





*SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE*

**Notice**

While reasonable efforts have been made to assure the accuracy of this document, Telit assumes no liability resulting from any inaccuracies or omissions in this document, or from use of the information obtained herein. The information in this document has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies or omissions. Telit reserves the right to make changes to any products described herein and reserves the right to revise this document and to make changes from time to time in content hereof with no obligation to notify any person of revisions or changes. Telit does not assume any liability arising out of the application or use of any product, software, or circuit described herein; neither does it convey license under its patent rights or the rights of others.

It is possible that this publication may contain references to, or information about Telit products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that Telit intends to announce such Telit products, programming, or services in your country.

**Copyrights**

This instruction manual and the Telit products described in this instruction manual may be, include or describe copyrighted Telit material, such as computer programs stored in semiconductor memories or other media. Laws in the Italy and other countries preserve for Telit and its licensors certain exclusive rights for copyrighted material, including the exclusive right to copy, reproduce in any form, distribute and make derivative works of the copyrighted material. Accordingly, any copyrighted material of Telit and its licensors contained herein or in the Telit products described in this instruction manual may not be copied, reproduced, distributed, merged or modified in any manner without the express written permission of Telit. Furthermore, the purchase of Telit products shall not be deemed to grant either directly or by implication, estoppel, or otherwise, any license under the copyrights, patents or patent applications of Telit, as arises by operation of law in the sale of a product.

**Computer Software Copyrights**

The Telit and 3rd Party supplied Software (SW) products described in this instruction manual may include copyrighted Telit and other 3rd Party supplied computer programs stored in semiconductor memories or other media. Laws in the Italy and other countries preserve for Telit and other 3rd Party supplied SW certain exclusive rights for copyrighted computer programs, including the exclusive right to copy or reproduce in any form the copyrighted computer program. Accordingly, any copyrighted Telit or other 3rd Party supplied SW computer programs contained in the Telit products described in this instruction manual may not be copied (reverse engineered) or reproduced in any manner without the express written permission of Telit or the 3rd Party SW supplier. Furthermore, the purchase of Telit products shall not be deemed to grant either directly or by implication, estoppel, or otherwise, any license under the copyrights, patents or patent applications of Telit or other 3rd Party supplied SW, except for the normal non-exclusive, royalty free license to use that arises by operation of law in the sale of a product.









- 4.4.1. Clock Date and Time ..... 52
  - 4.4.1.1. Adjusting the Date and Time ..... 52
  - 4.4.1.2. Reading the Current Date and Time ..... 53
- 4.4.2. Alarm Function ..... 54
  - 4.4.2.1. Setting the Alarm Time and Behavior ..... 54
  - 4.4.2.2. Stopping the Alarm Activity ..... 56
    - 4.4.2.2.1. Exit From the Alarm Status and Shut Down ..... 56
    - 4.4.2.2.2. Exit From the Alarm Status and Enter the Normal Operating Mode ..... 56
  - 4.4.2.3. Querying the Alarm Status ..... 57
    - 4.4.2.3.1. Alarm Operation Example ..... 57
- 4.5. Multi-Socket Data Session ..... 59
  - 4.5.1. Connecting to an Echo server using a TCP connection ..... 59
    - 4.5.1.1. Ensure that the Correct NAI Profile is selected and correct Mobile IP setting is selected 59
      - 4.5.1.1.1. VERIZON ..... 59
      - 4.5.1.1.2. SPRINT & AERIS ..... 60
    - 4.5.1.2. Turn on Data Release Notification Message (#DREL) ..... 61
    - 4.5.1.3. Configure the Socket Parameters as Needed ..... 61
    - 4.5.1.4. Verify that no socket has a current active session ..... 61
    - 4.5.1.5. Activate context ..... 62
    - 4.5.1.6. Connect socket to echo server (XXX.XXX.XXX.XXX) using TCP ..... 62
    - 4.5.1.7. Ensure that Socket Session is Still Open ..... 63
    - 4.5.1.8. Restore socket session ..... 63
    - 4.5.1.9. End socket session ..... 63
    - 4.5.1.10. Deactivate context ..... 64
  - 4.5.2. Sending and Receiving data to/from an Echo Server while in Command Mode ..... 64
    - 4.5.2.1. Turn on Data Release Notification Message (#DREL) ..... 64
    - 4.5.2.2. Turn off the display of the Escape Character (“+++”) ..... 64
    - 4.5.2.3. Activate context ..... 64
    - 4.5.2.4. Connect socket to echo server (XXX.XXX.XXX.XXX) using UDP ..... 64
    - 4.5.2.5. Send Data in Command Mode ..... 65
    - 4.5.2.6. Check Socket Information, specifically the buffer size ..... 65
    - 4.5.2.7. Receive Data in Command Mode ..... 65
    - 4.5.2.8. Close socket connection ..... 66
    - 4.5.2.9. Deactivate context ..... 66
- 4.6. FTP Operations ..... 66
  - 4.6.1. Opening and Closing an FTP Connection ..... 67



- 4.6.2. Setting the FTP Transfer Type ..... 67
- 4.6.3. FTP File transfer to the server ..... 68
- 4.6.4. FTP File download from the server ..... 70
- 4.7. Phone as Modem (PAM)..... 72
  - 4.7.1. External IP Stack..... 72
- 4.8. Email ..... 73
  - 4.8.1. General Information..... 73
  - 4.8.2. Sending an Email ..... 73
- 4.9. Telit IoT Portal ..... 75
  - 4.9.1. General Information..... 75
  - 4.9.2. Configure deviceWISE parameters..... 75
  - 4.9.3. Connect to M2M service..... 76
  - 4.9.4. Send data to M2M service ..... 77
  - 4.9.5. Receive data from M2M service ..... 79
  - 4.9.6. Query connection status ..... 79
  - 4.9.7. List information on message pending from M2M service ..... 80
- 5. MEID Format and Function..... 81**
  - 5.1. MEID Conversion, HEX to DEC..... 81
- 6. Service and Firmware Update ..... 83**
  - 6.1. Step-by-Step Upgrade Procedure (TFI)..... 83
  - 6.2. XFP Tool ..... 86
    - 6.2.1. Step-by-Step Upgrade Procedure ..... 86
- 7. Acronyms and Abbreviations ..... 90**
- 8. Document History ..... 91**





# 1. Introduction

## 1.1. Scope

This document describes software solutions that can be used to develop a product containing the Telit CL865 module, as follows:

- Describing the basic functions of the module
- Suggesting a proper software control solution for each function
- Describing typical wrong solutions and the most common errors to be avoided

This document is not intended to provide an overall description of all software solutions and all products that may be designed.

The solutions suggested serve as a guide or starting point for correctly developing a product with the Telit CL865 module.

However, avoiding the wrong solutions and most common errors described here should be regarded as mandatory.

## 1.2. Audience

This manual is intended for software and applications developers who design products that integrate the CL865 module.

## 1.3. Contact Information, Support

For general contact, technical support, to report documentation errors and to order manuals, contact Telit's Technical Support Center (TTSC) at:

TS-NORTHAMERICA@telit.com, TS-EMEA@telit.com, TS-APAC@telit.com or use <http://www.telit.com/en/products/technical-support-center/contact.php>

For detailed information about where you can buy Telit modules or for recommendations on accessories and components visit <http://www.telit.com>

To register for product news and announcements or for product questions contact Telit's Technical Support Center (TTSC).

Our aim is to make this guide as helpful as possible. Keep us informed of your comments and suggestions for improvements.

Telit appreciates feedback from the users of our information.

## 1.4. Product Overview

The CL865 is a CDMA-1XRTT wireless module designed to have the same form factor of its GSM/GPRS, UMTS/HSPA counterpart products of the xL865 family: GL865 V3 and UL865 respectively.



This enables integrators and developers to design their applications once and take advantage of the global coverage and service flexibility allowed by the combination of the most prevalent cellular technologies worldwide.

With its ultra-compact design and extended operating temperature range, the Telit CL865 is the perfect platform for low throughput wireless data applications on CDMA 1xRTT network, such as telemetry, telematics, smart metering, health care, tracking applications electronics and security.

### 1.4.1. General Specifications

Parameter	Description
External access	Code division multiple access
CDMA protocol	CDMA2000 1x
Data Rate	1x: 153.6 Kb/s (full-duplex)
Vocoder	EVRC, 13kQCELP
Operating temperature	-40° - +85°

## 1.5. Document Organization

This manual contains the following sections:

Section [Introduction](#) provides a scope for this manual, target audience, contact and support information, and text conventions.

Section [Getting Started](#) contains information on how to get started, including provisioning of service.

Section [Basic Operations](#) describes the basic operations of the module.

Section [Advanced Operations](#) provides in-depth information on some of the more advanced operations and commands of the module.

Section [MEID Format and functions](#) provides information on MEIDs and their representations.

Section [How to update SW](#) provides information on MEIDs and their representations.

Section [Acronyms and Abbreviations](#) provides definitions for all the acronyms and abbreviations used in this guide.









### 3. Basic Operations

#### 3.1. Command Syntax

In the next paragraphs the following notations are used:

- <cr> represents the Carriage Return Character (13)
- <lf> represents the Line Feed Character (10)
- <xx> represents a parameter. (<> characters are only for delimiting the parameter and must not be issued to the terminal).
- [<xx>] represents an optional parameter. ([] characters are only for delimiting the optional parameter and must not be issued to the terminal).

#### 3.2. Command Response Timeout

Every command issued to the Telit CL865 returns a result response if response codes are enabled (default) (see command ATQn). The time needed to process the given command and return the response varies and may depend on the network also. As a result every command is provided with a proper timeout period. If this time elapses without any result from the operation, the ERROR response is reported.

The timeout period is quite short for internal setup commands, but may be very long for commands that interact with the network (or networks).



**NOTE:** If no response is received after the timeout has expired, then try repeating the last command and if still no response is received before the timeout, an Unconditional Shutdown MUST be issued followed by powering ON the device again.

The table below lists all the commands with timeouts that differ from the default 100 ms:

Command	Time-Out (Seconds)
+CGMI	5
+CGMM	5
+CGMR	5
+CGSN	20
+CSQ	5
+GMI	5
+GMM	5
+GMR	5
+GSN	20
+CSMS	5
+CMGF	5
+CSMP	5
+CNMI	5

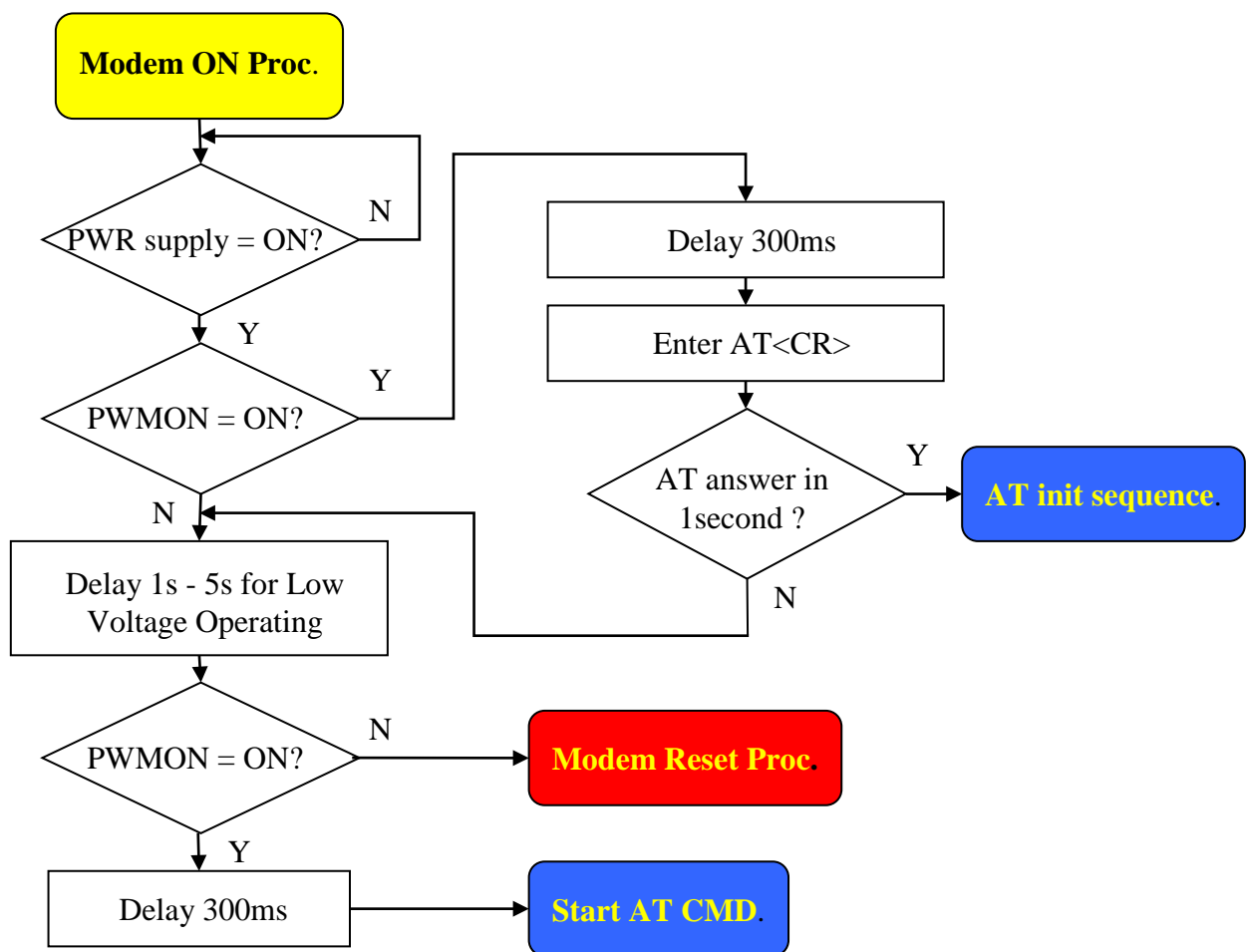




### 3.3. Auto-Turning ON the CL865

To Auto-turn on the CL865, the power supply must be applied on the power pins VBATT and VBATT\_PA, after 5 seconds, the V\_AUX / PWRMON pin will be at the high logic level and the module can be considered fully operating.

The following flow chart shows the proper turn on procedure:



**NOTE:**

The power supply must be applied either at the same time on pins VBATT and VBATT\_PA, or first applied on VBATT\_PA and then on VBATT. The opposite sequence shall be avoided. The reverse procedure applies for powering down the module: first disconnect VBATT, then VBATT\_PA, or both at once.



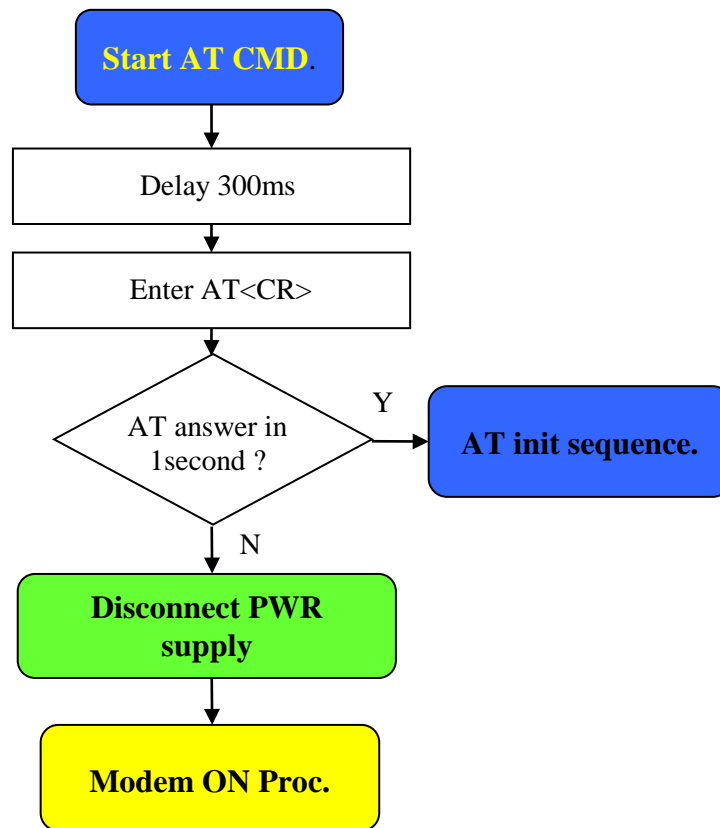




**NOTE:**

In order to prevent a back powering effect it is recommended to avoid having any HIGH logic level signal applied to the digital pins of the CL865 when the module is powered OFF or during an ON/OFF transition.

A flow chart showing the AT commands managing procedure is displayed below:



### 3.4. Turning OFF the CL865

Turning off the device can be done in two ways:

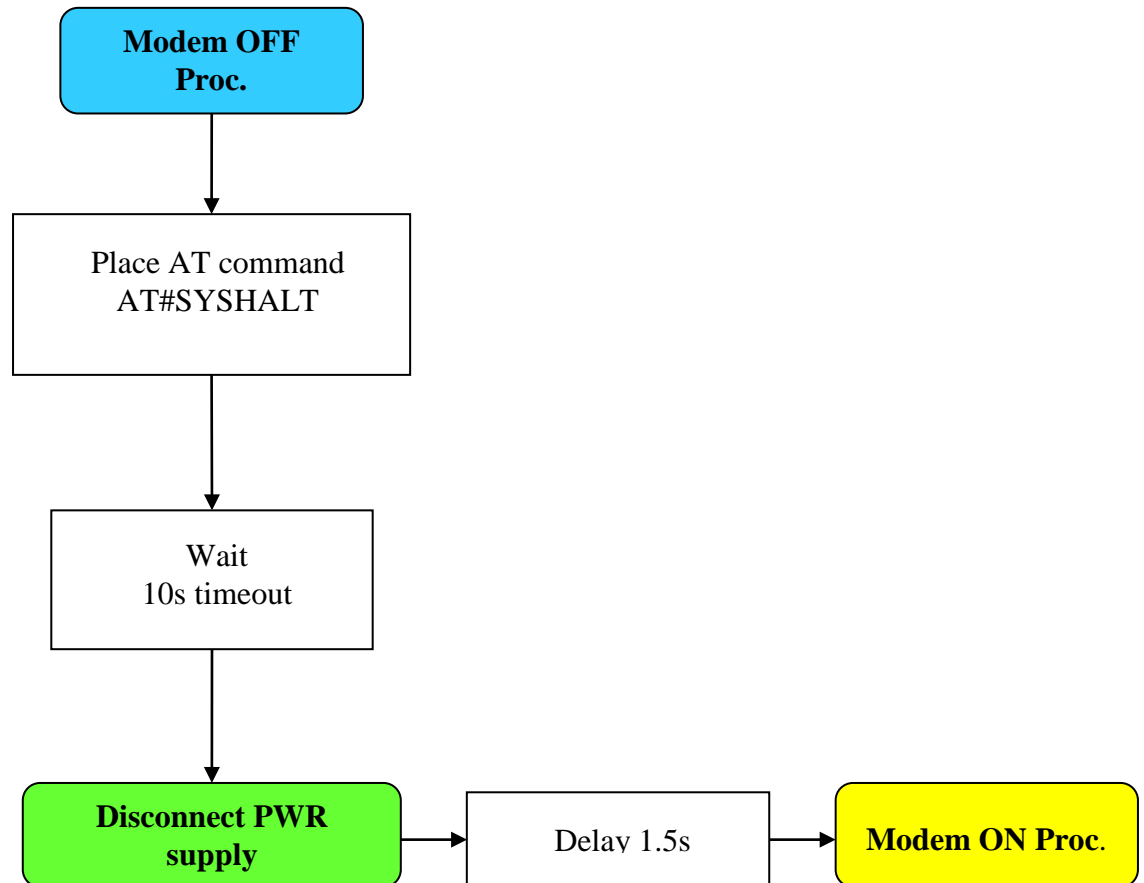
- by software command
- by hardware shutdown

When the device is shut down by software command or by hardware shutdown, it issues to the network a detach request indicating that the device will no longer be reachable.



### 3.4.1. Software Shutdown

The following flow chart shows the proper turnoff procedure:



**NOTE:**

The #SHDN will generate a shutdown of the module with an automatic restart.

### 3.4.2. Hardware Shutdown

**General turn OFF** – disconnect the power supply from the both power pins VBATT and VBATT\_PA at the same time. In this case all parts of the module are in OFF condition, no power consumption is present..

### 3.4.3. Hardware Unconditional restart

The Unconditional restart of the module could be activated using the RESET\* line.







<stat> is the network registration status

- 0 – Not registered, not currently searching for a new network operator.
- 1 – Registered, home network
- 2 – Reserved
- 3 – Registration denied
- 4 – Reserved
- 5 – Registered, roaming

<SID> is the system identification (only displayed when <mode> is set to “2”)

**Example:**

```
AT+CREG?
+CREG: 0,1
```

OK

### 3.5.2.2. Common Air Interface parameters (CDMA)

Once the module has registered on a network, it is possible to query the module for common network information:

1. Issue command:

```
AT#CAI?<cr>
```

2. Wait for response in the format:

```
<System ID>,<Network ID>,<.....>,<Number of Forward Supplemental Channel>
```

where parameter value numbers contain information on:

- 0 - Current System ID
- 1 - Current Network ID
- 2 - Current Base Station ID
- 3 - Current Packet Zone ID
- 4 - Current channel number
- 5 - Current Pilot PN Number
- 6 - Current Mobile Station Protocol Revision
- 7 - Current Base Station Protocol Revision
- 8 - Current In Use Protocol Revision













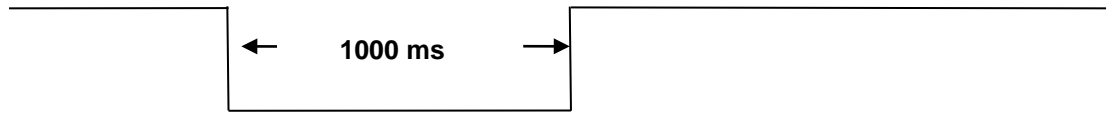






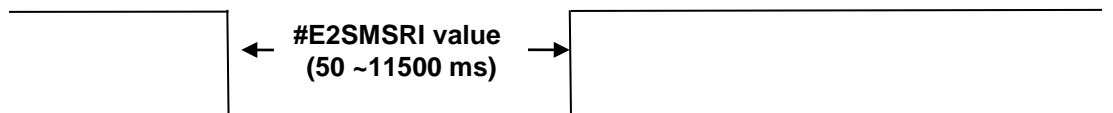






RI Signal for +CNMI=3,1

In case of #E2SMSRI = <x>, the negative going pulse is generated as below once



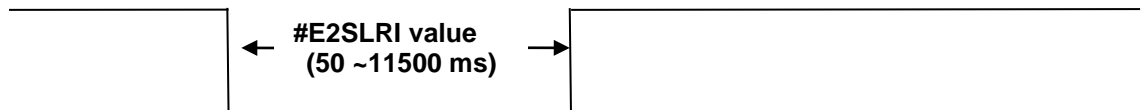
RI Signal for #E2SMSRI



**NOTE:** In case both +CNMI=3,1 and #E2SMSRI=<x> is issued, RI signal behavior by #E2SMSRI is ignored by +CNMI.

#### 4.1.2.3. RI Signal for socket listen

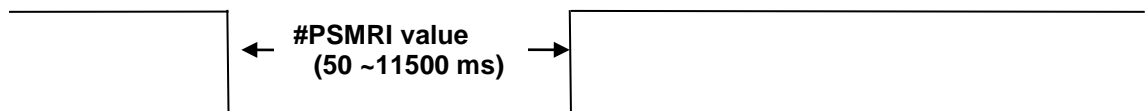
RI Signal is generated, when modem receive TCP connection request from remote client during socket server mode. This signal is the negative going pulse and is generated, one time.



RI Signal for #E2SLRI

#### 4.1.2.4. RI Signal for event in power saving mode

RI Signal for URC message is generated, only when MT is in PSM



RI Signal for #PSMRI



**NOTE:** RI signal for Incoming Call, SMS, Socket Listen and #PSMRI are generated at the same time in PSM, RI signal for #PSMRI will be ignored.



## 4.2. SMS

The Telit CL865 supports the Short Message Service. It is possible to store, delete, write, send and receive a SMS, which is a short text message up to 160 characters long.

### 4.2.1. SMS device setup

Before accessing the Short Message Service, the device has to be properly set up.

#### 4.2.1.1. Select SMS Message Format

Set command selects the “message format” in which SMS will be handled by the module and network.

You can choice between PDU (default) and TEXT formats.

Send command `AT+CMGF=<mode><cr>` where:

<mode> - message format

0 – PDU Mode

1 – TEXT Mode

#### 4.2.1.2. Select New Messages indication behavior

Set command selects the behavior of the device when receiving new messages from the network is indicated to the DTE.

Send command `AT+CNMI=<mode><cr>` where:



**NOTE:** The information below written in *italics* will be present depending on the last setting of `+CSDH`.

<mode> - *unsolicited result codes buffering option*

0 - No SMS-DELIVER indications are routed to the TE.

1 - Indication of the memory location is routed to the TE using the following *unsolicited result code*:

`+CMTI: <memr>,<index>`

where:

<memr> - memory storage where the new message is stored

"ME"





<index> - location on the memory where SM is stored.

2 - New message is routed to the TE using the following unsolicited result code:

**PDU MODE:**

+CMT: ,<length><CR><LF><pdu>

where:

<length> - Length of message

<pdu> - Message data

**TEXT MODE:**

+CMT: <orig\_num>,<callback>,<date>[,<tooa>,<tele\_id>  
,<priority>,<enc\_type>,<length>]<CR><LF><data>

where:

<orig\_num> - Origination number

Notice: If origination number is “6245” the message is from Email.

This interim solution puts the Internet e-mail address into the User Data field and uses a special address (6245) to label the message. A space will be used as a delimiter between the Internet e-mail address and the text of the message in the User Data field.

<callback> - Callback number

<date> - Received date in form as “YYYYMMDDHHMMSS”

<tooa> - Type of callback number

<tele\_id> - Teleservice ID

4097 - page

4098 - SMS message

<priority> - Priority

Note: The priority differs between carriers.

For Sprint and Aeris.Net:

- 0 – Normal (factory default)
- 1 - Interactive
- 2 - Urgent
- 3 - Emergency

For Verizon:

- 0 – Normal (factory default)
- 1 – High



<enc\_type> - Encoding type of message

- 0 - 8-bit Octet
- 2 - 7-bit ASCII
- 4 - 16-bit Unicode

<length> - Length of message

<data> - Message data. (Indicates the new voice mail count, if <tele\_id> is voice mail notification)

### Preparation for the examples

AT+CNMI=?

+CNMI: (0-2)

OK

AT+CNMI=1

OK

AT+CNMI?

+CNMI: 1

OK

+CMTI: "ME",98

AT+CNMI=2

OK

AT+CNMI?

+CNMI: 2

OK

+CMT: "01191775982",20071221163655,19,4098,0,2,8

TEST SMS

#SMSFULL

### Example 1

To eliminate all the unsolicited codes that may be sent when receiving SMS & Status Report, do the following:

1. Send command AT+CNMI= 0<cr>



2. Wait for response OK

**Examples for indicating a new message:**

**Example 2**

If a new SMS delivery (AT+CNMI=1) is received and this new message is stored in the "ME" storage at location number 7; the following unsolicited code (if code is enabled) will be displayed:

+CMTI: "ME",7

**Example 3**

If a new SMS Status Report delivery (AT+CNMI=2) is received and this new message is stored in the "ME" storage at the location number 8; the following unsolicited code will be displayed:

In case of AT+CMGF=0,

+CMT: "",20

068009686167030910281215501002000202858C

In case of AT+CMGF=1 and AT+CSDH=1,

+CMT: "9086167630","9086167630",20091028124045,129,4098,0,2,2

4.2.1.3. **Select SMS Memory and check for memory space**

There are various types of storage where the SMS can be stored; the Telit CL865 provides only "ME" - mobile equipment memory.

The CL865 allows the user to select a different storage for the read-delete, write-send, and reception-saving SMS operations.

1. Send command:

AT+CPMS=<memr>,<memw><cr>

where:

<memr>: memory storage for Read and Delete commands

"ME" only

<memw>: memory to which writing, sending, and receiving operations are made

"ME" only

2. Wait for response in the format:

+CPMS:<memr>,<usedr>,<totalr>,<memw>,<usedw>,<totalw>









The 9 octets from "hellohello" are D19766CDFA32ECD9BC, where **00** are padding bits.

### 4.2.3. Sending a New SMS without Storing It

A new SMS can be sent directly to the network without storing it.

1. send command AT+CMGS="<da>"<cr>  
where:  
<da> - destination address
2. wait for prompt ">"
3. send SMS text (MAX 160 characters)
4. end command with CTRL-Z character (0x1A hexadecimal) or abort command with ESC character (0x1B hexadecimal)
5. wait for response:

Response	Reason	Action
+CMGS: <mr> OK	Message has been successfully sent. <mr> represents the message reference number.	proceed ahead
ERROR	some error occurred	Enable the extended error codes report and retry.
OK	command aborted by user	ESC char was issued

#### Example

Send a new SMS to the destination address number 9194397977.

Assume the device has been set up for text SMS mode as described in the previous paragraphs:

Command:

```
AT+CMGS="9194397977"
```

Response:

```
>
```

Now insert the message text in IRA format (note that the IRA format and ASCII format coincide for the alphabet characters but not for the others).

...The SMS message text to be sent will be inserted here....

Conclude text with the character CTRL-Z.

Response:







**OK**

In this case, the SMS was successfully deleted.

**Example 2:**

Delete a received SMS that was stored to the index position number 7:

Command:

**AT+CMGD=7**

Response:

**OK**

## 4.2.5. Reading an SMS

A new SMS can be read using the following command:

1. Send command **AT+CMGR=<index><cr>**  
**where:**  
<index>: SMS location index
2. Wait for response in the format:  
**+CMGR: <stat>,<orig\_num>,<callback>,<date> [,<tooa>,<tele\_id>,<priority>,<enc\_type>,<length>] <CR><LF><data>**

**where:**

<stat> - status of the message

"REC UNREAD" - new received message unread

"REC READ" - received message read

"STO UNSENT" - message stored not yet sent

"STO SENT" - message stored already sent

<orig\_num> - origination number

<callback> - callback number

<date> - arrival time of the message to the MC

<tooa> - type of number

129 - number in national format







"REC READ" - read message

"STO UNSENT" - stored message not yet sent

"STO SENT" - stored message already sent

<oa/da> - sender number/destination number

< tooa/toda > - type of number <oa/da>

145 - international number (contains "+" character)

129 - national number

<callback> - callback number

<date> - arrival time of the message to the SC

<tooa> - type of number

129 - number in national format

145 - number in international format (contains the "+")

<priority> - Priority

Note: The priority is different with carrier.

For Sprint and Aeris.Net:

0 – Normal (factory default)

1 - Interactive

2 - Urgent

3 - Emergency

For Verizon:

0 – Normal (factory default)

1 - High

<tele\_id> - Teleservice ID

4097: page

4098: SMS message

<enc\_type> - Data Coding Scheme

<length> - Length of message

<data> - Message data. (Indicates the new voice mail count, if <tele\_id> is voice mail notification)



**NOTE:** If the status of the message is "received unread", status in the storage changes to "received read".



### Example

List all the SMSs “received read” that are stored.

Assume the device has been set up for text SMS mode as described in the previous paragraphs.

Command:

```
AT+CMGL="REC READ"
```

Response:

```
+CMGL: 0,"REC READ",,,,,,"01191775982",20071129013659,,4098,,16,12  
TEST MESSAGE  
+CMGL: 1,"REC READ",,,,,,"01191775982",20071217190804,,4098,,16,13  
TEST MESSAGE1  
+CMGL: 2,"REC READ",,,,,,"01191775982",20071221160610,,4098,,16,13  
TEST MESSAGE2  
+CMGL: 3,"REC READ",,,,,,"01191775982",20071224163607,,4098,,16,13  
TEST MESSAGE3
```

OK

## 4.3. Using General Purpose Input/Output Pins

The CL865 provides various General Purpose Input/Output pins. These pins can be configured via AT commands as Inputs, Outputs and some of them as "alternate function".

The "alternate function" is supported by the following pins:

- GPIO6, which can be configured to become an alarm output pin that reflects the alarm status.
- GPIO8, which can be configured to become the STAT\_LED output that reflects the modules current network interaction.

With these pins the application can control external hardware directly using the Telit CL865 pins with little or even no hardware added.



### 4.3.1. GPIO Pin Setup

GPIO pins must be configured to select their direction or alternate function.

#### 4.3.1.1. Setting a GPIO Pin as OUTPUT

When a GPIO is set up as output, the value must be specified that the pin output must take:

1. Send command `AT#GPIO=<pin>,<value>,1<cr>`

where:

<pin> is the GPIO pin number to which the command applies:

- |           |           |
|-----------|-----------|
| 1 - GPIO1 | 5 - GPIO5 |
| 2 - GPIO2 | 6 - GPIO6 |
| 3 - GPIO3 | 7 - GPIO7 |
| 4 - GPIO4 | 8 - GPIO8 |

<value> is the GPIO pin value that the pin will assume:

- 0 – LOW
- 1 – HIGH

2. Wait for response OK



**WARNING:** The #GPIO setting is not saved and will be lost on power off or reset and the GPIOs will be configured as INPUT by default. So at start-up repeat pin initialization commands.

Note that GPIO6 is handled differently from the above. The setting for GPIO6 is maintained even after a shutdown or reset to permit the alarm feature to always work.

#### Example

Set GPIO3 pin as Output with LOW status:

Command:

`AT#GPIO=3,0,1<cr>`

Response:

OK



In this case, the GPIO3 pin was successfully put in output direction and its status has been set to LOW.

#### 4.3.1.2. Setting a GPIO Pin as INPUT

When a GPIO is set as input, a dummy value for the pin state must be specified:

1. Send command `AT#GPIO=<pin>,<dummy_value>,0<cr>` where:  
<pin> is the GPIO pin number to which the command applies:

1 - GPIO1	5 - GPIO5
2 - GPIO2	6 - GPIO6
3 - GPIO3	7 - GPIO7
4 - GPIO4	8 - GPIO8

<value> is a dummy value which can be either:

- 0 – dummy value
- 1 – dummy value

2. wait for response OK.



**WARNING:** The #GPIO setting is not saved and will be lost on power off or reset and the GPIOs will be configured as INPUT by default.

Note that GPIO6 is handled differently from the above. The setting for GPIO6 is maintained even after a shutdown or reset to permit the alarm feature to always work.

#### Example

Set GPIO4 pin as Input:

Command:

`AT#GPIO=4,0,0<cr>`

Response:

OK

In this case, the GPIO4 pin was successfully configured for input direction.







Query the GPIO3 pin for its status:

Command:

```
AT#GPIO=3,2<cr>
```

Response:

```
#GPIO: 0,1
```

OK

In this case, the GPIO3 pin was set to input direction and its status has been measured to be HIGH.

### Example 2

Query the GPIO4 pin for its status:

Command:

```
AT#GPIO=4,2<cr>
```

Response:

```
#GPIO: 1,0
```

OK

In this case, the GPIO4 pin was set to output direction and its current status is LOW.

### Example 3

Query the GPIO6 pin for its status:

Command:

```
AT#GPIO=6,2<cr>
```

Response:

```
#GPIO: 2,0
```

OK

In this case, the GPIO6 pin was set to "alternate function" direction and therefore works as the alarm output.

The reported status = LOW has no meaning.

#### 4.3.2.2. Setting GPIO pin output status

To set the pin status (when pin is set as OUTPUT):

1. Send command `AT#GPIO=<pin>,<value>,1<cr>`

where:



<pin> is the GPIO pin number at which the command applies:

- |           |           |
|-----------|-----------|
| 1 - GPIO1 | 5 - GPIO5 |
| 2 - GPIO2 | 6 - GPIO6 |
| 3 - GPIO3 | 7 - GPIO7 |
| 4 - GPIO4 | 8 - GPIO8 |

<value> is the pin value to be set and can be:

- 0 – LOW
- 1 – HIGH

2. wait for response OK

**Example**

Set the GPIO3 pin HIGH:

Command:

AT#GPIO=3,1,1<cr>

Response:

OK

In this case, the GPIO3 pin was set to output direction and its status has been set to HIGH.

**4.3.2.3. Using the GPIO8 Pin as STAT\_LED (Alternate Function)**

When the GPIO8 is set as STAT\_LED function, the pin shows information on the network service availability and call status.

1. Send command AT#GPIO=8,0,2<cr>
2. Wait for response OK

To save the STAT\_LED setting for the next power cycle use the &W command.

**4.3.2.4. Using GPIO6 Pin as ALARM OUTPUT (Alternate Function)**

When GPIO6 pin is set as alarm output function, the pin reports the alarm state according to the +CALA settings.

To set the pin to an alternate function a dummy value must be specified for the pin state:



1. Send command `AT#GPIO=6,<dummy_value>,2<cr>`

where:

<value> is a dummy value can be either:

- 0 – dummy value
- 1 – dummy value

2. wait for response OK



**TIP:** Remember that the alternate function places the GPIO6 pin permanently in OUTPUT direction and since the GPIO6 pin value is controlled by the internal software, the corresponding function (+CALA) must be properly setup.



**NOTE:** The #GPIO6 direction setting is saved and will be kept after a power off or reset.

### Example

Set GPIO6 pin as ALARM OUTPUT:

command:

`AT#GPIO=6,0,2<cr>`

response:

OK

In this case, the GPIO6 pin was successfully put in alarm output direction.

## 4.4. Clock/Alarm Function

The Telit CL865 provides a Real Time Clock and Alarm embedded in the product. It is therefore possible to set-up the proper time, check the actual time, and set-up an alarm time at which the alarm will be triggered with various behavior depending on the +CALA setting.

The only requirement is that the power input to the Telit CL865 has to be guaranteed without interruptions. Since the CL865 has no backup battery, it will lose the time setting if its power supply is interrupted.

On Alarm trigger the Telit CL865 can:









- mm : two digits minute (00-59)
- ss : two digits seconds (00-59)
- ±zz: signed two digits timezone (-47 - +48)

<type> is the Alarm behavior:

- 0 - Reserved for other equipment use.
- 1 - The MODULE simply wakes up fully operative as in case of a normal startup. If the device is already ON at the alarm time, then it does nothing.
- 2 - The MODULE wakes up in "alarm mode" if at the alarm time it was off, otherwise it remains fully operative. In both cases the MODULE issues an unsolicited code every 3s:
  - +CALA: <text>
 where:
  - <text> is the +CALA optional parameter previously set.
 The device keeps on sending the unsolicited code every 3s until a #WAKE or #SHDN command is received or a 90s timeout occurs. If the device is in "alarm mode" and it does not receive the #WAKE command within 90s then it shuts down (default).
- 3 - The MODULE wakes up in "alarm mode" if at the alarm time it was off, otherwise it remains fully operative. In both cases the MODULE starts playing the alarm tone on the selected path for the ringer (see command #SRP)
  - The device keeps on playing the alarm tone until a #WAKE or #SHDN command is received or a 90s timeout occurs. If the device is in "alarm mode" and it does not receive the #WAKE command within 90s then it shuts down.
- 4 - The MODULE wakes up in "alarm mode" if at the alarm time it was off, otherwise it remains fully operative. In both cases the MODULE brings the pin GPIO6 high, provided its <direction> has been set to alarm output, and keeps it in this state until a #WAKE or #SHDN command is received or a 90s timeout occurs. If the device is in "alarm mode" and it does not receive the #WAKE command within 90s then it shuts down.
- 5 - The MODULE will make both the actions as for <type>=2 and <type>=3.
- 6 - The MODULE will make both the actions as for <type>=2 and <type>=4.
- 7 - The MODULE will make both the actions as for <type>=3 and <type>=4.

<text> - unsolicited alarm code text string. It has a meaning only if <type> equals 2, 5 or 6.

- 2. wait for response OK







If the device is already in normal operating mode (alarm was triggered when the module was already ON), then this command will terminate the alarm activity only.

#### 4.4.2.3. Querying the Alarm Status

When the device wakes up from an alarm time trigger, the module starts the alarm activity but no network activity, permitting some operations to be done by the controlling application without registering the mobile in the network.

To check if the module is in the "alarm status" and therefore no network activity is performed or the device is in the normal operating status:

1. Send command AT#WAKE?<cr>
2. wait for response in the format:  
+WAKE: <status>  
OK

where:

<status> is the operating mode:

- 0 - normal operating mode
- 1 - alarm mode



**NOTE:** If the device is in alarm mode then no network activity takes place, therefore the only commands that are accepted are the #WAKE and the #SHDN.

When the device is in alarm mode, no operation is allowed towards the network, therefore it is not possible to receive or send calls, SMS or any other CDMA services.

#### 4.4.2.3.1. Alarm Operation Example

##### Example

A battery powered meteorological unit that measures the conditions every hour and therefore needs to send a new SMS to the central server every hour indicating the weather status just measured.

The application shall consume the absolute minimum power to achieve the job since the device is placed in a remote location where its battery must last as long as possible. It should shut down completely and wake up every hour just for the time needed to measure and send the weather data before shutting down again.

#### 1. Set up the time in the internal clock (at first time only)



Command:

```
AT+CCLK="02/11/07,12:24:30+01"<cr>
```

Response:

OK

**2. Set up the next alarm in order to raise the GPIO6 pin to power up the controlling application as well.**

Command:

```
AT+CALA="02/11/07,13:24:30+01",0,6,"TIME TO MEASURE & SMS...!"<cr>
```

Response:

OK

**3. Shut down the CL865 module and subsequently the controlling application.**

Command:

```
AT#SHDN<cr>
```

Response:

OK

**4. After one hour**

The CL865 series turns itself ON in "Alarm Mode" and consequently raises the GPIO6 pin which turns on the power to the controlling application and every 3 seconds issues an unsolicited code +CALA: TIME TO MEASURE & SMS...!

**5. Turn on the keep alive line in the controlling application that keeps itself ON.**

**6. Stop the alarm activity in the CL865 (recognized by the +CALA unsolicited code) and bring the CL865 to operating mode.**

Command:

```
AT#WAKE=0<cr>
```

Response:

OK

**7. Take the weather measurement**

**8. Send the SMS about the weather data without storing the SMS**

See section [PDU Mode Breakdown](#) for more information.

**9. Read the current time.**

Command:



AT+CCLK?<cr>

Response:

+CCLK="02/11/07,13:24:47"<cr>

OK

**10. Calculate and set the next alarm in order to raise the GPIO6 pin to power up the controlling application again.**

Command:

AT+CALA="02/11/07,14:24:47+01",0,6,"TIME TO MEASURE & SMS...!"<cr>

Response:

OK

**11. Shut down the CL865 module and consequently the controlling application.**

Command:

AT#SHDN<cr>

Response:

OK

**12. Repeat steps from step 4 to step 11 as a cycle.**

## 4.5. Multi-Socket Data Session

### 4.5.1. Connecting to an Echo server using a TCP connection

#### 4.5.1.1. Ensure that the Correct NAI Profile is selected and correct Mobile IP setting is selected

This procedure is not applicable to CL865-SC project

##### 4.5.1.1.1. VERIZON

**NAI Profile 0 is selected and active**

Select NAI Profile 0

- 1 Send command AT\$QCMIPP?
- 2 Wait for response OK
- 3 Verify that \$QCMIPP: 0 was returned
- 4 If not then send command AT\$QCMIPP=0
- 5 Wait for response OK



Ensure that NAI Profile 1 is enabled

1. Send command AT\$QCMIP?
2. Wait for response OK
3. Verify that \$QCMIP: 1 was returned
4. If not then send command AT\$QCMIP=1
5. Wait for response OK

#### **Enable Mobile IP with SIP fallback**

1. Send command AT\$QCMIP?
2. Wait for response OK
3. Verify that \$QCMIP: 1 was returned  
(Note: Verizon is 1 and Sprint is 2)
4. IF NOT send command AT\$QCMIP=1
5. Wait for response OK

#### 4.5.1.1.2. SPRINT & AERIS

#### **NAI Profile 1 is selected and active**

Select NAI Profile 1

- 1 Send command AT\$QCMIPP?
- 2 Wait for response OK
- 3 Verify that \$QCMIPP: 1 was returned
- 4 If not then send command AT\$QCMIPP=1
- 5 Wait for response OK

Ensure that NAI Profile 1 is enabled

6. Send command AT\$QCMIP?
7. Wait for response OK
8. Verify that \$QCMIP: 1 was returned
9. If not then send command AT\$QCMIP=1
10. Wait for response OK

#### **Enable Mobile IP only**

1. Send command AT\$QCMIP?
2. Wait for response OK



3. Verify that \$QCMIP: 2 was returned
4. IF NOT send command AT\$QCMIP=2
5. Wait for response OK

#### 4.5.1.2. Turn on Data Release Notification Message (#DREL)

1. Send command AT#NOTI=11,1
2. Wait for response OK

#### 4.5.1.3. Configure the Socket Parameters as Needed

1. Send command AT#SCFG=<connId>,<cid>,<pktSz>,<maxTo>,<connTo>,<txTo>  
Where:  
<connId> is the socket connection identifier: 1...6  
<cid> is the PDP context identifier  
1 – specifies a particular PDP context  
<pktSz> is the packet size to be used by TCP/UDP/IP stack for data sending  
0 – automatically chosen by the device  
1..1500 – packet size in bytes  
<maxTo> is the exchange timeout (or socket inactivity timeout); if there is no data exchange within this timeout period the connection is closed.  
0 – no timeout  
n – timeout in seconds (1 to 65535, default is 90)  
<connTo> is the connection timeout; if we can't establish a connection to the remote within this timeout period, an error is raised.  
n – timeout value in hundreds of milliseconds (10 to 1200, default is 600)  
<txTo> is the data sending timeout; after this period data is sent even if they're less than the max packet size  
0 – no timeout  
n – timeout value in hundreds of milliseconds (1 to 255, default is 50)
2. Wait for response OK

#### 4.5.1.4. Verify that no socket has a current active session

1. Send command AT#SS
2. Wait for response OK
3. Check to make sure that all the second parameter values are 0



**Example:**

```
AT#SS
#SS: 1,0
#SS: 2,0
#SS: 3,0
#SS: 4,0
#SS: 5,0
#SS: 6,0
OK
```

4.5.1.5. **Activate context**

1. Send command AT#SGACT=1,1
2. Wait for response OK
3. The IP address should be displayed as #SGACT: XXX.XXX.XXX.XXX

4.5.1.6. **Connect socket to echo server (XXX.XXX.XXX.XXX) using TCP**

1. Send command AT#SD=1,0,"XXXXXX","XXX.XXX.XXX.XXX"
2. Wait for response CONNECT
3. Now data may be entered to be sent to the echo server.



**NOTE:** Typed data will not be displayed until it comes back from the server.

4. Use "+++" to suspend the socket session. If data is returned from the echo server after escaping from the session, SRING: 1 will be received.

**Example**

```
AT#SD=1,0,"XXXXXX","XXX.XXX.XXX.XXX" <--- TCP connection to an Echo server
CONNECT <--- Connected to server
Echo test to server <--- Text entered and only is displayed when the echo
returns to the module.
Used "+++" to escape socket connection.
OK <--- Successfully suspended socket connection using
"+++"
SRING: 1 <--- Notification of data pending on socket 1
```



#### 4.5.1.7. Ensure that Socket Session is Still Open

1. Send command AT#SS
2. Wait for response OK
3. A list of the sockets should be displayed and socket one should have information listed.

#### Example

AT#SS

```
#SS: 1,3,XXX.XXX.XXX.XXX,40509,XXX.XXX.XXX.XXX,10510
```

```
#SS: 2,0
```

```
#SS: 3,0
```

```
#SS: 4,0
```

```
#SS: 5,0
```

```
#SS: 6,0
```

OK

#### 4.5.1.8. Restore socket session

1. Send command AT#SO=1
2. Wait for response CONNECT
3. Once the session is restored the data that was present with the SRING: 1 will be visible and now data can continue to be sent to the echo server.
4. Escape the session by using “+++”

#### 4.5.1.9. End socket session

1. Send command AT#SH=1
2. Wait for response OK
3. Verify session shut down by sending command AT#SS. The list of sockets should not contain any information.



4.5.1.10. **Deactivate context**

1. Send command AT#SGACT=1,0
2. Wait for response OK
3. A #DREL notification should be displayed confirming the data session has been released.

4.5.2. **Sending and Receiving data to/from an Echo Server while in Command Mode**

4.5.2.1. **Turn on Data Release Notification Message (#DREL)**

1. Send command AT#NOTI=11,1
2. Wait for response OK

4.5.2.2. **Turn off the display of the Escape Character (“+++”)**

1. Send command AT#SKIPESC=1
2. Wait for response OK

4.5.2.3. **Activate context**

1. Send command AT#SGACT=1,1
2. Wait for response OK
3. IP address should be displayed as #SGACT: XXX.XXX.XXX.XXX

4.5.2.4. **Connect socket to echo server (XXX.XXX.XXX.XXX) using UDP**

1. Send command AT#SD=1,0,"XXXXXX","XXX.XXX.XXX.XXX"
2. Wait for response CONNECT
3. Now data can be entered and sent to the echo server or go to command mode using “+++”



**NOTE:** Typed data will not be displayed until it comes back from the server.

**Example:**

AT#SD=1,1,"XXXXXX","XXX.XXX.XXX.XXX" <--- UDP connection to an Echo server

CONNECT <--- Connected to server

<--- Use "+++" to enter command mode.





OK <--- Successfully suspended socket connection using "+++", now in command mode

4.5.2.5. **Send Data in Command Mode**

1. Send command AT#SSEND=1
2. Wait for response ">"
3. Enter data/text to be sent
4. "Ctrl+Z" to send data
5. Wait for response OK
6. Wait for response SRING:1

**Example:**

```

AT#SSEND=1                                <--- Send data in command mode to socket #1.
> TEST IN COMMAND MODE □                  <--- "Ctrl+Z" to send
OK

SRING: 1                                   <--- Notification of data arrived
  
```

4.5.2.6. **Check Socket Information, specifically the buffer size**

1. Send command AT#SI
2. Wait for socket information list, check last parameter for number of new bytes just received in buffer
3. Wait for response OK

**Example:**

```

AT#SI
#SI: 1,71,49,0,20                          <--- 20 new bytes in buffer
#SI: 2,0,0,0,0
#SI: 3,0,0,0,0
#SI: 4,0,0,0,0
#SI: 5,0,0,0,0
#SI: 6,0,0,0, 0
  
```

OK

4.5.2.7. **Receive Data in Command Mode**

1. Send command AT#SRECV=1,<maxByte>
  - a. <maxByte> is how many Bytes to be read from the buffer



2. Wait for response #SRECV=1,<maxByte>
3. Wait for “Data” from buffer to be displayed
4. Wait for response OK

**Example:**

```
AT#SRECV=1,20           <--- "#SI" informed us of 20 new bytes in buffer
#SRECV: 1,20
TEST IN COMMAND MODE    <--- Data displayed

OK
```

4.5.2.8. **Close socket connection**

1. Send command AT#SH=1
2. Wait for response OK

4.5.2.9. **Deactivate context**

1. Send command AT#SGACT=1,0
2. Wait for response OK
3. A #DREL notification should be displayed confirming the data session has been released.

## 4.6. FTP Operations

A set of AT Commands is available to support FTP activities. The first command is called #FTPTO (FTP Time-Out) which defines the time-out for FTP operations. The module already has a factory default time set to 10 seconds.

To modify the setting, the syntax is:

AT#FTPTO=[<tout>]

Where:

<tout> - time-out is 100 ms units

100..5000 – hundreds of ms (factory default is 100)



**NOTE:** The parameter is not saved in NVM.

**Example**

Set the timeout to 100 sec.

Command:





### 4.6.3. FTP File transfer to the server

To send a file to the FTP server during an FTP connection, use the command below:

```
AT#FTPPUT=<filename>
```

where:

<filename> - string type, name under which the file is saved on the server (must have the right extension: ex. If the file you are sending is .txt then the <filename> can be test.txt)

The command opens a data connection and starts sending the file to the FTP server to target file <filename>.

If the data connection succeeds, a CONNECT indication is sent, otherwise a NO CARRIER indication is sent.



**NOTE:** Use the escape sequence +++ to close the data connection.



**NOTE:** The command causes an ERROR result code to be returned if no FTP connection has been opened yet.

#### Example

Send a file to the FTP server.

Perform the following steps:

##### 1. Activate the Data Session.

The IP address of the module is provided as a response.

Command:

```
AT#SGACT=1,1<cr>
```

Response:

```
#SGACT: 174.156.82.131
```

```
OK
```

##### 2. Open the FTP connection.

Command:

```
AT#FTPTO=1000<cr> (FTP settings of time-out)
```



Response:

OK

Command:

AT#FTPOPEN="199.188.25.77","user","pass",0<cr>

Response:

OK



**NOTE:** In this case the FTP server port is not specified so the default value 21 is used.

Command:

AT#FTPTYPE=0<cr> (FTP settings of file type)

Response:

OK

### 3. Start the file transfer to the FTP server.

In this example the target file is named "file.txt".

Command:

AT#FTPPUT="file.txt"<cr> (sends the file)

Response:

CONNECT

### 4. Close the data session.

Command:

+++ (escape sequence +++ to close the data connection)

Response:

NO CARRIER

### 5. Close the FTP connection.

Command:

AT#FTPCLOSE<cr> <closing FTP connection>

Response:

OK



## 6. Deactivate the Data Session if required.

Command:

```
AT#SGACT=1,0<cr>
```

Response:

OK

### 4.6.4. FTP File download from the server

To open a data connection and start getting a file <filename> from the FTP server during an FTP connection use the command below.

Command:

```
AT#FTPGET=<filename>
```

Where:

<filename> - string type, file name

The command opens a data connection and starts downloading the <filename> file from the FTP server.

If the data connection succeeds, a CONNECT indication is sent, otherwise a NO CARRIER indication is sent. The file is received on the serial port.



**NOTE:** The command causes an ERROR result code to be returned if no FTP connection has been opened yet.

#### Example

Download a file from the FTP server.

Perform the following steps:

##### 1. Activate the Data Session.

The IP of the module is provided as a response.

Command:

```
AT#SGACT=1,1<cr>
```

Response:

```
#SGACT: 174.156.82.131
```

OK



**2. Open the FTP connection.**

Command:

AT#FTPTO=1000<cr> (FTP settings of time-out)

Response:

OK

Command:

AT#FTPOPEN="199.188.25.77","user","pass",0<cr>

Response:

OK



**NOTE:** In this case the FTP server port is not specified so the default value 21 is used.

Command:

AT#FTPTYPE=0<cr> (FTP settings of file type)

Response:

OK

**3. Change working directory if required.**

Command:

AT#FTPCWD="incoming"

Response:

OK



**NOTE:** In order to get the list of files on the working directory from the server the AT Command AT#FTPLIST should be used.

**4. Start the file transfer from the FTP server:**

In this example the source file on the FTP server is named "file.txt".

Command:

AT#FTPGET="file.txt"<cr> (receives the file)

Response:

CONNECT



5. **Data connection closes automatically with the response** below when the file downloading is terminated:

Response:

NO CARRIER

6. **Close the FTP connection.**

Command:

AT#FTPCLOSE<cr>

Response:

OK

7. **Deactivate the Data Session if required.**

Command:

AT#SGACT=1,0<cr>

Response:

OK

For more information about other available commands on the FTP functionality see the AT Commands Reference Guide.

## 4.7. Phone as Modem (PAM)

### 4.7.1. External IP Stack

Using Microsoft's IP Stack in Windows XP:

1. Click Start
2. Select Control Panel
3. Double-click Network Connects
4. Under "Network Tasks" select "Create a new connection"
5. A "Network Connection Wizard" will appear.
6. Click "Next"
7. Select "Connect to the Internet" than click "Next"
8. Select "Set up my connection manually" than click "Next"
9. Select "Connect using a dial-up modem" than click "Next"





10. Enter “Telit CL865” in the “ISP Name” field and click “Next”
11. Enter “#777” in the “Phone number” field and click “Next”
12. Select your “Connection Availability” and click “Next”
13. Leave all “Account information” fields blank and click “Next”
14. Now that the “New Connection Wizard” has been completed, select to add a shortcut to the desktop if you wish. Click “Finish” to start using this connection
15. Once the wizard is finished a new window will appear titled “Connect Telit CL865”
16. Ensure that “#777” is in the Dial box, if so click “Dial”
17. If everything is successful an internet connection should be established.

## 4.8. Email

### 4.8.1. General Information

The CL865 series only supports SMTP email service and a commercial grade SMTP provider is recommended.

During testing with free SMTP providers not all the accounts were able to send emails even though they might be with the same provider. This could be due to the fact that spammers prefer to use free SMTP email accounts, thus the SMTP servers may block certain messages.

Hotmail and Gmail accounts will not work on the CL865 series because they use HTTP and SSL respectively.

### 4.8.2. Sending an Email

To send an e-mail, do the following.



**NOTE:** The steps from step 1 to step 8 describe the SMTP parameter settings.

Enter SMTP Server:

1. Send command `AT#ESMTP="smtp.mydomain.com"<cr>`
2. Wait for response OK

Enter Sender Email Address:

3. Send command `AT#EADDR=me@email.box.com<cr>`
4. Wait for response OK



Enter Email Authentication User Name:

5. Send command `AT#EUSER="myE-Name"<cr>`
6. Wait for response OK

Enter Email Authentication Password:

7. Send command `AT#EPASSW="myPassword"<cr>`
8. Wait for response OK

Save Email Parameters:

9. Send command `AT#ESAV<cr>`
10. Wait for response OK

Activate Data Connection:

11. Send command `AT#SGACT=1,1<cr>`
12. Wait for response OK
13. IP address should be displayed as `#SGACT: XXX.XXX.XXX.XXX`

Compose and Send an Email:

14. Send command `AT#EMAILD="user@domain.com","email subject",0<cr>`
15. Wait for ">"
16. Compose body of email
17. Press "Ctrl+Z" to send email
18. Wait for response OK meaning email was sent successfully.
19. If an ERROR is returned send command `AT#EMAILMSG<cr>`. This will return information on the failed email attempt

Close Data Connection:

20. Send command `AT#SGACT=1,0<cr>`
21. Wait for response OK
22. A #DREL will only be displayed if `AT#NOTI=11,1` was entered before ending the data connection





<appToken> - The secure application token provided in the Management Portal, typically a string of 16 characters

<security> - Flag indicating if the SSL encryption is enabled

0 : SSL encryption disabled (default)

1 : SSL encryption enabled (not yet implemented and not available for setting)

<heartBeat> - If no packets are received in the number of seconds specified in the heartbeat field, a heartbeat message will be sent to keep the connection alive.

Default : 60

Range : 10 – 86400

<autoReconnect> - Flag indicating if the connection manager should automatically reconnect to the service

0 : auto-reconnect disabled

1 : auto-reconnect lazy – reconnect on next send and every 3600 seconds

2 : auto-reconnect moderate (default) – reconnect 120 seconds, then every 3600 seconds after the first day.

3 : auto-reconnect aggressive – reconnect every 120 seconds

<overflowHandling> - Flag indicating if the way to handle overflows in data management

0 : FIFO (default)

1 : LIFO

<atrunInstanceId> - AT instance that will be used by the service to run the AT Command

Default : 2

Range : 1 – 3

<serviceTimeout> - It defines in seconds the maximum time interval for a service request to the server

Default : 5

Range : 1 – 120

<contextID> - PDP context identifier

1 – numeric parameter which specifies a particular PDP context definition

2. Wait for response OK

### 4.9.3. Connect to M2M service

1. Send command AT#DWCONN=1
2. Wait for response OK





<param\_3 when status is set to zero> - return parameters for the method. Key value pair should be used. param\_i should be the name of the element and param\_i+1 should be the value of the element

2. Wait for response #DWSEND: <msgId>
3. Wait for response OK
4. Wait for response #DWRING: <type>,<msgId>,<len>

where:

<type> - type of the data message to receive  
 <msgId> - index of the data message to receive  
 <len> - length of data message to receive

Example:

```
AT#DWSEND=0,property.publish,key,temp,value,22.5
#DWSEND=1
OK
#DWRING: 0,1,20
```

<Send raw data to M2M Service>

Content must be valid JSON

1. AT#DSENDNR=<dataLen>  
 where:  
 <dataLen> - number of bytes to be sent  
 Range : 1 - 1500
2. Wait for response ">"
3. Enter data/text to be sent
4. Wait for response #DSENDNR: <msgId>
5. Wait for response OK
6. Wait for response #DWRING: <type>,<msgId>,<len>

Example:

```
AT#SENDNR=135
>{"1":{"command":"mailbox.send","params":{"thingKey":"5c794151","command":"module.diag",
,"params":{},"singleton":false,"ackTimeout":30}}}
#DSENDNR=2
OK
#DWRING: 1,2,20
```





**NOTE :** The response to the AT#DWSEND command reports the <msgId> value that identifies the sending



**NOTE :** It's possible to use AT#DWSEND only if the connection has been opened with AT#DWCONN

#### 4.9.5. Receive data from M2M service

1. Send command AT#DWRCV(R)=<msgId>
2. Wait for response OK
3. Wait for response #DW(R)DATA: <msgId>,<error>,<len>,<data>

Where:

- <msgId> - index of the data message to receive
- <error> - error code of the message to receive, 0 if there is no error
- <len> - length of data message to receive
- <data> - M2M Service data



**NOTE:** It is possible to use AT#DWRCV only if the connection has been opened with AT#DWCONN, else the ME is raising an error.

If the data received are the consequence of previous data sending issued by AT#DWSEND, then they can be read only using AT#DWRCV command and not AT#DWRCVR command(i.e : AT#DWRCV and AT#DWRCVR are not interchangeable)

#### 4.9.6. Query connection status

1. Send command AT#DWSTATUS
2. Wait for response #DWSTATUS:  
<connected>,<lastErrorCode>,<latency>,<pktsIn>,<pktOut>,<bytesIn>,<bytesOut>

where:

- <connected> - 0 for disconnected, 1 for trying to connect, Wait for response OK, 2 for connect, 3 for waiting to connect
- <lastErrorCode> - last error code encountered by the client
- <latency> - milliseconds measured between last request and reply
- <pktIn> - number of packets received, tracked by the server
- <pktOut> - number of packets sent
- <bytesIn> - number of bytes received, TCP/IP payload
- <bytesOut> - number of bytes sent



3. Wait for response OK

Example:

AT#DWSTATUS

#DWSTATUS: 2,0,1000,108,143,8281,10661

#### 4.9.7. List information on message pending from M2M service

1. Send command AT#DWLRCV
2. Wait for response #DWLRCV:  
<msg\_nmuber>,[,<msgId\_1>,<msg\_1\_len>[,<msgId\_2>,<msgId\_2\_len>[,...<msgId\_n>,<msg\_n\_len>]]]

where:

<msg\_number> - number of message pending from M2M Service

Range: >=0

<msgId\_i> - index of the i-th data message to receive

<msg\_i\_len> - length of the i-th data message to receive

3. Wait for response OK

Example:

AT#DWLRCV

#DWLRCV: 5,1,22,2,22,3,22,4,22,5,22

OK



**NOTE:** It is possible to use AT#DWLRCV only if the connection has been opened with AT#DWCONN, else the ME is raising an error.





## 5. MEID Format and Function

Telit provides Mobile Equipment Identifier (MEID) as part of the Global Hexadecimal Assignment Guidelines and Procedures

The 56-bit identifier structure is compatible between 3GPP IMEI and 3GPP2 MEID. MEIDs Each MS is assigned a unique MEID. When used as SF\_EUIMID, it is uniquely assigned to an R-UIM.

The MEID identifies the manufacturer of the MS. When SF\_EUIMID is assigned to an R-UIM, it identifies R-UIM manufacturer.

MEID Structure and Format:

The MEID digit range is hexadecimal and syntactically consistent with the IMEI structure. However, the MEID structure does not utilize all of the fields in the exact semantic manner as in IMEI. The MEID numbering space is allocated in a manner that does not impact the decimally encoded IMEI. The MEID structure is also consistent with the MEID allocation scheme which uses 24-bit Serial Numbers.

**The MEID structure:**

Manufacturer Code								Serial Number						CD
R	R	X	X	X	X	X	X	Z	Z	Z	Z	Z	Z	C

In the case of MEIDs for terminals conforming exclusively to 3GPP2 technology, all of these fields are defined as hexadecimal values with the following valid range:

RR - valid range A0 ... FF – globally administered

XXXXXX - valid range 000000... FFFFFFFF

ZZZZZZ - valid range 000000.... FFFFFFFF

C - valid range 0... F – not transmitted over the air

The MEID does not specify the frequency band, air-interface technology or supported service associated with the MS

### 5.1. MEID Conversion, HEX to DEC

To convert the MEID from HEX to DEC manually you will need to break the MEID down into two parts. Break the MEID into ‘first 8 digits’ (part 1) and ‘last 6 digits’ (part 2). Convert the two parts to decimal and then bring the two parts together again to make a single number again. See below example using MEID A100000009D6000D.



A1000009 = 2701131785

D6000D = 14024717

A1000009D6000D = 270113178514024717

AT#MEID?

#MEID: A1000009D6000D

OK

AT+CGSN

270113178514024717

OK

To convert the MEID from decimal to Hex you will need to do the reserve procedure from the example.



## 6. Service and Firmware Update

The **Telit Modules** firmware is updated through the Serial Interface normally used for the AT Commands.

It is suggested to provide an RS232 interface on the User Printed Circuit Board (where the **Telit Module** is soldered) to perform the physical connection between the **Telit module** and a Windows-based PC. That simple circuitry makes the firmware updating easy when a new firmware version is released.

During the User Application development or evaluation phase of the **Telit module**, the RS232 interface or the USB port implemented on the **Telit Evaluation Kit (EVK2)** [6] can be used to connect the **Telit module** to a Windows-based PC on which a dedicated tool for firmware updating is running.

Telit provides the User with two tools to update the firmware of the module. The following paragraphs describe them.




---

### NOTE:

GT terminals are complete encased modems. They do not need the Telit Evaluation Kit (EVK2) to perform testing, evaluation and Firmware Update.

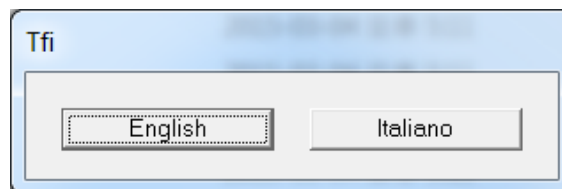
---

### 6.1. Step-by-Step Upgrade Procedure (TFI)

The firmware update can be done with a specific software tool provided by Telit that runs on Windows based PCs.

First the program will erase the content of flash memory, and then the program will write on the flash memory. To update the firmware of the module, we suggest the following procedure:

- Run the file **TFI\_xxxx.exe**. The following window must be displayed, select the language preferred by pressing the correspondent button.

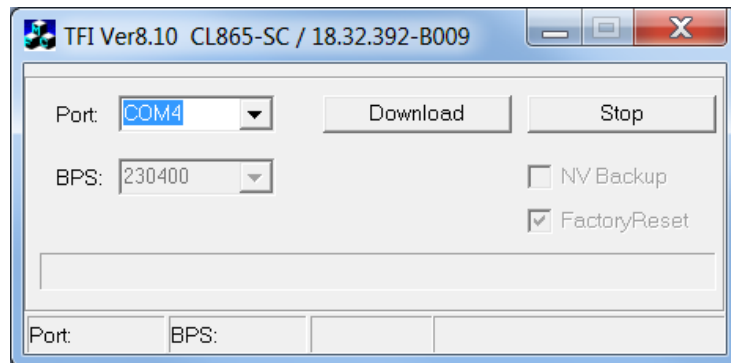


- The End User License Agreement will appear. Please, read it and accept the terms if you are going to proceed.

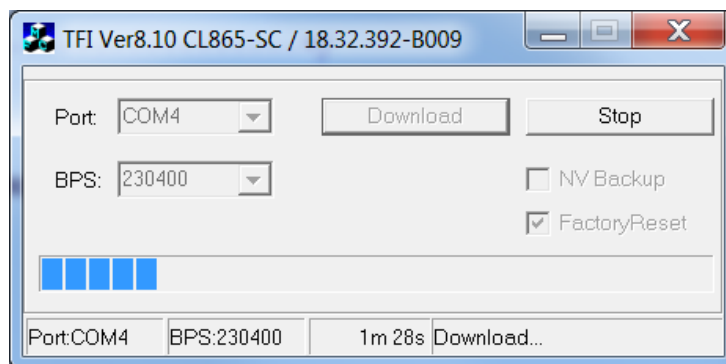
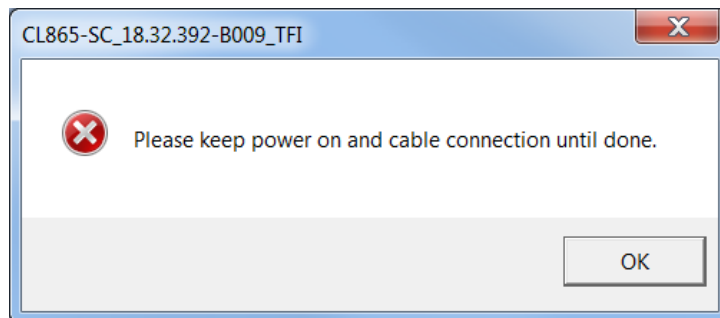




version and this version will be updated to the module.

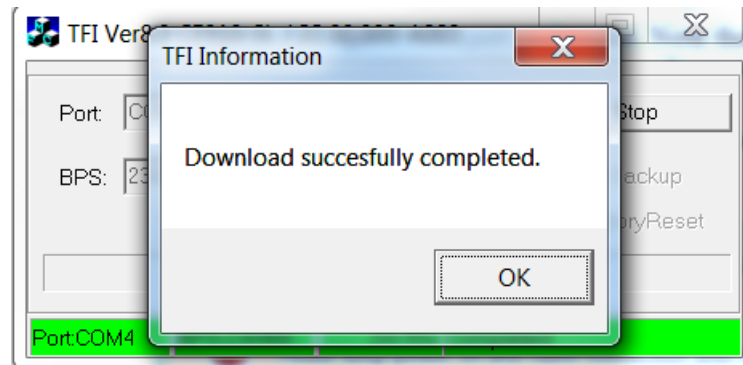


- Select the right COM port and speed. Note that to go faster than 115200 you need a special hardware on the PC. Then Press the Download button and within 5 seconds power-on the CL865-SC.
- Firmware Version displayed on Title bar is current firmware version.



Wait for the end of programming green message OK.





The Telit CL865-SC module is now programmed with the new firmware.

## 6.2. XFP Tool

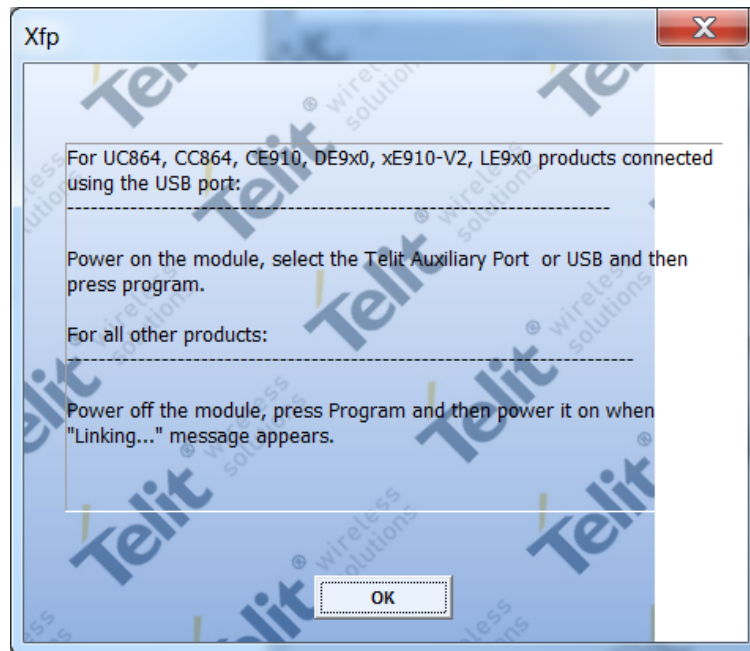
The firmware update of the module can be performed with the Xfp Tool provided by Telit. It runs on Windows based PCs. It erases the flash memory content, and then it downloads the new firmware on the flash memory.

### 6.2.1. Step-by-Step Upgrade Procedure

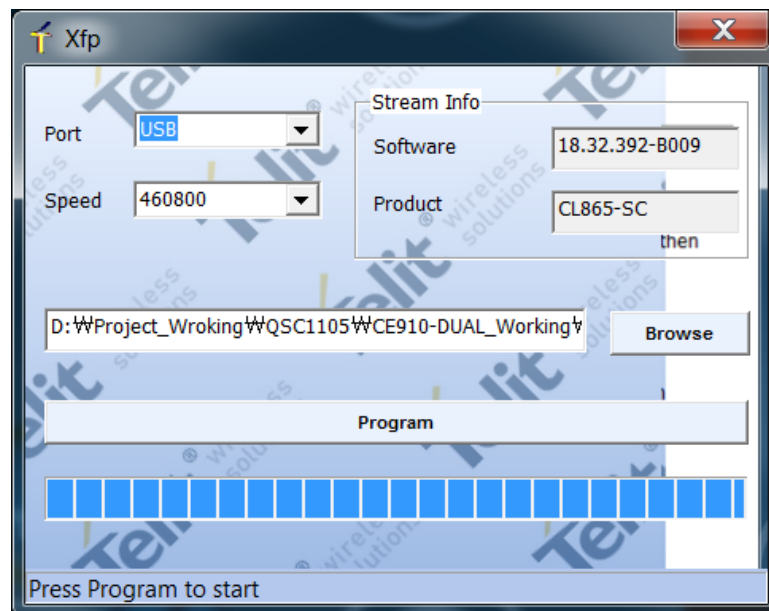
To update the **Telit Module** firmware, follow the procedure:

1. collect information about the **Telit Module** and Software version using the following AT commands:
  - **AT+CGMR<cr>**, returns the Software version information;
  - **AT+CGMM<cr>**, returns the **Telit Module** identification.
1. Switch OFF the Telit Module.
2. Run the *Xfp.exe* tool, the following windows are displayed. Power OFF the module if needed then press OK button.

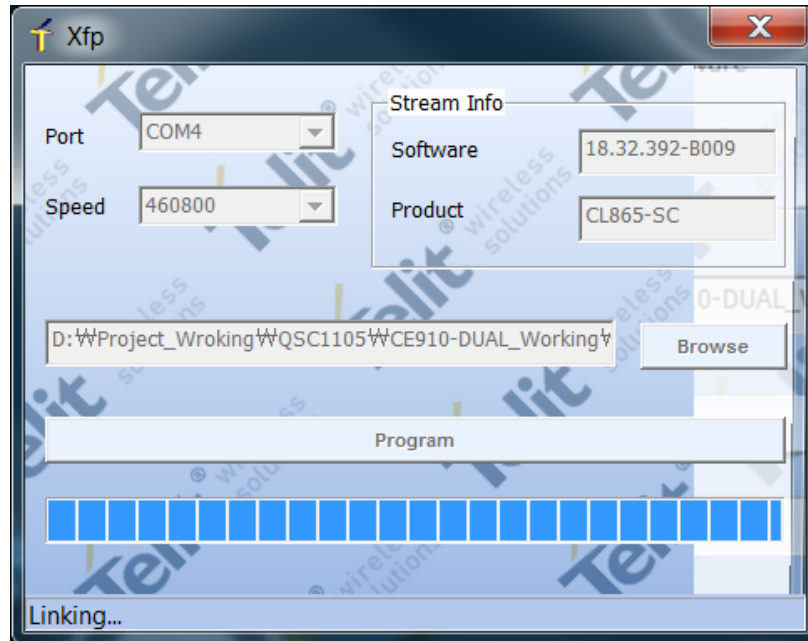




3. After pressing OK button on the screen is displayed only the following windows.



- After selecting COM port, speed<sup>1</sup> and stream file (stream files holds new firmware) press Program button, a flashing blue bar appears on the displayed window. The following window is displayed on the screen.

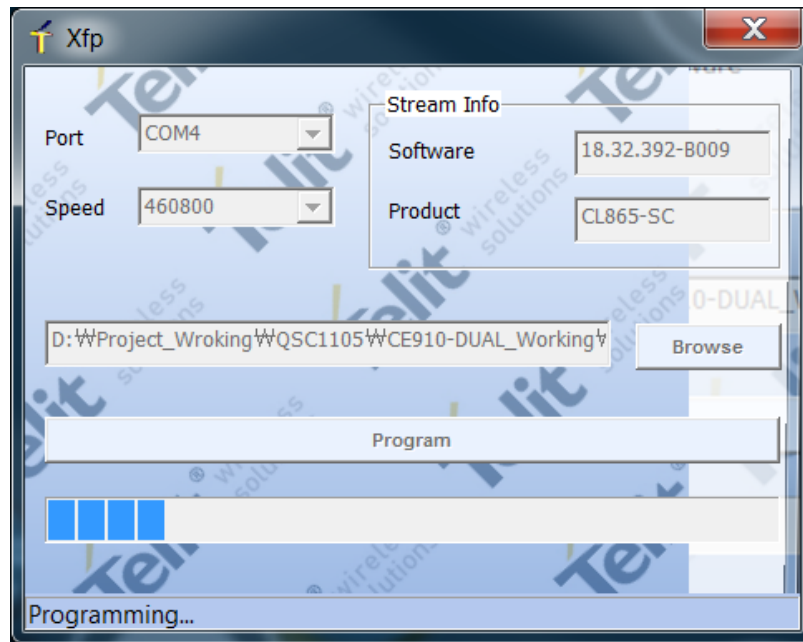


- After powering ON the module the following window is displayed on the screen. The programming is in progress

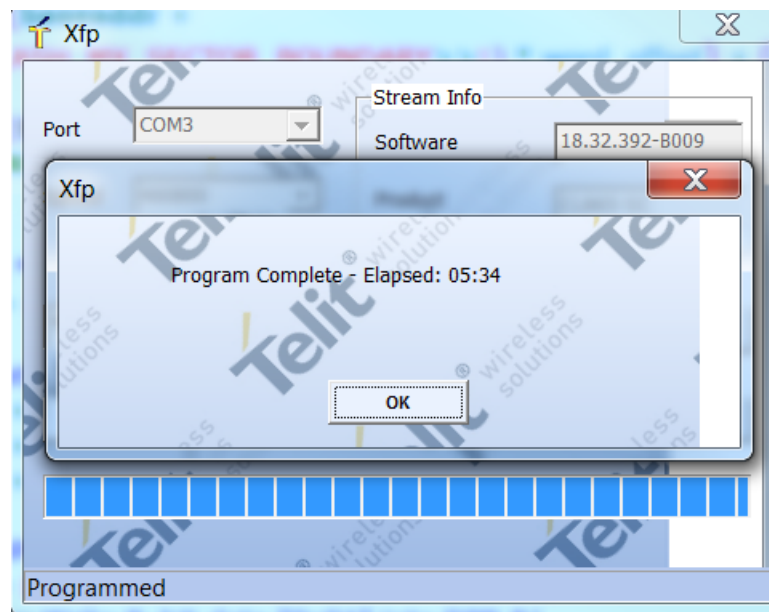
<sup>1</sup> To use speed greater than 115200 Bps a dedicated hardware on PC is needed.







- The following window is displayed on the screen when the module is successfully programmed.



- Now the **Telit module** is programmed with the new firmware. Press OK button and exit the tool.



## 7. Acronyms and Abbreviations

Term	Definition
ADC	Analog-to-Digital Converter
CDMA	Code Division Multiple Access
DAC	Digital-to-Analog Converter
DTE	Desk Top Equipment
EVRC	Enhanced Variable Rate CODEC
GPIO	General Purpose Input / Output
GPS	Global Positioning System
HF	Hands-free
I2C	Inter-Integrated Circuit
JDR	Jammer Detector
JTAG	Joint Test Action Group(ANSI/ICEEE Std. 1149.1-1990)
ME	Mobile Equipment
MS	Mobile Station
MT	Micro Telephone or HandSet (MT or HS)
PCM	Pulse Coded Modulation
PDM	Pulse Density Modulation (in a DAC)
RTC	Real Time Clock
R-UIM	Removable User Identity Module
S-GPS	Simultaneous-GPS
Sprint Vision	Trade name for 1xRTT service
Sprint Ready Link	Trade name for PTT
TA	Terminal Adapter
TE	Terminal Equipment
TGPIO	Telit General Purpose Input / Output
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
VAUX	Voltage Auxiliary
ZIF	Zero Intermediate Frequency



## 8. Document History

Revision	Date	Changes
0	2014-07-10	Initial release.
1	2015-08-25	Adding CL865-DUAL Sprint and CL865-SC models

