SBC840 SBC840M

Command Strings

Shure SBC840 and SBC840M command strings for third-party control systems, such as Crestron or Extron. Includes all supported programming commands.

Version: 1.1 (2020-K)

Table of Contents

SBC840 SBC840M Command Strings **3**

Command Strings Overview **3**

Command Strings **3**

SBC840 SBC840M Command Strings

Command Strings Overview

Shure command strings are a set of commands and status reports used by control system programmers to interface to Shure devices. The Shure device is connected via Ethernet to a control system, such as

- AMX, Crestron or Extron
- Symetrix, Biamp, other digital signal processors (DSP)
- Specialized custom programs

The Shure device is considered to be the server and the control system is considered to be the client.

Connection: Ethernet (TCP/IP; select “Client” in the AMX/Crestron program)
Port: 2202

Conventions

There are 4 types of strings

GET	Finds the status of a parameter. After the AMX/Crestron sends a GET command, the system responds with a REPORT string
SET	Changes the status of a parameter. After the AMX/Crestron sends a SET command, the system responds with a REPORT string to indicate the new value of the parameter.
REP	• When the system receives a GET or SET command, it replies with a REPORT command to indicate the status of the parameter. • REPORT is also automatically sent by the device when a status changes, for example: As a battery charges, the charger sends the reports without any GET commands: < REP 1 BATT_TIME_TO_FULL 00107 > < REP 1 BATT_TIME_TO_FULL 00106 > < REP 1 BATT_TIME_TO_FULL 00105 >
SAMPLE	Used for metering audio levels. (Not applicable with some Shure devices.)

Note:

- All messages sent and received are ASCII. Note that the level indicators and gain indicators are also ASCII.
- It is not necessary to constantly query parameters because most parameters send a REPORT command when they change.

Command Strings

ALL

Description	Discovery of device properties.
Commands	< GET *x* ALL > < REP ... >
Variables	When **x** is zero, the device responds with REP for all device-specific properties and ALL channel, module, or bay-related properties including all metered properties. When **x** is a channel, module, or bay number, the device responds with REP for all device-specific properties and ALL channel, module, or bay **x**-related properties, including all metered properties.
Notes	None.

BATT_BARS

Description	Discovers the number of bars for a battery.
Commands	< GET *x* BATT_BARS > < REP *x* BATT_BARS 003 > When the number of bars changes: < REP *x* BATT_BARS 004 >
Variables	Where *x* is the bay number. Using 0 returns information for all bays.
Notes	Numeric, three characters 000 - 005 : Number of bars reported 254 : An error has occurred, the value is not applicable at this time 255 : Unknown

BATT_CAPACITY_MAX

Description	Discovers the manufacturer's battery maximum capacity in mAh.
Commands	< GET *x* BATT_CAPACITY_MAX > < REP *x* BATT_CAPACITY_MAX 02393 >

Variables	Where *x* is the bay number. Using 0 returns information for all bays.
Notes	Numeric, five characters 00000 - 65533 : The manufacturer's battery maximum capacity in mAh 65534 : An error has occurred, the value is not applicable at this time 65535 : No battery or not applicable

BATT_CHARGE

Description	Discovers the charge in percent for a battery.
Commands	< GET *x* BATT_CHARGE > < REP *x* BATT_CHARGE 027 > < REP *x* BATT_CHARGE 028 > ... < REP *x* BATT_CHARGE 099 > < REP *x* BATT_CHARGE 100 >
Variables	Where *x* is the bay number. Using 0 returns information for all bays.
Notes	Numeric, three characters 000 - 100 : Percentage of charge 254 : An error has occurred, the value is not applicable at this time 255 : Unknown

BATT_CURRENT_CAPACITY

Description	Discovers the current battery capacity in mAh.
Commands	< GET *x* BATT_CURRENT_CAPACITY > < REP *x* BATT_CURRENT_CAPACITY 02189 >
Variables	Where *x* is the bay number. Using 0 returns information for all bays.

Notes	Numeric, five characters 00000 - 65533 : The current battery capacity in mAh 65534 : An error has occurred, the value is not applicable at this time 65535 : No battery or not applicable

BATT_CURRENT_CAPACITY_MAX

Description	Discovers the current maximum capacity in mAh.
Commands	< GET *x* BATT_CURRENT_CAPACITY_MAX > < REP *x* BATT_CURRENT_CAPACITY_MAX 02393 >
Variables	Where *x* is the bay number. Using 0 returns information for all bays.
Notes	Numeric, five characters 00000 - 65533 : The current battery maximum capacity in mAh 65534 : An error has occurred, the value is not applicable at this time 65535 : No battery or not applicable

BATT_CYCLE

Description	Discovers the number charging cycles for a battery.
Commands	Battery placed into charger bay *x*: < REP *x* BATT_CYCLE 00006 > ... < GET *x* BATT_CYCLE > < REP *x* BATT_CYCLE 00006 >
Variables	Where *x* is the bay number. Using 0 returns information for all bays.
Notes	Numeric, five characters 00000 - 65533 : Number of charging cycles 65534 : An error has occurred, the value is not applicable at this time

	65535 : Unknown or not applicable

BATT_DETECTED

Description	Discovers if a battery is detected.
Commands	< GET *x* BATT_DETECTED > < REP *x* BATT_DETECTED YES >
Variables	Where *x* is the bay number. Using 0 returns information for all bays.
Notes	Fixed String YES NO

BATT_ERROR

Description	Discovers the error status of a battery.
Commands	< GET *x* BATT_ERROR > < REP *x* BATT_ERROR 000 >
Variables	Where *x* is the bay number. Using 0 returns information for all bays.
Notes	Numeric, three characters 000 : No Active Error 001 : Unknown Module 002 : Unrecognized Battery 003 : Deep Discharge Recovery Failed 004 : Charge Failed 005 : Check Battery 006 : Check Charger 007 : Communication Failure 255 : No Battery Present

BATT_HEALTH

Description	Discovers the health in percent for a battery.
Commands	< GET *x* BATT_HEALTH > < REP *x* BATT_HEALTH 099 >
Variables	Where *x* is the bay number. Using 0 returns information for all bays.
Notes	Numeric, three characters 000 - 100 : Percentage of health 254 : An error has occurred, the value is not applicable at this time 255 : Unknown

BATT_MODULE_TYPE

Description	Discovers the type of the battery module.
Commands	< GET *x* BATT_MODULE_TYPE > < REP *x* BATT_MODULE_TYPE value >
Variables	Using 0 returns information for the module
Notes	Numeric string, 3 characters 000 : No module installed 128: SBC840 130 SBC840M 255 : Invalid or unsupported module

BATT_STATE

Description	Discovers the state of a battery.
Commands	< GET *x* BATT_STATE > < REP *x* BATT_STATE NORMAL > After some period of time, battery becomes fully charged: < REP *x* BATT_STATE FULL >

Variables	Where *x* is the bay number. Using 0 returns information for all bays.
Notes	Fixed String FULL CALCULATING NORMAL WARM WARM_FULL HOT COLD PRECHARGE READY_TO_STORE DISCHARGE_CALC DISCHARGING DISCHARGING_WARM DISCHARGING_COLD ERROR NO_BATT

BATT_TEMP_C

Description	Discovers the temperature in Celsius.
Commands	< GET *x* BATT_TEMP_C > < REP *x* BATT_TEMP_C 055 > There is an offset of 40 so the actual value = 55 - 40 = 15° C.
Variables	Where *x* is the bay number. Using 0 returns information for all bays.
Notes	The actual value = the reported value - 40 Numeric, three characters 000 - 253 : Temperature in Celsius 254 : An error has occurred, the value is not applicable at this time 255 : Unknown

BATT_TEMP_F

Description	Discovers the temperature in Fahrenheit.

Commands	< GET *x* BATT_TEMP_F > < REP *x* BATT_TEMP_F 095 > There is an offset of 40 so the actual value = 95 - 40 = 50° F.
Variables	Where *x* is the bay number. Using 0 returns information for all bays.
Notes	The actual value = the reported value - 40 Numeric, three characters 000 - 253 : Temperature in Fahrenheit 254 : An error has occurred, the value is not applicable at this time 255 : Unknown

BATT_TIME_TO_FULL

Description	Discovers the number of minutes for a battery to reach the target charging level.
Commands	< GET *x* BATT_TIME_TO_FULL > < REP *x* BATT_TIME_TO_FULL 00060 > Battery placed into charger bay *x*: < REP *x* BATT_TIME_TO_FULL 65533 > ... < REP *x* BATT_TIME_TO_FULL 00060 > ... < REP *x* BATT_TIME_TO_FULL 00001 > < REP *x* BATT_TIME_TO_FULL 00000 > < REP *x* BATT_TIME_TO_FULL 65529 > Battery removed: < REP *x* BATT_TIME_TO_FULL 65535 >
Variables	Where *x* is the bay number. Using 0 returns information for all bays.
Notes	Numeric, five characters Considered as time to target where: Charging Mode: Value is the estimated time to full charge. Storage Mode: Value is the estimated time to optimal storage voltage.

	00000 - 65528: Number of minutes estimated to reach the target 65529 : Battery is fully charged 65533 : Calculation in progress 65534 : Error has occurred 65535 : Unknown or not applicable

DEVICE_ID

Description	Controls the Device ID.
Commands	< GET DEVICE_ID > < REP DEVICE_ID {Name1yyyyyyyyyyyyyyyyyyyyyyyyyyy} > < SET DEVICE_ID {Name1} > < REP DEVICE_ID {Name1yyyyyyyyyyyyyyyyyyyyyyyyyyy} >
Variables	Where the repeating **y** represents the spaces returned by the device to pad the Device ID to 31 characters.
Notes	The device always responds with 31-character ID. SET accepts 1-8 Characters from the set: A-Z,a-z,0-9,!"#$%&'()*+,-./:;<=>?@[\]^_`~ and space.

FW_VER

Description	Discovery of the firmware version.
Commands	Self test passed: < GET FW_VER > < REP FW_VER {2.0.15.2yyyyyyyyyyyyyyyy} > Self test failed: < GET FW_VER > < REP FW_VER {2.0.15.2*yyyyyyyyyyyyyyy} >
Variables	Where the repeating **y** represents the spaces returned by the device to pad the response to 24 characters.
Notes	Package version number reported as Maj.Min.Pack.Build.

FLASH

Description	Controls the flash to identify a device.

Commands	< SET FLASH ON > < REP FLASH ON > The device will indicate < REP FLASH OFF > when the identification has completed.
Variables	None.
Notes	Device initiates an Identify then stops flashing.

MODEL

Description	Discovery of the model name of the device.
Commands	< GET MODEL > < REP MODEL {SBC840yyyyyyyyyyyyyyyyyyyyyyyyyy} > < REP MODEL {SBC840Myyyyyyyyyyyyyyyyyyyyyyyyyy} >
Variables	Where the repeating **y** represents the spaces returned by the device to pad the model name to 32 characters.
Notes	The device always responds with a 32-character model name.

STORAGE_MODE

Description	Controls the storage mode setting.
Commands	< GET STORAGE_MODE > < REP STORAGE_MODE OFF > < SET STORAGE_MODE ON > < REP STORAGE_MODE ON > < SET STORAGE_MODE TOGGLE > < REP STORAGE_MODE OFF >
Variables	None.
Notes	TOGGLE switches between ON and OFF.