



PROTOCOLLO DI COMUNICAZIONE PER SCHEDE A MICROCONTROLLORE

Codifica dei comandi - Revisione 1.0

F. Fiocchi, G. Maccaferri, A. Orlati, M. Morsiani

Rapporto interno IRA N°358/04



INDICE

1. INTRODUZIONE	4
2. CARATTERISTICHE DEL PROTOCOLLO	4
3. DESCRIZIONE DELLA CODIFICA	4
3.1. CALCOLO DEL CHECKSUM	5
3.2. TABELLA DEI CODICI DEL PROTOCOLLO	6
3.2. TABELLA DEI TIPI DI DATI	7
3.3. TABELLA DEI CODICI ASCII	7
4. COMANDI ESTESI	8
4.1. COMANDI ESTESI SENZA PARAMETRI	8
4.2. COMANDI ESTESI CON PARAMETRI	12
5. COMANDI ABBREVIATI	16
5.1. COMANDI ABBREVIATI SENZA PARAMETRI	16
5.2. COMANDI ABBREVIATI CON PARAMETRI	20
6. TRASMISSIONI DI GRANDI DIMENSIONI	24
CMD_INQUIRY	25
<i>COMANDO ESTESO</i>	25
<i>COMANDO ABBREVIATO</i>	25
CMD_RESET	26
<i>COMANDO ESTESO</i>	26
<i>COMANDO ABBREVIATO</i>	26
CMD_VERSION	27
<i>COMANDO ESTESO</i>	27
<i>COMANDO ABBREVIATO</i>	27
CMD_SAVE	28
<i>COMANDO ESTESO</i>	28
<i>COMANDO ABBREVIATO</i>	28
CMD_RESTORE	29
<i>COMANDO ESTESO</i>	29
<i>COMANDO ABBREVIATO</i>	29
CMD_GET_ADDR	30
<i>COMANDO ESTESO</i>	30
<i>COMANDO ABBREVIATO</i>	30
CMD_SET_ADDR	31
<i>COMANDO ESTESO</i>	31
<i>COMANDO ABBREVIATO</i>	31
CMD_GET_TIME	32
<i>COMANDO ESTESO</i>	32
<i>COMANDO ABBREVIATO</i>	32
CMD_SET_TIME	33
<i>COMANDO ESTESO</i>	33
<i>COMANDO ABBREVIATO</i>	33
CMD_GET_FRAME	34
<i>COMANDO ESTESO</i>	34
<i>COMANDO ABBREVIATO</i>	34
CMD_SET_FRAME	35
<i>COMANDO ESTESO</i>	35
<i>COMANDO ABBREVIATO</i>	35



CMD_GET_PORT	36
<i>COMANDO ESTESO</i>	36
<i>COMANDO ABBREVIATO</i>	37
CMD_SET_PORT	38
<i>COMANDO ESTESO</i>	38
<i>COMANDO ABBREVIATO</i>	39
CMD_GET_DATA	40
<i>COMANDO ESTESO</i>	40
<i>COMANDO ABBREVIATO</i>	41
CMD_SET_DATA	42
<i>COMANDO ESTESO</i>	42
<i>COMANDO ABBREVIATO</i>	43



1. INTRODUZIONE

Le nuove tecnologie elettroniche permettono la realizzazione di sistemi di monitoraggio e controllo ad intelligenza distribuita di basso costo. I vantaggi derivanti dall'utilizzo di questi sistemi risultano evidenti se si considerano le potenzialità di implementazione di apparecchiature in grado di essere controllate remotamente, di gestire in modo intelligente gli errori o gli allarmi, ma soprattutto flessibili nello sviluppo e nell'aggiornamento da non richiedere interventi fisici una volta installate.

Questo volume documenta dunque, il protocollo di comunicazione *multi-master* implementato sulle schede a microcontrollore sviluppate presso la stazione di Medicina ed utilizzate per la realizzazione dei sistemi di monitoraggio e di controllo necessari all'aggiornamento ed al potenziamento delle antenne.

2. CARATTERISTICHE DEL PROTOCOLLO

Per meglio comprendere le caratteristiche e la codifica del protocollo occorre chiarire la distinzione tra un dispositivo definito *master* ed uno *slave*, attraverso una definizione univoca.

Master: dispositivo che invia ad un altro dispositivo, un pacchetto di dati sul canale di comunicazione.

Slave: dispositivo che riconosce il pacchetto di dati a lui indirizzato e, se richiesto, torna al dispositivo chiamante un pacchetto di dati in risposta.

Il protocollo sviluppato è del tipo *multi-master*, in quanto permette la realizzazione di reti ad architettura semplice, come quella del tipo *master-slave*, oppure complessa come quella *multi-master* dove il controllo può essere ridondante o dove dispositivi diversi possono condividere le medesime risorse.

Se si realizza una rete semplice di tipo *master-slave*, dove il controllo è gestito da un unico dispositivo *master*, il protocollo *multi-master* risulta lievemente appesantito dalla presenza di un campo di indirizzamento, non necessario, che definisce l'indirizzo del *master*.

Nel caso si realizzi una rete complessa di tipo *multi-master*, dove il controllo può essere gestito in modo ridondante da più dispositivi *master* oppure più dispositivi (*master*) possono condividere le medesime risorse (*slave*), il protocollo *multi-master* risulta funzionale ed efficiente; infatti lo *slave*, mediante il campo del protocollo che identifica il *master*, è in grado di ritornare le risposte in modo univoco ai dispositivi che hanno inoltrato le domande. Perché la rete funzioni correttamente però, ogni dispositivo deve essere dotato dei meccanismi di gestione degli errori di collisione che si verificano quando più pacchetti di dati vengono inviati contemporaneamente.

3. DESCRIZIONE DELLA CODIFICA

La codifica del protocollo è stata realizzata tenendo conto delle esigenze attuali e di quelle che si possono presentare in futuro, in termini di interscambio dati fra dispositivi di elaborazione diversi per tecnologia e potenzialità. Per questo motivo i comandi e le relative risposte, codificati a sette bit per ragioni di compatibilità con i sistemi di comunicazione a controllo di errore, sono suddivisi in due gruppi principali: gli *estesi* e gli *abbreviati* che si distinguono unicamente per lunghezza, elemento determinante per una drastica riduzione del tempo di occupazione della linea di comunicazione. Entrambi i gruppi contengono i medesimi comandi contraddistinti con identificativi diversi che ne determinano l'appartenenza. All'interno dei due gruppi principali, è presente una ulteriore classificazione tra i comandi *senza parametri* e quelli *con parametri* caratterizzati, appunto, dalla presenza di parametri ovvero quei dati aggiuntivi necessari a definire completamente il comando.

La struttura di codifica prevede una suddivisione del pacchetto dei dati in più campi, come descritto in dettaglio nei capitoli **COMANDI ESTESI** e **COMANDI ABBREVIATI**. Il primo campo è sempre presente ed identifica se il pacchetto contiene un comando o una risposta mediante un carattere di apertura comunicazione diverso. Il secondo campo è presente solo nei comandi *con parametri* e nelle risposte che prevedono l'invio di dati. Il terzo ed ultimo campo è presente se il comando o la risposta sono di tipo *estesi* ed è costituito da due soli caratteri di cui quello di *checksum* viene utilizzato per controllare che la comunicazione tra due dispositivi sia esente da errori di trasmissione. Il calcolo del carattere di *checksum* è descritto dettagliatamente nel capitolo **CALCOLO DEL CHECKSUM**.

Le trasmissioni di pacchetti consistenti di dati possono mettere in difficoltà i dispositivi di elaborazione dotati di micro controllori, in quanto normalmente dotati di ristretti buffer di comunicazione. Nel protocollo è stato quindi previsto un meccanismo per frammentare la comunicazione in pacchetti di dati più piccoli denominati *frames* e per poter ricostruire in seguito la trasmissione nella sua pienezza. Una descrizione dettagliata dei meccanismi di frammentazione e ricostruzione sono descritti al capitolo **TRASMISSIONI DI GRANDI DIMENSIONI**.



3.1. CALCOLO DEL CHECKSUM

Il carattere di checksum si ottiene eseguendo la funzione logica EX-OR (OR esclusivo) di tutti i suoi precedenti a partire da quello di apertura della trasmissione incluso.

Comando: *master* → *slave*

“0x01,0x02,0x01,0x41,0x43,0x04”

Apertura:	“0x01”	(inizio trasmissione di comando <i>CMD_SOH</i>)
Indirizzo <i>slave</i> :	“0x02”	
Indirizzo <i>master</i> :	“0x01”	
Comando:	“0x41”	(comando <i>CMD_INQUIRY</i> esteso)
ID comando:	“0x00”	(numero progressivo del comando)
Checksum:	“0x01” hex-or “0x02” hex-or “0x01” hex-or “0x41” hex-or “0x00” hex-or	(apertore della trasmissione) (indirizzo dello <i>slave</i>) (indirizzo del <i>master</i>) (comando <i>CMD_INQUIRY</i>) (numero progressivo del comando)
Chiusura:	“0x43” “0x04”	(risultato del checksum) (fine trasmissione di comando <i>CMD_EOT</i>)

Risposta: *slave* → *master*

“0x02,0x01,0x02,0x41,0x00,0x00,0x0B,0x00,0x00,0x00,0x14,0x02,0x0C,0x10,0x11,0x37,0x00,0x00,0x67,0x03”

Apertura:	“0x02”	(inizio trasmissione di risposta <i>CMD_STX</i>)
Indirizzo <i>master</i> :	“0x01”	
Indirizzo <i>slave</i> :	“0x02”	
Risposta a:	“0x41”	(comando <i>CMD_INQUIRY</i> esteso)
ID comando:	“0x00”	(numero progressivo del comando)
Esito:	“0x00”	(<i>CMD_ACK</i> , tutto ok)
Data size:	“0x0B”	(seguono 10 caratteri)
Comando eseguito:	“0x00”	(nessun comando eseguito fino ad ora)
ID comando:	“0x00”	(numero progressivo del comando)
Esito:	“0x00”	(<i>CMD_ACK</i> , tutto ok)
Data:	“0x14,0x02,0x0C,0x10”	(2002/12/16 data di esecuzione)
Ora:	“0x11,0x37,0x00,0x00”	(17:55:00,0 ora di esecuzione)
Checksum:	“0x01” hex-or “0x02” hex-or “0x01” hex-or “0x41” hex-or “0x00” hex-or “0x00” hex-or “0x0B” hex-or “0x00” hex-or “0x00” hex-or “0x00” hex-or “0x14” hex-or “0x02” hex-or “0x0C” hex-or “0x10” hex-or “0x11” hex-or “0x37” hex-or “0x00” hex-or “0x00”	(apertore della trasmissione) (indirizzo dello <i>slave</i>) (indirizzo del <i>master</i>) (comando <i>CMD_INQUIRY</i>) (numero progressivo del comando) (esito <i>CMD_ACK</i> , tutto ok) (seguono 11 caratteri) (nessun comando eseguito fino ad ora) (numero progressivo del comando) (<i>CMD_ACK</i> , tutto ok) (secolo 20) (anno 02) (mese 12) (giorno 16) (ore 17) (minuti 55) (secondi 0) (centesimi 0)
Chiusura:	“0x67” “0x04”	(risultato del checksum) (fine trasmissione di risposta <i>CMD_ETX</i>)



3.2. TABELLA DEI CODICI DEL PROTOCOLLO

La codifica dei comandi utilizza soltanto sette bit in un campo esadecimale che va quindi da 0x00 a 0x7F; questo ne permette la validità anche con impostazioni particolari della porta di comunicazione (sette bit di dati ed uno di parità).

DESCRIPTION	NAME	HEX.	DEC.	ASCII
TRANSMISSION HEADER	CMD_SOH	0x01	1	SOH
	CMD_STX	0x02	4	EOT
TRANSMISSION ENDING	CMD_ETX	0x03	2	STX
	CMD_EOT	0x04	3	ETX
EXECUTION RESULT	CMD_ACK	0x00	0	NULL
	CMD_ERR_CMD	0x01	1	SOH
	CMD_ERR_CHKS	0x02	2	STX
	CMD_ERR_FORM	0x03	3	ETX
	CMD_ERR_DATA	0x04	4	EOT
	CMD_ERR_TOUT	0x05	5	ENQ
	CMD_ERR_ADDR	0x06	6	ACK
	CMD_ERR_TIME	0x07	7	BEL
	CMD_ERR_FRAME_SIZE	0x08	8	BS
	CMD_ERR_DATA_TYPE	0x09	9	HT
	CMD_ERR_PORT_TYPE	0x0A	10	LF
	CMD_ERR_PORT_NUMBER	0x0B	11	VT
CMD_ERR_DATA_SIZE	0x0C	12	FF	
EXTENDED COMMAND	CMD_INQUIRY	0x41	65	A
	CMD_RESET	0x42	66	B
	CMD_VERSION	0x43	67	C
	CMD_SAVE	0x44	68	D
	CMD_RESTORE	0x45	69	E
	CMD_GET_ADDR	0x46	70	F
	CMD_SET_ADDR	0x47	71	G
	CMD_GET_TIME	0x48	72	H
	CMD_SET_TIME	0x49	73	I
	CMD_GET_FRAME	0x4A	74	J
	CMD_SET_FRAME	0x4B	75	K
	CMD_GET_PORT	0x4C	76	L
	CMD_SET_PORT	0x4D	77	M
	CMD_GET_DATA	0x4E	78	N
CMD_SET_DATA	0x4F	79	O	
ABBREVIATED COMMAND	CMD_INQUIRY	0x61	97	a
	CMD_RESET	0x62	98	b
	CMD_VERSION	0x63	99	c
	CMD_SAVE	0x64	100	d
	CMD_RESTORE	0x65	101	e
	CMD_GET_ADDR	0x66	102	f
	CMD_SET_ADDR	0x67	103	g
	CMD_GET_TIME	0x68	104	h
	CMD_SET_TIME	0x69	105	i
	CMD_GET_FRAME	0x6A	106	j
	CMD_SET_FRAME	0x6B	107	k
	CMD_GET_PORT	0x6C	108	l
	CMD_SET_PORT	0x6D	109	m
	CMD_GET_DATA	0x6E	110	n
CMD_SET_DATA	0x6F	111	o	



3.2. TABELLA DEI TIPI DI DATI

DATA TYPE	RANGE	NAME	HEX.	DEC.	ASCII	BYTE	BITS
STRING	0 to 127	CMD_STRING	0x00	0	NULL	1	7 & 8
BIT	0 or 1	CMD_BIT	0x01	1	SOH	1	8
CHAR	-128 to +127	CMD_CHAR	0x02	2	STX	1	8
BYTE	0 to 255	CMD_BYTE	0x03	3	ETX	1	8
SHORT	-32768 to +32767	CMD_SHORT	0x04	4	EOT	2	8
WORD	0 to 65535	CMD_WORD	0x05	5	ENQ	2	8
LONG	-2147483648 to +2147483647	CMD_LONG	0x06	6	ACK	4	8
DOUBLE WORD	0 to 4294967295	CMD_DWORD	0x07	7	BEL	4	8
FLOAT	-3.402823466E-38 to +3.402823466E+38	CMD_FLOAT	0x08	8	BS	4	8
DOUBLE	-1.7976931348623158E-308 to +1.7976931348623158E+308	CMD_DOUBLE	0x09	9	HT	8	8
STRUCT		CMD_STRUCT	0x3F	63	?	x	8
BIT	0 or 1	CMD_7BIT	0x41	65	A	1	7
CHAR	-128 to +127	CMD_7CHAR	0x42	66	B	2	7
BYTE	0 to 255	CMD_7BYTE	0x43	67	C	2	7
SHORT	-32768 to +32767	CMD_7SHORT	0x44	68	D	3	7
WORD	0 to 65535	CMD_7WORD	0x45	69	E	3	7
LONG	-2147483648 to +2147483647	CMD_7LONG	0x46	70	F	5	7
DOUBLE WORD	0 to 4294967295	CMD_7DWORD	0x47	71	G	5	7
FLOAT	-3.402823466E-38 to +3.402823466E+38	CMD_7FLOAT	0x48	72	H	5	7
DOUBLE	-1.7976931348623158E-308 to +1.7976931348623158E+308	CMD_7DOUBLE	0x49	73	I	9	7
STRUCT		CMD_7STRUCT	0x7F	127	DEL	x	7

3.3. TABELLA DEI CODICI ASCII

DEC.	HEX.	ASCII	DEC.	HEX.	ASCII	DEC.	HEX.	ASCII	DEC.	HEX.	ASCII
0	0x00	NULL	32	0x20	SPACE	64	0x40	@	96	0x60	`
1	0x01	SOH	33	0x21	!	65	0x41	A	97	0x61	a
2	0x02	STX	34	0x22	“	66	0x42	B	98	0x62	b
3	0x03	ETX	35	0x23	#	67	0x43	C	99	0x63	c
4	0x04	EOT	36	0x24	\$	68	0x44	D	100	0x64	d
5	0x05	ENQ	37	0x25	%	69	0x45	E	101	0x65	e
6	0x06	ACK	38	0x26	&	70	0x46	F	102	0x66	f
7	0x07	BEL	39	0x27	·	71	0x47	G	103	0x67	g
8	0x08	BS	40	0x28	(72	0x48	H	104	0x68	h
9	0x09	HT	41	0x29)	73	0x49	I	105	0x69	i
10	0x0A	LF	42	0x2A	*	74	0x4A	J	106	0x6A	j
11	0x0B	VT	43	0x2B	+	75	0x4B	K	107	0x6B	k
12	0x0C	FF	44	0x2C	,	76	0x4C	L	108	0x6C	l
13	0x0D	CR	45	0x2D	-	77	0x4D	M	109	0x6D	m
14	0x0E	SO	46	0x2E	.	78	0x4E	N	110	0x6E	n
15	0x0F	SI	47	0x2F	/	79	0x4F	O	111	0x6F	o
16	0x10	DLE	48	0x30	0	80	0x50	P	112	0x70	p
17	0x11	DC1	49	0x31	1	81	0x51	Q	113	0x71	q
18	0x12	DC2	50	0x32	2	82	0x52	R	114	0x72	r
19	0x13	DC3	51	0x33	3	83	0x53	S	115	0x73	s
20	0x14	DC4	52	0x34	4	84	0x54	T	116	0x74	t
21	0x15	NAK	53	0x35	5	85	0x55	U	117	0x75	u
22	0x16	SYN	54	0x36	6	86	0x56	V	118	0x76	v
23	0x17	ETB	55	0x37	7	87	0x57	W	119	0x77	w
24	0x18	CAN	56	0x38	8	88	0x58	X	120	0x78	x
25	0x19	EM	57	0x39	9	89	0x59	Y	121	0x79	y
26	0x1A	SUB	58	0x3A	:	90	0x5A	Z	122	0x7A	z
27	0x1B	ESC	59	0x3B	;	91	0x5B	[123	0x7B	{
28	0x1C	FS	60	0x3C	<	92	0x5C	\	124	0x7C	
29	0x1D	GS	61	0x3D	=	93	0x5D]	125	0x7D	}
30	0x1E	RS	62	0x3E	>	94	0x5E	^	126	0x7E	~
31	0x1F	US	63	0x3F	?	95	0x5F	_	127	0x7F	DEL



4. COMANDI ESTESI

Il set dei *comandi estesi* è suddiviso in due sezioni. La prima sezione è costituita dai comandi *senza parametri*, ovvero che non richiedono il passaggio di parametri per l'esecuzione. Nella seconda sezione sono raccolti tutti i comandi *con parametri*, che richiedono quindi il passaggio di parametri per la completa definizione del comando.

4.1. COMANDI ESTESI SENZA PARAMETRI

I *comandi estesi senza parametri*, sono costituiti da stringhe di lunghezza fissa di sette caratteri, suddivise in due campi.

Il campo iniziale contiene cinque caratteri:

- COMMAND HEADER - carattere di sincronizzazione (inizio trasmissione del comando).
- SLAVE ADDRESS - indirizzo del dispositivo di destinazione del comando.
- MASTER ADDRESS - indirizzo del dispositivo che invia il comando.
- COMMAND - identificativo del comando.
- ID - numero progressivo del comando.

Il campo finale contiene due caratteri:

- CHECKSUM - checksum ottenuto eseguendo la funzione logica EX-OR di tutti i caratteri del 1° campo del comando.
- COMMAND ENDING - carattere che termina la trasmissione del comando.

La stringa del comando ha dunque il seguente formato.

COMMAND FORMAT						
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID	CHECKSUM	COMMAND ENDING

Le risposte ai *comandi estesi senza parametri*, sono costituite da stringhe di lunghezza variabile suddivise in due o tre campi.

Il campo iniziale contiene sei caratteri:

- ANSWER HEADER - carattere di sincronizzazione (inizio trasmissione della risposta).
- MASTER ADDRESS - indirizzo del dispositivo di destinazione della risposta.
- SLAVE ADDRESS - indirizzo del dispositivo che invia la risposta.
- COMMAND - identificativo del comando di pertinenza della risposta.
- ID - numero progressivo del comando.
- EXECUTION RESULT - esito dell'esecuzione del comando.

Il campo centrale è assente nel caso in cui il comando non sia stato riconosciuto od eseguito correttamente. Se presente, perché previsto dal tipo di comando, è costituito da almeno un carattere che identifica il numero dei seguenti:

- DATA SIZE - identifica il numero dei caratteri che seguono nel campo (da 0 a 126max.).
- DATA - caratteri che costituiscono i dati della risposta (max.126 caratteri).

Il campo finale contiene due caratteri:

- CHECKSUM - checksum ottenuto eseguendo la funzione logica EX-OR di tutti i caratteri del 1° e, se presente, del 2° campo della risposta.
- ANSWER ENDING - carattere che termina la trasmissione della risposta.



La stringa della risposta, nel caso in cui non vi siano dati da ritornare oppure ci siano errori nell'esecuzione, ha il seguente formato.

ANSWER FORMAT							
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTE N°8
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	CHECKSUM	ANSWER ENDING

Quando la dimensione dei dati richiesti dal comando risulta essere nulla (DATA SIZE = 0), ovvero i dati non sono disponibili, la risposta ha il seguente formato.

ANSWER FORMAT								
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTE N°8	BYTE N°9
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	DATA SIZE	CHECKSUM	ANSWER ENDING

Quando la dimensione dei dati richiesti dal comando non è nulla, la risposta ha il seguente formato.

ANSWER FORMAT									
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTES N°8 to N	BYTE N° N + 1	BYTE N° N + 2
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	DATA SIZE	DATA	CHECKSUM	ANSWER ENDING

Esempio: se DATA SIZE = 10, DATA è di 10 bytes, il CHECKSUM è il 19° byte e ANSWER ENDING il 20° byte della stringa di risposta.



4.1.1. TABELLA DEI COMANDI ESTESI SENZA PARAMETRI

BYTE (N°)	BYTE NAME	VALUE (hex.)	NAME	DESCRIPTION
1	COMMAND HEADER	0x01	CMD_SOH	Carattere di sincronizzazione inizio trasmissione tra il dispositivo master e lo slave.
2	SLAVE ADDRESS	0x00		Tutti gli slave che ricevono il comando, lo devono eseguire senza inviare la risposta.
		0x7F		Tutti gli slave che ricevono il comando, lo devono eseguire ed inviare la risposta.
		0x01 : 0x7E		Lo slave indirizzato deve eseguire il comando ed inviare una risposta.
3	MASTER ADDRESS	0x01 : 0x7E		Indirizzo del dispositivo master a cui lo slave deve rispondere, se richiesto.
4	COMMAND	0x41	CMD_INQUIRY	Richiesta di stato del dispositivo slave selezionato. Il dispositivo ritorna una stringa contenente identificativo ed esito dell'ultimo comando eseguito, insieme a data ed ora dell'esecuzione.
		0x42	CMD_RESET	Reset a caldo del dispositivo slave indirizzato. Il dispositivo interrompe tutte le attività e rilancia l'esecuzione del suo programma da capo.
		0x43	CMD_VERSION	Richiesta della versione dell'hardware e del firmware del dispositivo slave indirizzato. Il dispositivo invia una stringa alfanumerica indicante la sua versione di hardware e firmware.
		0x44	CMD_SAVE	Il dispositivo slave deve salvare i dati di configurazione nella memoria non volatile.
		0x45	CMD_RESTORE	Il dispositivo slave deve ripristinare i dati di configurazione presenti nella memoria non volatile ed eseguire di seguito un reset a caldo.
		0x46	CMD_GET_ADDR	Il dispositivo slave deve ritornare il suo indirizzo di protocollo.
		0x48	CMD_GET_TIME	Il dispositivo slave deve ritornare la data e l'orario del suo orologio interno.
		0x4A	CMD_GET_FRAME	Il dispositivo slave deve ritornare la lunghezza massima del frame di comunicazione, che dipende sostanzialmente dalla dimensione del suo buffer di comunicazione. Questo valore limita il master nella dimensione dei parametri dei comandi.
5	ID	0x00 : 0x7F		Questo è un numero progressivo inviato al dispositivo slave e che lo slave ritorna quando invia la risposta relativa all'esito del processo.
6	CHECKSUM	0x00 : 0x7F		Checksum della trasmissione calcolato eseguendo la funzione di EX-OR dei byte dal 1° al 5° incluso.
7	COMMAND ENDING	0x04	CMD_EOT	Carattere che conclude la trasmissione tra il dispositivo master e lo slave.



4.1.2. TABELLA DELLE RISPOSTE AI COMANDI ESTESI SENZA PARAMETRI

BYTE (N°)	BYTE NAME	VALUE (hex.)	NAME	DESCRIPTION
1	ANSWER HEADER	0x02	CMD_STX	Carattere di sincronizzazione inizio trasmissione tra il dispositivo slave ed il master.
2	MASTER ADDRESS	0x01 : 0x7E		Indirizzo del dispositivo master a cui tornare la risposta.
3	SLAVE ADDRESS	0x01 : 0x7E		Indirizzo del dispositivo slave che ha eseguito il comando.
4	COMMAND	0x41 0x42 0x43 0x44 0x45 0x46 0x48 0x4A	CMD_INQUIRY CMD_RESET CMD_VERSION CMD_SAVE CMD_RESTORE CMD_GET_ADDR CMD_GET_TIME CMD_GET_FRAME	Il dispositivo slave replica il comando ricevuto dal master.
5	ID	0x00 : 0x7F		Il dispositivo slave ritorna lo stesso campo ID trovato nel comando eseguito.
6	EXECUTION RESULT	0x00	CMD_ACK	Comando eseguito dallo slave senza errori.
		0x01	CMD_ERR_CMD	Comando non previsto dallo slave selezionato.
		0x02	CMD_ERR_CHKS	Errore nel checksum del comando ricevuto.
		0x03	CMD_ERR_FORM	Errore nel formato dei parametri ricevuti.
		0x04	CMD_ERR_DATA	Errore nei parametri del comando ricevuto.
		0x05	CMD_ERR_TOUT	Scaduto il timeout di esecuzione del comando
		0x06	CMD_ERR_ADDR	Indirizzo errato.
		0x07	CMD_ERR_TIME	Orario errato.
		0x08	CMD_ERR_FRAME_SIZE	Errore nella dimensione del frame, troppo grande per il dispositivo.
		0x09	CMD_ERR_DATA_TYPE	Tipo di dati non gestito.
		0x0A	CMD_ERR_PORT_TYPE	Il tipo di porta non esistente
		0x0B	CMD_ERR_PORT_NUMBER	Il numero di porta non esistente
		0x0C	CMD_ERR_DATA_SIZE	Errore nella dimensione dei dati.
		0x10 : 0x7F		Errore dello slave nell'esecuzione del comando. Il valore ritornato dipende dal tipo di comando e dal firmware del dispositivo slave.
7	DATA SIZE	0x00 : 0x7F		Dimensione in byte dei dati che seguono. N.B: questo byte è presente se richiesto dal tipo di comando e dall'esito positivo della sua esecuzione.
8 to N	DATA	0x00 : 0x7F		Pacchetto dei dati richiesti dal master, la cui lunghezza dipende dal comando. N.B: questi bytes sono presenti se richiesto dal tipo di comando e dall'esito positivo della sua esecuzione.
N + 1	CHECKSUM	0x00 : 0x7F		Checksum della trasmissione dello slave, calcolato eseguendo la funzione di EX-OR dei byte dal 1° all'N° incluso.
N + 2	ANSWER ENDING	0x03	CMD_ETX	Carattere che conclude la trasmissione dello slave verso il master.



4.2. COMANDI ESTESI CON PARAMETRI

I *comandi estesi con parametri*, sono costituiti da stringhe di lunghezza variabile suddivise in tre campi.

Il campo iniziale contiene cinque caratteri:

- COMMAND HEADER - carattere di sincronizzazione (inizio trasmissione del comando).
- SLAVE ADDRESS - indirizzo del dispositivo di destinazione del comando.
- MASTER ADDRESS - indirizzo del dispositivo che invia il comando.
- COMMAND - identificativo del comando.
- ID - numero progressivo del comando.

Il campo centrale è costituito da almeno un carattere che identifica il numero dei seguenti:

- PARAMETER SIZE - identifica il numero dei caratteri che seguono nel campo (da 1 a 126max.).
- PARAMETER - caratteri che costituiscono i parametri del comando (max.126 caratteri).

Il campo finale contiene due caratteri:

- CHECKSUM - checksum ottenuto eseguendo la funzione logica EX-OR di tutti i caratteri del 1° e del 2° campo del comando.
- COMMAND ENDING - carattere che termina la trasmissione del comando.

La stringa del comando ha dunque il seguente formato.

COMMAND FORMAT								
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTES N°7 to N	BYTE N° N + 1	BYTE N° N + 2
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID	PARAMETER SIZE	PARAMETER	CHECKSUM	COMMAND ENDING

Le risposte ai *comandi estesi con parametri*, sono costituite da stringhe di lunghezza variabile suddivise in due o tre campi.

Il campo iniziale contiene sei caratteri:

- ANSWER HEADER - carattere di sincronizzazione (inizio trasmissione della risposta).
- MASTER ADDRESS - indirizzo del dispositivo di destinazione della risposta.
- SLAVE ADDRESS - indirizzo del dispositivo che invia la risposta.
- COMMAND - identificativo del comando di pertinenza della risposta.
- ID - numero progressivo del comando.
- EXECUTION RESULT - esito dell'esecuzione del comando.

Il campo centrale è assente nel caso in cui il comando non sia stato riconosciuto od eseguito correttamente. Se presente, perché previsto dal tipo di comando, è costituito da almeno un carattere che identifica il numero dei seguenti:

- DATA SIZE - identifica il numero dei caratteri che seguono nel campo (da 0 a 126max.).
- DATA - caratteri che costituiscono i dati della risposta (max.126 caratteri).

Il campo finale contiene due caratteri:

- CHECKSUM - checksum ottenuto eseguendo la funzione logica EX-OR di tutti i caratteri del 1° e, se presente, del 2° campo della risposta.
- ANSWER ENDING - carattere che termina la trasmissione della risposta.



La stringa della risposta, nel caso in cui non vi siano dati da ritornare oppure ci siano errori nell'esecuzione, ha il seguente formato.

ANSWER FORMAT							
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTE N°8
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	CHECKSUM	ANSWER ENDING

Quando la dimensione dei dati richiesti dal comando risulta essere nulla (DATA SIZE = 0), ovvero i dati non sono disponibili, la risposta ha il seguente formato.

ANSWER FORMAT								
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTE N°8	BYTE N°9
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	DATA SIZE	CHECKSUM	ANSWER ENDING

Quando la dimensione dei dati richiesti dal comando non è nulla, la risposta ha il seguente formato.

ANSWER FORMAT									
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTES N°8 to N	BYTE N° N + 1	BYTE N° N + 2
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	DATA SIZE	DATA	CHECKSUM	ANSWER ENDING

Esempio: se DATA SIZE = 10, DATA è di 10 bytes, il CHECKSUM è il 19° byte e ANSWER ENDING il 20° byte della stringa di risposta.



4.2.1. TABELLA DEI COMANDI ESTESI CON PARAMETRI

BYTE (N°)	BYTE NAME	VALUE (hex.)	NAME	DESCRIPTION
1	COMMAND HEADER	0x01	CMD_SOH	Carattere di sincronizzazione inizio trasmissione tra il dispositivo master e lo slave.
2	SLAVE ADDRESS	0x00		Tutti gli slave che ricevono il comando, lo devono eseguire senza inviare la risposta.
		0x7F		Tutti gli slave che ricevono il comando, lo devono eseguire ed inviare la risposta.
		0x01 : 0x7E		Lo slave indirizzato deve eseguire il comando ed inviare una risposta.
3	MASTER ADDRESS	0x01 : 0x7E		Indirizzo del dispositivo master a cui lo slave deve rispondere, se richiesto.
4	COMMAND	0x47	CMD_SET_ADDR	Il dispositivo slave deve aggiornare il suo indirizzo di protocollo con il nuovo valore comunicato dal master. Vedi (*).
		0x49	CMD_SET_TIME	Il dispositivo slave deve aggiornare data e ora del suo orologio interno con i nuovi valori comunicati dal master.
		0x4B	CMD_SET_FRAME	
		0x4C	CMD_GET_PORT	Il dispositivo slave deve ritornare la configurazione della porta specificata dal comando, dipendentemente dalla sua versione di firmware ed hardware.
		0x4D	CMD_SET_PORT	Il dispositivo slave deve configurare la porta specificata dal comando, dipendentemente dalla sua versione di firmware ed hardware. Vedi (*)
		0x4E	CMD_GET_DATA	Il dispositivo slave deve ritornare i dati presenti sul canale della porta specificata dal comando, dipendentemente dalla sua versione di firmware ed hardware.
		0x4F	CMD_SET_DATA	Il dispositivo slave deve inviare i dati sul canale della porta specificata dal comando, dipendentemente dalla sua versione di firmware ed hardware.
5	ID	0x00 : 0x7F		Questo è un numero progressivo inviato al dispositivo slave e che lo slave ritorna quando invia la risposta relativa all'esito del processo.
6	PARAMETER SIZE	0x01 : 0x7F		Dimensione in byte dei parametri che seguono.
7 to N	PARAMETER	0x00 : 0x7F		Pacchetto dei parametri previsti dal comando.
N + 1	CHECKSUM	0x00 : 0x7F		Checksum della trasmissione del master, calcolato eseguendo la funzione di EX-OR dei byte dal 1° all'N° incluso.
N + 2	COMMAND ENDING	0x04	CMD_EOT	Carattere che conclude la trasmissione tra il dispositivo master e lo slave.

(*) N.B.: Senza un successivo comando CMD_SAVE_CONF, l'impostazione viene perduta in caso di reset a caldo od una mancanza di alimentazione allo slave.



4.2.2. TABELLA DELLE RISPOSTE AI COMANDI ESTESI CON PARAMETRI

BYTE (N°)	BYTE NAME	VALUE (hex.)	NAME	DESCRIPTION
1	ANSWER HEADER	0x02	CMD_STX	Carattere di sincronizzazione inizio trasmissione tra il dispositivo slave ed il master.
2	MASTER ADDRESS	0x01 : 0x7E		Indirizzo del dispositivo master a cui tornare la risposta.
3	SLAVE ADDRESS	0x01 : 0x7E		Indirizzo del dispositivo slave che ha eseguito il comando.
4	COMMAND	0x47 0x49 0x4B 0x4C 0x4D 0x4E 0x4F	CMD_SET_ADDR CMD_SET_TIME CMD_SET_FRAME CMD_GET_PORT CMD_SET_PORT CMD_GET_DATA CMD_SET_DATA	Il dispositivo slave replica il comando ricevuto dal master.
5	ID	0x00 : 0x7F		Il dispositivo slave ritorna lo stesso campo ID trovato nel comando eseguito.
6	EXECUTION RESULT	0x00	CMD_ACK	Comando eseguito dallo slave senza errori.
		0x01	CMD_ERR_CMD	Comando non previsto dallo slave selezionato.
		0x02	CMD_ERR_CHKS	Errore nel checksum del comando ricevuto.
		0x03	CMD_ERR_FORM	Errore nel formato dei parametri ricevuti.
		0x04	CMD_ERR_DATA	Errore nei parametri del comando ricevuto.
		0x05	CMD_ERR_TOUT	Scaduto il timeout di esecuzione del comando
		0x06	CMD_ERR_ADDR	Indirizzo errato.
		0x07	CMD_ERR_TIME	Orario errato.
		0x08	CMD_ERR_FRAME_SIZE	Errore nella dimensione del frame, troppo grande per il dispositivo.
		0x09	CMD_ERR_DATA_TYPE	Tipo di dati non gestito.
		0x0A	CMD_ERR_PORT_TYPE	Il tipo di porta non esistente
		0x0B	CMD_ERR_PORT_NUMBER	Il numero di porta non esistente
		0x0C	CMD_ERR_DATA_SIZE	Errore nella dimensione dei dati.
		0x10 : 0x7F		Errore dello slave nell'esecuzione del comando. Il valore ritornato dipende dal tipo di comando e dal firmware del dispositivo slave.
7	DATA SIZE	0x00 : 0x7F		Dimensione in byte dei dati che seguono. N.B: questo byte è presente se richiesto dal tipo di comando e dall'esito positivo della sua esecuzione.
8 to N	DATA	0x00 : 0x7F		Pacchetto dei dati richiesti dal master, la cui lunghezza dipende dal comando. N.B: questi bytes sono presenti se richiesto dal tipo di comando e dall'esito positivo della sua esecuzione.
N + 1	CHECKSUM	0x00 : 0x7F		Checksum della trasmissione dello slave, calcolato eseguendo la funzione di EX-OR dei byte dal 1° all'N° incluso.
N + 2	ANSWER ENDING	0x03	CMD_ETX	Carattere che conclude la trasmissione dello slave verso il master.



5. COMANDI ABBREVIATI

Il set dei *comandi abbreviati* è suddiviso in due sezioni. La prima sezione è costituita dai comandi *senza parametri*, ovvero che non richiedono il passaggio di parametri per l'esecuzione. Nella seconda sezione sono raccolti tutti i comandi *con parametri*, che richiedono quindi il passaggio di parametri per la completa definizione del comando.

5.1. COMANDI ABBREVIATI SENZA PARAMETRI

I *comandi abbreviati senza parametri*, sono costituiti da stringhe di lunghezza fissa di cinque caratteri:

- COMMAND HEADER - carattere di sincronizzazione (inizio trasmissione del comando).
- SLAVE ADDRESS - indirizzo del dispositivo di destinazione del comando.
- MASTER ADDRESS - indirizzo del dispositivo che invia il comando.
- COMMAND - identificativo del comando.
- ID - numero progressivo del comando.

La stringa del comando ha dunque il seguente formato.

COMMAND FORMAT				
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID

Le risposte ai *comandi abbreviati senza parametri*, sono costituite da stringhe di lunghezza variabile suddivise in uno o due campi.

Il campo iniziale contiene sei caratteri:

- ANSWER HEADER - carattere di sincronizzazione (inizio trasmissione della risposta).
- MASTER ADDRESS - indirizzo del dispositivo di destinazione della risposta.
- SLAVE ADDRESS - indirizzo del dispositivo che invia la risposta.
- COMMAND - identificativo del comando di pertinenza della risposta.
- EXECUTION RESULT - esito dell'esecuzione del comando.
- ID - numero progressivo del comando.

Il campo finale è assente nel caso in cui il comando non sia stato riconosciuto od eseguito correttamente. Se presente, perché previsto dal tipo di comando stesso, è costituito da almeno un carattere che identifica il numero dei seguenti:

- DATA SIZE - identifica il numero dei caratteri che seguono nel campo (da 0 a 126max.).
- DATA - caratteri che costituiscono i dati della risposta (max.126 caratteri).



La stringa della risposta, nel caso in cui non vi siano dati da ritornare oppure ci siano errori nell'esecuzione, ha il seguente formato.

ANSWER FORMAT					
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT

Quando la dimensione dei dati richiesti dal comando risulta essere nulla (DATA SIZE = 0), ovvero i dati non sono disponibili, la risposta ha il seguente formato.

ANSWER FORMAT						
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	DATA SIZE

Quando la dimensione dei dati richiesti dal comando non è nulla, la risposta ha il seguente formato.

ANSWER FORMAT							
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTES N° 8 to N
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	DATA SIZE	DATA

Esempio: se DATA SIZE = 10, DATA è di 10 bytes e l'intera trasmissione è lunga 17 bytes.



5.1.1. TABELLA DEI COMANDI ABBREVIATI SENZA PARAMETRI

BYTE (N°)	BYTE NAME	VALUE (hex.)	NAME	DESCRIPTION
1	COMMAND HEADER	0x01	CMD_SOH	Carattere di sincronizzazione inizio trasmissione tra il dispositivo master e lo slave.
2	SLAVE ADDRESS	0x00		Tutti gli slave che ricevono il comando, lo devono eseguire senza inviare la risposta.
		0x7F		Tutti gli slave che ricevono il comando, lo devono eseguire ed inviare la risposta.
		0x01 : 0x7E		Lo slave indirizzato deve eseguire il comando ed inviare una risposta.
3	MASTER ADDRESS	0x01 : 0x7E		Indirizzo del dispositivo master a cui lo slave deve rispondere se richiesto.
4	COMMAND	0x61	CMD_INQUIRY	Richiesta di stato del dispositivo slave selezionato. Il dispositivo ritorna una stringa contenente identificativo ed esito dell'ultimo comando eseguito, insieme a data ed ora dell'esecuzione.
		0x62	CMD_RESET	Reset a caldo del dispositivo slave indirizzato. Il dispositivo interrompe tutte le attività e rilancia l'esecuzione del suo programma da capo.
		0x63	CMD_VERSION	Richiesta della versione dell'hardware e del firmware del dispositivo slave indirizzato. Il dispositivo invia una stringa alfanumerica indicante la sua versione di hardware e firmware.
		0x64	CMD_SAVE	Il dispositivo slave deve salvare i dati di configurazione nella memoria non volatile.
		0x65	CMD_RESTORE	Il dispositivo slave deve ripristinare i dati di configurazione presenti nella memoria non volatile ed eseguire di seguito un reset a caldo.
		0x66	CMD_GET_ADDR	Il dispositivo slave deve ritornare il suo indirizzo di protocollo.
		0x68	CMD_GET_TIME	Il dispositivo slave deve ritornare la data e l'orario del suo orologio interno.
		0x6A	CMD_GET_FRAME	Il dispositivo slave deve ritornare la lunghezza massima del frame di comunicazione, che dipende sostanzialmente dalla dimensione del suo buffer di comunicazione. Questo valore limita il master nella dimensione dei parametri dei comandi.
5	ID	0x00 : 0x7F		Questo è un numero progressivo inviato al dispositivo slave e che lo slave ritorna quando invia la risposta relativa all'esito del processo.



5.1.2. TABELLA DELLE RISPOSTE AI COMANDI ABBREVIATI SENZA PARAMETRI

BYTE (N°)	BYTE NAME	VALUE (hex.)	NAME	DESCRIPTION
1	ANSWER HEADER	0x02	CMD_STX	Carattere di sincronizzazione inizio trasmissione tra il dispositivo slave ed il master.
2	MASTER ADDRESS	0x01 : 0x7E		Indirizzo del dispositivo master a cui tornare la risposta.
3	SLAVE ADDRESS	0x01 : 0x7E		Indirizzo del dispositivo slave che ha eseguito il comando.
4	COMMAND	0x61 0x62 0x63 0x64 0x65 0x66 0x68 0x6A	CMD_INQUIRY CMD_RESET CMD_VERSION CMD_SAVE CMD_RESTORE CMD_GET_ADDR CMD_GET_TIME CMD_GET_FRAME	Il dispositivo slave replica il comando ricevuto dal master.
5	ID	0x00 : 0x7F		Il dispositivo slave ritorna lo stesso campo ID trovato nel comando eseguito.
6	EXECUTION RESULT	0x00	CMD_ACK	Comando eseguito dallo slave senza errori.
		0x01	CMD_ERR_CMD	Comando non previsto dallo slave selezionato.
		0x02	CMD_ERR_CHKS	Errore nel checksum del comando ricevuto.
		0x03	CMD_ERR_FORM	Errore nel formato dei parametri ricevuti.
		0x04	CMD_ERR_DATA	Errore nei parametri del comando ricevuto.
		0x05	CMD_ERR_TOUT	Scaduto il timeout di esecuzione del comando
		0x06	CMD_ERR_ADDR	Indirizzo errato.
		0x07	CMD_ERR_TIME	Orario errato.
		0x08	CMD_ERR_FRAME_SIZE	Errore nella dimensione del frame, troppo grande per il dispositivo.
		0x09	CMD_ERR_DATA_TYPE	Tipo di dati non gestito.
		0x0A	CMD_ERR_PORT_TYPE	Il tipo di porta non esistente
		0x0B	CMD_ERR_PORT_NUMBER	Il numero di porta non esistente
		0x0C	CMD_ERR_DATA_SIZE	Errore nella dimensione dei dati.
		0x10 : 0x7F		Errore dello slave nell'esecuzione del comando. Il valore ritornato dipende dal tipo di comando e dal firmware del dispositivo slave.
7	DATA SIZE	0x00 : 0x7F		Dimensione in byte dei dati che seguono. N.B: questo byte è presente se richiesto dal tipo di comando e dall'esito positivo della sua esecuzione.
8 to N	DATA	0x00 : 0x7F		Pacchetto dei dati richiesti dal master, la cui lunghezza dipende dal comando. N.B: questi bytes sono presenti se richiesto dal tipo di comando e dall'esito positivo della sua esecuzione.



5.2. COMANDI ABBREVIATI CON PARAMETRI

I *comandi abbreviati con parametri*, sono costituiti da stringhe di lunghezza variabile suddivise in due campi.

Il campo iniziale contiene cinque caratteri:

- COMMAND HEADER - carattere di sincronizzazione (inizio trasmissione del comando).
- SLAVE ADDRESS - indirizzo del dispositivo di destinazione del comando.
- MASTER ADDRESS - indirizzo del dispositivo che invia il comando.
- COMMAND - identificativo del comando.
- ID - numero progressivo del comando.

Il campo finale è costituito da almeno un carattere che identifica il numero dei seguenti:

- PARAMETER SIZE - identifica il numero dei caratteri che seguono nel campo (da 1 a 126max.).
- PARAMETER - caratteri che costituiscono i parametri del comando (max.126 caratteri).

La stringa del comando ha dunque il seguente formato.

COMMAND FORMAT						
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTES N° 7 to N
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID	PARAMETER SIZE	PARAMETER

Le risposte ai *comandi abbreviati con parametri*, sono costituite da stringhe di lunghezza variabile suddivise in uno o due campi.

Il campo iniziale contiene sei caratteri:

- ANSWER HEADER - carattere di sincronizzazione (inizio trasmissione della risposta).
- MASTER ADDRESS - indirizzo del dispositivo di destinazione della risposta.
- SLAVE ADDRESS - indirizzo del dispositivo che invia la risposta.
- COMMAND - identificativo del comando di pertinenza della risposta.
- EXECUTION RESULT - esito dell'esecuzione del comando.
- ID - numero progressivo del comando.

Il campo finale è assente nel caso in cui il comando non sia stato riconosciuto od eseguito correttamente. Se presente, perché previsto dal tipo di comando, è costituito da almeno un carattere che identifica il numero dei seguenti:

- DATA SIZE - identifica il numero dei caratteri che seguono nel campo (da 0 a 126max.).
- DATA - caratteri che costituiscono i dati della risposta (max.126 caratteri).



La stringa della risposta, nel caso in cui non vi siano dati da ritornare oppure ci siano errori nell'esecuzione, ha il seguente formato.

ANSWER FORMAT					
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT

Quando la dimensione dei dati richiesti dal comando risulta essere nulla (DATA SIZE = 0), ovvero i dati non sono disponibili, la risposta ha il seguente formato.

ANSWER FORMAT						
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	DATA SIZE

Quando la dimensione dei dati richiesti dal comando non è nulla, la risposta ha il seguente formato.

ANSWER FORMAT							
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTES N° 8 to N
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	DATA SIZE	DATA

Esempio: se DATA SIZE = 10, DATA è di 10 bytes e l'intera trasmissione è lunga 17 bytes.



5.2.1. TABELLA DEI COMANDI ABBREVIATI CON PARAMETRI

BYTE (N°)	BYTE NAME	VALUE (hex.)	NAME	DESCRIPTION
1	COMMAND HEADER	0x01	CMD_SOH	Carattere di sincronizzazione inizio trasmissione tra il dispositivo master e lo slave.
2	SLAVE ADDRESS	0x00		Tutti gli slave che ricevono il comando, lo devono eseguire senza inviare la risposta.
		0x7F		Tutti gli slave che ricevono il comando, lo devono eseguire ed inviare la risposta.
		0x01 : 0x7E		Lo slave indirizzato deve eseguire il comando ed inviare una risposta.
3	MASTER ADDRESS	0x01 : 0x7E		Indirizzo del dispositivo master a cui lo slave deve rispondere, se richiesto.
4	COMMAND	0x67	CMD_SET_ADDR	Il dispositivo slave deve aggiornare il suo indirizzo di protocollo con il nuovo valore comunicato dal master. Vedi (*).
		0x69	CMD_SET_TIME	Il dispositivo slave deve aggiornare data e ora del suo orologio interno con i nuovi valori comunicati dal master.
		0x6B	CMD_SET_FRAME	
		0x6C	CMD_GET_PORT	Il dispositivo slave deve ritornare la configurazione della porta specificata dal comando, dipendentemente dalla sua versione di firmware ed hardware.
		0x6D	CMD_SET_PORT	Il dispositivo slave deve configurare la porta specificata dal comando, dipendentemente dalla sua versione di firmware ed hardware. Vedi (*).
		0x6E	CMD_GET_DATA	Il dispositivo slave deve ritornare i dati presenti sul canale della porta specificata dal comando, dipendentemente dalla sua versione di firmware ed hardware.
		0x6F	CMD_SET_DATA	Il dispositivo slave deve inviare i dati sul canale della porta specificata dal comando, dipendentemente dalla sua versione di firmware ed hardware.
5	ID	0x00 : 0x7F		Questo è un numero progressivo inviato al dispositivo slave e che lo slave ritorna quando invia la risposta relativa all'esito del processo.
6	PARAMETER SIZE	0x01 : 0x7F		Dimensione in byte dei parametri che seguono.
7 to N	PARAMETER	0x00 : 0x7F		Pacchetto dei parametri previsti dal comando.

(*) N.B.: Senza un successivo comando CMD_SAVE_CONF, l'impostazione viene perduta in caso di reset a caldo od una mancanza di alimentazione allo slave.



5.2.2. TABELLA DELLE RISPOSTE AI COMANDI ABBREVIATI CON PARAMETRI

BYTE (N°)	BYTE NAME	VALUE (hex.)	NAME	DESCRIPTION
1	ANSWER HEADER	0x02	CMD_STX	Carattere di sincronizzazione inizio trasmissione tra il dispositivo slave ed il master.
2	MASTER ADDRESS	0x01 : 0x7E		Indirizzo del dispositivo master a cui tornare la risposta.
3	SLAVE ADDRESS	0x01 : 0x7E		Indirizzo del dispositivo slave che ha eseguito il comando.
4	COMMAND	0x67 0x69 0x6B 0x6C 0x6D 0x6E 0x6F	CMD_SET_ADDR CMD_SET_TIME CMD_SET_FRAME CMD_GET_PORT CMD_SET_PORT CMD_GET_DATA CMD_SET_DATA	Il dispositivo slave replica il comando ricevuto dal master.
5	ID	0x00 : 0x7F		Il dispositivo slave ritorna lo stesso campo ID trovato nel comando eseguito.
6	EXECUTION RESULT	0x00	CMD_ACK	Comando eseguito dallo slave senza errori.
		0x01	CMD_ERR_CMD	Comando non previsto dallo slave selezionato.
		0x02	CMD_ERR_CHKS	Errore nel checksum del comando ricevuto.
		0x03	CMD_ERR_FORM	Errore nel formato dei parametri ricevuti.
		0x04	CMD_ERR_DATA	Errore nei parametri del comando ricevuto.
		0x05	CMD_ERR_TOUT	Scaduto il timeout di esecuzione del comando
		0x06	CMD_ERR_ADDR	Indirizzo errato.
		0x07	CMD_ERR_TIME	Orario errato.
		0x08	CMD_ERR_FRAME_SIZE	Errore nella dimensione del frame, troppo grande per il dispositivo.
		0x09	CMD_ERR_DATA_TYPE	Tipo di dati non gestito.
		0x0A	CMD_ERR_PORT_TYPE	Il tipo di porta non esistente
		0x0B	CMD_ERR_PORT_NUMBER	Il numero di porta non esistente
		0x0C	CMD_ERR_DATA_SIZE	Errore nella dimensione dei dati.
		0x10 : 0x7F		Errore dello slave nell'esecuzione del comando. Il valore ritornato dipende dal tipo di comando e dal firmware del dispositivo slave.
7	DATA SIZE	0x00 : 0x7F		Dimensione in byte dei dati che seguono. N.B: questo byte è presente se richiesto dal tipo di comando e dall'esito positivo della sua esecuzione.
8 to N	DATA	0x00 : 0x7F		Pacchetto dei dati richiesti dal master, la cui lunghezza dipende dal comando. N.B: questi bytes sono presenti se richiesto dal tipo di comando e dall'esito positivo della sua esecuzione.



6. TRASMISSIONI DI GRANDI DIMENSIONI

Attraverso i comandi **CMD_GET_DATA** e **CMD_SET_DATA** è possibile richiedere o dirigere i pacchetti di dati verso porte o risorse identificabili mediante alcuni caratteri (*port type* e *port number*) dei campi **PARAMETER** o **DATA**. Quando le dimensioni dei dati sono superiori alle risorse dei buffer di comunicazione presenti sui dispositivi coinvolti nello scambio, è possibile frammentare la trasmissione in più pacchetti detti *frames* in cui il primo riporta il numero complessivo. Mediante i comandi **CMD_GET_FRAME** e **CMD_SET_FRAME** è possibile conoscere o limitare la dimensione massima del pacchetto singolo; la ricostruzione della trasmissione complessiva è a carico del destinatario, mentre la frammentazione deve essere eseguita dal mittente.

Supponendo di avere un frame limitato a 5 caratteri, l'invio di un pacchetto di dati da 20 caratteri si svolge nel seguente modo:

1 - Richiesta allo slave della sua dimensione di frame

master → *slave* "0x01,0x02,0x01,0x6A,0x00" (**CMD_GET_FRAME**)
slave → *master* "0x02,0x01,0x02,0x6A,0x00,0x00,0x01,0x05" (*il frame del dispositivo è di 5 caratteri*)

2 - Invio del primo pacchetto contenente la dimensione totale

master → *slave* "0x01,0x02,0x01,0x6F,0x00,0x7F, 0x00,0x00,0x00, 0x00,0x00,0x00,0x14" (**CMD_SET_DATA**)

Apertura: "0x01" (*inizio trasmissione di comando **CMD_SOH***)
Indirizzo slave: "0x02"
Indirizzo master: "0x01"
Comando: "0x6F" (*comando **CMD_SET_DATA** abbreviato*)
ID comando: "0x00" (*numero progressivo del comando*)
Parameter size: "0x7F" (*indica che la trasmissione sarà più lunga del limite consentito*)
Parameters:
 Data type: "0x00" (**CMD_STRING** - richiesta dati di tipo *STRING*)
 Port type: "0x00" (*tipo di porta selezionata, valore 0x00:0x7F*)
 Port number: "0x00" (*numero di porta selezionata, valore 0x00:0x7F*)
 Total size: "0x00,0x00,x00,0x14" (*parola di 32bit indicante la dimensione totale della trasmissione, in questo caso 0x00000014 = 20 decimale*)

Il dispositivo slave risponde in base alla sua capacità di memoria, se è in grado di eseguire l'operazione risponde:

slave → *master* "0x02,0x01,0x02, 0x6F,0x00,0x00" (**CMD_ACK** comando eseguibile)

Se non è in grado risponde:

slave → *master* "0x02,0x01,0x02, 0x6F,0x00,0x01" (**CMD_ERR_CMD** comando non eseguibile)

3 - Invio dei rimanenti pacchetti di dati

Il dispositivo master invia quattro pacchetti di 5 byte (dimensione del frame) attendendo tra un invio e l'altro la risposta dello slave.

master → *slave* "0x01,0x02,0x03,0x04,0x05" (*primo pacchetto di dati da 5 byte*)
slave → *master* "0x02,0x01,0x02, 0x6F,0x00,0x00" (**CMD_ACK** comando eseguito)

master → *slave* "0x06,0x07,0x08,0x09,0x0A" (*secondo pacchetto di dati da 5 byte*)
slave → *master* "0x02,0x01,0x02, 0x6F,0x00,0x00" (**CMD_ACK** comando eseguito)

master → *slave* "0x01,0x02,0x03,0x04,0x05" (*terzo pacchetto di dati da 5 byte*)
slave → *master* "0x02,0x01,0x02, 0x6F,0x00,0x00" (**CMD_ACK** comando eseguito)

master → *slave* "0x01,0x02,0x03,0x04,0x05" (*quarto pacchetto di dati da 5 byte*)
slave → *master* "0x02,0x01,0x02, 0x6F,0x00,0x00" (**CMD_ACK** comando eseguito)



CMD_INQUIRY

Il comando di INQUIRY, chiede allo *slave* di ritornare identificativo ed esito dell'ultimo comando eseguito, insieme a data ed ora dell'esecuzione. Può essere utilizzato, per esempio, dopo un comando indirizzato a tutti gli *slave* che non preveda la risposta (indirizzo = 0x00 hex.), al fine di conoscere l'esito di uno *slave* specifico.

Se come identificativo dell'ultimo comando eseguito, il dispositivo torna zero (0x00 hex.), significa che non ha ancora eseguito operazione dal momento dell'accensione. L'esecuzione di un comando INQUIRY, non viene memorizzata dai dispositivi, quindi CMD_INQUIRY non compare mai come identificativo dell'ultimo comando eseguito.

COMANDO ESTESO

COMMAND FORMAT						
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID	CHECKSUM	COMMAND ENDING

ANSWER FORMAT									
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTE N°8 to 18	BYTE N°19	BYTE N°20
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	DATA SIZE	DATA	CHECK SUM	ANSWER ENDING

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x41,0x00,0x43,0x04
slave → *master* 0x02,0x01,0x02,0x41,0x00,0x00,0x0B,
 0x63,0x00,0x00,0x14,0x02,0x0C,0x10,0x11,0x37,0x00,0x00,0x04,0x03

Comando: 0x63 (*CMD_GET_TIME* abbreviato)
 Id: 0x00
 Esito: 0x00 (*CMD_ACK*, tutto ok)
 Data: 0x14,0x02,0x0C,0x10 (2002/12/16)
 Ora: 0x11,0x37,0x00,0x00 (17:55:00,0)

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.
CMD_ERR_CHKS	0x02	Errore nella trasmissione del comando, il checksum non torna.

COMANDO ABBREVIATO

COMMAND FORMAT				
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID

ANSWER FORMAT							
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTES N°8 to 18
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	DATA SIZE	DATA

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x61,0x00
slave → *master* 0x02,0x01,0x02,0x61,0x00,0x00,0x0B,0x63,0x00,0x00,0x14,0x02,0x0C,0x10,0x11,0x37,0x00,0x00"

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.



CMD_RESET

Il comando di RESET, forza lo *slave* a terminare ogni attività che ha in coda d'esecuzione ed a ricominciare da capo l'esecuzione del suo programma applicativo; questo equivale a riportare il dispositivo nella stessa condizione iniziale, quando gli viene applicata l'alimentazione.

COMANDO ESTESO

COMMAND FORMAT						
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID	CHECKSUM	COMMAND ENDING

ANSWER FORMAT							
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTE N°8
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	CHECKSUM	ANSWER ENDING

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x42,0x00,0x40,0x04
slave → *master* 0x02,0x01,0x02,0x42,0x00,0x00,0x43,0x03

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.
CMD_ERR_CHKS	0x02	Errore nella trasmissione del comando, il checksum non torna.

COMANDO ABBREVIATO

COMMAND FORMAT				
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID

ANSWER FORMAT					
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x62,0x00
slave → *master* 0x02,0x01,0x02,0x62,0x00,0x00

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.

NOTA: ulteriori valori di errore in esecuzione, possono essere definiti nello sviluppo di firmware specifici.



CMD_VERSION

Il comando VERSION, chiede allo *slave* di ritornare il codice identificativo. Il codice ritornato dipende dall'hardware del dispositivo *slave* e dall'applicativo su di esso sviluppato. In generale è composto da tre sezioni riportanti un identificativo della scheda (primi quattro caratteri), la versione del firmware (i due caratteri seguenti) e la relativa revisione (ultimi due caratteri).

COMANDO ESTESO

COMMAND FORMAT						
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID	CHECKSUM	COMMAND ENDING

ANSWER FORMAT									
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTE N°8 to N	BYTE N°N+1	BYTE N°N+2
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	DATA SIZE	DATA	CHECK SUM	ANSWER ENDING

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x43,0x00,0x41,0x04
slave → *master* 0x02,0x01,0x02,0x43,0x00,0x08,0x30,0x30,0x32,0x30,0x30,0x32,0x30,0x31,0x4B,0x03

Board ID: 0x30,0x30,0x32,0x30 (“0020” - C.N.R.-I.R.A. Display ad keyboard Interface)
Firmware ver.: 0x30,0x32 (“02” - Strumento di allineamento pannelli per superficie attiva)
Firmware rev.: 0x30,0x31 (“01” - Prima emissione)

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.
CMD_ERR_CHKS	0x02	Errore nella trasmissione del comando, il checksum non torna.

COMANDO ABBREVIATO

COMMAND FORMAT				
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID

ANSWER FORMAT							
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTES N°8 to N
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	DATA SIZE	DATA

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x63,0x00
slave → *master* 0x02,0x01,0x02,0x63,0x00,0x08,0x30,0x30,0x32,0x30,0x30,0x32,0x30,0x31

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.



CMD_SAVE

Il comando di SAVE, permette di salvare su memoria non volatile i parametri di configurazione dello *slave*; questi dipendono dall'hardware del dispositivo e dall'applicativo su di esso sviluppato. I dati di configurazione sono le impostazioni di default, ovvero quelle caricate al momento dell'alimentazione della scheda.

COMANDO ESTESO

COMMAND FORMAT						
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID	CHECKSUM	COMMAND ENDING

ANSWER FORMAT							
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTE N°8
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	CHECKSUM	ANSWER ENDING

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x44,0x00,0x46,0x04
slave → *master* 0x02,0x01,0x02,0x44,0x00,0x00,0x45,0x03

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.
CMD_ERR_CHKS	0x02	Errore nella trasmissione del comando, il checksum non torna.
	0x10	Errore nella memoria (difettosa o non presente)
	0x11	Errore nel salvataggio dei dati (non torna il checksum).

COMANDO ABBREVIATO

COMMAND FORMAT				
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID

ANSWER FORMAT					
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x64,0x00
slave → *master* 0x02,0x01,0x02,0x64,0x00,0x00

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.
	0x10	Errore nella memoria (difettosa o non presente)
	0x11	Errore nel salvataggio dei dati (non torna il checksum).

NOTA: ulteriori valori di errore in esecuzione, possono essere definiti nello sviluppo di firmware specifici.



CMD_RESTORE

Il comando di RESTORE, permette di recuperare dalla memoria non volatile, i parametri di configurazione dello *slave*; questi dipendono dall'hardware del dispositivo e dall'applicativo su di esso sviluppato. Dopo il recupero della configurazione, il dispositivo *slave* esegue un reset a caldo in modo da rendere operativa la nuova configurazione, che utilizzerà tutte le volte che verrà alimentato.

COMANDO ESTESO

COMMAND FORMAT						
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID	CHECKSUM	COMMAND ENDING

ANSWER FORMAT							
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTE N°8
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	CHECKSUM	ANSWER ENDING

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x45,0x00,0x47,0x04
slave → *master* 0x02,0x01,0x02,0x45,0x00,0x00,0x44,0x03

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.
CMD_ERR_CHKS	0x02	Errore nella trasmissione del comando, il checksum non torna.
	0x10	Errore nella memoria (difettosa o non presente)
	0x11	Errore nel recupero dei dati (non torna il checksum).

COMANDO ABBREVIATO

COMMAND FORMAT				
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID

ANSWER FORMAT					
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x65,0x00
slave → *master* 0x02,0x01,0x02,0x65,0x00,0x00

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.
	0x10	Errore nella memoria (difettosa o non presente)
	0x11	Errore nel recupero dei dati (non torna il checksum).

NOTA: ulteriori valori di errore in esecuzione, possono essere definiti nello sviluppo di firmware specifici.



CMD_SET_ADDR

Il comando SET ADDRESS, attraverso il carattere DATA, comunica il nuovo indirizzo di protocollo che lo *slave* deve utilizzare; la risposta al comando viene data col vecchio indirizzo, dopodiché il dispositivo risponderà solo ai comandi inviati al nuovo indirizzo. L'impostazione viene persa se viene a mancare l'alimentazione allo *slave*; per salvarla in modo non volatile, è necessario inviare allo *slave* (al suo nuovo indirizzo) un comando SAVE.

COMANDO ESTESO

COMMAND FORMAT								
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTE N°8	BYTE N°9
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID	PARAMETER SIZE	PARAMETER	CHECKSUM	COMMAND ENDING

ANSWER FORMAT							
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTE N°8
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	CHECKSUM	ANSWER ENDING

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x47,0x00,0x01,0x03,0x47,0x04
slave → *master* 0x02,0x01,0x02,0x47,0x00,0x00,0x46,0x03

New address: 0x03 (indirizzi possibili 0x01:0x7E)

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.
CMD_ERR_CHK	0x02	Errore nella trasmissione del comando, il checksum non torna.
CMD_ERR_FORM	0x03	Errore nel formato dei parametri ricevuti.
CMD_ERR_DATA	0x04	Errore nei parametri del comando ricevuto.

COMANDO ABBREVIATO

COMMAND FORMAT						
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTES N°7
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID	PARAMETER SIZE	PARAMETER

ANSWER FORMAT					
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x67,0x00,0x01,0x03
slave → *master* 0x02,0x01,0x02,0x67,0x00,0x00

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.
CMD_ERR_FORM	0x03	Errore nel formato dei parametri ricevuti.
CMD_ERR_DATA	0x04	Errore nei parametri del comando ricevuto.



CMD_GET_TIME

Il comando GET TIME, chiede allo *slave* di ritornare la data e l'orario del suo orologio interno. I caratteri del campo DATA, indicano in sequenza il secolo, l'anno, il mese, il giorno, l'ora, i minuti, i secondi ed i centesimi.

COMANDO ESTESO

COMMAND FORMAT						
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID	CHECKSUM	COMMAND ENDING

ANSWER FORMAT									
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTE N°8 to 15	BYTE N°16	BYTE N°17
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	DATA SIZE	DATA	CHECK SUM	ANSWER ENDING

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x48,0x00,0x4A,0x04

slave → *master* 0x02,0x01,0x02,0x48,0x00,0x00,0x08,0x14,0x02,0x0C,0x10,0x11,0x37,0x00,0x00,0x6D,0x03

Data: 0x14,0x02,0x0C,0x10 (2002/12/16)
Time: 0x11,0x37,0x00,0x00 (17:55:00,00)

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.
CMD_ERR_CHKS	0x02	Errore nella trasmissione del comando, il checksum non torna.

COMANDO ABBREVIATO

COMMAND FORMAT				
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID

ANSWER FORMAT							
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTES N° 8 to 15
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	DATA SIZE	DATA

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x68,0x00

slave → *master* 0x02,0x01,0x02,0x68,0x00,0x00,0x08,0x14,0x02,0x0C,0x10,0x11,0x37,0x00,0x00

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.



CMD_SET_TIME

Il comando SET TIME, chiede allo *slave* di ritornare l'orario e la data del suo orologio interno. I caratteri del campo DATA, indicano in sequenza il secolo, l'anno, il mese, il giorno, l'ora, i minuti, i secondi ed i centesimi.

COMANDO ESTESO

COMMAND FORMAT								
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTES N° 7 to 14	BYTE N°15	BYTE N°16
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID	PARAMETER SIZE	PARAMETER	CHECKSUM	COMMAND ENDING

ANSWER FORMAT							
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTE N°8
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	CHECKSUM	ANSWER ENDING

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x49,0x00,0x08,0x14,0x02,0x0C,0x10,0x11,0x37,0x00,0x00,0x6F,0x04

slave → *master* 0x02,0x01,0x02,0x49,0x00,0x00,0x48,0x03

New data: 0x14,0x02,0x0C,0x10 (2002/12/16)

New time: 0x11,0x37,0x00,0x00 (17:55:00,00)

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.
CMD_ERR_CHKS	0x02	Errore nella trasmissione del comando, il checksum non torna.
CMD_ERR_FORM	0x03	Errore nel formato dei parametri ricevuti.
CMD_ERR_DATA	0x04	Errore nei parametri del comando ricevuto.

COMANDO ABBREVIATO

COMMAND FORMAT						
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTES N° 7 to 14
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID	PARAMETER SIZE	PARAMETER

ANSWER FORMAT					
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x69,0x00,0x08,0x14,0x02,0x0C,0x10,0x11,0x37,0x00,0x00

slave → *master* 0x02,0x01,0x02,0x69,0x00,0x00

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.
CMD_ERR_FORM	0x03	Errore nel formato dei parametri ricevuti.
CMD_ERR_DATA	0x04	Errore nei parametri del comando ricevuto.



CMD_GET_FRAME

Il comando GET FRAME, chiede allo *slave* di ritornare la dimensione massima del campo PARAMETER (per i comandi) e DATA (per le risposte); questa dimensione dipende dall'hardware del dispositivo e dall'applicativo su di esso sviluppato.

COMANDO ESTESO

COMMAND FORMAT						
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID	CHECKSUM	COMMAND ENDING

ANSWER FORMAT									
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTE N°8	BYTE N°9	BYTE N°10
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	DATA SIZE	DATA	CHECK SUM	ANSWER ENDING

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x4A,0x00,0x48,0x04
slave → *master* 0x02,0x01,0x02,0x4A,0x00,0x00,0x01,0x78,0x32,0x03

Frame size: "0x78" (120 bytes - il valori massimo consentito è 126)

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.
CMD_ERR_CHKS	0x02	Errore nella trasmissione del comando, il checksum non torna.

COMANDO ABBREVIATO

COMMAND FORMAT				
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID

ANSWER FORMAT							
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTES N°8
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	DATA SIZE	DATA

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x6A,0x00
slave → *master* 0x02,0x01,0x02,0x6A,0x00,0x00,0x01,0x78

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.



CMD_SET_FRAME

Il comando SET FRAME, chiede allo *slave* di limitare la dimensione massima del campo PARAMETER (per i comandi) e DATA (per le risposte); questa dimensione dipende dall'hardware del dispositivo e dall'applicativo su di esso sviluppato.

COMANDO ESTESO

COMMAND FORMAT								
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTE N°8	BYTE N°9
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID	PARAMETER SIZE	PARAMETER	CHECKSUM	COMMAND ENDING

ANSWER FORMAT							
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTE N°8
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	CHECKSUM	ANSWER ENDING

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x4B,0x00,0x01,0x78,0x30,0x04
slave → *master* 0x02,0x01,0x02,0x4B,0x00,0x00,0x4A,0x03

New frame size: 0x78 (120 bytes - il valori massimo consentito è 126)

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.
CMD_ERR_CHKS	0x02	Errore nella trasmissione del comando, il checksum non torna.
CMD_ERR_FORM	0x03	Errore nel formato dei parametri ricevuti.
CMD_ERR_DATA	0x04	Errore nei parametri del comando ricevuto.
CMD_ERR_FRAME_SIZE	0x08	Errore nella dimensione del frame, troppo grande.

COMANDO ABBREVIATO

COMMAND FORMAT						
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTES N°7
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID	PARAMETER SIZE	PARAMETER

ANSWER FORMAT					
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x6B,0x00,0x01,0x78
slave → *master* 0x02,0x01,0x02,0x6B,0x00,0x00

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.
CMD_ERR_FORM	0x03	Errore nel formato dei parametri ricevuti.
CMD_ERR_DATA	0x04	Errore nei parametri del comando ricevuto.
CMD_ERR_FRAME_SIZE	0x08	Errore nella dimensione del frame, troppo grande.



CMD_GET_PORT

Il comando GET PORT, chiede allo *slave* di ritornare l'impostazione della porta, ovvero i parametri di configurazione necessari a definirne il funzionamento; una porta parallela, ad esempio, necessita la configurazione della direzione (ingresso/uscita) dei vari bit che costituiscono la parola.

Il primo carattere del campo PARAMETER, indica il tipo di dato richiesto (sono ammessi quelli elencati nella [TABELLA DEI TIPI DI DATI](#)), il secondo carattere definisce il tipo di porta ed il terzo il numero della porta. Il tipo di dati, il tipo ed il numero di porta, dipendono dall'hardware del dispositivo e dall'applicativo su di esso sviluppato.

NOTA: se la dimensione dei campi PARAMETER (per i comandi) e DATA (per le risposte) è superiore al FRAME di comunicazione, la trasmissione deve essere spezzata in più blocchi come specificato al capitolo [TRASMISSIONI DI GRANDI DIMENSIONI](#).

COMANDO ESTESO

COMMAND FORMAT								
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N° 7 to N	BYTE N° N + 1	BYTE N° N + 2
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID	PARAMETER SIZE	PARAMETER	CHECKSUM	COMMAND ENDING

ANSWER FORMAT									
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTE N° 8 to N	BYTE N° N + 1	BYTE N° N + 2
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	DATA SIZE	DATA	CHECK SUM	ANSWER ENDING

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x4C,0x00,0x03,0x00,0x00,0x00,0x4D,0x04

Data type: 0x00 (*CMD_STRING* - richiesta dati di tipo *STRING*)
 Port type: 0x00 (tipo di porta selezionata, valore 0x00:0x7F)
 Port number: 0x00 (numero di porta selezionata, valore 0x00:0x7F)

slave → *master* 0x02,0x01,0x02,0x4C,0x00,0x00,0x01,0x78,0x34,0x03

Port setting: 0x78 (data è di tipo *STRING*, valore 0x00:0x7F)

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.
CMD_ERR_CHKS	0x02	Errore nel checksum del comando ricevuto.
CMD_ERR_FORM	0x03	Errore nel formato dei parametri ricevuti.
CMD_ERR_DATA	0x04	Errore nei parametri del comando ricevuto.
CMD_ERR_TOUT	0x05	Scaduto il timeout di esecuzione del comando ricevuto.
CMD_ERR_DATA_TYPE	0x09	Tipo di dati non accettato.
CMD_ERR_PORT_TYPE	0x0A	Tipo di porta non presente.
CMD_ERR_PORT_NUMBER	0x0B	Numero di porta non presente.
CMD_ERR_DATA_SIZE	0x0C	Errore nella dimensione dei dati.

NOTA: ulteriori valori di errore in esecuzione, possono essere definiti nello sviluppo di firmware specifici.



COMANDO ABBREVIATO

COMMAND FORMAT						
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTES N°7 to N
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID	PARAMETER SIZE	PARAMETER

ANSWER FORMAT							
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTES N° 8 to N
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	DATA SIZE	DATA

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x6C,0x00,0x03,0x00,0x00,0x00
slave → *master* 0x02,0x01,0x02,0x6C,0x00,0x00,0x01,0x78

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.
CMD_ERR_FORM	0x03	Errore nel formato dei parametri ricevuti.
CMD_ERR_DATA	0x04	Errore nei parametri del comando ricevuto.
CMD_ERR_TOUT	0x05	Scaduto il timeout di esecuzione del comando ricevuto.
CMD_ERR_DATA_TYPE	0x09	Tipo di dati non accettato.
CMD_ERR_PORT_TYPE	0x0A	Tipo di porta non presente.
CMD_ERR_PORT_NUMBER	0x0B	Numero di porta non presente.
CMD_ERR_DATA_SIZE	0x0C	Errore nella dimensione dei dati.

NOTA: ulteriori valori di errore in esecuzione, possono essere definiti nello sviluppo di firmware specifici.



CMD_SET_PORT

Il comando SET PORT, chiede allo *slave* di modificare l'impostazione della porta, ovvero i parametri di configurazione necessari a definirne il funzionamento; una porta parallela, ad esempio, necessita la configurazione della direzione (ingresso/uscita) dei vari bit che costituiscono la parola.

Il primo carattere del campo PARAMETER, indica il tipo di dato inviato (sono ammessi quelli elencati nella [TABELLA DEI TIPI DI DATI](#)), il secondo carattere definisce il tipo di porta ed il terzo il numero della porta. Il tipo di dati, il tipo ed il numero di porta, dipendono dall'hardware del dispositivo e dall'applicativo su di esso sviluppato.

NOTA: se la dimensione del campo PARAMETER è superiore al FRAME di comunicazione, la trasmissione deve essere spezzata in più blocchi come specificato al capitolo [TRASMISSIONI DI GRANDI DIMENSIONI](#).

COMANDO ESTESO

COMMAND FORMAT								
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7 to N	BYTE N°N + 1	BYTE N°N + 2
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID	PARAMETER SIZE	PARAMETER	CHECKSUM	COMMAND ENDING

ANSWER FORMAT							
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTE N°8
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	CHECKSUM	ANSWER ENDING

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x4D,0x00,0x04,0x00,0x00,0x00,0x0F,0x44,0x04
slave → *master* 0x02,0x01,0x02,0x4D,0x00,0x00,0x4C,0x03

Data type: 0x00 (*CMD_STRING* - richiesta dati di tipo *STRING*)
Port type: 0x00 (*tipo di porta selezionata, valore 0x00:0x7F*)
Port number: 0x00 (*numero di porta selezionata, valore 0x00:0x7F*)
Port setting: 0x0F (*impostazione della porta selezionata*)

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.
CMD_ERR_CHKS	0x02	Errore nel checksum del comando ricevuto.
CMD_ERR_FORM	0x03	Errore nel formato dei parametri ricevuti.
CMD_ERR_DATA	0x04	Errore nei parametri del comando ricevuto.
CMD_ERR_TOUT	0x05	Scaduto il timeout di esecuzione del comando ricevuto.
CMD_ERR_DATA_TYPE	0x09	Tipo di dati non accettato.
CMD_ERR_PORT_TYPE	0x0A	Tipo di porta non presente.
CMD_ERR_PORT_NUMBER	0x0B	Numero di porta non presente.
CMD_ERR_DATA_SIZE	0x0C	Errore nella dimensione dei dati.

NOTA: ulteriori valori di errore in esecuzione, possono essere definiti nello sviluppo di firmware specifici.



COMANDO ABBREVIATO

COMMAND FORMAT						
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTES N°7 to N
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID	PARAMETER SIZE	PARAMETER

ANSWER FORMAT					
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT

Esempio di comando e risposta:

master → *slave* *0x01,0x02,0x01,0x6D,0x00,0x04,0x00,0x00,0x00,0x00,0x0F*
slave → *master* *0x02,0x01,0x02,0x6D,0x00,0x00*

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.
CMD_ERR_FORM	0x03	Errore nel formato dei parametri ricevuti.
CMD_ERR_DATA	0x04	Errore nei parametri del comando ricevuto.
CMD_ERR_TOUT	0x05	Scaduto il timeout di esecuzione del comando ricevuto.
CMD_ERR_DATA_TYPE	0x09	Tipo di dati non accettato.
CMD_ERR_PORT_TYPE	0x0A	Tipo di porta non presente.
CMD_ERR_PORT_NUMBER	0x0B	Numero di porta non presente.
CMD_ERR_DATA_SIZE	0x0C	Errore nella dimensione dei dati.

NOTA: ulteriori valori di errore in esecuzione, possono essere definiti nello sviluppo di firmware specifici.



CMD_GET_DATA

Il comando GET DATA, chiede allo *slave* di ritornare i dati presenti sulla porta; per una porta parallela, ad esempio, ritorna lo stato dei vari bit che costituiscono la parola.

Il primo carattere del campo PARAMETER, indica il tipo di dato richiesto (sono ammessi quelli elencati nella [TABELLA DEI TIPI DI DATI](#)), il secondo carattere definisce il tipo di porta ed il terzo il numero della porta. Il tipo di dati, il tipo ed il numero di porta, dipendono dall'hardware del dispositivo e dall'applicativo su di esso sviluppato.

NOTA: se la dimensione dei campi PARAMETER (per i comandi) e DATA (per le risposte) è superiore al FRAME di comunicazione, la trasmissione deve essere spezzata in più blocchi come specificato al capitolo [TRASMISSIONI DI GRANDI DIMENSIONI](#).

COMANDO ESTESO

COMMAND FORMAT								
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N° 7 to N	BYTE N° N + 1	BYTE N° N + 2
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID	PARAMETER SIZE	PARAMETER	CHECKSUM	COMMAND ENDING

ANSWER FORMAT									
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTE N° 8 to N	BYTE N° N + 1	BYTE N° N + 2
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	DATA SIZE	DATA	CHECK SUM	ANSWER ENDING

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x4E,0x00,0x03,0x00,0x00,0x00,0x4F,0x04

Data type: 0x00 (*CMD_STRING* - richiesta dati di tipo *STRING*)

Port type: 0x00 (tipo di porta selezionata, valore 0x00:0x7F)

Port number: 0x00 (numero di porta selezionata, valore 0x00:0x7F)

slave → *master* 0x02,0x01,0x02,0x4E,0x00,0x00,0x01,0x78,0x36,0x03

Data on port: 0x78 (data è di tipo *STRING*, valore 0x00:0x7F)

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.
CMD_ERR_CHKS	0x02	Errore nel checksum del comando ricevuto.
CMD_ERR_FORM	0x03	Errore nel formato dei parametri ricevuti.
CMD_ERR_DATA	0x04	Errore nei parametri del comando ricevuto.
CMD_ERR_TOUT	0x05	Scaduto il timeout di esecuzione del comando ricevuto.
CMD_ERR_DATA_TYPE	0x09	Tipo di dati non accettato.
CMD_ERR_PORT_TYPE	0x0A	Tipo di porta non presente.
CMD_ERR_PORT_NUMBER	0x0B	Numero di porta non presente.
CMD_ERR_DATA_SIZE	0x0C	Errore nella dimensione dei dati.

NOTA: ulteriori valori di errore in esecuzione, possono essere definiti nello sviluppo di firmware specifici.



COMANDO ABBREVIATO

COMMAND FORMAT						
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTES N°7 to N
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID	PARAMETER SIZE	PARAMETER

ANSWER FORMAT							
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTES N°8 to N
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	DATA SIZE	DATA

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x6E,0x00,0x03,0x00,0x00,0x00
slave → *master* 0x02,0x01,0x02,0x6E,0x00,0x00,0x01,0x78

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.
CMD_ERR_FORM	0x03	Errore nel formato dei parametri ricevuti.
CMD_ERR_DATA	0x04	Errore nei parametri del comando ricevuto.
CMD_ERR_TOUT	0x05	Scaduto il timeout di esecuzione del comando ricevuto.
CMD_ERR_DATA_TYPE	0x09	Tipo di dati non accettato.
CMD_ERR_PORT_TYPE	0x0A	Tipo di porta non presente.
CMD_ERR_PORT_NUMBER	0x0B	Numero di porta non presente.
CMD_ERR_DATA_SIZE	0x0C	Errore nella dimensione dei dati.

NOTA: ulteriori valori di errore in esecuzione, possono essere definiti nello sviluppo di firmware specifici.



CMD_SET_DATA

Il comando SET DATA, chiede allo *slave* di modificare i dati presenti sulla porta; per una porta parallela, ad esempio, imposta lo stato dei vari bit che costituiscono la parola.

Il primo carattere del campo PARAMETER, indica il tipo di dato inviato (sono ammessi quelli elencati nella [TABELLA DEI TIPI DI DATI](#)), il secondo carattere definisce il tipo di porta ed il terzo il numero della porta. Il tipo di dati, il tipo ed il numero di porta, dipendono dall'hardware del dispositivo e dall'applicativo su di esso sviluppato.

NOTA: se la dimensione del campo PARAMETER è superiore al FRAME di comunicazione, la trasmissione deve essere spezzata in più blocchi come specificato al capitolo [TRASMISSIONI DI GRANDI DIMENSIONI](#).

COMANDO ESTESO

COMMAND FORMAT								
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7 to N	BYTE N°N + 1	BYTE N°N + 2
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID	PARAMETER SIZE	PARAMETER	CHECKSUM	COMMAND ENDING

ANSWER FORMAT							
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTE N°7	BYTE N°8
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT	CHECKSUM	ANSWER ENDING

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x4F,0x00,0x04,0x00,0x00,0x00,0x0F,0x46,0x04
slave → *master* 0x02,0x01,0x02,0x4F,0x00,0x00,0x4E,0x03

Data type: 0x00 (*CMD_STRING* - richiesta dati di tipo *STRING*)
Port type: 0x00 (*tipo di porta selezionata, valore 0x00:0x7F*)
Port number: 0x00 (*numero di porta selezionata, valore 0x00:0x7F*)
Data on port: 0x0F (*impostazione della porta selezionata*)

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.
CMD_ERR_CHKS	0x02	Errore nel checksum del comando ricevuto.
CMD_ERR_FORM	0x03	Errore nel formato dei parametri ricevuti.
CMD_ERR_DATA	0x04	Errore nei parametri del comando ricevuto.
CMD_ERR_TOUT	0x05	Scaduto il timeout di esecuzione del comando ricevuto.
CMD_ERR_DATA_TYPE	0x09	Tipo di dati non accettato.
CMD_ERR_PORT_TYPE	0x0A	Tipo di porta non presente.
CMD_ERR_PORT_NUMBER	0x0B	Numero di porta non presente.
CMD_ERR_DATA_SIZE	0x0C	Errore nella dimensione dei dati.

NOTA: ulteriori valori di errore in esecuzione, possono essere definiti nello sviluppo di firmware specifici.



COMANDO ABBREVIATO

COMMAND FORMAT						
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6	BYTES N°7 to N
COMMAND HEADER	SLAVE ADDRESS	MASTER ADDRESS	COMMAND	ID	PARAMETER SIZE	PARAMETER

ANSWER FORMAT					
BYTE N°1	BYTE N°2	BYTE N°3	BYTE N°4	BYTE N°5	BYTE N°6
ANSWER HEADER	MASTER ADDRESS	SLAVE ADDRESS	COMMAND	ID	EXECUTION RESULT

Esempio di comando e risposta:

master → *slave* 0x01,0x02,0x01,0x6F,0x00,0x04,0x00,0x00,0x00,0x0F
slave → *master* 0x02,0x01,0x02,0x6F,0x00,0x00

EXECUTION RESULT		
Name	Value (hex.)	Description
CMD_ACK	0x00	Esecuzione senza errori.
CMD_ERR_FORM	0x03	Errore nel formato dei parametri ricevuti.
CMD_ERR_DATA	0x04	Errore nei parametri del comando ricevuto.
CMD_ERR_TOUT	0x05	Scaduto il timeout di esecuzione del comando ricevuto.
CMD_ERR_DATA_TYPE	0x09	Tipo di dati non accettato.
CMD_ERR_PORT_TYPE	0x0A	Tipo di porta non presente.
CMD_ERR_PORT_NUMBER	0x0B	Numero di porta non presente.
CMD_ERR_DATA_SIZE	0x0C	Errore nella dimensione dei dati.

NOTA: ulteriori valori di errore in esecuzione, possono essere definiti nello sviluppo di firmware specifici.

