

INTERNAL REPORT

DEROTATORE DEL RICEVITORE MULTI-BEAM

Descrizione dell'apparecchio e del software a corredo

Marco Buttu

Report N. 13, released: 09/12/2011

Reviewer: Sergio Poppi



Osservatorio
Astronomico
di Cagliari

Derotatore del ricevitore multi-beam.**Descrizione dell'apparecchio e del software a corredo.**

Marco Buttu <mbuttu@oa-cagliari.inaf.it>

Questo documento è strutturato in modo da poter risultare utile sia come guida all'utilizzo del software sviluppato che di quello di configurazione dei componenti del sistema *derotatore*¹. Per poter far ciò ciascun argomento verrà introdotto cercando di delineare il contesto in cui si opera; verrà quindi descritto il derotatore, il framework ACS (ALMA Common Software) che rappresenta una infrastruttura alla quale sarà legato il software sviluppato, gli strumenti di design utilizzati e quelli in fase di studio.

Nel primo capitolo viene descritto il sistema derotatore, vengono riportate le sue specifiche tecniche ed illustrato brevemente come utilizzare il software fornito dal produttore al fine di configurare l'interfaccia di comunicazione con i componenti del sistema. Il secondo capitolo introduce alcuni concetti base del framework ACS e descrive il software ACS che consente la movimentazione del derotatore.

¹Il derotatore è il sistema che permette la rotazione del ricevitore multi-beam a 22 GHz attorno al proprio feed centrale.

Indice

1	Descrizione del Derotatore	7
1.1	Panoramica dell'apparecchio	7
1.2	Descrizione dei Componenti del Sistema	8
1.2.1	Specifiche tecniche	11
1.3	Controllo Remoto del Motore-Azionamento	12
1.4	Collocazione del Derotatore sull'Antenna	14
2	Software di Controllo del Derotatore	15
2.1	Panoramica su ACS	15
2.2	Struttura del Software di Movimentazione del Derotatore	16
2.3	Convenzioni sul Verso della Rotazione	19
2.4	Utilizzo del Client	20
	Bibliografia	23
	Elenco delle figure	25
	Elenco delle tabelle	27

Capitolo 1

Descrizione del Derotatore

1.1 Panoramica dell'apparecchio

Il derotatore è il sistema che permette la rotazione del ricevitore multi-beam a 22 GHz attorno al proprio feed centrale.

Il sistema si basa sull'utilizzo di un motore con azionamento integrato in grado di gestire autonomamente tutte le funzioni necessarie per il controllo in posizione di una movimentazione.

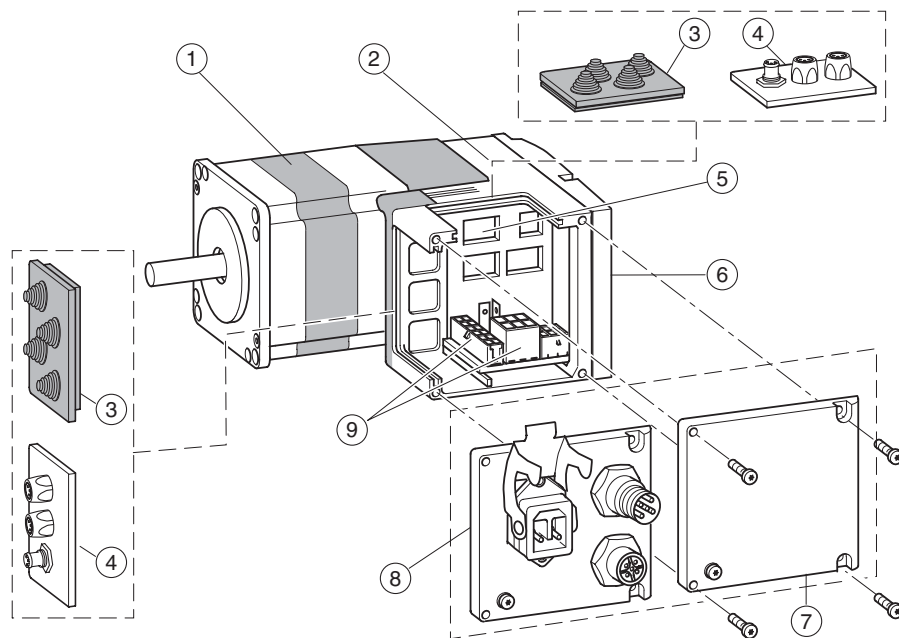


Figura 1.1: Panoramica dell'apparecchio

La figura 1.1 riporta una panoramica dell'apparecchio. Questo è composto dalle seguenti parti:

1. motore EC
2. carcassa dell'elettronica
3. passacavo (accessorio)
4. scheda I/O con connettore industriale (accessorio)
5. possibilità di impostazione mediante interruttori
6. coperchio della carcassa dell'elettronica
7. coperchio della scatola connettori
8. coperchio con connettore industriale per la tensione di alimentazione VDC e la connessione bus di campo IN/OUT
9. interfacce elettriche

Il motore-azionamento, una volta ricevuto un comando seriale di posizione finale da raggiungere, provvede a muoversi e a posizionarsi rispettando i parametri di velocità massima, accelerazione e decelerazione configurati tramite il software fornito dal produttore.

Il comando di posizione, come pure il controllo completo dell'azionamento, avviene attraverso un convertitore LAN/RS485 che permette di inserire il derotatore all'interno della rete LAN dell'antenna.

Sempre tramite LAN è leggibile un trasduttore di posizione assoluto posto direttamente sul rotore porta ricevitore. Questo componente consente di fare un confronto tra la sua posizione e quella del motore.

1.2 Descrizione dei Componenti del Sistema

Il sistema è composto dalle seguenti parti:

- **Motore brushless** costituito da un motore a corrente continua eccitato da un magnete permanente, con una risoluzione interna di 12 incrementi al giro. Il motore ha le seguenti caratteristiche:
 - alimentazione in bassa tensione a 24 VDC con una sola tensione di alimentazione per la potenza ed il controllo;

- presenta un'elevata coppia di arresto automatico a motore spento che nella maggior parte delle applicazioni rende superfluo l'impiego di un freno;
 - azionamento incorporato con funzione di posizionario punto-punto. Il loop di posizione si richiude sull'encoder interno al motore;
 - encoder assoluto¹ di posizione incorporato a 12 step/giro;
 - interfaccia I/O per fine corsa e stop di sicurezza;
 - interfaccia seriale RS485 per comando di configurazione.
- **Catena cinematica** con rapporto di riduzione 5494:1 per ridurre l'elevata velocità di rotazione del motore ed avere il corretto valore di momento torcente per poter ruotare l'intero ricevitore. La catena cinematica è composta da un riduttore epicicloidale (60:1) calettato direttamente sul motore che a sua volta si accoppia con un riduttore a vite senza fine (10.25:1) sulla cui uscita è fissata la puleggia per la cinghia dentata che trasmette il moto alla ralla centrale sulla quale viene fissato il ricevitore. Il rapporto di riduzione dell'accoppiamento tramite cinghia è pari a 8,933:1.
 - **Trasduttore di posizione assoluto** posto direttamente sul rotore del ricevitore per verificare il posizionamento del derotatore a valle della catena cinematica. Tale trasduttore è composto da un nastro magnetico codificato arrotolato sulla ralla porta ricevitore e dal relativo sensore di lettura. Il sensore è dotato di interfaccia seriale RS422 per la lettura remota dell'informazione di posizione che è la lunghezza dell'arco di circonferenza su cui è avvolto il nastro magnetico.
 - **Control box**, all'interno del quale è alloggiato un convertitore da LAN a interfaccia seriale RS485 e RS422 per poter accedere rispettivamente al motore-azionamento e al trasduttore assoluto di posizione. All'interno di tale scatola avvengono anche le interconnessioni tra i vari interruttori di fine corsa e il motore-azionamento.
 - **Power supply box**, all'interno del quale sono presenti due alimentatori a 24 VDC, uno di potenza non regolato per l'alimentazione del motore-azionamento, il secondo per l'alimentazione del convertitore LAN/seriale e dei fine corsa. Il power supply box è alimentato dalla linea UPS.

¹L'azionamento nel momento in cui gli viene tolta l'alimentazione provvede a salvare in memoria la posizione attuale. Se alla riaccensione la posizione del motore non corrisponde a quella memorizzata l'azionamento segnala una condizione di avaria.

- **Switch** di fine corsa. Complessivamente sono presenti i seguenti interruttori:
 - LIMP fine corsa positivo “proximity switch”: se attivo l’azionamento si ferma con un “quick stop” ed è possibile uscire da tale condizione con un comando manuale nella direzione opposta. L’interruttore LIMP viene utilizzato anche quale posizione di riferimento;
 - LIMN fine corsa negativo “proximity switch”: se attivo l’azionamento si ferma con una “quick stop” ed è possibile uscire da tale condizione con un comando manuale nella direzione opposta;
 - final limit: è uno switch meccanico di sicurezza che viene attivato se vengono oltrepassati i prelimit LIMP e LIMN. Qualora attivato l’azionamento viene disabilitato (funzione di sicurezza “power removal” EN954-1 cat. 3) sfruttando gli ingressi ridondanti “pwrr_a” e “pwrr_b” del motore azionamento. È possibile uscire da tale condizione realizzando esclusivamente un “bypass” di tale switch;
 - interruttore a chiave sul control box: permette di abilitare o disabilitare il motore-azionamento. Sfrutta il medesimo circuito del “final limit” descritto precedentemente.

Lo stato degli switch è leggibile in maniera inequivocabile tramite gli input/output dell’azionamento, riportati in tabella 1.1.

	I/O 0	I/O 1	I/O 2	I/O 3	PWR_A	PWR_B
LIMP	1					
LIMN		1				
EmStp			0	0	0	0
KeySw			1	0	0	0
FinLim			1	1	0	0

Tabella 1.1: Codifica dello stato degli switch

Quando l’azionamento compatto si trova nel campo dell’interruttore di fine corsa è possibile riportarlo in un campo di spostamento valido tramite uno spostamento manuale. Lo spostamento manuale deve essere eseguito in direzione negativa se ad attivarsi è stato il segnale dell’interruttore di finecorsa positivo LIMP e in direzione positiva se invece si è attivato LIMN.

In figura 1.2 è riportato lo schema a blocchi del sistema di controllo.

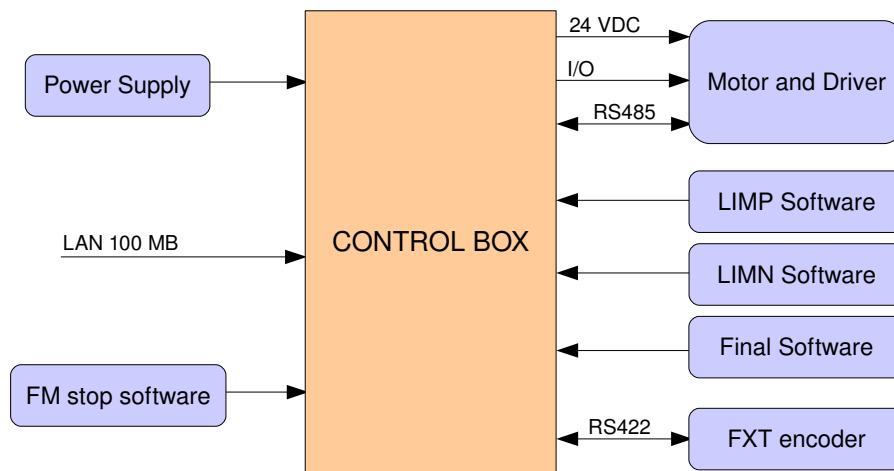


Figura 1.2: Schema a blocchi del sistema di controllo del derotatore

1.2.1 Specifiche tecniche

Corse utile	264° ($\pm 132^\circ$)
Velocità di rotazione	3.276 °/sec @3000 rpm
Accelerazione	2.67 °/sec ²
Ripetibilità	0.03 °
Risoluzione	0.0054622 °/step motore
Coppia max uscita	550 Nm
Alimentazione	24 VDC
Numero di giri max	3000 rpm
Numero di giri min	300 rpm
Coppia nominale	0.175 Nm
Step giro	12
Minimo spostamento	2 step

Tabella 1.2: Specifiche motore Berger Lahr - IcIA IFE

Risoluzione lineare	0.01 mm
Diametro montaggio traduttore	680.52 mm
Risoluzione angolare	0.00168°

Tabella 1.3: Specifiche traduttore esterno assoluto ELGO Electric EMAX

Rapporto riduzione riduttore n.1	60:1 $\eta = 90\%$
Rapporto riduzione riduttore n.2	10.25:1 $\eta = 70\%$
Rapporto riduzione riduttore n.3	8.933:1 $\eta = 90\%$
Rapporto riduzione totale	5494
Rendimento totale	60%
Coppia max uscita	550 Nm

Tabella 1.4: Specifiche trasmissione cinematica

Software limit	$\pm 132^\circ$
Pre limit switch (LIMP-LIMN)	$\pm 133^\circ.2$
Final limit software	$\pm 135^\circ$
Posizione di riferimento (LIMP + 100 step)	$+132^\circ.68$

Tabella 1.5: Posizione switch

1.3 Controllo Remoto del Motore-Azionamento

Il controllo remoto del motore-azionamento è possibile in due modalità:

1. con il software IclA Easy fornito dal produttore;
2. con il software ACS sviluppato, tema centrale di questo report.

I due software sono complementari e hanno differenti ambiti di utilizzo. IclA Easy viene utilizzato per la messa in servizio, la diagnosi approfondita, l'esecuzione di prove ed eventuale risoluzione di problemi di funzionamento, mentre il software ACS sviluppato gestisce la movimentazione del componente durante le osservazioni in multi-beam.

IclA Easy comunica con il motore-azionamento tramite interfaccia seriale, mentre il software ACS tramite LAN. I due software non possono quindi essere utilizzati in contemporanea ed è necessario configurare il convertitore LAN/RS485 con il software *Nport Administrator*. Per poter passare da una modalità di comunicazione all'altra si deve impostare l'*operating mode*. Per utilizzare IclA Easy l'*operating mode* dovrà essere impostato su *Real COM Mode* mentre per utilizzare il software ACS lo si dovrà impostare in *TCP Server Mode*, come mostrato in figura 1.3.

I principali comandi che possono essere impartiti al motore-azionamento tramite IclA Easy sono riportati in tabella 1.6.

Per informazioni dettagliate sui componenti rimandiamo alla documentazione fornita dal produttore [3] [2] [1].

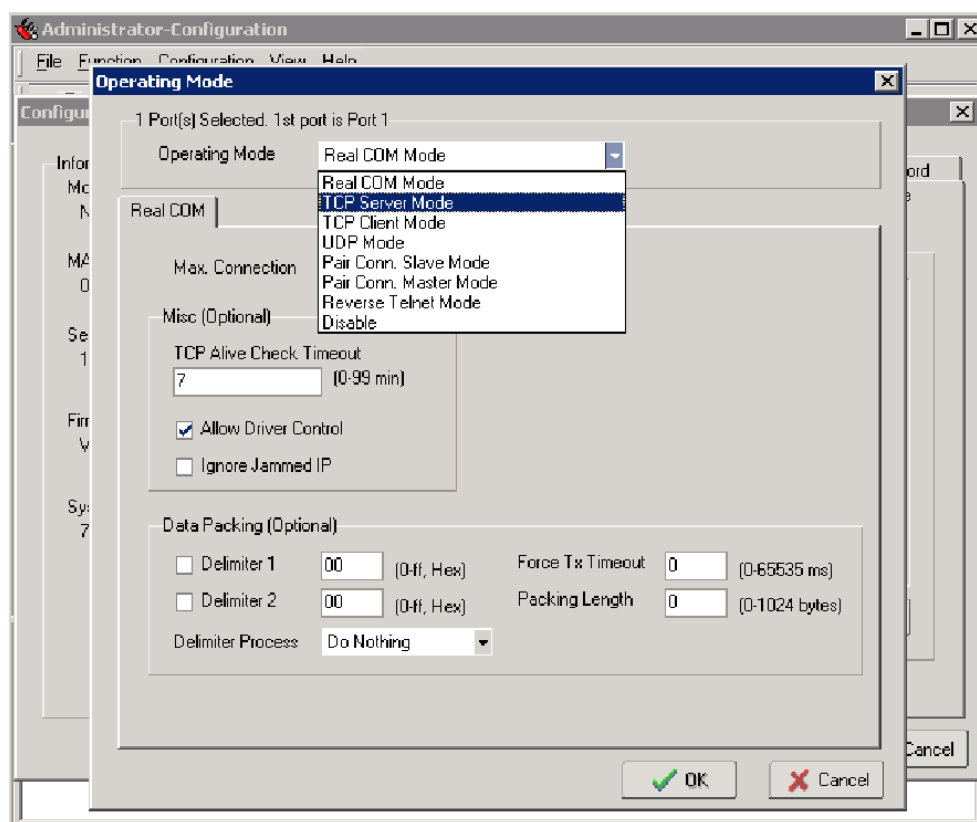


Figura 1.3: Configurazione dell'operating mode per l'utilizzo del software ACS.

Categoria	Nome	Funzione	Idx	SIdx	W	R
Coomands	driveCtrl	Drive Disable	28	1	0	
Coomands	driveCtrl	Drive Enable	28	1	2	
Coomands	driveCtrl	Drive QuickStop	28	1	4	
Coomands	driveCtrl	Drive Reset Fault	28	1	8	
Coomands	driveCtrl	QuickStop Release	28	1	16	
Homing	StartHome	Start Homing	40	1	1	
PTP	P_absPTP	Posizione Target	35	1	xxxx	
PTP	StatePTP	Posizione Target	35	2		xxxx
I/O	I0_act	I/O Status	33	1		xxxx
Status	P_act	Actual Position	31	6		xxxx
Status	driveStat	Drive Status	28	2		xxxx
Status	driveStat	Drive Status	28	3		xxxx

Tabella 1.6: Comandi base per il controllo remoto del motore-azionamento

1.4 Collocazione del Derotatore sull'Antenna

Il derotatore verrà collocato sul ricevitore multi-beam a 22 GHz, situato sul tamburo in fuoco secondario. In figura 1.4 il tamburo è indicato con il numero 26.

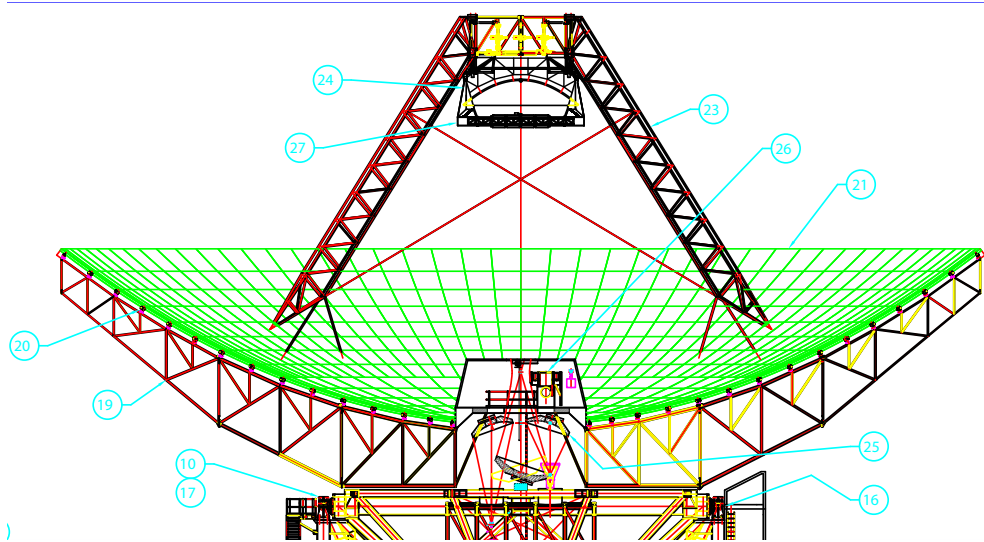


Figura 1.4: Dislocazione sull'antenna del derotatore (numero 26).

Capitolo 2

Software di Controllo del Derotatore

2.1 Panoramica su ACS

In questo capitolo verrà descritto il software di controllo del derotatore, sviluppato quando il componente era installato insieme al ricevitore multi-beam sul radiotelescopio di Medicina (BO). Il software è stato implementato utilizzando gli strumenti messi a disposizione del framework ACS (ALMA Common Software) e può essere visto come un componente passivo, inquanto la sua unica funzione è far ruotare il derotatore di un angolo fornitogli in ingresso. La logica che sta alla base della movimentazione (legata all'osservazione in corso) verrà delegata ad un componente attualmente in fase di studio.

Prima di addentrarci nei dettagli del software sviluppato chiariamo brevemente cosa è il framework ACS (ALMA Common Software). Atacama Large Millimeter Array (ALMA) è un progetto che coinvolge le organizzazioni astronomiche europee e del nord America e che prevede la realizzazione di almeno 64 antenne da 12 metri (situate a 5000 m di quota), nel deserto di Atacama in Cile. ACS fornisce una infrastruttura software comune a tutti i partner del progetto e consiste in una collezione di pattern e dei corrispondenti componenti che li implementano. Il cuore di ACS è un modello Distributed Objects (DOs) basato su CORBA. Il team che sarà responsabile dello sviluppo del sistema di controllo userà il DOs come base per i suoi componenti e dispositivi.

L'architettura di ACS è detta *Component/Container Model*. I container forniscono un ambiente nel quale i component vengono eseguiti, garantendo il supporto per servizi fondamentali come un sistema di logging, la gestione degli errori e degli allarmi, un database di configurazione e la gestione del ciclo

di vita dei component. In questo modo gli sviluppatori possono concentrarsi sulle funzionalità che dovrà avere il loro component piuttosto che su aspetti tecnici inerenti l'ambiente di esecuzione dei loro component. In figura 2.1 è mostrato uno schema esemplificativo di questa architettura.

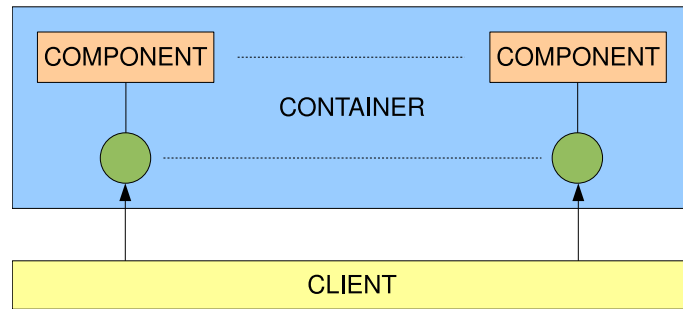


Figura 2.1: Component/Container Model

Uno dei punti di forza di ACS è la possibilità che viene data agli sviluppatori di implementare i propri component in un linguaggio a scelta tra C++, Python e Java. Per poter far ciò ACS definisce un IDL (Interface Definition Language), ovvero un linguaggio di descrizione delle interfacce dei component; ogni component quindi avrà una interfaccia descritta seguendo le specifiche dell'IDL e una implementazione in un linguaggio a scelta tra Python, C++ o Java.

Dopo questa breve panoramica sul framework ACS ci si potrebbe sentire un pò confusi. Per esemplificare ulteriormente si può pensare ad ACS come ad un intermediario tra l'hardware dell'antenna e l'operatore. Come intermediario ACS accede tramite i suoi component alle informazioni sensibili del sistema antenna; queste informazioni verranno rese disponibili da ACS mediante le interfacce pubbliche del componente, consentendo così ad altri software, detti client, di mostrare in modo amichevole all'operatore lo stato dell'antenna, tipicamente mediante delle intuitive interfacce grafiche.

2.2 Struttura del Software di Movimentazione del Derotatore

Il software sviluppato è costituito da un component (che comunica con il sistema derotatore) e da un client testuale che sostanzialmente utilizza le interfacce esportate dal component, permettendo così di poter effettuare dei rapidi test e comunicare in modo semplice ed intuitivo con il derotatore.

Il component ACS che si occupa della movimentazione del derotatore è stato implementato in C++, mentre il client è implementato in Python. Tutto il software descritto in questo capitolo è reperibile sul repository SVN, prevalentemente al path `/SRT/Servers/SRTMinorServo`, d'ora in poi chiamato *directory di lavoro*. La struttura dei file all'interno della directory di lavoro (box di colore ciano sulla sinistra) è schematizzata in figura 2.2.

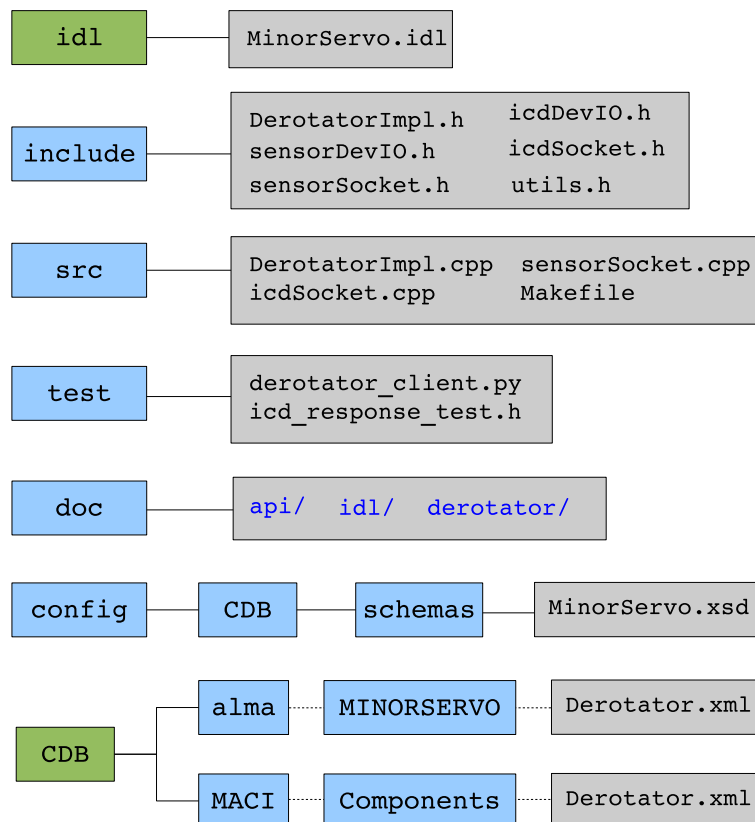


Figura 2.2: Gerarchia dei file del software ACS del derotatore

Le directory di colore verde in figura 2.2 sono esterne alla directory di lavoro. Diamo una breve descrizione della gerarchia di file e directory schematizzata in figura 2.2:

- **idl**: contiene il file che descrive l'interfaccia del component. Si trova su `Common/Interfaces/MinorServoInterface/`;
- **include**: contiene i file header dei sottosistemi del derotatore ed un file di libreria `utils.h` che definisce delle funzioni utilizzate per la trasformazione dei dati scambiati con il sistema derotatore;

- **src**: contiene le implementazioni dei componenti il sistema derotatore;
- **test**: contiene tre file, uno utilizzato come client (`derotator_test.py`), uno verifica che sia possibile attivare un canale di comunicazione con il derotatore (`socket_test.py`) ed un terzo (`icd_response_test.h`) utilizzato per simulare malfunzionamenti del sistema derotatore o per generare messaggi inattesi in risposta a certi comandi;
- **doc**: contiene i manuali forniti dai produttori dei componenti del derotatore, la documentazione automatica generata dai commenti presenti nel codice sorgente e formattati nel linguaggio *Doxygen*, una guida all'utilizzo del software;
- **config**: contiene lo schema xml del component.
- **CDB**: contiene la struttura di file e directory del CDB (Configuration Data Base). Si trova sulla directory **SRT**.

Fondamentalmente il software è suddiviso in due parti, una che si occupa della comunicazione con il sensore di posizione e l'altra che fa altrettanto con il motore-azionamento (ICD, Intelligent Compact Drive). I parametri tipici di un particolare derotatore vengono impostati configurando il CDB; nel listato 2.1 è mostrato uno snippet del file `Derotator.xml` relativo alla configurazione del derotatore dell'antenna di Medicina.

```
6 <Derotator
7   xmlns="urn:schemas-cosylab-com:derotator:1.0"
8   xmlns:baci="urn:schemas-cosylab-com:BACI:1.0"
9   xmlns:cdb="urn:schemas-cosylab-com:CDB:1.0"
10  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
11
12  PS_IP="192.168.51.7"
13  PS_PORT="4002"
14  PS_TIMEO="5000000"
15  PS_REFERENCE="688671"
16  PS_CF="0.001683884"
17
18  ICD_IP="192.168.51.7"
19  ICD_PORT="4001"
20  ICD_TIMEO="5000000"
21  ICD_REFERENCE="0"
22  ICD_CF="0.0054622">
```

Listato 2.1: Parametri di configurazione del sistema derotatore di Medicina

In figura 2.3 è schematizzato il derotatore ed i 7 ricevitori. I riferimenti ai punti cardinali sono relativi ad una vista del derotatore dal basso, cioè guardando il derotatore secondo il verso di puntamento dell'antenna.

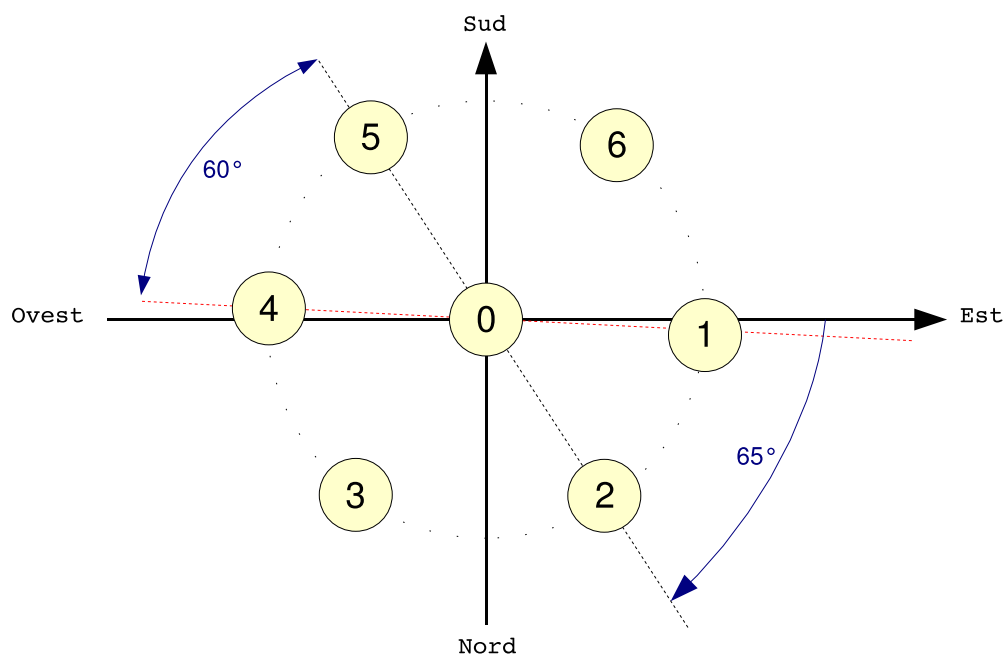


Figura 2.3: Vista dal basso (verso di puntamento dell'antenna) del derotatore.

Assumiamo come positive le rotazioni che avvengono in senso orario, sempre secondo lo schema riportato in figura 2.3. Con questa convenzione per poter portare il ricevitore numero 6 al posto del ricevitore numero 1 il derotatore deve ruotare di un angolo di 60° .

La figura 2.3 mostra il derotatore nella sua posizione di zero assoluto, ovvero quando la posizione fisica del motore-azionamento è pari a 0. In questo modo quando il derotatore si trova in posizione zero la congiungente tra il ricevitore numero 4 e quello numero 1 (indicata con una linea rossa tratteggiata in figura 2.3) è inclinata di 5° rispetto alla direzione ovest/est¹. Poichè il riferimento per l'utente sarà l'asse ovest/est, allora se il derotatore si trova nella sua posizione di zero e l'utente imporsta una nuova posizione pari a 65° il derotatore ruoterà di 60° .

¹Lo zero del derotatore (relativo al sistema di riferimento dell'utente) è configurabile sul CDB.

Inoltre, poichè il derotatore ha una escursione massima di 262° (le due posizioni più estreme sono $\pm 131^\circ$), in virtù dell'offset di 5° i posizionamenti che l'utente può impartire sono compresi nell'intervallo $[-126^\circ, 136^\circ]$.

2.4 Utilizzo del Client

Attualmente è disponibile un solo client testuale per l'interazione con il component, ma è in fase di sviluppo anche un client grafico. Per poter utilizzare il client la prima cosa da fare è avviare ACS ed il container. Questo può essere fatto eseguendo prima lo script `./acsconf` (modificando eventualmente i percorsi presenti al suo interno) e successivamente avviando ACS (`acsStart`) ed il container *MinorServoContainer* con il comando `getup_mscontainer`.

A questo punto è possibile avviare il client:

```
6 $ python derotator_client.py
```

Comparirà il seguente menu:

```
#-----
0. Exit
1. Get sensor position
2. Get ICD position
3. Set ICD position
4. Show position difference (ICD - Sensor)
5. Show ICD status

Enter a choice:
#-----
```

Questo è piuttosto autoesplicativo e, come si può intuire:

1. stampa la posizione letta dal sensore di posizione;
2. stampa la posizione letta dal motore-azionamento;
3. permette di impostare un valore per la posizione del motore-azionamento²;
4. stampa la differenza tra la posizione letta dal sensore e quella del motore-azionamento;

²Il component ha come parametro in ingresso l'angolo target che dovrà avere il derotatore (rispetto ad una posizione di riferimento) al termine della movimentazione. Questo angolo verrà convertito dal component nel corrispondente numero di step e successivamente comunicato al motore-azionamento.

5. mostra lo stato del motore-azionamento;

Lo stato del motore-azionamento viene visualizzato nel seguente modo:

```
#-----  
# STATUS:  
#  POLL_ERR:      0  
#  WRONG_RFLAG:   0  
#  CMDERR:        0  
#  WRONG_RCODE:   0  
#  IMS_FLTSIG:    0  
#  EMS_SIGN:      0  
#  WARNING:       0  
#  INIT_ERR:      0  
#  QS_ACTIVE:     0  
#  NOT_READY_SO:  0  
#  NOT_OP_ENABLE: 0  
#  MOVING:        0  
#  MOVING_ERR:    0  
#  WRONG_POSITION: 0  
#  
# Fine!  
#-----
```

Le variabili di stato possono assumere solo i valori 0 o 1, ed il loro significato è il seguente:

- **POLL_ERR**: quando questo bit è alto allora non è possibile mettersi in comunicazione con il motore-azionamento. Prima di iniziare una comunicazione infatti è necessario trasmettere un comando di *polling* che attiva un canale con il motore-azionamento e, se il polling non va a buon fine, allora non è possibile la trasmissione.
- **WRONG_RFLAG**: quando questo bit è alto c'è un errore di comunicazione; ogni risposta da parte del motore azionamento deve impostare un flag uguale a quello impostato nella word destinata al motore. Se il flag della word diretta al motore ed il flag della risposta di questo ultimo non coincidono allora questo bit viene messo a 1.
- **ERRCMD**: questo bit viene posto a 1 quando il motore-azionamento non può eseguire il comando che gli è stato impartito; la causa di questa impossibilità di esecuzione viene registrata nel file di log.
- **WRONG_RCODE**: questo bit viene posto a 1 quando la risposta è corrotta, ovvero viene ricevuta una risposta che non può essere decodificata;

- **IMS_FLTSIG**: questo è il bit **FltSig** della risposta (IMS sta per Internal Monitoring Signals); questo viene posto a 1 quando viene riscontrato un errore dall'*internal monitoring*.
- **EMS_FLTSIG**: questo è il bit **Sign** della risposta (EMS sta per External Monitoring Signals); questo viene posto a 1 quando viene riscontrato un errore dall'*external monitoring*.
- **WARNING**: questo bit viene posto a 1 quando il motore segnala un warning.
- **INIT_ERR**: quando c'è un *Initialization Error* dovuto al fatto che l'azionamento non è stato avviato questo bit viene posto a 1. Il motore azionamento infatti torna in uno stato di *switch off* appena non vi sono canali di comunicazioni attivi ed in questo caso è necessario dare un comandi di *driveEnable* prima di poter iniziare una comunicazione.
- **QS_ACTIVE**: questo bit viene posto a 1 quando vi è un *Quick Stop* attivo.
- **NOT_READY_S0**: dopo un comando di **Drive Enable** lo stato operativo dovrebbe essere *Ready to Switch On*; se non si trova in questo stato questo bit viene posto a 1. L'*operating status* inoltre va considerato solo dopo un comando di **Drive Enable**.
- **NOT_OP_ENABLE**: quando questo bit è posto a 1 significa che non è possibile impostare un nuovo valore di posizione perchè il motore-azionamento di trova in uno stato di *Not Operation Enable*.
- **MOVING**: Questo bit viene posto a 1 quando il motore-azionamento è in movimento.
- **MOVING_ERR**: questo è il bit *x_err* e viene posto a 1 quando avviene un errore durante la movimentazione.
- **WRONG_POSITION**: viene posto a 1 quando si verifica un errore durante la lettura della posizione.

Bibliografia

- [1] Elgo Electric. EMAX & FEMAX series, Installationi Manual, Magnetic Linear Encoders with 0.01 mm Resolution . Documento: 799000006.
- [2] Berger Lahr GmbH. Field bus protocol for the intelligent compact drive IclA IFx, RS485, 2002. Documento: 0098441113187.
- [3] Berger Lahr GmbH. Manuale del prodotto, Azionamento compatto intelligente con interfaccia bus di campo e motore EC, IclA IFE, 2006. Documento: 0098441113324, Edizione: V1.04, 05.2006.
- [4] Marco Morsiani. Descrizione del Derotatore per il multi-beam a 22 GHz, GAI05 Memo Series . Versione Preliminare, 26 Febbraio 2008.

Elenco delle figure

1.1	Panoramica dell'apparecchio	7
1.2	Schema a blocchi del sistema di controllo del derotatore	11
1.3	Configurazione dell'operating mode per l'utilizzo del software ACS.	13
1.4	Dislocazione sull'antenna del derotatore (numero 26).	14
2.1	Component/Container Model	16
2.2	Gerarchia dei file del software ACS del derotatore	17
2.3	Vista dal basso (verso di puntamento dell'antenna) del dero- tatore.	19

Elenco delle tabelle

1.1	Codifica dello stato degli switch	10
1.2	Specifiche motore Berger Lahr - IcIA IFE	11
1.3	Specifiche traduttore esterno assoluto ELGO Electric EMAX .	11
1.4	Specifiche trasmissione cinematica	12
1.5	Posizione switch	12
1.6	Comandi base per il controllo remoto del motore-azionamento	13