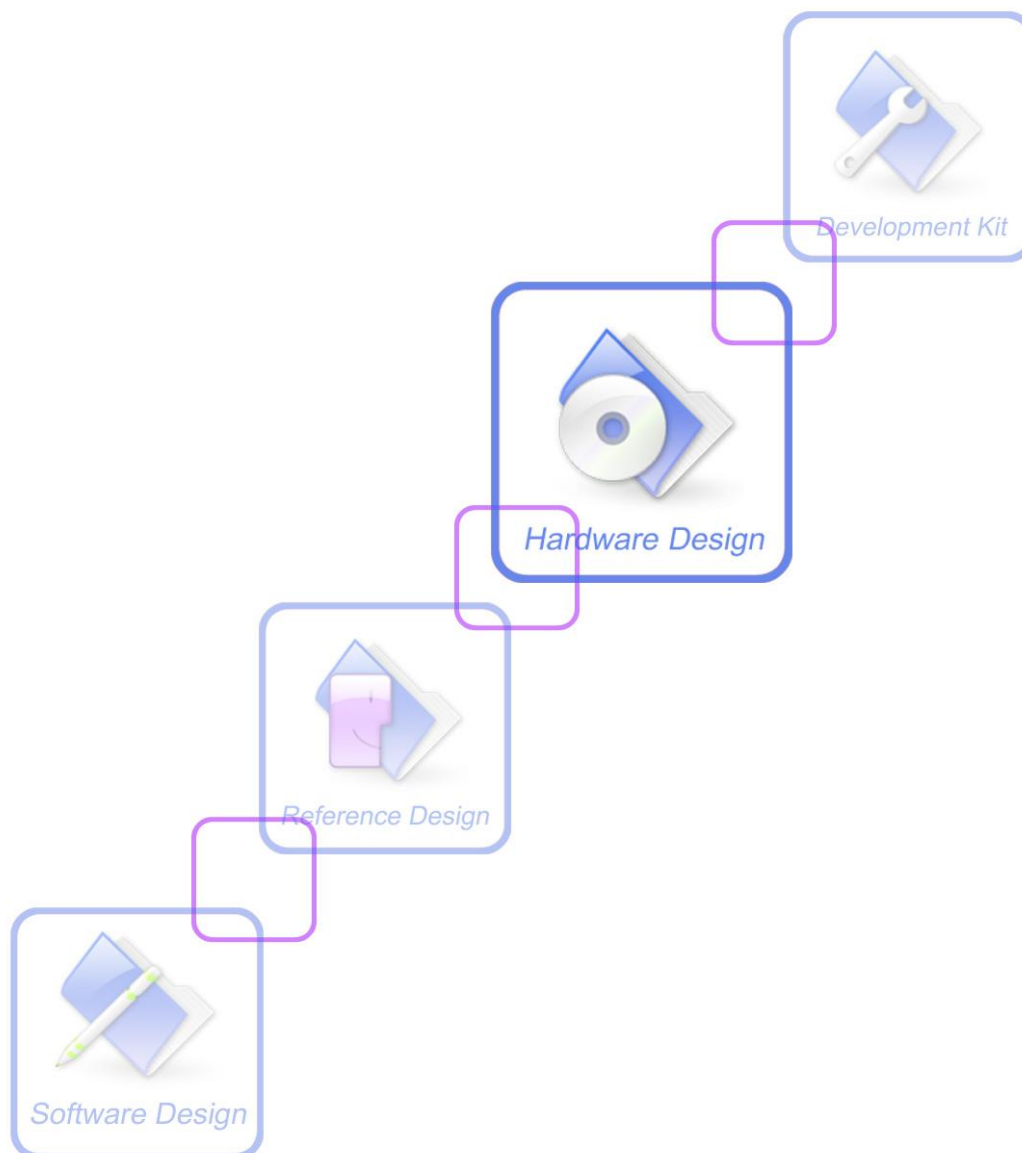




A company of SIM Tech

# Locus Manual for MTK GNSS Platform



<b>Document Title:</b>	Locus Manual for MTK GNSS Platform
<b>Version:</b>	1.00
<b>Date:</b>	2014-02-20
<b>Status:</b>	Release
<b>Document Control ID:</b>	Locus Manual for MTK GNSS Platform V1.00

### General Notes

SIMCom offers this information as a service to its customers, to support application and engineering efforts that use the products designed by SIMCom. The information provided is based upon requirements specifically provided to SIMCom by the customers. SIMCom has not undertaken any independent search for additional relevant information, including any information that may be in the customer's possession. Furthermore, system validation of this product designed by SIMCom within a larger electronic system remains the responsibility of the customer or the customer's system integrator. All specifications supplied herein are subject to change.

### Copyright

This document contains proprietary technical information which is the property of SIMCom Limited., copying of this document and giving it to others and the using or communication of the contents thereof, are forbidden without express authority. Offenders are liable to the payment of damages. All rights reserved in the event of grant of a patent or the registration of a utility model or design. All specification supplied herein are subject to change without notice at any time.

**Copyright © Shanghai SIMCom Wireless Solutions Ltd. 2014**

## Contents

1 LOCUS Introduction.....	4
1.1 LOCUS benefits.....	4
1.2 LOCUS operation.....	4
2 LOCUS PMTK Command List.....	5
2.1 Packet Type 183 – PMTK_LOCUS_QUERY_STATUS .....	5
2.2 Packet Type 184 – PMTK_LOCUS_ERASE_FLASH.....	6
2.3 Packet Type 185 – PMTK_LOCUS_STOP_LOGGER .....	6
2.4 Packet Type 186 – PMTK_LOCUS_LOG_NOW.....	7
2.5 Packet Type 187 – PMTK_LOCUS_CONFIG .....	7
2.6 Packet Type 622 – PMTK_LOCUS_QUERY_DATA .....	7
3 LOCUS Special ACK Description.....	10
4 LOCUS Parse.....	10
4.1 LOCUS dump sample code.....	10
4.2 LOCUS parse sample code .....	10
4.3 LOCUS Flash data format.....	12
5 LOCUS Operation during Normal and PowerSaving Mode.....	13

# 1 LOCUS Introduction

LOCUS is the name of MTK innate logger solution. This locus manual describes data logging on GNSS modules designed by SIMCOM, which are based on the MediaTek's chipsets. This manual serves as a reference tool, currently supported products are as follows:

- SIM28\SIM28C\SIM28M\  
• SIM39EA\SIM39EAU\SIM33ELA\  
• SIM68R\SIM68V\SIM68M \SIM68RB\SIM68VB\SIM68MB.

## 1.1 LOCUS benefits

1. Auto logging data to MTK chip internal flash, no need to wake up the host side.
2. Smart overlapping mechanism to keep the latest logger data (4KB base).
3. Flexible configuration to support most logging types, modes and contents.
4. Logger capability in MTK chip internal flash:
  - (1) With 1 sector flash (64KB), user can log > 16 hours.
  - (2) With AlwaysLocate™, user can log up to 48 hours (2 days) under standard scenario.

## 1.2 LOCUS operation

### Step 1: Save configuration to firmware setting by pre-setup

- Logging type
  - 1. Overlap
  - 2. Full stop
- Logging mode
  - Normal mode (logging per positioning, e.g. 1 sec)
  - Interval mode (logging per pre-setting interval, e.g. 10 secs)
  - AlwaysLocate™ mode (logging with AlwaysLocate™)
  - Fix-only mode (logging when 3D-fix only)
- Logging contents

### Step 2: Send PMTK command to do real-time application (if necessary)

- PMTK183: Query logger status and configuration
- PMTK184: Erase flash
- PMTK185: Stop/Start logger operation

**Step 3: Use PC tool (PowerGPS) or host sample code to dump and parse logger data in module.**

## 2 LOCUS PMTK Command List

This chapter describes commands that can be used to configure settings for LOCUS and to dump out the data.

### 2.1 Packet Type 183 - PMTK\_LOCUS\_QUERY\_STATUS

**Packet Meaning:**

Query LOCUS logging status.

**Command number:**

183

**DataField:**

NONE

**Return:**

\$PMTKLOG, Serial#, Type, Mode, Content, Interval, Distance, Speed, Status, Number, Percent\*CH

Serial#: Logging serial number: 0~65535

Type: Logging type. 0: Overlap, 1: FullStop

Mode: Logging mode. 0x08: Interval logger

Content: Logging contents of configuration

Interval: Logging interval setting (valid when interval mode is selected)

Distance: Logging distance setting (valid when distance mode is selected)

Speed: Logging speed setting (valid when speed mode is selected)

Status: Logging status. 1: Stop Logging, 2: Logging

Number: Logging number of data record

Percent: Logging life used percentage (0~100%)

**Example:**

Inspu: \$PMTK183\*38<CR><LF>

Output: \$PMTKLOG,456,0,11,31,2,0,0,0,3769,46\*48

\$PMTK001,183,3\*3A

Note: input a <CR><LF> is need, which means as follows:

<CR>: \r, ASCII value is 0x0D

<LF>: \n, ASCII value is 0x0A

**Details:**

It is configurable in custom firmware build.

**Mode:**

Logging mode, e.g. 0x08 means Interval logger, details as follows:

//1<<0: AlwaysLocate™ mode (logging with AlwaysLocate™)

//1<<1: Fix only mode (logging when 3D-fix only)

//1<<2: Normal mode (logging per positioning. e.g. 1 sec)

//1<<3: Interval mode (logging per pre-setting interval. e.g. 15 secs)

//1<<4: Distance mode logger (by distance. e.g. 50m)

//1<<5: Speed mode (by speed. e.g. 10m/s)

**Content:**

Logging contents of configuration(by Bit-Map).

```
#define LOCUS_CONTENT_UTC (1<<0) // 4-byte
#define LOCUS_CONTENT_VALID (1<<1) // 1-byte
#define LOCUS_CONTENT_LAT (1<<2) //4-byte
#define LOCUS_CONTENT_LON (1 <<3) //4-byte
#define LOCUS_CONTENT_HGT (1<<4) //2-byte
#define LOCUS_CONTENT_SPD (1 <<5) // 2-byte
#define LOCUS_CONTENT_TRK (1<<6) // 2-byte
#define LOCUS_CONTENT_HDOP (1<<10) // 2-byte
#define LOCUS_CONTENT_NSAT (1<<12) // 1-byte
```

## 2.2 Packet Type 184 – PMTK\_LOCUS\_ERASE\_FLASH

**Packet Meaning:**

Erase Logger Flash

**Command number:**

184

**DataField:**

\$PMTK184, Type

Type: Erase type. 1:erase all logger internal flash data

**Example:**

Inspu: \$PMTK184,1\*22<CR><LF>

Outspu: \$PMTK001,184,3\*3D

## 2.3 Packet Type 185 – PMTK\_LOCUS\_STOP\_LOGGER

**Packet Meaning:**

Stop or Start logging data

**Command number:**

185

**DataField:**

\$PMTK185, Status

Status: Stop logging. 1: Stop logging

0: Start logging

**Example:**

Stop logging:

Input: \$PMTK185,1\*23<CR><LF>

Output: \$PMTK001,185,3\*3C

Start logging:

Input: \$PMTK185,0\*22<CR><LF>

Output: \$PMTK001,185,3\*3C

## 2.4 Packet Type 186 - PMTK\_LOCUS\_LOG\_NOW

**Packet Meaning:**

Snapshot write log

**Command number:**

186

**DataField:**

\$PMTK186, Type

Type: 1 means snapshot log data.

**Example:**

Input: \$PMTK186,1\*20<CR><LF>

Output: \$PMTK001,186,3\*3F

## 2.5 Packet Type 187 - PMTK\_LOCUS\_CONFIG

**Packet Meaning:**

Configure LOCUS setting by command.

**Command number:**

187

**DataField:**

\$PMTK187,Type,Setting

Type: 1 means interval mode

Setting: New setting instead of the original configuration (e.g. change to 5 seconds interval as in the example below).

**Example:**

Input: \$PMTK187,1,5\*38<CR><LF>

Output: \$PMTK001,187,3\*3E

## 2.6 Packet Type 622 - PMTK\_LOCUS\_QUERY\_DATA

**Packet Meaning:**

Dump LOCUS flash data

**Command number:**

622

**DataField:**

\$PMTK622, Type

Type: 0: dump full flash data

1: dump partial in used flash data

Output data:

PMTKLOX packet type:

Type1: LOCUS start (n is the number PMTKLOX packets will be sent)

PMTKLOX,0,n

Type2: LOCUS data (data will be sent by 8-byte HEX sting, at most 24 events)

("FFFFFFFF" if empty) commas separate one log item.

Type3: LOCUS end

\$PMTKLOX,2\*47

\$PMTK001,622,3\*36

LOCUS data format and size as follows:

UTC: 4 bytes

Fix: 1 byte

Lat: 4 bytes

Lon: 4 bytes

Alt: 2 bytes

Spd: 2 bytes

Sat: 2 bytes

Cks: 1 byte

**Example:**

Input: \$PMTK622,1\*29<CR><LF>

Output:

\$PMTKLOX,0,171\*5E

\$PMTKLOX,1,0,0100000B,7F000000,05000000,00007B0B,00000000,00000000,00000000,00000000,00000000,007FFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,000C8C14,877CFC52,0475C6F9,414BB5F2,42250000,00720043,8C7CFC52,0475C6F9,414BB5F2\*53

\$PMTKLOX,1,1,42250000,00480173,917CFC52,0475C6F9,414BB5F2,42250000,00D800FF,967CFC52,0475C6F9,414BB5F2,42250000,00710051,9B7CFC52,0475C6F9,414BB5F2,42250000,0022010E,A07CFC52,0475C6F9,414BB5F2,42250000,00C200D4,A57CFC52,0475C6F9\*59

\$PMTKLOX,1,2,414BB5F2,4225000,00150107,AA7CFC52,0475C6F9,414BB5F2,42250000,00B500A9,AF7CFC52,0475C6F9,414BB5F2,42250000,0015000C,B47CFC52,0475C6F9,414BB5F2,42250000,00F300F1,B97CFC52,0475C6F9,414BB5F2,4225000,00880087,BE7CFC52\*27

\$PMTKLOX,1,3,0475C6F9,414BB5F2,42250000,0004000C,C37CFC52,0475C6F9,414BB5F2,42250000,00C300B6,C87CFC52,0474C6F9,414AB5F2,42250000,0021005F,CD7CFC52,0474C6F9,414AB5F2,42250000,00FF0084,D27CFC52,0474C6F9,414AB5F2,42250000,009D00F9\*53

\$PMTKLOX,1,4,D77CFC52,0474C6F9,414AB5F2,42250000,00670107,DC7CFC52,0473C6



F9,414AB5F2,42240000,00CF00A3,E17CFC52,0473C6F9,414AB5F2,422  
40000,00670137,E67CFC52,0473C6F9,414AB5F2,42240000,00560000,  
EB7CFC52,0473C6F9,414AB5F2,42240000\*56  
\$PMTKLOX,1,5,008400DF,F07CFC52,0473C6F9,414AB5F2,42240000,00  
9A00DA,F57CFC52,0473C6F9,414AB5F2,42240000,00B900FC,FA7CFC52  
,0473C6F9,414AB5F2,42240000,00020048,FF7CFC52,0473C6F9,414AB  
5F2,42240000,00290167,047DFC52,0473C6F9,414AB5F2\*5B  
\$PMTKLOX,1,6,42240000,00E10054,097DFC52,0473C6F9,414AB5F2,42  
240000,006101D8,0E7DFC52,0473C6F9,414AB5F2,42240000,002A009  
5,137DFC52,0473C6F9,414AB5F2,42240000,000200A0,187DFC52,0473  
C6F9,414AB5F2,42240000,000701AF,1D7DFC52,0473C6F9\*25  
\$PMTKLOX,1,7,414AB5F2,42240000,005400F8,227DFC52,0473C6F9,41  
4AB5F2,42240000,00FF006C,277DFC52,0473C6F9,414AB5F2,42240000  
,009D000B,2C7DFC52,0473C6F9,414AB5F2,42240000,000F0092,317DF  
C52,0473C6F9,414AB5F2,42240000,00E90069,367DFC52\*25  
\$PMTKLOX,1,8,0473C6F9,414AB5F2,42250000,003E01B9,3B7DFC52,04  
73C6F9,414AB5F2,42250000,000B0181,407DFC52,0473C6F9,414AB5F2  
,42250000,00D30023,417DFC52,0473C6F9,414AB5F2  
,42250000,001F00EE,467DFC52,0473C6F9,414AB5F2,42250000,00230  
0D5\*5D  
\$PMTKLOX,1,9,4B7DFC52,0473C6F9,414AB5F2,42250000,00C6003D,50  
7DFC52,0473C6F9,414AB5F2,42250000,003E01DF,557DFC52,0473C6F9  
,414AB5F2,42250000,004C01A8,5A7DFC52,0473C6F9,414AB5F2,42250  
000,00AA0040,5F7DFC52,0473C6F9,414AB5F2,42250000\*50  
\$PMTKLOX,1,10,0090007F,647DFC52,0473C6F9,414AB5F2,42250000,0  
0930047,697DFC52,0474C6F9,414AB5F2,42250000,000000DE,6E7DFC5  
2,0474C6F9,414AB5F2,42250000,00B3006A,737DFC52,0474C6F9,414A  
B5F2,42250000,002100E5,787DFC52  
,0474C6F9,414AB5F2\*11  
...  
\$PMTKLOX,1,133,009E0071,B388FC52,0476C6F9,414AB5F2,42240000,  
005601A5  
,B888FC52,0476C6F9,414AB5F2,42240000,001101E9,BD88FC52,0476C  
6F9,414AB5F2,42240000,00D50029,C288FC52,0476C6F9,414AB5F2,42  
240000,006401E6,C788FC52,0476C6F9,414AB5F2\*54  
\$PMTKLOX,1,134,42240000,002400A2,CC88FC52,0476C6F9,414AB5F2,  
42240000,00E40069,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFF  
FF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFF  
FFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF\*2B  
\$PMTKLOX,1,135,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,  
FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFF  
FFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFF  
FFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFF  
\*5F

...

```
$PMTKLOX,1,170,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,
FFFFFFFF,FFFFFFFF
,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFFFFF,FFFFF
FFF,FFFFFFFF,FFFFFFFF*5E
$PMTKLOX,2*47
$PMTK001,622,3*36
```

## 3 LOCUS Special ACK Description

Several LOCUS special ACK as follows:

No.	Output	Description of ACK
1	\$PMTKLOG,FULL_STOP*3E	Notify Host LOCUS become full and stop
2	\$PMTKLOG,WRITE_ERR*2D	Notify Host LOCUS write error
3	\$PMTKLOG,ERASE_ERR*30	Notify Host LOCUS erase error

**Table 1: LOCUS special ACK description for Host reference**

## 4 LOCUS Parse

Dump the LOCUS data in a binary file and parse the data.

### 4.1 LOCUS dump sample code

Sample the code for LOCUS dumping when the host issues PMTK622. PMTKLOX is used to dump the contents of Flash data(e.g. 128 KB).

PMTKLOX packet types and format see “**2.6 Packet Type 622 – PMTK\_LOCUS\_QUERY\_DATA**”.

Refer to the LOCUS attachment for sample codes:

Locus\_Dump\_Sample\_Code.cpp

Locus\_Dump\_Sample\_Code.h

### 4.2 LOCUS parse sample code

Sample code for LOCUS parsing. To test the function, you need to dump the dump sentences into **Locus Manual for MTK GNSS Platform V1.00**

a file in your PC. Below is the API function to feed LOCUS binary file for the parser:

```
//-----
// Locus_Parser_Start
// Function : Parse Locus' data with binary format
// Parameter:
//      InputDataFileName : The file name of Locus' binary data
// Return : 0 :success
//      1 : The type of files is wrong
//      2 : Fail to Create/Open output file
//      3 : File name is NULL
//      4 : Fail to read locus.bin
//-----
```

int Locus\_Parser\_Start( char\* InputData File Name);

• Refer to the LOCUS attachment for sample codes:

- LocusParser.cpp
- LocusParser.h

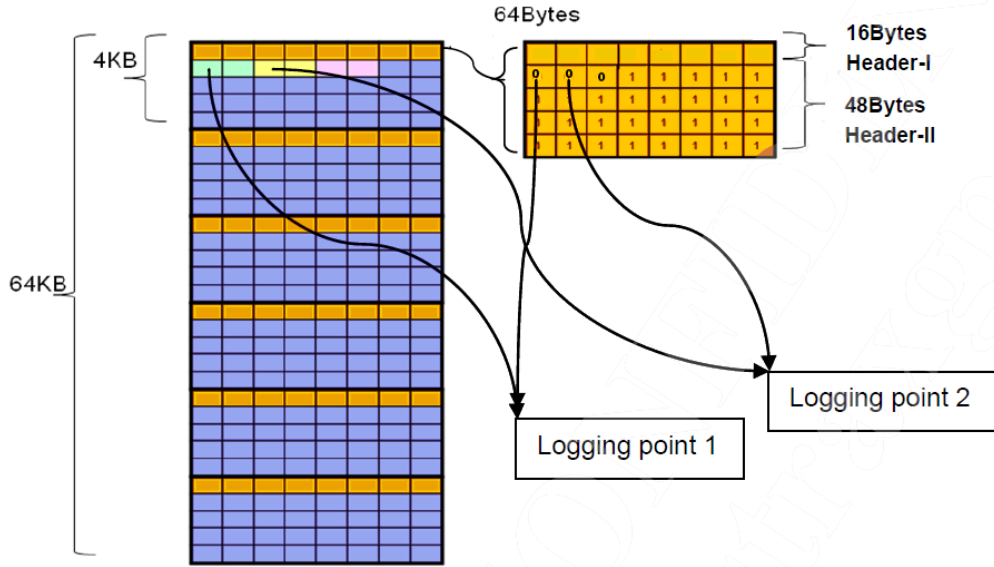
Details of logging content availability in different functional modes are described in Table 2.

No.	Name	Size (bytes)	UTC (4)	Fix_Type (1)	Lat (4)	Lon (4)	Alt (2)	Speed (2)	Heading (2)	HDO P(2)	SatNo (1)	Checksum(1)
1	Basic	16	•	•	•	•	•					•
2	Racing	20	•	•	•	•	•	•	•			•
3	Search	19	•	•	•	•	•			•	•	•
4	Saving	13	•		•	•						•
5	All	23	•	•	•	•	•	•	•	•	•	•

**Table 2: Logging content availability**

### 4.3 LOCUS Flash data format

Locus header/data arrangement:



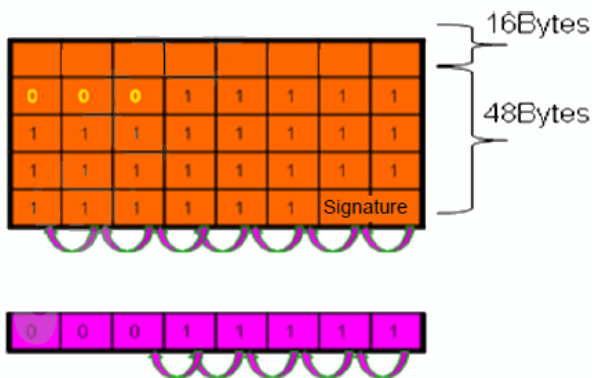
Locus Header-I in each sector (4KB): Every 4KB records its own logging format.

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Serial #	Type		Mode		Content	Interval		Distance		Speed	Checksum				

Signature (4Bytes) is located at the end of Header-II.

C	D	E	F
Signature			

Locus Header-II in each sector (4KB): bitmap 0 means data logged, and 1 means data are not used.



# 5 LOCUS Operation during Normal and PowerSaving Mode

Table 3 shows the logging mode behavior during positioning normal mode to power saving mode.

LOCUS config	Positionin	Normal mode	Power saving mode (Periodic/AlwaysLocate running)
AL		No logging	Logging once before goto sleep
Normal		Logging per fix	Logging per fix
Customization		Logging when over the customization criterion	Logging when over the customization criterion
AL + Normal		Logging per fix	Logging once before goto sleep
AL + Customization		Logging when over the customization criterion	Logging once before goto sleep
Normal + Customization		Logging per fix	Logging per fix
AL + Normal + Customization		Logging per fix	Logging once before goto sleep

**Table 3: Logging Mode behavior during Normal mode to PowerSaving mode**

Note:

1. The “Fix Only” is compatible with all other options.
2. The “AL” is used to save flash data and only Log once before going to sleep when AL running.
3. The “Interval”, ”Distance”, ”Speed” are called “Customization” in this table, and all of them are && condition with each other condiguration.