OCPP 1.6

Implementation Overview

Notice

This document contains information about one or more ABB products and may include a description of or a reference to one or more standards that may be generally relevant to the ABB products. The presence of any such description of a standard or reference to a standard is not a representation that all of the ABB products referenced in this document support all of the features of the described or referenced standard. In order to determine the specific features supported by a particular ABB product, the reader should consult the product specifications for the particular ABB product.

ABB may have one or more patents or pending patent applications protecting the intellectual property in the ABB products described in this document.

The information in this document is subject to change without notice and should not be construed as a commitment by ABB. ABB assumes no responsibility for any errors that may appear in this document.
In no event shall ABB be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB be liable for incidental or consequential damages arising from use of any software or hardware described in this document.
This document is originally written in English. Other language versions are a translation of the original document and ABB cannot be held liable for errors in the translation.
This document and parts thereof must not be reproduced or copied without written permission from ABB, and the contents therefore must not be imparted to a third party nor used for any unauthorized purpose.

Copyrights

Table of Contents

Revisions

Revision	Description	Date	Author
A	First issue	28 Jul 2020	Sarah Sun
V1.1	Add few details	11 Dec 2020	Sarah Sun
V1.2	Clean up & fixes	31 Mar 2021	Loredana Negriu
V1.3	Update some latest info	17 Jun 2021	Sarah Sun
V1.4	Update some latest info	30 Jun 2021	Loredana Negriu
V1.5	Update some latest Info	2 Sep 2021	Sarah Sun

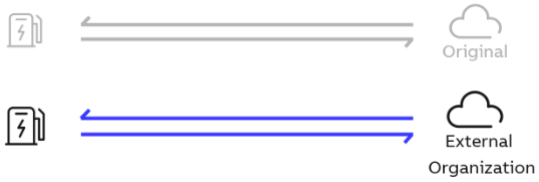
Overview

ABB Terra AC chargers support OCPP 1.6 J. This document describes OCPP 1.6 functionality supported by ABB Terra AC chargers according to OCPP protocol specification.

ABB has implemented OCPP 1.6-J version, which means using JSON over Websockets. The charger is either connected to an OCPP Server directly or is connected to the ChargerSync server by default. The connection to the ChargerSync portal allows for efficient remote support and enables additional features next to OCPP. This concept is referred to as below.

Link to the ABB ChargerSync server

Link to a 3rd party OCPP backend server



OCPP server configuration

Contact your local ABB sales representative to arrange access to TerraConfig App account and facilitate company creation and new OCPP configuration via the ChargerSync Portal.

TerraConfig App: configuring the charger (repeated for every charger commissioning that needs connection to a 3rd party OCPP backend).

1. Download TerraConfig app and use the credentials sent in the email, generated by the creation of the account.
2. Pair the TerraConfig app with the charger and check firmware version
   If required, update FW to v1.4.2 or higher (via TerraConfig app)
3. Make sure the Terra AC is connected to the internet (over WiFi, LAN or 4G)
4. Enable external OCPP server and choose the correct backend URL which has been preconfigured in the portal according to the steps above. Then press configure and afterwards OK
5. Check via Device info that the URL is the correct one and that connection has been established. Via OCPP logs (of the 3rd party backend) validate that BootNotification is successfully sent
6. Run some remote commands to confirm good communication between charger and backend

WebSocket communication

For the connection between a Charge Point and a Central System using OCPP-J, the Central System acts as a WebSocket server and the Charge Point acts as a WebSocket client.

Client Request:

The following is an example of an opening HTTP request of an OCPP-J connection handshake:

***GET** http://*some.server.com:33033*/webServices/ocpp/CP3211 HTTP/1.1*

***Host:** some.server.com:33033*

Upgrade: websocket

Connection: Upgrade

***Sec-WebSocket-Key:** x3JJHMbDL1EzLkh9GBhXDw==*

Authorization: Basic BASE64(ChargerboxID + password)*

Sec-WebSocket-Protocol: ocpp1.6, ocpp1.5

Sec-WebSocket-Version: 13

The bold parts are found as such in every WebSocket handshake request, the other parts are specific to this example. In this example, the Central System's OCPP-J endpoint URL is "ws://some.server.com:33033/webServices/ocpp". The Charge Point's unique identifier is "CP3211", so the path to request becomes "webServices/ocpp/CP3211".

Remark:

The FW before 1.3.x We will still send full URL address instead of the one after path

*The Password was configured in TerraConfig portal and send to TerraAC charger via Bluetooth with TerraConfig APP

Server Response:

Upon receiving the Charge Point's request, the Central System has to finish the handshake with a response as described in [RFC6455].

So, if the Central System accepts the above example request and agrees to using OCPP 1.6J with the Charge Point, the Central System's response will look as follows:

HTTP/1.1 101 Switching Protocols

Upgrade: websocket

Connection: Upgrade

***Sec-WebSocket-Accept:** s3pPLMBiTxaQ9kYGzzhZRbK+xOo=*

***Sec-WebSocket-Protocol:** ocpp1.6*

The bold parts are found as such in every WebSocket handshake response, the other parts are specific to this example.

Remark:

For the FW before 1.3.x a different response required that is the sever should response with HTTP/1.1 101 **Switch Protocols**, other than this one, the charger point will reject

For the FW after 1.3.x , the server could response with HTTP/1.1 101 and anything different from Switch

protocol could also be accepted by charger point.

WebSocket Ping in relation to OCPP Heartbeat

The WebsSocket specification defines Ping and Pong frames that are used to check if the remote endpoint is still responsive. In practice this mechanism is also used to prevent the network operator from quietly closing the underlying network connection after a certain period of inactivity. This websocket feature can be used as a substitute for most of the OCPP Heartbeat messages but cannot replace all of its functionality.

Remark:

ABB Chargersync Platform does not use this PING PONG mechanism, and our charger will not send a PING request.

ABB TerraAC Charger will only respond with PONG to the PING request from the server.

Before FW 1.3.x the charger will not respond to the PING request from the server if the message has an empty body.

Supported functionality

The implementation is following OCPP 1.6 specification of Open Charge Alliance. According to OCPP 1.6 specification all of features and associated messages are grouped into Feature Profiles.

OCPP 1.6 specified following Feature profiles:

Profile name	Description	Mandatory
Core	Basic Charge Point functionality comparable with OCPP 1.5 without support for firmware updates, local authorization list management and reservations.	Yes
Firmware Management	Support for firmware update management and diagnostic log file download.	No
Local Auth List Management	Features to manage the local authorization list in Charge Points.	No
Reservation	Support for reservation of a Charge Point	No
Smart Charging	Support for basic Smart Charging	No
Remote Trigger	Support for remote triggering of Charge Point initiated messages	No

For more about Feature Profiles please see [1] “3.3 Feature Profiles” in fill in link to OCPP 1.6 official

Please see below which ABB Charging products support which OCPP 1.6 Feature profiles from which software version:

Profile / Model	Core	Firmware* Management	Local Auth List Management	Reservation	Smart Charging	Remote Trigger
Terra AC Chargers	0.4.x	0.4.x*	0.5.x	To be supported**	1.x.x	To be supported**

* ABB provides a URL link to download the bin file package in zip format for each firmware release. The customer then uploads the zip file to their own OCPP back server and generates an URL address for it. The URL address is dropped in the location of firmware update while the charger connects to the customer server. During the update procedure, the charger will link to the URL address to download the firmware package and install it. After finishing this procedure, the charger will reboot and reconnect to the customer server.>>add on same FW folder cross ref doc

** Reservation and Remote Trigger profiles are not yet implemented in current version of software for the chargers. ABB intends to provide this functionality in the future software versions.

Please see below which messages are supported per OCPP feature profile.

Message	Supported (Y/N)	Comment
Core profile		
Authorize	Y	
BootNotification	Y	
ChangeAvailability	Y	
ChangeConfiguration	Y	
ClearCache	Y	The cache is empty, while the charger received the command, it will response accept but do nothing
DataTransfer	Y	While the charger connects to customer's own back end, customer could use this message for log transfer NOTE: Please see document for data transfer
GetConfiguration	Y	Before FW 1.3 the charger will only response with support keys. After FW1.3 charger will response with all keys, in case the key not supported, the charger will response "unknown" See section xx for Support key
HeartBeat	Y	
MeterValues	Y	ABB supports following Measurand types for AC: 1. Energy.Active.Import.Register 2. current import, 3. voltage, 4. power active import 5. Current.Offer
RemoteStartTransaction	Y	
RemoteStopTransaction	Y	
Reset	Y	AC chargers only support hard reset Hard reset fully reboot charger. The resets gracefully stops charging session if one is in progress before resetting.
StartTransaction	Y	
StatusNotification	Y	
StopTransaction	Y	
UnlockConnector	Y	Message is supported only to socket variants, upon receiving this message, socket variants charger will release the E-lock of socket. If send the message to cable variants, the message will be rejected.

Smart Charging		
SetChargingProfile	Y	1.x.x only support Max stack =0 ChargingScheduleAllowedChargingRateUnit = A ChargingScheduleMaxPeriods = 3
ClearChargingProfile	Y	Before FW 1.3 the charger will not response with

		the message if the body is empty
GetCompositeSchedule	N	
FirmwareManagement profile*		
GetDiagnostics	N	ABB AC charger implement the diagnostic in the self-defined data transfer message, detail info will be shared in the data transfer document.
DiagnosticsStatusNotification	N	
FirmwareStatusNotification	Y	ABB TerraAC Charger will response the status: Downloading Installed DownloadFailed InstallationFailed
UpdateFirmware	Y	
Local Authorization List Management		
GetLocalListVersion	Y	
SendLocalList	Y	Each list is limited to 8 ID tag, each ID tag with max 20 characters; The charger has a total limit of 16 ID tags

Configuration keys

Supported configuration keys

Please see below which configuration keys are supported per OCPP feature profile.

Key Name	Required/ Optional	Description	Type	Accessibility	Default Value
Core profile					
GetConfigurationMaxKeys	required	The number of configuration keys requested in a single PDU may be limited by the Charge Point. This maximum can be retrieved by reading this configuration key.	int	R	14
HeartbeatInterval	required	Interval of inactivity (no OCPP exchanges) with central system after which the Charge Point should send a Heartbeat.req PDU. If the interval less than 10, the AC charger will accept but execute 10	int	RW	120
MeterValuesSampleInterval	required	Interval between sampling of metering (or other) data, intended to be transmitted by "MeterValues" PDUs. The range of this value:0, 4 - 65534 If the interval less than 4, the AC charger will accept but execute 4	int	RW	30
LocalAuthorizeOffline	required	Controls whether a Charge Point will authorize a user when offline using the Authorization Cache and/or the Local Authorization List.	Boolean	RW	TRUE
LocalPreAuthorize	required	Controls whether a Charge Point will use the Authorization Cache and/or the Local Authorization List to start a transaction without waiting for an authorization response from the Central System.	Boolean	RW	FALSE

NumberOfConnectors	required	The number of physical charging connectors of this Charge Point.	int	R	1
SupportedFeatureProfiles	required	A list of supported Feature Profiles. Possible profile identifiers: Core, FirmwareManagement, LocalAuthListManagement, Reservation, SmartCharging and RemoteTrigger.	CSL	R	Core,Firmware,Local Authication List,smart charging
AllowOfflineTxForUnknownId	optional	When offline, a Charge Point may allow automatic authorization of any "unknown" identifiers that cannot be explicitly authorized by Local Authorization List or Authorization Cache entries. Identifiers with status other than "Accepted" (Invalid, Blocked, Expired) must be rejected. Now the charger will not allow any ID except in local authentication list while it is offline	Boolean	RW	False
AuthorizeRemoteTxRequests	required	Whether a remote request to start a transaction in the form of a RemoteStartTransaction.req message should be authorized beforehand like a local action to start a transaction. Now the charger will not send the authorize.req	Boolean	RW	False

Local Authorization List Management					
LocalAuthListEnabled	required	Whether the Local Authorization List is enabled	Boolean	RW	TRUE
LocalAuthListMaxLength	required	Maximum number of identifications that can be stored in the Local Authorization List	int	R	16
SendLocalListMaxLength	required	Maximum number of identifications that can be send in a single SendLocalList.req	int	R	8

Smart charging profile					
ChargeProfileMaxStackLevel	required	Max StackLevel of a Charging. The number defined also indicates the max allowed number of installed charging schedules per Charging Purposes.	int	R	0
ChargingScheduleAllowedChargingRateUnit	required	A list of supported quantities for use in a ChargingSchedule. Allowed values: 'Current' and 'Power'.	CSL	R	Current
ChargingScheduleMaxPeriods	required	Maximum number of periods that may be defined per ChargingSchedule.	int	R	3
MaxChargingProfileInstalled	required	Maximum number of Charging installed at a time.	int	R	1

To be Supported configuration keys

Key Name	Required/ Optional	Description	Type	Accessibility	Default Value
Core profile					
ClockAlignedDataInterval	required	Size (in seconds) of the clock-aligned data interval. This is the size (in seconds) of the set of evenly spaced aggregation intervals per day, starting at 00:00:00 (midnight). For example, a value of 900 (15 minutes) indicates that every day should be broken into 96 15-minute intervals.	int	RW	Unknown
ConnectionTimeOut	required	Interval(from successful authorization) until incipient charging session is automatically canceled due to failure of EV user to (correctly) insert the charging cable connector(s) into the appropriate connector(s). Now The timeout is 120	int	RW	Unknown
ConnectorPhaseRotation	required	For individual connector phase rotation information, the Central System may query the ConnectorPhaseRotation configuration key on the Charging Point via GetConfiguration. The Charge Point shall report the phase rotation in respect to the grid connection.	CSL	RW	Unknown
MeterValuesAlignedData	required	Clock-aligned measurand(s) to be included in a MeterValues.req PDU, every ClockAlignedDataInterval seconds	CSL	RW	Unknown
MetetrValuesSampledData	required	Sampled measurands to be included in a MeterValues.req PDU, every MeterValueSampleInterval seconds. Now the value is Energy.Active.Import.Register,current import,voltage,power active import, current. Offered	CSL	RW	Unknown

ResetRetries	required	Number of times to retry an unsuccessful reset of the Charge Point. Now the value is 0	int	RW	Unknown
StopTransactionOnInvalidId	required	Whether the Charge Point will stop an ongoing transaction when it receives a non- Acceptedauthorization status in a StartTransaction.conf for this transaction. Now the default value is true.	Boolean	RW	Unknown
StopTxnAlignedData	required	Clock-aligned periodic measurand(s) to be included in the TransactionData element of StopTransaction.req MeterValues.req PDU for every ClockAlignedDataInterval of the charging session.	CSL	RW	Unknown
StopTxnSampledData	required	Sampled measurands to be included in the TransactionData element of StopTransaction.req PDU, every MeterValueSampleInterval seconds from the start of the charging session	CSL	RW	Unknown
TransactionMessageAttempts	required	How often the Charge Point should try to submit a transaction-related message when the Central System fails to process it. Now the transaction data will always attempt to send to central system until it response	int	RW	Unknown
TransactionMessageRetryInterval	required	How long the Charge Point should wait before resubmitting a transaction related message that the Central System failed to process.	int	RW	Unknown

Not supported configuration keys

Following configuration keys are NOT SUPPORTED:

Key Name	Required/ Optional	Description	Type
Core profile			
AuthorizationCacheEnabled	optional	A Charge Point may implement an Authorization Cache that autonomously maintains a record of previously presented identifiers that have been successfully authorized by the Central System.	Boolean
MinimumStatusDuration	optional	The minimum duration that a Charge Point or Connector status is stable before a StatusNotification.req PDU is sent to the Central System.	int
WebSocketPingInterval	optional	Only relevant for websocket implementations. 0 disables client side websocket Ping/Pong. In this case there is either no ping/pong or the server ini- tiates the ping and client responds with Pong. Positive values are interpreted as number of sec- onds between pings. Negative values are not al- lowed. ChangeConfiguration is expected to return a REJECTED result. *NOTE: A value of 0 disables client side websocket Ping / Pong. In this case there is either no ping / pong or the server initiates the ping and client re- sponds with Pong. Positive values are interpreted* *as number of seconds between pings*	int
BlinkRepeat	optional	Number of times to blink Charge Point lighting when signaling	int
ConnectorPhaseRotationMaxLength	optional	Maximum number of items in a ConnectorPhaseRotation Configuration Key	int
LightIntensity	optional	Percentage of maximum intensity at which to illuminate Charge Point lighting	int
MaxEnergyOnInvalidId	optional	Maximum energy in Wh delivered when an identifier is invalidated by the Central System after start of a transaction.	int
StopTxnAlignedDataMaxLength	optional	Maximum number of items in a StopTxnAlignedData Configuration Key.	int
StopTransactionOnEVSideDisconnect	required	When set to true, the Charge Point shall administratively stop the transaction when the cable is unplugged from the EV. NOTE: this parameter is not being used, Transaction will always stop on EV disconnect or even before.	Boolean

UnlockConnectorOnEVSideDisconnect	required	When set to true, the Charge Point shall unlock the cable on Charge Point side when the cable is unplugged at the EV. NOTE: not applicable for ABB TerraAC chargers, not implemented	Boolean
SupportedFeaturesMaxLength	optional	Maximum number of items in a SupportedFeatures Configuration Key.	int
ConnectorSwitch3to1PhaseSupported	optional	If defined and true, this Charge Point support switching from 3 to 1 phase during a charging session.	Boolean
Reservation profile			
ReserveConnectorZeroSupported	optional	If this configuration key is present and set to true: Charge Point support reservations on connector 0.	Boolean

Security

Encryption

In addition to network level security ABB OCPP 1.6 implementation supports OCPP-J over TLS security. TLS 1.2 is supported. It is up to Central System operator to decide if TLS with Websocket (WSS) is used or not. No additional configuration changes are required to enable it. For more information on encryption with OCPP 1.6-J please see chapter “6.2.1 Encryption” of [2].

Authentication

ABB OCPP 1.6 implementation supports basic HTTP authentication. Username equals charge point ID and password/authorization keys can optionally be set during installation.

Setting authorization key over OCPP after installation is not supported.
For more information on OCPP 1.6-J authentication please see chapter “6.2.2 Authentication” of [2]

Reference documentation

[1] Open Charge Point Protocol 1.6

[2] Open Charge Point Protocol JSON 1.6, OCPP 1.6-J Specification