
Specifica protocollo data-logger

Sommario

1	Generalità.....	1
2	Modbus standard RTU.....	2
2.1	Lettura e scrittura parametri di configurazione.....	3
2.1.1	Read Holding Registers – FC 03 (0x03).....	3
2.1.2	Write Multiple Registers – FC 16 (0x10).....	4
2.2	Lettura valori istantanei.....	5
2.2.1	Read Input Registers - 04 (0x04).....	5
3	Specifica comandi Modbus.....	6
3.1	Modbus User-Defined – FC 65 (0x41).....	6
3.2	Richiesta dati correnti.....	7
3.3	Richiesta dati di archivio.....	8
3.4	Tracciato record binario.....	10
3.5	Sincronizzazione orologio.....	11
3.6	Disattivazione risparmio energetico.....	11
3.7	Lettura e scrittura parametro.....	12
3.8	Riavvio e reset data-logger.....	13
3.9	Richiesta dati istantanei.....	13
3.10	Lettura configurazione data-logger.....	14
4	Tracciato record Dati Istantanei.....	15
4.1	CORPO.....	15
4.2	TERMINATORE.....	18
4.3	Esempio di tracciato record Dati istantanei.....	18

1 Generalità

Il seguente documento riporta la specifica del protocollo di comunicazione per il data-logger.

2 Modbus standard RTU

Il data-logger implementa il protocollo standard *Modbus RTU* in risposta alle richieste in ingresso sulla comunicazione seriale (dispositivo seriale Modbus Slave).

Si utilizzano gli *Holding Registers* per leggere e scrivere i parametri di configurazione e gli *Input Registers* per leggere le locazioni di memoria interna. I valori sono espressi in codifica 32-bit *Floating point*.

Le tabelle che seguono riportano i codici delle funzioni supportate e la mappatura dei registri Modbus in relazione alle variabili di memoria.

<i>Modbus function</i>		03 (0x03) Read Holding Registers			PARAMETRI DI CONFIGURAZIONE
		16 (0x10) Write Multiple Registers			
<i>N. Registro</i>	<i>Accesso</i>	<i>Tipo</i>	<i>Codifica valore</i>	<i>Descrizione</i>	
0001	Read-Write	16-bit word HIGH	32-bit Floating point point (4 bytes)	Valore del 1° parametro	
0002	Read-Write	16-bit word LOW			
0003	Read-Write	16-bit word HIGH	32-bit Floating point point (4 bytes)	Valore del 2° parametro	
0004	Read-Write	16-bit word LOW			
.....	
.....			
0199	Read-Write	16-bit word HIGH	32-bit Floating point point (4 bytes)	Valore del 100° parametro	
0200	Read-Write	16-bit word LOW			

Modbus function		04 (0X04) Read Input Registers		LOCAZIONI DI MEMORIA	
N. Registro	Accesso	Tipo	Codifica valore	Descrizione	
00001	Read-Only	16-bit word HIGH	32-bit Floating point (4 bytes)	Valore della locazione M0001	
00002	Read-Only	16-bit word LOW			
00003	Read-Only	16-bit word HIGH	32-bit Floating point (4 bytes)	Valore della locazione M0002	
00004	Read-Only	16-bit word LOW			
.....	
.....			
17999	Read-Only	16-bit word HIGH	32-bit Floating point (4 bytes)	Valore della locazione M9000	
18000	Read-Only	16-bit word LOW			

2.1 Lettura e scrittura parametri di configurazione

I parametri di configurazione sono variabili numeriche impostabili dall'utente e usati dal data-logger durante il funzionamento (es.: intervalli di elaborazione e/o cadenze di trasmissione dei dati, soglie di superamento allarme, flag di abilitazione/disabilitazione delle funzioni, ecc.). I valori sono mantenuti dal data-logger in modo permanente, possono essere letti e modificati tramite codici di funzione standard.

2.1.1 Read Holding Registers – FC 03 (0x03)

Questo codice funzione viene utilizzato per leggere il contenuto di un blocco contiguo di *Holding Registers* da un dispositivo Modbus remoto. Sul data-logger questi registri corrispondono ai valori dei parametri di configurazione.

Il comando di richiesta, oltre a ID stazione, deve specificare l'indirizzo del registro iniziale e la quantità di registri da leggere. Poiché i parametri di configurazione sono codificati in valori floating-point 32-bit è necessario specificare due registri per ogni valore, quindi l'indirizzamento e le quantità devono essere multipli di due. I registri sono indirizzati a partire da zero, pertanto i registri numerati 0001 - 0200 sono indirizzati come 0x0000 - 0x00C7.

I dati nel messaggio di risposta sono impacchettati con due byte per ogni registro. In ogni registro il primo byte contiene il bit di ordine elevato e il secondo contiene i bit di ordine basso (registro a 16 bit).

Il valore del parametro si ottiene dalla codifica floating-point 32-bit di due registri adiacenti (4 bytes).

Richiesta lettura (*Request to read*):

Item	Size	Value
Station ID	1 Byte	0x00 to 0xF7, (0xFF = broadcast ID)
Function code	1 Byte	0x03
Starting address	2 Bytes	0x0000 to 0x00C7 (multiple of 2)
Quantity of registers (N)	2 Bytes	0x0002 to 0x007C (multiple of 2, max. 124 regs.)
CRC-16	2 Bytes	0x0000 to 0xFFFF

Risposta dati (*Valid response*):

Item	Size	Value
Station ID	1 Byte	0x00 to 0xF7, (0xFF = broadcast ID)
Function code	1 Byte	0x03
Bytes count	1 Bytes	0x0004 to 0x00F8 (N x 2)
Registers values	N x 2 Bytes	<data values>
CRC-16	2 Bytes	0x0000 to 0xFFFF

Risposta errore (*Error response*):

Item	Size	Value
Station ID	1 Byte	0x00 to 0xF7, (0xFF = broadcast ID)
Error code	1 Byte	0x83
Exception code	1 Bytes	0x01 or 0x02 or 0x03 or 0x04 (ref. to Modbus spec.)
CRC-16	2 Bytes	0x0000 to 0xFFFF

2.1.2 Write Multiple Registers – FC 16 (0x10)

Il codice funzione viene utilizzato per scrivere il contenuto di un blocco contiguo di *Holding Registers* in un dispositivo Modbus remoto. Questi registri corrispondono ai valori dei parametri di configurazione impostabili dall'utente.

Il comando di richiesta deve specificare l'indirizzo del registro iniziale e la quantità di registri da scrivere.

Poiché i parametri di configurazione sono codificati in valori *floating-point* 32-bit è necessario specificare due registri per ogni valore, quindi l'indirizzamento e le quantità devono essere multipli di due.

I dati nel messaggio di richiesta sono impacchettati con quattro byte adiacenti in codifica *floating-point* 32-bit per ogni valore di parametro.

La risposta corretta restituisce il codice funzione, l'indirizzo iniziale e la quantità di registri scritti.

Richiesta scrittura (*Request to write*):

Item	Size	Value
Station ID	1 Byte	0x00 to 0xF7, (0xFF = broadcast ID)
Function code	1 Byte	0x10
Starting address	2 Bytes	0x0000 to 0x00C7 (multiple of 2)
Quantity of registers (N)	2 Bytes	0x0002 to 0x007C (multiple of 2, max. 124 regs.)
Bytes count	1 Bytes	0x0004 to 0x00F8 (N x 2)
Registers values	N x 2 Bytes	<data values>
CRC-16	2 Bytes	0x0000 to 0xFFFF

Risposta corretta (*Normal response*):

Item	Size	Value
Station ID	1 Byte	0x00 to 0xF7, (0xFF = broadcast ID)
Function code	1 Byte	0x10
Starting address	2 Bytes	0x0000 to 0x00C7 (as requested)
Quantity of registers	2 Bytes	0x0002 to 0x007C (number of registers writted)
CRC-16	2 Bytes	0x0000 to 0xFFFF

Risposta errore (*Error response*):

Item	Size	Value
Station ID	1 Byte	0x00 to 0xF7, (0xFF = broadcast ID)
Error code	1 Byte	0x90
Exception code	1 Bytes	0x01 or 0x02 or 0x03 or 0x04 (ref. to Modbus spec.)
CRC-16	2 Bytes	0x0000 to 0xFFFF

2.2 Lettura valori istantanei

2.2.1 Read Input Registers - 04 (0x04)

Questo codice funzione viene usato per leggere il contenuto di un blocco contiguo di *Input Registers* da un dispositivo Modbus remoto. Il data-logger utilizza questi registri come valori delle locazioni interne di memoria, solitamente i dati istantanei acquisiti o in elaborazione.

Il comando di richiesta, oltre a ID stazione, deve specificare l'indirizzo del registro iniziale e la quantità di registri da leggere. Poiché i valori delle locazioni sono codificati in valori *floating-point* 32-bit è necessario specificare due registri per ogni valore, quindi l'indirizzamento e le quantità devono essere multipli di due. I registri sono indirizzati a partire da zero, pertanto i registri numerati 00001 - 18000 sono indirizzati come 0x0000 - 0x4650.

I dati nel messaggio di risposta sono impacchettati con due byte per ogni registro. In ogni registro il primo byte contiene il bit di ordine elevato e il secondo contiene i bit di ordine basso (registro a 16 bit).

Il valore finale della locazione si ottiene dalla codifica *floating-point* 32-bit di due registri adiacenti (4 bytes).

Richiesta lettura (*Request to read*):

Item	Size	Value
Station ID	1 Byte	0x00 to 0xF7, (0xFF = broadcast ID)
Function code	1 Byte	0x04
Starting address	2 Bytes	0x0000 to 0x00C7 (multiple of 2)
Quantity of registers (N)	2 Bytes	0x0002 to 0x007C (multiple of 2, max. 124 regs.)
CRC-16	2 Bytes	0x0000 to 0xFFFF

Risposta dati (*Valid response*):

Item	Size	Value
Station ID	1 Byte	0x00 to 0xF7, (0xFF = broadcast ID)
Function code	1 Byte	0x04
Bytes count	1 Bytes	0x0004 to 0x00F8 (N x 2)
Registers values	N x 2 Bytes	<data values>
CRC-16	2 Bytes	0x0000 to 0xFFFF

Risposta errore (*Error response*):

Item	Size	Value
Station ID	1 Byte	0x00 to 0xF7, (0xFF = broadcast ID)
Error code	1 Byte	0x84
Exception code	1 Bytes	0x01 or 0x02 or 0x03 or 0x04 (ref. to Modbus spec.)
CRC-16	2 Bytes	0x0000 to 0xFFFF

3 Specifica comandi Modbus

3.1 Modbus User-Defined – FC 65 (0x41)

In aggiunta alle funzioni standard il data-logger implementa una funzione *Modbus User-Defined* (codice 65) con la quale è possibile veicolare un set di comandi predefiniti (vedi pagine successive).

Viene di seguito descritta la specifica del protocollo:

Richiesta (*Request*):

<i>Item</i>	<i>Size</i>	<i>Value</i>
Station ID	1 Byte	0x00 to 0xF7, (0xFF = broadcast ID)
Function code	1 Byte	0x41
Bytes count	2 Bytes	0x0000 to 0xFFFF: Message Length (ML)
Message request	ML Bytes	< <i>user message</i> >
CRC-16	2 Bytes	0x0000 to 0xFFFF

Risposta dati (*Valid response*):

<i>Item</i>	<i>Size</i>	<i>Value</i>
Station ID	1 Byte	0x00 to 0xF7, (0xFF = broadcast ID)
Function code	1 Byte	0x41
Bytes count	2 Bytes	0x0000 to 0xFFFF: Data Length (DL)
Data response	DL Bytes	< <i>data response</i> >
CRC-16	2 Bytes	0x0000 to 0xFFFF

Risposta errore (*Error response*):

<i>Item</i>	<i>Size</i>	<i>Value</i>
Station ID	1 Byte	0x00 to 0xF7, (0xFF = broadcast ID)
Error code	1 Byte	0xC1
Exception code	1 Bytes	0x01 - MODBUS_ILLEGAL_FUNCTION Il codice funzione ricevuto nella richiesta non è consentito. 0x02 - MODBUS_ILLEGAL_DATA_ADDRESS Il messaggio ricevuto ha una lunghezza diversa dal numero di byte specificato nella richiesta.
CRC-16	2 Bytes	0x0000 to 0xFFFF

La sintassi dei messaggi da inviare è costituita da campi stringa separati da uno o più spazi e i messaggi stessi devono essere incapsulati nella funzione 65.

E' possibile che la centrale Polaris possa inviare più di un comando all'interno della stessa richiesta Modbus. I comandi concatenati possono essere due o più (di norma due). Nel caso di più comandi, la risposta inviata dal datalogger sarà relativa solo all'ultimo comando ricevuto.

Qualora i comandi non vengano inviati in modo corretto il data logger risponderà con **?<comando>** per comando non riconosciuto oppure **-1** per parametri non corretti.

Nota:

Il protocollo descritto non prevede una limitazione sulla lunghezza della risposta, tuttavia se il messaggio è veicolato in RF sarà limitato dal buffer radio. Riferirsi alle caratteristiche dell'apparato radio utilizzato per individuare la dimensione massima della memoria buffer di trasmissione (tipicamente: 1 KByte).

3.2 Richiesta dati correnti

Comando !LBR (*Last Binary Records*)

Il comando viene utilizzato per richiedere gli ultimi records dei dati registrati nel periodo di riferimento configurato (normalmente nell'intervallo di polling 30 minuti). Ad esempio, inviando una richiesta di dati !LBR nella fascia oraria 08:30÷08:59, la risposta conterrà i dati presenti dalle ore 08.01 alle 08.30 comprese. Successivamente, inviando la stessa richiesta dopo le 09:00, si riceveranno i records dalle ore 08.31 alle ore 09.00 comprese.

Il messaggio di richiesta deve specificare il numero di archivio da cui si vogliono richiedere i dati (*file*).

Sintassi:

Messaggio	!LBR file [yyyy mm dd hh nn ss]
Risposta	<binary_records>

Il formato dei dati ricevuti in risposta (<binary_records>) è descritto al par. 3.4 – *Tracciato binario*.

Di seguito un esempio di chiamata e risposta per il comando !LBR in formato Hexadecimal:

Comando: C6 41 00 06 21 4C 42 52 20 36 F4 77

Risposta: C6 41 00 7F 15 0B 0B 0A 05 00 27 17
29 00 00 00 00 15 0B 0B 0A 0A 00 26
B5 22 51 EC 40 08 27 17 29 00 00 00
00 15 0B 0B 0A 0F 00 26 AD 21 CC CD
41 B8 26 B4 21 33 33 41 53 27 17 29
00 00 00 00 15 0B 0B 0A 14 00 26 B5
22 1E B8 40 05 27 17 29 00 00 00 00
15 0B 0B 0A 19 00 27 17 29 00 00 00
00 15 0B 0B 0A 1E 00 26 AD 21 00 00
41 B8 26 B4 21 33 33 41 53 26 B5 22
33 33 40 03 27 17 21 00 00 00 00 1F
57

Decodifica della risposta:

Date: 11/11/2021 10:05:00
CS: 10007 CM: 3 VAL: 0.0 (Precipitazione 5')

Date: 11/11/2021 10:10:00

CS: 9909 CM: 2 VAL: 2.13 (Livello 10')
CS: 10007 CM: 3 VAL: 0.0 (Precipitazione 5')

Date: 11/11/2021 10:15:00
CS: 9901 CM: 2 VAL: 23.1 (Temperatura 15')
CS: 9908 CM: 2 VAL: 13.2 (Batteria 15')
CS: 10007 CM: 3 VAL: 0.0 (Precipitazione 5')

Date: 11/11/2021 10:20:00
CS: 9909 CM: 2 VAL: 2.08 (Livello 10')
CS: 10007 CM: 3 VAL: 0.0 (Precipitazione 5')

Date: 11/11/2021 10:25:00
CS: 10007 CM: 3 VAL: 0.0 (Precipitazione 5')

Date: 11/11/2021 10:30:00
CS: 9901 CM: 2 VAL: 23.0 (Temperatura 15')
CS: 9908 CM: 2 VAL: 13.2 (Batteria 15')
CS: 9909 CM: 2 VAL: 2.05 (Livello 10')
CS: 10007 CM: 2 VAL: 0.0 (Precipitazione 5')

In aggiunta a quanto descritto, il comando !LBR può effettuare la sincronizzazione ^(*) dell'orologio del data-logger se vengono specificate data e ora da impostare. I parametri opzionali sono: *yyyy* (anno), *mm* (mese), *dd* (giorno), *hh* (ora), *nn* (minuti) e *ss* (secondi).

(*) Nota: la sincronizzazione dell'orologio ha effetto solo se l'impostazione attuale del data-logger differisce tra ± 3 sec. e ± 50 min.

3.3 Richiesta dati di archivio

Comando !DBR (*Data Binary Records*)

Questo comando può essere utilizzato per richiede/recuperare i record dati registrati in uno specifico periodo storico, ossia per recuperare eventuali 'buchi di dati' presenti in centrale.

La risposta restituisce tutti i record precedenti all'istante richiesto entro il periodo di riferimento (es. 30 min. a ritroso) compreso il record richiesto stesso. La ricerca dei dati viene eseguita nell'archivio per data/ora di avvenuta registrazione.

Il messaggio di richiesta deve specificare il numero di archivio da cui si vogliono estrarre i dati (*file*).

I parametri da specificare sono inoltre data e ora della registrazione: *yyyy* (anno), *mm* (mese), *dd* (giorno), *hh* (ora), *nn* (minuti) e *ss* (secondi).

Sintassi:

Messaggio	!DBR file yyyy mm dd hh nn ss
Risposta	<binary_records>

Ad esempio, volendo recuperare i dati della stazione ID n.198 dalle 9.31 alle 10.00 del 21/10/2021 il sistema centrale dovrà inviare una richiesta Modbus così costruita:

Comando (text): `◆A !DBR 6 2021 10 21 10 00 00◆\`

Hex: `C6 41 00 1A 21 44 42 52 20 36 20 32 30 32 31 20
31 30 20 32 31 20 31 30 20 30 30 20 30 30 82 5C`

Il formato dei dati ricevuti in risposta (<*binary_records*>) è identico a quello ricevuto per i dati correnti ed è descritto al par. 3.4 – *Tracciato binario*.

Di seguito un esempio di chiamata e risposta per il comando !DBR in formato Hexadecimal:

Comando: `C6 41 00 1A 21 44 42 52 20 36 20 32 30 32 31 20
31 31 20 31 30 20 32 32 20 30 30 20 30 30 BD 89`

Risposta: `C6 41 00 7F 15 0B 0A 15 23 00 27 17 29 00 00 00
00 15 0B 0A 15 28 00 26 B5 22 B8 52 40 1E 27 17
29 00 00 00 00 15 0B 0A 15 2D 00 26 AD 21 99 9A
41 AD 26 B4 21 33 33 41 53 27 17 29 00 00 00 00
15 0B 0A 15 32 00 26 B5 22 5C 29 40 1F 27 17 29
00 00 00 00 15 0B 0A 15 37 00 27 17 29 00 00 00
00 15 0B 0A 16 00 00 26 AD 21 99 9A 41 AD 26 B4
21 33 33 41 53 26 B5 22 00 00 40 20 27 17 21 00
00 00 00 48 FF`

Decodifica della risposta:

Date: 10/11/2021 21:35:00
CS: 10007 CM: 3 VAL: 0.0 (Precipitazione 5')

Date: 10/11/2021 21:40:00
CS: 9909 CM: 2 VAL: 2.48 (Livello 10')
CS: 10007 CM: 3 VAL: 0.0 (Precipitazione 5')

Date: 10/11/2021 21:45:00
CS: 9901 CM: 2 VAL: 21.7 (Temperatura 15')
CS: 9908 CM: 2 VAL: 13.2 (Batteria 15')
CS: 10007 CM: 3 VAL: 0.0 (Precipitazione 5')

Date: 10/11/2021 21:50:00
CS: 9909 CM: 2 VAL: 2.49 (Livello 10')
CS: 10007 CM: 3 VAL: 0.0 (Precipitazione 5')

Date: 10/11/2021 21:55:00
CS: 10007 CM: 3 VAL: 0.0 (Precipitazione 5')

Date: 10/11/2021 22:00:00
CS: 9901 CM: 2 VAL: 21.7 (Temperatura 15')
CS: 9908 CM: 2 VAL: 13.2 (Batteria 15')
CS: 9909 CM: 2 VAL: 2.50 (Livello 10')
CS: 10007 CM: 2 VAL: 0.0 (Precipitazione 5')

3.4 Tracciato record binario

Il messaggio di risposta alle richieste dati contiene uno o più records che vengono trasmessi in tracciato "binario compresso". La struttura di ciascun record è di seguito descritta:

Record Binario :

Campo	Dim. (Bytes)	Descrizione
Y	1	Anno a partire da 2000 (0 – 99)
M	1	Mese (1 – 12)
D	1	Giorno (1 – 31)
H	1	Ora (00 – 23)
N	1	Minuto (00 – 59)
S	1	Secondo (00 – 59)
DATA1	7	1° Buffer dato
DATA2	7	2° Buffer dato
...
DATAn	7	n° Buffer dato

Struttura del buffer dato:

Campo	Dim. (Bytes)	Descrizione
SENSOR	2	Codice sensore di misura (1 – 65535) ⁽¹⁾
TYPE	1	Tipo misura Numero decimali ⁽²⁾
VALUE	4	Valore del dato (codifica in floating-point 32-bit) ⁽³⁾

⁽¹⁾ Codice sensore univoco assegnato come: [ID Stazione] x 50 + [ID Misura]

⁽²⁾ bit 0...2: Numero decimali di rappresentazione (0 – 7).

bit 4...7: Tipo misura (1 = Istantanea; 2 = Media; 3 = Minima; 4 = Massima; ecc.).

bit 3: Flag (0 = Record unico o finale; 1 = Segue ulteriore record oltre a questo).

Normalmente il flag è 0, se vale 1 nell'ultimo buffer dato di ciascun record sta ad indicare che segue un altro record. Esempio:

Flag
YMDHNS
Dato1 0
Dato2 0
Dato3 1
YMDHNS
Dato1 0
Dato2 1
YMDHNS
Dato1 1
YMDHNS
Dato1 0
Dato2 0
Dato3 0
Dato4 0

⁽³⁾ In caso di dato non valido il valore risulterà = 0xFFFFFFFF.

3.5 Sincronizzazione orologio

Comando CLK

Il comando può essere utilizzato per leggere o scrivere la data/ora corrente sul data-logger ^(*).

I parametri da specificare per la scrittura sono: *hh* (ora), *nn* (minuti), *ss* (secondi), *dd* (giorno), *mm* (mese) e *yy[yy]* (anno).

Sintassi:

Messaggio	CLK <i>hh nn ss dd mm yy[yy]</i>
Risposta	<i>hh nn ss dd mm yyyy</i>

Il comando per leggere è semplicemente CLK (senza parametri).

^(*) Nota: Opzione non utilizzata dalla centrale, di norma la sincronizzazione dell'orologio viene eseguita tramite il comando !LBR.

3.6 Disattivazione risparmio energetico

Comando !WA

Il consumo energetico della stazione è dovuto in buona parte al consumo dell'apparato di trasmissione radio. Il data-logger gestisce il risparmio energetico accendendo periodicamente la radio solo in prossimità dell'intervallo di trasmissione (impostazione di default: 10 minuti ogni ½ ora).

Nel caso sia necessario mantenere accesa la radio oltre il tempo prestabilito si utilizza il seguente comando per disattivare temporaneamente il risparmio energetico impostando i minuti di accensione successivi.

Sintassi:

Messaggio	!WA 1 <i>pos val</i>
Risposta	<i>val</i>

I parametri da specificare sono:

- ^(*) *pos*: Posizione della locazione di memoria su cui scrivere il valore
val: Valore da impostare (in questo caso periodo minuti di accensione)

A comando correttamente eseguito la risposta conterrà il valore impostato.

^(*) Nota: nella configurazione specifica per ARPAV il campo *pos* utilizzato corrisponde alla locazione n. 99

Esempio:

!WA 1 99 240 imposta l'accensione della radio per 4 ore consecutive

3.7 *Lettura e scrittura parametro*

Seguono i comandi per la lettura e la scrittura di un parametro di configurazione del data-logger.

Con:

id: identificativo numerico del parametro (*)

val: valore decimale del parametro

Comando !RP (*Read Parameter*)

Esegue la lettura del valore di un parametro.

Sintassi:

Messaggio	!RP <i>id</i>
Risposta	<i>val</i>

Esempio:

Comando: !RP 11

Risposta: 1.23

Comando !WP (*Write Parameter*)

Esegue la scrittura del valore di un parametro.

Sintassi:

Messaggio	!WP <i>id val</i>
Risposta	<i>val</i>

Esempio:

Comando: !WP 11 1.45

Risposta: 1.45

(*) Nota: gli identificativi numerici dei parametri possono variare a seconda della configurazione del data-logger e dovranno corrispondere alla configurazione parametri stazione nella centrale Polaris.

3.8 Riavvio e reset data-logger

Il comando di riavvio esegue un reboot hardware/software del data-logger (spegnimento e riaccensione). La sintassi è la seguente:

Sintassi:

Messaggio	!TW
Risposta	Terminating Watchdog

Il comando di reset esegue invece solo un riavvio software del data-logger (chiusura e riavvio programma). La sintassi è la seguente:

Sintassi:

Messaggio	RESET MICROS
Risposta	RESET MICROS

3.9 Richiesta dati istantanei

Comando !RD (Read Data)

Il comando !RD esegue la lettura sequenziale dei dati da un file di archivio ritornando i tracciati record in formato testo. Il messaggio di richiesta deve specificare quanto segue:

file: n° archivio dati
user: id utente
pack: n. pacchetti (da 200 bytes):

Sintassi:

Messaggio	!RD file user pack
Risposta	<text_records>

Nello specifico, per la lettura dei dati istantanei (archivio n°1) il comando inviato dalla centrale Polaris è il seguente:

Comando: !RD 1 1 1

Risposta: *<tracciato record istantanei>*

In questo caso il tracciato record istantanei è sempre unico perché viene sovrascritto periodicamente dal data-logger. Per la documentazione dettagliata sul formato dei dati istantanei si consulti il par. 4 – *Tracciato record Dati Istantanei*.

3.10 Lettura configurazione data-logger

Comando !RD (Read Data)

Il comando !RD può essere utilizzato anche per la lettura sequenziale del file di configurazione del data-logger. La risposta ritornerà il contenuto del file in formato XML. Il messaggio di richiesta deve specificare quanto segue:

file: n° 0 (riservato per il file di configurazione)
user: id utente
pack: n. pacchetti (da 200 bytes):

Sintassi:

Messaggio	!RD 0 user pack
Risposta	<i><text_XML></i>

Nello specifico, la lettura del file di configurazione potrebbe essere utilizzato allo scopo di estrarre automaticamente i parametri di stazione contenuti nella sezione nodo XML **<PARAMETERS>** ... **</PARAMETERS>** della configurazione stessa (funzione "Ricostruisci" di Polaris).

In questo caso la centrale Polaris invia il comando concatenato **!RE 0 1 !RD 0 1 1** con il quale vengono richiesti i primi 200 bytes della configurazione. Il comando **!RE** (Restore) viene inviato solo la prima volta per portare il puntatore di lettura all'inizio del file.

Nel caso la prima richiesta non fosse sufficiente per ottenere il nodo completo, verranno ripetuti comandi concatenati del tipo **!RS 0 1 !RD 0 1 1** per richiedere i successivi 200 bytes di configurazione. Il comando **!RS** (Read Set) serve per far avanzare il puntatore di lettura del file.

Nota: Al fine di semplificare l'operazione di ricostruzione parametri, si noti che Polaris potrebbe accettare in risposta il nodo XML completo già al primo comando **!RD 0 1 1** anche se dovesse superare in totale i 200 bytes richiesti.

Esempio di risposta attesa (nodo XML):

```
<PARAMETERS>
  <PARAMETER name="Intervallo dati" id="11" unit="sec" value="600"/>
  <PARAMETER name="Scansione invio" id="12" unit="sec" value="600"/>
  <PARAMETER name="Offset idrometro" id="13" unit="m" value="0.05"/>
</PARAMETERS>
```

Nota:

Ciascun elemento **<PARAMETER .../>** presente nel nodo deve specificare i seguenti attributi:

name, id, unit, value

4 Tracciato record Dati Istantanei

I *Dati Istantanei* sono disponibili con questo tracciato nell'**area n°1** del buffer del supporto di memorizzazione (CIC, CRM, CBM, ecc.).

TESTA

La *Testa* del *tracciato record* contiene le informazioni generali che permettono l'identificazione del record sulla base dell'evento temporale e sulla stazione di appartenenza. In dettaglio il suo formato è il seguente:

STid_stazione , 6 , hh.nn.ss , gg , mm , aa , 1 , Mnum_input ,

il significato dei vari campi viene riassunto nella tabella che segue:

STid_stazione	Header che identifica l'inizio della <i>Testa</i> del record. E' composto dai caratteri "ST" più un numero indicante il codice (ID) della stazione. Ad esempio l' <i>Header ST01</i> identifica il record della Stazione 1 .
6	Codice costante che identifica il tracciato record a <i>struttura dinamica</i> .
hh.nn.ss	Ora, minuto e secondo dell'evento nel formato: hh.nn.ss .
gg	Giorno dell'evento di memorizzazione del record nel formato: gg .
mm	Mese dell'evento di memorizzazione del record nel formato: mm .
aa	Anno dell'evento di memorizzazione del record nel formato: aa .
1	Codice costante che identifica questo tipo di tracciato record per i <i>dati istantanei</i> .
Mnum_input	Header che identifica la fine della <i>Testa</i> del record. E' composto dal carattere "M"+un numero indicante il numero totale di parametri (input analogici e/o digitali ON/OFF) contenuti nel <i>Corpo</i> del tracciato record. Ad esempio M9 indica un totale di 9 parametri, mentre M50 indica un totale di 50 parametri.

4.1 CORPO

Il *Corpo* del *tracciato record* contiene i *Dati Istantanei* relativi ad ogni parametro di misura. Il *corpo* del tracciato record è quindi costituito dalla ripetizione di un blocco di informazioni ripetute tante volte quanti sono i parametri da rappresentare.

In particolare il formato del *corpo* si presenta nel modo seguente:

CORPO RECORD = blocco_1 + “,” + blocco_2 + “,” + blocco_3 +.....

Ogni blocco è composto da più sezioni che assieme costituiscono l'anagrafica del parametro rappresentato. Il blocco ha una *struttura dinamica*, pertanto ci possono essere molteplici *sezioni* atte a rappresentarlo. Nel caso specifico dei *dati istantanei*, sono sufficienti 3 sezioni.

In particolare ogni blocco è così strutturato:

BLOCCO_1 = id + “,” + sezione _1 + “,” + sezione _2 + “,” + sezione_3

dove in dettaglio i vari campi assumono il seguente significato:

id	Codice numerico che identifica in maniera univoca il parametro (l'input analogico o digitale on/off) al quale le sezioni successive fanno riferimento.
sezione_1	Codice alfanumerico composto da due campi contenenti una sola informazione riguardante il parametro di misura.
sezione_2	Codice alfanumerico composto da due campi contenenti una sola informazione riguardante il parametro di misura.
sezione_3	Codice alfanumerico composto da due campi contenenti una sola informazione riguardante il parametro di misura.

A sua volta ogni sezione è scomposta in due campi che assieme concorrono per dare una sola informazione riguardante il parametro. In particolare ogni sezione è così strutturata:

SEZIONE_N = costante + “,” + valore

dove in dettaglio i due campi assumono il seguente significato:

costante	Carattere ASCII (da ASCII 65 a ASCII 90) che esprime il significato del campo “ valore ” successivo.
valore	Valore numerico che rappresenta una caratteristica del parametro in esame. Tale caratteristica può essere <i>il valore di misura del parametro, un particolare codice, un particolare stato</i> , ecc.

La seguente tabella fornisce una descrizione dei valori che può assumere il campo “**costante**” e il campo successivo (**valore**) nella particolare applicazione dei *Dati Istantanei*:

costante	significato assunto dal campo successivo (valore).
T	TIPO (analogico o digitale) del parametro, identificato dal campo id .
A	VALORE ISTANTANEO del parametro, identificato dal campo id .
S	STATUS del parametro, identificato dal campo id .

Dalla tabella si nota che il campo “**valore**” assume significati diversi a seconda del campo che lo precede. Le tabelle che seguono forniscono una descrizione dei valori che può assumere il campo “**valore**” per ogni tipo di “**costante**”.

costante	valore
T	0 Per identificare un parametro analogico
	1 Per identificare un parametro digitale ON / OFF

costante	valore
A	Qualsiasi valore numerico (<i>Dato istantaneo</i> del parametro in oggetto)

costante	valore
S	0 Dato istantaneo del parametro acquisito correttamente
	1 Errore elettrico di acquisizione
	2 Parametro in condizione "Fuori Monitor"
	3 Dato istantaneo nella condizione di "Fuori Scala" (Over-Range)
	4 Dato istantaneo non valido per una variazione degli input digitali associati
	5 Dato istantaneo non valido per una invalidazione degli input analogici associati
	6 Dato istantaneo nella condizione di Allarme di Minimo
	7 Dato istantaneo nella condizione di Preallarme di Minimo
	8 Dato istantaneo nella condizione di Preallarme di Massimo
	9 Dato istantaneo nella condizione di Allarme di Massimo
	10 Dato non disponibile per attesa ricondizionamento
	11 Test
	12 Allarme (solo per un parametro digitale ON / OFF)
	13 Riferimento per calibrazione di Zero
	14 Dato in attesa di calibrazione di Zero
	15 Dato in calibrazione di Zero
	16 Dato in attesa di calibrazione di Span
	17 Dato in calibrazione di Span
	18 Dato in spurgo

Quindi un **blocco** atto a rappresentare un *Dato Istantaneo* di un parametro di misura potrebbe essere così strutturato:

id + “,” + T + “,” + valore_1 + “,” + A + “,” + valore_2 + “,” + S + “,” + valore_3

```
graph TD
    subgraph " "
        direction LR
        id[id] --- comma1[“,"]
        comma1 --- T[T]
        T --- comma2[“,"]
        comma2 --- v1[valore_1]
        v1 --- comma3[“,"]
        comma3 --- A[A]
        A --- comma4[“,"]
        comma4 --- v2[valore_2]
        v2 --- comma5[“,"]
        comma5 --- S[S]
        S --- comma6[“,"]
        comma6 --- v3[valore_3]
    end
    sezione1[sezione_1] --> T
    sezione2[sezione_2] --> A
    sezione3[sezione_3] --> S
```

Le tre sezioni rappresentate non sono posizionali, nel senso che non è necessario che rispettino un ordine preciso, ma è buona norma rappresentarle con la successione sopradescritta. A seconda dello *status* in cui si trova il parametro (*in acquisizione, in allarme, in calibrazione, ecc.*) definito dalla sezione **S + “,” + valore_3**, il suo corrispondente dato istantaneo definito dalla sezione **A + “,” + valore_1**, può trovarsi nella situazione di *invalidità*. In tale situazione, il **valore_1** non è più un dato numerico, ma viene sostituito con un carattere specifico che è il “*” (ASCII 42). Questo carattere viene anche definito con il nome di “**manca dato**”.

4.2 TERMINATORE

Infine il *Terminatore* è un campo che identifica la fine del tracciato record. Il *terminatore* è costituito dal carattere “#” (ASCII 35) + un numero indicante il totale dei campi (*fields*) del record tra *header d’inizio “ST”* e *terminatore “#”* compresi. Ad esempio **#50** indica che il record è composto in totale da 50 campi con *header “ST”* e *terminatore “#”* compresi.

4.3 Esempio di tracciato record Dati istantanei

La stazione di monitoraggio n°2 il giorno 4 Febbraio 1996 alle ore 13.45.15 ha memorizzato un *tracciato record* di *Dati Istantanei* simile al seguente:

ST02,6,13.45.15,04,02,96,1,M02,1,T,0,A,13.5,S,0,2,T,0,A,*,S,1,#23

dove:

ST02	Stazione n°2	STid_stazione
6	Codice costante del tracciato record a <i>struttura dinamica</i>	6
13.45.15	<i>Ora, minuto e secondo</i> dell'evento	hh.nn.ss
04	<i>Giorno</i> dell'evento	gg
02	<i>Mese</i> dell'evento	mm
96	<i>Anno</i> dell'evento	aa
1	Codice costante identificativo dei <i>Dati Istantanei</i>	1
M02	Significa che seguono 2 blocchi atti a rappresentare 2 parametri	Mnum_input
1	ID del <i>primo</i> parametro	id
T	Specifica che il campo seguente è il <i>tipo</i> di parametro	costante
0	Parametro di tipo <i>Analogico</i>	valore
A	Indica che il campo seguente è il <i>dato istantaneo</i> del parametro	costante
13.5	<i>Dato istantaneo</i> del parametro con id=1	valore
S	Specifica che il campo seguente è lo <i>status</i> del parametro	costante
0	Indica che il dato istantaneo è <i>stato acquisito correttamente</i>	valore
2	ID del <i>secondo</i> parametro	id
T	Specifica che il campo seguente è il <i>tipo</i> di parametro	costante
0	Parametro di tipo <i>Analogico</i>	valore
A	Indica che il campo seguente è il <i>dato istantaneo</i> del parametro	costante
*	<i>Manca dato</i> (non è stato possibile acquisire il dato)	valore
S	Specifica che il campo seguente è lo <i>status</i> del parametro	costante
1	Indica che il dato istantaneo <i>non è stato acquisito correttamente</i> per un errore elettrico	valore
#23	<i>Terminatore</i>	terminatore

N.B.: Il formato usato è quello con cui il *dato* è archiviato in maniera vera e propria, senza alcuna notazione.