C8P UHF RDR G2 – REST API and MQTT description

UHF RFID READER

(v1.6)

CONTENT

Version 1.6
November 2022

RoHS

Callidus C8P UHF RDR supports REST API based interface for directly getting data from device.

Request are done via an HTTP GET or POST request. These requests are sent to corresponding RFID device IP address and port 80, where they are processed by embedded web server.

For all commands is necessary to have HASH key, that is used for verification. This HASH key is obtained as an answer after authorization and is valid 10 minutes after last command. If no command is sent in that interval, new authorization has to be performed.

All responses to commands are JSON objects.

Authorization

HTTP method:	POST
Command:	/douserauth.json
Request body:	{"username":"user","password":"123456"}

Successful response: {"status":"ok", "connectedClients": 2, "hash":"1234ABCD"}
Response in case of fail: {"status":"error", "message":"incorrect login", "hash":""}
Response in case of too many clients already logged in:
{"status":"error", "message":"too many logins", "connectedClients":3,"hash":""}

Login details

Username for user with limited rights:	**user**
Default password:	Call + last 6 numbers of MAC
Example for MAC '1A:2B:3C:4D:5E:6F':	Call4D5E6F
Username for admin with full rights:	**admin**
Default password:	Call + last 6 numbers of MAC with reverted double digits
Example for MAC '1A:2B:3C:4D:5E:6F':	CallD4E5F6

It is strongly recommended to change default passwords!

Log out

HTTP method:	GET
Command:	/douserlogout.json?hash=XXXXXX
Request body:	none

Successful response: {"status": "ok", "message": "client logged out"}
Response in case of fail: {"status": "error", "message": "invalid authentication"}

Callidus C8P UHF RDR also supports MQTT, which is very simple and unpretentious protocol built over TCP/IP. Designed for devices even with unreliable networks, narrow bandwidth or high latency.

Device publishing data to corresponding MQTT server (broker) specified by IP address and port, usually 1883.

Authentication is part of the transport and application-level security in MQTT.

All payloads are JSON objects.

Messages and topics

Device publishing 3 types of messages:

- **Status** published every 30 seconds
- **Event** published immediately when event happened
- **Answer** published after processing required command

Topic structure: topic_prefix/device_identification/message_type/additional_data

Status example: monitoring/rfid/C8PURDRG2_123456ABCDDE_000000123456/status
Event example: monitoring/rfid/C8PURDRG2_123456ABCDDE_000000123456/event/info/inventoryReading/start
Answer example: monitoring/rfid/C8PURDRG2_123456ABCDDE_000000123456/answer/get/control/time

Prefix in topic can be changed in MQTT settings.

In order to receive messages, client has to subscribe to the topics specified above.
Below is a cheat sheet to subscribe topics to getting started quickly:

MEANING	TOPIC
subscribe all topics from all devices	#
subscribe all topics from all RFID devices	monitoring/rfid/#
subscribe only status from all RFID devices	monitoring/rfid/+/status
subscribe all topics for specific RFID device	monitoring/rfid/C8PURDRG2_123456ABCDDE_000000123456/#
subscribe only events for specific RFID device	monitoring/rfid/C8PURDRG2_123456ABCDDE_000000123456/events/#

Sending commands

Sending command is very simple. Device subscribe on “command” topic and based on the rest of the topic will do appropriate action. Get and set is equivalent for GET and POST method in API. Command name is also the same as command for API, but does **not** expect “.json” extension nor parameters in topic, such as hash.
Payload data are always JSON objects. Answers have the same format as for API.

Topic structure: topic_prefix/device_identification/command/get_or_set/command_name

Get time example: monitoring/rfid/C8PURDRG2_123456ABCDDE_000000123456/command/get/control/time
Set GPIO config example: monitoring/rfid/C8PURDRG2_123456ABCDDE_000000123456/command/set/config/GPIO
Payload: {"in1":[1,0],"in2":[2,1],"out1":[1],"out2":[2]}
Start read example: monitoring/rfid/C8PURDRG2_123456ABCDDE_000000123456/command/set/control/tagRead
Payload: {"reading":true}

Changing configuration over MQTT has to be allowed in MQTT settings, otherwise command changing configuration will have no effect and answer will **not** be sent.

Tree list of MQTT events

	topic
Information	
Tags deleted from buffer	**/event/info/tagBuffer/deletedAll**
Reading started	**/event/info/inventoryReading/start**
Reading stopped	**/event/info/inventoryReading/stop**
GPI falling	**/event/info/GPI/GPIportNumber*/falling**
GPI rising	**/event/info/GPI/GPIportNumber*/rising**
GPO closed	**/event/info/GPO/GPOportNumber*/closed**
GPO opened	**/event/info/GPO/GPOportNumber*/opened**
Incoming people	**/event/info/peopleCounter/PCnumber*/incoming**
Outgoing people	**/event/info/peopleCounter/PCnumber*/outgoing**
Alarm	
Outgoing tag alarm	**/event/alarm/antennaPort/portNumber*/tagOutgoing**
Incoming tag alarm	**/event/alarm/antennaPort/portNumber*/tagIncoming**
False tag alarm	**/event/alarm/antennaPort/portNumber*/tagFalse**
Tag in field	**/event/alarm/antennaPort/portNumber*/tagInField**
Wrong way incoming	**/event/alarm/wrongWay/PCnumber*/incoming**
Wrong way outgoing	**/event/alarm/wrongWay/PCnumber*/outgoing**
GPI alarm trigger	**/event/alarm/GPI/GPIportNumber***

* Instead of GPIportNumber, GPOportNumber, PCnumber and portNumber are actual numbers, indicating which port triggered event.

Message payload

Payload for status and answers is the same as response body for API.
Payload for events contains actual time when the event happened and information depending on that event.
For more information, see below. Names and values have the same meaning as for API.

Event payloads

Event: tags deleted from buffer, reading started, reading stopped
Topic: monitoring/rfid/C8PURDRG2_123456ABCDDE_000000123456/event/info/tagBuffer/deletedAll
monitoring/rfid/C8PURDRG2_123456ABCDDE_000000123456/event/info/inventoryReading/start
monitoring/rfid/C8PURDRG2_123456ABCDDE_000000123456/event/info/inventoryReading/stop
Event payload:
{"status":"ok","totalTagsRead":0,"uniqueTagsRead":0,"freeSpace":2000,"totalRows":0,"readTime":0.000,"tagsReadingSpeed":0.00,"actualTime":[13,24,40,7,11,22]}

Event: GPI falling, GPI rising, GPO closed, GPO opened
Topic: monitoring/rfid/C8PURDRG2_123456ABCDDE_000000123456/event/info/GPI/1/falling
monitoring/rfid/C8PURDRG2_123456ABCDDE_000000123456/event/info/GPI/2/rising
monitoring/rfid/C8PURDRG2_123456ABCDDE_000000123456/event/info/GPO/1/closed
monitoring/rfid/C8PURDRG2_123456ABCDDE_000000123456/event/info/GPO/2/opened
Event payload:
{"status":"ok","GPIO":{"in1State":0,"in2State":1,"out1State":0,"out2State":0},"actualTime":[13,27,24,7,11,22]}

Event: incoming people, outgoing people
Topic: monitoring/rfid/C8PURDRG2_123456ABCDDE_000000123456/event/info/peopleCounter/1/incoming
monitoring/rfid/C8PURDRG2_123456ABCDDE_000000123456/event/info/peopleCounter/2/outgoing
Event payload:
{"status":"ok","peopleCounters":{"counter1":[1,7,0],"counter2":[0,0,0],"states1":[1,1],"states2":[1,1]},"actualTime":[13,29,7,7,11,22]}

Event: outgoing tag alarm, incoming tag alarm, false tag alarm, tag in field
Topic: monitoring/rfid/C8PURDRG2_123456ABCDDE_000000123456/event/alarm/antennaPort/1/tagOutgoing
monitoring/rfid/C8PURDRG2_123456ABCDDE_000000123456/event/alarm/antennaPort/4/tagIncoming
monitoring/rfid/C8PURDRG2_123456ABCDDE_000000123456/event/alarm/antennaPort/7/tagFalse
monitoring/rfid/C8PURDRG2_123456ABCDDE_000000123456/event/alarm/antennaPort/8/tagInField
Event payload:
{"status":"ok","tag":{"epc":"1234","tid":"CDEF","aisle":1,"port":1,"count":1,"firstTimestamp":"20221107T133237","lastTimestamp":"20221107T133237","lastRSSI":63,"maxRSSI":63,"direction":3,"alarmTriggered":true,"alarmCurrently":true},"actualTime":[13,32,37,7,11,22]}

Event: wrong way incoming, wrong way outgoing
Topic: monitoring/rfid/C8PURDRG2_123456ABCDDE_000000123456/event/alarm/wrongWay/1/incoming
monitoring/rfid/C8PURDRG2_123456ABCDDE_000000123456/event/alarm/wrongWay/2/outgoing
Event payload:
{"status":"ok","peopleCounters":{"counter1":[1,64,2],"counter2":[0,0,0],"states1":[1,1],"states2":[1,1]},"actualTime":[13,30,40,7,11,22]}

Event: GPI alarm trigger
Topic: monitoring/rfid/C8PURDRG2_123456ABCDDE_000000123456/event/alarm/GPI/1
monitoring/rfid/C8PURDRG2_123456ABCDDE_000000123456/event/alarm/GPI/2
Event payload:
{"status":"ok","GPIO":{"in1State":0,"in2State":0,"out1State":1,"out2State":1},"alarmCounters":{"gpi1":0,"gpi2":1},"actualTime":[13,29,56,7,11,22]}

Response: {"status":"error", "message":"insufficient user rights."}
{"status":"error", "message":"invalid authentication"}
Meaning: User does NOT have sufficient rights or is not logged at all.

Response: {"status":"error", "message":"unknown command."}
Meaning: If unknown command was received.

Response: {"status":"error", "message":"wrong param. 'ParameterName', invalid format."}
Meaning: Appears when parameter was sent with invalid format.

Response: {"status":"error", "message":"wrong param. 'ParameterName', value is too low."}
{"status":"error", "message":"wrong param. 'ParameterName', value is too high."}
{"status":"error", "message":"wrong param. 'ParameterName', value is invalid."}
Meaning: Occurs when value of parameter is lower than minimum or higher than maximum.

Response: {"status":"error", "message":"wrong param. 'ParameterName', param. name not found."}
Meaning: Message is missing required parameter.

Response: {"status":"error","message":"RTOS, radio busy"}
Meaning: Radio is already doing some work.
If inventory reading is running, stop it, otherwise wait until current work is done and try again later.

Response: {"status":"error", "message":"TM API, code ErrorCode."}
Meaning: There was some error with ThingMagic API, where
ErrorCode refer to „Mercury API programmer guide“.

Response: {"status":"error", "message":"RTOS, code ErrorCode."}
{"status":"error", "message":"HAL, code ErrorCode."}
Meaning: If any of these errors appears, please contact us with screenshot of the error and steps that lead to this error.

Reading of parameters (GET method)

Configuration	**command**	
Reader		
Networking	**/config/network.json**	
MQTT	**/config/MQTT.json**	
Get time settings	**/config/time.json**	
Antenna ports	**/config/antennaPort.json**	
Alarms	**/config/alarms.json**	
GPIO	**/config/GPIO.json**	
People counters	**/config/peopleCounters.json**	
Gen2	**/config/gen2.json**	
Radio	**/config/radio.json**	
Region	**/config/region.json**	
Inventory settings	**/config/inventorySettings.json**	
POS	**/config/pos.json**	
Diagnostic		
Reader		
~~Status~~	~~**/diag/status.json**~~	obsolete
Version	**/diag/versionReader.json**	
Radio		
Version	**/diag/versionRadio.json**	
Voltage Standing Wave Ratio	**/diag/VSWR.json**	
Controls and others		
Status	**/status.json**	
Get current time	**/control/time.json**	
Get tags statistics	**/control/tagStatistics.json**	
Get tag buffer	**/control/tagBuffer.json**	
Get reading state	**/control/tagRead.json**	
Get GPIO states	**/control/GPIO.json**	
Get alarm counters	**/control/alarmCounters.json**	
Get people counters	**/control/peopleCounters.json**	

Obsolete commands will be removed in future versions!

Changing of parameters (POST method)

	command
Configuration	
Passwords	/config/changePasswords.json
Networking	/config/network.json
MQTT	/config/MQTT.json
Set time	/config/time.json
Antenna ports	/config/antennaPort.json
Reset to default	/config/factoryReset.json
FW update	/config/fwUpdateReader.json
Alarms	/config/alarms.json
GPIO	/config/GPIO.json
People counters	/config/peopleCounters.json
Gen2	/config/gen2.json
Radio	/config/radio.json
Region	/config/region.json
Inventory settings	/config/inventorySettings.json
POS	/config/pos.json
Controls and others	
Test status LED	/control/LED/readerStatusLED.json
Test read LED	/control/LED/radioStatusLED.json
Test internal buzzer	/control/buzzer/internal.json
Test external buzzer	/control/buzzer/external.json
Start read tags	/control/tagRead.json
Delete tag buffer	/control/tagBufferDelete.json
Inspect tag	/control/tagInspector.json
Write new EPC	/control/tagWriteEPC.json
Write new data	/control/tagWriteData.json
Write locks	/control/tagLocks.json
Kill tag	/control/tagKill.json
Trigger predefined alarm	/control/alarm.json
Trigger custom alarm	/control/alarmTrigger.json
Trigger GPO	/control/setGPO.json
Reset alarm counters	/control/alarmCountersReset.json
Reset people counters	/control/peopleCountersReset.json

Obsolete commands will be removed in future versions!

Networking

Method: GET
Command: /config/network.json?hash=XXXXXX
Request body: none

Response:
{"status":"ok","netbiosName":"RFID-READER","dhcp":true,"ip":"192.168.0.100","netmask":"255.255.255.0","gw":"192.168.0.1","pridns":"192.168.0.1","port1":"connected","port2":"disconnected","connectedClients":1}

Name of field	Type of value	Possible values	Description
netbiosName	String	-	Name of device
dhcp	Boolean	true/false	DHCP enabled
ip	String	IP address	IP address of device
netmask	String	IP address	Subnet mask
gw	String	IP address	Gateway
pridns	String	IP address	Primary DNS
portX	String	connected/disconnected	Indicates if LAN ports are connected (1-2)
connectedClients	Integer	-	Currently connected clients to the (API) reader

MQTT

Method: GET
Command: /config/MQTT.json?hash=XXXXXX
Request body: none

Response:
{"status":"ok","enabled":false,"allowRemoteConfig":true,"server":"broker.mqttdashboard.com","port":1883,"username":"user","password":"pass","topicPrefix":"monitoring/rfid","connection":"not connected"}

Name of field	Type of value	Possible values	Description
enabled	Boolean	true/false	MQTT enabled
allowRemoteConfig	Boolean	true/false	Allow configure reader over MQTT
server	String	-	MQTT broker
port	Integer	-	Port (usually 1883)
username	String	-	Username
password	String	-	Password
topicPrefix	String	-	Topic prefix
connection	String	connecting/ connected/ not connected	MQTT connection state

Get time settings

Method: GET
Command: /config/time.json?hash=XXXXXX
Request body: none

Response:
{"status":"ok","actualTime":[15,37,25,15,12,20],"timezoneDiff":60,"DST":{"enabled":true,"begin":{"month":3,"time":2,"date":28},"end":{"month":10,"time":3,"weekday":1,"weekOfMonth":6}},"ntpSync":true,"ntp":"pool.ntp.org","ntpSyncInterval":1,"driftCorrection":0}

Name of field	Type of value	Possible values	Description
actualTime	Array of integers	-	Current time (hours, minutes, seconds, day, month, year)
timezoneDiff	Integer	-720 ~ 720	Time zone difference in minutes
DST	Object	-	Daylight saving time (DST) settings
enabled	Boolean	true/false	Enabled or disabled DST
begin/end	Object	-	Beginning and end of DST
month	Integer	1 ~ 12	Month where 1 - January and 12 - December
time	Integer	0 ~ 23	Hour of the beginning and end of DST
date	Integer	1 ~ 31	Day of the beginning and end of DST
weekday	Integer	1 ~ 7	Day where 1 - Monday and 7 - Sunday
weekOfMonth	Integer	1 ~ 6	Week where 1 - first and 5 - fifth, 6 - last week
ntpSync	Boolean	true/false	Enable or disable NTP synchronization
ntp	String	-	NTP server (name or IP)
ntpSyncInterval	Integer	1 ~ 65535	Synchronization interval in minutes
driftCorrection	Integer	-127 ~ 128	Time correction per day $\Delta t = n \cdot \frac{86400}{32 \cdot 32768}$ s/day

Get current time

Method: GET
Command: /control/time.json?hash=XXXXXX
Request body: none

Response:
{"status":"ok","actualTime":[15,37,25,15,12,20],"timezoneDiff":60,"uptime":8507}

Name of field	Type of value	Possible values	Description
actualTime	Array of integers	-	Current time (hours, minutes, seconds, day, month, year)
timezoneDiff	Integer	-720 ~ 720	Time zone difference in minutes
uptime	Integer	-	Time of device from power on in seconds

Antenna ports

Method: GET
Command: /config/antennaPort.json?hash=XXXXXX
Request body: none

Response:
{"status":"ok","port1":false,"port2":true,"port3":false,"port4":true}

Name of field	Type of value	Possible values	Description
portX	Boolean	true/false	Indicates if antenna port (1-8) is enabled or disabled

GEN2

Method: GET
Command: /config/gen2.json?hash=XXXXXX
Request body: none

Response:
{"status":"ok","accessPassword":"#1234CDEF","BLF":320,"tari":6.25,"tagEncoding":"FM0","session":"S2","target":"AB","q":7,"sendSelectEveryQuery":false}

Name of field	Type of value	Possible values	Description
accessPassword	String	#00000000 ~ #FFFFFFFF	Access password for manipulation with tags. 8 hexadecimal characters.
BLF	Integer	250/320/640	Backscatter link frequency in kHz
tari	Float	25/12.5/6.25	Tari value in µs
tagEncoding	String	FM0/M2/M4/M8	Tag encoding method
session	String	S0/S1/S2/S3	Session flag state
target	String	A/B/AB/BA	Target value
q	Integer	0 ~ 15	Q value
sendSelectEveryQuery	Boolean	true/false	Disable or enable sending select command every query

For more information about GEN2 settings, refer to „Mercury API programmer guide“.

Radio

Method: GET
Command: /config/radio.json?hash=XXXXXX
Request body: none

Response:
{"status":"ok","readingPower":20,"writingPower":20,"readingPowerPerPort":{"port1":25,"port2":30},"writingPowerPerPort":{},"temperature":34}

Name of field	Type of value	Possible values	Description
readingPower	Integer	-10 ~ 30	Used global power when reading in dBm
writingPower	Integer	-10 ~ 30	Used global power when writing in dBm
readingPowerPerPort	Object	-	Overwrites reading global power for individual ports if specified in dBm
writingPowerPerPort	Object	-	Overwrites writing global power for individual ports if specified in dBm
portX	Integer	-10 ~ 30	Power for individual ports (1-8)
temperature	Integer	-	Temperature of radio

Region

Method: GET
Command: /config/region.json?hash=XXXXXX
Request body: none

Response:
{"status":"ok","id":"North America","supportedRegions":["North America"],"hopTable":[918250, 913250],"hopTime":250}

Name of field	Type of value	Possible values	Description
id	String	-	Currently selected region
supportedRegions	Array of strings	-	Array of supported regions
hopTable	Array of integers	-	Array of frequencies
hopTime	Integer	1 ~ 4000	Time for each frequency in ms

Get reading state

Method: GET
Command: /control/tagRead.json?hash=XXXXXX
Request body: none

Response:
{"status":"ok","reading":true}

Name of field	Type of value	Possible values	Description
reading	Boolean	true/false	Indicates if reader is currently reading

Inventory settings

Method: GET
Command: /config/inventorySettings.json?hash=XXXXXX
Request body: none

Response:

{"status":"ok","antPorts":[1,4],"targetBank":["EPC","TID"],"readingOnBoot":false,"autostartReading":0,
"fullBuffer":"overwrite","filters":[{"bank":"EPC","offsetInBits":32,"lengthInBits":8,"mask":"E2","action":"ON_N_OFF","t
arget":"SL","invert":false}]}

Name of field	Type of value	Possible values	Description
antPorts	Array of integers	-	Enabled antenna ports
targetBank	Array of strings	EPC/TID	Banks to inventory
readingOnBoot	Boolean	true/false	Start inventorying automatically on boot
autostartReading	Integer	0 ~ 86400	Start inventorying automatically after X minutes span if not running
fullBuffer	String	stop/overwrite	Once the inventory buffer is full, new tags will be ignored or overwrite the oldest one
filters	Array of objects	-	Radio HW filters
bank	String	EPC/TID/user/ EPCtruncate/EPClength	Memory bank to apply filter on
offsetInBits	Integer	0 ~ 512	Bit offset starting at the bank position
lengthInBits	Integer	0 ~ 512	Bit length of the comparison
mask	String	-	Pattern to be matched
action	String	table below	Set, reset, flip or leave alone the flag
target	String	SL/S0/S1/S2/S3	Which flag should be changed
invert	Boolean	true/false	Invert the selection

Action	Tag matching	Tag not matching	Behavior if flag is "SL" (selected)
ON_N_OFF	Assert target	De-assert target	Matching tags will respond, and non-matching tags will NOT respond
ON_N_NOP	Assert target	Do nothing	Matching tags will respond, and non-matching tags will respond based on previous SL flag status from last Action
NOP_N_OFF	Do nothing	De-assert target	Matching tags will respond based on previous SL flag status from last Action and non-matching tags will NOT respond
NEG_N_NOP	Negate target	Do nothing	Previous SL flag will be nullified for matching tags and non-matching tags will respond based on previous SL flag status from last Action
OFF_N_ON	De-assert target	Assert target	Matching tags will NOT respond, and non-matching tags will respond
OFF_N_NOP	De-assert target	Do nothing	Matching tags will NOT respond, and non-matching tags will respond based on previous SL flag status from last Action
NOP_N_ON	Do nothing	Assert target	Matching tags will respond based on previous SL flag status from last Action and non-matching tags will respond
NOP_N_NEG	Do nothing	Negate target	Matching tags will respond based on previous SL flag status from last Action and previous SL flag will be nullified for nonmatching tags

Status (obsolete)

Method: GET
Command: /diag/status.json?hash=XXXXXX
Request body: none

Response:
{"status":"ok","bus5v":4.95,"bus12v":11.85,"bckpBatt":3.03,"rtc":true,"uart":true,"i2cMem":true,"spiMem":false}

Name of field	Type of value	Possible values	Description
bus5v	Float	-	Voltage of 5V bus
bus12v	Float	-	Voltage of 12V bus
bckpBatt	Float	-	Voltage of backup battery
rtc	Boolean	true/false	Indicates if RTC works correctly
uart	Boolean	true/false	Indicates if UART works correctly
i2cMem	Boolean	true/false	Indicates if memory works correctly over I2C
spiMem	Boolean	true/false	Indicates if memory works correctly over SPI

Reader version

Method: GET
Command: /diag/versionReader.json?hash=XXXXXX
Request body: none

Response:
{"status":"ok","sn":"000000123456","mac":"12:34:56:AB:CD:DE","hwName":"C8PURDR","hwVer":"G2","maxPorts":8,"fwVer":"0.01","fwBuildDate":"Oct 1 2020 12:34:56"}

Name of field	Type of value	Possible values	Description
sn	String	-	Serial number
mac	String	-	MAC address
hwName	String	-	Reader name
hwVer	String	-	Hardware version
maxPorts	Integer	-	Total ports
fwVer	String	-	Firmware version
fwBuildDate	String	-	Date of Firmware build

Module version

Method: GET
Command: /diag/versionRadio.json?hash=XXXXXX
Request body: none

Response:
{"status":"ok","hardware":"20.00.00.01","model":"M6e micro","productGroup":"Embedded Reader","serial":"123456789abc","software":"01.12.2a.bc.de","supportedProtocols":["GEN2"]}

Name of field	Type of value	Possible values	Description
hardware	String	-	Version identifier for the reader hardware
model	String	-	Model identifier for the reader hardware
productGroup	String	-	Product group type
serial	String	-	Serial number of the reader
software	String	-	Version identifier for the reader software
supportedProtocols	Array of strings	-	Array of supported protocols

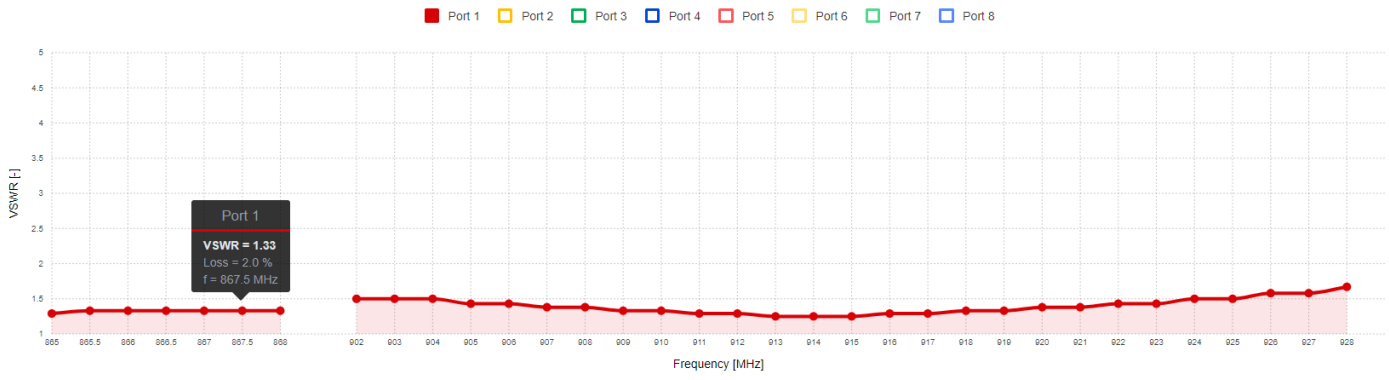
Voltage Standing Wave Ratio

Method: GET
Command: /diag/VSWR.json?hash=XXXXXX
Request body: none

Response:
{"status":"ok","frequencykHz":[865000,865500,…,928000],"VSWRport1":[1.67,1.67,…,1.58],…,"VSWRport8":[5.85,5.85,…,17.39]}

Name of field	Type of value	Possible values	Description
frequencykHz	Array of integers	-	X-axis – specify in which frequencies was VSWR measured in kHz
VSWRportX	Array of integers	-	Y-axis per port (1-8) – VSWR values for each frequency specified above

Measurement can take up to 10s. Make sure that timeout is sufficient.
VSWR value can be -1, which indicates exceeding the measurable value.
To visualize such value, maximum is used.



Visualization of measured data.

Status

Method: GET
Command: /status.json?hash=XXXXXX
Request body: none

Response:

{"status":"ok","lwt":"on-line","versionReader":{"sn":"000000123456","mac":"12:34:56:AB:CD:DE","hwName":"C8PURDR","hwVer":"G2","maxPorts":8,"fwVer":"0.01","fwBuildDate":"Oct 1 2020 12:34:56"},"network":{"netbiosName":"RFID-READER","dhcp":true,"ip":"192.168.0.100","netmask":"255.255.255.0","gw":"192.168.0.1","pridns":"192.168.0.1","port1":"connected","port2":"disconnected","connectedClients":1},"MQTT":{"enabled":false,"allowRemoteConfig":true,"server":"broker.mqttdashboard.com","port":1883,"username":"user","password":"pass","topicPrefix":"monitoring/rfid","connection":"not connected"},"time":{"actualTime":[15,37,25,15,12,20],"timezoneDiff":60, "uptime":69667},"hwStatus":{"bus5v":5.05,"bus12v":11.81,"bckpBatt":3.25,"radio":true,"rtc":true,"uart":true,"i2cMem":true,"spiMem":true,"spiPsramMem":true},"tagRead":{"reading":false},"tagStatistics":{"totalTagsRead":46,"uniqueTagsRead":2,"freeSpace":1998,"totalRows":4,"readTime":0.000,"tagsReadingSpeed":0.00,"tagOpSuccess":0,"tagOpFailures":0},"radio":{"readingPower":20,"writingPower":20,"readingPowerPerPort":{"port1":25,"port2":30},"writingPowerPerPort":{},"temperature":33},"alarmCounters":{"aisle":{"no1":{"tagOutgoingRegular":0,"tagIncoming":0,"tagFalse":0},"no2":{"tagOutgoingRegular":0,"tagIncoming":0,"tagFalse":0}},"gpi1":0,"gpi2":0,"hwError":0},"peopleCounters":{"counter1":[0,0,0],"counter2":[0,0,0],"states1":[0,0],"states2":[0,0]},"GPIO":{"in1State":0,"in2State":0,"out1State":0,"out2State":0},"errCnt":{"cntWdtReset":0,"cntModComErr":0,"cntSpiMemErr":0,"cntI2cMemErr":0,"cntHardFaultErr":0,"cntMemAllocErr":0,"cntStackOverflowErr":0,"cntHalErrorHandler":0,"cntAssertErr":0,"cntSpiBusErr":0,"cntSpiPsramErr":0,"cntFsActOpenFile":1,"cntFsMaxOpenFile":4,"dynamic":{"freeHeapSize":82640,"minimumEverFreeHeapSize":39768}}}

{"status":"ok","lwt":"off-line","versionReader":{"sn":"000000123456","mac":"12:34:56:AB:CD:DE","hwName":"C8PURDR","hwVer":"G2","maxPorts":8,"fwVer":"0.01","fwBuildDate":"Oct 1 2020 12:34:56"}}

Name of field	Type of value	Possible values	Description
lwt	String	on-line/off-line/ reconnected	Last will and testament. API always return on-line.
versionReader	Object	-	Reader version
network	Object	-	Networking settings and status
MQTT	Object	-	MQTT settings and connection state
time	Object	-	Current time
hwStatus	Object	-	Hardware status
bus5v	Float	-	Voltage of 5V bus
bus12v	Float	-	Voltage of 12V bus
bckpBatt	Float	-	Voltage of backup battery
radio	Boolean	true/false	Indicates if RFID radio works correctly
rtc	Boolean	true/false	Indicates if RTC works correctly
uart	Boolean	true/false	Indicates if UART works correctly
i2cMem	Boolean	true/false	Indicates if memory works correctly over I2C
spiMem	Boolean	true/false	Indicates if memory works correctly over SPI
spiPsramMem	Boolean	true/false	Indicates if PS memory works correctly over SPI
tagRead	Object	-	Reading state
tagStatistics	Object	-	Tag statistics
radio	Object	-	Radio settings and temperature
alarmCounters	Object	-	Alarm counters
peopleCounters	Object	-	People counters
GPIO	Object	-	Current GPIO states
errCnt	Object	-	Developer counters

Get tag statistics

Method: GET
Command: /control/tagStatistics.json?hash=XXXXXX
Request body: none

Response:
{"status":"ok","totalTagsRead":10,"uniqueTagsRead":2,"freeSpace":998,"totalRows":2,"readTime":12.321,"tagsReadingSpeed":50.15,"tagOpSuccess":0,"tagOpFailures":0}

Name of field	Type of value	Possible values	Description
totalTagsRead	Integer	-	The total number of read tags
uniqueTagsRead	Integer	-	The total number of unique tags in buffer
totalRows	Integer	-	Total rows of read tags
freeSpace	Integer	-	Free space in buffer
readTime	Float	-	Time since reading started in seconds
tagsReadingSpeed	Float	-	Reading rate
tagOpSuccess	Integer	-	Successful operations with tags
tagOpFailures	Integer	-	Failed operations with tags

Get tag buffer

Method: GET
Command: /control/tagBuffer.json?hash=XXXXXX
Request body: none

Response:
{"status":"ok","totalTagsRead":10,"uniqueTagsRead":1,"freeSpace":999,"readTime":1.568,"tagsReadingSpeed":142.59,"tags":[{"epc":"1234","tid":"CDEF","aisle":1,"port":1,"count":10,"firstTimestamp":"20201214T073635","lastTimestamp":"20201214T073644","lastRSSI":65,"maxRSSI":60,"direction":3,"alarmTriggered":true,"alarmCurrently":false}]}

Name of field	Type of value	Possible values	Description
tags	Array of objects	-	Array of all tags in buffer
epc	String	-	Tag‘s EPC
tid	String	-	Tag’s TID
aisle	Integer	-	Aisle where tag was seen
port	Integer	-	Antenna that read tag
count	Integer	-	Number of reads
firstTimestamp	Datetime	-	Timestamp when the tag was first read in ISO 8601 format: YYYYMMDDThhmmss
lastTimestamp	Datetime	-	Timestamp when tag was last read in ISO 8601 format: YYYYMMDDThhmmss
lastRSSI	Integer	-	Last received signal strength indication in dBm
maxRSSI	Integer	-	Maximal received signal strength indication in dBm
direction	Integer	0 ~ 3	0 – evaluating, 1 – incoming, 2 – outgoing, 3 – unknown
alarmTriggered	Boolean	true/false	Tag triggered alarm at least once
alarmCurrently	Boolean	true/false	Tag is currently in field and indicates alarm

Undescribed items have the same meaning as in Get tag statistics.

Alarms

Method: GET
Command: /config/alarms.json?hash=XXXXXX
Request body: none

Response:

{"status":"ok","tagReadOutgoingRegularLight":[100, 1, 100, 0, 0, 0, 10, 0, 0],"tagReadOutgoingRegularSound":[100, 100, 100, 100, 0, 0, 3, 0, 2, 2],"tagReadIncomingLight":[100, 1, 100, 0, 0, 0, 10, 0, 0],"tagReadIncomingSound":[100, 100, 100, 100, 0, 0, 3, 0, 2, 2],"tagReadFalseLight":[100, 1, 100, 0, 0, 0, 10, 0, 0],"tagReadFalseSound":[100, 100, 100, 100, 0, 0, 3, 0, 2, 2],"peopleCounterOutgoingLight":[100, 3, 100, 0, 0, 0, 10, 0, 0],"peopleCounterOutgoingSound":[100, 100, 100, 100, 0, 0, 0, 0, 2, 2],"peopleCounterIncomingLight":[100, 1, 100, 0, 0, 0, 10, 0, 0],"peopleCounterIncomingSound":[100, 100, 100, 100, 0, 0, 0, 0, 2, 2],"gpi1Light":[100, 1, 100, 0, 0, 0, 10, 0, 0],"gpi1Sound":[100, 100, 100, 100, 0, 0, 0, 3, 2, 2],"gpi2Light":[100, 1, 100, 0, 0, 0, 10, 0, 0],"gpi2Sound":[100, 100, 100, 100, 0, 0, 3, 0, 2, 2],"hwErrorLight":[100, 3, 100, 0, 0, 0, 20, 0, 4],"hwErrorSound":[100, 100, 100, 100, 500, 100, 0, 3, 2, 2], "autonomousMode":{"enabled":true,"soundIndicationOnlyOnce":true,"reevaluateTagAfter":60,"removeInactiveTagAfter":1800,"conditions":{"targetBank":["EPC","TID"],"regex":"(1234)|(F.*)","validationByDirection":"disabled"},"tagInField":{"indicateAfter":300,"repeatInterval":300}}}

Name of field	Type of value	Possible values	Description
'alarmName'Light	Array of integers	-	Light settings for alarm
1st item	Integer	0 ~ 1000	Time of first color in ms
2nd item	Integer	0 ~ 7	First color where each bit represents RGB e.g. $(5)_{DEC} = (101)_{BIN}$ = 1*red + 0*green +1*blue = magenta
3rd item	Integer	0 ~ 1000	Time of second color in ms
4th item	Integer	0 ~ 7	Second color
5th item	Integer	0 ~ 1000	Time of third color in ms
6th item	Integer	0 ~ 7	Third color
7th item	Integer	0 ~ 100	Number of repetitions
8th item	Integer	0 ~ 1000	Delay after alarm in ms
9th item	Integer	0 ~ 7	Decoration color
'alarmName'Sound	Array of integers	-	Sound settings for alarm
1st item	Integer	0 ~ 1000	First step on time in ms
2nd item	Integer	0 ~ 1000	First step off time in ms
3rd item	Integer	0 ~ 1000	Second step on time in ms
4th item	Integer	0 ~ 1000	Second step off time in ms
5th item	Integer	0 ~ 1000	Third step on time in ms
6th item	Integer	0 ~ 1000	Third step off time in ms
7th item	Integer	0 ~ 100	Number of repetitions
8th item	Integer	0 ~ 1000	Delay after alarm in ms
9th item	Integer	0 ~ 3	Volume of internal buzzer (muted/low/medium/high)
10th item	Integer	0 ~ 3	Volume of external buzzer (muted/low/medium/high)

All possible alarm names:

- tagReadOutgoingRegular, tagReadIncoming, tagReadFalse, tagReadInField
- peopleCounterOutgoing, peopleCounterIncoming
- gpi1, gpi2
- hwError

Name of field	Type of value	Possible values	Description
autonomousMode	Object	-	Setting of autonomous mode
enabled	Boolean	true/false	Enabled or disabled autonomous mode
soundIndicationOnlyOnce	Boolean	true/false	Sound indication of alarm only once if alarm persist, otherwise repeat alarm
reevaluateTagAfter	Integer	0 ~ 86400	Tag that is not visible given period of time in seconds can be reevaluated again – tag in field mark is deleted and direction can be reassigned. 0 - disabled
removeInactiveTagAfter	Integer	0 ~ 86400	Tag that is not visible given period of time in seconds will be removed from buffer. 0 - disabled
conditions	Object	-	Conditions of autonomous mode to trigger alarm
targetBank	Array of strings	EPC/TID	Searching in EPC/TID or both banks
regex	String	-	Regular expression used to evaluate alarm on target bank(s)
validationByDirection	String	disabled/outgoing/ incoming/both	Validate alarm by direction
tagInField	Object	-	Tag in field settings
indicateAfter	Integer	0 ~ 3600	Tag that is visible whole given period of time in seconds is then marked as tag in field. 0 - disabled
repeatInterval	Integer	0 ~ 3600	If some tag is marked as tag in field and is still visible, then alarm is repeated by this given interval in seconds. 0 - never repeat, trigger only first time

Regex specifications:

- matches are greedy and possessive
- only anchored expressions are supported (^ and $ are implied whether specified or not)

Expression	Matches
.	any character
\xXX	character 0xXX
\d	any digit character (0-9)
\D	any non-digit character
\h	any hexadecimal character (0-9, A-F or a-f)
\H	any non-hexadecimal character
\w	any word character (0-9, A-Z, a-z or _)
\W	any non-word character
\? * \+	meta character
?	zero or one time
*****	zero or more times
+	one or more times
A|B	A or B
()	capturing group
(?:)	non-capturing group
(?=)	positive look-ahead
(?!)	negative look-ahead
(?i)	enable case insensitivity
(?I)	disable case insensitivity

Example: (1234)|(F.*) – matches tag containing exactly 1234 or tag starting with F

Get alarm counters

Method: GET
Command: /control/alarmCounters.json?hash=XXXXXX
Request body: none

Response:
{"status":"ok","aisle":{"no1":{"tagOutgoingRegular":4,"tagIncoming":0,"tagFalse":0},"no4":{"tagOutgoingRegular":0,"tagIncoming":7,"tagFalse":0}},"gpi1":1,"gpi2":3,"hwError":0}

Name of field	Type of value	Possible values	Description
aisle	Object	-	Counters for all aisles
noX	Object	-	Number of aisle (1-255)
tagOutgoingRegular	Integer	-	Regular or outgoing tag alarm counter
tagIncoming	Integer	-	Incoming tag alarm counter
tagFalse	Integer	-	False tag alarm counter
gpiX	Integer	-	GPI (1-2) alarm counter
hwError	Integer	-	Hardware error counter

GPIO

Method: GET
Command: /config/GPIO.json?hash=XXXXXX
Request body: none

Response:
{"status":"ok","in1":[3, 0],"in2":[5, 0],"out1":[0],"out2":[0]}

Name of field	Type of value	Possible values	Description
inX	Array of integers	-	Input settings
1st item	Integer	0 ~ 5	Action (none/read control/alarm trigger/ POS - activate tag/POS - deactivate tag/ POS - deactivate tag and remove password)
2nd item	Integer	0 ~ 2	Edge (falling/rising/both)
outX	Array of integers	-	Output settings
1st item	Integer	0 ~ 3	Trigger (none/read indication/alarm indication/ REST API or MQTT controlled)

Get GPIO states

Method: GET
Command: /control/GPIO.json?hash=XXXXXX
Request body: none

Response:
{"status":"ok","in1State":0,"in2State":0,"out1State":0,"out2State":0}

Name of field	Type of value	Possible values	Description
in1State	Integer	0 ~ 1	Current state of first input (off/on)
in2State	Integer	0 ~ 1	Current state of second input (off/on)
out1State	Integer	0 ~ 1	Current state of first output (off/on)
out2State	Integer	0 ~ 1	Current state of second output (off/on)

POS

Method: GET
Command: /config/pos.json?hash=XXXXXX
Request body: none

Response:
{"status":"ok","passwordType":"static","accessPassword":"00000000","rewriteEPCmode":"TID","activationFlag":"FFFF","deactivationFlag":"0000","activationOffset":0,"deactivationOffset":0}

Name of field	Type of value	Possible values	Description
passwordType	String	none/static/dynamic	Indicates if password is used and will be static or generated for each tag individually
accessPassword	String	00000000 ~ FFFFFFFF	Access password for manipulation with tags (only if static password is used). 8 hexadecimal characters.
rewriteEPCmode	String	TID/EPC/flag	New EPC will be based on the tag TID or EPC combined with flag. If a flag is specified, only the flag will be written. In that case, offset has to be a multiple of 4 and cannot be negative.
activationFlag	String	-	Flag that will replace EPC once the tag is de/activated.
deactivationFlag	String	-	
activationOffset	Integer	-	Offset position of the flag in EPC memory once the tag is de/activated. Positive from the bank beginning, negative number from the end. Offset is in words.
deactivationOffset	Integer	-	

For more information about POS settings, refer to „Callidus UHF RFID POS“ document.

People counters

Method: GET
Command: /config/peopleCounters.json?hash=XXXXXX
Request body: none

Response:
{"status":"ok","PC1mode":[0,0,3],"PC2mode":[0,0,3],"port1PCbinding":[2,2,15,1],"port2PCbinding":[1,2,15,1],"port3PCbinding":[2,0],"port4PCbinding":[2,1,2]}

Name of field	Type of value	Possible values	Description
PCXmode	Array of integers	-	People counting settings
1st item	Integer	0 ~ 3	Operational mode (disabled/receiver/ transmitter/receiver and transmitter)
2nd item	Integer	0 ~ 3	Wrong way indication (disabled/incoming/outgoing/both directions)
3rd item	Integer	0 ~ 1	Swap counting direction (off/on)
portXPCbinding	Array of integers		Tag directionality evaluation
1st item	Integer	1 ~ 255	Pedestal aisle
2nd item	Integer	0 ~ 2	Evaluating sensor (none/internal/external)
3rd item	Integer	1 ~ 2 or 0 ~ 31	PC (1/2) if internal PC selected Group ID (0~31) if external PC selected
4th item	Integer	0 ~ 31	Pedestal ID if external PC selected

Get people counters

Method: GET
Command: /control/peopleCounters.json?hash=XXXXXX
Request body: none

Response:

{"status":"ok","counter1":[0,0,5],"counter2":[20,3,10],"states1":[0,0],"states2":[1,2]}

Name of field	Type of value	Possible values	Description
counterX	Array of integers	-	Counters
1st item	Integer	-	Incoming counter
2nd item	Integer	-	Outgoing counter
3rd item	Integer	-	Wrong way events counter
statesX	Array of integers	-	Current sensors state
1st item	Integer	0 ~ 2	First sensor (off/active/idle)
2nd item	Integer	0 ~ 2	Second sensor (off/active/idle)

Passwords

Method: POST
Command: /config/changePasswords.json
Request body:
{"hash":"XXXXXX","adminPassword":{"current":"abc","new":"123","confirmNew":"123"},"userPassword":{"new":"aaa","confirmNew":"aaa"}}

Response: {"status":"ok"}

Name of field	Required	Type of value	Possible values	Description
adminPassword	No	Object	-	Change password for admin with full rights
userPassword	No	Object	-	Change password for user with limited rights
current	Yes*	String	-	Current password for specific user
new	Yes	String	-	New password for specific user
confirmNew	Yes	String	-	Confirmation of new password for specific user

Admin can change password for user without needing to know his current password.
User can change only his own password, but has to specify his current password.

Networking

Method: POST
Command: /config/network.json
Request body:
{"hash":"XXXXXX","netbiosName":"RFID","dhcp":false,"ip":"192.168.1.100","netmask":"255.255.255.0","gw":"192.168.0.1","pridns":"192.168.0.1"}

Response: {"status":"ok"}

Name of field	Required	Type of value	Possible values	Description
netbiosName	No	String	-	Name of device
dhcp	No	Boolean	true/false	DHCP enabled
ip	No	String	IP address	IP address of device
netmask	No	String	IP address	Subnet mask
gw	No	String	IP address	Gateway
pridns	No	String	IP address	Primary DNS

MQTT

Method: POST
Command: /config/MQTT.json
Request body:
{"hash":"XXXXXX","enabled":true,"server":"broker.mqttdashboard.com","port":1883,"username":"user","password":"pass","topicPrefix":"monitoring/rfid","allowRemoteConfig":true}

Response: {"status":"ok"}

Name of field	Required	Type of value	Possible values	Description
enabled	No	Boolean	true/false	MQTT enabled
allowRemoteConfig	No	Boolean	true/false	Allow configure reader over MQTT
server	No	String	-	MQTT broker
port	No	Integer	-	Port (usually 1883)
username	No	String	-	Username
password	No	String	-	Password
topicPrefix	No	String	-	Topic prefix

Antenna ports

Method: POST
Command: /config/antennaPort.json
Request body:
{"hash":"XXXXXX","port1":false,"port2":true,"port3":false,"port4":true}

Response: {"status":"ok"}

Name of field	Required	Type of value	Possible values	Description
portX	No	Boolean	true/false	Enable or disable antenna port (1-8)

Reset to default

Method: POST
Command: /config/factoryReset.json
Request body:
{"reader":true,"radio":true,"network":false,"MQTT":false,"hash":"XXXXXX"}

Response: {"status":"ok"}

Name of field	Required	Type of value	Possible values	Description
reader	No	Boolean	true/false	Reset reader settings
radio	No	Boolean	true/false	Reset radio settings
network	No	Boolean	true/false	Reset networking configuration
MQTT	No	Boolean	true/false	Reset MQTT configuration

Networking configuration reset will enable DHCP and communication can be lost.
MQTT configuration reset will reset to default, which will disable MQTT.
If sent from MQTT, communication will be lost!

FW update

Method: POST
Command: /config/fwUpdateReader.json
Request body:
{"hash":"XXXXXX"}

Response: {"status":"ok"}

GEN2

Method: POST
Command: /config/gen2.json
Request body:
{"hash":"XXXXXX","accessPassword":"#1234CDEF","BLF":320,"tari":6.25,"tagEncoding":"FM0","session":"S2","target":"AB","q":7,"sendSelectEveryQuery":false}

Response: {"status":"ok"}

Name of field	Required	Type of value	Possible values	Description
accessPassword	No	String	#00000000 ~ #FFFFFFFF	Access password for manipulation with tags. 8 hexadecimal characters.
BLF	No	Integer	250/320/640	Backscatter link frequency in kHz
tari	No	Float	25/12.5/6.25	Tari value in µs
tagEncoding	No	String	FM0/M2/M4/M8	Tag encoding method
session	No	String	S0/S1/S2/S3	Session flag state
target	No	String	A/B/AB/BA	Target value
q	No	Integer	0 ~ 15	Q value
sendSelectEveryQuery	No	Boolean	true/false	Disable or enable sending select command every query

For more information about GEN2 settings, refer to „Mercury API programmer guide“.

Radio

Method: POST
Command: /config/radio.json
Request body:
{"hash":"XXXXXX","readingPower":20,"readingPowerPerPort":{"port1":25,"port2":30},"writingPower":20,"writingPowerPerPort":{"port3":null},"hash":"14030514302686306"}

Response: {"status":"ok"}

Name of field	Required	Type of value	Possible values	Description
readingPower	No	Integer	-10 ~ 30	Set global reading power in dBm
writingPower	No	Integer	-10 ~ 30	Set global writing power in dBm
readingPowerPerPort	No	Object	-	Overwrites reading global power for individual ports if specified in dBm
writingPowerPerPort	No	Object	-	Overwrites writing global power for individual ports if specified in dBm
portX	No	Integer	null/-10 ~ 30	Power for individual ports (1-8) Null will remove overwriting.

Region

Method: POST
Command: /config/region.json
Request body:
{"hash":"XXXXXX","id":"North America","hopTable":[918250, 913250],"hopTime":250}

Response: {"status":"ok"}

Name of field	Required	Type of value	Possible values	Description
id	No	String	-	Set region
hopTable	No	Array of integers	-	Array of frequencies
hopTime	No	Integer	1 ~ 4000	Time for each frequency in ms

You can get supported regions with GET request.

Inventory settings

Method: POST
Command: /config/inventorySettings.json
Request body:
{"hash":"XXXXXX","antPorts":[2,3],"targetBank":["TID"],"readingOnBoot":true,"autostartReading":3,"fullBuffer":"overwrite","filters":[{"bank":"EPC","offsetInBits":32,"lengthInBits":24,"mask":"ABCDEF","action":"ON_N_NOP","target":"S1","invert":false},{"bank":"user","offsetInBits":8,"lengthInBits":16,"mask":"FFFF","action":"NOP_N_OFF","target":"S3","invert":false}]}

Response: {"status":"ok"}

Name of field	Required	Type of value	Possible values	Description
antPorts	No	Array of integers	-	Enabled antenna ports
targetBank	No	Array of strings	EPC/TID	Banks to inventory
readingOnBoot	No	Boolean	true/false	Start inventorying on boot
autostartReading	No	Integer	0 ~ 86400	Start inventorying automatically after X minutes span if not running
fullBuffer	No	String	stop/overwrite	Once the inventory buffer is full, new tags will be ignored or overwrite the oldest one
filters	No	Array of objects	-	Radio HW filters
bank	Yes	String	EPC/TID/user/ EPCtruncate/EPClength	Memory bank to apply filter on
offsetInBits	No	Integer	0 ~ 512	Bit offset starting at the bank position
lengthInBits	Yes	Integer	0 ~ 512	Bit length of the comparison
mask	Yes	String	-	Pattern to be matched
action	No	String	table in GET request	Set, reset, flip or leave alone the flag
target	No	String	SL/S0/S1/S2/S3	Which flag should be changed
invert	No	Boolean	true/false	Invert the selection

Test LED

Method: POST
Command: /control/LED/readerStatusLED.json
/control/LED/radioStatusLED.json
Request body:
{"hash":"XXXXXX","color1":"#FF0000","time1":250,"color2":"#00FF00","time2":100,"color3":"#000000","time3":0,"delay":100,"repeat":3}

Response: {"status":"ok"}

Name of field	Required	Type of value	Possible values	Description
colorX	No	String	#000000 ~ #FFFFFF	Color (1-3) of each individual phase in RGB HEX code (dimming is NOT supported)
timeX	No	Integer	0 ~ 1000	Time (1-3) of each color in ms
delay	No	Integer	0 ~ 1000	Delay between repeated sequences in ms
repeat	No	Integer	0 ~ 100	Number of repetitions

Test buzzer

Method: POST
Command: /control/buzzer/internal.json
/control/buzzer/external.json
Request body:
{"hash":"XXXXXX","onTime1":100,"offTime1":50,"onTime2":200,"offTime2":50,"delay":100,"repeat":3,"volume":2}

Response: {“status”:”ok”}

Name of field	Required	Type of value	Possible values	Description
onTimeX	No	Integer	0 ~ 1000	Time on (1-2) of each individual phase in ms
offTimeX	No	Integer	0 ~ 1000	Time off (1-2) of each individual phase in ms
delay	No	Integer	0 ~ 1000	Delay between repeated sequences in ms
repeat	No	Integer	0 ~ 100	Number of repetitions
volume	No	Integer	0 ~ 2	Buzzer volume (low/medium/high)

Start read tags

Method: POST
Command: /control/tagRead.json
Request body:
{"hash":"XXXXXX","reading":true}

Response: {"status":"ok"}

Name of field	Required	Type of value	Possible values	Description
reading	Yes	Boolean	true/false	Enable or disable reading

Delete tag buffer

Method: POST
Command: /control/tagBufferDelete.json
Request body:
{"hash":"XXXXXX"}

Response: {"status":"ok", "deleted": 50}

Inspect tag

Method: POST
Command: /control/tagInspector.json
Request body:
{"hash":"XXXXXX","readingTimeMs":1000,"targetBank":["reserved","EPC","user"],"antPorts":[1,4],"filters":[{"bank":"EPC","offsetInBits": 32,"lengthInBits":8,"mask":"E2"}]}

Response:
{"status":"ok","tags":[{"port":4,"count":1,"lastRSSI":68,"phase":101,"frequency":866900,"EPCID":"E2005179981901422820026E","reservedBank":"0000000000000000","EPCBank":"FF9A3000E2005179981901422820026E","userBank":"00000000000000000000000000000000"}]}

Name of field	Required	Type of value	Possible values	Description
readingTimeMs	No	Integer	0 ~ 5000	Reading time in ms
targetBank	Yes	Array of strings	reserved/EPC/TID/user	Memory banks to inspect
antPorts	No	Array of integers	-	Antenna ports used for inspection
filters	No	Array of objects	-	Radio HW filters
bank	Yes	String	EPC/TID/user/ EPCtruncate/ EPClength	Memory bank to apply filter on
offsetInBits	No	Integer	0 ~ 512	Bit offset starting at the bank position
lengthInBits	Yes	Integer	0 ~ 512	Bit length of the comparison
mask	Yes	String	-	Pattern to be matched

Write new EPC

Method: POST
Command: /control/tagWriteEPC.json
Request body:
{"hash":"XXXXXX","targetDataInWords":"1234","antPorts":[4],"filters":[{"bank":"TID","offsetInBits":0,"lengthInBits":96,"mask":"E28068102000000127BE0B80"}]}

Response: {"status":"ok"}

Name of field	Required	Type of value	Possible values	Description
targetDataInWords	Yes	String	-	New EPC data in HEX
antPorts	No	Array of integers	-	Antenna ports used for writing
filters	No	Array of objects	-	Radio HW filters

Filter parameters are the same as in tag inspector.

Write new data

Method: POST

Command: /control/tagWriteData.json

Request body:

{"hash":"XXXXXX","targetBank":"reserved","antPorts":[4],"targetOffsetInWords":0,"targetDataInWords":"00000001","filters":[{"bank":"TID","offsetInBits":0,"lengthInBits":96,"mask":"E28068102000000127BE0B80"}]}

Response: {"status":"ok"}

Name of field	Required	Type of value	Possible values	Description
targetBank	Yes	String	Reserved/EPC/user	Bank to write
antPorts	No	Array of integers	-	Antenna ports used for writing
targetOffsetInWords	No	Integer	0 ~ 65535	Word offset starting at the bank position
targetDataInWords	Yes	String	-	New data in HEX
targetLengthInWords	No	Integer	0 ~ 65535	Length of data to write in words
filters	No	Array of objects	-	Radio HW filters

Filter parameters are the same as in tag inspector.

Write locks

Method: POST

Command: /control/tagLocks.json

Request body:

{"hash":"XXXXXX","antPorts":[4],"accessPassword":"000000001","locks":{"accessPassword":"unlock","killPassword":"unchanged","EPCmemory":"unchanged","userMemory":"unchanged"},"filters":[{"bank":"TID","offsetInBits":0,"lengthInBits":96,"mask":"E28068102000000127BE0B80"}]}

Response: {"status":"ok"}

Name of field	Required	Type of value	Possible values	Description
antPorts	No	Array of integers	-	Antenna ports used for writing
accessPassword	Yes	String	00000000 ~ FFFFFFFF	Access password for manipulation with tags. 8 hexadecimal characters.
locks	Yes	Array of objects	-	Action performed on locks
accessPassword	No	String	unchanged/ unlock/ lock/ unlockPermanently/ lockPermanently	Lock for access password
killPassword	No	String		Lock for kill password
EPCmemory	No	String		Lock for EPC memory
userMemory	No	String		Lock for user memory
filters	No	Array of objects	-	Radio HW filters

Filter parameters are the same as in tag inspector.

Kill tag

Method: POST
Command: /control/tagKill.json
Request body:
{"hash":"XXXXXX","antPorts":[4],"killPassword":"000000002","filters":[{"bank":"TID","offsetInBits":0,"lengthInBits":96,"mask":"E28068102000000127BE0B80"}]}

Response: {"status":"ok"}

Name of field	Required	Type of value	Possible values	Description
antPorts	No	Array of integers	-	Antenna ports used for writing
killPassword	Yes	String	00000000 ~ FFFFFFFF	Kill password for manipulation with tags. 8 hexadecimal characters.
filters	No	Array of objects	-	Radio HW filters

Filter parameters are the same as in tag inspector.

This action can NOT be undone and tag will be **permanently** killed!
If no filter is specified it will kill first tag that respond.

Set time

Method: POST
Command: /config/time.json
Request body:
{"hash":"XXXXXX","actualTime":[15,37,0,15,12,20],"timezoneDiff":60,"DST":{"enabled":true,"begin":{"month":3,"time":2,"date":28},"end":{"month":10,"time":3,"weekday":1,"weekOfMonth":6},"beginBy":1,"endBy":0},"ntpSync":true,"ntp":"pool.ntp.org","ntpSyncInterval":1,"driftCorrection":-1}

Response: {"status":"ok"}

Name of field	Required	Type of value	Possible values	Description
actualTime	No	Array of integers	-	Set current time (hours, minutes, seconds, day, month, year)
timezoneDiff	No	Integer	-720 ~ 720	Time zone difference in minutes
DST	No	Object	-	Daylight saving time (DST) settings
enabled	No	Boolean	true/false	Enable or disable DST
begin/end	No	Object	-	Beginning or end of DST
month	No	Integer	1 ~ 12	Month where 1 – January and 12 - December
time	No	Integer	0 ~ 23	Hour of the beginning and end of DST
date	No	Integer	1 ~ 31	Day of the begging and end of DST
weekday	No	Integer	1 ~ 7	Day where 1 – Monday and 7 - Sunday
weekOfMonth	No	Integer	1 ~ 6	Week where 1 – first and 5 – fifth, 6 – last week
ntpSync	No	Boolean	true/false	Enable or disable NTP synchronization
ntp	No	String	-	NTP server (name or IP)
ntpSyncInterval	No	Integer	1 ~ 65535	Synchronization interval in minutes
driftCorrection	No	Integer	-127 ~ 128	Time correction per day $\Delta t = n \cdot \frac{86400}{32 \cdot 32768}$ s/day

Alarms

Method: POST
Command: /config/alarms.json
Request body:
{"hash":"XXXXXX","tagReadFalseLight":[100,1,100,0,0,0,10,0,0],"tagReadFalseSound":[100,100,100,100,0,0,3,0,0,0],"autonomousMode":{"enabled":true,"soundIndicationOnlyOnce":false,"reevaluateTagAfter":60,"removeInactiveTagAfter":1800,"conditions":{"targetBank":["EPC","TID"],"regex":"(1234)|(F.*)","validationByDirection":"disabled"},"tagInField":{"indicateAfter":300,"repeatInterval":300}}}

Response: {"status":"ok"}

You can set multiple alarms at once.
All alarm names and values in array have the same meaning as in GET request and are all required.
All autonomous mode object names and values have the same meaning as in GET request and are all optional.

Trigger predefined alarm

Method: POST
Command: /control/alarm.json
Request body:
{"hash":"XXXXXX","type":"tagReadFalse"}

Response: {"status":"ok"}

Name of field	Required	Type of value	Possible values	Description
type	Yes	String	-	Name of alarm to trigger

All alarm names are listed in GET request example.

Trigger custom alarm

Method: POST
Command: /control/alarmTrigger.json
Request body:
{"hash":"XXXXXX","light":[150,1,150,0,0,0,5,0,0],"sound":[50,100,200,300,0,0,3,0,2,2]}

Response: {"status":"ok"}

Name of field	Required	Type of value	Possible values	Description
light	No	Array of integers	-	Light settings for alarm
sound	No	Array of integers	-	Sound settings for alarm

All values in array have the same meaning as in GET request and are all required.

Reset alarm counters

Method: POST

Command: /control/alarmCountersReset.json

Request body:

{"hash":"XXXXXX","aisle":{"no1":{"tagOutgoingRegular":true,"tagFalse":true},"no3":{"tagOutgoingRegular":true,"tagIncoming":true}},"gpi1":true, "gpi2":false,"hwError":true}

Response: {"status":"ok"}

Name of field	Required	Type of value	Possible values	Description
aisle	No	Object	-	Counters for all aisles
noX	No	Object	-	Number of aisle (1-255)
tagOutgoingRegular	No	Boolean	true/false	Clear regular/outgoing alarm counter
tagIncoming	No	Boolean	true/false	Clear incoming tag alarm counter
tagFalse	No	Boolean	true/false	Clear false tag alarm counter
gpiX	No	Boolean	true/false	Clear GPI (1-2) alarm counter
hwError	No	Boolean	true/false	Clear HW error counter

GPIO

Method: POST

Command: /config/GPIO.json

Request body:

{"hash":"XXXXXX","in1":[0,0],"in2":[1,2], "out1":[2],"out2":[2]}

Response: {"status":"ok"}

Name of field	Required	Type of value	Possible values	Description
inX	No	Array of integers	-	Input settings
1st item	Yes	Integer	0 ~ 5	Action (none/reading control/alarm trigger/POS - activate tag/ POS - deactivate tag/POS - deactivate tag and remove password)
2nd item	Yes	Integer	0 ~ 2	Edge (falling/rising/both)
outX	No	Array of integers		Output settings
1st item	Yes	Integer	0 ~ 3	Trigger (none/read indication/alarm indication/ REST API or MQTT controlled)

Trigger GPO

Method: POST
Command: /control/setGPO.json
Request body:
{"hash":"XXXXXX","out1":[0],"out2":[1,50]}

Response: {"status":"ok"}

Name of field	Required	Type of value	Possible values	Description
outX	No	Array of integers	-	Desired output
1st item	No	Integer	0 ~ 3	State (do not change/off/on/toggle)
2nd item	No	Integer	0 ~ 60000	Time in ms, 0 means permanently

POS

Method: POST
Command: /config/pos.json
Request body:
{"hash":"XXXXXX","passwordType":"static","accessPassword":"12345678","rewriteEPCmode":"flag","activationFlag":"FFFF","activationOffset":8,"deactivationFlag":"0000","deactivationOffset":8}

Response: {"status":"ok"}

Name of field	Required	Type of value	Possible values	Description
passwordType	No	Boolean	none/static/ dynamic	Indicates if password is used and will be static or generated for each tag individually.
accessPassword	No	String	00000000 ~ FFFFFFFF	Access password for manipulation with tags (only if static password is used). 8 hexadecimal characters.
rewriteEPCmode	No	String	TID/EPC/flag	New EPC will be based on the tag TID or EPC combined with flag. If a flag is specified, only the flag will be written. In that case, offset has to be a multiple of 4 and cannot be negative.
activationFlag	No	String	-	Flag that will replace EPC once the tag is de/activated.
deactivationFlag	No	String	-	
activationOffset	No	Integer	-	Offset position of the flag in EPC memory once the tag is de/activated. Positive from the bank beginning, negative number from the end. Offset is in words.
deactivationOffset	No	Integer	-	

For more information about POS settings, refer to „Callidus UHF RFID POS“ document.

People counters

Method: POST
Command: /config/peopleCounters.json
Request body:
{"hash":"XXXXXX","PC1mode":[0,0,3],"PC2mode":[1,1,2],"port1PCbinding":[2,2,15,1],"port3PCbinding":[2,0],"port4PCbinding":[2,1,2]}

Response: {"status":"ok"}

All names and values in array have the same meaning as in GET request.
Not all objects have to be sent but all values in that array are required.

Reset people counters

Method: POST
Command: /control/peopleCountersReset.json
Request body:
{"hash":"XXXXXX","counter1":true,"counter2":false}

Response: {"status":"ok"}

Name of field	Required	Type of value	Possible values	Description
counterX	No	Boolean	true/false	Clear counter (1-2)