytes**

Request

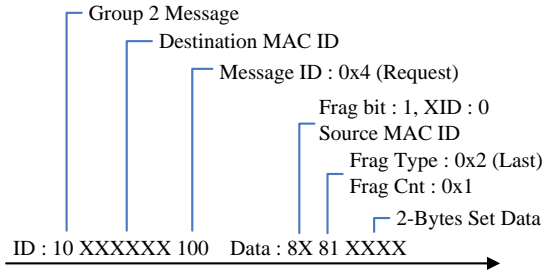


Response



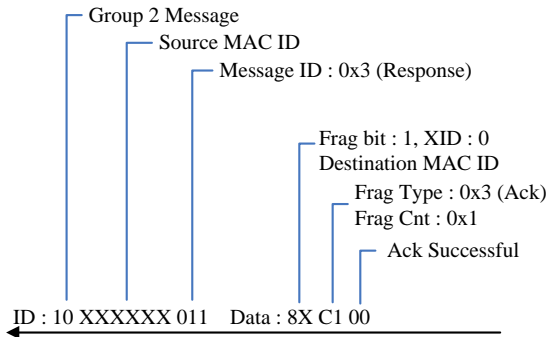
**Last Fragment request 2-Bytes and successful acknowledgement**

Request



Response

\*1



Note

\*1 When slave got request last fragmentation form master then response ack successful to master

## DeviceNet Errors Code

Define the DeviceNet response error code standard setting.

Error Condition	General Error Code (Hex)	Additional Error Condition	Additional Error Code (Hex)
Resource Unavailable	0x02	Not Support Mode	0x01
		Data Out of Range	0x02
		Data Transmit Fail	0x03
		Subindex no Find	0x04
Service Code not Support	0x08	None	0Xff or Null
Attribute Invalid	0x09	None	0Xff or Null
Already in Requested mode/state	0x0B	Invalid Allocation Choice	0x02
		None	0Xff or Null
Object State Conflict	0x0C	Class Specific Error	0x01
		None	0Xff or Null
Attribute not Setting	0x0E	None	0Xff or Null
Attribute not Support	0x14	None	0Xff or Null
Object does not exist	0x16	None	0Xff or Null



# APPENDIX

## Factory Default Settings

The following default settings are the factory configuration settings for the ASR-6000 series. For details on how to return to the factory default settings, see the User's Manual.

Continuous Mode	ASR-6450		ASR-6600	
	3P4W	1P2W	3P4W	1P2W
MODE	AC+DC-INT		AC+DC-INT	
Range	100V		100V	
ACV	0.00 Vrms		0.00 Vrms	
DCV	+0.00 Vdc		+0.00 Vdc	
FREQ	50.00Hz		50.00 Hz	
IRMS	15.75 A	47.25 A	21 A	63 A
ON Phs	Fixed 0.0°		Fixed 0.0°	
OFF Phs	Fixed 0.0°		Fixed 0.0°	
GAIN	100		100	
SIG	L1 LINE		L1 LINE	
Syc Phs	0.0		0.0	
SRC	L1 EXT		L1 EXT	
Wave	SIN		SIN	
Freq Limit	2000		2000	
Vrms Limit	175.0 Vrms		175.0 Vrms	
VPK+ Limit	+250 V		+250 V	
VPK- Limit	-250 V		-250 V	
IPK+ Limit	+63.00 A	+189.00 A	+84.00 A	+252.00 A
IPK- Limit	-63.00 A	-189.00 A	-84.00 A	-252.00 A

<b>MISC Configuration</b>	<b>ASR-6450</b>	<b>ASR-6600</b>
T peak , hold(msec)	1	1
Phase Mode	Unbalance	Unbalance
Peak CLR	ALL	ALL
Power ON	OFF	OFF
Buzzer	ON	ON
Remote Sense	OFF	OFF
V Response	Medium	Medium
Output Relay	Enable	Enable
THD Format	IEC	IEC
External Control	OFF	OFF
V Unit(TRI,ARB)	rms	rms
Set Change Phase	OFF	OFF
Monitor Output1	L1 Voltage	L1 Voltage
Monitor Output2	L1 Current	L1 Current
Monitor Output Amp	±2.5	±2.5
TrgOut Width(ms)	0.1	0.1
TrgOut Source	L1	L1
Re-Lock	ON	ON
Data Average Count	8	8
Data Update Rate	Fast	Fast

<b>LAN</b>	<b>ASR-6450</b>	<b>ASR-6600</b>
DHCP	ON	ON

<b>USB Device</b>	<b>ASR-6450</b>	<b>ASR-6600</b>
Speed	Full	Full
Mode	TMC	TMC

<b>RS232C</b>	<b>ASR-6450</b>	<b>ASR-6600</b>
Baudrate	9600	9600
Databits	8bits	8bits
Parity	None	None
Stopbits	1bit	1bit

<b>GPIO</b>	<b>ASR-6450</b>	<b>ASR-6600</b>
Address	10	10

<b>CAN BUS</b>	<b>ASR-6450</b>	<b>ASR-6600</b>
Baudrate	125K	125K
Node ID	127	127

<b>DeviceNet</b>	<b>ASR-6450</b>	<b>ASR-6600</b>
Baudrate	125K	125K
MAC ID	63	63

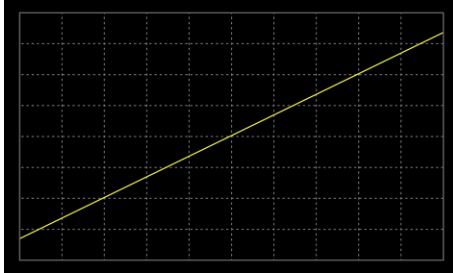
<b>Output Impedance</b>	<b>ASR-6450</b>	<b>ASR-6600</b>
Output Impedance	OFF	OFF
L1 Output Inductance	0.1 $\mu$ H	0.1 $\mu$ H
L2 Output Inductance	0.1 $\mu$ H	0.1 $\mu$ H
L3 Output Inductance	0.1 $\mu$ H	0.1 $\mu$ H
L1 Output Resistance	0.1 $\Omega$	0.1 $\Omega$
L2 Output Inductance	0.1 $\Omega$	0.1 $\Omega$
L3 Output Inductance	0.1 $\Omega$	0.1 $\Omega$

<b>Sequence Mode</b>	<b>ASR-6450</b>			<b>ASR-6600</b>		
Step	0			0		
Time	0.1000 s			0.1000 s		
Jump To	OFF			OFF		
Jump Cnt	1			1		
Branch 1	OFF			OFF		
Branch 2	OFF			OFF		
Term	CONTI			CONTI		
Sync Code	LL			LL		
Item	L1	L2	L3	L1	L2	L3
ACV	0.00, CT	0.00, CT	0.00, CT	0.00, CT	0.00, CT	0.00, CT
DCV	0.00, CT	0.00, CT	0.00, CT	0.00, CT	0.00, CT	0.00, CT
Fset	50.0 , CT	50.0 , CT	50.0 , CT	50.0 , CT	50.0 , CT	50.0 , CT
Wave	SIN	SIN	SIN	SIN	SIN	SIN
ON Phs	Free	Free	Free	Free	Free	Free
OFF Phs	Free	Free	Free	Free	Free	Free
Phase	Fixed (0)	120	240	Fixed (0)	120	240

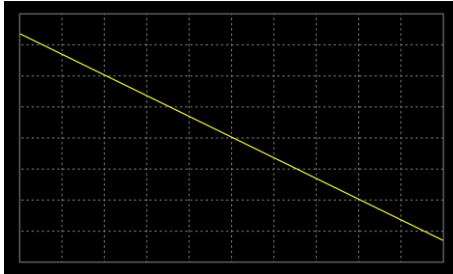
<b>Simulate Mode</b>	<b>ASR-6450</b>			<b>ASR-6600</b>		
Step	Initial			Initial		
Repeat	OFF			OFF		
Time	0.1000 s			0.1000 s		
Term	Free			Free		
Code	LL			LL		
Item	L1	L2	L3	L1	L2	L3
ACV	0.00	0.00	0.00	0.00	0.00	0.00
Fset	50.00	50.00	50.00	50.00	50.00	50.00
Wave	SIN	SIN	SIN	SIN	SIN	SIN
ON Phs	Free	Free	Free	Free	Free	Free
OFF Phs	Free	Free	Free	Free	Free	Free

## Default Waveform Setting

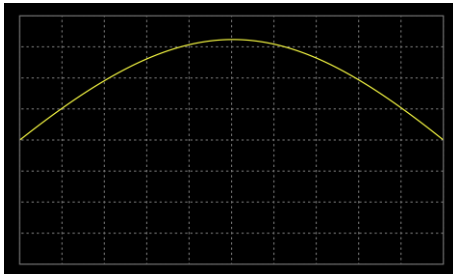
ARB 1 Ramp (rising)



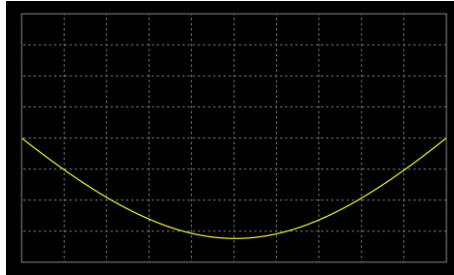
ARB 2 Ramp (falling)



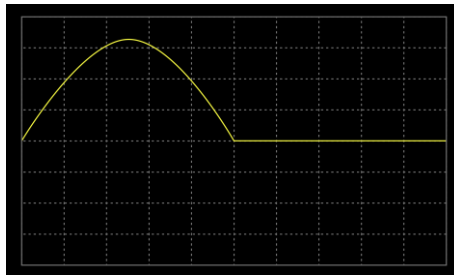
ARB 3 Sine wave, half-cycle(positive pole)



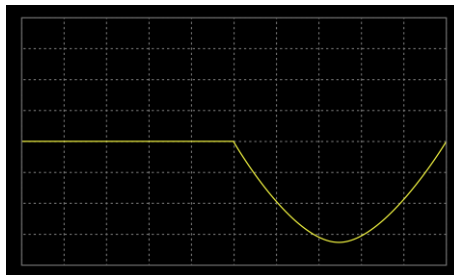
ARB 4 Sine wave, half-cycle(negative pole)



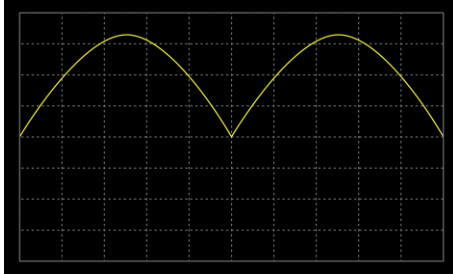
ARB 5 Sine wave, half-wave rectification (positive polarity)



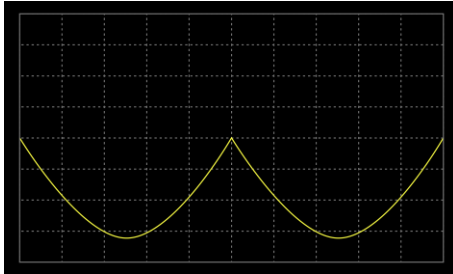
ARB 6 Sine wave, half-wave rectification(negative polarity)



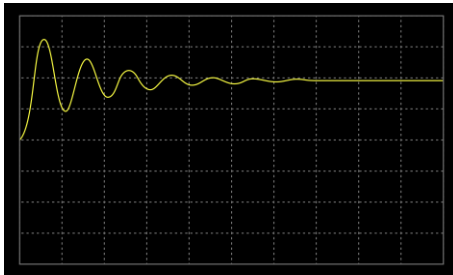
ARB 7 Sine wave, full-wave  
rectification(positive polarity)



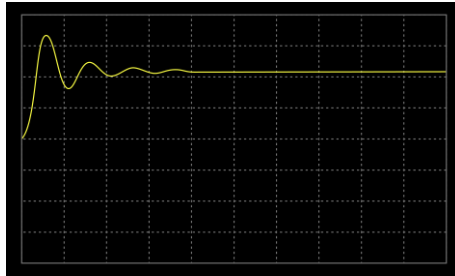
ARB 8 Sine wave, full-wave  
rectification(negative polarity)



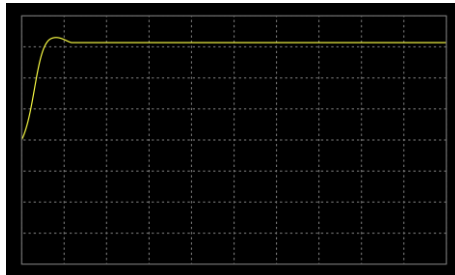
ARB 9 Second order step response(damping  
coefficient 0.1)



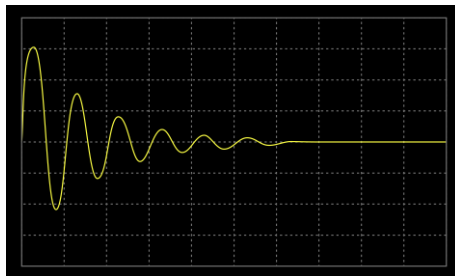
ARB 10 Second order step response(damping coefficient 0.2)



ARB 11 Second order step response(damping coefficient 0.7)

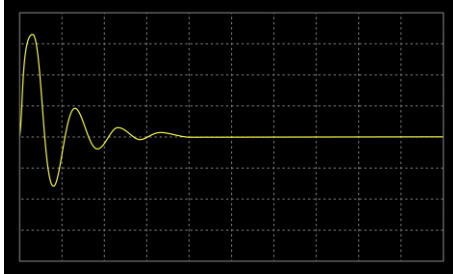


ARB 12 Second order impulse response(damping coefficient 0.1)

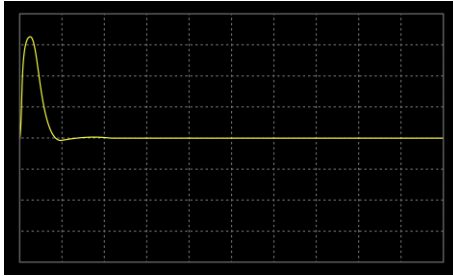




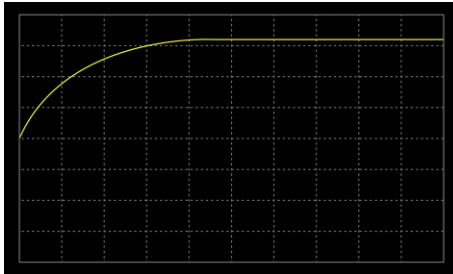
ARB 13 Second order impulse  
response(damping coefficient 0.2)



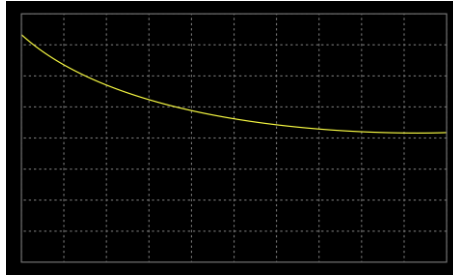
ARB 14 Second order impulse  
response(damping coefficient 0.7)



ARB 15 Exponential (rising)

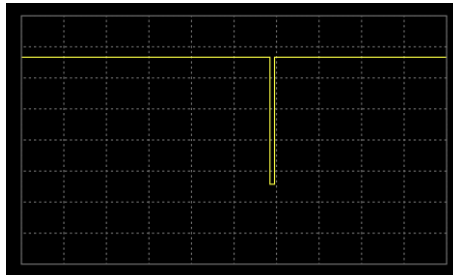


ARB 16 Exponential (falling)

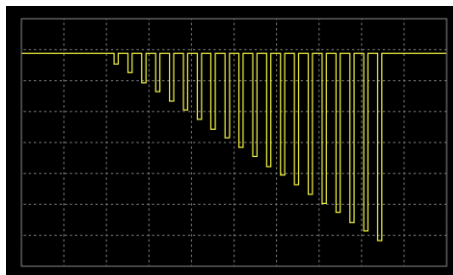


Default Sequence Setting

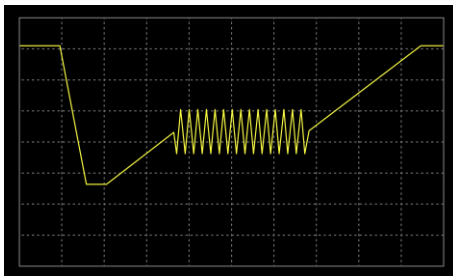
SEQ6 Momentary drop in supply voltage



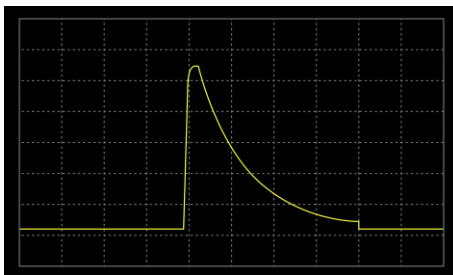
SEQ7 Reset test for Level1 systems with 12V



## SEQ8 Starting Profile



## SEQ9 Test2 Tr: 10ms, Td: 40ms



# INDEX

Accessories .....	11	symbol.....	4
Caution symbol .....	4	List of features.....	10
Cleaning the instrument.....	7	Model differences .....	9
Default settings.....	312, 321, 322	Power on/off	
Disposal instructions.....	7	safety instruction .....	6
EN61010		Remote control.....	21, 49
pollution degree .....	7	Command list.....	83
Environment		RS232.....	50
safety instruction.....	7	Service operation	
Front panel diagram .....	13	about disassembly .....	5
Ground		Warning symbol .....	4