



Polte Device Integration Guide

Revision	Date	Changes	Author
1	March 5, 2019	Initial Release	Wondoloski
2	March 7, 2019	Editing Updates	Elliott
3	April 26, 2019	AT command updates, for new firmware version 1.0.5,1.0.4.	Wondoloski

Table of Contents

Introduction.....	3
Software Version	3
Description.....	3
Usage Models	5
Usage Model 1: No External Processor	5
Usage Model 2: Limited External Processor.....	6
Usage Model 3: Full External Processor	7
AT Commands.....	9
Documentation Style.....	9
Location Commands	9
AT+POLTEVER?.....	10
AT+POLTEREG	11
AT+AUTOLOC.....	13
AT+POLTELOC.....	14
AT+POLTECOMPRESS	16
References	20

Introduction

This datasheet provides information about Polte -enabled modems, and the requirements for implementation of the Polte location service on a device. It provides system integrators a description of the feature set provided by a modem equipped with the Polte Compression Engine.

Software Version

This document covers the CE firmware versions shown here.

Firmware version number: 1.0.5,1.0.4

Description

The Polte solution is a chip to cloud architecture that provides precise location determination for Polte-enabled devices. Figure 1 provides a high-level overview of how the Polte location solution works.

The IoT device consists of the chipset (SOC) of the Licensee, which is embedded into other components to make up the device. The other components may be provided by Licensee or other third parties.

The Polte Cloud provides the location determination information via APIs to other third parties or, optionally, back to the device's application.

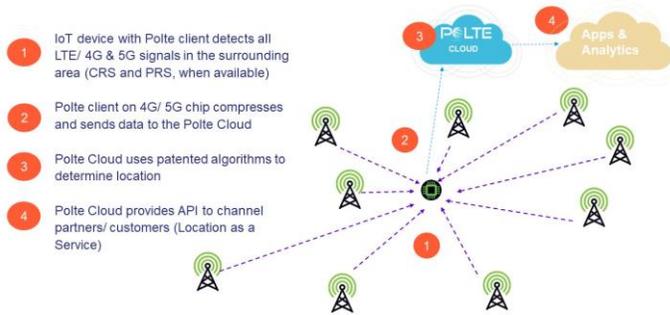


Figure 1: Polte Location Solution

The Polte location solution provides the best accuracy available for cellular systems by using advanced multipath mitigation techniques, super-resolution, interference and noise cancellation, and heuristic algorithms.

The device maintains a low power profile by not requiring additional hardware for location (Polte provides firmware on the communication module). In addition, the uplink location data size to the cloud is extremely small. For a CAT-M device, it is approximately 200 to 600 bytes of data.

Figure 2 shows a system block diagram of the solution. There are three primary usage models for the solution.

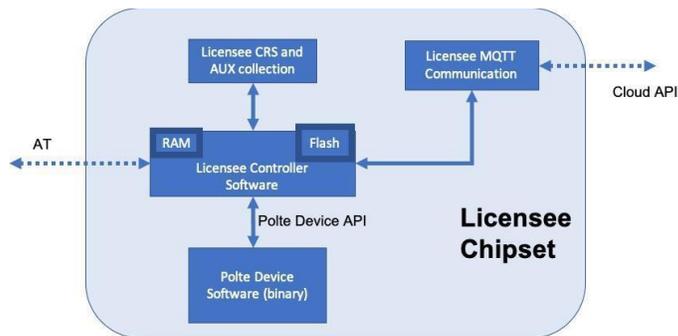


Figure 2: System Block Diagram

In the first usage model - **no external processor** - the modem is in full control of its power state, when to locate, and when and how to send data to the cloud. The modem does not have access to external sensors, so locations are either periodic, or are triggered via SMS message. The SOC reports location information to the Polte Cloud independently from any external processors and is able to be controlled from the Polte Cloud through configuration messages. This is a very basic usage scenario and is therefore the most power efficient.

In the second usage model - **hybrid use** - an external processor is in control of the power state of the modem and when to locate; however, the modem could retain its ability to control how to send data to the cloud. The external processor triggers locations via AT command, either periodically, or during an event triggered itself by one of the sensors. The external processor could be extremely limited, while providing the ability to trigger locations based on a sensor input.

In the third usage model - **full external processor** - an external processor is in control of the power state of the modem and when to locate, as well as when and how to send data to the cloud. The external processor triggers Polte CE captures via AT command, either periodically, or during an event triggered by one of the sensors. Location information is then packaged and sent to a 3rd party cloud, which subsequently triggers a 'locate' event to the Polte Cloud via a REST interface. Polte processes the location information data and responds with a location, where the 3rd party cloud makes use of that location.

Usage Models

Usage Model 1: No External Processor

In this usage model, the modem is in full control of its power state, when to locate, and when and how to send data to the cloud. The modem does not have access to external sensors, so locations are either periodic, or are triggered via SMS message. All communication is to the Polte Cloud.

This is a very basic usage scenario but would be low power and size efficient.

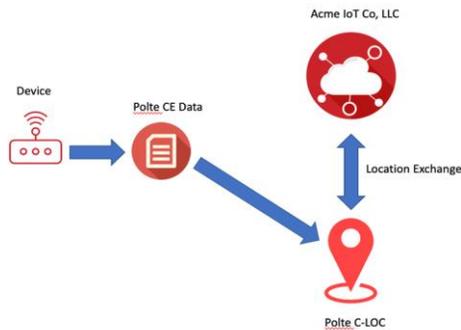


Figure 1: No External Processor

Valid AT Commands for this Model

The details on the AT commands are given in the AT Commands section of this document.

There is only one valid AT+ command for this model of operation and that is the command to activate the locations or to turn off the locations. This is completed using the AT+AUTOLOC command. The modem processor is in full control of the Polte location service once it is activated. The configuration and control of this service is completed through the Polte Cloud interface. Please see Polte Quick Start Guide for details on the configuration of the device settings through the cloud.

An example of activating the device locations to Polte Cloud directly:
`AT+AUTOLOC=1`

Which returns the following:
`+AUTOLOC: 0`
`OK`

An example of deactivating the device locations to Polte Cloud directly:
`AT+AUTOLOC=0`

Which returns the following:
`+AUTOLOC: 0`
`OK`

Usage Model 2: Limited External Processor

In this usage model, an external processor is in control of the power state of the modem and when to locate; however, the modem could retain its ability to control how to send data to the cloud.

The external processor triggers locations via AT command, either periodically, or during an event triggered itself by one of the sensors.

In this case, the external processor could be extremely limited, while providing the ability to trigger location based on sensor input. A simple example of this mode would be a button triggered location device.

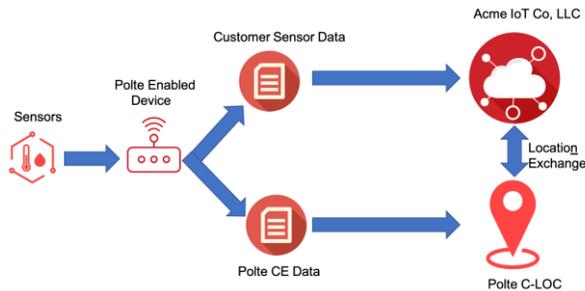


Figure 2: Limited External Processor

Valid AT Commands for this Mode

The details on the AT commands are given in the AT Commands section of this document.

The main function of the external processor in this model is triggering a location request. This is done using the AT+POLTELOC command to trigger the request. The modem then returns the location data and status information. Please see the following section for the details of this command. If the modem has not previously registered to the Polte Cloud, this command will register it directly to the Polte Cloud. In the event the modem is unable to register to the cloud, an error code will be returned.

The modem will communicate the necessary location information to the Polte cloud. If a location response is requested in the AT+POLTELOC command, the modem will subsequently provide this location to the external processor.

An example of a location request with high accuracy mode:
`at+polteloc=1`

Which returns the following:
`+POLTELOC: 0`

An example of a location request with low power mode also returning lat/long and timestamp:
`at+polteloc=0,1`

Which returns the following:
`+POLTELOC: 0,32.97582516403683,-96.71043140500139,1548815518`

Commented [RM1]: Need to introduce different modes before using.. Maybe a sentence explaining there are two modes..

Formatted: Space After: 0 pt, Line spacing: single

Formatted: Indent: Before: 0"

Usage Model 3: Full External Processor

In this usage model, an external processor is in control of the power state of the modem and when to locate, as well as when and how to send data to the cloud.

The external processor triggers locations via AT command, either periodically, or during an event triggered by one of the sensors.

This model is anticipated for those systems that prefer to treat Polte as just another sensor. In this scenario, the modem reports the compression information to the external micro-processor VIA AT command.

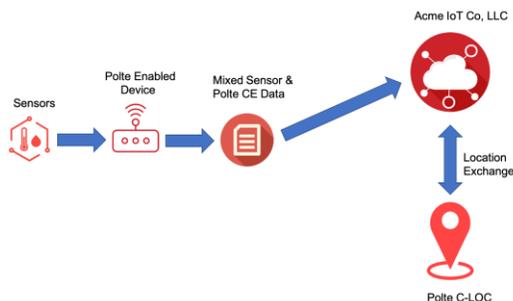


Figure 3: Full External Processor

Valid AT Commands for this Mode

The details on the AT commands are given in the AT Commands section of this document.

In this mode, access to all the AT+ commands defined in this document are valid for use. The use of this mode provides the most flexible usage of the system and allows the location information to be bundled with other sensor information on the device and sent to a 3rd party cloud. This 3rd party cloud provider can then integrate directly to the Polte Cloud through a REST interface to receive the location information.

The basic requirements for communicating location information with the device in this mode is to register the device and generate location commands. These are done from the device to the 3rd party cloud with the 3rd party cloud proxying this information to the Polte Cloud where necessary.

Registration

AT+POLTEREG?

The flexibility requires that more control and information for the location service be handled directly by the device manufacturer's processor.

In the full external model, you may register the device through your 3rd party cloud as a proxy to the Polte Cloud using the REST API. The AT+POLTEREG? command can be used to return the necessary token information, which needs to be relayed to the Polte Cloud through the get_ue_register API.

An example of Registering the device to PoLTE cloud through 3rd party
AT+POLTEREG?

Which returns the following:

+POLTEREG: 0,123456789123456,12-345678-912345-8,123456abcde,f123456789ab

OK

Please note that the command to register information to the 3rd party cloud returns the manufacturer and model token in addition to the IMEI and ue-token. This information is required to be sent to the Polte Cloud through the REST API from the 3rd party cloud application.

For detailed information on the use of the cloud REST APIs, please see the Polte Quick Start Guide.

[Get Location Data from the device](#)

AT+POLTECOMPRESS

To receive the current device location information, the external processor issues the AT+POLTECOMPRESS command. This will return compressed location data as a bitstream. This information needs to be relayed to the Polte Cloud through the API command /ue/{ueToken}/locate, which will return the location information in a latitude/ longitude format back to the 3rd party cloud.

An example of Output compression data, high accuracy mode, hexadecimal format [default, not specified in command]:

AT+POLTECOMPRESS=1

Which returns the following:

+POLTECOMPRESS:

```
0,2549582289,607,020000025801000001000306326A3E0310041013F6003F003F006007F3611113F
6FFFFE204FFFFFB6300000E87030A5CB7391107F3611101003F1AF1FF6F46AC48A93FA134A2469
F4FA23EA358A047A16F9FF1A0569D509EEFA0F0A26DA1DCA3E7A23DA149A1429C479F3C9E489
D0001771AD00000BAD6B648C93569364D383B42345337CB43C059B3CBB5B8BDB3D7B343C034C
E374433373A315D36C43EBB51B4D1B4BABC0000F018DE0000F5BAC6B8BBC0C5CDB6EABD53C
23DCA38E039DA344A353F43424F37F93BCB3BBA4FBB66BDDBB5C5B9B8BCBADABC5EBA48000
04013EC0000D0CC6DD9CEC8D3CED2E3B8E6CE55C44EC255C668CC48D65EE046CE4CE33FD5
4E794EE0415858544F5749424E4A4B4D7701015612E100006BD0D76F546FDDDBEB4D5FD1CC76
4761F6E9EC494FDFD2655658E2D96F5056E1F7E2674E54CFD16568EAD6E3DF576ED901002C11
520000E24DD443D755F75BDD52D043654F6B69F050E44278555F574760445D5BD35B5E425F4478
4BDF6CC4EAC266C56FD35DE60001651140000067FDE3DC5E5CFFDE5F63E1CF6A4EFDDFD659
58E6D9695BE9DDEA6B5AECDE786169E16E656BD9EB596CFECA666CEFD86D0101A910530000E
6C95A4F46F3CADCEB4359D8C4D5605E497EF4EDCBF75AD4FDD8CF5E4F6C5CDBCD5E5F6865
E3D2D95B4C536CCAC8D1650100360E130000F7F0656AE26FF979FBE264F9F3F7E259E66F6B7D
5D6AE56BE6FEDE5F5EEC5D62EB66DFE7FEDF7E6C656AFE7AF8FDFBE800006E0D9800005D41
67E86DE17B525AC65D6652C2D1585AC5C65D5244DAF1475ACDCF5669EAE2D44FE8CFC25051C
CD3766FE1CBDC555E00AA
```

OK

For detailed information on the use of the cloud REST APIs, please see the Polte Quick Start Guide.

Please note that there are current limitations in the way in which this command functions that require the device manufacturer to avoid other function calls to the chipset while this command is operating, as there is the potential for a race condition. If this race condition occurs, the AT+ POLTECOMPRESS function will likely return an error.

AT Commands

This is a summary of the available AT+ commands relating to the device manufacturer.

Documentation Style

In the AT command definitions, the following syntax is used:

Syntax	Description
<value>	The value will always be returned in the response
[<value>]	The value might be returned in the response, depending on the parameters of the command, and if the command was successful

Formatted Table

Location Commands

Command	Response
AT+POLTEVER?	+POLTEVER: <v_app>, <v_ce>
AT+POLTEREG?,<mode>=1	+POLTEREG: <stat>, [, <imei>, <ue-token>]
AT+AUTOLOC=<mode>[,<band>]	+AUTOLOC : <stat>
AT+POLTELOC=[<mode>,<get_loc>]	+POLTELOC : <stat> [, <lat>, <lon>, <utc>]
AT+POLTECOMPRESS=[<mode>,<format>]	+POLTECOMPRESS : <stat>, <crc>, <len>, <data>

Formatted Table

Formatted: Font: 10 pt, Complex Script Font: 10 pt

Formatted: Font: 10 pt, Complex Script Font: 10 pt

Formatted: Font: 10 pt, Complex Script Font: 10 pt

Formatted: Font: 10 pt, Complex Script Font: 10 pt

Formatted: Font: 10 pt, Complex Script Font: 10 pt

Formatted: Font: 10 pt, Complex Script Font: 10 pt

AT+POLTEVER?

Operational Conditions

This command is available at all times.

Description

This command is used by the external processor to request the version of the application and CE.

Syntax

Command	Description
AT+POLTEVER?	+POLTEVER: <v_app>, <v_ce>

Parameter Definitions

v_app

Value	Description
X.Y.Z	ASCII text representing the version of the application

v_ce

Value	Description
X.Y.Z	ASCII text representing the version of the CE

Example

```
AT+POLTEVER?  
+POLTEVER: 1.0.4,1.0.3  
OK
```

AT+POLTEREG

Operational Conditions

This command is available at all times.

Description

This command is used by the external processor to register the modem with the Polte Cloud.

Syntax

Command	Description
AT+POLTEREG=1	+POLTEREG: <stat>, [, <imei>, <ue-token>]
AT+POLTEREG?	+POLTEREG: <stat>, [, <imei>, <ue-token>][,<mfg-token>,<model-token>]

Formatted Table

Parameter Definitions

reg

Value	Description
1	Register directly to Polte

Formatted Table

stat

Value	Description
0	Success
1	Device registration rejected
2	Network unavailable

: imei

Value	Description
X	The IMEI (International Mobile station Equipment Identity) of the modem

ue-token

Value	Description
X	A Polte specific token for this modem

mfg-token

Value	Description
X	A Polte specific token for this manufacturer

model-token

Value	Description
X	A Polte specific token for this model

Example

Register device to Polte Cloud directly

```
AT+POLTEREG=1
+POLTEREG: 0,123456789123456,12-345678-912345-8
OK
```

Register device to Polte Cloud through 3rd party

```
AT+POLTEREG?
+POLTEREG: 0,123456789123456,12-345678-912345-8,123456abcde,f123456789ab
OK
```

Formatted Table

Formatted Table

Formatted: Indent: Before: 0"

Formatted: Indent: Before: 0.5"

Formatted: Indent: Before: 0"

Formatted: Indent: Before: 0.5"

AT+AUTOLOC

Operational Conditions

This command is available in all modem states. The modem must be able to attach to the network to perform AUTOLOC. PoLTE recommends setting connection parameters (APN) before calling this command.

Description

This command is used by the integrator to inform the modem that it should perform periodic location without the need for additional AT command interaction. The values are stored non-volatilely and will persist through a device reset or power cycle. Parameters used by the location process are configured via an OTA API or by the AT+POLTEUPLLOADCONFIG command.

Syntax

Command	Description
AT+AUTOLOC=<mode>[,<band>]	+AUTOLOC : <stat>

Formatted Table

Parameter Definitions

mode

Value	Description
0	Disable
1	Enable

band

Value	Description
band	Admin Band Value. Recommended to use the default value of 0 (auto). Valid Values are operator and location, specific Supported bands are 1-32, 65-76, 85, 252, and 255

stat

Value	Description
0	Success
-1	Value or Format Error

Example

Disable

```
AT+AUTOLOC=0
+AUTOLOC: 0
OK
```

Enable

```
AT+AUTOLOC=1
+AUTOLOC: 0
OK
```

AT+POLTELOC

Operational Conditions

This command is only available in RRC_IDLE.

The modem must have previously attached to the network for this command to work.

Description

This command is used by the external processor to trigger a location event. The modem will compress the data, send it to the Polte Cloud, and optionally, request the resultant location information.

If the modem has not previously registered to the Polte Cloud, this command will register it. In the event the modem is unable to register to the cloud, an error code will be returned.

Syntax

Command	Description
AT+POLTELOC=[<mode>,<get_loc>]	+POLTELOC : <stat> [, <lat>, <lon>, <utc>]

Formatted Table

Parameter Definitions

mode

Value	Description
0	Default: Low Power Mode
1	High Accuracy Mode

Formatted Table

get_loc

Value	Description
0	Do not retrieve lat/lon from Polte Cloud
1	Retrieve lat/lon from Polte Cloud

Formatted Table

stat

Value	Description
0	Success
-1	Error
-2	Bad RF Data
-9	Memory Allocation Error
-10	Capture Failure
-11	Compression Fail
-13	Network Connection Unavailable
-15	Aux Capture Fail
-17	No Location Published
-18	Serving Cell Bandwidth Not Supported

Formatted Table

lat

Value	Description
X	UE latitude

lon

Value	Description
X	UE longitude

UTC

Value	Description
X	UTC timestamp of location

Example

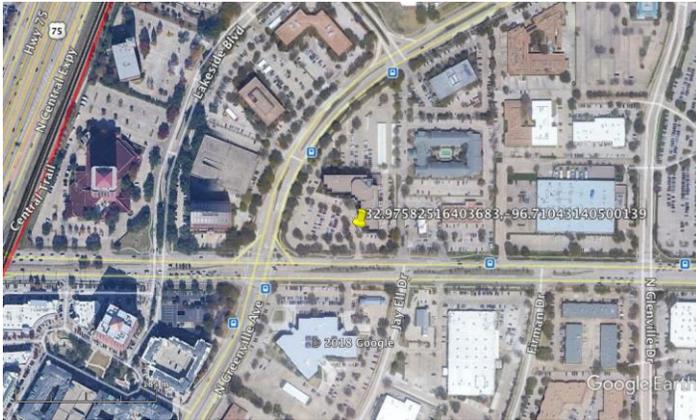
Example of a high accuracy mode locate

at+polteloc=1
+POLTELOC: 0

Example of a, low power mode, return lat/lon

at+polteloc=0,1
+POLTELOC: 0,32.97582516403683,-96.71043140500139,1548815518

Commented [RW2]: These examples are no longer relevant with the new AT parameters....



AT+POLTECOMPRESS

Operational Conditions

This command is available in RRC_IDLE.

Description

This command is used by the external processor to request compressed CRS information from the modem. It is only available in RRC_IDLE mode, since RRC_IDLE is guaranteed to allow the modem to capture any contiguous CRS symbols at any narrowband index.

Syntax

Command	Description
AT+POLTECOMPRESS=[<mode>,<format>]	+POLTECOMPRESS : <stat>, <crc>, <len>, <data>

Formatted Table

Parameter Definitions

mode

Value	Description
0	Default: Low Power Mode
1	High Accuracy Mode

Formatted Table

format

Value	Description
0	ASCII Hexadecimal format. <i>Default output mode if format parameter is omitted.</i>
1	Binary data (0x00 – 0xFF in data stream)

crc

Value	Description
X	CRC32, computed based on 0x04C11DB7 polynomial

stat

Value	Description
0	Success
-1	Error
-2	Bad RF Data
-7	Bad BSA Data
-9	Memory Allocation Error
-10	Capture Failure
-11	Compression Fail
-13	Network Connection Unavailable
-15	Aux Capture Fail
-17	No Location Published
-18	Serving Cell Bandwidth Not Supported

len

Value	Description
X	The length of the binary stream following, in bytes.

← Formatted Table

Data

The compressed data as a bitstream. The expected size of the bitstream is 300-600 bytes.

The compressed data is returned as an ASCII character array when *<format>* is set to 0. The text contains the hexadecimal representation of the binary data.

In either output format, the data will immediately follow the “,” after the *<size>* parameter. ASCII format will be 2 x *<size>* and binary format will be *<size>* bytes.

Example

Output compression data, high accuracy mode, hexadecimal format [default, not specified in command]

```
AT+POLTECOMPRESS=1
+POLTECOMPRESS:
0,2549582289,607,020000025801000001000306326A3E0310041013F6003F003F006007F3611113F
6FFFFE204FFFFFFB6300000E87030A5CB7391107F3611101003F1AF1FF6F46AC48A93FA134A2469
F4FA23EA358A047A16F9FF1A0569D509EEFA0F0A26DA1DCA3E7A23DA149A1429C479F3C9E489
D0001771AD00000BAD6B648C93569364D383B42345337CB43C059B3CBB5B8BDB3D7B343C034C
E374433373A315D36C43EBB51B4D1B4BABC0000F018DE0000F5BAC6B8BBC0C5CDB6EABD53C
23DCA38E039DA344A353F43424F37F93BCB3BBA4FBB66BDDBB5C5B9B8BCBADABC5EBA48000
04013EC0000D0CC6DD9CEC8D3CED2E3B8E6CE55C44EC255C668CC48D65EE046CE4CE33FD5
4E794EE0415858544F5749424E4A4B4D7701015612E100006BD0D76F546FDDDBEB4D5FD1CC76
4761F6E9EC494FDFD2655658E2D96F5056E1F7E2674E54CFD16568EAD6E3DF576ED901002C11
520000E24DD443D755F75BDD52D043654F6B69F050E44278555F574760445D5BD35B5E425F4478
4BDF6CC4EAC266C56FD35DE60001651140000067FDE3DC5E5CFFDE5F63E1CF6A4EFDDFD659
58E6D9695BE9DDEA6B5AECDE786169E16E656BD9EB596CFECA666CFD86D0101A910530000E
6C95A4F46F3CADCEB4359D8C4D5605E497EF4EDCBF75AD4FDD8CF5E4F6C5CDBCD5E5F6865
E3D2D95B4C536CCAC8D1650100360E130000F7F0656AE26FF979FBE264F9F3F7E259E66F6B7D
5D6AE56BE6FEDE5F5EEC5D62EB66DFE7FEDF7E6C656AFE7AF8DFBE800006E0D9800005D41
67E86DE17B525AC65D6652CD1585AC5C65D5244DAF1475ACDCF5669EAE2D44FE8CFC25051C
CD3766FE1CBDC555E00AA
OK
```

References

- [1] 3GPP TS 36.331 v13.7 Radio Resource Control (RRC); Protocol specification
- [2] 3GPP TS 36.304 v13.7 User Equipment (UE) procedures in idle mode
- [3] 3GPP TS 36.133 v13.9 Requirements for support of radio resource management
- [4] 3GPP TS 36.211 v13.7 Physical channels and modulation
- [5] http://www.sharetechnote.com/html/Handbook_LTE_BL_CE_Narrowband.html