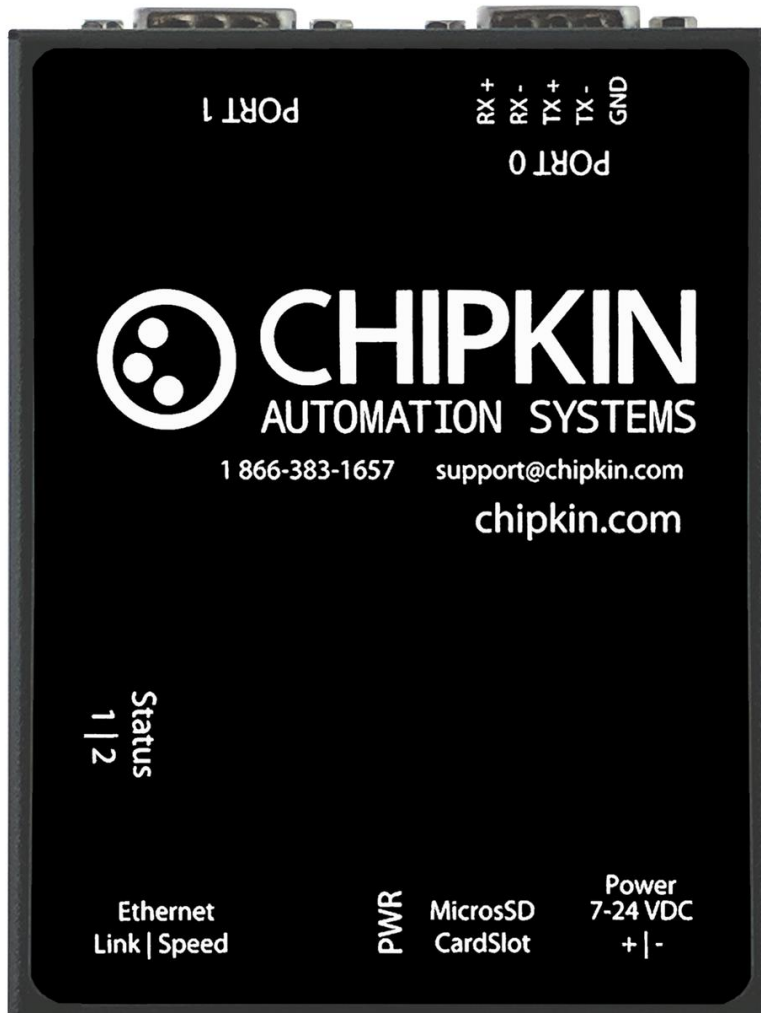


Chipkin™ CAS 2700 HTTP REST API USER MANUAL



VERSION: 1

PRODUCT NUMBER: CAS-2700

DOCUMENT REVISION: 3

TABLE OF CONTENTS

TABLE OF CONTENTS	2
LIST OF FIGURES	3
LIST OF TABLES	3
COPYRIGHT & TRADEMARKS	4
DISCLAIMER	4
CONFIDENTIALITY	4
DOCUMENT REVISION HISTORY	4
1. PREFACE.....	5
1.1 WELCOME	5
1.2 CHIPKIN.....	5
1.3 SAFETY WARNINGS	5
1.4 CUSTOMER SUPPORT	5
2. CONNECTIONS	7
2.1 NETWORK CONNECTIONS.....	7
2.2 COMUNCATION PORTS.....	8
3. HTTP REST API.....	9
3.1 CSV.....	9
3.1.1 Parameters	9
3.1.2 Example - Get system information as CSV with headers.	9
3.2 Act Get Value	10
3.2.1 Parameters	10
3.2.2 Example - Get a specific data point as JSON	10
3.2.3 Example - Get all data point as JSON	11
3.3 Act Set Value	12
3.3.1 Parameters	12
3.3.2 Write thru.....	12
3.3.3 Example - Set a specific data point.....	13
3.4 Act Set	14
3.4.1 Parameters	14
3.4.2 Example - Set the current time.....	15
3.5 Act Get.....	15

3.5.1 Parameters15

3.5.2 Example – Get the current stats15

3.6 Act Info17

3.6.1 Example – Get the current drivers stats.....17

THANK YOU.....19

LIST OF FIGURES

Figure 3.1-1. Network Connections Block Diagram..... 7

LIST OF TABLES

Table 1 - Document Revision History 4

Table 2 - Communication ports 8

Table 3 - Endpoint formats 9

COPYRIGHT & TRADEMARKS

Copyright © 2017 Chipkin Automation Systems All rights reserved.
TM (™) are trademarks of Chipkin Automation Systems

DISCLAIMER

Chipkin Automation Systems™ has limited its liability for damages incurred by the customer or its personnel in the contractual documents pursuant to which the product is provided to the customer. The information and specifications contained throughout this user manual are up to date at the time of publication. Chipkin Automation Systems has used, and continues to use, its best efforts to maintain this user manual to reflect the most current configuration of the product. Chipkin Automation Systems reserves the right to change the contents of this user manual at any time without notice and assumes no liability for its accuracy. In the preparation of this user manual, Chipkin Automation Systems has incorporated, and/or compiled service information and maintenance procedures sourced from manufacturers and vendors of parts and components used in the manufacturing of this product. Therefore, Chipkin Automation Systems shall not be liable for omissions or missing data. It is not the intention of this user manual to instruct service technicians in using common sense, basic skills and rules of service repair.

CONFIDENTIALITY

The information contained in this document is the intellectual property of Chipkin Automation Systems and is Commercially Confidential. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Chipkin Automation Systems.

DOCUMENT REVISION HISTORY

REVISION	DATE	AUTHOR	NOTE
1	2018-Mar-15	SWS	Document created
2	2018-Sep-20	SWS	Added "3.3.4 Example - Set a multiple data point" section.
3	2019-Apr-24	ACF	Added additional optional parameters to Get Value API call

Table 1 - Document Revision History

1. PREFACE

1.1 WELCOME

As a new owner of Chipkin Automation Systems'™ (CAS) Gateway you have joined thousands of satisfied customers who use Chipkin's protocol gateways, data clients and integration services to meet their building and industrial automation requirements. Our configuration expertise in this field combined with free BACnet and other tools ensure your success; and our customer support via phone, email and remote desktop tools means that we're there when you need us. Thank you for choosing Chipkin's products.

1.2 CHIPKIN

Chipkin offers expert solutions for your building and industrial automation requirements. We develop, configure, install and support gateways (protocol converters), data loggers, and remote monitor and controlling applications. Founded in October 2000, Chipkin provides expert solutions for converting BACnet®, Modbus®, and Lonworks®—to name just a few—and enabling interfaces for HVAC, fire, siren, intercom, lighting, transportation and fuel systems. The high-quality products we offer (including those from other vendors) interface with Simplex™, Notifier™, McQuay™, GE™ and many others—so you can rest assured that Chipkin will select the most appropriate solution for your application.

1.3 SAFETY WARNINGS

The CAS Gateway User Manual provides information on how to install and configure the gateway and is intended for engineers, project management consultants and building management services. Before you install the device, please observe the safety warnings described in in this manual.

1.4 CUSTOMER SUPPORT

Chipkin is a small responsive company, and we live or die by the quality of our service—and with offices in two time-zones—we can provide support when you need it. For information on sales, service, obtaining documentation or submitting a service request, please call us toll free at 1-866-383-1657. Thanks for choosing Chipkin's protocol gateways, data clients and integration services to meet your building and industrial automation requirements.



SALES AND CUSTOMER SUPPORT

TOLL FREE: 1-866-383-1657

FAX: 1-416-915-4024

EMAIL: salesgroup1@chipkin.com

GENERAL

TOLL FREE: 1-866-383-1657

FAX: 1-416-915-4024

EMAIL: support@chipkin.com

SHIPPING ADDRESS

3381 Cambie St., #211

Vancouver, BC, Canada V5Z 4R3

2. CONNECTIONS

2.1 NETWORK CONNECTIONS

This block diagram lists common network connections that can monitor and control data from HTTP devices using BACnet IP, Modbus RTU/TCP, etc.

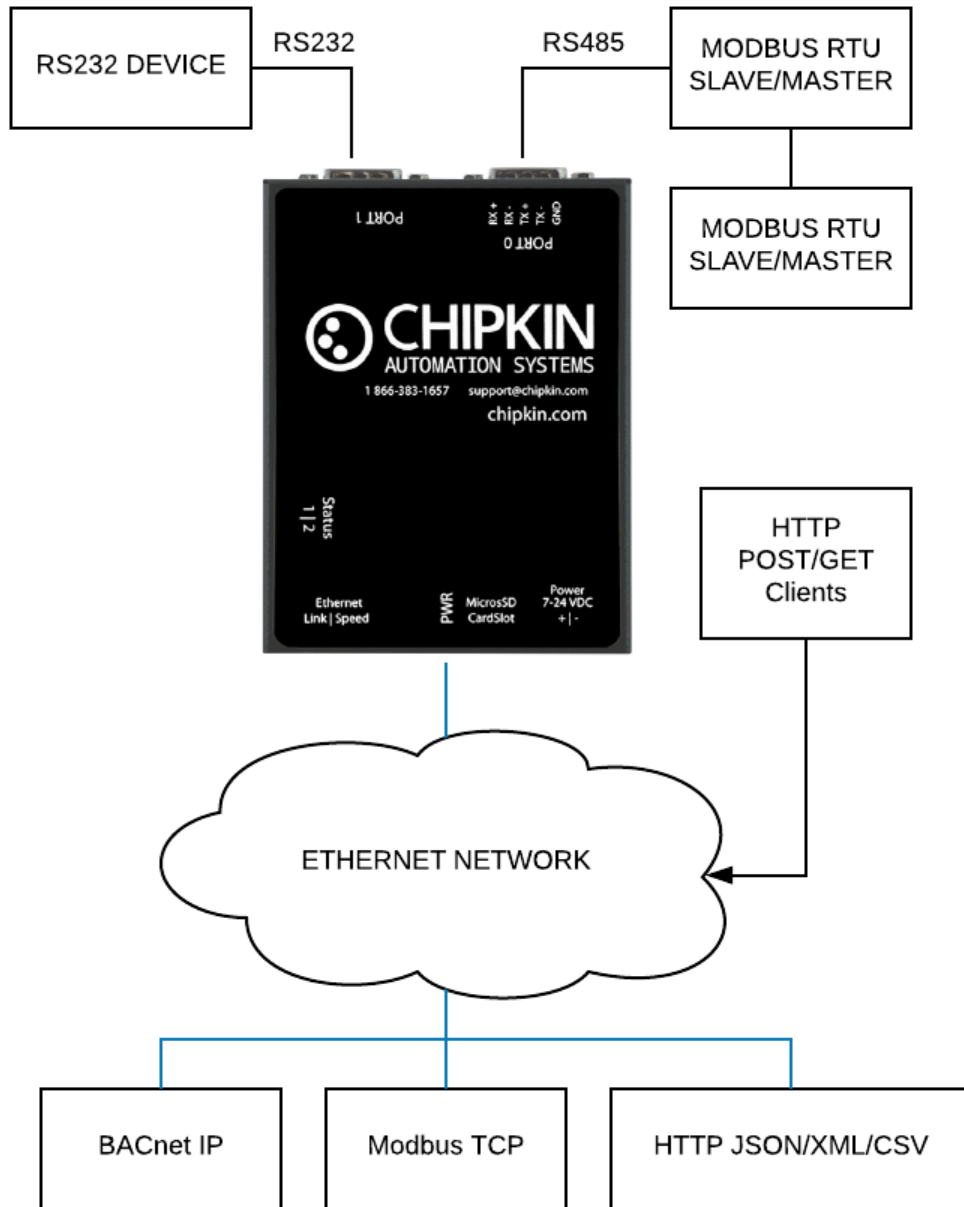


Figure 2.1-1. Network Connections Block Diagram

2.2 COMUNCATION PORTS

The Gateway uses the following ports for communication:

Protocol	Port	Notes
HTTP	TCP 80	Web server.
BACnet IP	UDP 47808	Default port, can be configured.
Modbus TCP	TCP 502	Default port, can be configured.
Syslog	UDP 514	Can be disabled.
FTP	TCP 21	Can be disabled.

Table 2 - Communication ports

3. HTTP REST API

This section contains instructions on how to query the CAS Gateway CAS-2700 via the HTTP REST API.

All examples below are in JSON format.

Endpoints

The CAS Gateway supports requesting information from the system via a HTTP REST API. The responses from the REST server can be formatted as either CSV, XML, or JSON.

Document format	Endpoint
XML - Extensible Markup Language	/bin/xml/
JSON - JavaScript Object Notation	/bin/json/
CSV - Comma Separated Values	/bin/csv/

Table 3 - Endpoint formats

3.1 CSV

The CSV endpoint has a URL parameter that can be used to turn on or off headers in the response.

3.1.1 Parameters

- **use_headers [Optional]** - Include the headers of the columns for the first row. Accepted values are 0 for no headers, or 1 for include headers. Default: 0 (No headers)

3.1.2 Example - Get system information as CSV with headers.

Request:

```
GET /bin/csv/?act=get_value&use_headers=1 HTTP/1.1
```

Response

```
id, modified, data,
1, 0, 0,
2, 0, 0,
3, 0, 0,
4, 0, 0,
5, 0, 0,
6, 0, 0,
7, 0, 0,
```

```
8, 0, 0,
9, 0, 0,
10, 0, 0,
```

3.2 Act Get Value

Gets the current values from the database or a specific value from the database.

Endpoint: /bin/json/?act=get_value

3.2.1 Parameters

- **id [Optional]** - The offset into the database of the data point that is being requested. If the ID is not defined, then the entire database will be returned (Get all).
- **table [Optional]** - The table to extra the data from. By default, primary data table is used da_data.
- **length [Option]** - The number of records to get. Must be used with the id parameter.

3.2.2 Example - Get a specific data point as JSON

In this example the id parameter is specified and only a single data point will be returned.

Request:

```
GET /bin/json/?act=get_value&id=1 HTTP/1.1
```

Response

```
{
  "query": {
    "act": "get_value",
    "id": "1"
  },
  "response": {
    "data": [{
      "data": "1001",
      "id": "1",
      "modified": "1346272391"
    }]
  }
}
```

3.2.3 Example - Get all data point as JSON

In this example the id parameter is not specified and all the data in the database will be return in a single response.

Request:

```
GET /bin/json/?act=get_value HTTP/1.1
```

Response

```
{
  "query": {
    "act": "get_value",
  },
  "response" : {
    "data": [{
      "data": "1001",
      "id": "1",
      "modified": "1346272391"
    }, {
      "data": "1002",
      "id": "2",
      "modified": "1346272391"
    }, {
      "data": "1003",
      "id": "3",
      "modified": "1346272391"
    }, {
      "data": "1004",
      "id": "4",
      "modified": "1346272391"
    }, {
      "data": "1005",
      "id": "5",
      "modified": "1346272391"
    }
  ],
}]
```

```
}

```

3.3 Act Set Value

Sets the current values for a specific offset in the database.

Endpoint: /bin/json/?act=set_value

3.3.1 Parameters

- **id [Required]** - The offset into the database of the data point that is being set.
- **value [Required]** - The value to be written as text.
- **table [Optional]** - The table to extra the data from. By default, primary data table is used da_data.
- **writethru** - Enables the CAS Gateway to attempt to create a write task for the protocol drivers that are associated with this offset. See writethru below for more information. Accepted values are 1 for attempt writethrough and 0 for do not attempt writethrough. Default: 0 (do not attempt writethrough)

3.3.2 Write thru

The CAS Gateway can be configured to read values from many other protocols such as BACnet IP, Modbus, etc. These values are stored in the CAS Gateway internal database. These values can then be read via the HTTP API using the act get value command. Some protocol such as BACnet IP or Modbus TCP/RTU support writing as well as reading.

If the CAS Gateway receives a set_value message, with the writethru flag, to an offset that has been configured to read from protocol that supports writes. The CAS gateway will attempt to write the value to the end device via the supported protocol.

Not all protocols support writes, or this write thru command. Below is a list of commonly used protocols that do support this feature. Please contact Chipkin support to see if a protocol is supported.

- BACnet IP
- Modbus RTU
- Modbus TCP
- IDP600
- Eyedro
- MQTT
- Azure

3.3.3 Example - Set a specific data point

In this example a value will be written to a specified data point and the results will be returned via JSON.

Request:

```
GET /bin/xml/?act=set_value&id=1&value=1000.3
```

Response:

```
{
  "query": {
    "act": "set_value",
    "id": "1",
    "value": "1000.3"
  },
  "response": {
    "data": [{
      "data": "OK"
    }]
  }
}
```

3.3.4 Example - Set a multiple data point

In this example a value will be written to multiple data point and the results will be returned via JSON. This request will set the first 5 data points of the da_data table

Request:

```
POST /bin/xml/?act=set_value

{"set_value": [{"id": "1","value": "101.9"}, {"id": "2","value": "102.9"}, {"id": "3","value": "103.9"}, {"id": "4","value": "104.9"}, {"id": "5","value": "105.9"}]}
```

Response:

```

{
  "query": {
    "act": "set_value",
    "post": "{\"set_value\": [{\"id\": \"1\", \"value\": \"101.9\"}, {\"id\": \"2\", \"value\": \"102.9\"}, {\"id\": \"3\", \"value\": \"103.9\"}, {\"id\": \"4\", \"value\": \"104.9\"}, {\"id\": \"5\", \"value\": \"105.9\"}]}"
  },
  "response": {
    "data": [{
      "data": "OK"
    }]
  }
}
    
```

3.4 Act Set

Sets an internal configuration setting on the CAS Gateway or starts a system task.

Endpoint: /bin/json/?act=set

3.4.1 Parameters

- **current_time [Optional]** – Sets the current time on the CAS gateway. The time is in Unix EPOCH time or seconds since 1970.
- **save [Optional]** – Saves the database to disk. The only accepted value is 1.
- **reset [Optional]** – Reboots the CAS Gateway as if the power was cycled. The only accepted value is 1.
- **delete_database [Optional]** – Deletes the existing database and the configuration. The only accepted value is 1. **Note:** This command deletes data and cannot be recovered from, use with caution.
- **delete_file [Optional]** – Deletes a file from the file system. The path to the file is used as the value. **Note:** This command deletes data and cannot be recovered from, use with caution.
- **clear_history [Optional]** – Removes all the values stored in the history database. The only accepted value is 1. **Note:** This command deletes data and cannot be recovered from, use with caution.

- **delete_history_files** – When combined with the clear_histroy flag. The archived history files will also be deleted. The only accepted value is 1. **Note:** This command deletes data and cannot be recovered from, use with caution.

3.4.2 Example – Set the current time

Sets the current time of the CAS Gateway. This can be useful if the CAS Gateway does not have access to a NTP server.

Request:

```
GET /bin/json/?act=set&current_time=1234567890
```

Response:

```
{
  "query": {
    "act": "set",
    "current_time": "1234567890"
  },
  "response": {
    "current_time": "ok"
  }
}
```

3.5 Act Get

Get system information from the CAS Gateway.

Endpoint: /bin/json/?act=get

3.5.1 Parameters

- **type [Required]** – The type of information to be retrieved. Currently only a value of “stats” is accepted.

3.5.2 Example – Get the current stats

This command will get the current stats from the CAS Gateway as JSON.

Request:

```
GET /bin/json/?act=get&type=stats
```

Response:

```
{
  "query": {
    "act": "get",
    "type": "stats"
  },
  "response": {
    "stats": {
      "BACnetIP.init.success": "1",
      "Http.requests": "10",
      "MQTT.init.success": "1",
      "Memory.Free": "3251952",
      "Memory.Total": "4398568",
      "Memory.Used": "1146616",
      "ModbusRTU.init.success": "1",
      "ModbusRTU.recv.bytes": "1",
      "ModbusTCP.init.success": "1",
      "ModbusTCPClient.System.errors": "6",
      "ModbusTCPClient.init.success": "1",
      "ModbusTCPClient.task.1.read": "6",
      "ModbusTCPClient.task.1.read.error.fail": "6",
      "ModbusTCPClient.task.1.read.status": "Fail",
      "MultipleTCP.max.connections": "10",
      "Serial0.bytes.recv": "1",
      "Serial0.connection.count": "1",
      "System.drivers.success": "5",
      "System.freespace": "921",
      "System.http.drivers": "19",
      "System.saveDB.success": "1",
      "System.start_up": "1521156253",
      "TCP.connection.failed": "6"
    }
  }
}
```


3.6 Act Info

Get the driver and subsystem version information.

Endpoint: /bin/json/?act=info

3.6.1 Example – Get the current drivers stats

This command will get the current driver versions.

Request:

```
GET /bin/json?act=info
```

Response:

```
{
  "query": {
    "act": "info"
  },
  "response": [
    {
      "application_version": "0.780",
      "core_driver_version": "0.32",
      "core_http_version": "0.48",
      "title": "header"
    },
    {"data": "1234569130", "title": "current_time"},
    {"data": "943664", "title": "available_disk_space"},
    {"data": "975536", "title": "total_disk_space"},
    {"title": "BACnetIP", "version": "0.93"},
    {"title": "ModbusRTU", "version": "0.19"},
    {"title": "ModbusTCP", "version": "0.21"},
    {"title": "ModbusTCPClient", "version": "0.11"},
    {"title": "MQTT", "version": "0.12"},
    {"title": "HttpSystem", "version": "0.66"},
    {"title": "HttpXML", "version": "0.09"},
    {"title": "HttpCSV", "version": "0.08"},
    {"title": "HttpJSON", "version": "0.08"}
  ]
}
```



THANK YOU

Thanks for choosing Chipkin's protocol gateways, data clients and integration services to meet your building and industrial automation requirements!

Chipkin Automation Systems™ (Chipkin) is a building and industrial automation protocol expert. We develop, configure, install and support gateways (protocol converters), data loggers and remote monitor and controlling applications.

Founded in October 2000, Chipkin provides expert solutions for converting BACnet®, Modbus®, and Lonworks®—to name just a few—and enabling interfaces for HVAC, fire, siren, intercom, lighting, transportation and fuel systems. The high-quality products we offer (including those from other vendors) interface with Simplex™, Notifier™, McQuay™, GE™ and many others—so you can rest assured that we will select the most appropriate solution for your application.

With Chipkin you are buying a solution. Our configuration expertise in this field combined with free BACnet tools and other tools ensure your success; and our customer support via phone, email and remote desktop tools means that we're there when you need us. Chipkin is a small responsive company, and we live or die by the quality of our service—and with offices in two-time zones—we can provide support when you need it. Give us a call now!

Sales and Customer Service

Toll free: 1-866-383-1657

Email: salesgroup1@chipkin.com