LiveBooster-Heracles-Arduino library

User Manual for LiveBooster-Heracles-Arduino library

Table of contents

1. INTRODUCTION 4
   1.1. Document purpose 4
   1.2. Reference documents 4
   1.3. Glossary 4
2. OVERVIEW 5
   2.1. What is LiveBooster-Heracles-Arduino ? 5
   2.2. Arduino 5
   2.3. Heracles modem 5
3. GETTING STARTED 7
   3.1. Hardware environment and compatibility 7
   3.2. Library examples 7
      3.2.1. "Soft Serial Test" example 7
      3.2.2. "Heracles Sample Basic" example 8
   3.3. Packages dependencies 8
   3.4. Configure library 8
4. DETAILED FEATURES 10
   4.1. Introduction 10
   4.2. TCP connections 10
   4.3. Other functionalities 13
      4.3.1. Basic functions 14
      4.3.2. Power functions 14
      4.3.3. SIM card functions 14
      4.3.4. Generic network functions 15
      4.3.5. Phone call functions 15
      4.3.6. Messaging functions 15
      4.3.7. Location functions 16
      4.3.8. Battery functions 16

Table of figures

Figure 1 - Heracles module 5
Figure 2 - Heracles module embedded on an evaluation / demonstration board.......... 6
Figure 3 - Wiring diagram 7
Figure 4 - attachGPRS sequence diagram 11
Figure 5 - connect sequence diagram 12
Figure 6 - write sequence diagram 12
Figure 7 - read sequence diagram 13
Figure 8 - close sequence diagram 13

1. Introduction

1.1. Document purpose

This document is a manual for the **LiveBooster-Heracles-Arduino** library for Arduino presenting the following:

- Overview
- Getting started
- Detailed Features

1.2. Reference documents

#	Origin	Title
1	SIMCom	SIM800 Series_AT Command Manual_V1.10
2	SIMCom	SIM800 Series_SSL_Application Note_V1.02
3	SIMCom	Heracles_AT Command Manual_V1.01

1.3. Glossary

DNS	Domain Name System
GPRS	General Packet Radio Service
HTTP	Hypertext Transfer Protocol
IoT	Internet Of Things
MQTT	Message Queuing Telemetry Transport
PCB	Printed Circuit Board
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
UART	Universal Asynchronous Receiver Transmitter

2. Overview

2.1. What is LiveBooster-Heracles-Arduino ?

LiveBooster-Heracles-Arduino is an Arduino library for Heracles modem, based on the TinyGSM library. It is released under the GNU Lesser General Public License (LGPL-3.0).

This library provides an Arduino standard Client interface, so that it is easy to integrate with lots of usages based on TCP (MQTT, HTTP, ...).

As an example, this library can be used with the IoTSoftBox library to connect devices to Live Objects server, but it is not limited to this usage.

2.2. Arduino

Arduino is a well-known platform for educational purpose. It is also widely used for testing and prototyping projects.

2.3. Heracles modem

The connected module **Heracles** results from the fruitful partnership between **Orange** and **EBV Elektronik**, combining Orange's assets in cellular connectivity and EBV's expertise on electronic market in an "all inclusive" solution, available off the shelf.

Figure 1 - Heracles module

Designed to simplify the IoT device making process, **Heracles** provides a ready to use cellular module embedding a SIM card with a prepaid data plan, valid for a long term period, at no additional cost and without any monthly fee.

The **Heracles** module, also called modem in this document, can be used embedded on an EBV evaluation/development board, or directly embedded on another connected device.

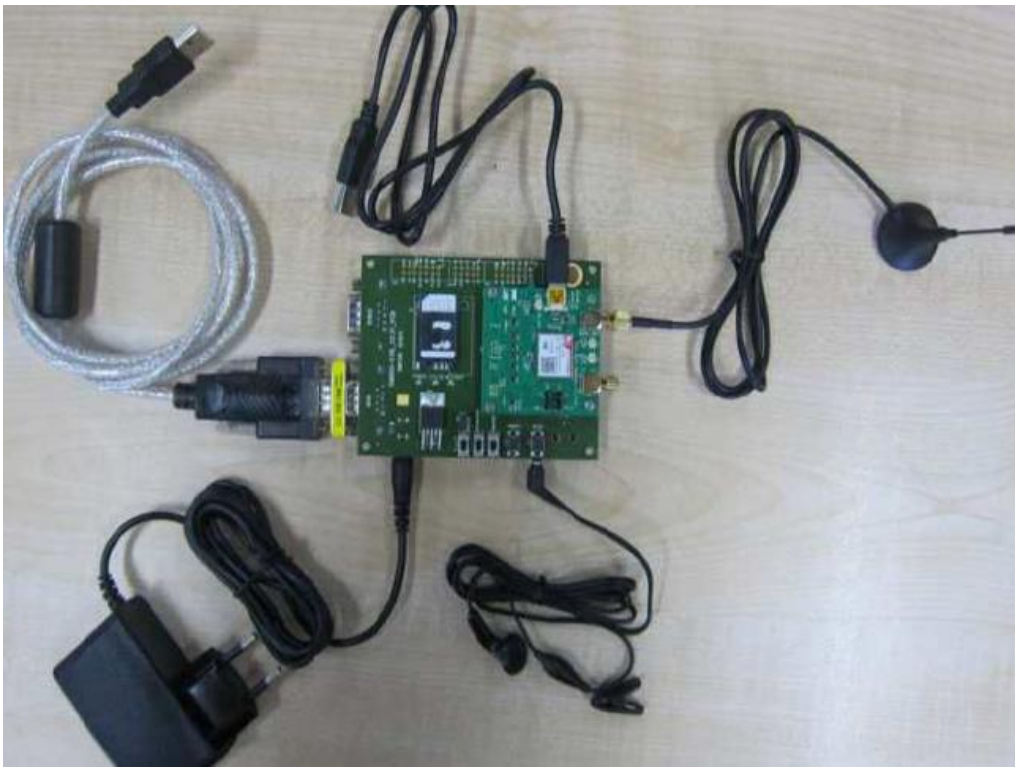


Figure 2 - Heracles module embedded on an evaluation / demonstration board

The LiveBooster-Heracles-Arduino library helps developers to make easy usage of the Heracles module.

3. Getting started

3.1. Hardware environment and compatibility

As for today, we are compatible with the following boards:

- Mediatek LinkIt One
- Arduino MEGA ADK
- Other Arduino boards should be compatible but were not tested.

The Arduino board communicates with the Heracles modem through a standard serial UART interface.

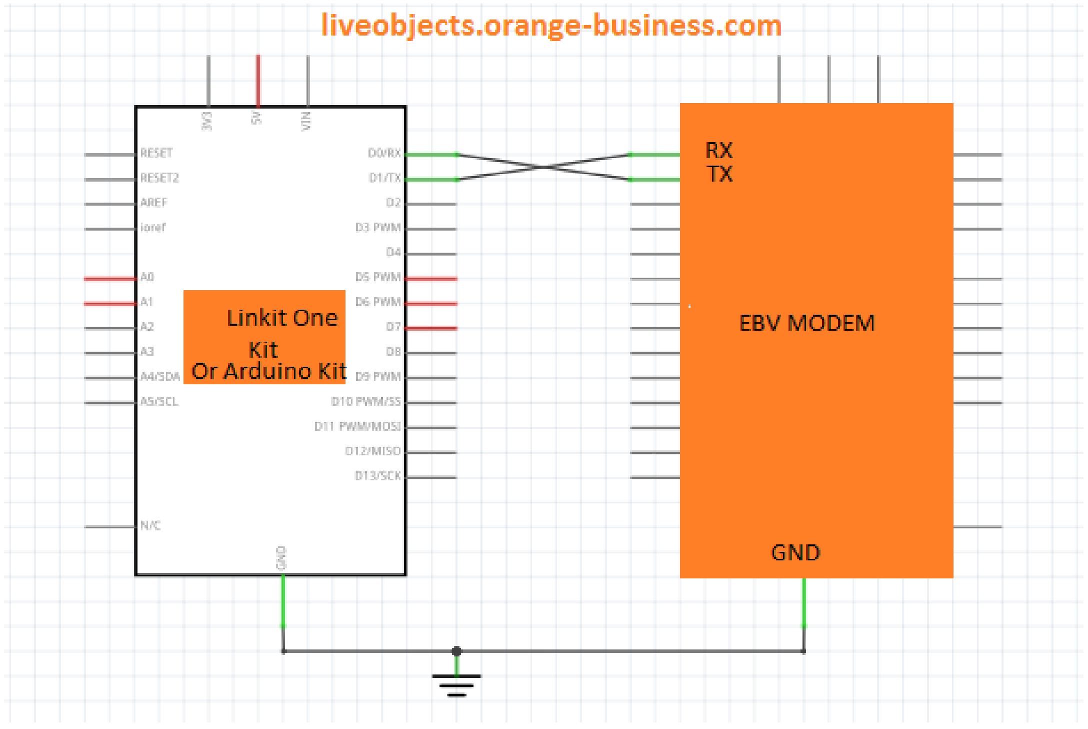


Figure 3 - Wiring diagram

3.2. Library examples

A good way to discover **LiveBooster-Heracles-Arduino** library features is to use our examples.

First of all, the library shall be imported into your Arduino IDE: refer to guideline https://www.arduino.cc/en/Guide/Libraries#toc4.

3.2.1. "Soft Serial Test" example

This example sketch doesn't really use the **LiveBooster-Heracles-Arduino** library: it should be used to check correct communication between the Arduino board and the Heracles modem.

All AT commands sent on Arduino Serial Monitor are transmitted to modem, and answers from modem are transmitted to Arduino Serial Monitor.

AT commands are described by **SIMCom** documents (reference documents 1, 2 and 3).

If this example is not working, you should check:

- That your modem is correctly powered. Try to reset it.
- Hardware connection between the modem and your board (Ground, Rx, Tx).
- Serial UART and baud rates used in example sketch and in Arduino Serial monitor.

3.2.2. "Heracles Sample Basic" example

This example sketch uses the **LiveBooster-Heracles-Arduino** library to:

- Connect to website arduino.cc through a TCP connection.
- Get file http://www.arduino.cc/asciilogo.txt.

Finally, the downloaded file is printed on Arduino Serial Monitor.

3.3. Packages dependencies

The **LiveBooster-Heracles-Arduino** library doesn't require any additional package or library dependency.

3.4. Configure library

The **LiveBooster-Heracles-Arduino** library doesn't require any prior configuration before use, but the following parameters can be modified according to user needs:

- GSM_MUX_COUNT

  This parameter defines the number of maximum simultaneous TCP connections managed by the library.

  The default value is 2 (two simultaneous TCP connections are required for a **Live Objects** usage: one for the MQTT connection, and a second one to get resource files).

  This parameter is defined in source file **HeraclesGsmModem.h**:

  ```
  #define GSM_MUX_COUNT 2
  ```

- dnsEnabled

  This parameter is provided at HeraclesGsmModem object declaration:

```
HeraclesGsmModem(Stream& stream, bool dnsEnabled = true)
```

The default value is **true**. When set to **false**, the DNS is not configured on Heracles modem (the command AT+CDNSCFG is not sent to the modem).

- mux
  This parameter is provided at GsmClient object declaration:

  ```
  GsmClient(HeraclesGsmModem& modem, uint8_t mux = 0, bool sslEnabled = true)
  ```

  The default value is **0**. It allows managing simultaneous TCP connections.
  It shall always be less than the GSM_MUX_COUNT value (see below).

- sslEnabled
  This parameter is provided at GsmClient object declaration:

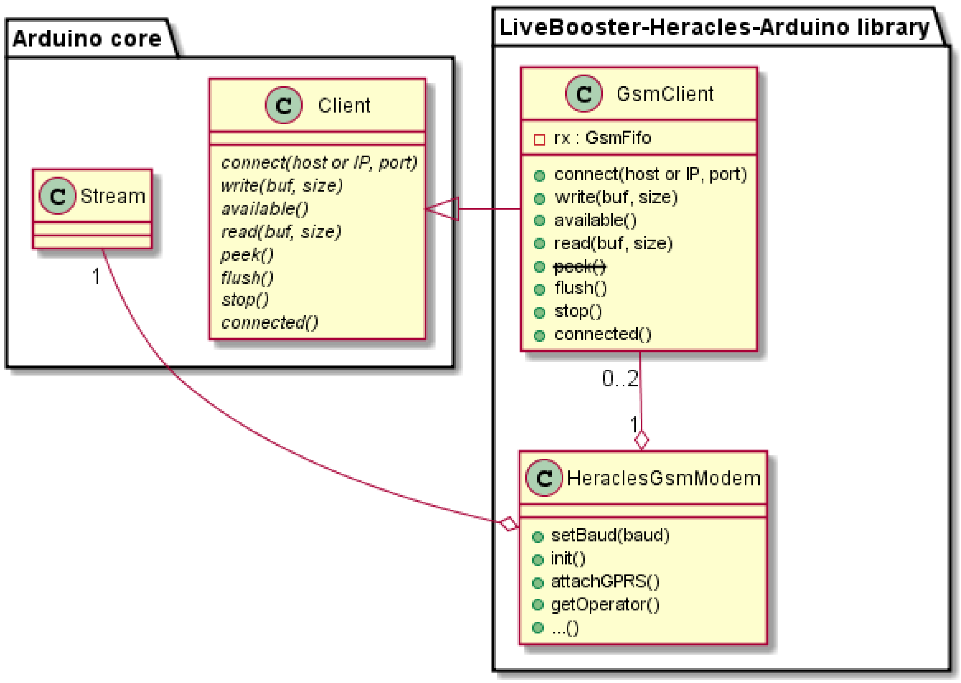
  ```
  GsmClient(HeraclesGsmModem& modem, uint8_t mux = 0, bool sslEnabled = true)
  ```

  The default value is **true**. It sets if the TCP connection shall be encrypted with the modem Secure Socket Layer (**SSL**) or not. For example, it shall be set to **true** to initiate a **HTTPS** or **MQTTS** connection, and set to **false** to initiate a **HTTP** or **MQTT** connection.

4. Detailed Features

4.1. Introduction

The **LiveBooster-Heracles-Arduino** library provides an Arduino standard **Client** interface, and needs an Arduino standard **Stream** interface (typically the UART Serial interface connected to the modem).



The library communicates with the Heracles modem by using **AT commands**, described by **SIMCom** documents (reference documents 1, 2 and 3).

4.2. TCP connections

The main functionality of the library is to ease the communication of the user application with a distant server through a TCP connection.

In following diagrams, only nominal cases (when no error occurs) are described. Typically when an error is returned by the modem, the library method will return an error code to the user application.

Before trying to open a TCP communication, the user application shall request the modem to attach to the GPRS network.

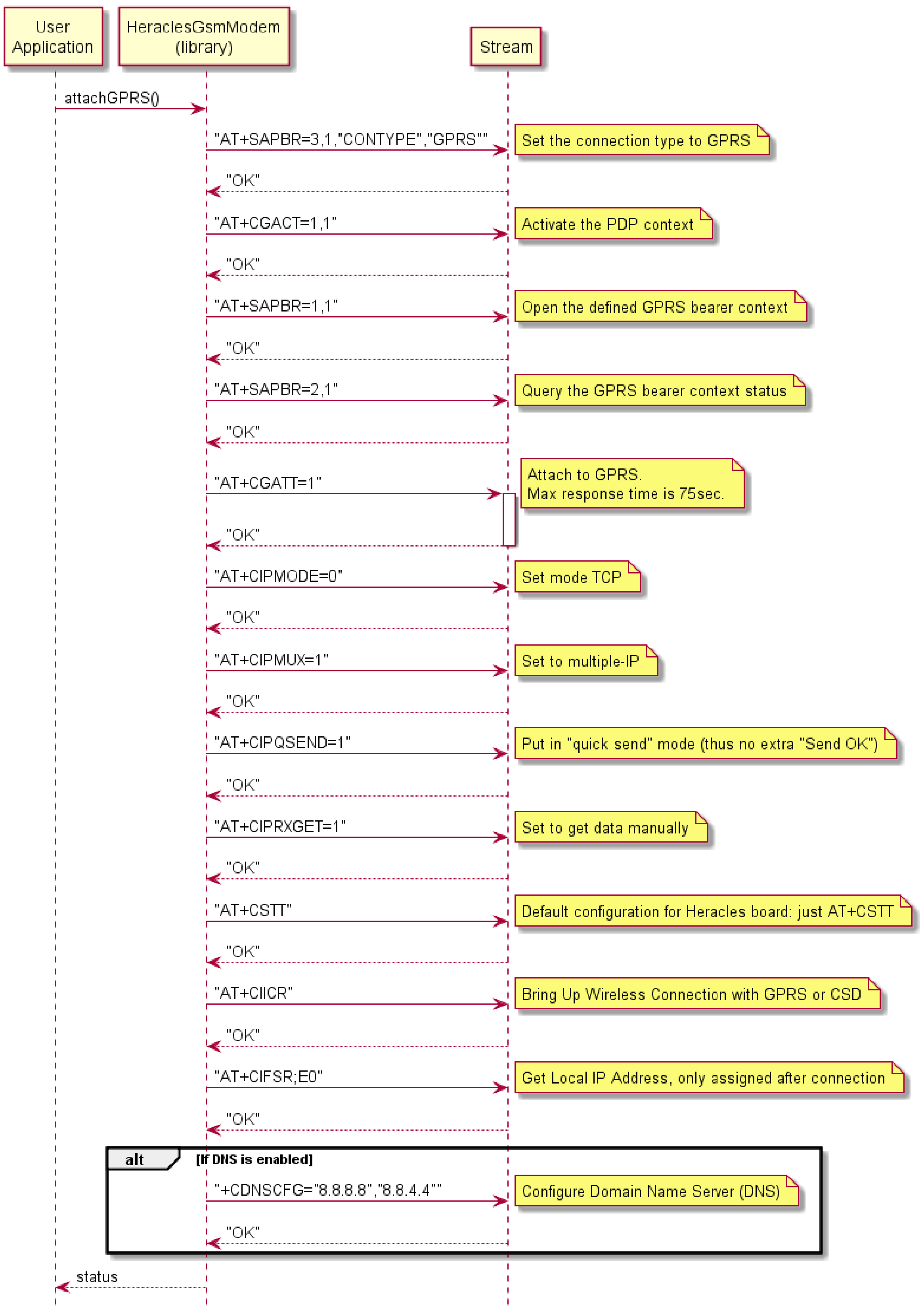

User Application
HeraclesGsmModem (library)
Stream
attachGPRS()
"AT+SAPBR=3,1,"CONTYPE","GPRS""
Set the connection type to GPRS
"OK"
"AT+CGACT=1,1"
Activate the PDP context
"OK"
"AT+SAPBR=1,1"
Open the defined GPRS bearer context
"OK"
"AT+SAPBR=2,1"
Query the GPRS bearer context status
"OK"
"AT+CGATT=1"
Attach to GPRS.
Max response time is 75sec.
"OK"
"AT+CIPMODE=0"
Set mode TCP
"OK"
"AT+CIPMUX=1"
Set to multiple-IP
"OK"
"AT+CIPQSEND=1"
Put in "quick send" mode (thus no extra "Send OK")
"OK"
"AT+CIPRXGET=1"
Set to get data manually
"OK"
"AT+CSTT"
Default configuration for Heracles board: just AT+CSTT
"OK"
"AT+CIICR"
Bring Up Wireless Connection with GPRS or CSD
"OK"
"AT+CIFSR;E0"
Get Local IP Address, only assigned after connection
"OK"
alt
[If DNS is enabled]
"+CDNSCFG="8.8.8.8","8.8.4.4""
Configure Domain Name Server (DNS)
"OK"
status

Figure 4 - attachGPRS sequence diagram

Then the user application can initiate a TCP connection. If SSL is enabled, this step includes the SSL handshake step (performed by the Heracles modem).

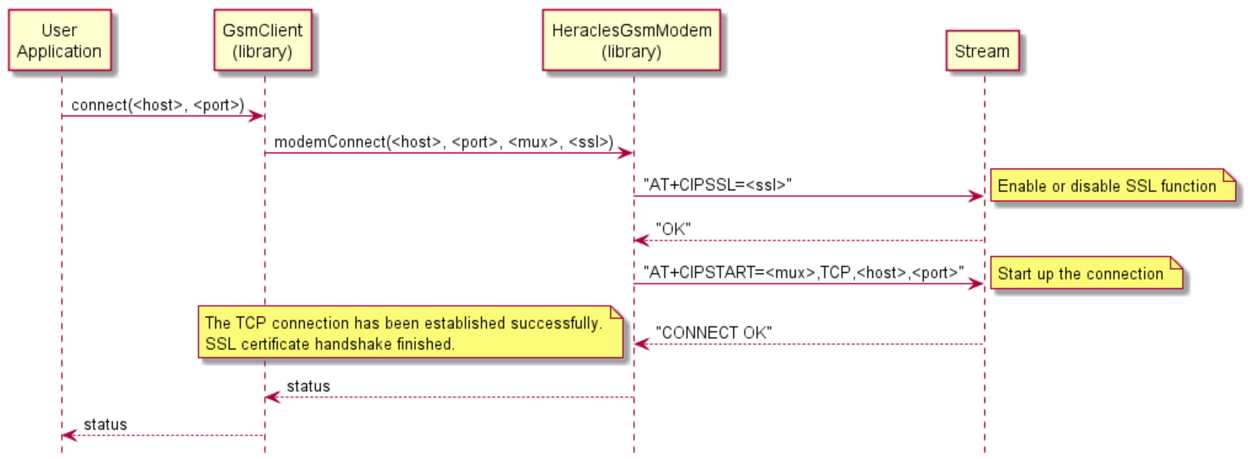

User Application
GsmClient (library)
HeraclesGsmModem (library)
Stream
connect(<host>, <port>)
modemConnect(<host>, <port>, <mux>, <ssl>)
"AT+CIPSSL=<ssl>"
Enable or disable SSL function
"OK"
"AT+CIPSTART=<mux>,TCP,<host>,<port>"
Start up the connection
The TCP connection has been established successfully. SSL certificate handshake finished.
"CONNECT OK"
status
status

Figure 5 - connect sequence diagram

Once the TCP connection is opened, the user application can read and write bytes to/from the TCP layer:

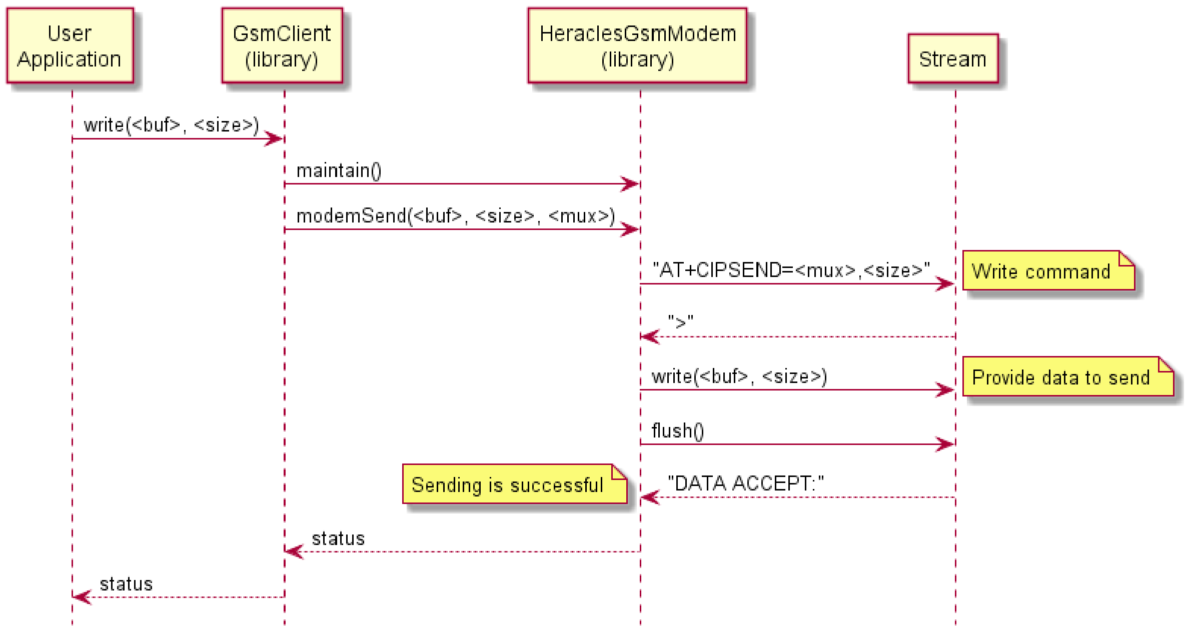


Figure 6 - write sequence diagram

When SSL is enabled, the SSL encryption/decryption is managed directly by the Heracles modem. It means that data on the serial link between the Arduino board and the modem are not encrypted: if security is a major need for the user project, this link should be protected (for example, not accessible on a surface PCB layer).

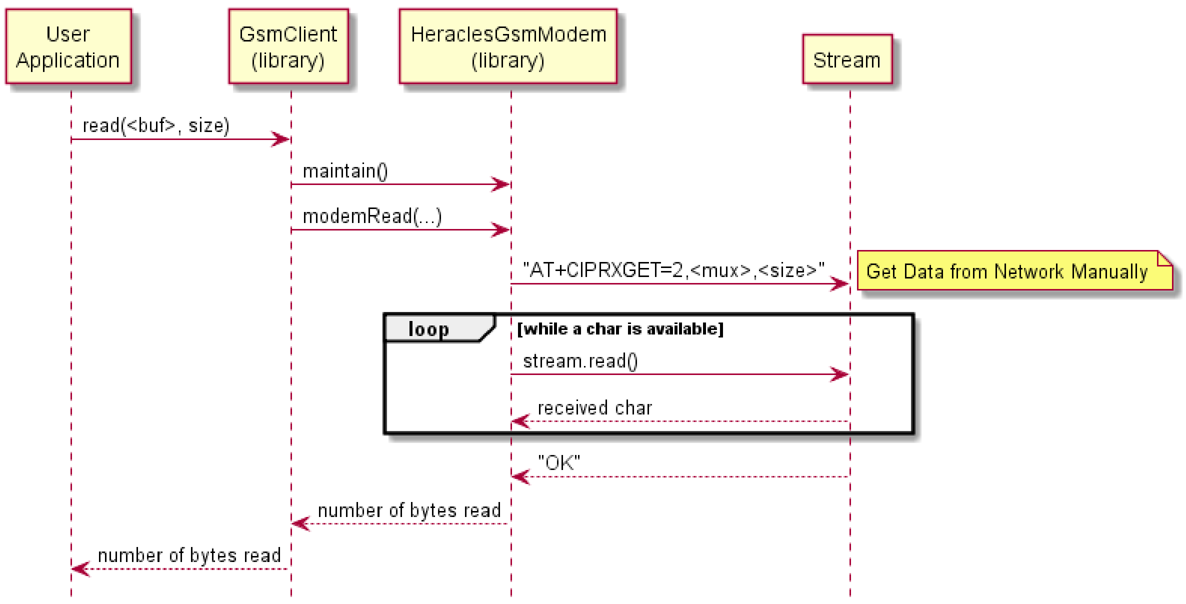


Figure 7 - read sequence diagram

It is important to note that the read() function is not working on a closed TCP connection, even if all data have been received from server before connection closing. It means that if the user application doesn't read available data as soon as they are available, the TCP connection shall be kept alive. In the case of a HTTP GET request for example, the "Connection: keep-alive" key should be sent to the server.

If not closed by the server, the user application can finally request the connection closure.

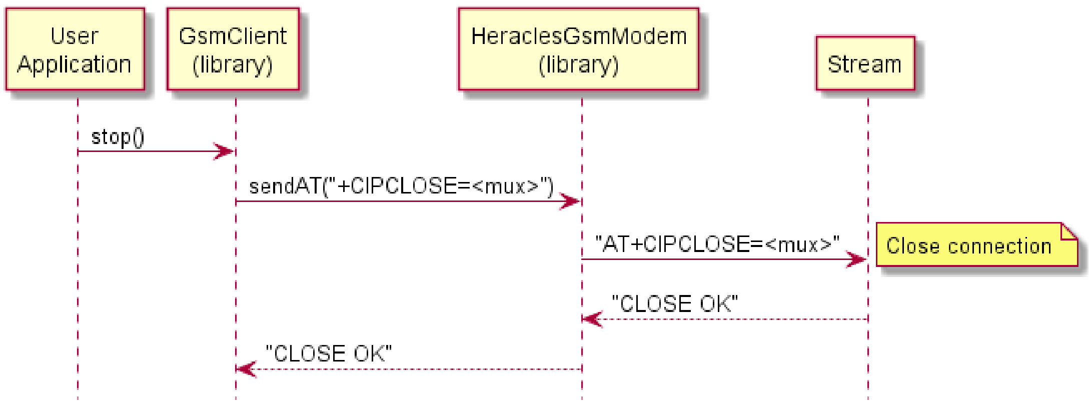


Figure 8 - close sequence diagram

4.3. Other functionalities

The library provides functions to use the Heracles modem capabilities:

4.3.1. Basic functions

Function	Description	Parameters	Return
init()	Initialize modem	Void	boolean
setBaud(unsigned long baud)	Set modem baud rate	unsigned long	Void
getModemInfo()	Get information about modem	Void	String

4.3.2. Power functions

Function	Description	Parameters	Return
restart()	Restart modem	Void	Boolean
poweroff()	Power down	Void	Boolean
radioOff()	Minimum functionality	Void	Boolean
sleepEnable(bool enable = true)	Sleep	Boolean	Boolean

4.3.3. SIM card functions

Function	Description	Parameters	Return
setInternalSim()	Use internal Heracles SIM card	Void	Boolean
setExternalSim()	Use external SIM card	Void	Boolean
simUnlock(const char *pin)	Unlock SIM card	Char	Boolean
getSimCCID()	Get CCID	Void	String
getIMEI()	Get IMEI	Void	String
getSimStatus(unsigned long timeout = 10000L)	SIM Status	Unsigned long	SimStatus : *SIM_ERROR* *SIM_READY* *SIM_LOCKED*
getRegistrationStatus()	Registration type	Void	RegStatus : *REG_UNREGISTERED* *REG_SEARCHING* *REG_DENIED* *REG_OK_HOME* *REG_OK_ROAMING* *REG_UNKNOWN*

getOperator()	Get operator	Void	String

4.3.4. Generic network functions

Function	Description	Parameters	Return
getSignalQuality()	Get signal quality	Void	Int
waitForNetwork(unsigned long timeout = 60000L)	Wait for network	Unsigned long	Boolean
attachGPRS(const char* apn, const char* user, const char* pwd)	Attach to GPRS with personal data; for external SIM	const char* apn, const char* user, const char* pwd	Boolean
attachGPRS()	Attach to GPRS; for internal SIM	Void	Boolean
gprsDisconnect()	Disconnect from GPRS	Void	Boolean
isGprsConnected()	Test if GPRS connected or not	Void	Boolean
getLocalIP()	Get local IP address	Void	String

4.3.5. Phone call functions

Function	Description	Parameters	Return
setGsmBusy(bool busy = true)	Gsm busy	boolean	Boolean
callAnswer()	Call answer	Void	Boolean
callNumber(const String& number)	Call number	String number	Boolean
callHangup()	Call hang-up	Void	Boolean

4.3.6. Messaging functions

Function	Description	Parameters	Return
sendUSSD(const String& code)	Send USSD code	String	String
sendSMS(const String& number, const String& text)	Send SMS	String, String	Boolean

4.3.7. Location functions

Function	Description	Parameters	Return
getGsmLocation()	Get GSM location	Void	String

4.3.8. Battery functions

Function	Description	Parameters	Return
getBattVoltage()	Get battery voltage	Void	uint16_t
getBattPercent()	Get battery percent	Void	Int