GT125 Public API

Version 1.1.0
2016-04-15

Sinopé Technologies Inc.
705 Montrichard Avenue
St-Jean-sur-Richelieu (Quebec)
J2X 5K8
Phone: 450 741-7700
Fax: 450 741-7710

Content

Revision History

Revision	Date	Changes
0.7.0	2015/10/23	Preliminary release
1.0.0	2016/01/28	Added instructions on how to set manual IP address. Clarifications in Authentication Key Request description.
1.1.0	2016/02/23	Add Sunrise & Sunset Data Objects. Add index byte for Application Data Objects that are formed of “Struct” data (Date and Time).
1.2.0	2016/04/14	Data Report Answer may now be generated as a notification (push) command.

1. Scope

This document defines the requirements as well as the protocol used to directly interface with the GT125 and the wireless system.

The details and behavior of the wireless devices connected to the GT125 is out of the scope of this document.

2. Intended Audience

This document is intended to be distributed to professionals with a good knowledge of communication protocols and embedded systems programming.

3. Legal / Confidentiality

The information contained in this document is confidential. It may not be reproduced in whole, or in part, nor may any of the information contained therein be disclosed without the prior written consent of Sinopé Technologies' engineering department.

Any form of reproduction, dissemination, copying, disclosure, modification, distribution and or publication of this material is strictly prohibited.

All the information contained in this document is intellectual property owned solely by Sinopé Technologies and is protected as such. They include patents, trademarks, trade names, design rights, copyright (including rights in computer software and moral rights), database rights, rights in know-how and other intellectual property rights, in each case whether registered or unregistered and including applications for the grant of any of the foregoing and all rights or forms of protection having equivalent or similar effect to any of the foregoing which may subsist anywhere in the world.

4. References

1) TH1120RF Application data objects
2) TH1300RF Application data objects
3) SW2500RF Application data objects

5. Connecting

The GT125 local API uses a raw TCP/IP socket on port 4550.

TCP is a stream protocol, an application frame may be received in multiple chunks of data or there might be more than one frame per chunk of data received. The application is responsible to re-assemble or split the data into application frames.

This section describes how the data is exchanged between the GT125 and the client device once the connection is established.

TCP Port:	4550
Encryption:	None
Encoding:	Raw / Binary
Byte order:	LSB first (little endian)
Maximum number of connections:	Only one (1) device may connect to the API at a time.

6. Frame Format

Frame Header			Frame Payload		Frame CRC
Preamble	**FrameCtl**	**Size**	**Command**	**Data**	**CRC**
1u	1u	2u	2u	var	1u

The same frame format is used to transmit and receive data to/from the GT125.

See appendix A for frame examples.

Preamble:	Always: 0x55
FrameCtl:	Always: 0x00
Size:	Frame payload size in bytes. The frame payload size includes the length of the payload command and its data. The length of the header and CRC are excluded.
Command:	Type of message exchanged between the server and client. See section "Command Reference" for a detailed description of the commands and their data.
Data:	Data associated with the command. See section "Command Reference" for the format and length of the associated data.
CRC:	Frame integrity verification using a CRC-8. Frames without a valid CRC will be ignored. The CRC is calculated on the whole frame starting with the preamble and stopping at the last payload data byte. Information on the implementation used: Byte order: LSB first Bit order: msb first Initial value: 0x00 Final XOR: 0x00 (none) Polynomial representation: $C(x) = x^8 + x^2 + x + 1$ (0x07) The following websites may be used to help you find CRC values: http://smbus.org/faq/crc8Applet.htm https://ghsi.de/CRC/index.php?Polynom=100000111 See Appendix B for a C code example of CRC calculation.

7. Command Reference

0x0012 - Ping Request

	Frame Payload	
	Command	Data
	0x0012	
Size (sign)	2 (U)	0 (U)

Command ID:	0x0012
Issued by:	Client
Type:	Request
Authentication required:	No
Expect answer:	Yes
Will answer with:	0x0013
Is the answer to:	-
Description:	Ping the GT125 and/or keep the connection alive.
Data description:	No associated data

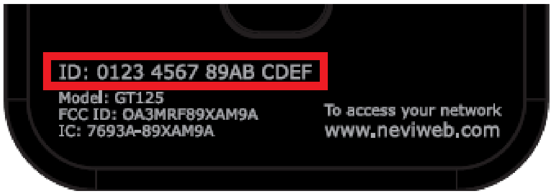
0x0013 - Ping Answer

	Frame Payload	
	Command	Data
	0x0013	
Size (sign)	2 (U)	0 (U)

Command ID:	0x0013
Issued by:	Server
Type:	Answer
Authentication required:	No
Expect answer:	No
Will answer with:	-
Is the answer to:	0x0012
Description:	Answer to a ping request.
Data description:	No associated data

0x010A - Authentication Key Request

	Frame Payload	
	Command	Data
	0x010A	**Id**
Size (sign)	2 (U)	8 (U)

Command ID:	0x010A
Issued by:	Client
Type:	Request
Authentication required:	No
Expect answer:	Yes
Will answer with:	0x010B
Is the answer to:	-
Description:	Request the API authentication key to the GT125. The authentication key is required to initiate an authenticated session and access the wireless devices. To retrieve the key: 1) Send a valid key request. 2) Press the "Web" button on the GT125 within 300 seconds. 3) Retrieve the key from the answer (if successful). **Invalid requests will only generate an error after the 300 seconds delay.** The following events will cause an invalid request: • The "Web" button on the GT125 is not pressed. • The GT125 ID does not match. • An authenticated session with the GT125 is already established. The authentication key can be erased to remove access to the devices that were previously authorized to use the API. To erase the key: 1) Send a valid key request. 2) Press and hold the "Web" button for about 5 seconds on the GT125 within 300 seconds. 3) Wait for confirmation that the key was erased. After the key was erased, a new key can be retrieved by doing a new retrieve sequence. If other clients are connected when the key erase occurs, they will be disconnected.
Data description:	
ID	GT125 ID (printed on the product).  ID: 0123 4567 89AB CDEF Model: GT125 FCC ID: OA3MRF89XAM9A IC: 7693A-89XAM9A To access your network www.neviweb.com ID for the above illustration: 0x0123456789ABCDEF

0x010B - Authentication Key Answer

	Frame Payload			
	Command	Data		
	0x010B	**Status**	**Backoff**	**API Key**
Size (sign)	2 (U)	1 (S)	2 (U)	8 (U)

Command ID:	0x010B
Issued by:	Server
Type:	Answer
Authentication required:	No
Expect answer:	No
Will answer with:	-
Is the answer to:	0x010A
Description:	Answer to the authentication request. See the authentication key request command for more details.
Data description:	
Status	Result of the request

Value	Status
1	Success (Authentication Key is provided)
2	Authentication Key erased
-1 (0xFF)	Authentication failed / timeout

Backoff	When status signals an error (negative values), this value represents the backoff period in seconds that the client must wait before retrying to connect / authenticate with the server.
API Key	API Key to use to start an authenticated session. Valid only if status is “Success”.

0x0110 – API Login Request

	Frame Payload		
	Command	Data	
	0x0110	**Id**	**API Key**
Size (sign)	2 (U)	8 (U)	8 (U)

Command ID:	0x0110
Issued by:	Client
Type:	Request
Authentication required:	No
Expect answer:	Yes
Will answer with:	0x0111
Is the answer to:	-
Description:	Use this command to start an authenticated session with the GT125. An authenticated session is required to access the wireless devices.
Data description:	
ID	GT125 ID as printed on the product. (see authentication key request)
API Key	API key provided by the authentication process (authentication key answer).

0x0111 – API Login Answer

	Frame Payload						
	Command	Data					
	0x0111	**Status**	**Backoff**	**SwRevMaj**	**SwRevMin**	**SwRevBug**	**DeviceID**
Size (sign)	2 (U)	1 (S)	2 (U)	1 (U)	1 (U)	1 (U)	4 (U)

Command ID:	0x0111
Issued by:	Server
Type:	Answer
Authentication required:	No
Expect answer:	No
Will answer with:	-
Is the answer to:	0x0110
Description:	Indicate the status of the request to start an authenticated session.
Data description:	
Status	Result of the authentication

Value	Status
0	Success
-1 (0xFF)	Login / Authentication failed
-2 (0xFE)	*Reserved*
-3 (0xFD)	*Reserved*
-4 (0xFC)	Client blacklisted / banned

Backoff	When status signals an error (negative values), this value represents the backoff period in seconds that the client must wait before retrying to connect / authenticate with the server.
SwRevMaj	Software version of the GT125. (Major)
SwRevMin	Software version of the GT125. (Minor)
SwRevBug	Software version of the GT125. (BugFix)
DeviceID	DeviceID of the GT125.

0x0116 - Device Link Report

	Frame Payload		
	Command	Data	
	0x0116	**Status**	**DeviceID**
Size (sign)	2 (U)	1 (S)	4 (U)

Command ID:	0x0116
Issued by:	Server
Type:	Report
Authentication required:	Yes
Expect answer:	No
Will answer with:	-
Is the answer to:	-
Description:	Report generated when a device joins the network or when a locate report is initiated from a device. This report can be used to identify a physical device by its DeviceID.
Data description:	

Status

Value	Status
0	Locate report
1	Device joined
-1 (0xFF)	Device left

DeviceID

DeviceID of the device generating the report.
The DeviceID is fixed and does not change over time.

A DeviceID value of 0xFFFFFFFF is used with a status of -1 (0xFF) to report that the wireless network has been erased and that no devices are available.

0x0240 - Data Read Request

0x0242 - Data Report Request

0x0244 - Data Write Request

	Frame Payload									
	Command	Data								
	0x0240 / 0x0242 / 0x0244	**Seq #**	**Request Type**	**Res.1**	**Res.2**	**Res.3**	**Res.4**	**Dest. DeviceID**	**AppData Size**	**AppData**
Size (sign)	2 (U)	4 (U)	1 (U)	1 (U)	1 (U)	2 (U)	2 (U)	4 (U)	1 (U)	var

<table>
<tr><td>Command ID:</td><td>0x0240 / 0x0242 / 0x0244</td></tr>
<tr><td>Issued by:</td><td>Client</td></tr>
<tr><td>Type:</td><td>Request</td></tr>
<tr><td>Authentication required:</td><td>Yes</td></tr>
<tr><td>Expect answer:</td><td>Yes</td></tr>
<tr><td>Will answer with:</td><td>0x0241 / 0x0243 / 0x0245</td></tr>
<tr><td>Is the answer to:</td><td>-</td></tr>
<tr><td>Description:</td><td>The read, report and write requests are the commands used to access application data on the wireless network.

Use the read request to get the data value from a device.

Use the write request to assign a data value to a device.

Use the report request to inform the device(s) of a data value (e.g. outdoor temperature). Report request are usually sent as broadcast to all devices.

Report requests should be viewed as a suggestion while write requests should be viewed as an order.</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Data description:</td><td></td></tr>
<tr><td>Seq #</td><td>Unique sequence number. This value will be used to identify the answer commands to the originating request.

Although the initial value can be randomly generated, it should not be randomly generated on every request to prevent having two identical sequence numbers within a short time frame. It is recommended that the sequence number is incremented on every request sent to the GT125.</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Request Type</td><td><table><tr><th>Value</th><th>Status</th></tr><tr><td><b>0</b></td><td>Request</td></tr><tr><td><b>1</b></td><td>Abort</td></tr></table>
The abort type can be used to stop waiting for the device answer. When sending an abort request, the sequence number and DeviceID must match the original request.</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Res.1</td><td>Reserved fields.</td></tr>
</table>

Res.2 Res.3 Res.4	Use the value 0.
Dest. DeviceID	DeviceID of the device to send the request to. The DeviceID is provided in the “Device Link Report” messages. The DeviceID of the GT125 is provided when the session is established (“API login answer”). Use the value 0xFFFFFFFF to broadcast a report to all devices. There will be no confirmation that the broadcast was received by all devices.
AppData Size	Size of the application data to send on the wireless network in bytes.
AppData	Application data to request or to send on the wireless network. See the “Application Data” and “Application Data Objects” section for the application objects formats and descriptions.

0x0241 - Data Read Answer

0x0243 - Data Report Answer / Notification

0x0245 - Data Write Answer

	Frame Payload							
	Command	Data						
	0x0241 / 0x0243 / 0x0245	**Seq #**	**Status**	**Attempt #**	**More**	**Source DeviceID**	**AppData Size**	**AppData**
Size (sign)	2 (U)	4 (U)	1 (S)	1 (U)	1 (U)	4 (U)	1 (U)	var

Command ID:	0x0241 / 0x0243 / 0x0245
Issued by:	Server
Type:	Answer / Notification
Authentication required:	Yes
Expect answer:	No
Will answer with:	-
Is the answer to:	0x0240 / 0x0242 / 0x0244
Description:	The data read, report and write answers are the answers to the data request commands used to access application data on the wireless network. The "Data Report Answer / Notification" command may also be generated by the server as a data notification (push) message. Data report notifications are identified by a different value of the "Status" field and also are not associated with a "Data Report Request" command (generated asynchronously). Data report notifications requirements: - GT125 firmware v2.3.0 and later - Devices that support the generation of data reports (see the device "Application Data Objects" specification).
Data description:	
Seq #	Sequence number of the request generating the answer message. Shall be ignored when Status is "Data Notification".

Status

Value	Status
0	Ack / Wait for answer Request received and queued; waiting for device answer. Should receive another answer of type "Answer" or "No Answer received" for this request.
1	No Wait Request received and queued; wait for device answer not supported. No more answer for this request will be generated.

2	Aborted Request aborted / removed from queue. No more answer for this request will be generated.
10	Data Answer Answer containing the data to the request. No more answer for this request will be generated.
11	Data Notification A device on the network is sending an unsolicited data report to the server (API). This mess
-1 (0xFF)	Request failed (generic) No more answer for this request will be generated.
-2 (0xFE)	Buffer full Queue full; retry later No more answer for this request will be generated.
-3 (0xFD)	*Reserved*
-4 (0xFC)	No answer received Device is not responding. No more answer for this request will be generated.
-5 (0xFB)	Abort failed Request not found in queue.
-6 (0xFA)	Unknown device The destination DeviceID is invalid or not a member of this network. No more answer for this request will be generated.

Values that are not present in the above table are reserved.

Attempt #

[Advanced setting] Ignore this value.

More

Value	Status
0	Last answer for this request (Final answer).
1	Expect more answer messages associated with this request.

Source DeviceID

DeviceID of the device answering the request.

AppData Size

Size of the application data received from the wireless network in bytes.

AppData

Application data received from the wireless network.

See the “Application Data” and “Application Data Objects” section for the application objects formats and descriptions.

8. Application Data

Object Format

Application data is accessed by objects. Each data object is made of three components: the object identifier, the object data size and the object data value.

	Object Header		Object Data
	DataID	**Data Size**	**Data**
Size (sign)	4 (U)	1 (U)	var

DataID:	Application data object identifier.
Data Size:	Size of the application object data in bytes.
Data:	Application object data value. See the “Application Data Objects” section for the objects descriptions.

Usage

Data Read Request

When sending read requests, only the object identifiers part (DataID) of the objects to read are sent.

Data Report Answer

“Data Report Answer” commands will not return application data objects.

Data Report Request / Data Write Request / Data Read Answer / Data Write Answer

These commands use the complete object format as described on the previous section.

9. Application Data Objects

This section describes the application data objects available in the GT125.

The other wireless devices will be covered in separate documents.

GT125

DataID:	0x00000204
Name:	Outdoor temperature
Accept Read:	Yes
Accept Write:	Yes
Accept Report:	Yes
Size:	2 bytes
Format:	16bit signed integer
Scale:	1 = 0.01°C
Range:	
Description:	Outdoor temperature in Celsius. Must be sent every 60 minutes to remain valid. See the temperature error code section in the API specification for more information about out of range values.

DataID:	0x00000600
Name:	Local Time
Accept Read:	Yes
Accept Write:	Yes
Accept Report:	Yes
Size:	3 bytes
Format:	Struct
[Byte 0] Seconds	8bit unsigned integer
[Byte 1] Minutes	8bit unsigned integer
[Byte 2] Hours	8bit unsigned integer
Scale:	
Seconds	1 = 1 second
Minutes	1 = 1 minute
Hours	1 = 1 hour
Range:	
Seconds	0 @ 59
Minutes	0 @ 59
Hours	0 @ 23 (Normal), 128 @ 151 (DST active)
Description:	Local time in 24h format. Must be sent every 24h.
Hours	The msb (bit 7) in the “Hours” byte is used to distinguish between standard (0) or Daylight Saving Time (1).

DataID:	0x00000601
Name:	Local Date
Accept Read:	Yes
Accept Write:	Yes
Accept Report:	Yes
Size:	4 bytes
Format:	Struct
[Byte 0] Day of week	Enum
[Byte 1] Day of month	8bit unsigned integer
[Byte 2] Month	8bit unsigned integer
[Byte 3] Year	8bit unsigned integer
Scale:	
Day of week	
Day of month	
Month	
Year	
Range:	
Day of week	0 @ 6
Day of month	1 @ 31
Month	1 @ 12
Year	0 @ 99
Description:	Local date.
Day of week	Values for the day of week:

Value	Description
0	Monday
1	Tuesday
2	Wednesday
3	Thursday
4	Friday
5	Saturday
6	Sunday

Year	Years are represented as years from the 2000s.

DataID:	0x00000620
Name:	Sunrise Time
Accept Read:	Yes
Accept Write:	Yes
Accept Report:	Yes
Size:	3 bytes
Format:	Struct
[Byte 0] Seconds	8bit unsigned integer
[Byte 1] Minutes	8bit unsigned integer
[Byte 2] Hours	8bit unsigned integer
Scale:	
Seconds	1 = 1 second
Minutes	1 = 1 minute
Hours	1 = 1 hour
Range:	
Seconds	0 @ 59
Minutes	0 @ 59
Hours	0 @ 23 (Normal), 128 @ 151 (DST active), 255 = No sunrise
Description:	Sunrise time in 24h format.
	Must be sent every 24h for devices with sunrise events such as the SW2500RF.
Hours	
	The msb (bit 7) in the “Hours” byte is used to distinguish between standard (0) or Daylight Saving Time (1).

DataID:	0x00000621
Name:	Sunset Time
Accept Read:	Yes
Accept Write:	Yes
Accept Report:	Yes
Size:	3 bytes
Format:	Struct
[Byte 0] Seconds	8bit unsigned integer
[Byte 1] Minutes	8bit unsigned integer
[Byte 2] Hours	8bit unsigned integer
Scale:	
Seconds	1 = 1 second
Minutes	1 = 1 minute
Hours	1 = 1 hour
Range:	
Seconds	0 @ 59
Minutes	0 @ 59
Hours	0 @ 23 (Normal), 128 @ 151 (DST active), 255 = No sunset
Description:	Sunset time in 24h format. Must be sent every 24h for devices with sunset events such as the SW2500RF.
Hours	The msb (bit 7) in the “Hours” byte is used to distinguish between standard (0) or Daylight Saving Time (1).

DataID:	0x00000700
Name:	Occupancy setback
Accept Read:	Yes
Accept Write:	Yes
Accept Report:	Yes
Size:	1 byte
Format:	enum
Scale:	1 = 1
Range:	
Description:	

Value	Description
0	None
1	*Reserved*
2	Away

Temperature Error Codes

Application objects that have a temperature as their data may return values that are out of range as error codes.

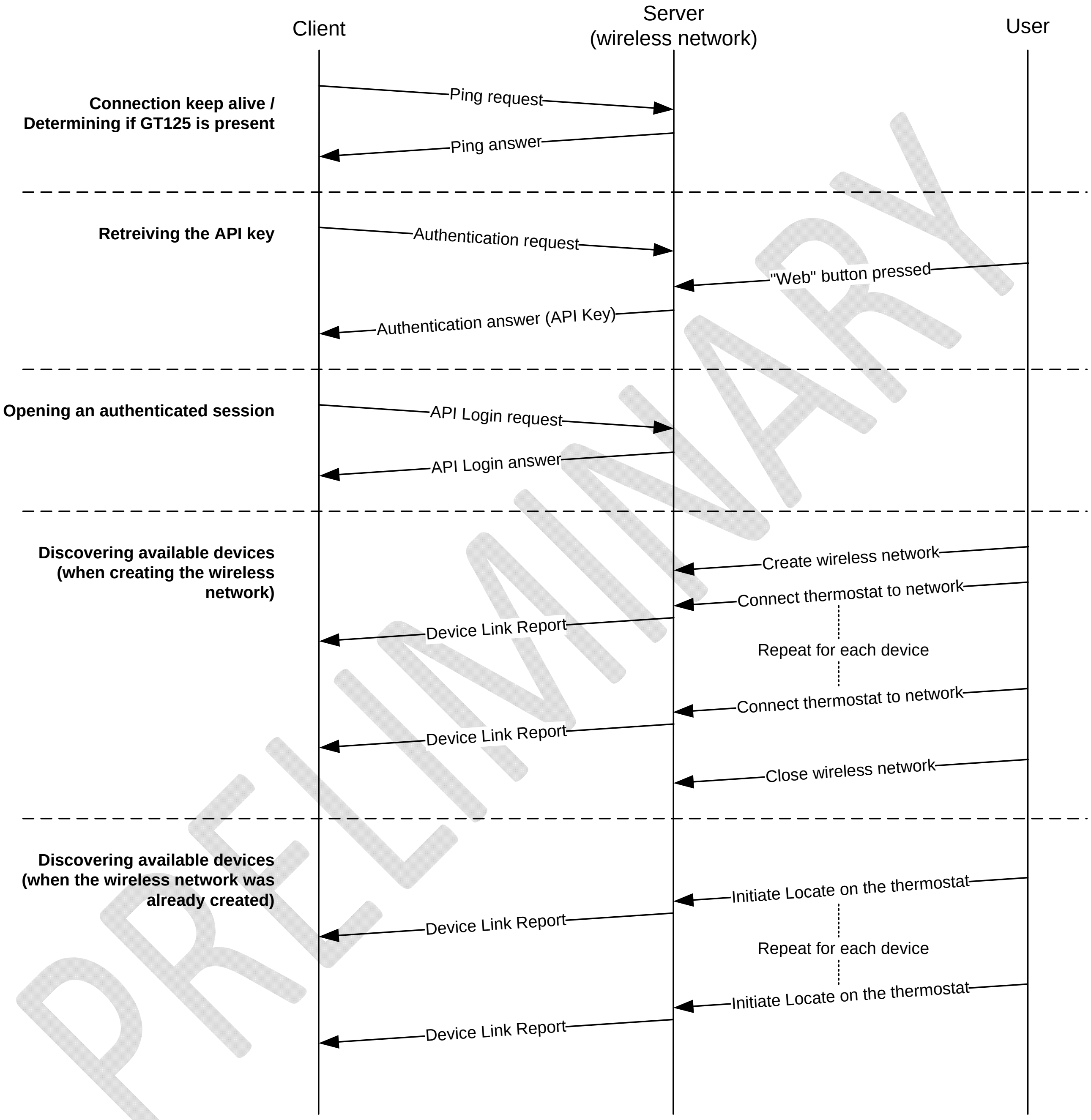
Value (hex)	Description
0x7FF5	Internal error
0x7FF6 **0x7FF7** **0x7FFD** **0x7FFE**	Defective temperature sensor
0x7FF8 **0x7FFF**	Temperature higher than maximum range
0x7FF9 **0x8000** **0x8001**	Temperature lower than minimum range
0x7FFA **0x7FFC**	No value / Invalid / Disabled
0x7FFB	Overload

Local time and Sunrise/Sunset

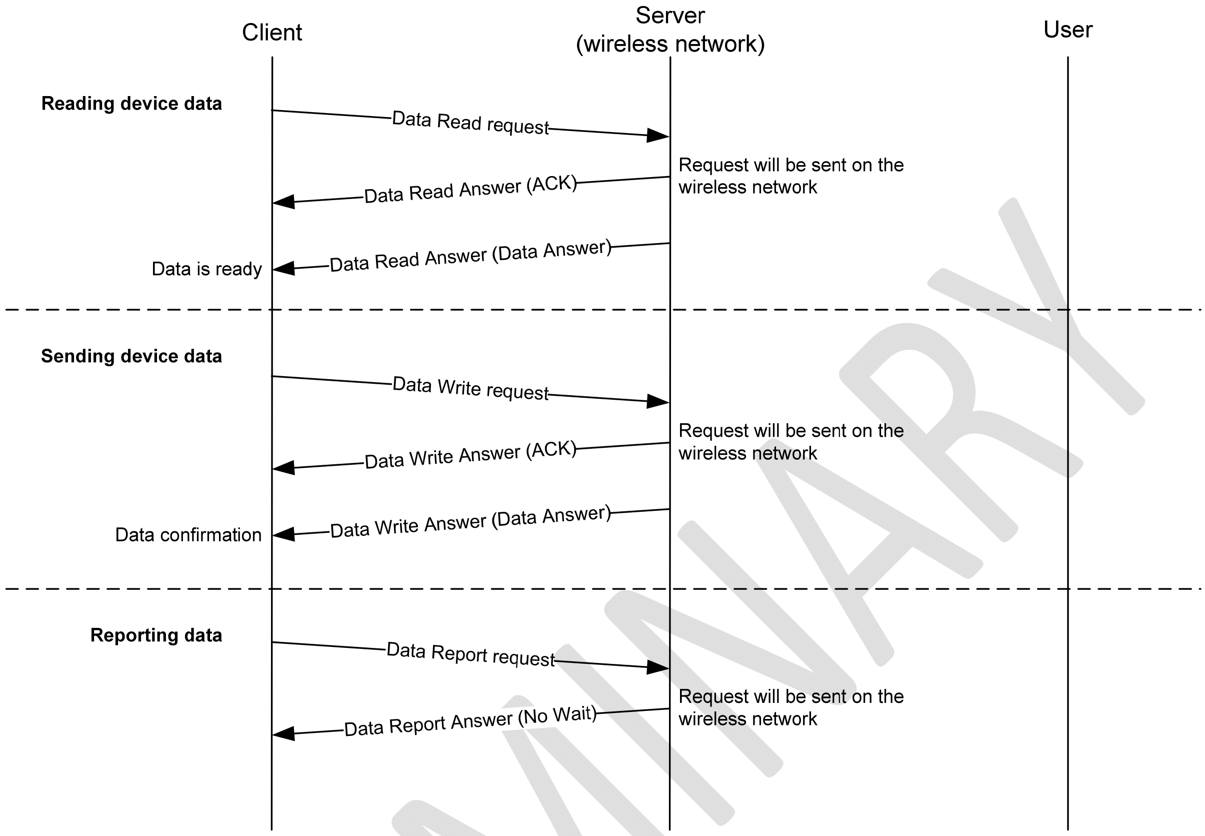
If the time is provided through the API instead of a neviweb® account and the wireless network contains devices with sunrise/sunset events such as the SW2500RF, it is also required to provide the sunrise and sunset through the API.

10. Using the Local API

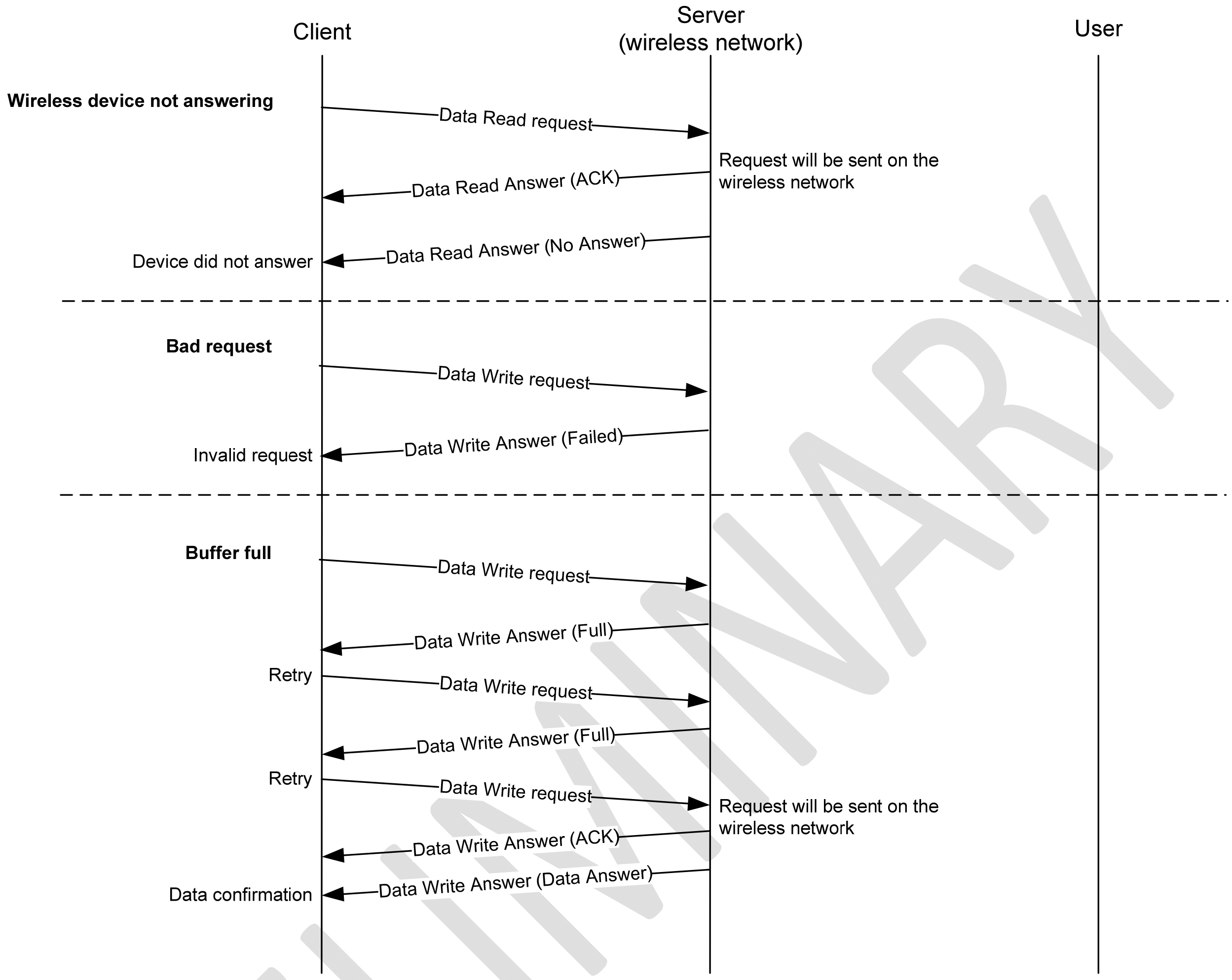
Initial Handshake and Discovery



Accessing Application Data



Errors when Accessing Application Data



Appendix A – Frame Examples

The data in the frame examples is represented in hexadecimal form.

API Key Request

Example of an API key request for a GT125 with an ID of “0123 4567 89AB CDEF":

						Data								
Preambule	FrameCtl	Size LSB	Size MSB	Command LSB	Command MSB	ID (LSB)							ID (MSB)	CRC
55	00	0A	00	0A	01	EF	CD	AB	89	67	45	23	01	DA

Data Read – Thermostat Room Temperature

Example of a Data Read for the room temperature (object DataID 0x00000203) to the thermostat having the DeviceID 0x00000444:

Request to send:

						Data																				
																						AppData				
Preamble	FrameCtl	Size LSB	Size MSB	Command LSB	Command MSB	Seq # (LSB)			Seq # (MSB)	Req. Type	Reserved 1	Reserved 2	Reserved 3	Reserved 3	Reserved 4	Reserved 4	DeviceID (LSB)			DeviceID (MSB)	AppData Size	DataID (LSB)			DataID (MSB)	CRC
55	00	16	00	40	02	78	56	34	12	00	00	00	00	00	00	00	44	04	00	00	04	03	02	00	00	29

Request acknowledge received:

						Data												
Preamble	FrameCtl	Size LSB	Size MSB	Command LSB	Command MSB	Seq # (LSB)			Seq # (MSB)	Status	Attempt #	More	DeviceID (LSB)			DeviceID (MSB)	AppData Size	CRC
55	00	0E	00	41	02	78	56	34	12	00	00	01	44	04	00	00	00	0D

Application data answer received:

						Data																			
																		AppData							
Preamble	FrameCtl	Size LSB	Size MSB	Command LSB	Command MSB	Seq # (LSB)			Seq # (MSB)	Status	Attempt #	More	DeviceID (LSB)			DeviceID (MSB)	AppData Size	DataID (LSB)			DataID (MSB)	Data Size	Temperature (LSB)	Temperature (MSB)	CRC
55	00	15	00	41	02	78	56	34	12	0A	01	00	44	04	00	00	07	03	02	00	00	02	82	07	43

The returned room temperature value is 0x0782 = 19.22°C.

Data Write – Thermostat Room Setpoint

Example of a Data Write for the thermostat setpoint (object DataID 0x00000208) to the thermostat having the DeviceID 0x00000444:

The value to send as the setpoint will be 21.0°C (0x0834).

Request to send:

						Data																							
																						AppData							
Preamble	FrameCtl	Size LSB	Size MSB	Command LSB	Command MSB	Seq # (LSB)			Seq # (MSB)	Req. Type	Reserved 1	Reserved 2	Reserved 3	Reserved 3	Reserved 4	Reserved 4	DeviceID (LSB)			DeviceID (MSB)	AppData Size	DataID (LSB)			DataID (MSB)	Data Size	Temperature LSB	Temperature MSB	CRC
55	00	19	00	44	02	8C	1D	15	B4	00	00	00	00	00	00	00	44	04	00	00	07	08	02	00	00	02	34	08	01

Request acknowledge received:

						Data												
Preamble	FrameCtl	Size LSB	Size MSB	Command LSB	Command MSB	Seq # (LSB)			Seq # (MSB)	Status	Attempt #	More	DeviceID (LSB)			DeviceID (MSB)	AppData Size	CRC
55	00	0E	00	45	02	8C	1D	15	B4	00	00	01	44	04	00	00	00	B9

Application data confirmation (answer) received:

						Data																			
																		AppData							
Preamble	FrameCtl	Size LSB	Size MSB	Command LSB	Command MSB	Seq # (LSB)			Seq # (MSB)	Status	Attempt #	More	DeviceID (LSB)			DeviceID (MSB)	AppData Size	DataID (LSB)			DataID (MSB)	Data Size	Temperature LSB	Temperature MSB	CRC
55	00	15	00	45	02	8C	1D	15	B4	0A	01	00	44	04	00	00	07	08	02	00	00	02	34	08	5E

Data Report – Broadcast Local Time

Example of a Data Report for the local time (object DataID 0x00000600) to all devices (broadcast) on the network:

The value to send for the local time will be 16:09:00 (4:09:00pm) while DST is active.

Request to send:

						Data																								
																						AppData								
Preamble	FrameCtl	Size LSB	Size MSB	Command LSB	Command MSB	Seq # (LSB)			Seq # (MSB)	Req. Type	Reserved 1	Reserved 2	Reserved 3	Reserved 3	Reserved 4	Reserved 4	DeviceID (LSB)			DeviceID (MSB)	AppData Size	DataID (LSB)			DataID (MSB)	Data Size	Seconds	Minutes	Hours	CRC
55	00	1A	00	42	02	8B	FF	03	6F	00	00	00	00	00	00	00	FF	FF	FF	FF	08	00	06	00	00	03	00	09	90	51

Request acknowledge received:

						Data												
Preambule	FrameCtl	Size LSB	Size MSB	Command LSB	Command MSB	Seq # (LSB)			Seq # (MSB)	Status	Attempt #	More	DeviceID (LSB)			DeviceID (MSB)	AppData Size	CRC
55	00	0E	00	43	02	8B	FF	03	6F	01	00	00	FF	FF	FF	FF	00	E7

Data reports and broadcast do not confirm the sent values.

Appendix B – CRC Calculation C Code

```
const uint8_t CrcTable[256] = {
    0x00, 0x07, 0x0e, 0x09, 0x1c, 0x1b, 0x12, 0x15
    0x38, 0x3f, 0x36, 0x31, 0x24, 0x23, 0x2a, 0x2d
    0x70, 0x77, 0x7e, 0x79, 0x6c, 0x6b, 0x62, 0x65
    0x48, 0x4f, 0x46, 0x41, 0x54, 0x53, 0x5a, 0x5d
    0xe0, 0xe7, 0xee, 0xe9, 0xfc, 0xfb, 0xf2, 0xf5
    0xd8, 0xdf, 0xd6, 0xd1, 0xc4, 0xc3, 0xca, 0xcd
    0x90, 0x97, 0x9e, 0x99, 0x8c, 0x8b, 0x82, 0x85
    0xa8, 0xaf, 0xa6, 0xa1, 0xb4, 0xb3, 0xba, 0xbd
    0xc7, 0xc0, 0xc9, 0xce, 0xdb, 0xdc, 0xd5, 0xd2
    0xff, 0xf8, 0xf1, 0xf6, 0xe3, 0xe4, 0xed, 0xea
    0xb7, 0xb0, 0xb9, 0xbe, 0xab, 0xac, 0xa5, 0xa2
    0x8f, 0x88, 0x81, 0x86, 0x93, 0x94, 0x9d, 0x9a
    0x27, 0x20, 0x29, 0x2e, 0x3b, 0x3c, 0x35, 0x32
    0x1f, 0x18, 0x11, 0x16, 0x03, 0x04, 0x0d, 0x0a
    0x57, 0x50, 0x59, 0x5e, 0x4b, 0x4c, 0x45, 0x42
    0x6f, 0x68, 0x61, 0x66, 0x73, 0x74, 0x7d, 0x7a
    0x89, 0x8e, 0x87, 0x80, 0x95, 0x92, 0x9b, 0x9c
    0xb1, 0xb6, 0xbf, 0xb8, 0xad, 0xaa, 0xa3, 0xa4
    0xf9, 0xfe, 0xf7, 0xf0, 0xe5, 0xe2, 0xeb, 0xec
    0xc1, 0xc6, 0xcf, 0xc8, 0xdd, 0xda, 0xd3, 0xd4
    0x69, 0x6e, 0x67, 0x60, 0x75, 0x72, 0x7b, 0x7c
    0x51, 0x56, 0x5f, 0x58, 0x4d, 0x4a, 0x43, 0x44
    0x19, 0x1e, 0x17, 0x10, 0x05, 0x02, 0x0b, 0x0c
    0x21, 0x26, 0x2f, 0x28, 0x3d, 0x3a, 0x33, 0x34
    0x4e, 0x49, 0x40, 0x47, 0x52, 0x55, 0x5c, 0x5b
    0x76, 0x71, 0x78, 0x7f, 0x6a, 0x6d, 0x64, 0x63
    0x3e, 0x39, 0x30, 0x37, 0x22, 0x25, 0x2c, 0x2b
    0x06, 0x01, 0x08, 0x0f, 0x1a, 0x1d, 0x14, 0x13
    0xae, 0xa9, 0xa0, 0xa7, 0xb2, 0xb5, 0xbc, 0xbb
    0x96, 0x91, 0x98, 0x9f, 0x8a, 0x8d, 0x84, 0x83
    0xde, 0xd9, 0xd0, 0xd7, 0xc2, 0xc5, 0xcc, 0xcb
    0xe6, 0xe1, 0xe8, 0xef, 0xfa, 0xfd, 0xf4, 0xf3
};

uint8_t Crc8( const uint8_t * pBuffer, uint16_t length )
{
    uint8_t crc = 0;

    while (length > 0) {
        length--;
        crc = CrcTable[crc ^ *pBuffer];
        pBuffer++;
    }

    return crc;
}
```

Appendix C – Setup a static IP address

The GT125 can be configured to use a static IP address instead of the default dynamic address.

If you're having trouble connecting to the internal configuration of the GT125, or if you have multiple GT125 on the network, try connecting the GT125 and the computer directly using the Ethernet cable and disable WiFi connections.

Access the GT125 network settings

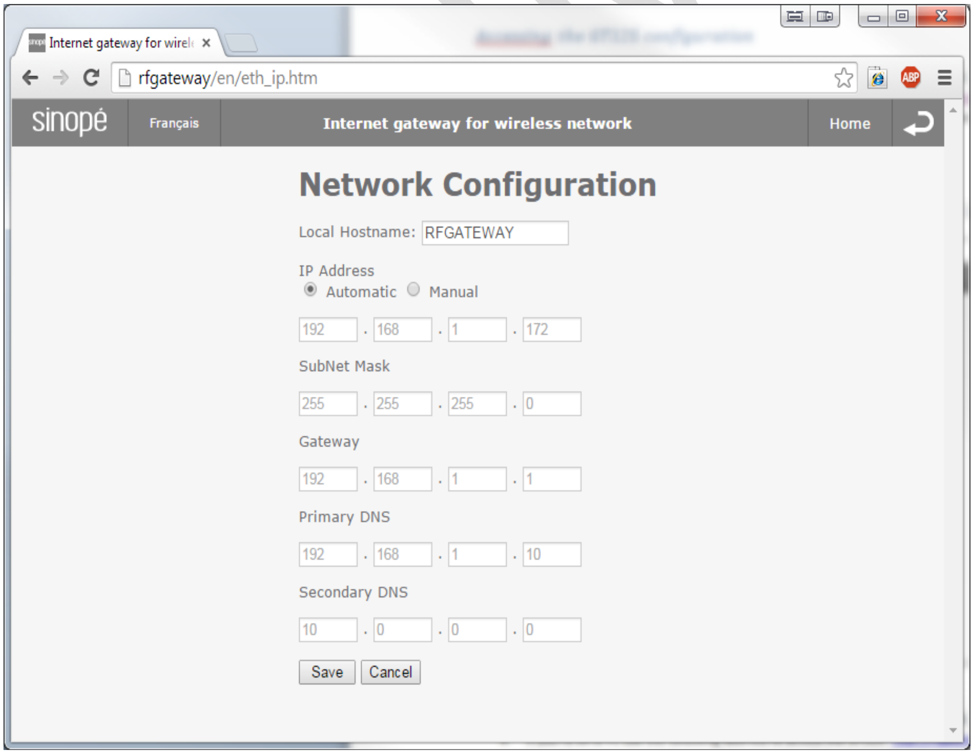
- Open your web browser from a computer on the same network as the GT125.
  - If you're on a PC use the following address to access the GT125: http://rfgateway/en/eth_ip.htm
  - If you're on a MAC or iOS use the following address to access the GT125: http://rfgateway.local/en/eth_ip.htm
- If / when asked for login information, enter the GT125 password (this is NOT your neviweb.com username and password)
  - The default password is "admin" (without the quotes)
  - Username should be left blank.

Network name

- Local hostname is the "friendly" name to access this configuration without using the numeric IP address.
  Unless you have multiple GT125 on the network, this should be left as is.
  Use only alpha-numeric characters.

IP configuration

- Decide between Automatic (dynamic) or Manual (static) IP address.
- If setting to Manual, enter the desired IP address, subnet mask, gateway and DNS servers.
- Click the "Save" button.
- Wait for the configuration page to reload.


Internet gateway for wirel
rfgateway/en/eth_ip.htm
sinopé
Français
Internet gateway for wireless network
Home

Network Configuration

Local Hostname: RFGATEWAY

IP Address
● Automatic ○ Manual

192 . 168 . 1 . 172

SubNet Mask

255 . 255 . 255 . 0

Gateway

192 . 168 . 1 . 1

Primary DNS

192 . 168 . 1 . 10

Secondary DNS

10 . 0 . 0 . 0

Save Cancel