

Protocollo ASCII

TraceBox e PowerLac

Rev. 1.5.0

All copyright and industrial rights in this document and in the technical knowledge it contains are owned by Powersoft and/or the third parties rightfully concerned.

No part of this document nor any data herein shall be disclosed, reproduced or used for any purpose whatsoever without the prior written consent of POWERSOFT as foreseen by the law.

Drawings and specifications are subject to change.

All trademarks and registered trademarks are the property of their respective holders.

Printed in Italy.

ELENCO DELLE REVISIONI

REV	DATA	MOTIVO DELLA REVISIONE	Visto
1.0	06/2006	1 ^a EDIZIONE	
1.1	12/07/06	Aggiornamento protocollo	
1.2	2016/01/29	Miglioramento indice e formattazione	
1.3	2016/0315	Modifica al comando \$RESET Spiegazione del protocollo ed esempio di calcolo del checksum	
1.3.1	2016/03/30	Aggiunta la documentazione del tag TOWEXT Aggiunta la documentazione del comando \$PLGT	
1.4.0	2016/05/18	Aggiunti i comandi \$PLSIM e \$PCSIM	
1.5.0	2017/01/17	Aggiunti comandi speciali GE \$ADDAP (aggiunta aeroporti) e \$CLEARAP (cancellazione tabella aeroporti)	

INDICE

1. PREMESSA E MODALITÀ DI NUMERAZIONE DOCUMENTO	7
2. DEFINIZIONE SINTASSI DEL PROTOCOLLO	7
3. <i>IMPOSTAZIONE PARAMETRI DI COMUNICAZIONE</i>	8
\$0 - Aggiornamento parametri di sistema	8
\$PCSIM – Abilita o disabilita pin nella SIM	10
\$PLSIM – Esegue il login in una SIM con pin	10
4. LOCALIZZAZIONE DEL DISPOSITIVO	11
\$10 - Aggiornamento parametri di LOG	11
5. INVIO DELLO STATO DEL DISPOSITIVO	12
\$11 - struttura dati 'struct LOG_Struct' senza ACK	12
\$12 - LOG_Struct con ACK	13
\$18 - LOG_Struct con ACK e stato di dispositivi slave	14
Valori validi per il campo TAG	15
6. GESTIONE I/O	17
\$20 - Aggiornamento parametri ingressi	17
\$30 - Aggiornamento parametri uscite (messaggio non attivo)	18
7. CONFIGURAZIONE PARAMETRI DI SISTEMA	19
\$DC - Correzione differenziale	19
\$SET – impostazione parametri odometro e girobussola	20
\$GET - Lettura parametri odometro e girobussola	20

\$VER - Lettura versione firmware:	21
\$GETPARAM - richiesta parametri	21
\$SETMAXVEL - Impostazione velocità massima	21
\$SEMT - Impostazione Metri Percorsi totali	22
\$SETREFRSEC - Impostazione secondi funzionamento frigo	22
8. GESTIONE LOGIN AUTISTI	23
\$ADDUSER – aggiunta utente e password	23
\$DELUSER – rimozione utente	24
\$SETIDPIN - Abilitazione login autista	24
9. GESTIONE PUNTI NOTEVOLI	25
\$ADDPOINT - Aggiungi punto notevole	25
\$DELPOINTS - Rimuovi punti notevoli	26
\$ADDAP – Aggiungi aeroporto	26
\$CLEARAP – Cancella tabella aeroporti	27
10. INVIO MESSAGGI DI TESTO A BORDO	28
\$MSG – Invio messaggio a bordo	28
11. GESTIONE SOGLIE DI TEMPERATURA	28
\$SETTHRESH - Definizione soglia temperatura	28
\$SELTHRESH - Selezione soglia temperatura	29
12. GESTIONE ALLARMI ANTIFURTO	30
\$ENANTITHEFT - Abilitazione antifurto	30
\$ANTITHEFTTEL - Impostazione numeri telefonici antifurto	30
13. COMUNICAZIONI INTER-DISPOSITIVO	31
\$PLGT – comunicazione seriale master-slave (solo su Powerlac1)	31
14. AGGIORNAMENTO FIRMWARE	32
\$INITDOWNLOAD – cancellazione buffer RAM	32
\$BLOCKDOWNLOAD – scrittura di un blocco di RAM	32
\$IMAGECHECKSUM– verifica checksum immagine in RAM	33

\$WRITEFLASH – copia del RAM buffer nella flash	34
\$HTTPUPDATE – aggiornamento firmware da URL	34
\$RESET – reset di sistema	35
15. APPENDICE A – FLUSSI LOGICI	37
Aggiornamento firmware	37
Richiesta connessione del mezzo	38

1. Premessa e modalità di numerazione documento

Questo documento descrive il protocollo di comunicazione con i dispositivi Mobility di Powersoft (attualmente TraceBox e Powerlac2).

Il documento segue una numerazione a tre cifre secondo questa modalità:

- Il primo numero indica le revisioni principali. Si ha un cambio di revisione principale solo se avvengono modifiche così profonde al protocollo da renderlo incompatibile in tutto o in gran parte con la versione precedente.
- Il secondo numero cambia quando si hanno aggiunta o eliminazione di comandi, oppure modifiche ai parametri di comandi esistenti.
- L'ultimo numero indica le revisioni più marginali, ad esempio una modifica ad una descrizione o una riformattazione.

2. Definizione sintassi del protocollo

Il protocollo ASCII di localizzazione Powersoft ha la seguente sintassi:

\$<CMD>[:<PARAM1>[,<PARAM2>[,<PARAM3>...]]]*<CHECKSUM>

<CMD> è il comando scelto tra quelli riconosciuti dal protocollo.

Se ci sono parametri, dopo <CMD> è presente il simbolo ':' ed i parametri separati da virgole, ed espressi come stringa ASCII anche nel caso di valori numerici. Se non ci sono parametri, dopo <CMD> è immediatamente presente il simbolo '*'.

<checksum> è il valore di controllo a 8 bit, calcolato come XOR di tutti i dati dal carattere \$ fino al carattere immediatamente precedente il simbolo *. Per il calcolo, si considera il valore ASCII dei caratteri.

Ipotesizzando per esempio di avere il comando:

\$LOGSET:0,1*53

Si calcola il checksum come XOR dei caratteri da \$ fino a 1:

\$LOGSET:0,1 → [36] [76] [79] [71] [83] [69] [84] [58] [48] [44] [49]

Eseguendo XOR dei valori si ottiene 53 come atteso.

Il messaggio va scartato se il risultato dell'operazione differisce dal checksum.

3. Impostazione parametri di comunicazione

In questa sezione si definiscono i comandi di configurazione relativi alla comunicazione del PowerLac con la centrale, quali ad esempio i dati per l'accesso alla rete GPRS, l'indirizzo IP della centrale e la porta socket sulla quale inviare i dati di localizzazione. Queste impostazioni sono quindi necessarie per consentire al dispositivo di comunicare con la centrale.

\$0 - Aggiornamento parametri di sistema

Centro→PowerLac

```
$0:ID_SMS,  
    ID_Vehicle,  
    SetupNumber1, SetupNumber2, SetupNumber3,  
    AskNumber1,AskTypeSMS1,AskChannelType1,  
    AskNumber2,AskTypeSMS2,AskChannelType2,  
    AskNumber3,AskTypeSMS3,AskChannelType3,  
    AskNumber4,AskTypeSMS4,AskChannelType4,  
    AskNumber5,AskTypeSMS5,AskChannelType5,  
    TimeOutLogOff,GPRS_APNUN,GPRS_APNPW,GPRS_APNSERV,  
    TCP_IP_Center_Addr,TCP_IP_Center_Port,  
    TempCoeff_m, TempCoeff_q, GPRS_ConnectionTimeOut  
    *[CKS<CR><LF>]
```

dove:

ID_SMS:	Identificativo dell'SMS
ID_Vehicle:	Identificativo del mezzo
SetupNumber1-3:	3 Numeri di telefono abilitati all'aggiornamento parametri.
AskNumber1-5:	5 numeri di telefono abilitati all'interrogazione.
AskTypeSMS1-5:	Tipo di comando di protocollo da inviare.
AskChannelType1-5:	Tipo di canale utilizzato per l'invio:
	0: SMS
	1: GPRS
TimeOutLogOff:	Dopo che sono trascorsi questi secondi dallo spengimento del mezzo,

rilevabile dallo stato della chiave, viene //eseguito un log off.

- GPRS_APNUM: Access point - User Name
- GPRS_APNPW: Access point - Password
- GPRS_APNSERV: Access point - Server
- TCP_IP_Center_Addr: Indirizzo TCP/IP del socket verso il server del centro.
- TCP_IP_Center_Port: Porta dell'indirizzo TCP/IP del socket verso il server del
- TempCoeff_m Coefficiente m della formula di conversione per la temperatura: $T=m*x+q$
- TempCoeff_q Coefficiente q della formula di conversione per la temperatura: $T=m*x+q$
- GPRS_ConnectionTimeout: Time out di trasmissione in modalità GPRS.
- [CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Il campo AskTypeSMS indica il tipo di comando di protocollo che il PowerLac deve usare per l'invio della localizzazione. I valori ammissibili sono i seguenti:

- 2: invio della localizzazione testuale
- 254: invio della localizzazione con Ack (vedere comando di protocollo \$12)
- 255: invio della localizzazione senza Ack (vedere il comando di protocollo \$11)

Se il campo AskChannelType è 0 (SMS) è necessario che AskTypeSMS sia 2, ovvero invio della localizzazione senza Ack.

Attenzione: Se un campo è vuoto → il relativo parametro sul PowerLac viene lasciato invariato.
Se un campo è composto da uno spazio → il relativo parametro sul PowerLac viene resettato.

PowerLac→Centro

\$0:ID_SMS,ID_MEZZO,[OK|ERR]*[CKS<CR><LF>]

dove:

ID_SMS Identificativo dell'SMS ricevuto

ID_MEZZO Identificativo del mezzo

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

\$PCSIM – Abilita o disabilita pin nella SIM

Questo comando ha lo scopo di abilitare o disabilitare la richiesta di pin nella SIM.

È necessario eseguire prima un comando \$PLSIM con risposta OK, perché \$PCSIM parte dal presupposto che il login alla SIM sia già riuscito.

Centro→Dispositivo

\$PCSIM:ID_VEHICLE,PIN,STATE*CKS[<CR><LF>]

Dove:

- **ID_Vehicle** è l'identificativo del mezzo espresso in formato ASCII e con terminatore 0 finale. Il campo può essere vuoto solo se il messaggio arriva tramite SMS o seriale.
- **PIN** è il pin attuale della SIM espresso in formato ASCII (ad esempio il numero 0 va fornito come valore esadecimale 0x30 e non come 0x00). Il campo **non può essere vuoto**.
- **State** è lo stato desiderato espresso in formato ASCII. Il valore "0" indica che la richiesta pin va disattivata, mentre "1" chiede l'attivazione. Altri valori sono considerati un errore e respinti.

Dispositivo→Centro

\$PCSIM:ID_VEHICLE,<OK|ERR>*CKS[<CR><LF>]

dove:

- **ID_Vehicle** è l'identificativo del mezzo espresso in formato ASCII e con terminatore 0 finale.
- <OK|ERR> indica l'esito del comando. Una risposta OK indica che la SIM ha accettato il comando, mentre ERR indica pin attuale rifiutato, mancata risposta ed in generale qualsiasi altra condizione anomala.

\$PLSIM – Esegue il login in una SIM con pin

Questo comando tenta di eseguire il login in una SIM con il pin abilitato.

Centro→Dispositivo

\$PCSIM:ID_VEHICLE,PIN *CKS[<CR><LF>]

Dove:

- **ID_Vehicle** è l'identificativo del mezzo espresso in formato ASCII e con terminatore 0 finale. Il campo può essere vuoto solo se il messaggio arriva tramite SMS o seriale.

- **PIN** è il pin attuale della SIM espresso in formato ASCII (ad esempio il numero 0 va fornito come valore esadecimale 0x30 e non come 0x00). Il campo **non può essere vuoto**.
- **State** è lo stato desiderato espresso in formato ASCII. Il valore "0" indica che la richiesta pin va disattivata, mentre "1" chiede l'attivazione. Altri valori sono considerati un errore e respinti.

Dispositivo→Centro

\$PCSIM:ID_VEHICLE,<OK|ERR|NOPIN>*CKS[<CR><LF>]

dove:

- **ID_Vehicle** è l'identificativo del mezzo espresso in formato ASCII e con terminatore 0 finale.
- <OK|ERR|NOPIN> indica l'esito del comando. OK indica che il login è riuscito, ERR indica una generica condizione di errore, mentre NOPIN segnala che la SIM non richiede login.

4. Localizzazione del dispositivo

La comunicazione della posizione e dello stato del dispositivo avviene in modo spontaneo o su richiesta del centro. In tal caso il centro interroga il mezzo effettuando una chiamata al numero del dispositivo che, identificando il chiamante, interrompe la comunicazione e si collega via GPRS. Il PowerLac comunica il proprio stato mediante una struttura in formato esadecimale stampabile. La frequenza e la logica di invio dello stato del dispositivo è configurabile da protocollo.

\$10 - Aggiornamento parametri di LOG

Centro→PowerLac

\$10:ID_SMS,
DeltaT_LOG,DeltaS_LOG,
DeltaT_SEND1,DeltaS_SEND1,SendOff_DateTime1,NumSend1,TypeSMS1,
DeltaT_SEND2,DeltaS_SEND2,SendOff_DateTime2,NumSend2,TypeSMS2,
DeltaT_SEND3,DeltaS_SEND3,SendOff_DateTime3,NumSend3,TypeSMS3,
EnablePointGrpIN,EnablePointGrpOUT*
[CKS<CR><LF>]

dove:

DeltaT_LOG:	Intervallo temporale ogni quanto loggo (0=>No Log)
DeltaS_LOG:	Intervallo spaziale ogni quanto loggo (0=>No Log)
DeltaT_SEND1-3:	Intervallo temporale ogni quanto invio lo stato per l'utente 1-3
DeltaS_SEND1-3:	Intervallo spaziale ogni quanto invio lo stato per l'utente 1-3
SendOff_DateTime1-3:	- Data di scadenza per l'invio spontaneo all'utente 1-3 - Formato:dd/mm/yy hh:mm (totale:14 caratteri)
NumSend1-3:	Numero di telefono a cui inviare l'SMS per l'utente 1-3
TypeSMS1-3:	Tipo di SMS da inviare per l'utente 1-3
EnablePointGrpIN	Long nel quale vengono specificati quali gruppi sono controllati per generare l'evento di entrata in un Punto Notevole.
EnablePointGrpOUT	Long nel quale vengono specificati quali gruppi sono controllati per generare l'evento di uscita da un PuntoNotevole.
[CKS<CR><LF>]:	FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Attenzione: Se un campo è vuoto → il relativo parametro sul PowerLac viene lasciato invariato.
Se un campo è composto da uno spazio → il relativo parametro sul PowerLac viene resettato.

PowerLac→Centro

\$10:ID_SMS,ID_MEZZO,[OK|ERR]*[CKS<CR><LF>]

dove:

ID_SMS Identificativo dell'SMS ricevuto

ID_MEZZO Identificativo del Veicolo

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

5. Invio dello stato del dispositivo

\$11 - struttura dati 'struct LOG Struct' senza ACK

In base alla strategia impostata con il comando

PowerLac → Centro

\$11:ID_Vehicle,StrHex *[CKS<CR><LF>]

dove:

ID_Vehicle: Identificativo del mezzo

StrHex: Dati esadecimali stampabili della struttura in '**struct LOG_Struct**' (**vedere**

AppendiceA) .

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

\$12 - LOG Struct con ACK

PowerLac → Centro

\$12:IdMsg, ID_Vehicle,StrHex *[CKS<CR><LF>]

dove:

IdMsg: Identificativo del messaggio

ID_Vehicle: Identificativo del mezzo

StrHex: Dati esadecimali stampabili della struttura in '**struct LOG_Struct**' (**vedere**

AppendiceA) .

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Centro → PowerLac

\$12:ACK,IdMsg *[CKS<CR><LF>]

dove:

IdMsg: Identificativo del messaggio arrivato dal PowerLac

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Nella struttura LOG_Struct è presente la stringa TAG, che è usata per notificare loggare logici che avvengono a bordo.

\$18 - LOG Struct con ACK e stato di dispositivi slave

In alcune circostanze un dispositivo può essere associato a un dispositivo slave che acquisisce i dati degli ingressi (un caso tipico è l'installazione di un dispositivo nella motrice di un camion e di un dispositivo slave in un rimorchio). In questo caso lo stato del dispositivo sarà l'unione (fatta dal dispositivo master) dei dati dei due dispositivi. I dati inviati in centrale conterranno anche l'identificativo del dispositivo slave nel campo id slave. Nel caso in cui il dispositivo slave non sia presente il campo IDSlave deve essere omesso e quindi la struttura deve essere trattata come se fosse inviata mediante il comando \$12.

PowerLac → Centro

\$18:IdMsg, ID_Vehicle, ID_Slave, StrHex *[CKS<CR><LF>]

dove:

IdMsg:	Identificativo del messaggio
ID_Vehicle:	Identificativo del mezzo
ID_Slave:	Identificativo del dispositivo slave
StrHex:	Dati esadecimali stampabili della struttura in ' struct LOG_Struct ' (vedere

AppendiceA) .

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Centro → PowerLac

\$18:ACK,IdMsg *[CKS<CR><LF>]

dove:

IdMsg:	Identificativo del messaggio arrivato dal PowerLac
[CKS<CR><LF>]:	FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Nella struttura LOG_Struct è presente la stringa TAG, che è usata per notificare loggare logici che avvengono a bordo.

Valori validi per il campo TAG

Attualmente gli stati gestiti del campo TAG sono i seguenti:

TAG	Significato
AllarmIn:NumInput	Ingresso NumInput attivo
AllarmOut:NumInput	Ingresso NumInput non attivo
C:CodCliente IdCarico	Carico merce IdCarico per il cliente CodCliente
EndServ:User DataInizioServizio SecondiGuida MetriPercorsi	Fine servizio dell'utente user. Vengono ritornati anche i dati sulla DataInizioServizio ,SecondiGuida dell'autista e MetriPercorsi.
EndTravel:CodCliente DataInizioGita MetriPercorsi ClientsAttuali	Fine gita per il cliente CodCliente. DataInizioGita indica la data di inizio, mentre metri Percorsi indica i metri percorsi nella gita.
Fuel:Val Unità	Indica il rifornimento da parte del mezzo. Val: valore(-1 indica il pieno di carb.); Unità: <E L> (euro litri)
IServ:User	Fine servizio invalido da parte dell'autista user. Questo accade quando l'autista user non ha chiuso il servizio e un altro autista si logga a bordo.
LogIn:user	È avvenuto il login dell'utente user
LogInE:user passw	Errore nel login dell'utente <user> con password <passw>
LogInNotKey:User	Login senza contatto chiave
LogOut:User	È avvenuto il logout dell'utente user
MaxVel:Velocità	Il mezzo ha superato il limite di velocità raggiungendo la velocità indicata.
NewFirmware	il mezzo ha ricevuto il nuovo firmware e stà per effettuare l'aggiornamento
NewFirmware	Il dispositivo ha scaricato il nuovo firmware ed è pronto a riavviarsi per effettuare l'aggiornamento.
OutputChanged:NumOutput NewValue	L'uscita NumOutput è cambiata e ha assunto il nuovo valore New Value
Point:PNId In/Out	Il mezzo è entrato/uscito da un punto notevole. Il punto è indicato da PNid , il campo In/Out è I se il mezzo entra in nel punto,O se esce dal mezzo.
S:CodCliente IdCarico	Scarico merce IdCarico per il cliente CodCliente
SetIdPin:x	Il controllo dello user name e della password è attivo o inattivo
StartA	Avvio dell'applicativo
StartServ:User	Inizio servizio dell'utente user

StartTravel:CodCliente ClientiAttuali	Inizio gita per il cliente CodCliente, ClientiAttuali indica la lista dei clienti per i quali si sta facendo servizio
Theft:StatoAntifurto SecondiAttivo	Lo stato dell'allarme antifurto è cambiato
TRange:TempMin TempMax TypeTemp	La soglia di temperature è stata cambiata
TowExt:trailerID,state	Utilizzata nelle configurazioni tipo "trailer" dove un Powerlac (su motrice) comunica con una Tracebox (su rimorchio) tramite seriale. Il tag comunica alla centrale l'aggancio o sgancio del dispositivo <trailerID> inviando rispettivamente 0 oppure 1 nel valore <state>.

NOTA: le date sono in formato dd/mm/yyyy hh:mm:ss.

I clienti per i quali il mezzo sta lavorando sono separati da spazio.

6. Gestione I/O

È possibile definire per ogni ingresso del dispositivo delle soglie di attivazione. Quando il dispositivo legge un valore fuori da tali soglie, l'ingresso è considerato attivo. Per ogni uscita è possibile definire la logica di attivazione.

\$20 - Aggiornamento parametri ingressi

Centro → PowerLac

\$20:ID_SMS,NumIngresso,Enable,
GroupEnableIn,GroupEnableOut,
Min,Max,Hysteresis,MinAllarmTime,
NumTel1,TypeSMS1,
NumTel 2,TypeSMS2,
NumTel 3,TypeSMS3*[CKS<CR><LF>]

dove:

NumIngresso: Numero dell'allarme per il quale si stanno settando i parametri

Enable: 1: Allarme in funzione; 0:Allarme disattivato.

GroupEnableIn: Gruppi di punti notevoli nei quali l'allarme è abilitato

GroupEnableOut:	Gruppi di punti notevoli fuori dei quali l'allarme è abilitato
Min:	Estremo inferiore del range nel quale non siamo in allarme
Max:	Estremo superiore del range nel quale non siamo in allarme
Hysteresis:	Isteresi
MinAlarmTime:	Numero di secondi che devono trascorrere prima che un allarme scatti
NumTel 1-3:	Numero 1-3 di telefono da avvisare in caso di allarme
TypeSMS1-3:	Tipo di SMS da inviare
[CKS<CR><LF>]:	FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

PowerLac → Centro

\$20:ID_SMS,ID_MEZZO,ID_ALLARM,[OK|ERR]*[CKS<CR><LF>]

dove:

ID_SMS	Identificativo dell'SMS ricevuto
ID_MEZZO	Identificativo del Veicolo
ID_ALLARM	Identificativo dell'allarme impostato
[CKS<CR><LF>]:	FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

\$30 - Aggiornamento parametri uscite (messaggio non attivo)

Centro → PowerLac

\$30:ID_SMS,NumOutput,Input,MaxVel,LogOff,ActiveMode,ActiveTime*[CKS<CR><LF>]

dove:

NumOutput:	Numero dell'uscita presa in esame.
Input:	Word da 16 bit in formato decimale (es:"24536"). (Vedi AppendiceA "Descrizione Struttura Uscite")
MaxVel:	Velocità massima dopo la quale si attiva l'uscita.
LogOff:	1: Attiva l'uscita quando si è in stato di LogOff.
ActiveMode:	1: Attivo basso. 0: Attivo alto.

ActiveTime:	Tempo di durata attivo (anche se gli ingressi decadono).Se zero, attivo direttamente in funzione degli ingressi.
Force:	(Default 0) Indica se lo stato dell'uscita debba essere forzato da centrale al valore ForceStatus. Nel caso in cui Force sia uguale a 1, l'uscita sarà impostata al valore ForceStatus a prescindere dallo stato degli ingressi e delle altre grandezze che determinano lo stato dell'uscita.
ForceStatus:	Valore da applicare all'uscita nel caso in cui Force sia diverso da 0.
[CKS<CR><LF>]:	FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

PowerLac → Centro

\$30:ID_SMS,ID_MEZZO,ID_USCITA,[OK|ERR]*[CKS<CR><LF>]

dove:

ID_SMS	Identificativo dell'SMS ricevuto
ID_MEZZO	Identificativo del Veicolo
ID_USCITA	identificativo dell'uscita impostata
[CKS<CR><LF>]:	FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

7. Configurazione parametri di sistema

\$DC - Correzione differenziale

Comando:

\$DC:STR*[CKS<CR><LF>]

dove:

STR è la stringa di correzione secondo il protocollo RTCM SC-104 versione 2.1 (tipo 1, 2, 9)

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

\$SET – impostazione parametri odometro e girobussola

\$SET:OP,GYDX,GYSN*[CKS<CR><LF>]

dove:

OP è il numero di impulsi per km

GYDX è il fattore di scala per la destra uV/gradi/sec

GYSN è il fattore di scala per la sinistra uV/gradi/sec

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Risposta:

\$SET: ID_MEZZO , [OK|ERR] * [CKS<CR><LF>]

Dove:

ID_MEZZO identificatore del veicolo

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

\$GET - Lettura parametri odometro e girobussola

Comando:

\$GET*CKS<CR><LF>

risposta:

\$GET: ID_MEZZO , [OK|ERR] , OP , GYDX , GYSN * [CKS<CR><LF>]

dove:

ID_MEZZO identificatore del veicolo

OP è il numero di impulsi per km

GYDX è il fattore di scala per la destra uV/gradi/sec

GYSN è il fattore di scala per la sinistra uV/gradi/sec

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

\$VER - Lettura versione firmware:

Comando:

\$VER* [CKS<CR><LF>]

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

risposta:

\$VER: ID_MEZZO, [OK|ERR], strVer* [CKS<CR><LF>]

dove:

ID_MEZZO identificatore del veicolo

strVer: stringa alfanumerica che descrive la versione presente sul Lac.

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

\$GETPARAM - richiesta parametri

Comando:

\$GETPARAM* [CKS<CR><LF>]

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Risposta:

Il PowerLac in risposta al comando invia una localizzazione seguita da SYSTEM PARAMETERS:

[Parametri di sistema]. I Parametri terminano con [END.]

\$SETMAXVEL - Impostazione velocità massima

Comando:

\$SETMAXVEL:MaxVel* [CKS<CR><LF>]

dove:

MaxVel: Massima Velocità dopo la quale viene 'loggato'

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Risposta:

`$SETMAXVEL: ID_MEZZO, [OK|ERR] * [CKS<CR><LF>]`

dove:

ID_MEZZO identificatore del veicolo

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

\$SEMT - Impostazione Metri Percorsi totali

Comando:

`$SETMT: TotMt* [CKS<CR><LF>]`

dove:

TotMt: Nuovo valore del contatore metri.

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Risposta:

`$SETMT: ID_MEZZO, [OK|ERR] * [CKS<CR><LF>]`

dove:

ID_MEZZO identificatore del veicolo

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

\$SETREFRSEC - Impostazione secondi funzionamento frigo

Comando:

`$SETREFRSEC: TotSEC* [CKS<CR><LF>]`

dove:

TotSEC: Nuovo valore del contatore di secondi di funzionamento frigo.
[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Risposta:

`$SETREFRSEC: ID_MEZZO , [OK|ERR] * [CKS<CR><LF>]`

dove:

ID_MEZZO identificatore del veicolo

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

8. Gestione login autisti

\$ADDUSER – aggiunta utente e password

Comando:

`$ADDUSER: Users , Password* [CKS<CR><LF>]`

dove:

Users: Stringa: utente

Password: Stringa: password

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Attenzione: per ovvi motivi, Users e Password non possono contenere i caratteri ‘,’ e ‘*’.

Risposta:

`$ADDUSER: ID_MEZZO , [OK|ERR|FULL] , USERS* [CKS<CR><LF>]`

dove:

ID_MEZZO identificatore del veicolo

FULL ram piena utente non aggiunto

USERS identificativo dell'utente aggiornato

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

\$DELUSER – rimozione utente

Comando:

```
$DELUSER:Users* [CKS<CR><LF>]
```

dove:

Users: Stringa: utente

Se questo campo ha il valore di: 'ALL USERS', tutti gli utenti verranno cancellati.

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Attenzione: per ovvi motivi,Users non può contenere i caratteri ',' e '*'.0

Risposta:

```
$DELUSER: ID_MEZZO, [OK|ERR|NOT FOUND] * [CKS<CR><LF>]
```

dove:

ID_MEZZO identificatore del veicolo

NOT FOUND utente non trovato

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

\$SETIDPIN - Abilitazione login autista

Comando:

```
$SETIDPIN:Val* [CKS<CR><LF>]
```

dove:

Val: 0: Disabilita id e pin autista.

1: Abilita id e pin autista.

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Risposta:

```
$SETIDPIN: ID_MEZZO, [OK|ERR] * [CKS<CR><LF>]
```


dove:

ID_MEZZO identificatore del veicolo

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

9. Gestione punti notevoli

\$ADDPOINT - Aggiungi punto notevole

Comando:

\$ADDPOINT: Lat , Long , RayMt , Group , ID , Ang , AngTolerance , Note* [CKS<CR><LF>]

dove:

Lat: Latitudine del punto (es. 43.7647667)

Long: Longitudine del punto (ed. 11.1629629)

RayMt: Raggio in metri tra 0 e 65535

Group: Valore decimale 8bit (0 .. 255) che definisce l'appartenenza al gruppo 0-7

Id: Id del punto

Ang: Angolo di provenienza

AngTolerance: Tolleranza sull'angolo di provenienza

Note: Nota o descrizione del punto notevole

VelLimit: Limite di velocità

FullSignaling: Tipologia di segnalazione internamente al punto notevole:

0: Solo in avvicinamento al PN

1: Sia in avvicinamento che in allontanamento

A: Campo da definire

B: Campo da definire

C: Campo da definire

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Risposta:

`$ADDPOINT: ID_MEZZO, [OK|ERR|FULL] * [CKS<CR><LF>]`

dove:

ID_MEZZO identificatore del veicolo

FULL RAM piena punto non aggiunto

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

\$DELPOINTS - Rimuovi punti notevoli

Comando:

`$DELPOINTS* [CKS<CR><LF>]`

dove:

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Risposta:

`$DELPOINTS: ID_MEZZO, [OK|ERR] * [CKS<CR><LF>]`

dove:

ID_MEZZO identificatore del veicolo

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

\$ADDAP – Aggiungi aeroporto

Questo comando è stato scritto per il firmware personalizzato GE, ma potrebbe essere facilmente aggiunto in quello generale. ADDAP è stato pensato per offrire un'alternativa molto più semplice al comando ADDPOINT, riducendo il numero di parametri a quelli strettamente indispensabili. Inoltre, il firmware prevede una funzione speciale di ricerca binaria che consente di identificare il punto in modo rapido anche in presenza di migliaia di aeroporti caricati in memoria.

Comando:

`$ADDAP:Lat,Long,RayMt,TextID * [CKS<CR><LF>]`

dove:

Lat: Latitudine del punto (es. 43.7647667) , formato virgola mobile
Long: Longitudine del punto (ed. 11.1629629) , formato virgola mobile
RayMt: Raggio in metri tra 0 e 65535
TextID Identificativo internazionale dell'aeroporto (tre lettere senza terminatore)
Spare: valore al momento non utilizzato, formato virgola mobile

Risposta:

`$ADDAP:<ID_MEZZO> , [OK|ERR] * [CKS<CR><LF>]`

dove:

ID_MEZZO identificatore del veicolo

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

\$CLEARAP – Cancella tabella aeroporti

Questo comando è stato scritto per il firmware personalizzato GE, ma potrebbe essere facilmente aggiunto in quello generale. CLEARAP cancella la tabella aeroporti eventualmente creata tramite comandi ADDAP.

Non è prevista la cancellazione di un singolo aeroporto, perché la tabella deve essere strettamente ordinata ed al momento non è previsto un algoritmo di riordinamento.

Comando:

`$CLEARAP* [CKS<CR><LF>]`

Risposta:

`$ADDAP:<ID_MEZZO> ,OK* [CKS<CR><LF>]`

dove:

ID_MEZZO identificatore del veicolo

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

10. Invio messaggi di testo a bordo

\$MSG – Invio messaggio a bordo

Comando:

`$MSG:IdMsg,TipoMsg,Messaggio* [CKS<CR><LF>]`

dove:

IdMsg: Id del messaggio (x Espansioni future)

TipoMsg: Tipo messaggio (x Espansioni future)

Messaggio: Messaggio da inviare al mezzo

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Risposta:

`$MSG:ID_MEZZO, ID_Msg, [OK|ERR] * [CKS<CR><LF>]`

dove:

ID_MEZZO identificatore del veicolo

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

11. Gestione soglie di temperatura

\$SETTHRESH - Definizione soglia temperatura

Comando:

`$SETTHRESH:NumSoglia,Descrizione,Tmin,Tmax,Hysteresis, MinAllarmTime* [CKS<CR><LF>]`

dove:

NumSoglia: Numero della soglia che si vuole impostare (tra 0 e 6)

Descrizione: Descrizione testuale della soglia (es: "LATTICINI")

Tmin: Temperatura minima (in °C) sotto la quale scatta l'allarme

Tmax: Temperatura massima (in °C) sopra la quale scatta l'allarme

Hysteresis: Isteresi in °C

MinAllarmTime: Tempo minimo per cambiare stato

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Risposta:

```
$SETHRESH: ID_MEZZO, NumSoglia, [OK|ERR] * [CKS<CR><LF>]
```

dove:

ID_MEZZO identificatore del veicolo

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

\$SELTHRESH - Selezione soglia temperatura

Comando:

```
$SELTHRESH: NumSoglia * [CKS<CR><LF>]
```

dove:

NumSoglia: Numero della soglia che si vuole selezionare (tra 0 e 6)

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Risposta:

```
$SELTHRESH: ID_MEZZO, NumSoglia, [OK|ERR] * [CKS<CR><LF>]
```

dove:

ID_MEZZO identificatore del veicolo

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

12. Gestione allarmi antifurto

E' possibile attivare per il dispositivo l'allarme antifurto. Quando il mezzo è in allarme antifurto invia sms e effettua una serie di squilli in fonia ai numeri previsti.

\$ENANTITHEFT - Abilitazione antifurto

Comando:

```
$ENANTITHEFT: ENABLE, ONOFF, ENSPECIAL, ENDRIVERDOOR, ENMOVEALL* [CKS<CR><LF>]
```

dove:

Enable: [0|1] abilita o disabilita l'allarme antifurto a bordo del mezzo

Onoff [0] spegne l'allarme; [1] accende l'allarme; [2] lascia l'allarme nello stato in cui si

trova

EnsPECIAL [0|1] abilita o disabilita l'attivazione dell' antifurto per carichi speciali

EnDriverDoor [0|1] abilita o disabilita l'attivazione dell' antifurto per apertura porta autista

EnMoveAll [0|1] abilita o disabilita l'attivazione dell' antifurto per mezzo in movimento

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Risposta:

```
$ENANTITHEFT: ID_MEZZO, [OK|ERR] * [CKS<CR><LF>]
```

dove:

ID_MEZZO identificatore del veicolo

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

\$ANTITHEFTTEL - Impostazione numeri telefonici antifurto

Comando:

```
$ANTITHEFTTEL: NUM1, NUM2, NUM3, NUM4, NUM5, NUM6, NUM7, NUM8* [CKS<CR><LF>]
```

dove:

NUM1-8: Numeri di telefono per l'antifurto (1-4: SMS – 5-8: Numeri ai quali fare lo squillo)

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Risposta:

`$ANTITHEFTTEL: ID_MEZZO, [OK|ERR] * [CKS<CR><LF>]`

dove:

ID_MEZZO identificatore del veicolo

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

13. Comunicazioni inter-dispositivo

Alcune configurazioni prevedono la possibilità di collegare tramite seriale un Powerlac ed una Tracebox, formando un sistema master-slave, in cui lo slave trasferisce il proprio stato al master invece di inviarlo tramite log. Un caso tipico si ha quando si collega un rimorchio ad una motrice.

\$PLGT – comunicazione seriale master-slave (solo su Powerlac1)

Questo messaggio al momento è attivo solo su Powerlac1. Però è disponibile nella versione binaria del protocollo per Powerlac2.

Comando (Powerlac a Tracebox):

`$PLGT:<dati master>*<CKS><CR><LF>`

Risposta (Tracebox a Powerlac):

`$PLGT:<dati slave>*<CKS><CR><LF>`

In entrambi i casi, i dati sono espressi come coppie di caratteri ASCII che esprimono il valore esadecimale di un byte.

I simboli <CR> e <LF> sono obbligatori perché questo tipo di comunicazione può avvenire solo via seriale.

14. Aggiornamento firmware

Attenzione: Durante l'aggiornamento del firmware verrà utilizzata la RAM adibita per il LOG dati, con l'inevitabile perdita degli stessi. Ricordarsi, se necessario, di scaricarli prima di iniziare l'aggiornamento.

Modo 1:

STEPS:

\$INITDOWNLOAD – cancellazione buffer RAM

1. Inizializzare il buffer in RAM con 0xFF per lo scarico dell'immagine del firmware con il comando:

```
$INITDOWNLOAD* [CKS<CR><LF>]
```

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Risposta:

```
$INITDOWNLOAD: Esito* [CKS<CR><LF>]
```

dove:

Esito: OK → Inizializzazione avvenuta con successo
 WP → Jumper write protect non presente, quindi non è possibile la
 programmazione.

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

\$BLOCKDOWNLOAD – scrittura di un blocco di RAM

2. Scaricare nel buffer in RAM l'immagine dati del firmware tramite il seguente comando:

```
$BLOCKDOWNLOAD: AddrBlock, SizeBlock* [CKS<CR><LF>]
```

dove:

AddrBlock: Indirizzo del blocco dati da inviare.

SizeBlock: Dimensione del blocco da scaricare

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Successivamente verranno trasmessi i dati binari del blocco, terminato da checksum (relativo al blocco inviato).

Risposta:

\$BLOCKDOWNLOAD:ESITO* [CKS<CR><LF>]

dove:

ESITO=OK Blocco ricevuto con successo.

ESITO=ERROR Blocco non ricevuto correttamente.

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

\$IMAGECHECKSUM– verifica checksum immagine in RAM

3. Dopo che sono stati trasferiti tutti i dati, fare verifica del checksum dell'immagine dati nel buffer RAM:

\$IMAGECHECKSUM* [CKS<CR><LF>]

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Risposta:

\$ IMAGECHECKSUM:CheckSumFlash, CheckSumRAM* [CKS<CR><LF>]

dove:

CheckSumFlash : Checksum dell'immagine dati presente in FLASH.

CheckSumRAM : Checksum dell'immagine dati in RAM pronta per essere trasferita nella Flash

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

\$WRITEFLASH – copia del RAM buffer nella flash

4. Per procedere con l'aggiornamento della FLASH con l'immagine dati preparata in RAM, utilizzare il seguente comando:

\$WRITEFLASH* [CKS<CR><LF>]

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Risposta:l

\$WRITEFLASH : SUCCESSFUL* [CKS<CR><LF>]

dove:

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Prima della risposta il microprocessore esegue i seguenti passi:

- a. Esegue l'aggiornamento della FLASH
- b. Inizializza la RAM per il LOG con 0x00
- c. Termina, se presente, la comunicazione GSM
- d. Reset del micro per eseguire il firmware nuovo

Modo2:

\$HTTPUPDATE – aggiornamento firmware da URL

\$HTTPUPDATE : URL , HOST , FORCE* [CKS<CR><LF>]

dove:

URL: IndirizzoURL del file immagine del firmware. Deve avere estensione .cde

HOST: Indirizzo IP dell'URL(normalmente non usato).

FORCE: Se 1 indica al PowerLac di effettuare l'aggiornamento a prescindere da eventuali gite, servizi aperti.

FILEChecksum:Checksum calcolato sul file del firmware.

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

Risposta:

\$HTTPUPDATE: ID_MEZZO, [OK|ERR] * [CKS<CR><LF>]

dove:

ID_MEZZO identificatore del veicolo

[CKS<CR><LF>]: FACOLTATIVO. Questa parte è presente solo se il protocollo transita attraverso una comunicazione seriale.

\$RESET – reset di sistema

Esegue il soft reset del dispositivo ed opzionalmente cancella una o più tabelle in flash esterna.

Il comando è esclusivamente disponibile nel protocollo ASCII.

Comando:

\$RESET: Tipo* [CKS<CR><LF>]

<Tipo> indica quali tabelle si richiede eventualmente di cancellare, tramite una struttura a bit: si possono quindi cancellare più tabelle mettendo in OR i bit desiderati.

Se <Tipo> vale 0, il comando esegue semplicemente un soft reset.

Bit Tabella cancellata

0	Parametri di sistema e relativo backup
1	Accelerometro
2	Parametri di connessione e relativo backup
3	Punti notevoli e gruppi di punti (entrambe le tabelle)
4	Configurazione generale e relativo backup

5	Parametri di log (nota: non si cancellano i log, per i quali si rimanda al comando \$LOGSET)
6	Riservato per cancellazione tabelle utente
7	Riservato per cancellazione tabelle utente

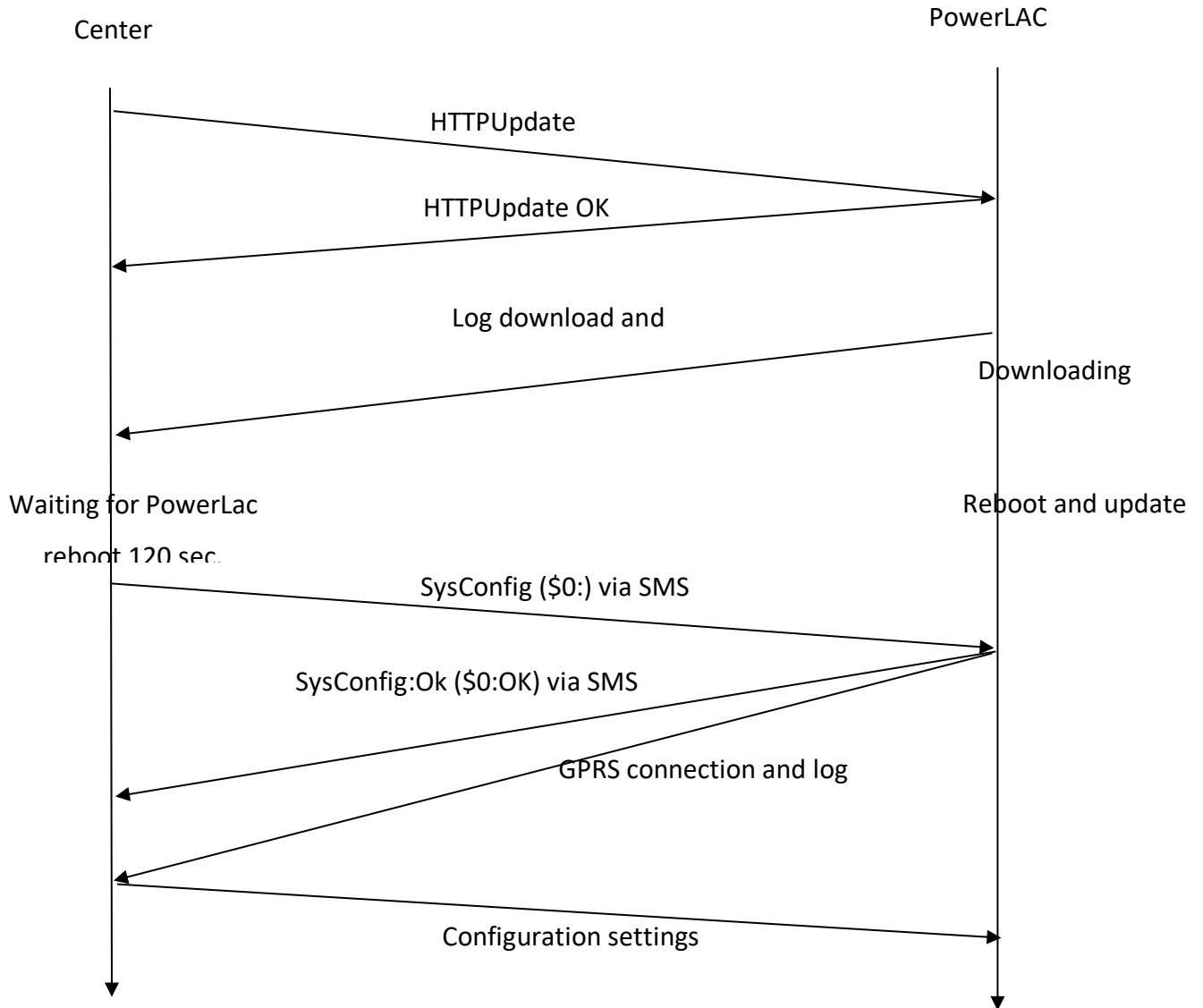
NOTA:

I bit 6 e 7 consentono di cancellare eventuali tabelle utente.

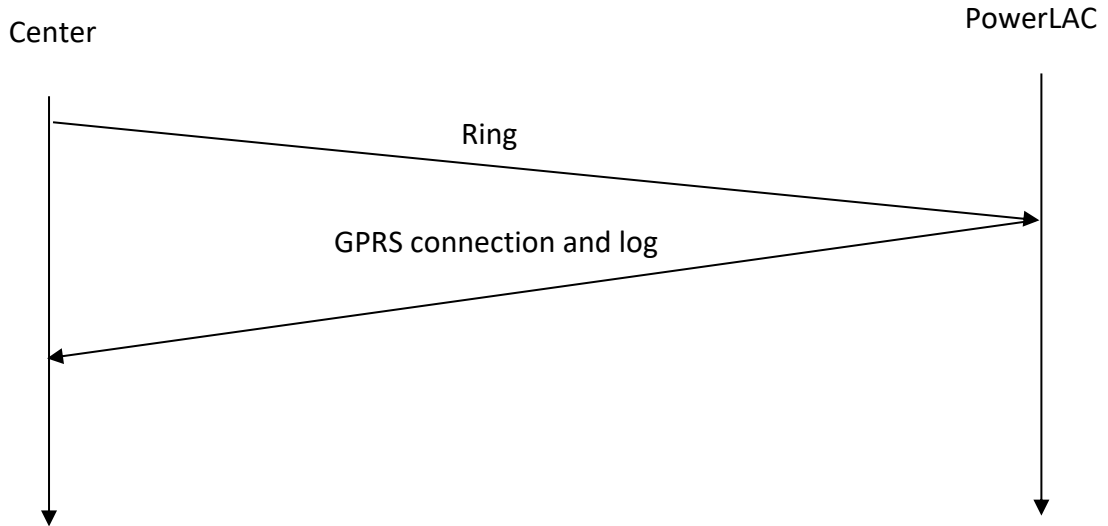
A questo scopo il firmware definisce un puntatore a funzione, normalmente NULL, che il firmware personalizzato può ridefinire in modo da intercettare richieste di cancellazioni ai parametri non standard.

15. Appendice A – Flussi logici

Aggiornamento firmware



Richiesta connessione del mezzo



Quando il mezzo riceve la chiamata aggancia la comunicazione e si riconnette subito al centro.