

Software di Controllo per
ARcetri CORrelation Spectrometer
Lettura dei total power dei campionatori

F. Palagi¹ e G. Comoretto ²

¹ IRA - CNR, Sezione di Firenze, l.go E. Fermi 5, 50125 Firenze (Italy)

² Osservatorio Astrofisico di Arcetri, l.go E. Fermi 5, 50125 Firenze (Italy)

Arcetri Technical Report N° 1/2002
Florence January 2002

8 gennaio 2002

Sommario

I campionatori dello spettrometro *ARcetri COrrelation Spectrometer, ARCOS* sono forniti di dispositivi per la misura di *total power* all'ingresso di ciascun canale. Nella prima versione del programma di controllo di ARCOS la misura di potenza totale veniva fatta per mezzo di rivelatori quadratici esterni e voltmetri digitali.

In questo rapporto si descrive lo sviluppo del software necessario per l'utilizzo dei misuratori di total power interni e le modifiche apportate al programma esistenze per la sua integrazione.

Il progetto segue una metodologia orientata agli oggetti, anche se la realizzazione e' fatta in linguaggio C.

1 Descrizione del problema

La lettura dei misuratori di Total Power dei campionatori di ARCOS richiede alcune modifiche sia al programma `adlb4arc` (PC-DOS) che al programma `adlb_serv` (VME-WxWorks).

I paragrafi 1.1 e 1.2 descrivono la situazione attuale mentre il paragrafo 1.3 descrive l'operazione che si vuole aggiungere al programma.

1.1 ADLB4ARC

Nel programma `adlb4arc`, il voltmetro, che misura il livello di total power all'ingresso di ciascuna sezione del correlatore, è definito al momento della configurazione iniziale associando opportuni valori (parametri) alle parole chiave:

- `LO2TPTYPE`
- `LO2TPADDR`

La misura della temperatura di sistema è demandata alla funzione

```
measure_tsys(void)
```

ed è effettuata sequenzialmente per ciascuna sezione del correlatore.

La lettura dei contatori collegati ai rivelatori di total power è delegata alla funzione

```
getvolt(int type, /* valore definito dal comando LO2TPTYPE */
        int addr, /* valore definito dal comando LO2TPADDR */
        double tint, /* tempo di integrazione in sec. */
        double *value /* Valore letto */
        );
```

Questa funzione è costituita essenzialmente da un costrutto di tipo *switch* sulla variabile `type`. Per ognuno dei tipi di contatore supportati si eseguono le seguenti operazioni:

- a) calcolo del numeri di cicli di lettura elementari sulla base del tempo di integrazione richiesto.
- b) Inizializzazione di `*value` a zero.
- c) Configurazione del contatore per la lettura.
- d) lettura del conteggio
- e) assegnazione del conteggio a `*value`

I comandi scambiati con il contatore devono rispettare il protocollo del bus su cui questo è installato. Nel nostro caso si tratta della linea parallela IEEE488.

1.2 ADLB_SERV

Il correlatore ARCOS è equipaggiato da una CPU Motorola PowerPC su scheda madre VME, sotto sistema operativo WxWorks. Esso è collegato al PC di controllo attraverso un'interfaccia parallela IEEE488 ed è identificato dall'unità logica 13.

Il programma di controllo `adlb_serv` è basato su 4 moduli concorrenti (*thread*) la cui attività è coordinata attraverso altrettante code di comandi di tipo FIFO, (`CMD_Q`, `INTEG_Q`, `PRINT_Q`, `OUT_Q`), che regolano il flusso dei comandi interni necessari per eseguire le funzioni di integrazione e di restituzione dei risultati all'esterno. La coda `OUT_Q`, coordina il flusso dei messaggi in uscita sulla porta di comunicazione (linea IEEE488). La struttura del programma è descritta in [1]

I moduli fondamentali sono (fra parentesi è riportata la coda associata al modulo):

1. Modulo principale (`OUT_Q`).
2. `CommandInterpreter` (`CMD_Q`).
3. `Print` (`PRINT_Q`).
4. `Integrator` (`INTEG_Q`).

Nella versione attuale (Luglio 1998) la comunicazione con l'esterno avviene attraverso una porta IEEE488, ma in futuro essa sarà sostituita da una porta *ethernet* con protocollo TCP/IP. Essa è gestita dal modulo principale (`main`) del programma (file: `adlb_serv.c`).

I comandi sono ricevuti dal modulo `Interface` e successivamente sono inseriti nella coda `CMD_Q` come oggetti `AdlbCommands`. La lettura dei comandi in arrivo è disabilitata durante la trasmissione di dati (misure o messaggi) verso l'esterno.

I comandi della coda `CMD_Q` sono letti dal modulo `CommandInterpreter`, che confronta il comando letto con la tabella dei comandi validi. Ad ogni comando è associata una funzione. Se il comando è convalidato il modulo esegue la funzione associata, altrimenti invia alla coda di stampa il messaggio di comando non valido.

Se la funzione associata richiede la collaborazione di uno degli altri moduli, essa genera un comando interno e lo inserisce nella coda di comandi appropriata.

Il modulo `Integrator` legge i comandi dalla coda `INTEG_Q`, li verifica ed esegue la funzione associata. Alla fine dell'integrazione emette un comando interno sulla coda `CMD_Q`, per segnalare che l'integrazione è terminata e che la misura è pronta per essere trasmessa sulla coda di stampa.

Il modulo `Print` crea un oggetto `AdlbResponse`, che contiene il risultato della misura (ACF), lo aggiunge alla coda di uscita `OUT_Q` e segnala a `CommandInterpreter` che la stampa è terminata, inviando il comando interno `#EP`

1.3 MISURA di TOTAL POWER

I misuratori di total power del correlatore ARCOS sono integrati nei campionatori. Dal punto di vista del programma di controllo, la misura è del tutto simile alla misura della funzione di autocorrelazione e quindi essi possono essere implementati sulla falsariga dei comandi che effettuano tale misura, tenendo conto che ARCOS misura contemporaneamente (con un solo comando) i livelli di tutte le sezioni del correlatore mentre, nella versione attuale, i voltmetri sono letti in modo sequenziale.

Sono disponibili due funzioni di alto livello per la lettura dei contatori di ARCOS, che sono contenute nella libreria `correlib.o`:

read_tp Legge i contatori un numero prefissato di volte e restituisce tutte le misure fatte in un array dimensionato in maniera opportuna.

read_tp_stat Legge i contatori un numero specificato di volte e restituisce il valore medio e la standard deviation della misura.

Le funzioni di basso livello sono implementate nel driver del correlatore, `crdrv`, visto attraverso il file speciale `/dev/cr_tp`. Queste funzioni sono descritte in [1, p.13]. Rispetto alla prima versione del documento citato, sono state apportate al driver le seguenti modifiche e/o aggiunte:

- Le costanti dichiarate che identificano la configurazione dei contatori di total power sono `TP_MONP`, `TP_MONM`, `TP_MONZ` e `TP_MONT`, rispettivamente per la misura delle soglie `+1`, `-1`, `0` e del livello di total power.
- Il tempo di integrazione elementare è selezionabile tra i valori di 20, 100, 200 e 1000 ms.

Un esempio di uso delle funzioni del driver `/dev/cr_tp` si trova nel programma che verifica il corretto funzionamento delle soglie, `read_tp.c`.

2 Analisi dei requisiti

Esistono due moduli separati, `adlb4arc` e `adlb_serv`, che vengono eseguiti da due CPU's diverse. Questa situazione è rappresentata in fig. 1 che mostra come le classi di oggetti sono raggruppate in due *categorie* distinte. La relazione indica che esse comunicano attraverso un'interfaccia di tipo IEEE488.

Tuttavia, esiste un legame forte fra le due strutture, nel senso che ad un comando generato da `adlb4arc` corrisponde una funzione in `adlb_serv`.

2.1 Modello a Oggetti

Dalla descrizione di ADLB4ARC si individuano le seguenti classi di oggetti. **Attenuatore** e **Marca** sono prese in considerazione per descrivere un possibile contesto all'interno del quale si esegue una misura di total power. La descrizione completa delle classi, che comprende responsabilità attributi e funzioni è riportata in appendice e il relativo modello statico è rappresentato in Fig. 2.

Totalpower	Esegue le misure di total power con ogni tipo di contatore
Interface	Gestisce il colloquio sulla porta di comunicazione
Attenuatore	Controlla il livello del segnale d'ingresso
Marca	Controlla il generatore di rumore per la calibrazione

Total Power measurement

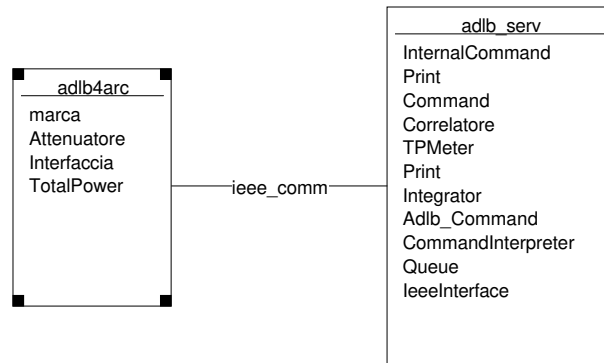


Figura 1: Categorie delle classi coinvolte nella misura di total power.

Dalla descrizione di `adlb_serv` si individuano le classi riportate nella tabella seguente (v. anche in appendice) e il relativo modello statico e' rappresentato in Fig. 3:

Thread	Gestisce la concorrenza fra moduli del programma
Interface	Gestisce il colloquio con l'esterno su una porta generica
IeeeInterface	Gestisce il colloquio sulla porta di comunicazione IEEE488
Queue	Coordina l'esecuzione di comandi
CommandInterpreter	Verifica la validità di un comando esterno ed esegue la funzione associata
Adlb_response	Contiene le risposte da inviare ad <code>adlb4arc</code>
Adlb_Command	Contiene il comando ricevuto da <code>adlb4arc</code>
Integrator	Controlla la temporizzazione delle misure
Print	Formatta e riporta i risultati delle misure sulla coda di uscita
Correlator	Misura la funzione di correlazione
TotalPower	Misura il segnale di ingresso a ciascuna sezione
InternalCommand	Comando interno gestito da Queue
Command	Comandi conosciuti dal <code>CommandInterpreter</code>

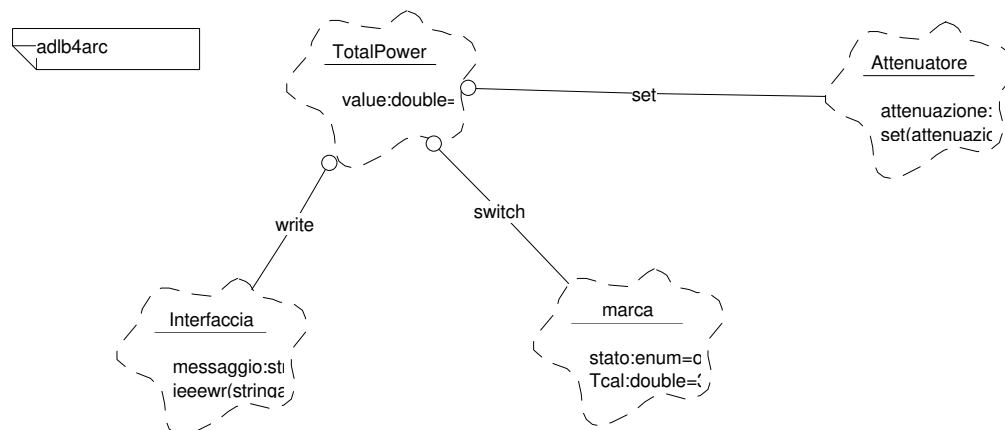


Figura 2: Modello a oggetti del sistema di misura della temperatura di sistema.

2.2 Modello dinamico

2.2.1 Scenario per la misura di tsys

Questo scenario è rappresentato nel diagramma a oggetti e diagramma degli eventi di fig. 4 e 5.

- * totalpower definisce le sezioni per le quali richiede la misura
- * totalpower invia all'interfaccia il comando per la configurazione dei lettori di total power
- * totalpower accende la marca
- * totalpower legge il valore corrispondente a marca accesa
- * totalpower spegne la marca
- * totalpower legge il valore di cielo (marca spenta)
- * totalpower legge il valore corrente degli attenuatori IF
- * totalpower pone gli attenuatori al massimo
- * totalpower legge il valore di zero
- * totalpower ripristina il valore degli attenuatori IF

2.2.2 Scenario di lettura dei contatori di ARCOS.

Questo scenario descrive la preparazione della stringa di comando e il suo invio sulla porta di comunicazione con il correlatore; la parte relativa all'esecuzione della lettura compresa la restituzione del risultato sarà descritta in uno scenario successivo, nella sezione dedicata ad `adlbserv`.

Lo scenario è rappresentato nel diagramma a oggetti e diagramma degli eventi di fig. 6).

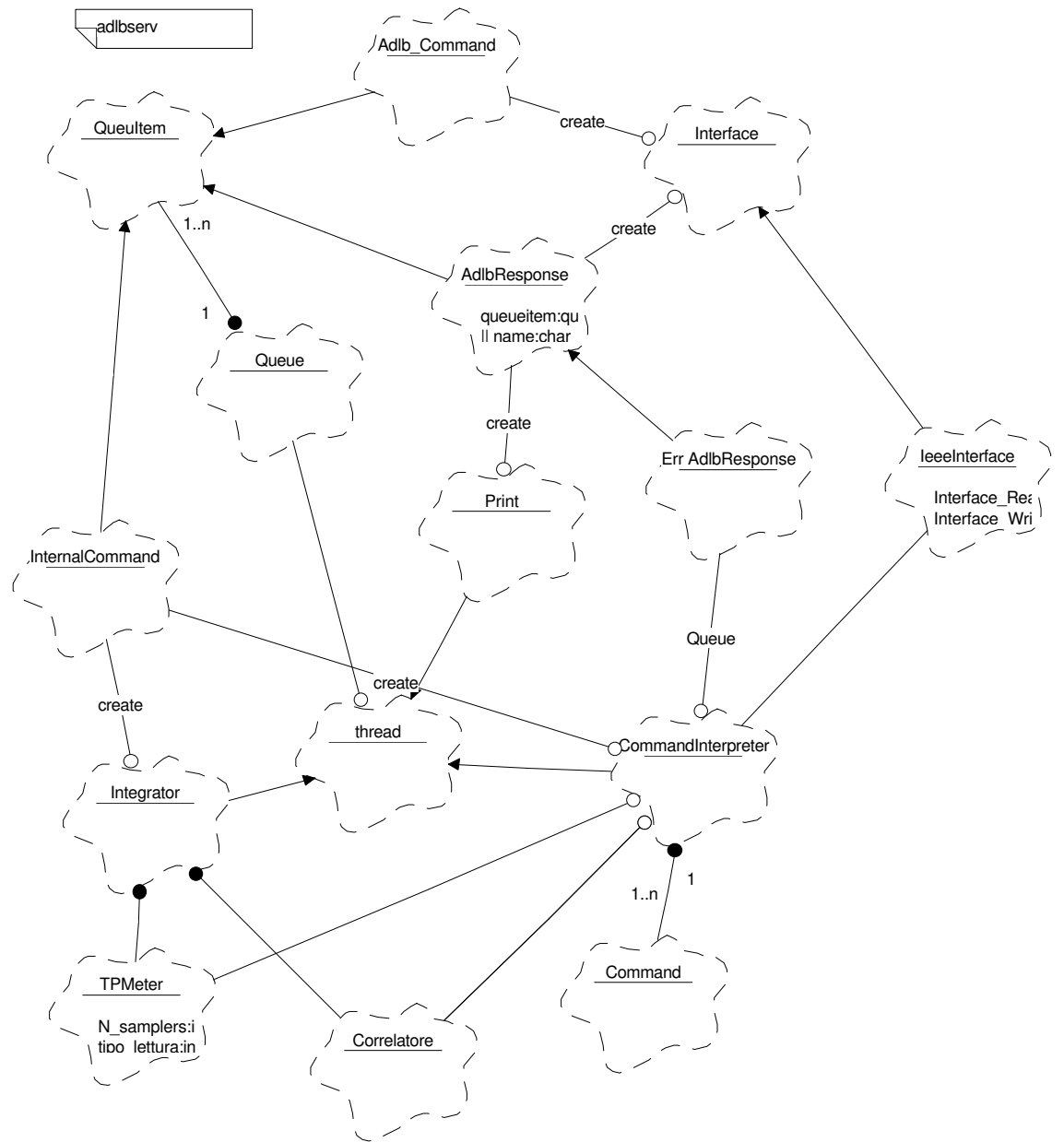


Figura 3: Modello a oggetti del sistema di misura della potenza all'ingresso del correlatore.

Misura Temp. Sistema

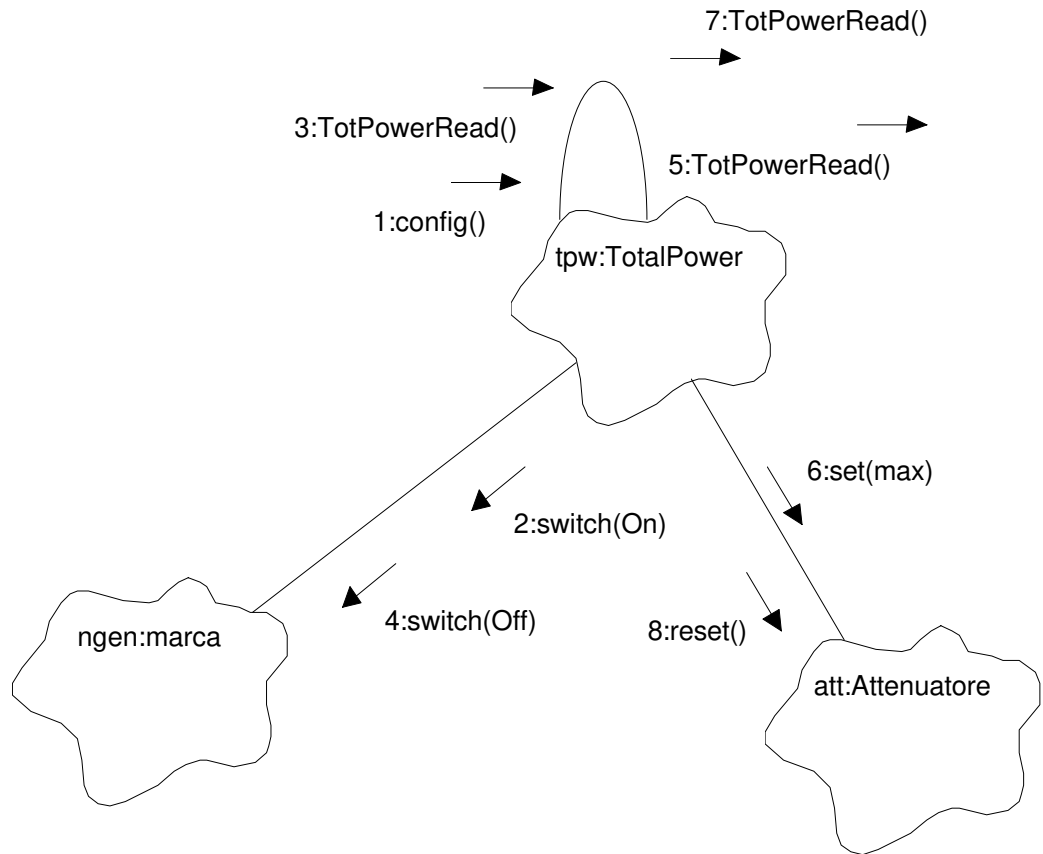


Figura 4: Diagramma degli oggetti per la misura della temperatura di sistema.

Misura della Temperatura di Sistema

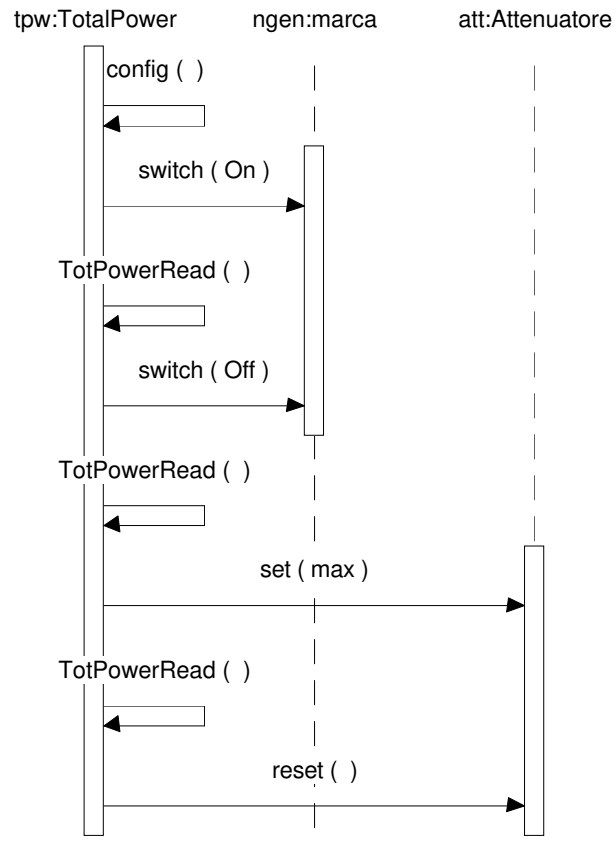


Figura 5: Diagramma delle interazioni per la misura della temperatura di sistema.

- * Si definiscono i contatori abilitati e si prepara la stringa di comando che contiene l'identificatore del comando seguito dai parametri che descrivono la configurazione e il tempo di integrazione richiesti.
- * Invia la stringa di comando tramite l'Interfaccia.
- * Attende un numero di secondi pari al tempo di integrazione.
- * Legge dalla porta di comunicazione il messaggio di risposta, che contiene il risultato delle misure per tutte le sezioni.
- * Assegna i valori validi alle variabili di conteggio in base alla configurazione richiesta.

2.2.3 Scenario per la lettura dei contatori di total power.

Questo scenario si riferisce alla parte implementata in `adlbserv` ed è rappresentato nelle figg. 7 e 8.

I comandi arrivano dalla porta di comunicazione sotto forma di stringhe di caratteri che iniziano con una parola chiave che identifica il comando seguita dai relativi parametri.

- * L'interfaccia (`Interface`) legge il messaggio e crea l'oggetto `adlb_command`. Invia il comando a `CommandInterpreter`.
- * `CommandInterpreter` verifica la validità del comando ed esegue la funzione associata.
- * `CommandInterpreter` invia il comando interno di lettura dei contatori di total power a `Integrator`.
- * `Integrator` segnala che i dati di total power non sono validi, esegue la misura con la collaborazione di `TPMeter`. I risultati sono memorizzati.
- * Al termine della lettura, `Integrator` invia il comando interno di fine misura a `CommandInterpreter`.
- * `CommandInterpreter` segnala che i dati sono validi e invia a `Print` il comando interno di stampa delle misure sulla porta di comunicazione.
- * `Print` chiede a `TPMeter` un oggetto `adlb_response` che contiene le misure e lo invia sulla porta di comunicazione (`OUT_QUE`).

2.3 Modello funzionale

Il modello funzionale identifica il flusso dei dati per ottenere il risultato voluto. Quindi:

- a) si identificano i valori di uscita: Conteggi dei total power
- b) si identificano i valori di ingresso: tipo di contatore da usare (per sezione), numero di sezioni del correlatore, tempo di integrazione.

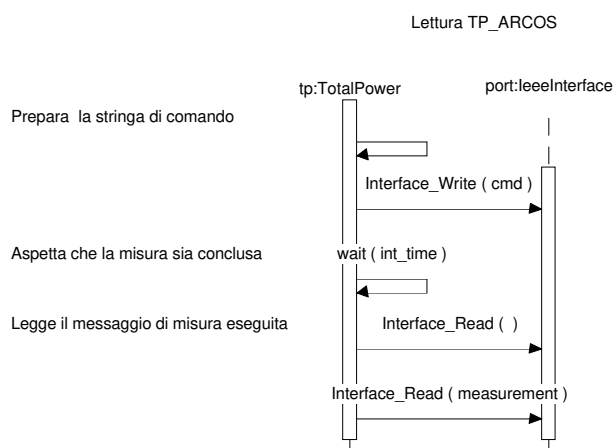
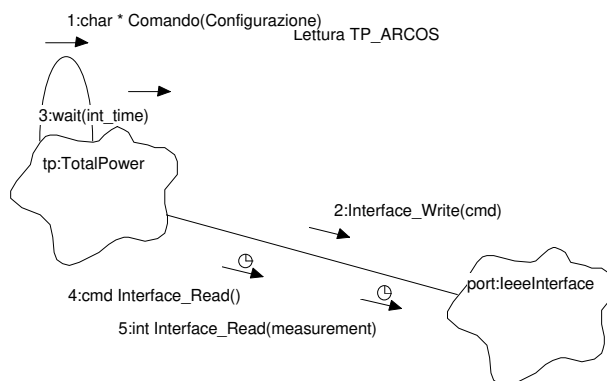


Figura 6: Diagramma degli oggetti e delle interazioni per la misura del livello di total power.

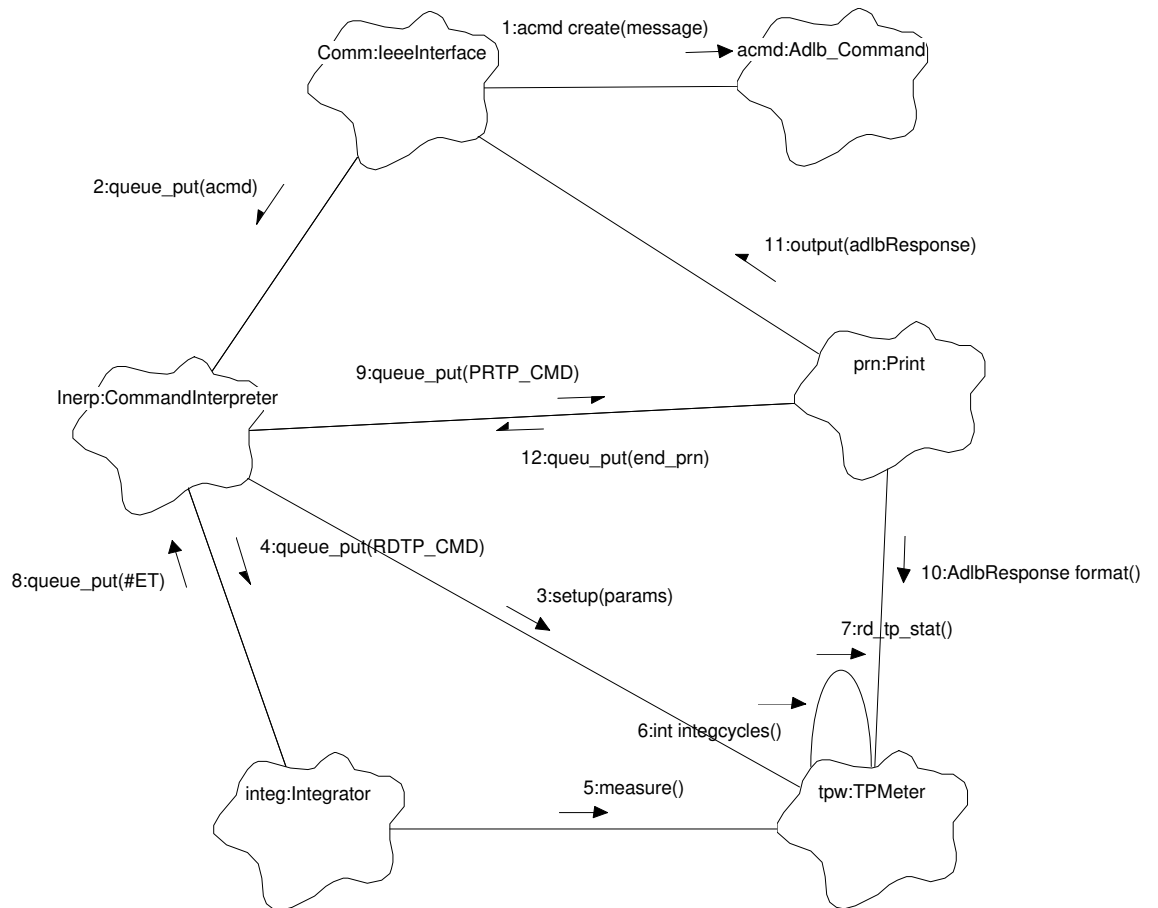


Figura 7: Diagramma degli oggetti che descrive la misura del livello di total power in adlbserv.

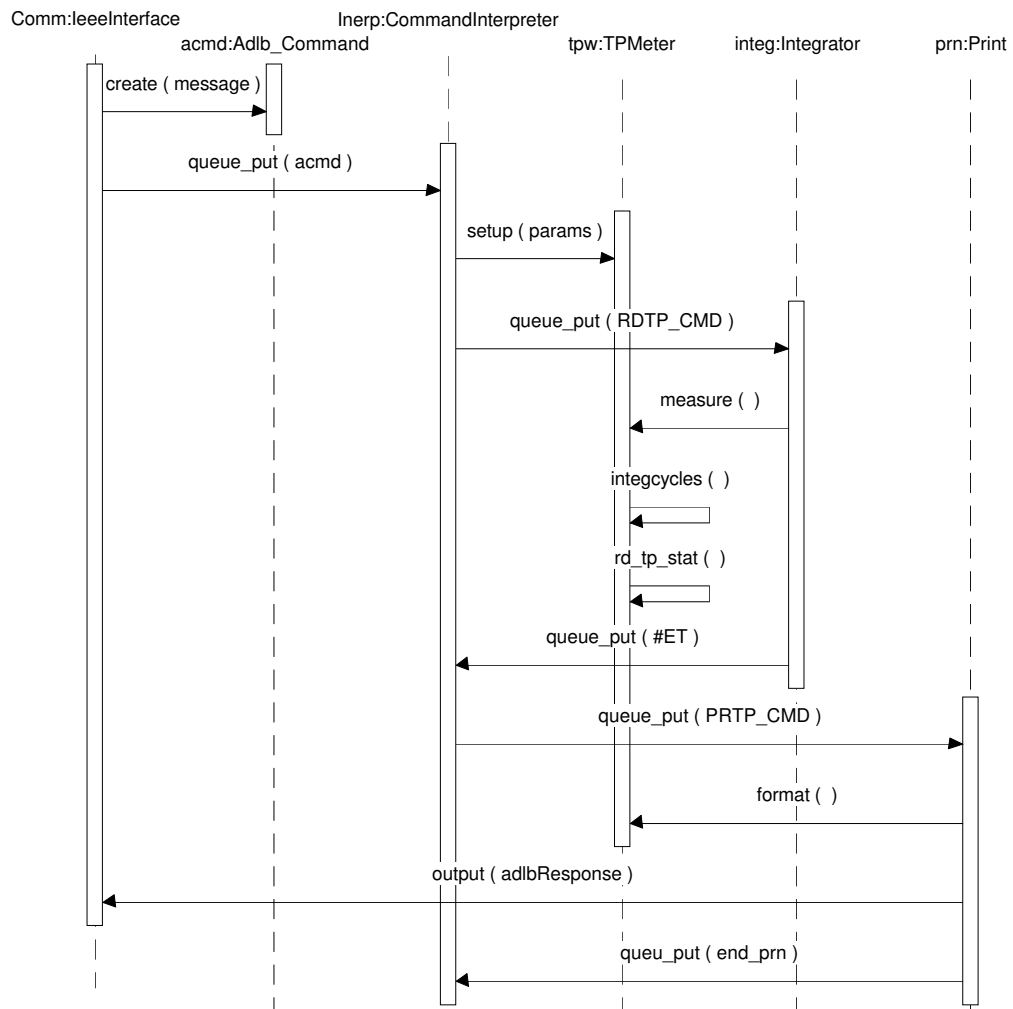


Figura 8: Diagramma delle interazioni che descrive la misura del livello di total power in adlbserv.

LETTURA dei contatori di TOTAL POWER

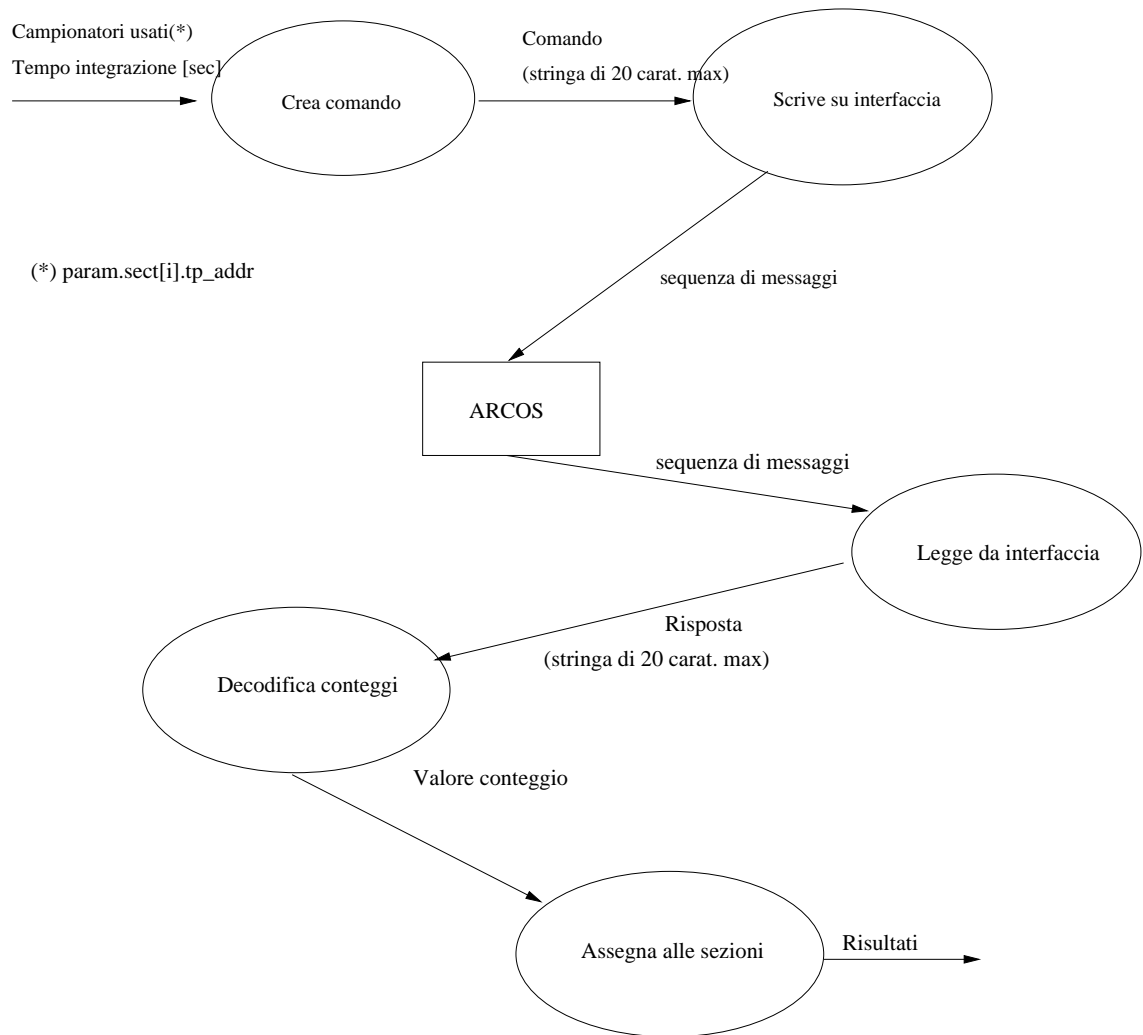


Figura 9: Modello funzionale del sistema di misura del Total Power.

2.3.1 Lettura dei contatori (adlb4arc).

Valore di uscita: misura del livello di total power del segnale di ingresso al campionatore.

I valori di ingresso sono: Configurazione e tempo di integrazione [sec].

Descrizione:

- * TotalPower conta il numero di contatori di tipo TP_ARCOS
- * Se il numero di tali contatori è $\neq 0$ esegue la lettura dei contatori di total power dal correlatore.
- * Esegue la lettura dei contatori diversi da TP_ARCOS.

2.3.2 Lettura dei contatori (adlb_serv).

La lettura avviene usando la funzione `rd_tp_stat` con un ciclo di misura elementare di 50 ms (`CK_TP_50M`).

Valore di uscita: Messaggio con la lettura degli otto contatori.

Valori di ingresso: messaggio di configurazione.

Descrizione:

- * Si costruisce il vettore di configurazione.
- * Si calcola il numero di cicli elementari che corrispondono al tempo di integrazione richiesto.
- * Si assegnano le misure fatte alle sezioni

3 Progetto

La struttura del modello a oggetti rimane quella delle figg. 2 e 3. Le funzioni di lettura dei contatori i TP_ARCOS (contenute nel driver `/dev/cr_tp`) riportano sempre i conteggi relativi a tutti gli 8 campionatori presenti nel sistema; i valori significativi si riferiscono solo ai contatori abilitati.

Al contrario la funzione `get_volt` è stata progettata per leggere i contatori (voltmetri) uno alla volta, per ciascuna sezione del correlatore.

Di conseguenza è necessario ristrutturare l'operazione di lettura, in modo da tener conto di entrambe le situazioni.

Si apportano le seguenti modifiche e aggiunte:

- In `adlb4arc`:
 1. Si definisce un nuovo tipo di contatore TP_ARCOS e si aggiungono i due comandi TPR e TPW nel protocollo di comunicazione con il correlatore, per realizzare il comando di lettura dei contatori di questo tipo.
 2. Nella funzione `t_sys` si estrae il loop di lettura delle sezioni creando una funzione specifica `TotPowerRead`.

3. **TotPowerRead** conta il numero di contatori TP_ARCOS richiesti. Se ne esiste almeno uno, essi vengono letti contemporaneamente inviando un opportuno comando ad ARCOS. Successivamente i contatori diversi da TP_ARCOS vengono letti in sequenza per mezzo della funzione `get_volt`.
4. Quando il tipo di contatore è definito TP_ARCOS i valori di LO2TPADDR vengono ignorati e in sostituzione si usa il valore del canale ritardato collegato (associato) alla sezione. Questa soluzione è valida solo nel caso di misura della funzione di AUTOCORRELAZIONE. Quando si misura la funzione di CROSSCORRELAZIONE occorre misurare la potenza totale di entrambi i canali.

- In `adlb_serv`:

1. Si aggiunge l'oggetto `totpower` della classe `TPMeter`. Questa classe ha come attributi `cfg` che definisce lo stato (abilitato/disabilitato) del contatore e `sect` che indica il numero di sezione a cui è collegato.
2. Si aggiunge il comando TPR nel modulo `CommandInterpreter` a cui corrisponde la funzione `ReadTpw`.
3. La funzione `ReadTpw` inserisce il comando interno RDTP_CMD nella coda INTEG_Q e gestisce la lettura dei risultati.
4. `Integrator` smista il comando di lettura a `totpower`.

3.1 Descrizione delle funzioni

Si descrivono le funzioni da modificare o aggiungere ai programmi `adlb4arc` e `adlb_serv`. Per ciascuna operazione si descrivono il flusso dei dati e le relative funzioni. Il diagramma di flusso dei dati descrive anche lo scambio di informazioni attraverso la porta di comunicazione e riguarda quindi entrambi i programmi.

3.1.1 Funzioni in `adlb4arc`

La funzione `TotPowerRead` ha il compito di gestire la lettura di qualsiasi tipo di contatore di total power. Essa identifica i tipi di contatori utilizzati; se esiste almeno un contatore di tipo TP_ARCOS esegue una lettura contemporanea di tutte le sezioni e, successivamente, una lettura sequenziale dei contatori degli altri tipi. La funzione è descritta dalla seguente lista di operazioni:

- a) Si contano i misuratori di tipo TP_ARCOS.
- b) Se esiste almeno un contatore di tipo TP_ARCOS si esegue la funzione `TpArcosRead`.
- c) Si esegue la funzione `get_volt` per i contatori \neq TP_ARCOS. Il valore ottenuto sostituisce quello riportato (ma non valido) dalla lettura dei TP_ARCOS.

La funzione `TpArcosRead` esegue la lettura di tutti i campionatori di tipo TP_ARCOS. Riporta risultati validi solo per quelli abilitati. Il protocollo di comunicazione è identico a quello usato per la misura della Funzione di Autocorrelazione: il messaggio di risposta è iniziato dalla *keyword* TPW seguita da una stringa di caratteri che contengono per tutti i misuratori di total power:

1. Numero del contatore.
2. Valore della misura.
3. Standard deviation.

La funzione è descritta dalla seguente lista di operazioni e dal diagramma del flusso dei dati do fig. 10.

- Prepara il vettore di configurazione in base a quanto specificato nella struttura che descrive la sezione. Usa la codifica seguente:

1 = abilitato

0 = disabilitato

- Invia il comando di lettura TPR seguito dal tempo di integrazione espresso in secondi e dai parametri di configurazione. A tale scopo usa la funzione `adlbsend`.
- Attende per il tempo di integrazione e invia il comando PAS.
- Legge la risposta di fine misura.
- Legge il risultato della misura per tutte le sezioni.
- Assegna i valori per quelle abilitate.
- Si prevedono stringhe di errore nei seguenti casi:
 1. Errore di lettura.
 2. Errore di configurazione.
 3. Comando non valido.

4 Implementazione

4.1 ADLB4ARC

Comando di lettura `tprcmd_send(nsect)`.

- Assegna il valore `disable` tutti i contatori;
- Per ciascuna sezione:
 - Se il contatore della sezione è di tipo `TP_ARCOS`:
assegna il valore `enable` al contatore il cui indirizzo è stato specificato da `LO2TPADDR`;
 - Costruisce la stringa dei parametri: tempo integrazione, configurazione usando il formato:
`%f,%d,%d,%d,%d,%d,%d,%d,%d`

LETTURA dei contatori di TOTAL POWER

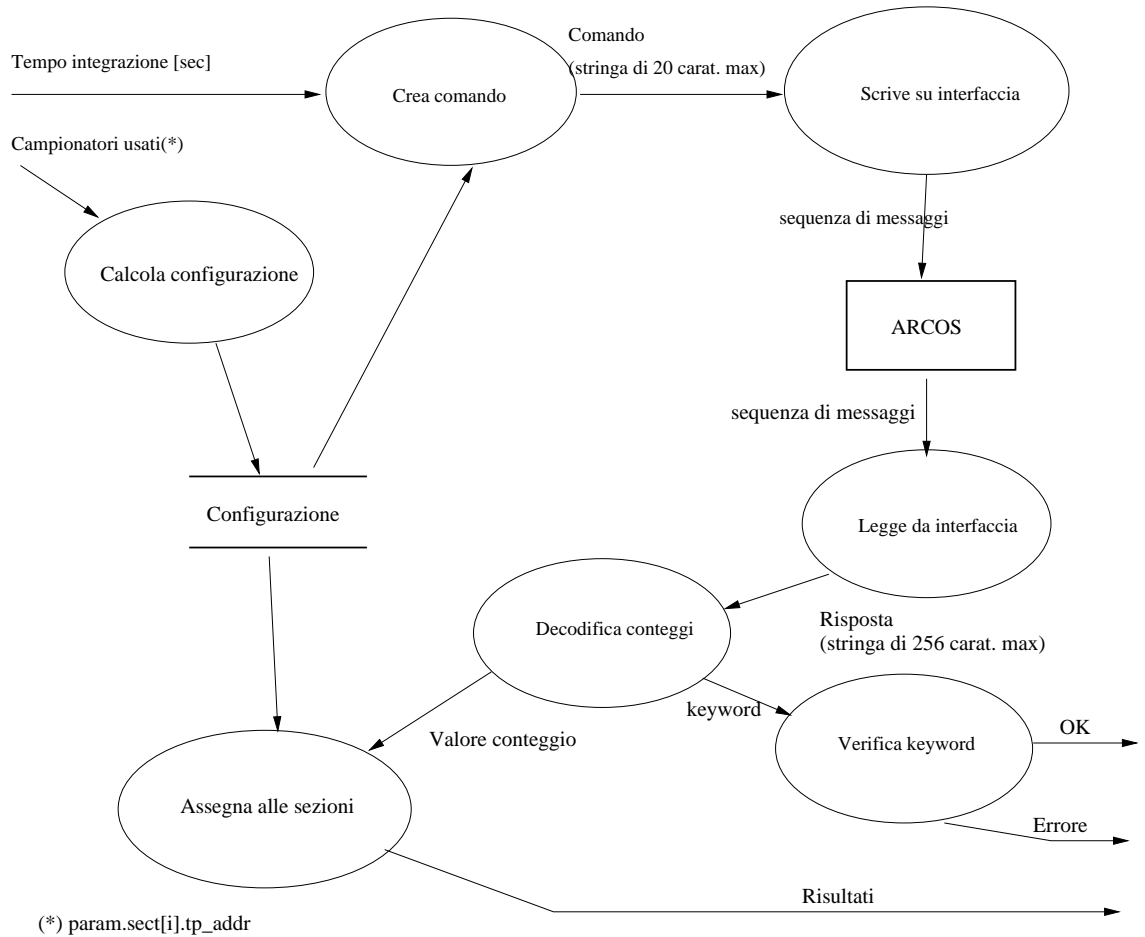


Figura 10: Adlb4arc: diagramma del flusso dei dati per la lettura dei Total Power come previsto dal progetto.

Lettura dei Risultati `tpw_answer_receive(void)`.

- Legge il comando dall'interfaccia `ieee488`.
- se il comando è `#TPW`.
- legge la stringa successiva, composta da 8 sequenze di formato `%d,%8.2e,%8.2e`, e terminata dal carattere `\n`.
- decodifica numero `totalpower`, valore, rms.

Funzione Assegna Risultato `SetResults(nsez, configurazione) → risultati`

- Per tutte le sezioni: assegna `risultato non valido`;
- Per tutte le sezioni:
- Indice Configurazione (`ic`) = indirizzo contatore;
- Se `configurazione[ic]` è abilitata
`risultato[nsezione] = conteggio`;

4.2 `adlb_serv`

Modifiche in `adlb_serv`. I contatori di total power sono descritti da un oggetto globale che contiene i parametri globali della misura (tempo integrazione, numero di campioni) e da una struttura che contiene i risultati della misura (counts, standard deviation e flag di configurazione) per ogni contatore.

- Si aggiunge il l'oggetto globale `totpower`.

Modifiche a `CommandInterpreter`.

- Si aggiunge il comando `TPR` associato alla funzione `ReadTpw` nella tabella dei comandi validi.
- Si aggiunge il comando `#ET` che segnala la fine della misura dei contatori di total power, associato alla funzione `EndTpw`.
- Si aggiunge la funzione `ReadTpw`.
- Si aggiunge la funzione `EndTpw`.

Funzione `ReadTpw`

- Segnala che è attiva la misura di total power.
- Richiede a `totpower` il setup del tempo di integrazione e del vettore di configurazione.
- Inserisce comando interno `RDTP_CMD` nella coda di `Integrator`.

Funzione `EndTpw`

- Abbassa la flag di misura attiva.
- Invia il comando di stampa dati total power (`PRTP_CMD`) alla coda di stampa.

Oggetto `totpower`. Questo oggetto contiene gli attributi:

1. tempo di integrazione.
2. Risultato delle misure
3. Errore delle misure
4. Configurazione.

e fornisce le seguenti operazioni:

1. Calcolo del numero di cicli elementari corrispondenti al tempo di integrazione richiesto.
2. Decodifica il valore del tempo di integrazione.
3. Costruisce il vettore di configurazione da usare per la lettura.
4. Chiama la funzione `rd_tp_stat`.

Modifiche a `Integrator` Il modulo `Integrator` smista i comandi ricevuti a `totpower` o `arcos`. Le modifiche da fare consistono nell'aggiungere le operazioni da eseguire quando riceve i comandi che riguardano `totpower`.

- Se il comando è `RDTP_CMD` :
- Converte il tempo di integrazione in cicli di misura di 50 ms.
- Chiama la funzione `read_tp_stat`.
- Assegna i risultati ai descrittori dei contatori.

Modifiche a `Printer`. La ricezione del comando `PRTP_CMD` provoca la scrittura dei seguenti messaggi nella coda `OUT_QUE`:

- il comando "TPW" che inizia la trasmissione.
- Un buffer con 8 sequenze di numeri che indicano:
 - Numero di totalpower (%d)
 - Valore medio delle misure (%8.2e)
 - Errore rms della misura (%8.2e)
- I risultati sono inviati alla coda di uscita come oggetto `AdlbResponse`.

Modifiche a `InternalCommand`.

- Si aggiunge il comando `PRTP_CMD` all'elenco dei comandi interni.

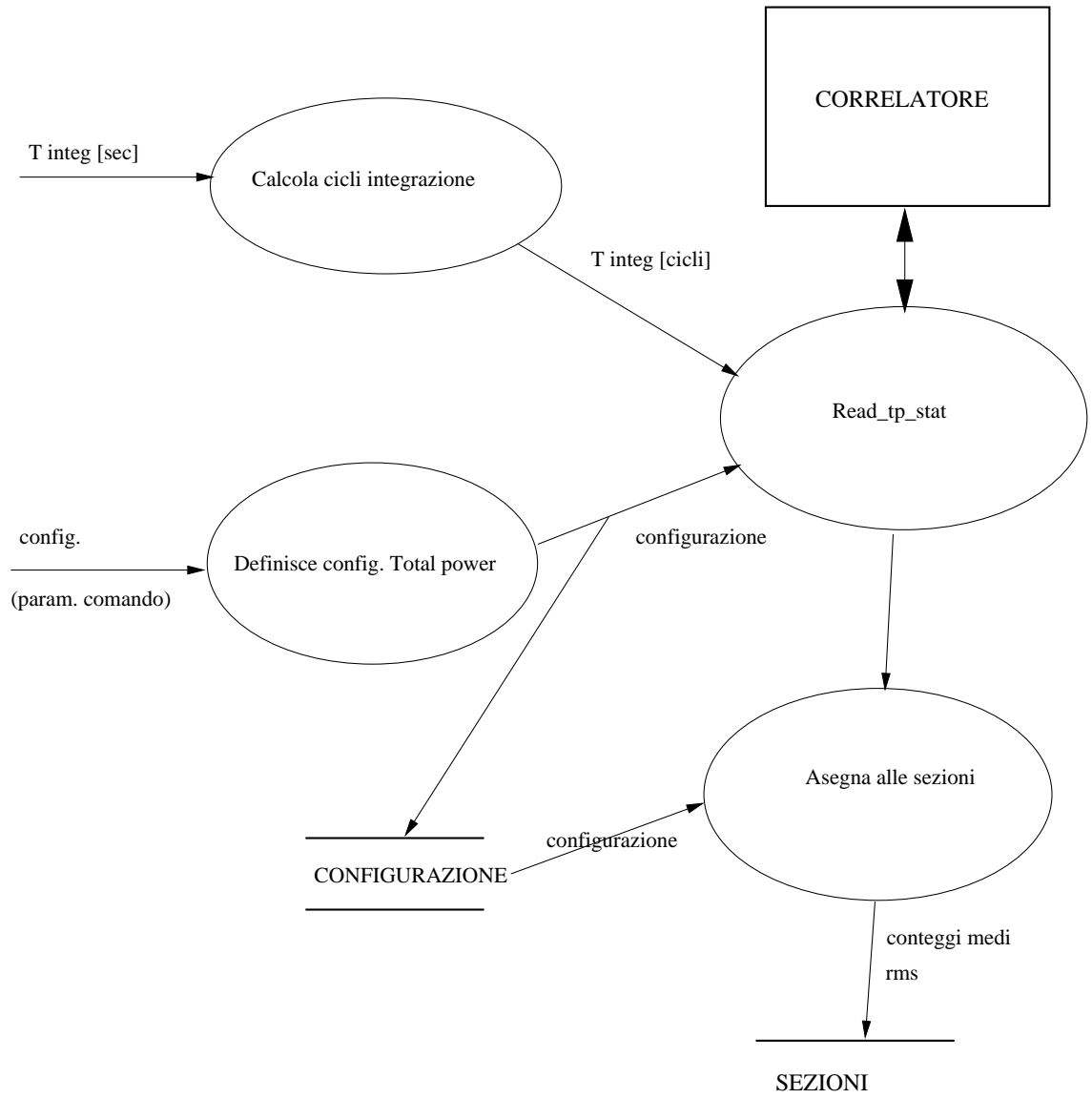


Figura 11: Adlbserv: diagramma del flusso dei dati per la lettura dei Total Power.

5 Descrizione delle Classi

CLASS	totalpower	
Superclass Subclass		
Responsabilità Operazioni	Attributi	Descrizione
Configura e legge i livelli di Total Power. Misura la temperatura di sistema. Lettura dei contatori.	misura	valore della lettura (conteggi)
	measure_tsys get_volt	

CLASS	interfaccia	
Superclass Subclass		
Responsabilità Operazioni	Attributi	Descrizione
Gestisce il controllo e i colloquio sul bus IEEE488. Scrittura di una stringa di caratteri	messaggio	stringa di comando da inviare
	risposta	stringa che contiene il risultato del comando inviato.
	ieeewr ieeerd	

Per la parte di interfaccia ci si limita a queste due funzioni, in quanto si sa che essa è già disponibile e funziona correttamente. Non richiede modifiche.

CLASS	Attenuatore	
Superclass Subclass		
Responsabilità Operazioni	Attributi	Descrizione
Definisce il livello di attenuazione IF. Aggiustamento del livello	livello	livello di attenuazione in dB
	att_set	

CLASS	marca	
Superclass Subclass		
Responsabilità Operazioni	Attributi	Descrizione
Controlla il generatore di rumore. Accende e spegne il generatore switch(stato)	tcal	Valore della marca di calibrazione Commuta allo stato

Dal lato adlb_serv si individuano i seguenti oggetti con relativi attributi e funzioni.

CLASS	Thread	
Superclass Subclass		
Responsabilità Operazioni	Attributi	Descrizione
Realizza la concorrenza fra oggetti Riunifica il controllo nel modulo principale del programma. Thread_Join		

CLASS	IeeeInterface	
Superclass Subclass	Interface	
Responsabilità Operazioni	Attributi	Descrizione
Gestisce le operazioni di I/O sulla porta IEEE488 del VME. Legge messaggio in arrivo Read Scrive messaggio in partenza Write	fileno	Unità logica dello stream

CLASS	Queue	
Superclass Subclass		
Responsabilità Operazioni	Attributi	Descrizione
Gestisce una sequenza di comandi nel modo FIFO Prende un comando dalla coda Pop Mette un comando nella coda Pop	first last n	Primo elemento della coda Ultimo elemento della coda Numero di elementi nella coda

CLASS	CommandInterpreter	
Superclass Subclass		
Responsabilità Operazioni	Attributi	Descrizione
Verifica la correttezza dei comandi e li esegue. CommandInterpreter	Cmdtable	Array di comandi

CLASS	adlb_response	
Superclass Subclass	QueueItem	
Responsabilità Operazioni	Attributi	Descrizione
Contiene le risposte ai comandi inviati da ADLB4ARC. Responsabilità Operazione	name param_buffer next_param paramlen	Identificatore Buffer dei parametri Indirizzo del parametro successivo Lunghezza della stringa dei parametri

CLASS	Adlb_Command	
Superclass Subclass	QueueItem	
Responsabilità Operazioni	Attributi	Descrizione
Contiene i Comandi inviati da ADLB4ARC. Responsabilità 1 Operazione	command params paramlen	Identificatore del comando Stringa dei parametri Lunghezza della stringa dei parametri

CLASS	Integrator	
Superclass Subclass		
Responsabilità Operazioni	Attributi	Descrizione
Esegue i comandi inseriti nella coda di integrazione. Segnala la fine dell'integrazione. Integrator	internal_cmd	Indirizzo del comando interno da eseguire

CLASS	Print	
Superclass Subclass		
Responsabilità Operazioni	Attributi	Descrizione
Legge risultati e risposte dalla coda di stampa, li formatta e li scrive sull'interfaccia. Invia il comando interno di fine stampa alla coda dei comandi. Print	cmd resp	Comando interno Oggetto di classe AdlbResponse

CLASS	Total_Power	
Superclass Subclass		
Responsabilità Operazioni	Attributi	Descrizione
Legge i conteggi del canale di total power dei samplers Predispone i samplers per eseguire la lettura richiesta. Configurazione dei contatori Config Lettura read_tp	nsamplers type *result	Numero di samplers nel sistema Tipo di lettura richiesta per ciascun sampler Risultato (conteggi) della lettura Usato per predisporre il tipo di lettura da effettuare

6 Files nuovi o modificati

Le tabelle che seguono riportano i nomi dei files nuovi o modificati, suddivisi per programma. Si riportano solo i files .c, le definizioni delle funzioni e delle strutture usate sono

contenute nei corrispondenti files `.h`.

6.1 ADLB4ARC

Nome	Funzioni contenute	Stato
adlbcom6.c	adlbget_tp	nuova
totalpow.c	tpcmd_send	nuova
	tpw_measures	nuova
	decode_answer	nuova
	assign_results	nuova
	print_results	nuova
rs_sms.c	rs_sms_freq_ampl	nuova
marca.c	measure_tsys	modificata
	TotPowerRead	nuova
	TpArcosRead	nuova

6.2 adlbserv

Nome	Funzioni contenute	Stato
AdlbCommand.h		modificato
CommandInterpreter.c	ReadTpw	nuova
	EndTpw	nuova
Print.c	Print	modificato
Integrator.c	Integrator	modificato
Totalpower.c	totpower_create	nuova
	totpower_initialize	nuova
	totpower_mesure	nuova
	totpower_setup	nuova
	totpower_integcycles	nuova
	totpower_format	nuova
	totpower_resp_destroy	nuova

Riferimenti bibliografici

- [1] Comoretto G., 1995, *ARCOS - Arcetri Correlator spectrometer, Stato del software di controllo dello strumento*, ATR 9/1995.

file: `palagi/strumenti/arcos/adlb4interface/totalpower.tex`