



Polte Device Integration Guide

Revision	Date	Changes	Author
A	March 5, 2019	Initial Release	Wondoloski
B	March 7, 2019	Editing Updates	Elliott

Table of Contents

Introduction	4
Description	4
Usage Models	6
Usage Model 1: No External Processor	6
Valid AT Commands for this mode	6
Usage Model 2: Limited External Processor	7
Valid AT Commands for this mode.	7
Usage Model 3: Full External Processor	8
Valid AT Commands for this mode.	8
Registration	8
BSA Management	8
Get Location Data from device	9
AT Commands	10
AT+POLTELOC	11
AT+POLTEVER?	14
AT+POLTEREG	14
AT+POLTEGETBSA	17
AT+POLTEUPLOADBSA	18
AT+POLTECOMPRESS	21
	22
References	24

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.

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.

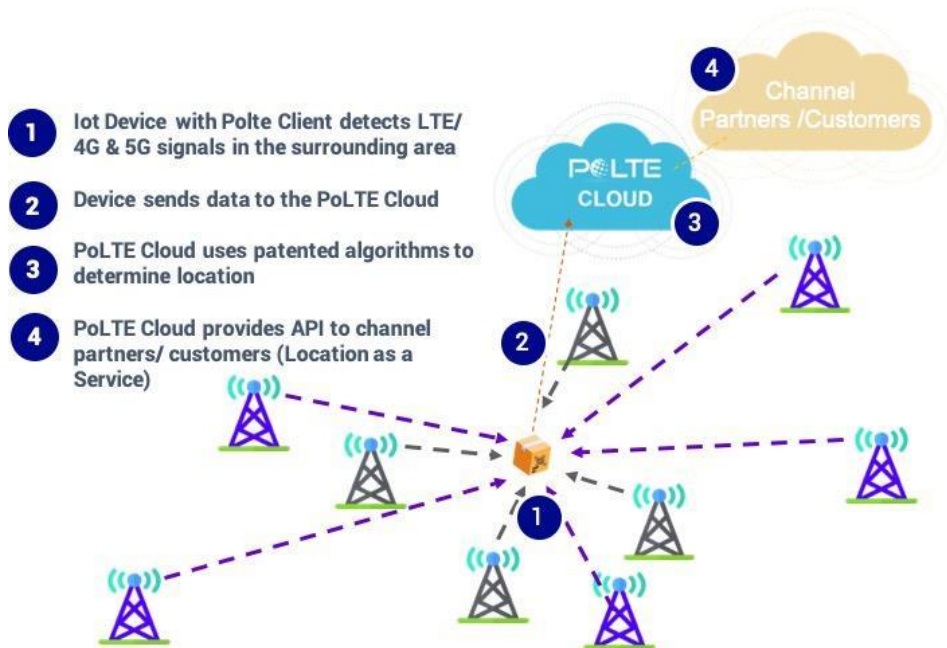


Figure 1: Polte Location Solution

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 any additional hardware for location (Polte is 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 300 bytes of data.

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

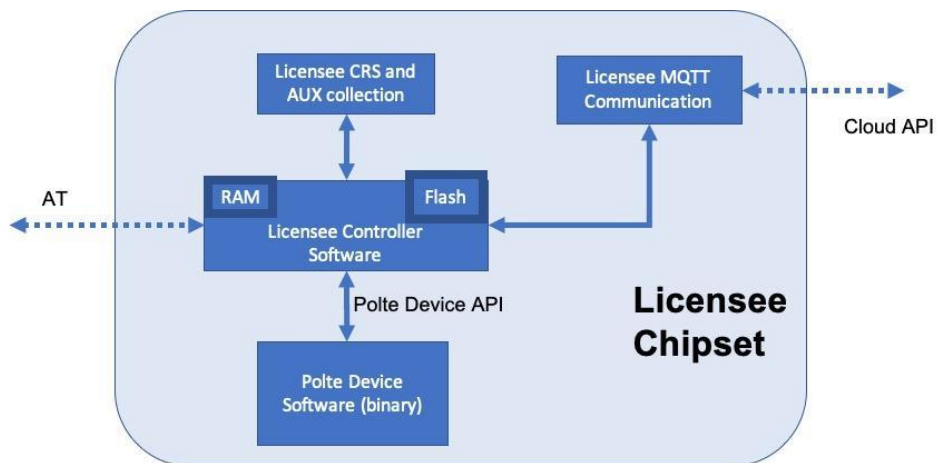


Figure 2: System Block Diagram

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. Sensor information is then packaged and sent to a 3rd party cloud, which subsequently triggers a 'locate' event to Polte C-LoC, via REST. Polte C-LoC processes the Polte 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.

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

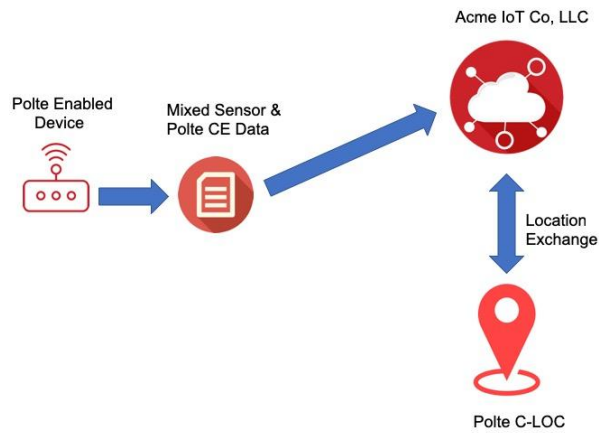


Figure 1: No External Processor

Valid AT Commands for this Mode

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

There are no valid external processor AT+ commands for this mode, since the modem processor is in full control of the Polte location service. The configuration and control of this service is completed through the Polte Cloud interface.

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.

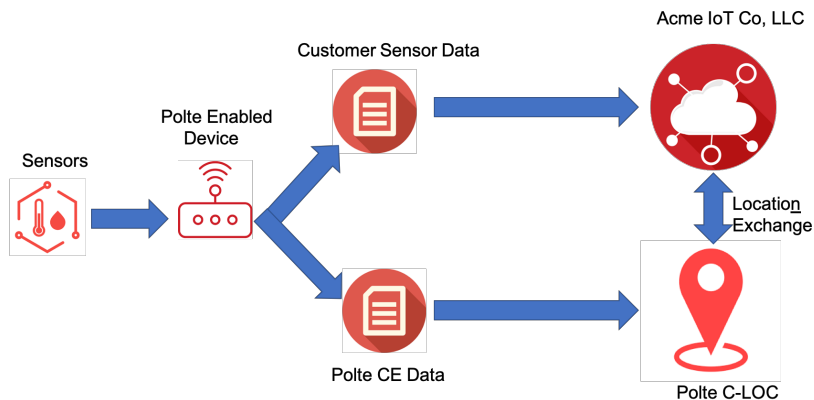


Figure 2: Limited External Processor

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+POTLELOC 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.

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.

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 itself 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.

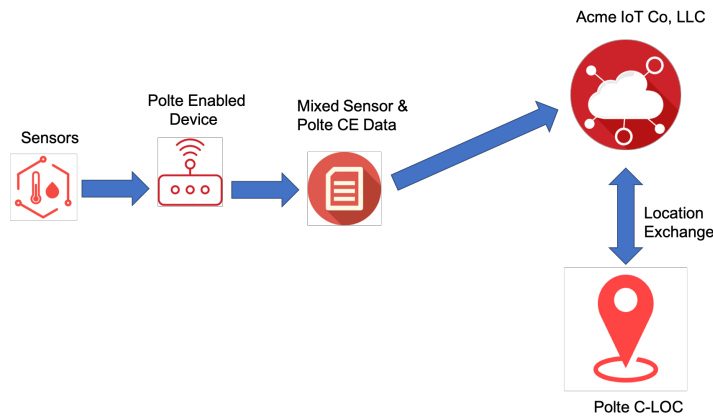


Figure 3: Full External Processor

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.

Registration

AT+POLTEREG

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

The sequence of events requires that the device initially be registered on the Polte Cloud by utilization of the AT+POLTEREG. This command directly communicates with the Polte Cloud to complete the registration process. This registration event is only required one time for the life of the device.

If you are unable or do not want the device to register directly to the Polte Cloud, you may register through your 3rd party cloud as a proxy to the Polte Cloud using the REST API. The AT+POLTEREG=1 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.

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

BSA Management

AT+POLTEGETBSA

AT+POLTEUPLOADBSA

Once the device is registered, the initial location request requires a base station almanac (BSA) to be loaded into the modem. In the current software, BSA management is an ongoing process required for getting valid location information from the device. This requirement will be removed in future software versions.

When new BSA updates are required (as the modem moves locations through the network), it is indicated by a message in the response to the location data request message AT+POLTECOMPRESS in the next section.

The device can manage the BSA request directly through the modem in a communication channel to the Polte Cloud utilizing the AT+POLTEGETBSA command. Issuing this command will cause the modem to directly communicate to the Polte Cloud and receive updated BSA information.

The device can also avoid direct communication with the Polte Cloud and rely on the 3rd party vendor cloud to proxy the information. If the BSA is managed in this method, and a BSA update is received from the 3rd party cloud, then the information is loaded into the modem utilizing the AT+POLTEUPLOADBSA command.

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

Get Location Data from 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.

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

The AT+POLTECOMPRESS command may return a value indicating that the BSA needs to be updated before the location information can be returned. If this is the case, then the BSA management process must occur before issuing the AT+POLTECOMPRESS command.

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

Location Commands

Command	Response
AT+POLTELOC=<cmd> [, <mode>]	+POLTELOC : <stat> [, <lat>, <lon>] [, <cmp_d>]
AT+POLTEVER?	+POLTEVER: <v_app>, <v_ce>
AT+POLTEREG[=<reg-option>]	+POLTEREG: <stat>, [, <imei>, <ue-token>]
AT+POLTEGETBSA=<output>[,<format>]	+POLTEGETBSA: <stat>[,<crc>,<size>,<data>]
AT+POLTEUPLOADBSA=<iter>,<data>,<crc>[,<full_crc>]	+POLTEUPLOADBSA: <stat>
AT+POLTECOMPRESS=<nbidx>,<nretry>[,<format>]	+POLTECOMPRESS : <stat> [,<crc>,<len>,<data>]

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. Removal of this condition is underway.

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.

Since CRS data is available only in RRC_IDLE, this command works only in RRC_IDLE.

If this command is called while the modem is not in RRC_IDLE, and the number of retries <nretry> is set to 0, the modem return an error code corresponding to this error.

If this command is called while the modem is not in RRC_IDLE, and the number of retries <nretry> is greater than 0, the modem attempt <nretry> captures of CRS data at 1 second intervals.

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.

If the modem does not have the appropriate Base Station Almanac (BSA) for the area, the modem will download it (after capturing CRS data, but before compressing the CRS data). In the event the modem is unable to download the BSA, an error code will be returned.

Syntax

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

Parameter Definitions

Table 19: mode

Value	Description
0	Perform location, but do not request location sent back to device
1	Perform location, and request location to be sent back to the device

Table 20: nframes

Value	Description
1..6	Number of frames to use for location

Note:

(1) The number of frames used in the location attempt influences the quality of the location generated. The higher the # of frames used, the better the location will be.

(2) The number of frames used in the location attempt influences the processing time of the compression engine.

Table 21: nbidx

Value	Description
0..15, 31	Narrowband index, 31 for center of band

Note:

(1) Changing the narrowband index for each location is recommended for best location performance

Table 22: nretry

Value	Description
0..999	Number of times to retry

Table 23: stat

Value	Description
0	Success
1	Modem in RRC_CONNECTED
2	Bad RF Data
3	Run time error
4	Unable to register to the Polte Cloud
5	Unable to query BSA

Table 24: lat

Value	Description
X	UE latitude

Table 25: lon

Value	Description
X	UE longitude

Table 26: utc

Value	Description
X	UTC timestamp of location

Example

at+Polteloc=1,6,0,10

+POLTELOC: 0,32.97582516403683,-96.71043140500139,1548815518

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

Table 27: v_app

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

Table 28: v_ce

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

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[=<reg-option>]	+POLTEREG: <stat>, [, <imei>, <ue-token>][, <mfg-token>, <model-token>]

Parameter Definitions

Table 29: reg

Value	Description
0	Register directly to Polte. <i>Default Mode.</i>
1	Does not directly register to Polte. Provides all tokens required to perform 3 rd part registration.

Table 30: stat

Value	Description
0	Success
1	Device registration rejected
2	Network unavailable

Table 31: imei

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

Table 32: ue-token

Value	Description
X	A Polte specific token for this modem

AT+POLTEGETBSA

Operational Conditions

This command is available at all times.

Description

This command is used by the external processor to request the modem performs a BSA update. The BSA is stored internally in the modem.

Syntax

Command	Description
AT+POLTEGETBSA=<output>[,<format>]	+POLTEGETBSA: <stat>,<crc>, <size>,<data>

Parameter Definitions

Table 33: output

Value	Description
0	Do not output the BSA
1	Output BSA on completion

Table 34: 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)

Table 35: stat

Value	Description
0	Success
1	BSA not available

Table 36: *crc*

Value	Description
X	CRC32, computed based on 0x04C11DB7 polynomial

Table 37: *size*

Value	Description
x	Size of the BSA in bytes. <i>In the response message there will be twice this many characters when ASCII format is selected.</i>

Table 38: *data*

Value	Description
X	Data will be 0x30-0x39, 0x41-46 when <i><format>=0</i> and 0x00-0xFF when <i><format>=1</i>

The BSA 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.

AT+POLTEUPLOADBSA

Operational Conditions

This command is available at all times.

Description

This command is used by the external processor to update the BSA data. The BSA is stored internally in the modem.

Syntax

Command	Description
AT+POLTEUPLOADBSA=<iter>,<data>,<crc>[,<full_crc>]	+POLTEUPLOADBSA: <stat>

Parameter Definitions

Data

The external micro-processor should use this command to upload the Polte BSA. The Polte BSA may be queried via the REST API or via MQTT. The response from the Polte cloud may then be uploaded to the device via UART using this AT command. The command breaks the BSA into several packets, encodes them in base 16 ASCII text, and uploads them to the modem.

Table 39: iter

Value	Description
0..30	The BSA Upload iteration count

Table 40: data

Value	Description
0x30-0x39, 0x41-46	BSA Payload, Base 16 encoded ASCII text, maximum of 256 bytes encoded in each data payload

Table 41: crc

Value	Description
X	CRC32, computed based on 0x04C11DB7 polynomial over the packet

Table 42: full_crc

Value	Description
X	CRC32, computed based on 0x04C11DB7 polynomial computed over the entire BSA inclusion in message indicates final packet

Table 43: stat

Value	Description
0	Success
1	CRC Fail
X	System level failures (non-zero values)

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=<nbidx>,<nretry>[,<format>]	+POLTECOMPRESS : <stat>, <crc>, <len>, <data>

Parameter Definitions

Table 44: nbidx

Value	Description
0..15	Narrow band index according to 3GPP TS36.211-5.2.4
31	Center of the band

Table 45: nretry

Value	Description
0..10	Number of times to retry

Table 46: 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)

Table 47: crc

Value	Description
X	CRC32, computed based on 0x04C11DB7 polynomial

Table 48: stat

Value	Description
0	Success
1	Modem in RRC_CONNECTED
2	Bad RF Data
3	Run time error

Table 49: len

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

Data

The compressed data as a bitstream.

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.

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