SaetCom
v0.25

Autore: Giancarlo Capella
Data: 10/08/2012

Cronologia

Versione	*Data*	*Varianti*
...	...	(Omissis)
0.4	23/03/2005	Specifica del processo di autenticazione
0.5	30/03/2005	Aggiunti eventi 82 e 83 (allineamento orologio)
0.6	26/01/2006	Aggiornamento alla reale implementazione definitiva. Non viene più usato l'MD5 per l'autenticazione ma solo AES per la crittografia.
0.7	29/05/2006	Aggiunto evento 84 (stato connessione)
0.8	23/01/2007	Inserito comando Y con codice 52 e spostata tutta la numerazione dei comandi successivi
0.9	07/02/2007	Aggiunti comando 69 ed evento 101 di allineamento centrale. Rinumerato vecchio evento 84 (stato connessione) come 102.
0.10	20/02/2007	Ridenominazione DelphiCom in SaetCom. Revisione generale del protocollo e aggiunta del dettaglio di implementazione.
0.11	26/02/2007	Corretto il formato dell'evento 80, valori analogici, per il quale i valori sono espressi su word e non su byte.
0.12	09/07/2008	Aggiunti i comandi e gli eventi per la gestione degli impianti EiB.
0.13	03/10/2008	Aggiunto comando 70 per invio file lara.gz
0.14	01/10/2009	Aggiunto evento 10 di allineamento per centrale Konnex EiB
0.15	27/11/2009	Definito il protocollo per dati ModBus (tipo 3) e scheduler (tipo 4)
0.16	02/12/2009	Aggiornati gli eventi ed i comandi ModBus (tipo 3), con l'aggiunta di dettaglio implementativo.
0.17	07/12/2009	Aggiornati gli eventi ed i comandi Scheduler (tipo 4), con l'aggiunta di dettaglio implementativo.
0.18	20/02/2010	Modificata la descrizione dei comandi ed eventi Scheduler secondo la nuova struttura XML di base. Aggiunta l'appendice B con lo schema XML della configurazione trasferita.
0.19	25/02/2010	Aggiornata la comunicazione scheduler, con l'invio di xml anche parziali al posto delle conferme generiche. Aggiornato lo schema XML dell'appendice B.
0.20	15/04/2010	Aggiunto un byte all'evento di conferma comando per lo scheduler.
0.21	10/01/2011	Aggiunto evento Delphi 103 per overflow dello storico
0.22	30/07/2012	Aggiunti comandi 71 e 72 per invio giorni festivi in blocco, 73 e 74 per le relative richieste, e gli eventi 104 e 105 per la lettura della configurazione.
0.23	30/07/2012	Creata classe 5 “serrature”
0.24	08/08/2012	Comando Modbus di impostazione base tempi e relativo evento di conferma
0.25	10/08/2012	Aggiunta l'indicazione dell'indirizzo dispositivo per il comando e l'evento ModBus di base tempi (dimenticanza)

Introduzione

Lo scopo di questo documento è descrivere il protocollo di comunicazione tra GEMSS e la centrale Delphi.

Architettura

La comunicazione avviene su evento in modo da non caricare la rete di comunicazione. Utilizza il protocollo UDP/IP per l'invio dei datagram ed ogni comando/evento viene confermato dal ricevente tramite apposito messaggio di acknowledge.

In caso di mancata ricezione dell'Ack per il periodo di timeout, il trasmittente ripete il messaggio inviato precedentemente, impostando l'apposito bit di stato “messaggio ritrasmesso”. Il periodo di timeout è fissato ad 1 secondo e il numero di tentativi di invio del messaggio è pari a 3. Se nessuno dei tentativi va a buon fine, il destinatario del messaggio è considerato non più raggiungibile (caduta rete).

Sia GEMSS che la centrale Delphi si comportano da client o da server a seconda del tipo di messaggio scambiato. Per i comandi il client è GEMSS ed il server è la centrale Delphi, per gli eventi il client è la centrale Delphi ed il server è GEMSS.

Formato datagram

Il pacchetto scambiato tra GEMSS e Delphi ha il seguente formato:

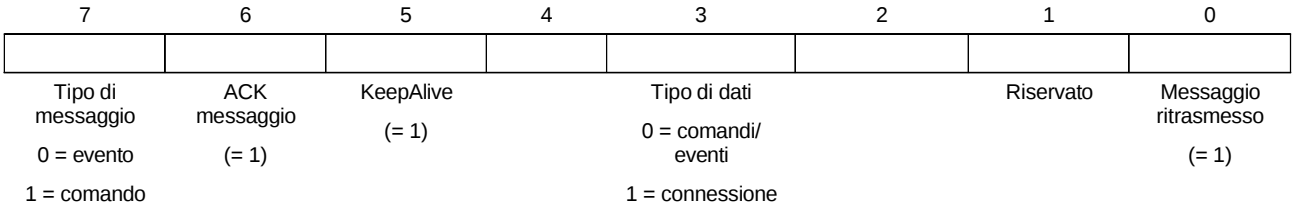
MsgID	Stato	Impianto	Nodo	DateTime	Lungh.	Estensioni	[Dati]

MsgID

Il MsgID è un contatore di pacchetto su 2 byte incrementato ad ogni pacchetto inviato. Un messaggio ripetuto mantiene la numerazione del pacchetto originario.

Stato

Il byte di stato ha il seguente formato:



La ritrasmissione di un messaggio forza il bit 0 a 1 e mantiene il numero di messaggio (MsgID).

L'Ack ad un messaggio avviene replicando il MsgID del comando e forzando il bit 6 a 1. Il tipo di messaggio è lo stesso del messaggio di cui si sta facendo l'acknowlegde. Non si fa ritrasmissione dell'Ack, se non arriva a destinazione, la destinazione stessa si preoccupa di ritrasmettere il messaggio originario.

GEMSS può verificare lo stato di attività della centrale Delphi inviando un pacchetto vuoto (solo l'header senza il campo dati) e con il bit KeepAlive a 1. La centrale Delphi risponde a questo messaggio con un Acknowledge.

Impianto

Il numero di impianto identifica l'impianto di destinazione nel caso di un comando, o l'impianto mittente nel caso di un evento, senza legare quindi l'identità di un impianto ad un indirizzo IP specifico. La dimensione del campo è di 2 byte.

Nodo

Il numero di nodo per ora è da considerarsi non utilizzato anche se può assumere valori diversi. Ignorare.

DateTime

Ogni messaggio viene marchiato dall'istante di insorgenza dell'evento che trasporta o dell'invio del messaggio stesso. Il campo è un intero su 4 byte, indicante il numero di secondi a partire dal 1/1/1990.

Lunghezza

Attraverso questo campo viene indicata la lunghezza del messaggio completo, header incluso. L'utilità di questa indicazione è principalmente legata alla trasmissione crittografata dei messaggi, poiché la lunghezza dei messaggi crittografati non è la stessa dei messaggi in chiaro.

Estensioni

Questo campo è riservato per eventuali future estensioni. Attualmente è impostato 0 zero.

Comando/Evento

Per i messaggi di comando o di evento (non Ack), il campo dati contiene la codifica degli stessi secondo il formato descritto nei capitoli seguenti. Per i messaggi di Ack o di KeepAlive il campo è assente.

Connessione

I messaggi di tipo connessione (bit Tipo Dati = 1) sono utilizzati per abilitare il dialogo tra una centrale ed un supervisore. Il campo dati per i messaggi in entrambe le direzioni, da supervisore a centrale e viceversa, contiene le informazioni necessarie al riconoscimento reciproco degli interlocutori. In assenza di riconoscimento, la centrale non accetta comandi e non invia eventi sulla rete.

Formato Comandi

Il comando è composto dal tipo comando, dal codice comando e dagli eventuali dati associati. Il tipo rappresenta la centrale alla quale è diretto il comando; attualmente sono definiti i valori 0 per centrale Tebe e 1 per centrale Delphi.

Il dato espresso con *b* è su 1 byte, il dato espresso da *ww* è su 2 byte in formato little endian (formato Intel).

Tipo	***Codice***	***Dati***	***Intervallo***	***Descrizione***	***Comando SaetNet***
Comandi centrale DELPHI					
1	1	ww	0 - 8192	Fuori servizio attuatore	A
1	2	ww	0 - 8192	In servizio attuatore	B
1	3	ww	0 - 8192	Fuori servizio sensore	C
1	4	ww	0 - 8192	In servizio sensore	D
1	5			Accettazione degli attuatori temporizzati	G
1	6			Accettazione di tutti gli attuatori	H
1	7	b	0 - 254	Attivazione zona	I
1	8	b	0 - 254	Disattivazione zona	J
1	9	ww	0 - 1023	On memoria (telecomando)	K
1	10			Sistema in prova	L
1	11	ww	0 - 8192	Invio rumore sul sensore indicato (se sistema in prova)	M
1	12			Invio rumore a tutti i sensori (se sistema in prova)	N
1	13			Sistema non in prova	O
1	14	wwb	0 - 8192 0 - 3	Nuovo valore di durata sensore (vecchio tipo)	P
1	15	wwb	0 - 8192 0 - 7	Nuovo valore sensibilità sensore (vecchio tipo)	Q
1	16	bb	0 - 2 1 - 32	Programmazione giorno festivo: 0=feriale; 1=semifestivo; 2=festivo	R
1	17	bbb	1 - 31 1 - 12 0 - 99	Programmazione data. (anno a partire dal 2000)	U

Tipo	Codice	Dati	Intervallo	Descrizione	Comando SaetNet
1	18			Visualizza ora corrente	V
1	19	bbb	0 - 23 0 - 59 0 - 59	Programmazione ora	W
1	20	ww[8b]	0 - 8192 0 - 255	Configurazione zona/sensori	Z
1	21			Genera evento risposta SCS	>
1	22			Refresh stato sensori	/35
1	23			Refresh stato attuatori	/36
1	24			Refresh stato tarature	/37
1	25			Invio memoria MU (128 eventi 200 terminati da 201)	/380
1	26	ww	0 - 8192	Accettazione allarme	/381
1	27	ww	0 - 8192	Accettazione manomissione	/382
1	28	ww	0 - 8192	Accettazione guasto	/383
1	29			Azzera memoria evento sensore	/389
1	30	bww	0 - 9 0 - 65535	Invio segreto	/52
1	31	bww	0 - 9 0 - 65535	Invio evento	/53
1	32	[32b]	0 - 2	Lista 32 giorni festivi programmabili a partire da domani.	/54
1	33	bbbb	0 - 255 0 - 1 0 - 23 0 - 59	Variazione fascia oraria	/7
1	34	ww	0 - 512	Lettura valori analogici sensori della periferica	/aR
1	35	wwb[5b]	0 - 512 0 - 3, 8 - 15 0 - 255	Invio soglie analogiche sensori alla periferica. Il primo b è il canale, seguono i 5 valori di soglia.	/aW
1	36	ww	0 - 512	Richiesta primo badge libero sulla periferica	/B1
1	37	wwwwb	0 - 512 0 - 1999 1, 2, 4, 128	Gestione badge su periferica: 1-abilitazione, 2-disabilitazione, 4-cancellazione, 128-invio di tutti i badge	/B2
1	38	wwww	0 - 512 0 - 1999	Acquisizione badge	/B3
1	39	ww[?]	0 - 512	Caricamento badge	/B4
1	40			Lista sensori fuori servizio	a
1	41			Lista attuatori fuori servizio	b
1	42			Lista stato zone	c
1	43			Lista stato sensori	d
1	44			Lista stato attuatori	e
1	45			Lista parametri di taratura	g
1	46			Lista giorni festivi	k
1	47			Lista periferiche	n

Tipo	Codice	Dati	Intervallo	Descrizione	Comando SaetNet
				(periferiche previste – periferiche presenti)	
1	48			Lista controllo memoria	t
1	49			Lista fasce orarie giornaliere	u
1	50			Lista chiavi	v
1	51			Lista associazione zona/sensore	w
1	52	bb	0 – 2 0 - 255	Attivazione speciale zona	Y
1	53	b	0 - 9	Chiamata manuale ronda	/550
1	54	b	0 - 9	Partenza ronda	/551
1	55	b	0 - 9	Annullamento ronda	/552
1	56	bbbww	0 - 9 0 - 31 0 - 99 0 - 65535	Impostazione tappa percorso ronda	/560
1	57	bb[4bww]	0 - 9 0 - 31 0 - 255 0 - 65535	Impostazione ronda zone da disattivare	/561
1	58	bb[4b]	0 - 9 0 - 31 0 - 255	Impostazione ronda zone da attivare	/562
1	59	bbbbb	0 - 2 0 - 31 0 - 23 0 - 59 0 - 9	Impostazione ronda orario di partenza	/563
1	60	b	0 - 8	Interroga lo stato del modem GSM generando eventi del tipo 253-0. Il parametro indica il tipo di messaggio richiesto: **event_index**=0 Identificazione dell'operatore di network. I dati sono letti dal modulo GSM una sola volta all'accensione del sistema. **event_index**=1 Numero del centro di servizi SMS. Il dato è letto dal modulo GSM una sola volta all'accensione del sistema. **event_index**=2 Genera eventi ciascuno contenete una cella valida di rubrica. **event_index**=3 Ultime 10 chiamate (memorizzate dal driver). **event_index**=4 Ultime 10 chiamate fallite (memorizzate dal driver). **event_index**=5 Ultime 10 ricezioni (memorizzate dal driver). **event_index**=6 Stato SIM **event_index**=7 Valori minimo e massimo del segnale radio rilevato e la data e l'ora del primo rilevamento. La richiesta genera anche gli eventi relativi all'event_index=8. **event_index**=8 Valori attuale e medio del segnale radio rilevato e per il secondo anche	/AGS

Tipo	Codice	Dati	Intervallo	Descrizione	Comando SaetNet
				l'intervallo di tempo sul quale e' eseguita la misura, espressa in giorni e ore, minuti e secondi. **event_index**=9 Azzeramento statistiche segnale radio.	
1	61	b	4 - 7	Interroga lo stato del modem analogico generando eventi del tipo 253-1. Il parametro indica il tipo di messaggio richiesto: **event_index**=4 Genera eventi ciascuno contenete una cella valida di rubrica. **event_index**=5 Ultime 10 chiamate (memorizzate dal driver). **event_index**=6 Ultime 10 chiamate fallite (memorizzate dal driver). **event_index**=7 Ultime 10 ricezioni (memorizzate dal driver).	/AMS
1	62			Restart centrale Delphi	/MR
1	63	wwww[b...]	0 - 65535 (num. pkt) 0 - 1400* (dim. pkt) 0 – 255 * su ISI: 0 – 192	Invio file libuser.so	/MU
1	64			Invio file saet.conf	/MC
1	65			Invio file saet.nv	/MN
1	66			Invio file consumer.conf	/MP
1	67			Invio file strings.conf	/MS
1	68			Invio file sprint.xml	/MX
1	69			Richiesta allineamento centrale	x
1	70	wwww[b...]	come comandi 63-68	Invio file lara.gz	/ML
1	71	32*[bbb]		Invio 32 festività fisse (giorno/mese/tipo) Le festività non impostate hanno i campi a FFh.	/542, /543
1	72	32*[bbbbb]		Invio 32 festività variabili (giorno/mese/anno/tipo) Anno su 2 byte per comprendere anche il secolo. Le festività non impostate hanno i campi a FFh.	/544, /545
1	73			Richiesta festività fisse	f
1	74			Richiesta festività variabili	z
-	-	-	-	Usato internamente da GEMSS	/X
Comandi centrale TEBE					
0	1				/La
0	2				/Lb
0	3				/Lc
0	4				/Ld
0	5				/Le
0	6				/Lf
0	7				/Lg
0	8				/Lh
0	9				/Li

Tipo	*Codice*	*Dati*	*Intervallo*	*Descrizione*	*Comando SaetNet*
0	10				/Lj
0	11				/Lk
0	12				/Ll
Comandi impianto EiB (*)					
2	1			Invio configurazione	
2	2			Richiesta di allineamento	
2	3			Lettura indirizzo di gruppo	
2	4			Comando indirizzo di gruppo (6bit)	
2	5			Comando indirizzo di gruppo (1byte)	
2	6			Comando indirizzo di gruppo (2byte)	
2	7			Comando indirizzo di gruppo (4byte)	
Comandi impianto ModBus (*)					
3	1	b		Richiesta di allineamento	
3	2	b b aa b [b...]		Scrittura dati (indirizzo dispositivo, function, indirizzo dati, tipo dati, dati)	
3	3	b wwww wwww		Impostazione base tempi (nuova base tempi [s], durata della base tempi [s])	
Comandi scheduler					
4	1	ww		Lettura regola	
4	2	[seq][dati]		Scrittura regola	
4	3	ww		Cancellazione regola	
4	4	wwb		Forzatura regola (on/off)	
4	5			Lettura anagrafica completa	
4	6	[seq][dati]		Scrittura anagrafica completa	
Comandi serrature					
5	1	b		Crea utente	
5	2	b		Elimina utente	
5	3	b		Abilita utente	
5	4	b		Disabilita utente	
5	5	b		Invalida utente	
5	6			Blocco serratura	
5	7			Sblocco serratura	
5	8			Apertura da remoto con conferma locale	
5	9			Lettura fasce orarie	
5	10			Programmazione fasce orarie	
5	11			Lettura festività fisse	
5	12			Programmazione festività fisse	
5	13			Lettura festività variabili	
5	14			Programmazione festività variabili	

(*) I comandi SaetCom per l'impianto EiB non ha un corrispettivo SaetNet. La comunicazione è gestita

unicamente dalle centrali ISI con gestione diretta in questo formato.

Dettaglio comandi ModBus

Il comando di richiesta di allineamento prevede il solo identificativo del dispositivo. Questa operazione restituisce le 4 tabelle ModBus come descritto nel paragrafo eventi.

La scrittura di dati può essere fatta solo in modo singolo, quindi specificando la function 05h per la scrittura nella tabella coils (quella che viene letta tramite function 01h) e indicando il tipo dati 0 (bit), oppure la function 06h per la scrittura nella tabella holding register (letta tramite la function 03h), indicando il tipo dati 2 (word, 2 byte) o 3 (dword, 4 byte). Il tipo 3 è valido solo per specifici impianti.

Es:

Richiesta allineamento impianto 3: 03 01 03

Scrittura di bit nella tabella coils all'indirizzo 123, dispositivo 2: 03 02 02 05 7b 00 00 01

Scrittura di registro nella tabella holding register all'indirizzo 1000, dispositivo 4: 03 02 04 06 e8 03 02 01 00

Dettaglio comandi Scheduler

Lo scheduler è implementato solo nella centrale ISI e risiede nel task di supervisione GEMSS/udp.

I comandi permettono di gestire le regole attive all'interno dello scheduler, numerate a partire da 1 ma non necessariamente consecutive. Ogni regola è descritta da una schema XML e di base è composta da una data/ora di inizio ed una data/ora di fine, entrambe associate al proprio codice di comando che viene attivato all'ingresso o all'uscita della fascia stessa.

I comandi 1 e 3 prevedono come parametro l'indice della regola di interesse. Il comandi 2 e 6 permettono di trasferire l'anagrafica in modo parziale (comando 2) o completo (comando 6). La differenza tra i due comandi è che il primo aggiunge o sostituisce le regola trasmesse, il comando 6 svuota l'anagrafica corrente e la sostituisce completamente con quella ricevuta.

Infine è possibile forzare esplicitamente lo stato di una regola tramite il comando 4. La forzatura ON (parametro = 1) invia il comando previsto all'attivazione della regola, la forzatura OFF (parametro = 2) invia il comando previsto alla disattivazione della regola. In entrambi i casi il comando viene comunque inviato solo se c'è variazione di stato rispetto allo stato corrente. Inviando il comando con parametro 0 si procede allo sblocco della regola che quindi si attiva/disattiva secondo le normali attività. Se il codice di forzatura è diverso da 0,1,2, il comando non ha effetto ma l'evento di conferma restituisce lo stato di forzatura corrente.

I comandi 2 e 6 restituiscono l'xml di ciò che è stato effettivamente processato, con gli ID delle regole che non ne possedevano uno (regole nuove).

I comandi 3 e 4 restituiscono un evento generico di conferma.

Dettaglio comandi Serrature

I comandi definiti sono ricavati a partire dall'implementazione ISI – Kaithron. Eventuali future integrazioni diverse da Kaithron potrebbero implicare la definizione di ulteriori comandi.

Formato Eventi

L'evento ha il seguente formato:

247	[tipo]	[codice]	[dati]

Il *tipo* indica che tipo di centrale ha emesso l'evento. Attualmente sono definiti i valori 0 per una centrale Tebe e 1 per una centrale Delphi. Il *codice* quindi ha una numerazione che dipende dal *tipo*.

Centrale Tebe

Il formato degli eventi per la centrale tipo Tebe (tipo = 0) è descritto nel documento specifico.

Centrale Delphi

Il formato degli eventi per la centrale tipo Delphi (tipo = 1) è il seguente. Il dato espresso con *b* è su 1 byte, il dato espresso da *ww* è su 2 byte in formato little endian (formato Intel).

Codice	***Dati***	***Intervallo***	***Descrizione***
0			Inizio test attivo

Codice	*Dati*	*Intervallo*	*Descrizione*
1			Fine test attivo
2	wwb		Allarme sensore **ww** Z=**b**
3	wwb		Fine allarme sensore **ww** Z=**b**
4	ww		Guasto periferica **ww**
5	ww		Fine guasto periferica **ww**
6	ww		Manomissione dispositivo **ww**
7	ww		Fine manomissione dispositivo **ww**
8	ww		Manomissione contenitore periferica **ww**
9	ww		Fine manomissione contenitore periferica **ww**
10	b		Attiva zona **b**
11	b		Disattiva zona **b**
12	b		Attivazione impedita zona **b**
13	ww		In servizio sensore **ww**
14	ww		Fuori servizio sensore **ww**
15	ww		In servizio attuatore **ww**
16	ww		Fuori servizio attuatore **ww**
17	b		Modulo Master **b**: ricevuto codice errato
18	bb		Rilevato Modulo Master **b**: tipo **b**
19	wwb		Periferica incongruente ind. **ww** tipo **b**
20	ww		Periferica manomessa ind. **ww**
21	ww		Ripristino periferica ind. **ww**
22	ww		Sospesa attivita linea **ww**
23	ww		Chiave falsa periferica **ww**
24	ww		Sentinella **ww** attivata
25	ww		Sentinella **ww** disattivata
26	ww		Sentinella **ww** disattivata per fuori tempo
27	b		No Tx modulo master n. **b**
28	b		Errore Rx modulo master n. **b**
29			Errore ricezione messaggio host
30	ww		Segnalazione evento n. **ww**
31	ww[8b]		Periferiche presenti da **ww > bbbbbbbb**
32	b[8b]		Stato zona **b> bbbbbbbb**
33	ww[8b]		Stato sensore **ww> bbbbbbbb**
34	ww[8b]		Stato attuatore **ww> bbbbbbbb**
35	ww		Guasto sensore **ww**
36	bb		Variazione ora **bb**
37	wwwwww		Codici di controllo: **ww ww ww**
38			Stato QRA I= **b** QRA II= **b**. Attivata ronda **b**
39	ww		Accettato allarme sensore **ww**
40	ww		Mancata punzonatura stazione **ww**

Codice	*Dati*	*Intervallo*	*Descrizione*
41	ww		On telecomando **ww**
42	wwbb		Parametri taratura periferica **ww** Sens. **b** Dur. **b**
43	b		Stato Prova = **b**
44			Test attivo gruppo
45			Risposta SCS
46	ww		Fine guasto sensore **ww**
47	wwb		Valore **ww** per segreto **b**
48	ww[8b]		Periferiche previste da **ww > bbbbbbbb**
49	wwb		Sensore in allarme **ww** Z=**b**
50	bbbb		Variata fascia oraria **b b** ore **bb**
51			Invio diretto dati tipo **b** a host computer
52			Salvataggio memoria
53			Fine invio dati
53			Cnf.ronda **b- b**,**b**,**b**,**b**,**b**,**b**,**b**,**b**,**b**,**b**,**b**,**b**,**b**,**b**,**b**
55			Ore ronda: **b**- **bb**,**bb**,**bb**,**bb**
56	b[16b]		Lista festivi **b- b**,**b**,**b**,**b**,**b**,**b**,**b**,**b**,**b**,**b**,**b**,**b**,**b**,**b**,**b**
57	b[4bb]		Lista fasce giornaliere **b – bb bb bb bb**
58	ww		ON Attuatore **ww**
59	wwww		Transitato identificativo **ww** lettore **ww**
60	wwww		Entrato identificativo **ww** lettore **ww**
61	wwww		Uscito identificativo **ww** lettore **ww**
62	b		Codice valido **b**
63	b		Chiave valida **b**
64	b		Operatore n. **b**
65	ww		Stazione n. **ww** punzonata

Codice	*Dati*	*Intervallo*	*Descrizione*
66	ww		Spegnimento n. **ww**
67	ww		Reset fumo n. **ww**
68			Livello abilitato **b**
69			Variato segreto
70	b		Avvenuta connessione modem tipo **b**
73	?		Accesso badge: periferica **b** id. **ww** gruppo **b bbbbbbbbbb**
74	wwbb		Evento sensore **ww b b**
75	ww[8b]		Stato MU **ww > bbbbbbbb**
76	ww[8b]		Associazione zone sensori **ww > bbbbbbbb**.
79	bb		Giorno **b** (offsett rispetto a oggi) variato a **b** (0=feriale, 1=semifestivo, 2=festivo).
80	ww[8ww]		Valori analogici **ww > ww ww ww ww ww ww ww ww**
81	?		Badge letto
82			Orologio disallineato
83			Orologio riallineato
84			Manca rete
85			Ripristino rete
86			Guasto batteria
87			Ripristino batteria
88			Inizio manutenzione
89			Fine manutenzione
90	b		Fine test negativo
91			Inizio straordinario
92			Fine straordinario
93			Inserimento anticipato

Codice	Dati	Intervallo	Descrizione
94	b		Attivata zona ritardata
95	b		Stato centrale[1]
96	wwb		On attuatore (0=normale, 1=lampeggiante)[2]
97	ww		Off attuatore[2]
98			Salvataggio database (variati parametri)[2]
99	ww		Fuori servizio sensore (sensore autoescluso)
100	ww		In servizio sensore (sensore autoescluso)
101	bww[8b]		Lista allineamento centrale[3]
102	b		Stato connessione (usato dall'ISI, ex 84)
103			Overflow storico
104	32*[bbb]		Giorni festivi fissi (giorno/mese/tipo)
105	32*[bbbbb]		Giorni festivi variabili (giorno/mese/anno/tipo)

Impianto EiB

Gli eventi definiti per l'impianto EiB (tipo = 2) sono i seguenti.

('aa': indirizzo individuale/gruppo, 'b': byte)

Codice	Dati	Intervallo	Descrizione
0	b	1=connesso 0=non connesso	Connessione ad impianto
1	aab	1=guasto 0=normale	Guasto elemento
2	aaaab		Comando indirizzo di gruppo (6bit)

1 Il byte di dato rappresenta una bitmap di stati specifici di centrale. In particolare sono definiti i seguenti bit:
  bit 0 – centrale in manutenzione
  bit 1 – manca rete
  bit 2 – batteria scollegata
  bit 3 – orologio disallineato
  bit 4 – guasto centrale (corrispondente al guasto della periferica 0 di centrale)
  bit 5 – manomissione centrale (corrispondente alla manomissione della periferica 0 di centrale)

2 Questi eventi sono normalmente filtrati e quindi non visibili. Affiché risultino visibili occorre abilitarli esplicitamente nella struttura PrintEventEx2.

3 Il primo byte dell'evento rappresenta l'oggetto in allineamento:
  0: fine allineamento
  1: sensori
  2: zone
  I byte di stato hanno una forma comune:
  bit 0 – allarme
  bit 1 – manomissione
  bit 2 – guasto
  bit 3 – allarme memorizzato
  bit 4 – manomissione memorizzata
  bit 5 – guasto memorizzato
  bit 6 – non usato
  bit 7 – fuori servizio (sensore), attivata (zona)

Codice	*Dati*	*Intervallo*	*Descrizione*
3	aaaab		Comando indirizzo di gruppo (1byte)
4	aaaabb		Comando indirizzo di gruppo (2byte)
5	aaaabbbb		Comando indirizzo di gruppo (4byte)
6	aaaab		Lettura indirizzo di gruppo (6bit)
7	aaaab		Lettura indirizzo di gruppo (1byte)
8	aaaabb		Lettura indirizzo di gruppo (2byte)
9	aaaabbbb		Lettura indirizzo di gruppo (4byte)
10	b	1 – inizio 0 – fine	Allineamento stato elementi

Impianto ModBus

Gli eventi definiti per l'impianto ModBus (tipo = 3) sono i seguenti.

Codice	*Dati*	*Intervallo*	*Descrizione*
0	bb	[0-255][0-1]	Stato comunicazione
1	(vedi sotto)		Variato valore
2			Lista allineamento
3	b wwww wwww		Variata base tempi, durata nuova impostazione

Lo stato comunicazione indica l'indirizzo del dispositivo ModBus a cui si riferisce e lo stato (0=caduta, 1=ripristino).

L'evento di variato valore ha il seguente formato:

1	ind	function	regaddrL	regaddrH	tipo	dati...

dove:

ind: indirizzo del dispositivo ModBus [0-255]
function: codice della function ModBus utilizzata per la lettura
regaddr: indirizzo del valore letto all'interno della tabella specificata da function
tipo: indica se si tratta di 1 bit (0), 1 byte (1), 2 byte/1 word (2), 2 word (3)
dati: 1, 2 o 4 byte per i dati da trasmettere.

I valori vengono sempre trasmessi uno per volta, nel caso di 1 bit viene comunque trasmesso un byte intero di valore 0x00 o 0x01.

Gli eventi di allineamento hanno il seguente formato:

2	ind	function	seqL	seqH	dati...

dove:

ind: indirizzo del dispositivo ModBus [0-255]
function: codice della function ModBus utilizzata per la lettura (01h coils, 02h input, 03h holding register, 04h input register)
seq: sequenza del pacchetto dati che andrà a formare il blocco dati completo (partendo da 0)
dati: blocco dati (max 64 byte) della tabella compressa con algoritmo Zlib. La quantità di dati si ricava in base alla dimensione del messaggio che lo trasporta, e l'assenza di dati indica il fine invio per la tabella.

La conferma della corretta interpretazione dei pacchetti di una tabella si può avere dalla dimensione dei dati decompressi. Per le function 01h e 02h si devono ottenere esattamente 8192 byte, per le function 03h e 04h si devono ottenere esattamente 131072 byte.

Gestione scheduler

Gli eventi definiti per lo scheduler (tipo = 4) sono i seguenti.

Codice	Dati	Intervallo	Descrizione
0	[b][id][status]		Conferma comando [b]
1	[seq][dati]		Lettura configurazione regola singola
2	[seq][dati]		Lettura anagrafica completa

L'evento codice 0 viene restituito per i comandi di cancellazione regola e forzatura regola. Nel primo caso lo status è sempre a 0, mentre in caso di forzatura viene restituito lo stato di forzatura impostato per la regola.

I due messaggi restituiscono un file compresso con algoritmo Zlib in modo analogo al sistema ModBus. Il trasferimento prevede un certo numero di messaggi numerati dal campo [seq] e contenente una parte del file che il ricevente deve ricostruire. Il contenuto del file è una struttura xml che descrive una o più regole definite. Benché il file trasferito abbia la stessa struttura in entrambi i casi, il codice evento specifica se si tratti di una regola singola o il trasferimento dell'anagrafica intera.

Gestione serrature

Gli eventi definiti per le serrature (tipo = 5) sono i seguenti.

Codice	Dati	Intervallo	Descrizione
0	b b		Variato stato utente
1	b		Utente invalidato
2	b	0=non autorizzato, 1-2=utente autorizzante	Autorizzazione locale alla programmazione
3	b	0=non autorizzato, 1-2=utente autorizzante	Autorizzazione locale all'apertura
4			Serratura bloccata
5			Serratura sbloccata
6	b b bb bb	giorno della settimana numero fascia nel giorno ore:minuti inizio ore:minuti fine	Fasce orarie
7	32*[bbb]	giorno/mese, tipo	Festività fisse
8	32*[bbbbb]	giorno/mese/anno, tipo	Festività variabili

Gli utenti sono identificati come 1-2 per gli utenti senior, 11-20 per gli utenti junior

Gli stati utente sono codificati come:

Codice	Descrizione
0	Cancellato
1	Abilitato
2	Disabilitato

Le fasce orarie sono identificate dal giorno della settimana di riferimento (0=domenica) e dall'indice di fascia del giorno. Nel caso Kaithron si hanno due fasce orarie per ogni giorno della settimana, ma altre applicazioni potrebbero richiederne di più.

L'anno nelle festività variabili è espresso su 2 byte poiché indica anche il secolo.

Evento stringa

L'evento stringa è codificato come...

Identificazione

La procedura di identificazione è necessaria affinché la centrale conosca l'indirizzo IP del supervisore. Questo permette di ottenere una certa flessibilità nella configurazione del sistema, mentre il numero di porta e l'indirizzo IP della centrale devono essere noti a priori.

I messaggi di identificazione sono caratterizzati dal bit di tipo dati impostato a 1 nel byte di stato. Il campo dati assume quindi una struttura specifica, formata da un byte di fase e da eventuali byte associati.

Il byte di fase può assumere i seguenti valori:

Fase	***Descrizione***	***Direzione***
0	Non valido	
1	Richiesta di identificazione	SV -> Centrale
2	Richiesta accettata	Centrale -> SV
3	Impostazione chiave	SV -> Centrale
4	Chiave impostata	Centrale -> SV

La procedura di identificazione avviene su iniziativa del supervisore che intende dialogare con la centrale, generando la richiesta con fase 1. Se la centrale è disponibile allo scambio di dati, ovvero non c'è un altro supervisore connesso, viene generata la risposta con fase 2. Nel caso la centrale abbia già un supervisore registrato ed attivo ed il dialogo sta avvenendo senza crittografia, la nuova richiesta di identificazione viene ignorata, a meno che non sia originata dal supervisore corrente.

Al termine della fase 2, i due sistemi possono regolarmente dialogare fra loro senza crittografia. L'attivazione della crittografia avviene procedendo con la fase 3, con la quale il supervisore invia 24 byte di chiave alla centrale. La chiave rimane valida fino a nuova impostazione o a decadimento per timeout della comunicazione in corso. L'accettazione della nuova chiave avviene con la generazione della risposta con fase 4.

Il timeout di decadimento della comunicazione in corso ha particolarmente significato nel caso di comunicazione senza crittografia. Tale valore deve essere impostato con un valore molto alto, dell'ordine di mezz'ora o anche di ore, nel caso si voglia garantire che sempre un solo supervisore possa dialogare con la centrale, senza generare varchi di possibilità per nuove connessioni in caso di caduta prolungata di rete. Durante questo timeout, la centrale non accetta nuove richieste di identificazione per prevenire attacchi dalla rete.

Nel caso in cui la crittografia è attiva, la centrale accetta qualunque richiesta di connessione in ogni istante, posto che la richiesta sia stata trasmessa cifrata con la chiave che si presume nota solo ai supervisori autorizzati. Per questo motivo il timeout può essere impostato a zero per indicare una validità della connessione a tempo indeterminato.

Il supervisore deve comunque considerare la ricezione di eventuali conferme di identificazione non originate dal supervisore stesso come tentativo di sovrapposizione fraudolenta di un altro supervisore con identico indirizzo IP.

Le fasi 3 e 4 sono facoltative solo all'avvio della centrale ovvero quando non sono mai state eseguite in precedenza. Se attraverso tali fasi viene programmata una chiave, da quell'istante in poi la centrale accetta esclusivamente messaggi crittografati con la chiave impostata, di qualunque tipo essi siano.

La variazione della chiave da parte del supervisore corrente può avvenire direttamente con le fasi 3 e 4, saltando le fasi 1 e 2 in questo caso opzionali.

Crittografia

Impostando una chiave di crittografia attraverso le fasi 3 e 4 di connessione, viene attivata la trasmissione cifrata di tutti i messaggi.

La cifratura dei messaggi avviene tramite algoritmo AES con chiave a 192bit e riguarda l'intero messaggio trasmesso, header compreso.

Come già indicato, la chiave impostata rimane valida per tutto il tempo di connessione. La chiave può essere modificata dal supervisore, ma in caso di timeout (prolungata assenza di attività tra centrale e supervisore) tale chiave viene azzerata per permettere il ristabilimento di una nuova connessione.

Per evitare lo scadere accidentale della chiave in caso di sistema a riposo, viene periodicamente trasmesso un messaggio di keepalive, con il duplice scopo di mantenere attiva la connessione corrente e di rilevare l'eventuale caduta della centrale stessa.

In caso di crittografia attiva, è inoltre possibile per qualunque supervisore connettersi alla centrale stessa senza dover attendere il timeout di connessione.

Questo perché si presume che se un supervisore è in possesso della chiave di comunicazione è già automaticamente autorizzato alla connessione. Il nuovo supervisore che vuole sostituirsi al precedente deve comunque effettuare le fasi 1 e 2 per dichiararsi alla centrale.

In un sistema di centralizzazione ridondato è quindi conveniente utilizzare la comunicazione cifrata in modo da poter scambiare velocemente le funzionalità tra i sistemi primari e secondari.

Attraverso lo studio di particolari procedure è anche possibile impostare a priori una chiave di crittografia sulla centrale così che il sistema possa partire fin da subito con l'utilizzo di messaggi cifrati senza far transitare mai la chiave sulla connessione di rete.

Appendice A

Vengono qui descritti i dettagli relativi all'apertura della connessione e le caratteristiche di sicurezza del protocollo SaetCom. Verranno indicati il supervisore come SV e la centrale come CA. Per pacchetto vuoto si intende un pacchetto formato dal solo header.

Apertura connessione

- SV invia a CA un pacchetto con i bit Command e Connect impostati, e nel campo dati indica nel primo byte il valore 01h seguito da 4 byte di valore casuale;
- CA risponde con Ack al messaggio ricevuto, restituendo nel campo dati gli stessi valori ricevuti nella richiesta che SV deve verificare;
- CA invia a SV un pacchetto con il solo bit Connect impostato, e nel campo dati indica nel primo byte il valore 02h seguito da 4 byte di valore casuale;
- SV risponde con Ack al messaggio ricevuto, restituendo nel campo dati gli stessi valori ricevuti nella richiesta che CA deve verificare;
- la connessione è a questo punto attiva e può partire la comunicazione in chiaro se non è già stata impostata una chiave, o cifrata se è presente una chiave di crittografia. I contatori per il campo MsgId nelle due direzioni vengono azzerati.

Il campo dati per i messaggi di Ack viene utilizzato esclusivamente nel dialogo per la connessione, poiché non viene eseguito il controllo sul campo MsgId ed occorre fornire un minimo di sicurezza a questa fase. In caso di comunicazione in chiaro, la robustezza fornita è facilmente scavalcabile da un interlocutore intelligente, ma non da messaggi replicati, mentre con crittografia attiva fornisce un meccanismo sufficiente a riconoscere eventuali pacchetti iniettati in rete.

Impostazione chiave di crittografia

- SV invia a CA un pacchetto con i bit Command e Connect impostati, e nel campo dati indica nel primo byte il valore 03h seguito da 24 byte di chiave;
- CA risponde con Ack al messaggio ricevuto;
- CA invia a SV un pacchetto con il solo bit Connect impostato, e nel campo dati indica nel primo byte il valore 04h seguito da 4 byte di valore casuale, come richiesta di conferma;
- SV risponde con Ack al messaggio ricevuto, restituendo nel campo dati gli stessi valori ricevuti nella richiesta che CA deve verificare;
- la chiave viene attivata e da questo punto in poi viene utilizzata per cifrare tutti i messaggi scambiati tra SV e CA;
- [preferibile] SV dovrebbe inviare immediatamente un KeepAlive a CA e verificare che risponda. In caso di mancata risposta, CA potrebbe aver perso l'Ack e non aver quindi attivato la nuova chiave. SV dovrebbe perciò riprendere la chiave precedente e procedere ad una nuova sessione di impostazione chiave.

La procedura per l'impostazione della chiave avviene nello stesso modo sia dopo una prima connessione, con comunicazione in chiaro, sia dopo aver impostato precedentemente una chiave. In questo caso la comunicazione avviene sotto cifratura della chiave precedente. Al termine della procedura la chiave viene sostituita con la nuova.

Verifiche di validità dei messaggi scambiati

Tutti i messaggi scambiati tra SV e CA devono sottostare ad un certo numero di regole che devono essere verificate prima di accettare come valido un messaggio ricevuto.

- il bit Command è sempre impostato per i messaggi da SV a CA;
- il bit Command non è mai impostato per i messaggi da CA a SV;
- il bit Resend non è mai impostato per i messaggi di Ack, a meno che non sia un Ack ad un messaggio di keepalive che richiede il riallineamento dei contatori; tale riallineamento deve essere eseguito immediatamente su iniziativa di SV attivando la procedura di connessione;
- tutti i bit di stato non codificati devono sempre risultare a 0;
- il campo Nodo non è attualmente utilizzato e non può assumere valori diversi da 0;

- il campo Estensione può assumere solo il valore 1, che indica l'attuale revisione del protocollo;
- se non sono impostati i bit Connect e KeepAlive, viene eseguito un controllo stretto sul campo MsgId. Tale valore deve essere sempre quello atteso, pari cioè al valore del contatore interno di messaggio atteso per i messaggi di comando od evento, e pari al valore del messaggio inviato precedentemente nel caso di messaggi di acknowledge;
- ad ogni messaggio ricevuto con il campo MsgId pari al valore atteso, viene incrementato il contatore atteso per il messaggio successivo. Sia SV che CA devono quindi preoccuparsi di incrementare a loro volta i contatori utilizzati per popolare il campo MsgId;
- in caso di bit Resend impostato, il controllo su MsgId deve avvenire in due fasi: se MsgId è allineato al contatore per il messaggio atteso significa che il messaggio originale è andato perso e deve quindi essere gestito come nuovo. Se invece MsgId è pari al contatore atteso decrementato di uno, è andato perso l'Ack di risposta: il messaggio deve essere ignorato ma deve essere reinviato l'Ack. In tutti gli altri casi, vale il controllo generale del campo MsgId e quindi il messaggio deve essere scartato;
- se il bit KeepAlive è impostato non viene eseguito il controllo stretto sul campo MsgId. Viene sempre inviato in risposta l'Ack, ma nel caso di contatori disallineati viene impostato anche il bit Resend. Quando SV riceve la combinazione di bit KeepAlive+Ack+Resend deve attivare la procedura di connessione per riallineare i contatori;
- se SV invia un comando senza ricevere l'Ack per un determinato numero di tentativi, potrebbe significare o che la CA non è più raggiungibile o che i contatori si sono disallineati per qualche motivo (es: reset della CA dopo l'ultimo KeepAlive). Prima di dichiarare non raggiungibile CA, SV deve verificare la comunicazione con un messaggio di KeepAlive. Se i contatori sono disallineati viene ricevuta la risposta di cui sopra ed a seguito del riallineamento può essere nuovamente inviato il comando accodato;
- se il bit Connect è impostato viene eseguito un controllo sulla dimensione del messaggio ricevuto, che a seconda delle fasi deve avere una dimensione precisa;
- per le proprie caratteristiche intrinseche, i pacchetti UDP trasmessi non possono essere spezzati da apparati di comunicazione intermedi. La dimensione di ogni pacchetto deve quindi essere pari a quella indicata nel messaggio stesso in caso di comunicazione in chiaro, o comunque non inferiore in caso di comunicazione cifrata. La dimensione minima del messaggio è pari a quella dell'header, non esistono messaggi di dimensione inferiore;
- se la chiave di crittografia è impostata, il pacchetto dati ricevuto deve necessariamente avere una dimensione pari ad un multiplo di 16.

In generale, se la chiave di crittografia è impostata le verifiche di cui sopra sono in grado di discriminare tra un messaggio valido e uno con contenuto casuale o generato con una chiave non valida (che può quindi essere considerato a tutti gli effetti un messaggio con contenuto casuale).

Appendice B

Lo schema XML per il trasferimento dell'anagrafica dello scheduler è il seguente:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Scheduler" nillable="true" type="Scheduler" />
  <xs:complexType name="Scheduler">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="SchedulerRules" type="ArrayOfSchedulerRule" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ArrayOfSchedulerRule">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="SchedulerRule" nillable="true"
type="SchedulerRule" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="SchedulerRule">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Description" type="xs:string" />
      <xs:element minOccurs="1" maxOccurs="1" name="StartDate" type="xs:dateTime" />
      <xs:element minOccurs="0" maxOccurs="1" name="EndDate" type="xs:dateTime" />
      <xs:element minOccurs="0" maxOccurs="1" name="DaysOfWeek" type="ArrayOfInt" />
      <xs:element minOccurs="0" maxOccurs="1" name="CommandStart" type="SchedulerCommand" />
      <xs:element minOccurs="0" maxOccurs="1" name="CommandEnd" type="SchedulerCommand" />
      <xs:element minOccurs="0" maxOccurs="1" name="RangeStart" type="RangeLimit" />
      <xs:element minOccurs="0" maxOccurs="1" name="RangeEnd" type="RangeLimit" />
      <xs:element minOccurs="0" maxOccurs="1" name="Exceptions" type="ArrayOfSchedulerRule" />
    </xs:sequence>
    <xs:attribute name="ID" type="xs:int" use="required" />
  </xs:complexType>
  <xs:complexType name="ArrayOfInt">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="int" type="xs:int" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="SchedulerCommand">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Command" type="xs:base64Binary" />
      <xs:element minOccurs="1" maxOccurs="1" name="Time" type="xs:dateTime" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="RangeLimit">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Day" type="xs:int" />
      <xs:element minOccurs="0" maxOccurs="1" name="Month" type="xs:int" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```