MAESTRO GPS - APPS NOTE

EXTENDED EPHEMERIS RECEIVER LOAD REV 0.1

A DESCRIPTION ON HOW TO LOAD MAESTRO'S GPS MODULES WITH EE DATA FILES

WWW.MAESTRO-WIRELESS.COM
EMAIL: CONTACT@MAESTRO-WIRELESS.COM
TEL: (+852) 2869 0688
FAX: (+852) 2525 4701
ADDRESS: ROOM 3603-3609, 36/F, 118 CONNAUGHT ROAD WEST, SHEUNG WAN, HONG KONG

Revision history

Rev.	Date	Details	Originated by
0.0	21 March 2011	First issue	Samuel Chereau
0.1	3 April 2011	CSR Approved, small mistype correction	Samuel Chereau

This manual is written without any warranty.

Maestro Wireless Solutions Ltd. reserves the right to modify or improve the product and its accessories which can also be withdrawn without prior notice.

Besides, our company stresses the fact that the performance of the product as well as accessories depends not only on the proper conditions of use, but also on the environment around the places of use.

Maestro Wireless Solutions Ltd. assumes no liability for damage incurred directly or indirectly from errors, omissions or discrepancies between the modem and the manual.

3603-9, 36/F., 118 Connaught Road West, Sheung Wan, Hong Kong
contact@maestro-wireless.com

Contents

3603-9, 36/F., 118 Connaught Road West, Sheung Wan, Hong Kong
contact@maestro-wireless.com

List of Figures

List of Tables

Chapter 1

Introduction

This document describes the way to work with a GPS receiver or antenna module incorporated within a unit with an additional processor and memory resources. The document covers the third part of a complete process of retrieving Extended Ephemeris (EE) data from Maestro Wireless Solutions's EE server, transferring them to the units in the field and then finally loading the receiver with the EE data.

The drawing 1.0.1 gives an overview.

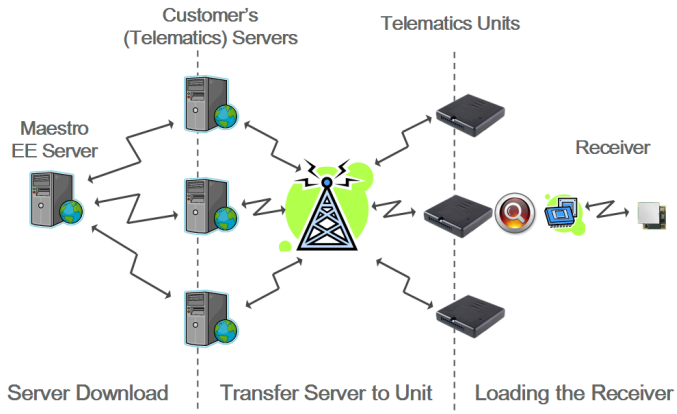

Figure 1.0.1: Overview of EE use

The first part "**Server Download**" is described within the application note "**GPS - Apps Note - EE Server Download**". The second part "**Transfer Server to Unit**" is a custom specific implementation of a file transfer and storage. The overview shows GSM / GPRS as the transfer mediumTypically FTP could be used to transmit the file from a server to a unit in the field. The file can then be stored in flash, SRAM, on an SD card, with or without file system in the unit. Alternative transmission methods could be WLAN or simply via a direct serial connection (e.g. USB connection). In the following chapters the third part "**Loading the Receiver**" of the complete process is described.

> **Note:** The GPS receiver or GPS antenna module must hold SiRF FW revision 3.6.0 or higher. Please verify the FW on your receiver (according information is giving on the label of the module! The part of the firmware supporting Extended Ephemeris is referred to as Embedded Client Location Manager (ECLM).

Chapter 2

Preparation

2.1 Setting the Baud Rate

Dependent on the validity of the EE data (currently 1, 3, 5, and 7 days are supported) the file sizes are accordingly from about 10kBytes to about 70kBytes. The default setting of firmware 3.6.0 is NMEA and 4800 baud. It is possible to use both SiRFbinary[1] and NMEA mode for the transfer. Anyhow it is recommended to switch the baud rate to a higher value, e.g. 57600 baud, in order to speed up this process. This can be done using the command **PSRF100**, described here in detail with an example for NMEA mode and 57600 baud, refer to Table 2.1:

– $PSRF100,1,57600,8,1,0*36

Name	Example	Description
Message ID	$PSRF100	PSRF100 protocol header
Protocol	1	0 SiRF binary / 1 NMEA
Baud	57600	4800, 9600, 19200, 38400, 57600, 115200
DataBits	8	8, 7
StopBits	1	0, 1
Parity	0	0 none / 1 odd / 2 even
Checksum	*36	End of message termination

Table 2.1: Serial Port Setup

One can return to the original baud rate of 4800 baud by sending the following command:

– $PSRF100,1,4800,8,1,0*0E

For switching to binary mode and back to NMEA, please refer to the Description in "**GPS - Apps Note - Firmware GSC3**".

[1] SiRF protocol is only valid for 8 data bits, 1 stop bit and no parity

2.2 Duration of Transfer

Protocol and baudrate will have an affect on the amount of time required to transfer data to the receiver. Refer to the Tables 2.2, 2.3, 2.4 when deciding whether to use SiRF binary or NMEA and which baud rate (It is not complete, but an excellent reference).

Baudrate	NMEA	SiRF Binary
4800	160s	80s
9600	55s	30s
57600	10s	4s

Table 2.2: Transfer time for a 1 Day Extended Ephemeris File

Baudrate	NMEA	SiRF Binary
4800	300s	180s
9600	130s	80s
57600	33s	20s

Table 2.3: Transfer time for a 3 Day Extended Ephemeris File

Baudrate	NMEA	SiRF Binary
4800	660s	330s
9600	270s	170s
57600	77s	30s

Table 2.4: Transfer time for a 7 Day Extended Ephemeris File

2.3 Data Transfer

The data transfer can be split into two stages, the initialization and the real transfer following it. The flow chart 2.3.1 on the next page shall serve as an overview and explanation.

2.3.1 Initialization Stage

Before the Extended Ephemeris File can be transferred to the GPS receiver, the following process must be carried out:

1. Send “Start Download” Message

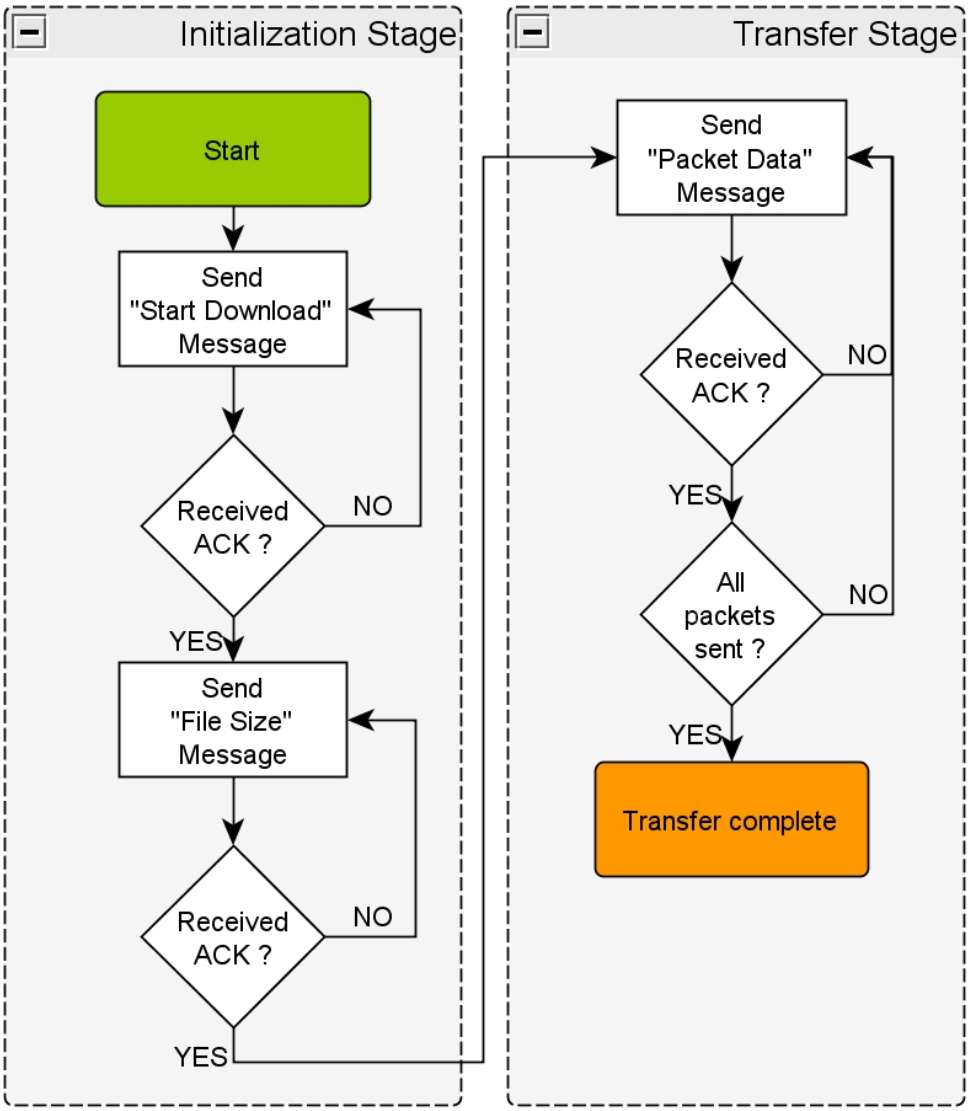

Figure 2.3.1: Data transfer

2. Wait for “Ack/Nack” Message from receiver
3. Send “File Size” Message
4. Wait for “Ack/Nack” Message from receiver

Please see the following sections for descriptions of each message.

NMEA:

- Start Download – Section 3.1
- File Size – Section 3.2
- Ack/Nack – Section 3.5

SiRF Binary:

- Start Download – Section 4.1
- File Size – Section 4.2
- Ack/Nack – Section 4.5

2.3.2 Transfer Stage

Once the initialization stage has completed, the following process may take place:

1. First file packet is sent to the receiver with the “Packet Data” message.
2. Wait for “Ack/Nack” message from receiver.
3. Loop between steps 1 and 2 until the entire file is transferred.

See the following sections for descriptions of each message.

NMEA:

- Packet Data – Section 3.3
- Ack/Nack – Section 3.5

SiRF Binary:

- Packet Data – Section 4.3
- Ack/Nack – Section 4.5

2.3.3 Additional Commands

In order to request the age of the EE data already loaded into the receiver it is possible to request their age. The response will return the age in seconds along with the total prediction interval. Therefore the remaining prediction time can be calculated.

Chapter 3

NMEA Data Transfer Messages

3.1 Start Download

ECLM Start Download - MsgID $PSRF114, SubMsgID 0x16

This message is to indicate to the GPS receiver that the host processor wants to initiate the EE file download. The SubMsgID for this message is fixed to 0x16. Refer to the Table 4.1 on page 16.

Example: $PSRF114,16*08

Name	Example	Unit	Description
Message ID	$PSRF114	-	ECLM Download Packet Header
Sub message ID	0x16	-	SubID $22_{(dec)}$: Sub Id for Start Download
Checksum	*08	-	-
<CR><LF>	-	-	End of message termination

Table 3.1: ECLM Start Download Message Description

Note: This message would be acknowledged to indicate SUCCESS/FAILURE.

- SUCCESS: would be acknowledged with “Ack: $PSRF156” using Command Acknowledgment, NMEA Message ID $PSRF156, Sub Msg ID 0x20 (see Section 3.5).
- FAILURE: would be acknowledged with “Nack: $PSRF156” using Command Negative Acknowledgment, NMEA Message ID $PSRF156, Sub Msg ID 0x20 (see Section 3.5).

3.2 Send File Size

ECLM Send File Size - MsgID $PSRF114, SubMsgId 0x17

This message is sent from the host processor to the GPS Receiver to indicate the size of the EE file to be downloaded. The SubMsgID for this message is fixed to 0x17. Refer to the Table 3.2 on page 13.

Example: $PSRF114,17,2859*23

In the example a file size of 10329 bytes is used which corresponds to 0x2859.

Note: This message would be acknowledged to indicate SUCCESS/FAILURE.

- SUCCESS: would be acknowledged with “Ack: $PSRF156” using Command Acknowledgment, NMEA Message ID $PSRF156, Sub Msg ID 0x20 (see Section 3.5).
- FAILURE: would be acknowledged with “Nack: $PSRF156” using Command Negative Acknowledgment, NMEA Message ID $PSRF156, Sub Msg ID 0x20 (see Section 3.5).

3.3 Send Data

ECLM Packet Data - MsgID $PSRF114, SubMsgId 0x18

This message is sent from the host processor to the GPS receiver to actually transmit the file’s data. The SubMsgID for this message is fixed to 0x18. Refer to the Table 3.3 on the following page.

Example: $PSRF114,18,1,32,62,12,31,6,3,2,7,d9,7,7,0,0,39,6d,8f,12,0,0,0,0,0,0,1,2d,9a,e7,5,2,ff,fe,28,5*3d

This is the first package, so packet Sequence No = 0x1 and the packet length is 20 (0x14).

Note: This message would be acknowledged to indicate SUCCESS/FAILURE.

- SUCCESS: would be acknowledged with “Ack: $PSRF156” using Command Acknowledgment, NMEA Message ID $PSRF156, Sub Msg ID 0x20 (see Section 3.5).
- FAILURE: would be acknowledged with “Nack: $PSRF156” using Command Negative Acknowledgment, NMEA Message ID $PSRF156, Sub Msg ID 0x20 (see Section 3.5).

3.4 Request EE Age

ECLM Get EE Age - MsgID $PSRF114, SubMsgId 0x1A

This message is sent from the host processor to the GPS receiver to get the age of the EE data of single satellites. The SubMsgID for this message is fixed to 0x1A. Refer to the Table 3.6 on page 15.

Example: $PSRF114,1a,1*42

This message requests age of satellite #1.

Note: This message would be acknowledged to indicate SUCCESS/FAILURE.

- SUCCESS: would be acknowledged with “Ack: $PSRF156” using Command Acknowledgment NMEA Message ID $PSRF156, Sub Msg ID 0x22 with EE Age (see Section 3.6) **this is the specific response for this request**.
- FAILURE: would be acknowledged with “Nack: $PSRF156” using Command Negative Acknowledgment NMEA Message ID $PSRF156, Sub Msg ID 0x20 (see Section 3.5).

3.5 Ack / Nack Responses

Ack/Nack Response - Message ID $PSRF156, SubMsgID 0x20

Name	Example	Unit	Description
Message ID	$PSRF114	-	ECLM Download Packet Header
Sub message ID	0x17	-	SubID $22_{(dec)}$: Sub Id for Start Download
File Length	0x2859	-	Length of the Extended Ephemeris File to be downloaded in bytes. Length of file does not include any overhead from adding start sequence, message checksums, or end sequences.
Checksum	*23	-	-
<CR><LF>	-	-	End of message termination

Table 3.2: ECLM File Size Message Description

Name	Example	Unit	Description
Message ID	$PSRF114	-	Message Id
Sub message ID	0x18	-	SubId: SGEE Packet Data
Packet Sequence No	0x1	-	Packet Sequence number of the current packet Starting from 1
Packet Length	32	bytes	Length of this packet. Number of bytes in decimal. Max. number is 400 (decimal).
Packet Data	62, 12, 31, 6, 3, 2, 7, d9, 7, 7, 0, 0, 39, 6d, 8f, 12, 0, 0, 0, 0, 0, 0, 1, 2d, 9a, e7, 5, 2, ff, fe, 28, 5	-	EE Data in this packet of length Packet Length
Checksum	*3d	-	-
<CR><LF>	-	-	End of message termination

Table 3.3: ECLM Send Data Message Description

Name	Example	Unit	Description
Message ID	$PSRF114	-	Message Id
Sub message ID	0x1a	-	SubId: EE Age
Satellite ID	0x1	-	Satellite ID (in hex) for which age is requested
Checksum	*42	-	-
<CR><LF>	-	-	End of message termination

Table 3.4: ECLM Get EE Age Message Description

This is the receiver's response to the message $PSRF114 with SubMsgIDs 0x16, 0x17, 0x18, 0x1a. The SubMsgID for this message is fixed to 0x20. The Ack Sub ID corresponds to the SubMsgID of the original message that triggered this response. Refer to the Table 3.5.

Example: $PSRF156,20,72,16,0,0*09

This example holds Sub Message ID = 0x20, AckMid = 114, AckSid = 0x16, Ack, Reason = SUCCESS. AckMid and AckSid are the Message IDs and SubID of the message for which this response Message will be sent.

Name	Example	Unit	Description
Message ID	$PSRF156	-	ECLM Response
Sub message ID	0x20	-	SubID $32_{(dec)}$: Sub Id for Ack/Nack
Ack Message ID	0x72	-	MsgID $114_{(dec)}$: Msg Id for ECLM Download
Ack Sub ID	0x16	-	SubID: Sub Id for ECLM Start Download. This field may take values 0x16, 0x17, 0x18, or 0x1a to ACK corresponding SID's.
Ack/Nack	0x0	-	0: Ack / 1: Nack
Reason	0x0	-	ECLM_SUCCESS = 0 ECLM_SPACE_UNAVAILABLE = 1 /* if flash space is not sufficient */ ECLM_PKT_LEN_INVALID = 2 /* if pkt length field is out of range */ ECLM_PKT_OUT_OF_SEQ = 3 /* if pkt received is out of sequence */ ECLM_DOWNLOAD_SGEE_NONEWFILE = 4 /* no new file */ ECLM_DOWNLOAD_CORRUPTFILE_ERROR = 5 /* corrupt file */ ECLM_DOWNLOAD_GENERIC_FAILURE = 6 /* generic failure */ ECLM_API_GENERIC_FAILURE = 7 /* generic failure calling CLM API */
Checksum	*08	-	-
<CR><LF>	-	-	End of message termination

Table 3.5: ECLM Message Response Description

3.6 Get Age Response

Get Age Response - Message ID $PSRF156, SubMsgID 0x1a

This is the receiver's response on the message $PSRF114, SubMsgID 0x1a. The SubMsgID for this message is fixed to 0x22. Refer to the Table 3.6 on the next page.

Example: $PSRF156,22,7da8,15180*3e

This example holds Sub Message ID = 0x22, the acknowledge itself, the age value of 0x7da8 and the prediction interval of 0x15180.

Name	Example	Unit	Description
Message ID	$PSRF156	-	Message Id
Sub message ID	0x22	-	SubId: EE Age
EE Age	0x7da8	-	Age of the satellite data in seconds
Prediction Interval	0x15180	second	Total validity period of the extended ephemeris in seconds. Prediction interval minus EE age would result in the remaining validity period of the EE file.
Checksum	*3e	-	-
<CR><LF>	-	-	End of message termination

Table 3.6: ECLM Get EE Age Response

Chapter 4

SiRF Binary Data Transfer Messages

4.1 Start Download

ECLM Start Download - Msg ID 0xe8, SubMsgID 0x16

This message is to indicate to the GPS receiver that the host processor wants to initiate the EE file download. The SubMsgID for this message is fixed to 0x16. Refer to the Table 4.1.

Example: Message = a0 a2 00 02 e8 16 00 fe b0 b3

```
a0 a2 00 02 – Start Sequence and Payload Length
e8 16 – Payload
00 fe b0 b3 – Message Checksum and End Sequence
```

Name	Bytes	Value(hex)		Unit	Description
		Scale	E.g.		
Message ID	1U	-	0xe8	-	Standard host applications to receive communication Message ID
Sub message ID	1U	-	0x16	-	SubID to indicate Start Download Message

Table 4.1: ECLM Start Download Payload Description

Playload length: 2 bytes.

Note: This message would be acknowledged to indicate SUCCESS/FAILURE.

- SUCCESS/FAILURE: would be acknowledged with Command Acknowledgment, SSB Message ID 0x38, Sub Msg ID 0x20 (see Section 4.5).

4.2 Send File Size

ECLM Send File Size - Msg ID 0xe8, SubMsgID 0x17

This message is sent from the host processor to the GPS Receiver to indicate the size of the EE file to be downloaded. The SubMsgID for this message is fixed to 0x17. Refer to the Table 4.2 on the next page.

Example: Message = a0 a2 00 06 e8 17 00 00 28 59 01 80 b0 b3

a0 a2 00 06 – Start Sequence and Payload Length
e8 17 00 00 28 59 – Payload
01 80 b0 b3 – Message Checksum and End Sequence

In the example a file size of 10329 bytes is used which corresponds to 0x2859.

Name	Bytes	Value(hex)		Unit	Description
		Scale	E.g.		
Message ID	1U	-	0xe8	-	Standard host applications to receive communication Message ID
Sub message ID	1U	-	0x17	-	SubID to indicate Start Download Message
File length	4U	-	0x00 0x00 0x28 0x59	-	Length of the Extended Ephemeris File to be downloaded in bytes. Length of file does not include any overhead from adding start se-quence, message checksums, or end sequences. (Big endian)

Table 4.2: ECLM File Size Payload Description

Playload length: 6 bytes.

Note: This message would be acknowledged to indicate SUCCESS/FAILURE.

- SUCCESS/FAILURE: would be acknowledged with Command Acknowledgment, SSB Message ID 0x38, Sub Msg ID 0x20 (see Section 4.5).

4.3 Send Data

ECLM Packet Data - Msg ID 0xe8, SubMsgID 0x18

This message is sent from the host processor to the GPS receiver to actually transmit the file's data. The SubMsgID for this message is fixed to 0x18. Refer to the Table Table 4.3 on page 18.

Example: Message = a0 a2 00 26 e8 18 00 01 00 20 62 12 31 06 03 02 07 d9 07 07 00 00 39 6d 8f 12 00 00 00 00 00 00 01 2d 9a e7 05 02 ff fe 28 05 07 e6 b0 b3

a0 a2 00 26 – Start Sequence and Payload Length
e8 18 00 01 00 20 62 12 31 06 03 02 07 d9 07 07 00 00 39 6d 8f 12 00 00 00 00 00 00 01 2d
07 e6 b0 b3 – Message Checksum and End Sequence

As this is the first package packet Sequence No = 0x1; packet length is 32 (0x20).

Playload length: 6 + <Packet Length> bytes.

Note: This message would be acknowledged to indicate SUCCESS/FAILURE.

- SUCCESS/FAILURE: would be acknowledged with Command Acknowledgment, SSB Message ID 0x38, Sub Msg ID 0x20 (see Section 4.5).

Name	Bytes	Value(hex)		Unit	Description
		Scale	E.g.		
Message ID	1U	-	0xe8	-	Standard host applications to receive communication Message ID
Sub message ID	1U	-	0x18	-	SubID to indicate Start Download Message
Packet Sequence Number	2U	-	0x00 0x01		Packet sequence number of the current packet starting from 1. (big endian)
Packet length	2U	-	0x00 0x20		Length of the Extended Ephemeris data in current packet in bytes (hex value). Value includes only size of EE data and not start sequence, payload length, message check-sums, or end sequences. Max. number is 400 (decimal) / 0x190 (hex). (big endian)
Packet data	packet length	-	62 12 31 06 03 02 07 d9 07 07 00 00 39 6d 8f 12 00 00 00 00 00 00 01 2d 9a e7 05 02 ff fe 28 05	-	EE data of length indicated in packet length of the message. (big endian)

Table 4.3: ECLM Packet Data Payload Description

4.4 Request EE Age

ECLM Get SGEE Age - Msg ID 0xe8, SubMsgID 0x11

This message is sent from the host processor to the GPS receiver to get the age of the EE data of single satellites. The SubMsgID for this message is fixed to 0x1a. Refer to the Table Table 4.4 on page 19.

Example: Message = a0 a2 00 03 e8 1a 01 01 03 b0 b3

```
a0 a2 00 03 – Start Sequence and Payload Length
e8 1a 01 – Payload
01 03 b0 b3 – Message Checksum and End Sequence
```

This message requests age of satellite #1.

Name	Bytes	Value$_{(hex)}$		Unit	Description
		Scale	E.g.		
Message ID	1U	-	0xe8	-	Standard host applications to receive communication Message ID
Sub message ID	1U	-	0x1a	-	SubID to indicate Start Download Message
Sat ID	1U	-	0x01	-	PRN (in hexa) for which EE age is requested

Table 4.4: ECLM Get EE Age Message Description

Playload length: 3 bytes.

Note: This message would be acknowledged to indicate SUCCESS/FAILURE.

- SUCCESS: would be acknowledged along with EE data using Command Acknowledgment, SSB Message ID 0x38, Sub Msg ID 0x22 (see Section 4.6) this is the specific response for this request.
- FAILURE: would be acknowledged with “Rejected: MID_ECLMAckNack” using Command Acknowledgment, SSB Message ID 0x38, Sub Msg ID 0x22 (see Section 4.5).

4.5 Ack / Nack Responses

Ack/Nack Response - Msg ID 0x22, SubMsgID 0x20

This is the receiver’s response tn the message ID 0xE8 with SubMsgIDs 0x16, 0x17, 0x18, or 0x1a. The SubMsgID for this message is fixed to 0x20. The Ack Sub ID corresponds to the SubMsgID of the original message that trig-gered this response. Refer to the Table Table 4.5 on page 21.

Example: Message = a0 a2 00 06 38 20 e8 16 00 00 01 56 b0 b3

```
a0 a2 00 06 – Start Sequence and Payload Length
38 20 e8 16 00 00 – Payload
01 56 b0 b3 – Message Checksum and End Sequence
```

This example is an acknowledge to message 0xe8 with SubID 0x16 (ECLM Start Download).

Playload length: 6 bytes.

4.6 Get Age Response

Get Age Response - Msg ID 0x38, SubMsgID 0x22

This is the receiver's response to the message Get EE Age with Message ID 0xe8, SubMsgID 0x1A. The SubMsgID for this message is fixed to 0x22. Refer to the Table Table 4.6 on page 21.

Example: Message = a0 a2 00 0a 38 22 00 00 80 ea 00 01 51 80 02 96 b0 b3

a0 a2 00 0a – Start Sequence and Payload Length
38 22 00 00 80 ea 00 01 51 80 – Payload
02 96 b0 b3 – Message Checksum and End Sequence

This example holds Sub Message ID = 0x22, acknowledge itself, the age value of 0x80ea and the prediction interval of 0x15180.

Playload length: 10 bytes.

Name	Bytes	Value(hex) Scale	Value(hex) E.g.	Unit	Description
Message ID	1U	-	0x38	-	Standard host applications to receive communication Message ID
Sub message ID	1U	-	0x20	-	SubID to indicate Start Download Message
Ack Msg ID	1U	-	0xe8	-	Ack Message ID 0xe8
Ack Sub ID	1U	-	0x16	-	Ack SubID, ECLM Start Download 0x16
Ack/Nack	1U	-	0x00	-	0 = Ack / 1 = Nack
Ack Nack Reason	1U	-	0x00	-	ECLM_SUCCESS = 0 ECLM_SPACE_UNAVILABLE = 1 /* if flash space is not sufficient */ ECLM_PKT_LEN_INVALID = 2 /* if pkt length field is out of range */ ECLM_PKT_OUT_OF_SEQ = 3 /* if pkt received is out of sequence */ ECLM_DOWNLOAD_SGEE_NONEWFILE = 4 /* no new file */ ECLM_DOWNLOAD_CORRUPTFILE_ERROR = 5 /* corrupt file */ ECLM_DOWNLOAD_GENERIC_FAILURE = 6 /* generic failure */ ECLM_API_GENERIC_FAILURE = 7 /* generic failure calling CLM API */

Table 4.5: ECLM Start Download Ack/Nack Message Description

Name	Bytes	Value(hex) Scale	Value(hex) E.g.	Unit	Description
Message ID	1U	-	0x38	-	Standard host applications to receive communication Message ID
Sub message ID	1U	-	0x22	-	SubID to indicate Get EE Age
EE Age	4U	-	0x00 0x00 0x80 0xea	second	Age of the satellite data (big endian)
Prediction Interval	4U	-	0x00 0x01 0x51 0x80	-	Prediction interval in seconds: Total validity period of the extended ephemeris in seconds. Prediction interval minus EE age would result in the remaining validity period of the EE file. (big endian)

Table 4.6: ECLM Get EE Age Response Description

Chapter 5

Related Information

5.1 Contact

This manual was created with due diligence. We hope that it will be helpful to the user to get the most out of the GPS module.

Inputs regarding errors or mistaken verbalization and comments or proposals to Maestro Wireless Solutions, for further improvements are highly appreciated.

maestro GPS receiver

www.maestro-wireless.com
Email: contact@maestro-wireless.com
Tel: (+852) 2869 0688
Fax: (+852) 2525 4701
Address: Room 3603-3609, 36/F, 118 Connaught Road West, Sheung Wan, Hong Kong

5.2 Related Documents

- GPS - Apps Note - Server Download (Maestro Wireless Solutions)
- GPS - Apps Note - Firmware GSC3 (Maestro Wireless Solutions)
- SiRF Binary Reference Manual (CSR)
- SiRF NMEA Reference Manual (CSR)

5.3 Related Tools

- GPS Cockpit (Maestro Wireless Solutions)
- SiRF GetEE (CSR)

– SiRF Flash (CSR)