

Manuale d'uso

HM58x EC

Elenco sezioni

- 1 Norme di sicurezza
- 2 Identificazione
- 3 Quick reference (TwinCAT)
- 4 Connessioni elettriche
- 5 Interfaccia EtherCAT®
- 6 Programmazione



1 - Norme di sicurezza

Sicurezza

- Durante l'installazione e l'utilizzo del dispositivo osservare le norme di prevenzione e sicurezza sul lavoro previste nel proprio paese;
- l'installazione e le operazioni di manutenzione devono essere eseguite da personale qualificato, in assenza di tensione e parti meccaniche in movimento;
- utilizzare il dispositivo esclusivamente per la funzione per cui è stato costruito: ogni altro utilizzo potrebbe risultare pericoloso per l'utilizzatore;
- alte correnti, tensioni e parti in movimento possono causare lesioni serie o fatali;
- non utilizzare in ambienti esplosivi o infiammabili;
- il mancato rispetto delle norme di sicurezza o delle avvertenze specificate in questo manuale è considerato una violazione delle norme di sicurezza standard previste dal costruttore o richieste dall'uso per cui lo strumento è destinato;
- Lika Electronic s.r.l. non si assume alcuna responsabilità per eventuali danni o lesioni derivanti dall'inosservanza delle norme di sicurezza da parte dell'utilizzatore.

Avvertenze elettriche

- Effettuare le connessioni elettriche esclusivamente in assenza di tensione;
- rispettare le istruzioni relative alle connessioni riportate nella sezione "Connessioni elettriche";
- in conformità alla normativa 2004/108/CE sulla compatibilità elettromagnetica rispettare le seguenti precauzioni:
 - prima di maneggiare e installare il dispositivo, eliminare la presenza di carica elettrostatica dal proprio corpo e dagli utensili che verranno in contatto con il dispositivo;
 - alimentare il dispositivo con tensione stabilizzata e priva di disturbi, se necessario, installare appositi filtri EMC all'ingresso dell'alimentazione;
 - utilizzare sempre cavi schermati e possibilmente "twistati";
 - non usare cavi più lunghi del necessario;
 - evitare di far passare il cavo dei segnali del dispositivo vicino a cavi di potenza;
 - installare il dispositivo il più lontano possibile da eventuali fonti di interferenza o schermarlo in maniera efficace;
 - per garantire un funzionamento corretto del dispositivo, evitare l'utilizzo di apparecchiature con forte carica magnetica in prossimità dell'unità;
 - collegare la calza del cavo e/o la custodia del connettore e/o il corpo del dispositivo a un buon punto di terra; assicurarsi che il punto di terra sia privo di disturbi. Il collegamento a terra può essere effettuato sul lato dispositivo e/o sul lato utilizzatore; è compito dell'utilizzatore valutare la soluzione migliore da adottare per minimizzare i disturbi.



Avvertenze meccaniche

- Montare il dispositivo rispettando rigorosamente le istruzioni riportate nella sezione "Istruzioni di montaggio";
- effettuare il montaggio meccanico esclusivamente in assenza di parti meccaniche in movimento;
- non disassemblare il dispositivo;
- non eseguire lavorazioni meccaniche sul dispositivo;
- dispositivo elettronico delicato: maneggiare con cura; evitare urti o forti sollecitazioni sia all'asse che al corpo del dispositivo;
- utilizzare il dispositivo in accordo con le caratteristiche ambientali previste dal costruttore,
- encoder con asse sporgente: utilizzare giunti elastici per collegare encoder e motore; rispettare le tolleranze di allineamento ammesse dal giunto elastico.

2 - Identificazione

Il dispositivo è identificato dal codice e dal numero di serie stampati sull'etichetta e attraverso i documenti di trasporto dello stesso. Per dettagli relativi alle caratteristiche elettriche fare riferimento al catalogo del prodotto.

3 - Quick reference (TwinCAT)

3.1 Configurazione su TwinCAT

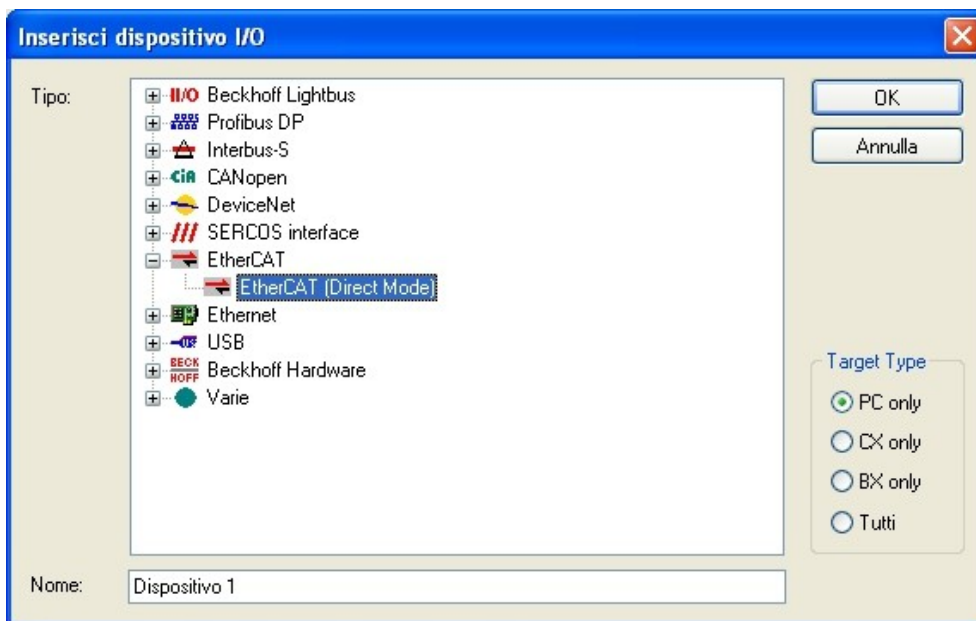
Impostazione scheda di rete

Avviare **TwinCAT System Manager**.

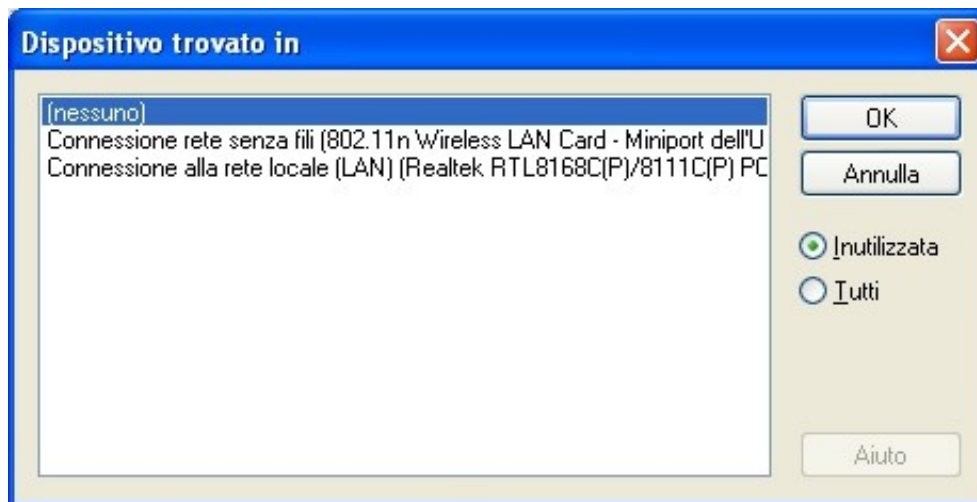
Nella finestra a sinistra estendere l'albero e selezionare **Dispositivi I/O**; quindi premere il tasto destro del mouse, aprire il menù a tendina e selezionare il comando **Aggiungi dispositivo...**



Nella finestra **Inserisci dispositivo I/O** selezionare il dispositivo **EtherCAT (Direct Mode)** e confermare premendo il pulsante **OK**.

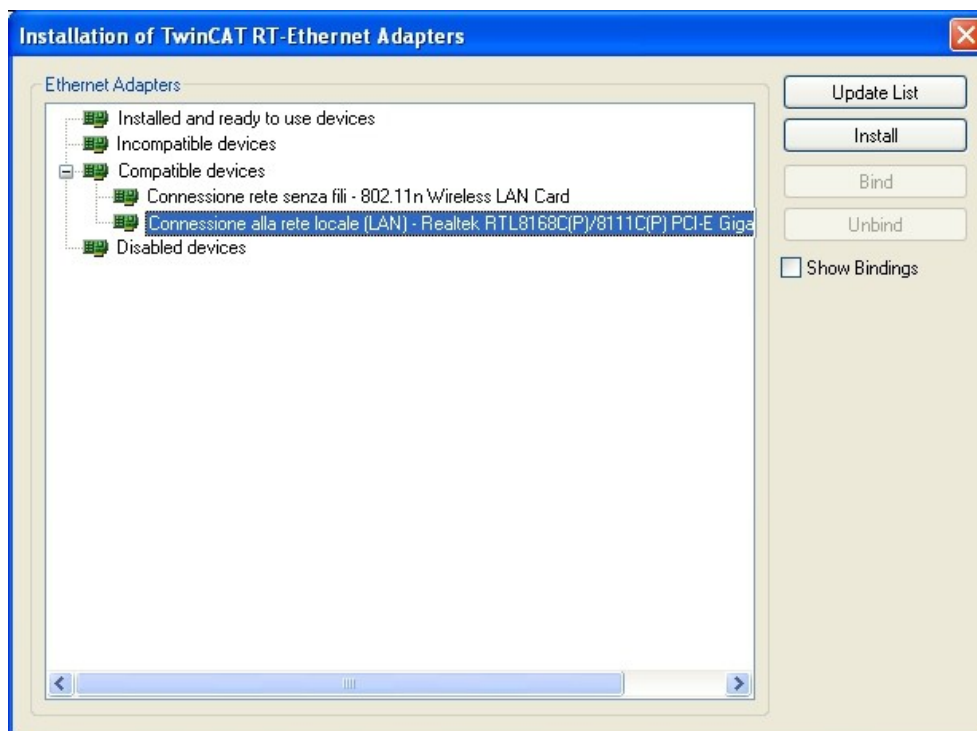


Se vi sono schede di rete installate apparirà una finestra simile a quella di seguito riportata.



Selezionare la scheda di rete che si vuole utilizzare e confermare la scelta premendo il pulsante **OK**.

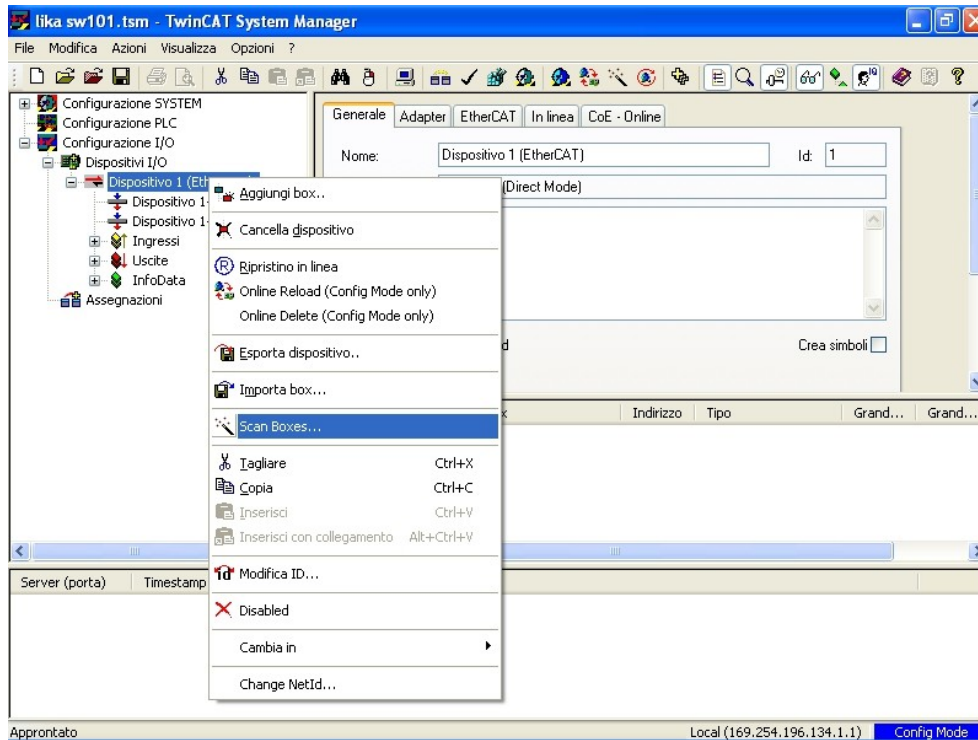
Se invece non vi sono schede di rete installate bisogna prima installarne una. Per fare questo aprire il menu **Opzioni** nella finestra principale di **TwinCAT System Manager** e selezionare poi il comando **Show Real Time Ethernet Compatible Devices...**



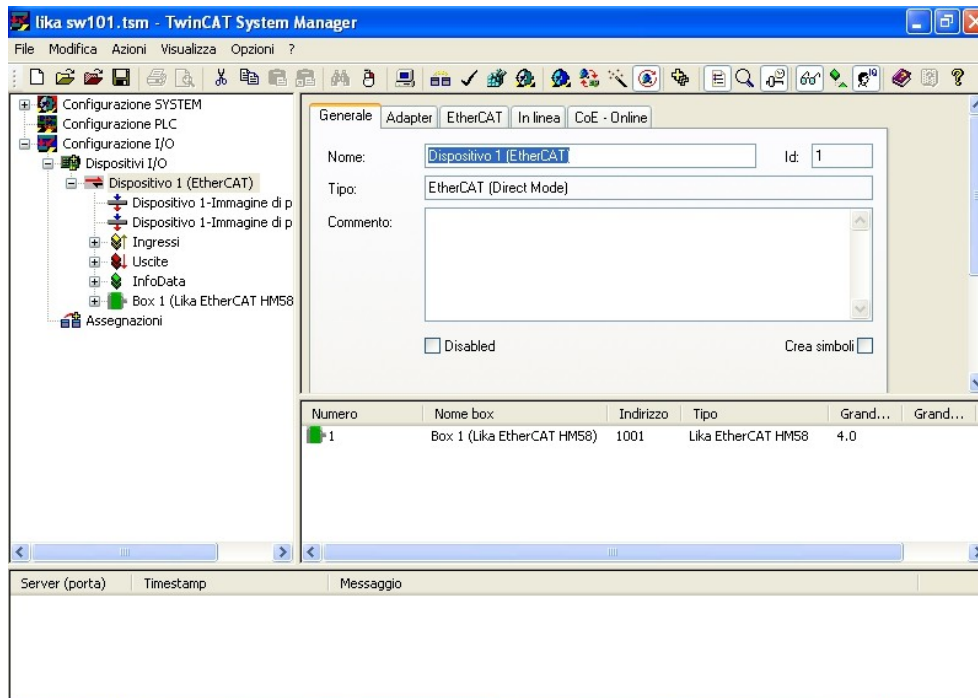
Nella finestra **Installation of TwinCAT RT – Ethernet Adapter** che si apre selezionare **Compatible devices** e scegliere poi la scheda di rete che si vuole installare. Confermare premendo il pulsante **Install**.

Aggiungere Moduli Input/Output (Box)

Se i dispositivi sono già collegati alla rete e sono alimentati, cliccare il tasto destro del mouse sulla voce **Dispositivo 1 (EtherCAT)** nella finestra principale di TwinCAT System Manager e premere il comando **Scan Boxes...**

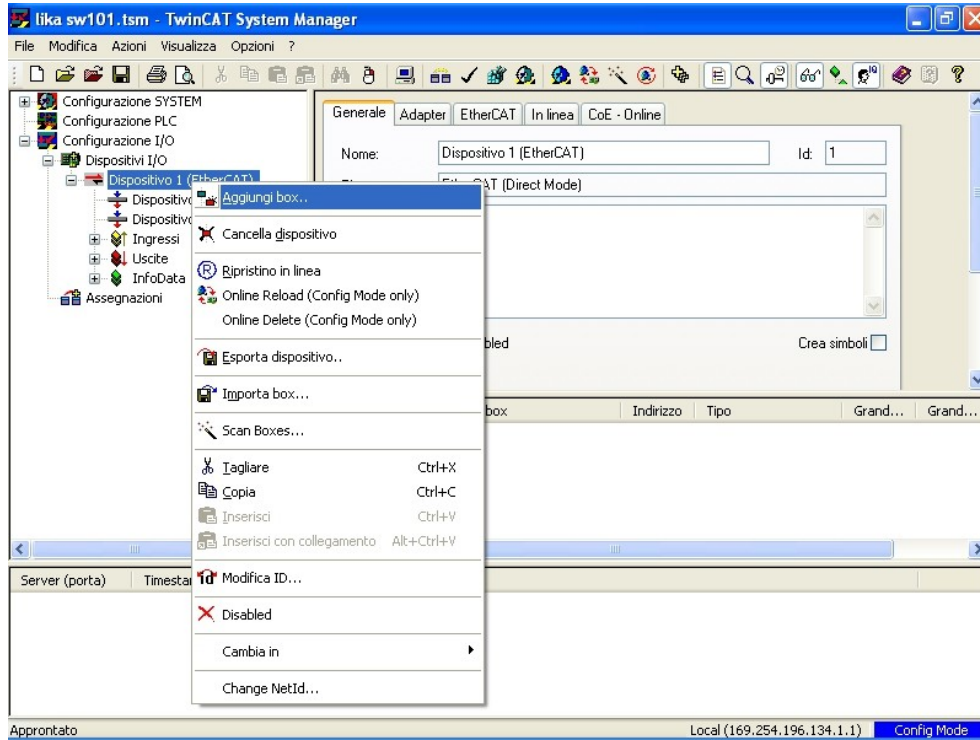


Al termine si avrà una situazione simile a quella descritta nella Figura seguente.



Se invece i dispositivi non sono già collegati alla linea è necessario utilizzare il file **Lika_Hx58_EC_Vx.xml** fornito con l'encoder (si veda il supporto informatico fornito con l'apparecchiatura oppure all'indirizzo www.lika.it > **PRODOTTI > ROTACOD > Hx58x EC**).

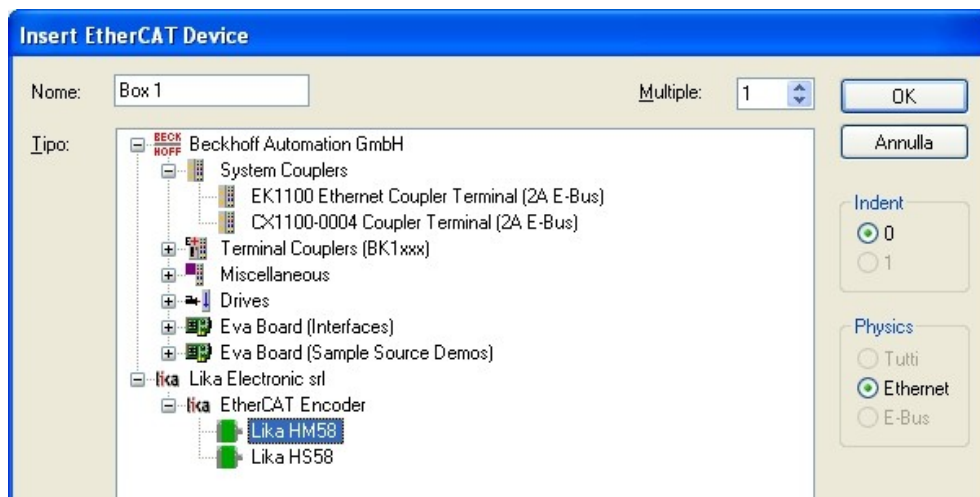
Nella finestra principale di **TwinCAT System Manager** selezionare **Dispositivo 1 (EtherCAT)**, premere il tasto destro del mouse e nel menu a tendina selezionare il comando **Aggiungi box...**



Nella finestra **Insert EtherCAT Device** che appare selezionare **Lika Electronic srl** e poi **EtherCAT Encoder**; scegliere quindi il tipo di encoder che si vuole installare:

- Lika HM58: encoder multigiuro;
- Lika HS58: encoder monogiuro.

Premere il pulsante **OK** per confermare.

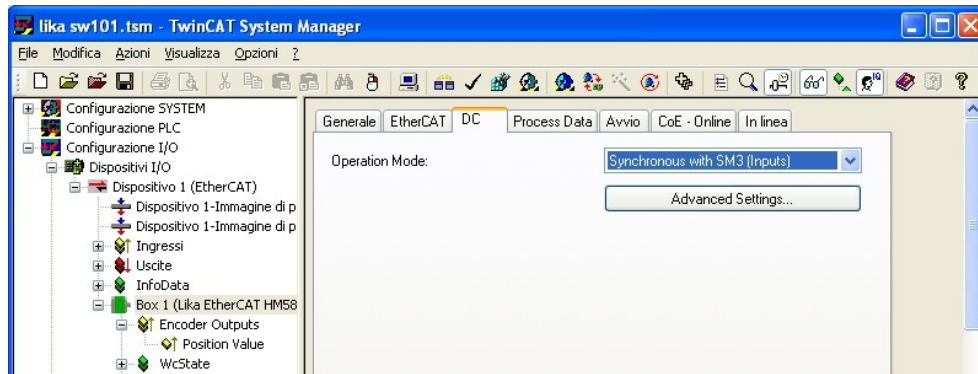


3.2 Impostazioni modalità di funzionamento

Sincrono con SM3

Nella finestra principale di **TwinCAT System Manager** selezionare il **Box (Lika EtherCAT HM58 o HS58)**: a destra compare la finestra principale dedicata alla configurazione del Box; accedere alla pagina **DC**.

Selezionare **Synchronous with SM3 (Inputs)** nel box **Operation Mode**.

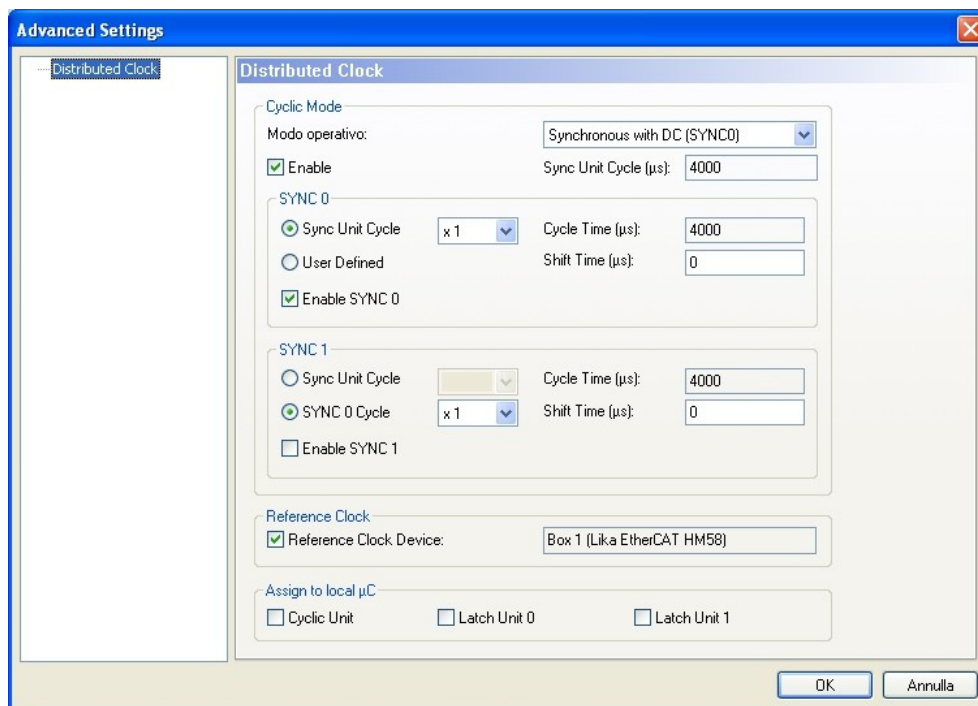


Sincrono con DC (SYNCO)

Nella finestra principale di **TwinCAT System Manager** selezionare il **Box (Lika EtherCAT HM58 o HS58)**: a destra compare la finestra principale dedicata alla configurazione del Box; accedere alla pagina **DC**.

Selezionare **Synchronous with DC (SYNCO)** nel box **Operation Mode**.

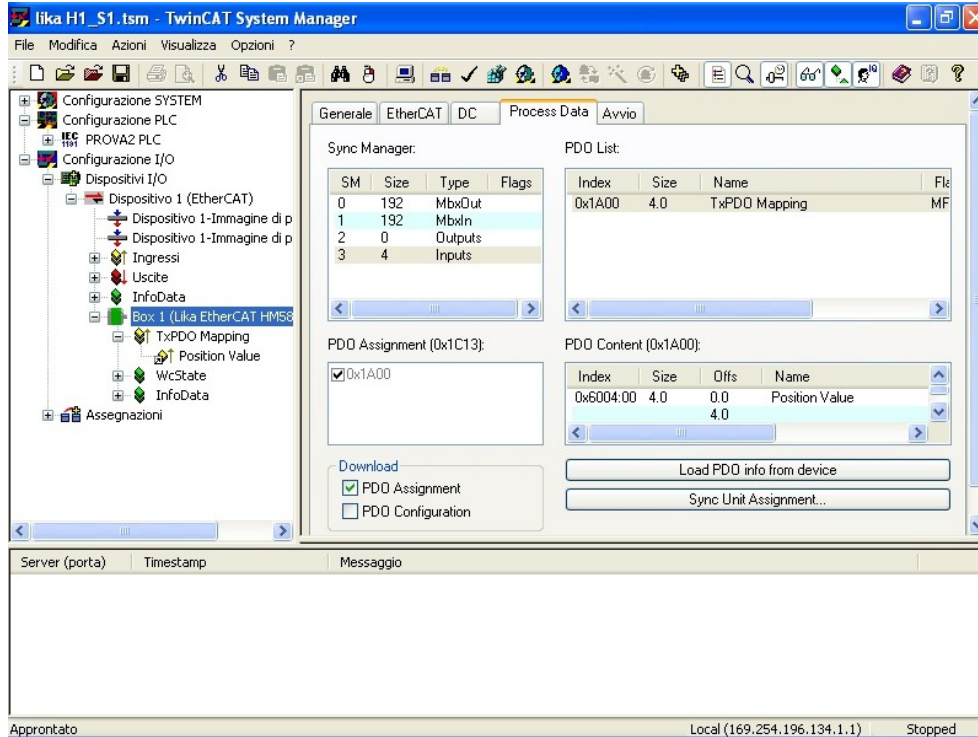
Premere poi il pulsante **Advanced Settings...**



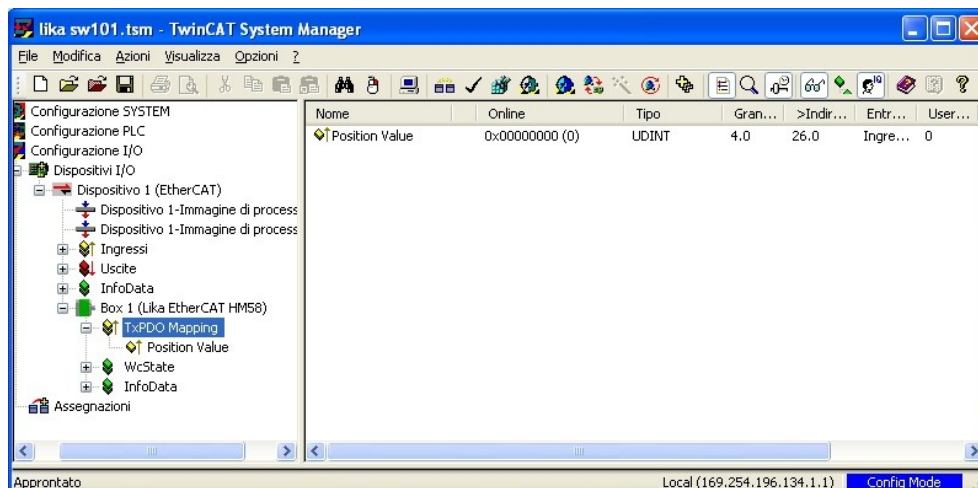
Nella sezione denominata **SYNCO** impostare il tempo di ciclo di SYNCO nel box **Sync Unit Cycle**; il tempo è dato come multiplo (o sottomultiplo) del valore specificato nel campo in alto a destra **Sync Unit Cycle (µs)**.

3.3 Process Data Objects

Nella finestra principale di **TwinCAT System Manager** selezionare il **Box (Lika EtherCAT HM58 o HS58)**; a destra compare la finestra principale dedicata alla configurazione del Box; accedere alla pagina **Process Data**. In questa pagina sono visualizzati i dati di processo (PDO Mapping).

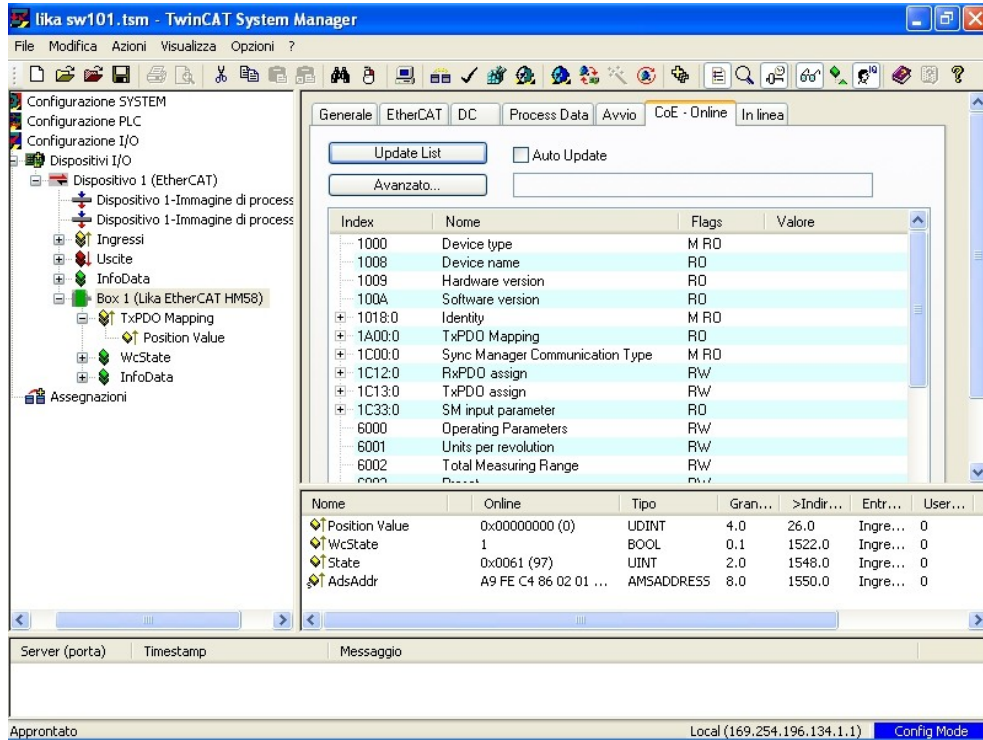


E' possibile visualizzare i dati di processo anche cliccando su **TxPDO Mapping** nella finestra principale di **TwinCAT System Manager** estendendo l'albero relativo al **Box (Lika EtherCAT HM58 o HS58)**; i dati sono visualizzati nella parte destra della finestra.

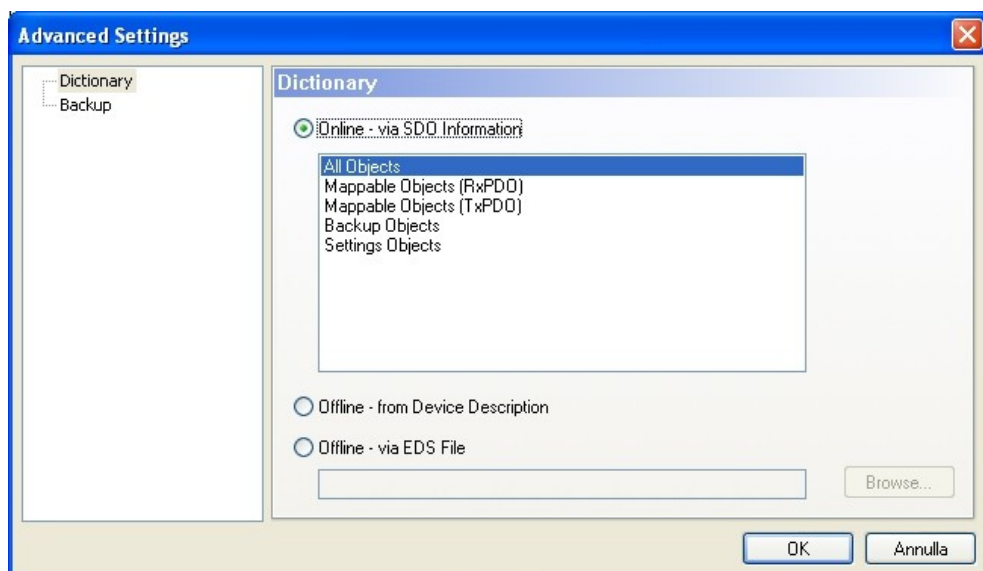


3.4 Dizionario Oggetti COE

Nella finestra principale di **TwinCAT System Manager** selezionare il **Box (Lika EtherCAT HM58 o HS58)**: a destra compare la finestra principale dedicata alla configurazione del Box; accedere alla pagina **CoE – Online**. In questa pagina sono visualizzati gli oggetti del dizionario.



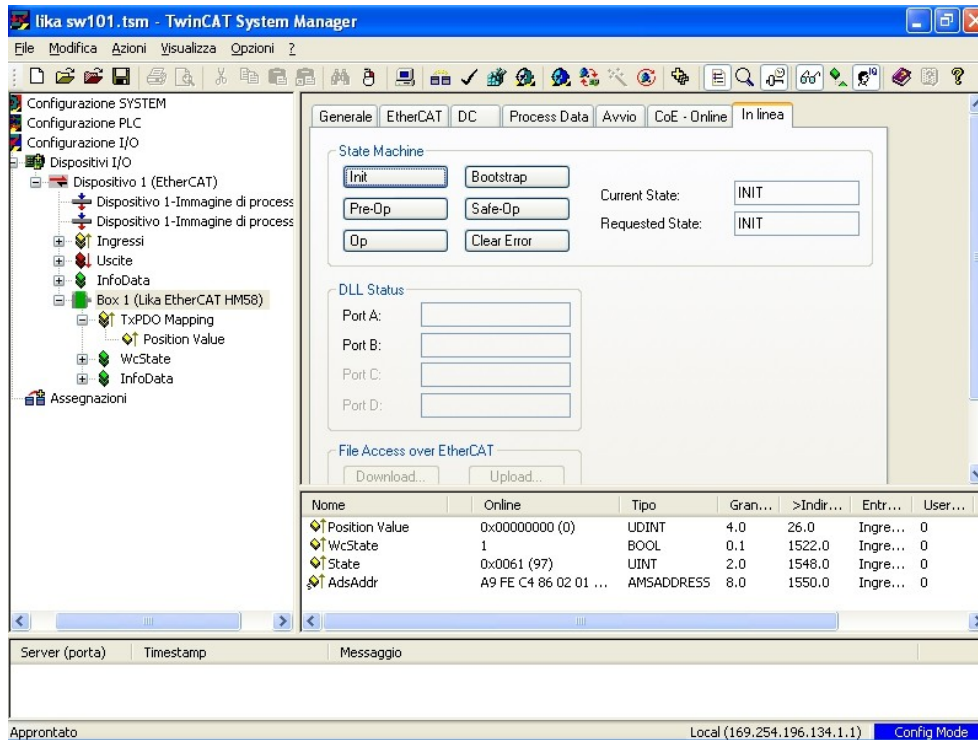
Se si vogliono visualizzare i valori direttamente online leggendoli dall'encoder cliccare il pulsante **Avanzato...**: si apre la finestra **Advanced Settings**.



Nel riquadro di sinistra selezionare **Dictionary** e nella pagina **Dictionary** a destra scegliere l'opzione **Online – via SDO Information**; premere infine il pulsante **OK**.

3.5 Dati Online

Nella finestra principale di **TwinCAT System Manager** selezionare il **Box (Lika EtherCAT HM58 o HS58)**: a destra compare la finestra principale dedicata alla configurazione del Box; accedere alla pagina **In linea**. In questa pagina è visualizzato lo stato dell'encoder.



Per visualizzare in tempo reale i dati di processo dell'encoder cliccare il pulsante **Safe-OP** per visualizzare gli input; cliccare il pulsante **OP** per visualizzare anche gli output.



ATTENZIONE

La struttura dei Data Objects (PDO e SDO) prevede l'inserimento dei dati dal meno significativo (LSB) al più significativo (MSB).

Nell'utilizzo di TwinCAT invece i Data byte devono essere inseriti dal'MSB all'LSB.

Nell'utilizzo di TwinCAT anche le stringhe devono essere inserite al contrario:

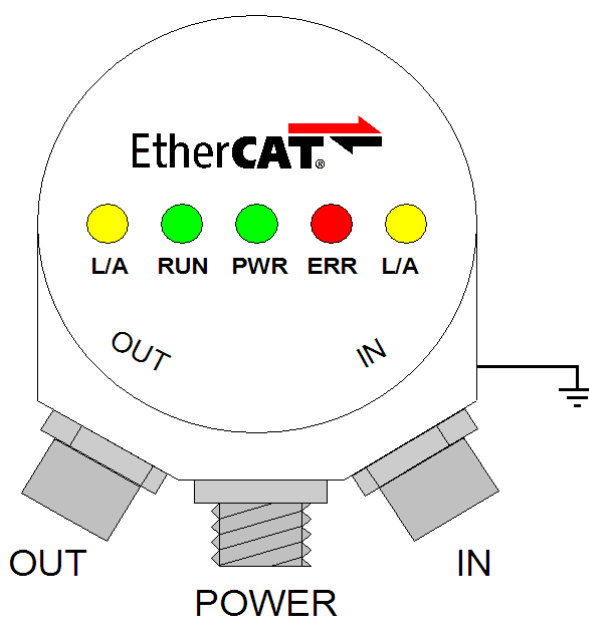
- lettura parametri di default: Data byte = 64 61 6F 6C hex = "daol" in codifica ACII (cioè "load" al contrario);
- salvataggio dati: Data byte = 65 76 61 73 hex = "evas" in codifica ASCII (cioè "save" al contrario).

4 - Connessioni elettriche



ATTENZIONE: non rimuovere il coperchio dall'encoder. Alcuni componenti interni potrebbero danneggiarsi.

Collegare la calza del cavo e/o la custodia del connettore e/o il corpo del dispositivo a un buon punto di terra; assicurarsi che il punto di terra sia privo di disturbi. Il collegamento a terra può essere effettuato sul lato dispositivo e/o sul lato utilizzatore; è compito dell'utilizzatore valutare la soluzione migliore da adottare per minimizzare i disturbi. Si consiglia di effettuare il collegamento a terra il più vicino possibile all'encoder.



4.1 Connettori M12

Il coperchio è provvisto di tre connettori M12 con pin-out secondo lo standard EtherCAT. Pertanto è possibile utilizzare cavi EtherCAT standard disponibili in commercio.

I connettori in ingresso (IN) e in uscita (OUT) non sono intercambiabili.

M12	BUS IN e BUS OUT	POWER
(vista lato contatti)	 codifica D femmina	 codifica A maschio

Pin	Descrizione	Descrizione
1	Tx Data +	+10VDC +30VDC
2	Rx Data +	n.c.
3	Tx Data -	0 VDC
4	Rx Data -	n.c.

n.c.= non connesso.

4.2 Indicatori LED

Cinque LED sulla parte posteriore della protezione del collegamento mostrano la condizione di funzionamento dell'interfaccia EtherCAT secondo la seguente tabella:

Link/ Activity (giallo)	indica lo stato del link fisico e l'attività su questo link
OFF	stato: porta chiusa, link: SI, activity: N.A.
Lampeggio veloce	stato: porta aperta, link: SI, activity: SI
ON	stato: porta aperta, link: SI, activity: NO
RUN (verde)	indica lo stato dell' EtherCAT State Machine (ESM)
OFF	l'encoder è nello stato INIT
Lampeggio lento	l'encoder è nello stato PRE-OPERATIONAL
Singolo flash	l'encoder è nello stato SAFE-OPERATIONAL
ON	l'encoder è nello stato OPERATIONAL
PWR (verde)	indica lo stato dell'alimentazione
OFF	encoder non alimentato
ON	encoder alimentato correttamente
ERR (rosso)	indica la presenza di errori
OFF	nessun errore
Lampeggio veloce	errore caricamento parametri da memoria allo start-up
Lampeggio veloce	errore salvataggio parametri
Lampeggio lento	configurazione non valida dei parametri
Singolo flash	errore locale (vedi ETG1000.6)
Doppio flash	watchdog timeout
ON	errore memoria e controller ESC inattivo

5 - Interfaccia EtherCAT®

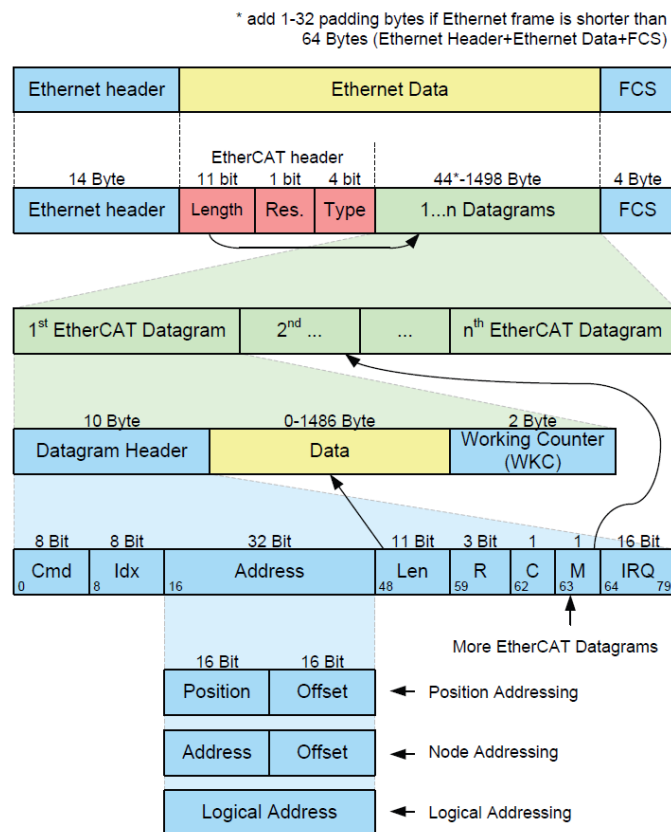
5.1 Nozioni di base sul protocollo EtherCAT®

Il protocollo EtherCAT si appoggia direttamente ai dataframes Ethernet standard per il trasferimento dei dati; inoltre dal punto di vista hardware ha il vantaggio di non richiedere l'utilizzo di master dedicati in quanto vengono utilizzate le schede di rete normalmente utilizzate in qualsiasi rete Ethernet.

Un bus EtherCAT può essere visto come una singola e grande subnet Ethernet che invia e riceve dati appoggiandosi alla struttura dei dataframes Ethernet senza alterarne la struttura.

All'interno di questa subnet possono tuttavia essere presenti solo un controller master e un certo numero di slave EtherCAT, ma nessun Ethernet controller che invii dati in rete.

Struttura del frame Ethernet con EtherCAT:



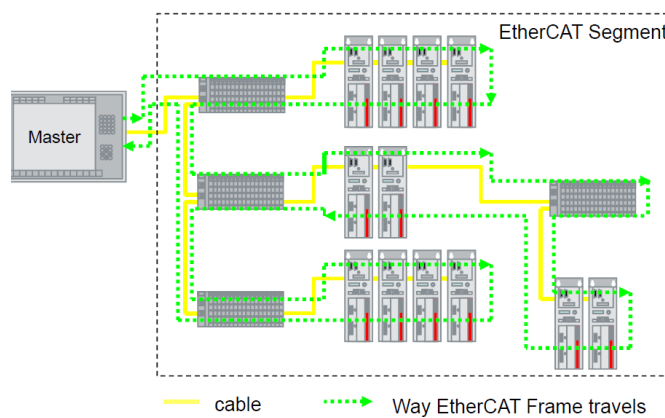
I dati all'interno del frame Ethernet vengono trasmessi tra master e slave sotto forma di dati di processo (PDO - process data objects). Ciascun PDO ha associato un indirizzo verso uno o più slave; l'associazione dati+indirizzo (unitamente ad altri elementi tra cui una checksum di validazione) forma un telegramma EtherCAT (Datagram). Un frame EtherCAT può contenere più telegrammi e spesso un ciclo completo di controllo può richiedere anche più di un frame.

5.1.1 Trasferimento dati

Generalmente in una struttura di trasferimento dati a Bus il controller master invia una richiesta dati in linea e attende che questi vengano elaborati e poi restituiti da ogni nodo slave; questo comporta una difficoltà nel rispettare le caratteristiche tipiche di un sistema "real time" in quanto il controller master acquisisce i dati in istanti diversi dai diversi slave.

Con EtherCAT questo problema viene superato in quanto i dati vengono processati "on-the-fly" in tempo reale utilizzando un unico frame di chiamata per tutti gli slave della rete.

Infatti lo stesso pacchetto di richiesta dati dal master circola su tutti gli slave e ciascun dispositivo, se indirizzato, inserisce o preleva i dati richiesti e trasmette il frame al dispositivo successivo per ulteriori elaborazioni. L'ultimo slave provvede a restituire a tutti gli slave e infine al master il frame completo con tutti i dati richiesti.



Tutto ciò è reso possibile dalla struttura full-duplex della rete EtherCAT che presenta linee separate per la trasmissione e la ricezione dei dati. Inoltre l'elaborazione del protocollo avviene all'interno dell'hardware ed è pertanto indipendente dalla CPU e dalla parte di elaborazione software.

5.1.2 ISO/OSI Layer model

OSI Layers		
7	Application Layer (AL)	IEC 61158-5/-6 Type 12
6	empty*	
5		
4		
3		
2	Data Link Layer (DLL)	IEC 61158-3/-4 Type 12
1	Physical Layer (PhL)	IEC 61158-2 Type 12

* empty: significa che la funzionalità del layer esiste, ma non è mostrata esplicitamente.

5.1.3 Topologia

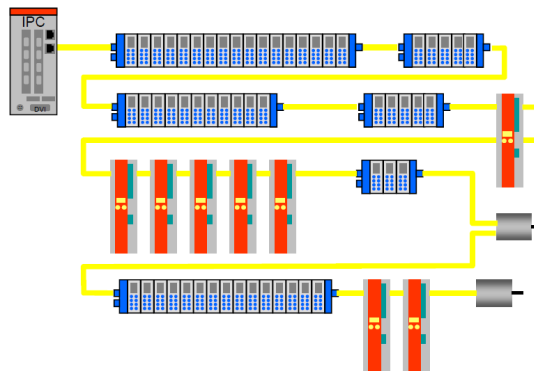
E' possibile utilizzare diverse topologie di connessione: line, tree, daisy chain + drop lines, star; esse possono essere implementate in qualunque combinazione. La lunghezza massima del cavo tra due slave è di 100 m e si utilizza un cablaggio standard Ethernet.

La scelta dell'una o dell'altra topologia viene fatta in base alle caratteristiche della struttura dell'impianto e alla semplicità di cablaggio.

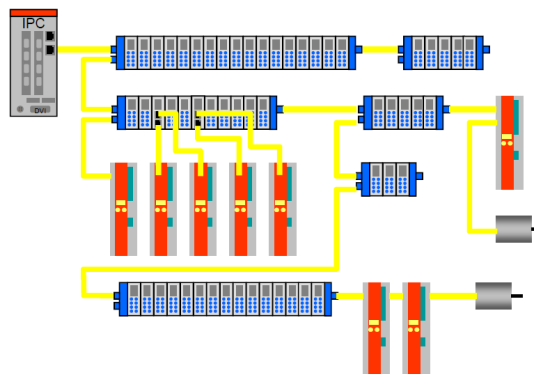
All'interno di una rete EtherCAT si possono collegare fino a 65535 dispositivi.

Alcuni esempi sono riportati nelle Figure qui sotto:

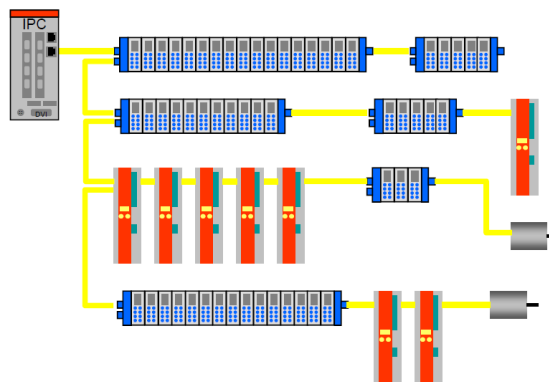
Topologia LINE:



Topologia TREE:



Topologia DAISY CHAIN con drop lines:



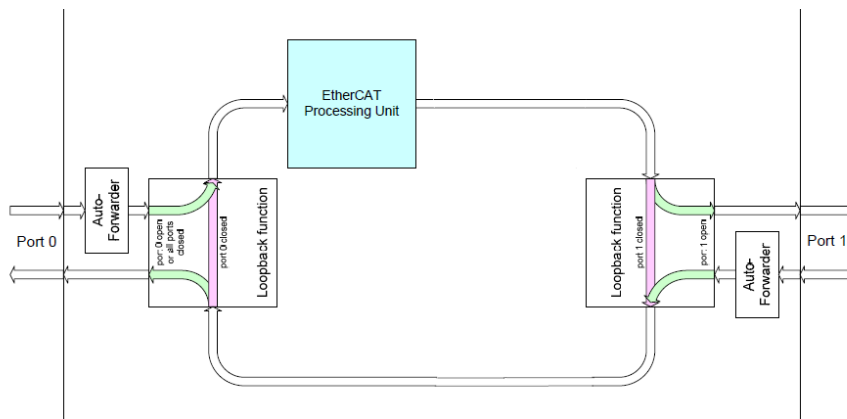
5.1.4 Line Termination

Non c'è la necessità di usare terminazioni di linea in quanto la chiusura della rete EtherCAT avviene in modo automatico; ogni slave infatti è in grado di rilevare o meno la presenza di altri slave a valle.

Uno slave EtherCAT è in grado di rilevare la presenza del segnale sulla linea uscente (outgoing line Port 0) o sulla linea di ritorno (Return line Port 1).

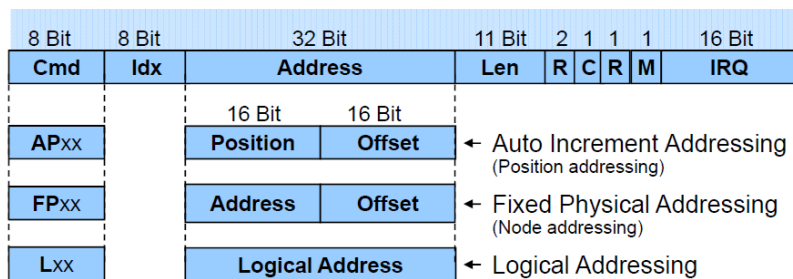
Se lo slave non rileva più il segnale sulla sua linea di ritorno allora cortocircuita il segnale TX della linea uscente con il segnale RX della linea di ritorno; in questo modo un telegramma ricevuto sulla linea uscente viene processato e rimandato indietro attraverso il TX della linea di ritorno.

Lo slave continua a inviare sul TX della linea uscente un "carrier signal" o un telegramma; non appena viene ripristinato lo slave a valle viene di nuovo rilevato un segnale su RX della linea di ritorno per cui il cortocircuito viene rimosso e i telegrammi vengono inviati al TX della linea uscente.



5.1.5 Indirizzo dispositivo

Non c'è bisogno di assegnare un indirizzo fisico al dispositivo (tramite per es. dip-switch) in quanto l'indirizzamento dello slave avviene in modo automatico all'accensione del sistema durante la fase di scan della configurazione hardware.



L'indirizzamento è a 32 bit e può essere fatto nei seguenti modi:

- Auto Increment Addressing = 16 bit rappresentano la posizione fisica dello slave nella rete e 16 bit vengono usati per indirizzare la memoria locale; quando riceve il frame, lo slave incrementa la posizione fisica e il dispositivo che riceve Position = 0 è quello indirizzato;
- Fixed Addressing = 16 bit rappresentano l'indirizzo fisico dello slave nella rete e 16 bit vengono usati per indirizzare la memoria locale;
- Logical Address = lo slave non è indirizzato individualmente, ma legge o scrive dati in una sezione dell'intero spazio di 4Gbyte disponibile.

5.1.6 Modalità di comunicazione

Gli encoder Lika con interfaccia EtherCAT supportano le seguenti modalità di funzionamento:

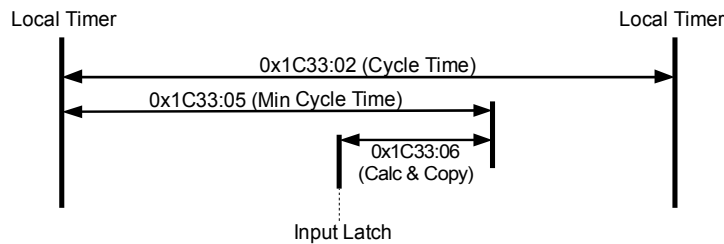
- FreeRun: modalità asincrona;
- SM3 event: modalità sincrona;
- DC: modalità a clock distribuiti.

Per un sistema che necessita di alte prestazioni real-time (anello chiuso) si può usare la modalità DC; nel caso in cui la necessità di real-time sia un requisito di bassa importanza si possono usare SM3 e Freerun.

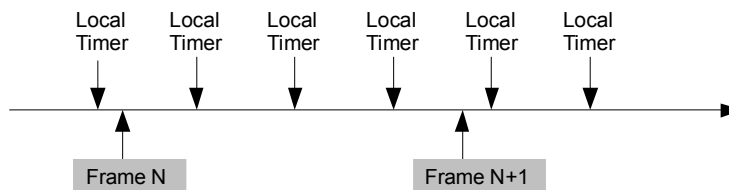
In particolare un parametro di riferimento è rappresentato dal "Jitter" ossia la variazione nel tempo dell'istante di campionamento del dato; in altre parole, il dato campionato dal controllore è reso disponibile nella memoria DPRAM dell'EtherCAT controller dopo un certo tempo con una fascia di incertezza pari al jitter.

FreeRun

Modalità di funzionamento asincrona in cui la quota encoder è prelevata direttamente dal frame EtherCAT inviato dal master; l'aggiornamento della posizione è effettuato da un timer interno al controllore ogni 100 microsecondi.



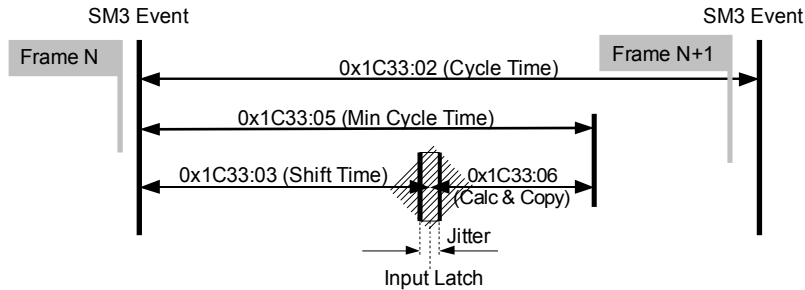
Questa modalità di funzionamento presenta un jitter di campionamento elevato che al massimo può valere 100 µs e può essere usata con tempi di ciclo sensibilmente maggiori rispetto al jitter se si vuole garantire un sistema real-time sufficientemente prestante.



Descrizione	Min	Typ	Max	
Jitter	0		100	µs
Tempo di ciclo	1000		64000	µs

Sincrono con SM3

In questa modalità i dati sono campionati e successivamente copiati nel buffer Sync Manager non appena i dati precedenti sono stati letti dal master (evento SM); quindi i nuovi valori campionati risultano sincroni con le letture da parte del master.



I nuovi dati saranno letti dal master col ciclo successivo rispetto a quello che ha generato l'evento SM per cui, se il tempo di ciclo è troppo grande, avremo dei valori relativamente vecchi per un sistema real-time.

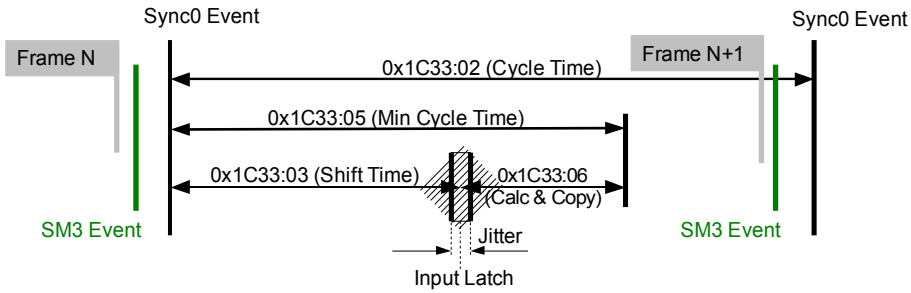
Il vantaggio principale è che l'aggiornamento dei dati avviene in modo sincrono con la lettura da parte del master.

Descrizione	Min	Typ	Max	
Jitter	0		7,2	µs
Tempo di ciclo	62,5		64000	µs

Sincrono con DC

In questa modalità i dati sono campionati e successivamente copiati nel buffer in corrispondenza del segnale SYNC0 generato dall'unità di capture/compare dell'ESC.

Il tempo necessario per queste operazioni è definito nell'oggetto 1C33hex, in particolare da "Shift Time" (1C33hex, sub3) e "Calc and Copy time" (1C33hex, sub6).

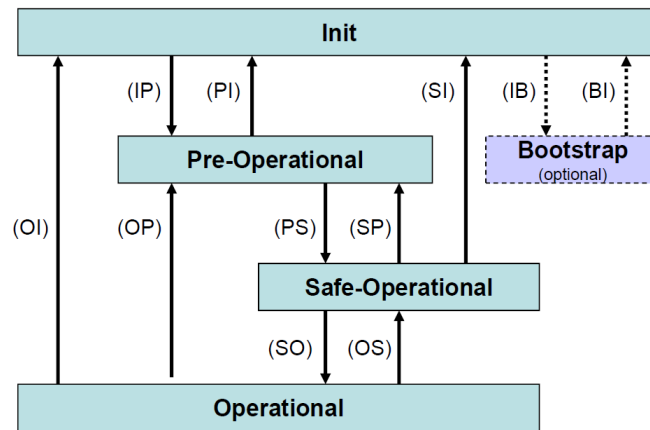


Un fattore importante da considerare è il jitter che si ha nel campionamento tra due dati successivi.

Il vantaggio principale di questa modalità è quello di avere una relazione diretta tra l'istante di campionamento del dato e il tempo assoluto del sistema per cui, conoscendo gli shift time dei vari dispositivi, si può avere un'esatta fotografia del sistema in un determinato istante (con incertezza pari al jitter).

Descrizione	Min	Typ	Max	
Jitter	0	100	200	µs
Tempo di ciclo	62,5		64000	µs

5.1.7 EtherCAT State Machine (ESM)



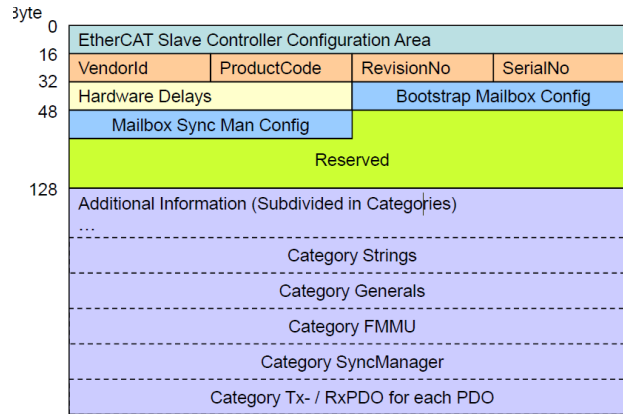
Lo slave EtherCAT è una macchina a stati; le caratteristiche di comunicazione e di funzionamento dipendono dallo stato in cui si trova lo slave:

- **INIT:** è lo stato di default dopo l'accensione; in questo stato non c'è comunicazione diretta tra master e slave sull'Application Layer; è inizializzata una serie di registri di configurazione ed effettuata la configurazione dei sync manager;
- **PRE-OPERATIONAL:** in questo stato è attiva la mailbox; il master e lo slave possono usare la mailbox e i corrispondenti protocolli per scambiare specifici parametri di inizializzazione per l'applicazione. Non è possibile lo scambio di dati di processo;
- **SAFE-OPERATIONAL:** in questo stato master e slave possono scambiarsi dati di processo solo per quanto riguarda gli input, mentre gli output rimangono nello stato "safe";
- **OPERATIONAL:** in questo stato master e slave possono scambiarsi dati di processo, sia input che output.

5.1.8 Slave configuration

La configurazione per la comunicazione dello slave (Configurazione Sync Manager, indirizzi, modi di sincronizzazione, PDO mapping, ecc.) può avvenire tramite il relativo file XML (EtherCAT Slave Information ESI) oppure tramite caricamento da EEPROM (Slave Information Interface SII).

Contenuto EEPROM (SII):



5.1.9 Temporizzazione e sincronizzazione

La caratteristica principale di EtherCAT è quella di rappresentare in maniera quasi ideale un sistema real-time.

Per far ciò il master deve essere in grado di sincronizzare contemporaneamente tutti i dispositivi slave in modo tale da avere un sistema in cui tutti i nodi hanno lo stesso tempo di riferimento; questo è realizzato mediante l'uso di "clock distribuiti".

Uno degli slave (di solito il primo) contiene il master clock di riferimento ed è incaricato di sincronizzare i clock degli altri dispositivi in rete; il controller master invia periodicamente uno speciale telegramma di sincronizzazione in cui lo slave di riferimento scrive il proprio "current time". Questo telegramma viene inviato poi a tutti gli altri slave che provvedono in questo modo a risincronizzare il proprio clock in modo da evitare fenomeni di deriva.

Questa sincronizzazione del tempo di riferimento è di fondamentale importanza per avere una fotografia istantanea del sistema e poter così effettuare delle azioni simultanee in applicazioni particolarmente delicate quali la coordinazione nelle operazioni di controllo assi.

L'EtherCAT Slave Controller (ESC) dispone inoltre di un'unità di comparazione in grado di generare segnali di sincronismo in direzione del controllore locale (SYNCO o interrupt) che permettono al controllore stesso di sincronizzare il proprio tempo locale con quello dello slave.

Sync Manager

Il Sync Manager è responsabile della sincronizzazione del trasferimento dei dati tra Master e Slave ed evita che la stessa zona di memoria sia scritta contemporaneamente da diversi eventi.

Ci sono due modalità di sincronizzazione:

- 3-Buffer Mode;
- 1-Buffer Mode.

L'inizializzazione della modalità usata avviene attraverso il file XML o caricando direttamente i dati da EEPROM (SII).

Buffered Mode (3-Buffer Mode)

In questa modalità si garantisce un accesso ai nuovi dati in qualsiasi momento e i dati sono accessibili da entrambe le parti (Master ECAT e ESC) senza nessuna restrizione di tempistiche. Sono necessari tre buffer (tre aree di memoria consecutive); un buffer è sempre disponibile per la scrittura da parte di ESC e un buffer contiene sempre dati aggiornati in lettura da parte del master.

E' solitamente usato per la comunicazione dei dati di processo.

Mailbox Mode (1-Buffer Mode)

In questa modalità si deve utilizzare un "handshake" tra master e slave in quanto viene utilizzato un unico buffer di memoria che può essere letto o scritto; la scrittura da parte del master (o da parte dello slave) può avvenire solo quando il buffer è vuoto ossia la controparte (slave o master) ha completamente letto i dati contenuti nel buffer; analogamente per quanto riguarda la lettura che deve avvenire solo quando il buffer è stato completamente scritto dalla controparte.

5.2 CANopen Over EtherCAT (CoE)

Gli encoder Lika sono dispositivi Slave e utilizzano il protocollo "CANopen Over EtherCAT (CoE)" per il trasferimento dei dati; in particolare supportano il "Device profile for encoders", Classe 2.

Per ogni specifica omessa relativa al protocollo CANopen® fare riferimento ai documenti "CiA Draft Standard 301" e "CiA Draft Standard 406" disponibili sul sito www.can-cia.org.

Per ogni specifica omessa relativa al protocollo EtherCAT fare riferimento ai documenti "ETG.1000 EtherCAT Specification" disponibili sul sito www.ethercat.org.

5.2.1 File XML

Gli encoder EtherCAT sono forniti con un loro file XML **Lika_Hx58_EC_Vx.xml** (si veda il supporto informatico fornito con l'apparecchiatura oppure all'indirizzo www.lika.it > PRODOTTI > ROTACOD > Hx58x EC).

Il file XML deve essere installato sul dispositivo master.

5.2.2 Tipi di messaggi

La modalità CoE prevede la seguente struttura dell'EtherCAT Datagram:

Mbx Header	CoE Cmd			Cmd specific data
type = 3	Number	res	Type	
6 byte	9 bit	3 bit	3 bit	0...1478 byte

Mbx Header = 3 modalità CoE

Number = 0 in caso di messaggi SDO
 ≠ 0 in caso di messaggi PDO, specifica il tipo di servizio

res bit riservati

Type = 0 valore riservato
 = 1 messaggi Emergency
 = 2 richiesta SDO
 = 3 risposta SDO
 = 4 PDO trasmessi (TxPDO)
 = 5 PDO ricevuti (RxPDO)
 = 6 Remote transmission request dei PDO trasmessi
 = 7 Remote transmission request dei PDO ricevuti
 = 8 informazioni SDO
 = 9...15 valori riservati

Cmd specific data messaggi PDO: sono i dati di processo, es. posizione
 messaggi SDO: contiene il frame standard CANopen

"Type" è considerato trasmesso (tx) o ricevuto (rx) dal punto di vista del nodo Slave.

5.2.3 Process Data Objects (PDO)

I messaggi PDO sono usati per trasmettere dati di processo in tempo reale; i dati trasmessi sono definiti nei PDO Mapping e gestiti dai Sync Manager PDO Mapping.

5.2.4 Service Data Objects (SDO)

I messaggi SDO sono trasferiti tramite Mailbox (dati a bassa priorità); non sono supportati il Segmented SDO Service e l'SDO Complete Access (trasferimento di dati con dimensioni contenute e un subindex per volta). Sono utilizzati per accedere al "Dizionario Oggetti" per leggere o modificare i parametri in esso contenuti.

"CoE Cmd type" = 2 o 3.

Struttura del "Cmd specific data":

Cmd specific data				
SDO control	Index	Sub index	Data	Data optional
8 bit	16 bit	8 bit	32 bit	1...1470 byte

SDO control comando CANopen per SDO standard.

Index indice del parametro da leggere o scrivere.

Sub index secondo indice del parametro da leggere o scrivere.

Data valore letto o scritto del parametro.

Data optional byte aggiuntivi a Data per parametri con più di 4 data byte.

I possibili valori di Index e Sub index sono specificati nel "Dizionario oggetti".

5.2.5 Dizionario oggetti

Di seguito sono riportati gli oggetti implementati nel dispositivo, per ciascuno è indicato:

Index-subindex Nome oggetto [tipo variabile, attributo]

- Index e subindex sono espressi in formato esadecimale.
- Attributo:
 - ro = oggetto accessibile in sola lettura;
 - rw = oggetto accessibile in lettura e scrittura.

Struttura oggetti Unsigned16:

Data byte	
byte 4	byte 5
LSByte	MSByte

Struttura oggetti Unsigned32:

Data byte			
byte 4	byte 5	byte 6	byte 7
LSByte	MSByte

Oggetti standard

1000-00 Tipo di dispositivo [Unsigned32, ro]

Default = 0001 0196h = encoder monogiro, DS 406

0002 0196h = encoder multigiro, DS 406

1008-00 Nome del dispositivo [String, ro]

Contiene il nome del dispositivo.

Default = "HS58xx" = encoder monogiro

"HM58xx" = encoder multigiro

1009-00 Versione hardware [String, ro]

Contiene la versione hardware del dispositivo.

100A-00 Versione software [String, ro]

Contiene la versione software del dispositivo.

1010-01 Salvataggio parametri [Unsigned32, rw]

Oggetto utilizzato per eseguire il salvataggio di tutti i parametri nella memoria non volatile. Scrivere "save" nei data byte:

Master → Encoder

Cmd specific data							
Cmd	Index		Sub	Data			
23	10	10	01	73	61	76	65

Encoder → Master (conferma)

Cmd specific data							
Cmd	Index		Sub	Data			
60	10	10	01	00	00	00	00

1011-01 Parametri di default [Unsigned32, rw]

Oggetto utilizzato per caricare tutti i parametri di default. Scrivere "load" nei data byte:

Master → Encoder

Cmd specific data							
Cmd	Index		Sub	Data			
23	11	10	01	6C	6F	61	64

Encoder → Master (conferma)

Cmd specific data							
Cmd	Index		Sub	Data			
60	11	10	01	00	00	00	00



NOTA

Per conservare i parametri di default impostati, eseguire "Salvataggio parametri" (vedi oggetto 1010h).

1018 Informazioni di identificazione

- **01 Identificativo del costruttore** [Unsigned32, ro]
- **02 Codice prodotto** [Unsigned32, ro]
- **03 Numero revisione** [Unsigned32, ro]
- **04 Numero seriale** [Unsigned32, ro]

1A00-01 Mappatura PDO [Unsigned32, ro]

In questo oggetto è mappata la posizione dell'encoder, secondo le specifiche DS406.

Default = 6004 0020h

1C00 Sync Manager supportati

- **01** SM MailBox Receive (SM0) [Unsigned8, ro] Default = 1
- **02** SM MailBox Send (SM1) [Unsigned8, ro] Default = 2
- **03** SM PDO output (SM2) [Unsigned8, ro] Default = 3
- **04** SM PDO input (SM3) [Unsigned8, ro] Default = 4

1C12-00 Sync Manager RxPDO assegnati [Unsigned8, ro]

Per questi dispositivi non è prevista la ricezione di PDO.

Default = 0

1C13-01 Sync Manager TxPDO assegnati [Unsigned32, ro]

Questi dispositivi prevedono l'uso dei TxPDO per l'invio della posizione.

Default = 0000 1A00hex

1C33 Parametri Sync Manager

Alcuni subindex dell'oggetto 1C33hex sono calcolati in modo dinamico e dipendono dalla configurazione dell'encoder (risoluzione impostata, direzione conteggio, ecc.) e dal modo di sincronizzazione attivo (SM o DC).

- **01 Tipo di sincronismo** [Unsigned16, rw] Default = 1
 - 0: FreeRun
 - 1: Sincronizzato con SM3
 - 2: Sincronizzato con DC SYNC0
- **02 Cycle time** [Unsigned32, ro]

Questo parametro dipende dal tipo di sincronismo impostato:

 - se "FreeRun": tempo fra due campionamenti di quota (timer interno);
 - se "Sincronizzato con SM3": tempo minimo tra due eventi SM3;
 - se "Sincronizzato con DC SYNC0": tempo di ciclo di Sync0.
- **03 Shift Time** [Unsigned32, ro]

Tempo tra l'evento di sincronizzazione e l'istante in cui viene fatto il latch per acquisire il dato dell'encoder. Questo parametro è calcolato dinamicamente.
- **04 Sincronismi supportati** [Unsigned16, ro] Default = 7
 - Bit 0: FreeRun (supportato);
 - bit 1: Sincrono con SM3 (supportato);
 - bit 2: Sincrono con DC SYNC0 (supportato).
- **05 Minimum cycle time** [Unsigned32, ro]

Durata massima del tempo di ciclo interno dell'encoder. Questo parametro è calcolato dinamicamente in base ai parametri operativi e alla posizione encoder.
- **06 Calc and Copy time** [Unsigned32, ro]

Tempo necessario al controllore per effettuare tutti i calcoli interni sul dato campionato e copiarlo nella memoria dell'ESC. Questo parametro è calcolato dinamicamente in base ai parametri operativi e alla posizione encoder.

Oggetti profilo encoder
6000-00 Parametri operativi [Unsigned16, rw]

Bit	Funzione	bit = 0	bit = 1
0	Direzione conteggio	Orario	Antiorario
1	non usato		
2	Funzione di scaling	Disabilitato	Abilitato
3...15	non usato		

Default = 0000h

- Con la definizione di "Direzione conteggio" si stabilisce in quale senso ruotare l'albero encoder, in modo che il valore in uscita dall'encoder incrementi quando l'albero ruota in senso orario oppure antiorario. Il senso di rotazione è stabilito guardando l'encoder dall'estremità dell'albero.
- Funzione di scaling: se disabilitato l'encoder utilizza la propria risoluzione fisica (vedi oggetto 6501h e 6502h); se abilitato utilizza la risoluzione impostata negli oggetti 6001h e 6002h con la seguente relazione:

$$\text{posTx} = \frac{\text{obj_6001}}{\text{obj_6501}} \cdot \text{posReale} \leq \text{obj_6002}$$

6001-00 Informazioni per giro [Unsigned32, rw]

Questo oggetto definisce il numero di informazioni per giro desiderate.

Attivo se il bit 2 dell'oggetto 6000h è impostato a "1".

Per evitare salti di quota verificare che $\frac{\text{obj_6501}}{\text{obj_6001}}$ sia un valore intero.

E' possibile impostare solo valori minori o uguali al numero di "Informazioni per giro fisiche" (vedi dati di targa).

6002-00 Risoluzione totale [Unsigned32, rw]

Questo oggetto definisce la risoluzione totale desiderata.

Attivo se il bit 2 dell'oggetto 6000h impostato a "1".

E' possibile impostare solo valori minori o uguali alla "Risoluzione totale fisica" (vedi dati di targa).

Esempio

Encoder multigirotto HM5816/16384EC-6.

Risoluzione:

- Informazioni per giro fisiche: "obj_6501" = 65536 inf./giro (2^{16})
- Numero di giri fisico: "obj_6502" = 16384 giri (2^{14})
- Risoluzione totale fisica: = 1073741824 (2^{30})

si desidera impostare 2048 inf./giro x 1024 giri:

1 -Attivare la funzione di scaling: "obj_6000", bit 2 = "1"

2 -Informazioni per giro: "obj_6001" = 2048 (0000 0800h)

3 -Risoluzione totale: "obj_6002" = 2048 x 1024 = 2097152 (0020 0000h)

**NOTA**

Per evitare possibili salti di quota si consiglia di impostare sempre valori di potenze di due (2^n : es. 2, 4, ..., 2048, 4096, 8192,...) negli oggetti 6001 e 6002.

Se si modificano le informazioni per giro e/o la risoluzione totale bisogna reimpostare eventuali azzeramenti o preset.

6003-00 Valore di preset [Unsigned32, rw]

Usare questo oggetto per assegnare e impostare un valore di Preset.

**NOTA**

- Se "Funzione di scaling" è disabilitata (vedi oggetto 6000), il "Valore di preset" deve essere inferiore a "Risoluzione totale fisica".
- Se "Funzione di scaling" è abilitata (vedi oggetto 6000), il "Valore di preset" deve essere inferiore a "Risoluzione totale" (oggetto 6002).

6004-00 Valore di posizione [Unsigned32, ro]

Questo oggetto contiene il valore di posizione (eventualmente modificato dalla "Funzione di scaling").

6500-00 Stato operativo [Unsigned16, ro]

Bit	Funzione	bit = 0	bit = 1
0	Direzione conteggio	Orario	Antiorario
1	Non utilizzato		
2	Funzione di scaling	Disabilitato	Abilitato
3...15	Non utilizzato		

- Con la definizione di "Direzione conteggio" si stabilisce in quale senso ruotare l'albero encoder, in modo che il valore in uscita dall'encoder incrementi quando l'albero ruota in senso orario oppure antiorario. Il senso di rotazione è stabilito guardando l'encoder dall'estremità dell'albero.
- Funzione di scaling: se disabilitato l'encoder utilizza la propria risoluzione fisica (vedi oggetto 6501h e 6502h); se abilitato utilizza la risoluzione impostata negli oggetti 6001h e 6002h.

6501-00 Informazioni per giro fisiche [Unsigned32, ro]

Questo oggetto definisce il numero fisico di informazioni per giro del dispositivo.

Per impostare una risoluzione diversa si veda all'oggetto 6001h.

6502-00 Numero di giri fisico [Unsigned32, ro]

Questo oggetto definisce il numero fisico di giri del dispositivo.

La risoluzione totale fisica del dispositivo risulta:

"Risoluzione totale fisica" = "obj_6501" x "obj_6502".

Per impostare una risoluzione totale diversa si veda agli oggetti 6001h e 6002h.

6503-00 Errori [Unsigned16, ro]

In questo parametro sono settati i bit relativi agli errori supportati.

6504-00 Errori supportati [Unsigned16, ro]

Default = 0000h (nessun errore supportato).

6505-00 Warning [Unsigned16, ro]

In questo parametro sono settati i bit relativi alle avvertenze supportate.

6506-00 Warning supportati [Unsigned16, ro]

bit 12: parametri caricati all'accensione errati.

Default = 1000h

6509-00 Valore di offset [Unsigned32, ro]

Questo oggetto contiene il valore di Offset che risulta dalla differenza tra la posizione fisica dell'encoder e quella relativa al Preset.

**NOTA**

Per salvare i parametri modificati eseguire "Salvataggio parametri" (vedi oggetto 1010h).

Nel caso di spegnimento del dispositivo i dati non salvati andranno persi.

5.2.6 SDO Abort codes

Il trasferimento degli SDO può non andare a buon fine; le cause di errore sono specificate negli "SDO Abort Codes" (vedi ETG1000.6 par. 5.6.2.7.2 table.40).

5.2.7 Emergency Error Codes

L'Emergency Service viene usato dal server per trasmettere messaggi di diagnostica al client attraverso la MailBox; i relativi Error Codes sono riportati in ETG1000.6 par. 5.6.4.2 table 50.

Error Code		Error Register	Diagnostic Data				
Byte (0)	Byte (1)	Byte (2)	Byte (3)	Byte (4)	Byte (5)	Byte (6)	Byte (7)

Error Code	<p>errori di transizione della macchina a stati: (per descrizione dettagliata vedi ETG1000.6 par. 5.6.4.3) A000hex: errore di transizione da PRE-OP a SAFE-OP A001hex: errore di transizione da SAFE-OP a OP</p> <p>errori encoder: 5000hex: errore hardware 5001hex: diagnostic data (errore lettura parametri encoder da memoria Flash)</p>
Error Register	stato corrente della macchina a stati EtherCAT (ESM)
Diagnostic Data	fornisce indicazioni sulle cause dell'errore (vedi ETG1000.6 par. 5.6.4.3.2-5).

5.2.8 AL Status Error Codes

Se il cambiamento di stato richiesto dal master attraverso "AL Control Register" non è andato a buon fine, lo slave imposta a 1 il bit "Error Ind" del registro "AL Status" e scrive nel registro "AL Status Code" la causa dell'errore.

I valori e le descrizioni di "AL Status Code" sono riportati in ETG1000.6 par. 5.3.2 Table 11.

5.3 File Over EtherCAT (FoE)

Gli encoder Lika sono dispositivi che permettono l'aggiornamento del firmware utilizzando il protocollo "File Over EtherCAT (FoE)".

Per le specifiche relative al protocollo FoE fare riferimento ai documenti "ETG.1000 EtherCAT Specification" disponibili sul sito www.ethercat.org.

Versione documento	Descrizione
1.0	Prima release
1.1	Revisione generale
1.2	Aggiunto capitolo 5.3



LIKA Electronic

Via S. Lorenzo, 25 - 36010 Carrè (VI) - Italy

Tel. +39 0445 806600

Fax +39 0445 806699

Italy: eMail info@lika.it - www.lika.it

World: eMail info@lika.biz - www.lika.biz

User's manual

Hx58x EC

Table of Contents

- 1 Safety summary
- 2 Identification
- 3 Quick reference (TwinCAT)
- 4 Electrical connections
- 5 EtherCAT® interface
- 6 Setup



EtherCAT® 

1 - Safety summary

Safety

- Always adhere to the professional safety and accident prevention regulations applicable to your country during device installation and operation;
- installation and maintenance operations have to be carried out by qualified personnel only, with power supply disconnected and stationary mechanical parts;
- device must be used only for the purpose appropriate to its design: use for purposes other than those for which it has been designed could result in serious personal and/or the environment damage;
- high current, voltage and moving mechanical parts can cause serious or fatal injury;
- warning ! Do not use in explosive or flammable areas;
- failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the equipment;
- Lika Electronic s.r.l. assumes no liability for the customer's failure to comply with these requirements.

Electrical safety

- Turn off power supply before connecting the device;
- connect according to explanation in section 4: "Electrical connections";
- in compliance with the 2004/108/EC norm on electromagnetic compatibility, following precautions must be taken:
 - before handling and installing, discharge electrical charge from your body and tools which may come in touch with the device;
 - power supply must be stabilized without noise, install EMC filters on device power supply if needed;
 - always use shielded cables (twisted pair cables whenever possible);
 - avoid cables runs longer than necessary;
 - avoid running the signal cable near high voltage power cables;
 - mount the device as far as possible from any capacitive or inductive noise source, shield the device from noise source if needed;
 - to guarantee a correct working of the device, avoid using strong magnets on or near by the unit;
 - minimize noise by connecting the shield and/or the connector housing and/or the frame to ground. Make sure that ground is not affected by noise. The connection point to ground can be situated both on the device side and on user's side. The best solution to minimize the interference must be carried out by the user.



Mechanical safety

- Install the device following strictly the information in the section "Mounting instructions";
- mechanical installation has to be carried out with stationary mechanical parts;
- do not disassemble the encoder;
- do not tool the encoder or its shaft;
- delicate electronic equipment: handle with care; do not subject the device and the shaft to knocks or shocks;
- respect the environmental characteristics declared by manufacturer;
- solid shaft: use a flexible coupling to connect the encoder to motor shaft. Make sure the misalignment tolerances of the flexible coupling are respected.

2 - Identification

Device can be identified through the label's data (ordering code, serial number). This information is listed in the delivery document. For any information on the technical features of the product, refer to the technical catalogue.

3 - Quick reference (TwinCAT)

3.1 System configuration using TwinCAT

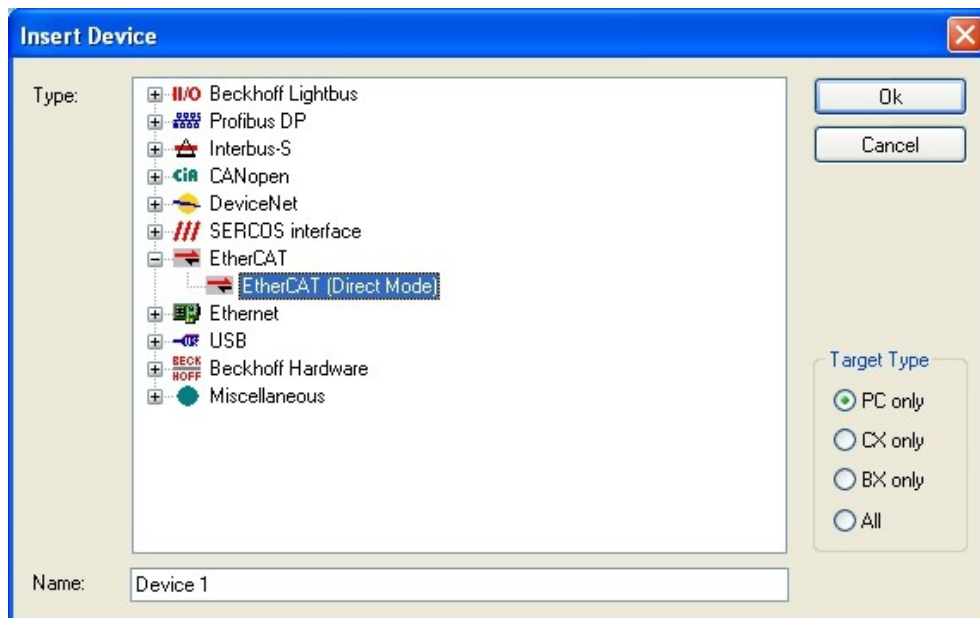
Setting the Network Card

Start TwinCAT System Manager.

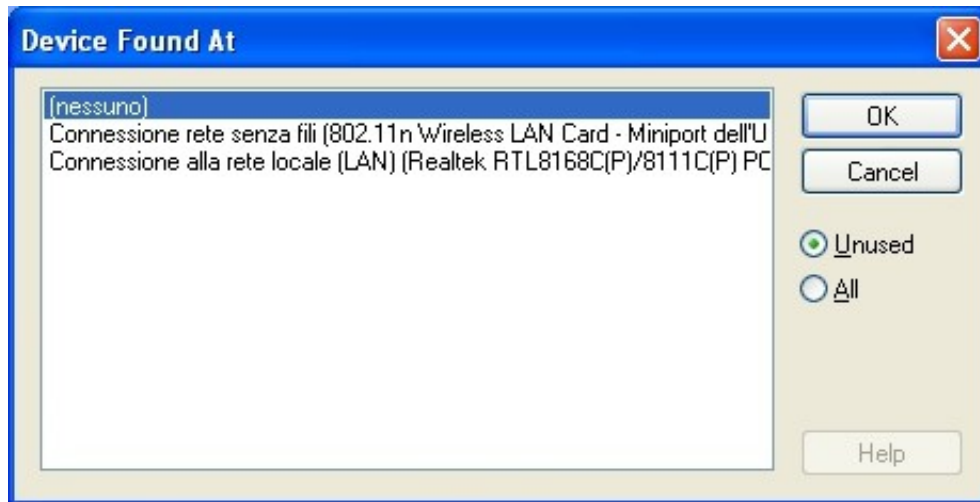
In the left pane of the main window extend the devices tree and select the **I/O Devices** item; right-click the **I/O Devices** item and then press the **Append Device...** command.



In the **Insert Device** window select the **EtherCAT (Direct Mode)** item and confirm pressing the **OK** button.

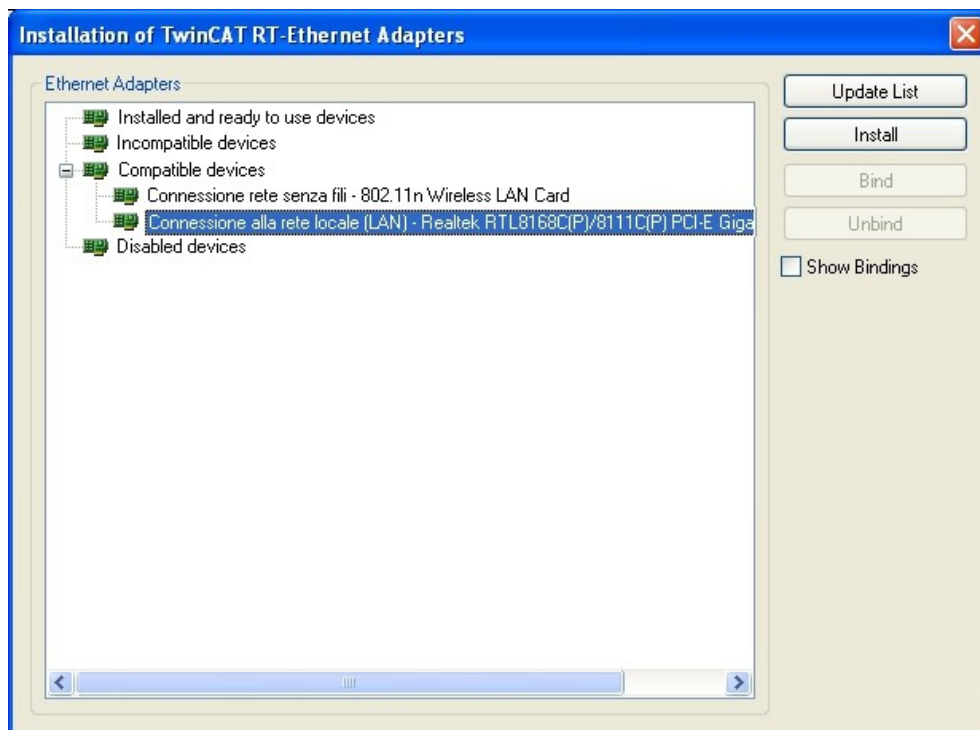


Should a network card be installed properly, the following window will appear showing the list of the installed devices.



Select the network card you want to use and then confirm the choice pressing the **OK** button.

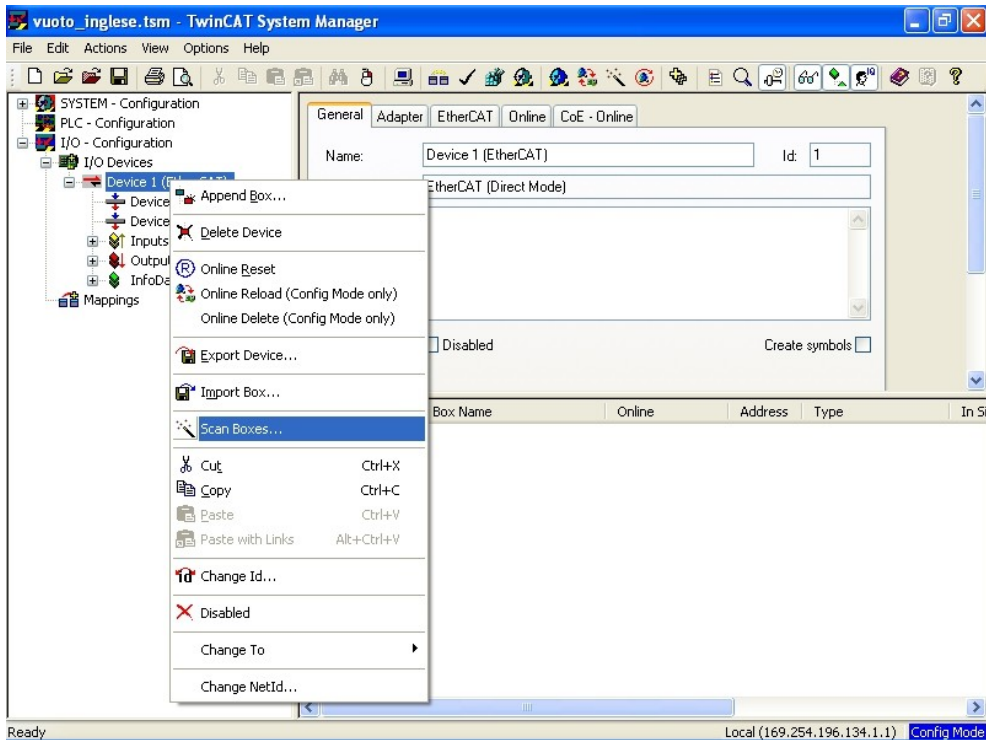
If there are no network cards installed, you must install one before proceeding. To do this, on the menu bar of the **TwinCAT System Manager** main window, select the **Options** menu and then press the **Show Real Time Ethernet Compatible Devices...** command.



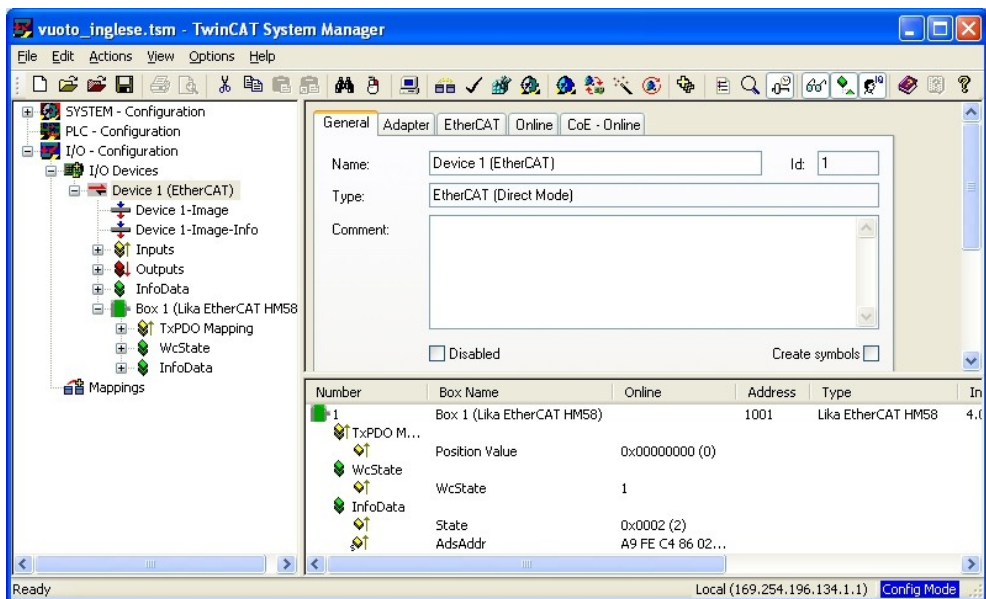
The **Installation of TwinCAT RT – Ethernet Adapter** window will appear. Now select **Compatible Devices** item and choose the network card you want to install; finally press the **Install** button to confirm your choice.

Add new I/O modules (Boxes)

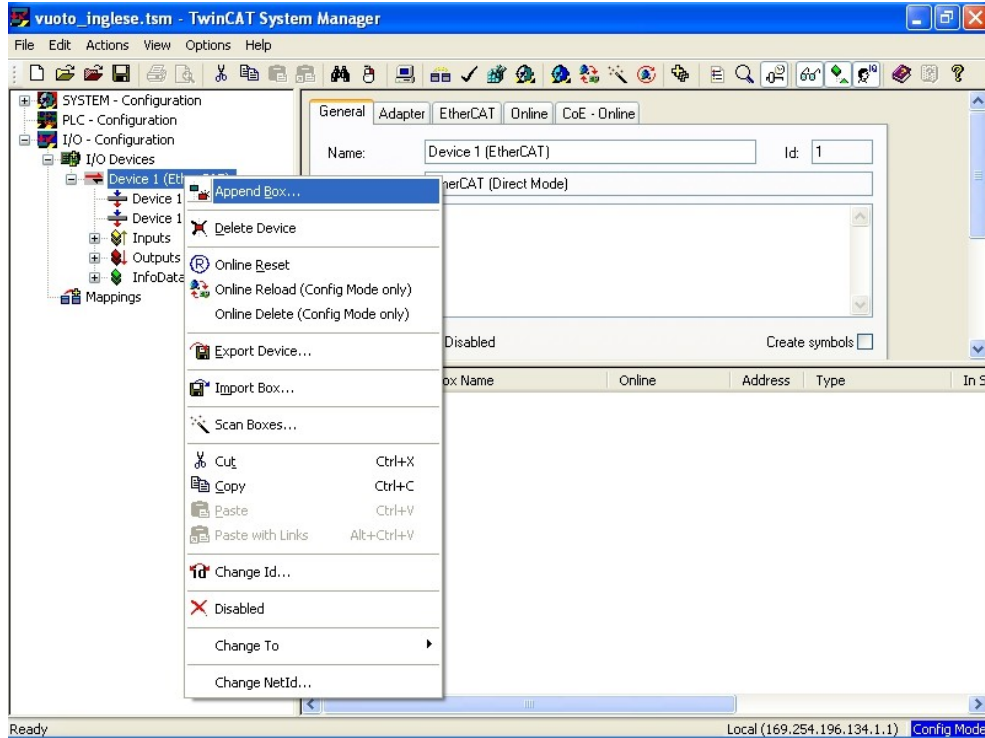
If devices are connected to the network and switched ON, right-click the **Device 1 (EtherCAT)** item in the left pane of the **TwinCAT System Manager** main window and press the **Scan Boxes...** command.



At the end of the process some information will be listed in the right page as in Figure here below.



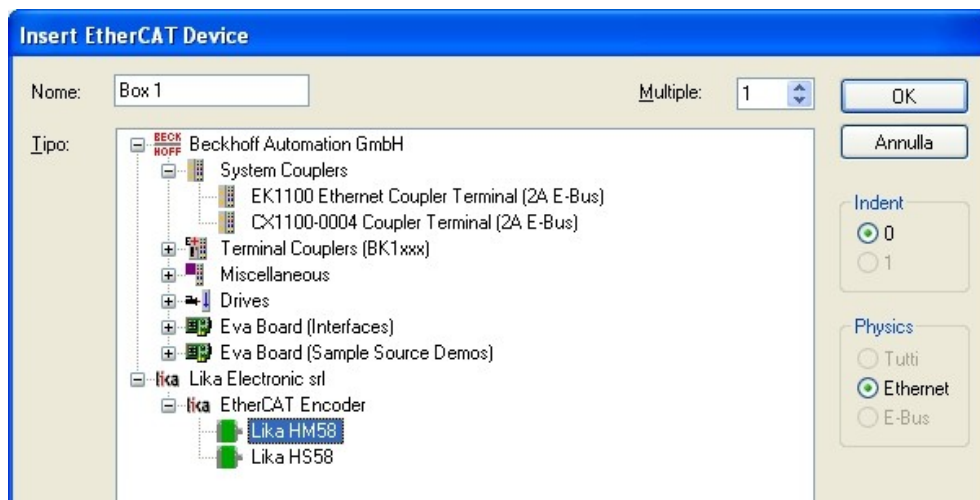
If devices are not already connected to the network it is necessary to use the XML file supplied with the encoder: **Lika_Hx58_EC_Vx.xml** (see enclosed documentation or click www.lika.biz > **PRODUCTS** > **ROTACOD** > **Hx58x EC**). Right-click the **Device 1 (EtherCAT)** item in the left pane of the **TwinCAT System Manager** main window and press the **Append Box...** command.



In the **Insert EtherCAT Device** window that appears select **Lika Electronic srl** and then **EtherCAT Encoder** items; now choose from the list the encoder you want to install:

- HM58: multi-turn encoder;
- HS58: single-turn encoder.

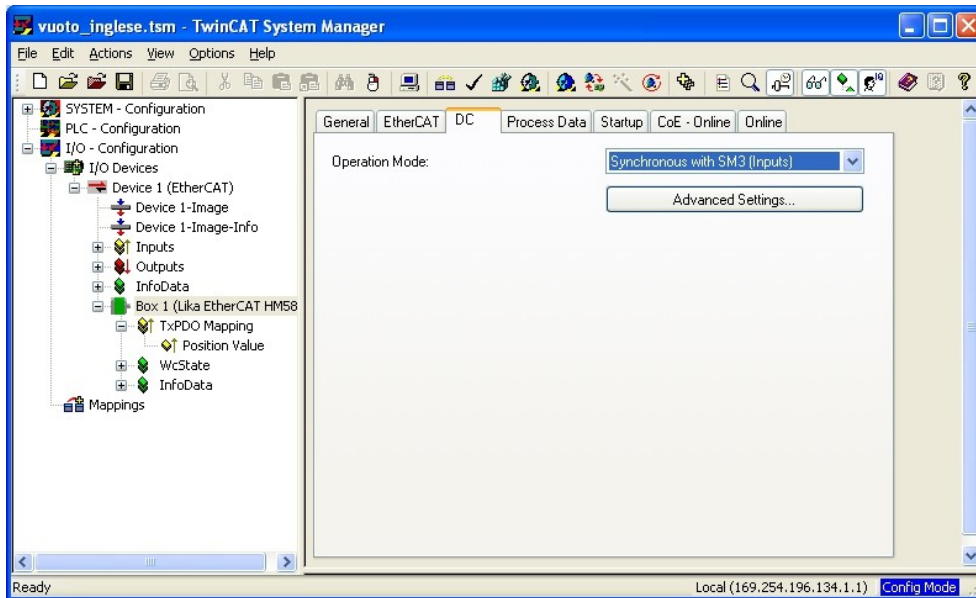
Press the **OK** button to confirm your choice.



3.2 Setting the communication mode

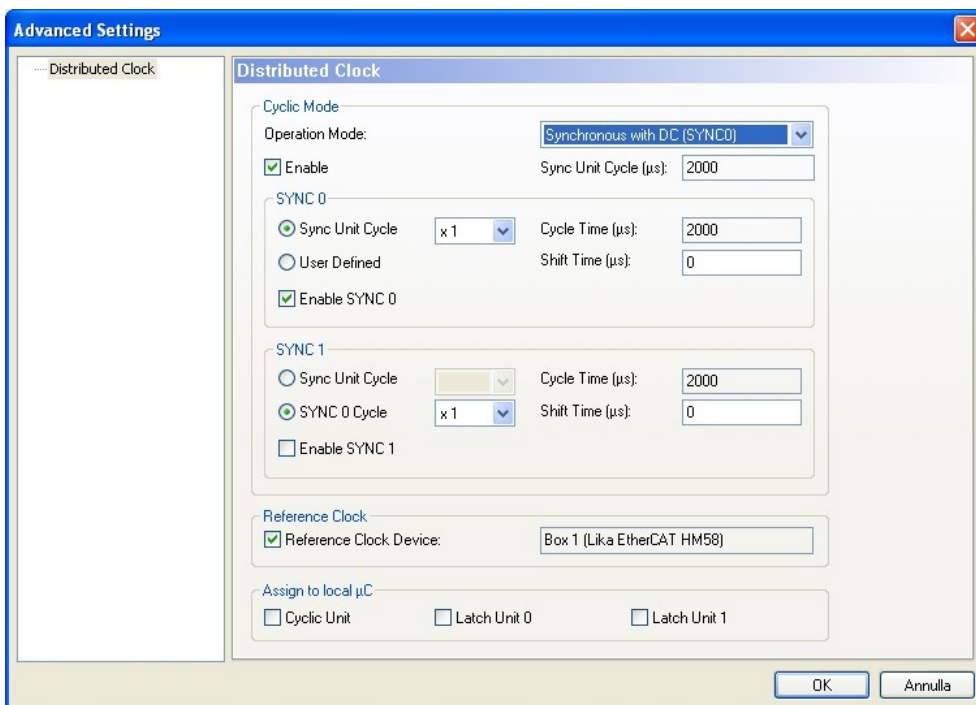
Synchronous with SM3

In the left pane of the **TwinCAT System Manager** main window press the **Box 1 (Lika EtherCAT HM58 or HS58)** item: some tabbed pages for configuring and managing the device will appear in the right pane. Enter **DC** page. Select the **Synchronous with SM3 (Inputs)** option in the **Operation Mode** box.



Synchronous with DC (SYNC0)

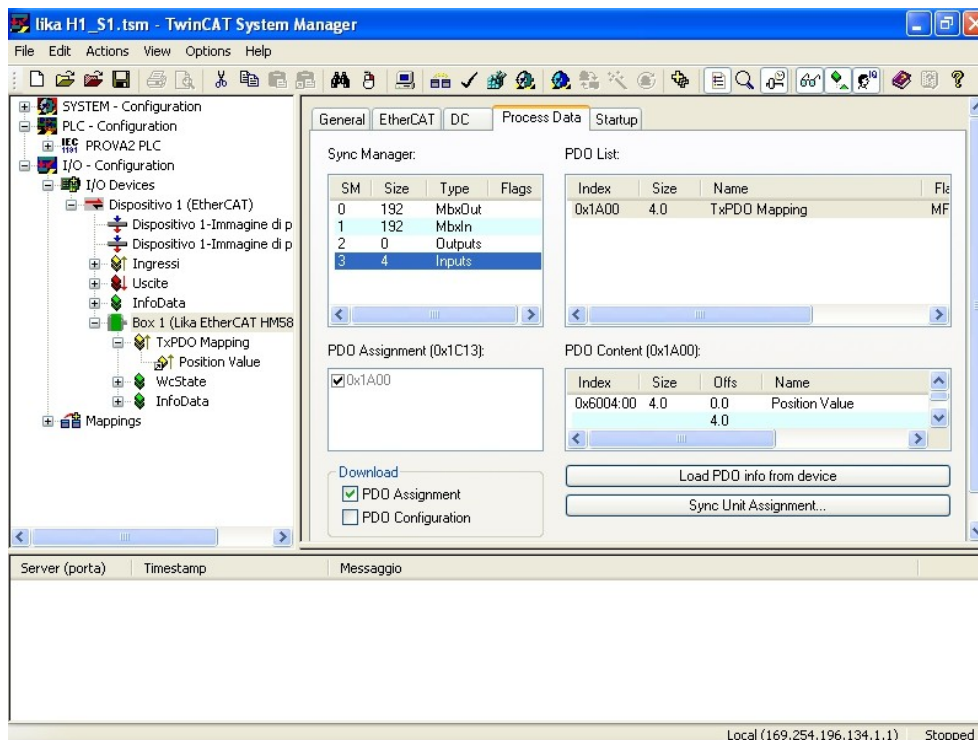
In the left pane of the **TwinCAT System Manager** main window press the **Box 1 (Lika EtherCAT HM58 or HS58)** item: some tabbed pages for configuring and managing the device will appear in the right pane. Enter **DC** page. Select the **Synchronous with DC (SYNC0)** option in the **Operation Mode** box. Then press the **Advanced Settings...** button.



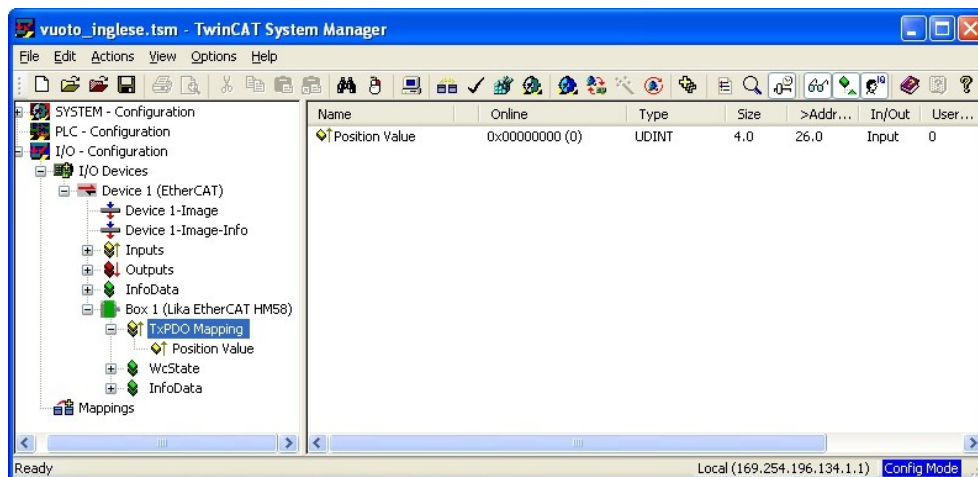
The **Advanced Settings** window will appear. In the section tabbed **SYNC 0** set the cycle time next to the **Sync Unit Cycle** box; sync time is calculated as multiple (or sub-multiple) of the value set in the **Sync Unit Cycle (µs)** item right above.

3.3 Process Data Objects

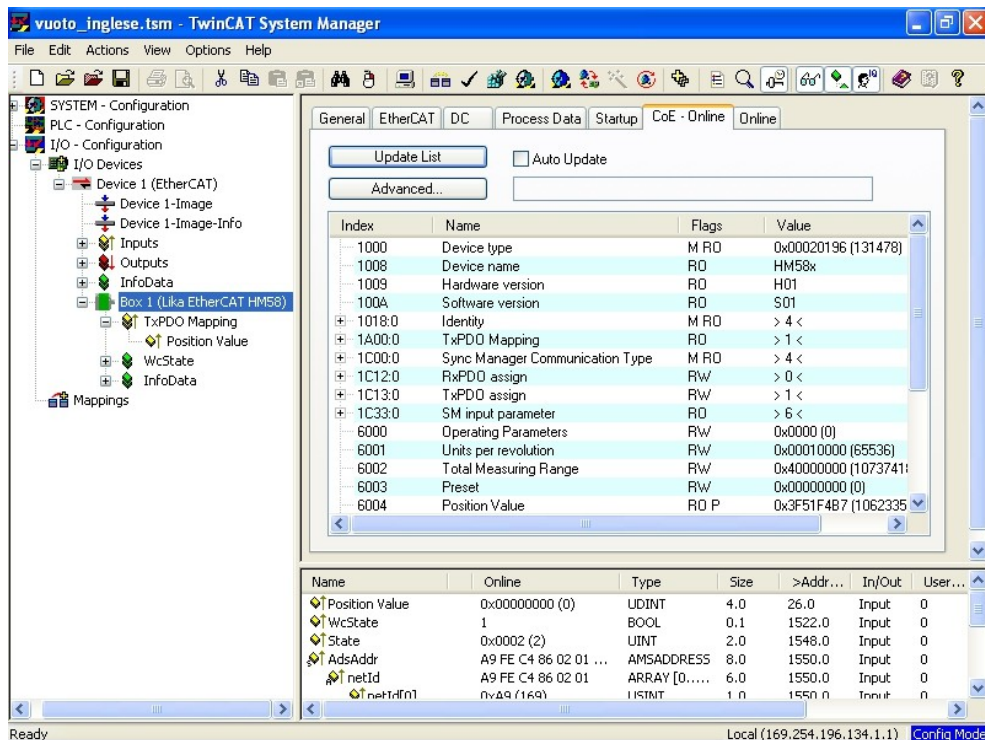
In the left pane of the **TwinCAT System Manager** main window press the **Box 1 (Lika EtherCAT HM58 or HS58)** item: some tabbed pages for configuring and managing the device will appear in the right pane. Enter **Process Data** page. In this page process data objects (PDO Mapping) are shown.



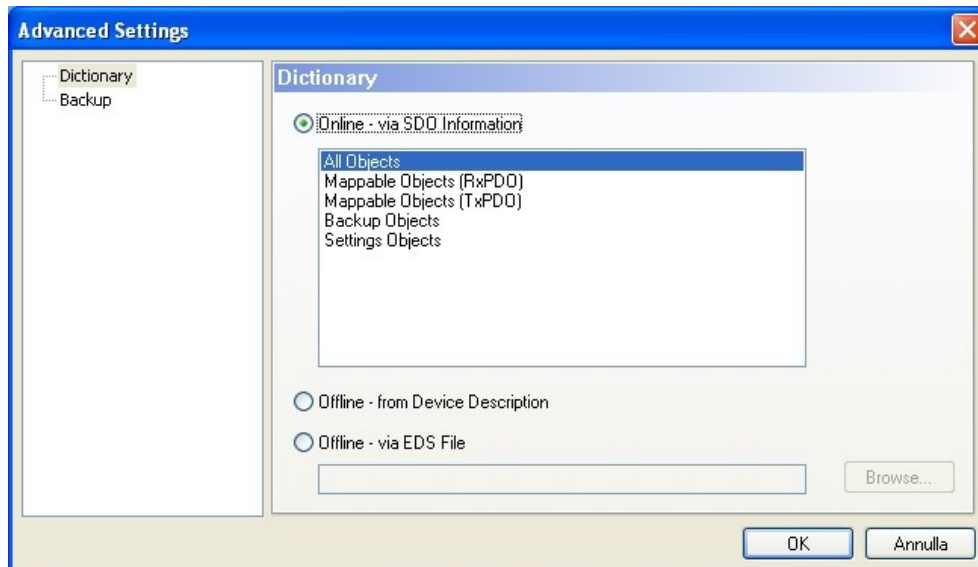
Process data objects can be displayed also by pressing the **TxPDO Mapping** item in the left pane of the **TwinCAT System Manager** main window; data are listed in the right pane.



In the left pane of the **TwinCAT System Manager** main window press the **Box 1 (Lika EtherCAT HM58 or HS58)** item: some tabbed pages for configuring and managing the device will appear in the right pane. Enter **CoE - Online** page. In this page the objects dictionary is shown.



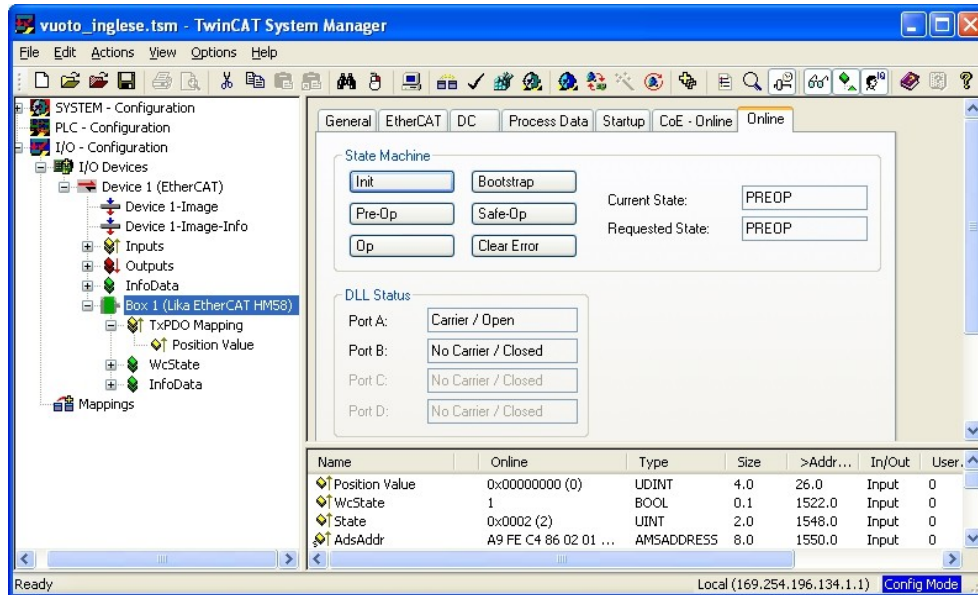
Objects can be read directly from the encoder; to do this click the **Advanced...** button: the **Advanced Settings** window will appear.



Select the **Dictionary** item in the left pane and then choose the **Online - via SDO Information** option in the **Dictionary** page; press **OK** button to confirm.

3.5 Online Data

In the left pane of the **TwinCAT System Manager** main window press the **Box 1 (Lika EtherCAT HM58 or HS58)** item: some tabbed pages for configuring and managing the device will appear in the right pane. Enter **Online** page to check the encoder status.



To display the encoder process data in real time, click the **Safe-OP** button if you want to display inputs only; click the **OP** button if you want to display both inputs and outputs.



ATTENTION

Structure of Data Objects (PDO and SDO) requires bytes to be sent from the Least Significant Byte (LSB) to the Most Significant Byte (MSB).

On the contrary in TwinCAT write and read data from MSB to LSB.

Furthermore in TwinCAT also strings must be entered in the reverse order:

- read default values: Data byte = 64 61 6F 6Chex = "daol" in ASCII code (means "load" in reverse);
- save parameters: Data byte = 65 76 61 73hex = "evas" in ASCII code (means "save" in reverse).

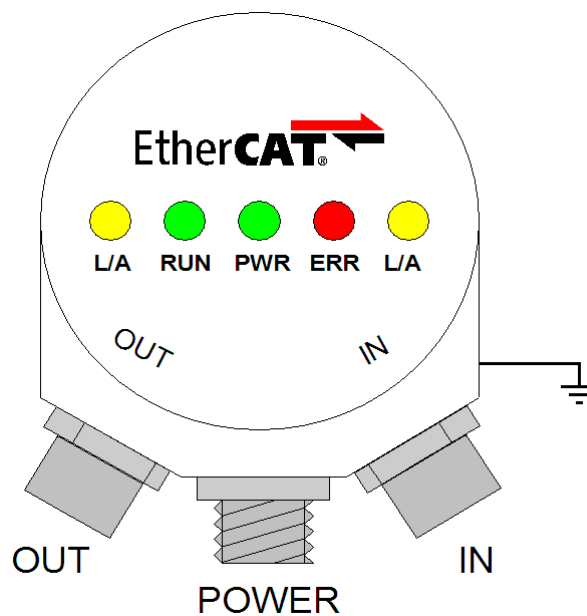
4 - Electrical connections



ATTENTION

Do not remove the connection cap. Damage may be caused to internal components.

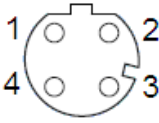

Minimize noise by connecting the shield and/or the connector housing and/or the frame to ground. Make sure that ground is not affected by noise. The connection point to ground can be situated both on the device side and on user's side. The best solution to minimize the interference must be carried out by the user. You are advised to provide the ground connection as close as possible to the encoder.



4.1 M12 connectors

The connection cap is fitted with three M12 connectors with pin-out in compliance with EtherCAT® standard. Therefore you can use standard EtherCAT cables commercially available.

Input (IN) and output (OUT) connectors are not interchangeable! IN connector must be placed in the direction of the EtherCAT Master.

M12 (frontal side)	BUS IN e BUS OUT	POWER
	 <p>D coding female</p>	 <p>A coding male</p>

Pin	Description	Description
1	Tx Data +	+10VDC +30VDC
2	Rx Data +	n.c.
3	Tx Data -	0 VDC
4	Rx Data -	n.c.

n.c.= not connected.

4.2 Diagnostic LEDs

Five LEDs located in the rear side of the connection cap are designed to show the operating or fault status of the EtherCAT® interface.

Link/ Activity (yellow)	it shows the status of the physical link and the activity in this link
OFF	condition: port closed, link: SI, activity: N.A.
FLICKERING	condition: port open, link: SI, activity: SI
ON	condition: port open, link: SI, activity: NO

RUN (green)	it shows the status of EtherCAT State Machine (ESM)
OFF	encoder is in INIT status
BLINKING	encoder is in PRE-OPERATIONAL status
SINGLE FLASH	encoder is in SAFE-OPERATIONAL status
ON	encoder is in OPERATIONAL status

PWR (green)	it shows the power supply status
OFF	encoder is switched OFF
ON	encoder is switched ON

ERR (red)	it shows the error status
OFF	no error
FLICKERING	error on loading parameters from flash memory at start-up
FLICKERING	encoder parameters have not been saved correctly on flash memory
BLINKING	invalid configuration
SINGLE FLASH	local error (see ETG1000.6)
DOUBLE FLASH	watchdog timeout
ON	memory error and ESC controller is not active

5 - EtherCAT® interface

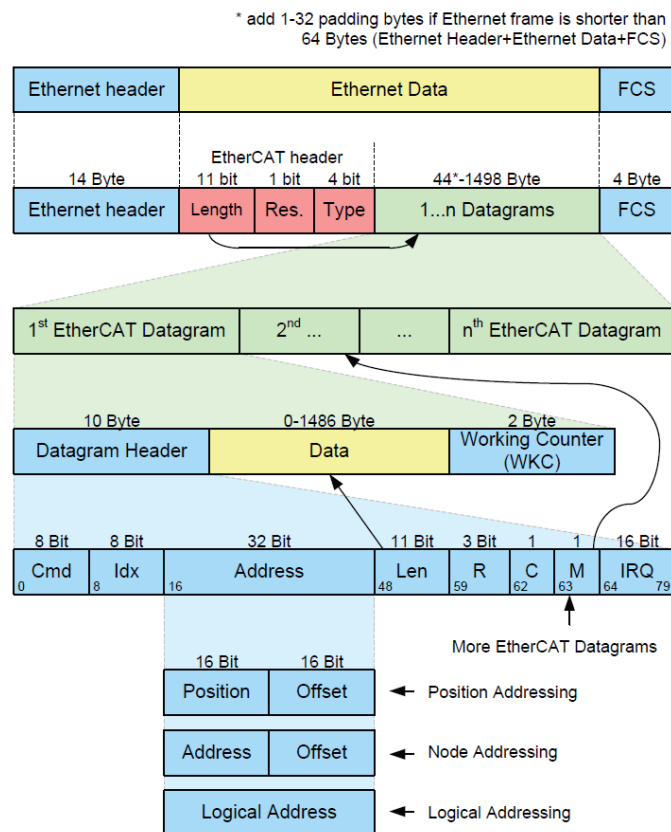
5.1 Basic Information on EtherCAT® Protocol

EtherCAT protocol is designed to use the standard Ethernet dataframes for issuing data; in addition, and as regards the hardware, it is not necessary to install dedicated masters for establishing and managing the EtherCAT communication because standard Ethernet network cards can be used. This results in an advantage in terms of lower costs and simplicity of use because Ethernet network cards are used in standard personal computers and are easily commercially available.

An EtherCAT bus can be viewed as a single and large Ethernet device that receives and sends Ethernet telegrams; it can be considered an Ethernet subnet supported by an Ethernet dataframes structure.

However this "subnet" must be fitted with one only EtherCAT master controller and several EtherCAT slaves, but no Ethernet controller with downstream microprocessor must be present.

Here follows an Ethernet frame structure with EtherCAT:



Inside the Ethernet frames data are transmitted among master and slaves using PDO (process data objects) protocol. Each PDO message has inside one or more addresses for issuing data to slaves; data + address/es (and additional elements such as a validation checksum) joined together forms an EtherCAT telegram (Datagram).

An EtherCAT frame can contain several telegrams and often a complete control cycle requires more than one frame.

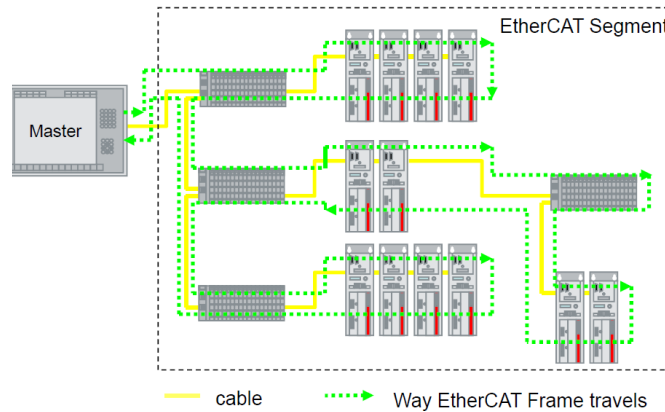
5.1.1 Data transfer

Usually, in a data bus system, master controller sends online a data request and then waits for data to be processed and sent back from each slave node; this does not comply with a real-time system because master receives data from slaves in different moments and the whole system cannot be synchronized.

In EtherCAT the real-time characteristic of the system is quite improved because data are processed "on-the-fly", using one single frame to acquire all data from all slaves.

In fact the frame sent by the master is read by each slave node data is addressed to while the telegram passes through the device; similarly, input data is inserted while the telegram passes through. Then the telegram is forwarded to the next device. Telegrams are only delayed by a few nanoseconds.

The last slave issues back the complete frame to the master with all the requested data (again passing through all the slaves).



This efficient data flow is guaranteed by the 100BASE-TX full-duplex structure of EtherCAT bus which is fitted with two separated lines for transmitting and receiving data.

Moreover the protocols exchange takes place inside the hardware and it is thus independent from CPU and software processing.

5.1.2 ISO/OSI Layer model

OSI Layers	
7	Application Layer (AL) IEC 61158-5/-6 Type 12
6	empty*
5	
4	
3	
2	Data Link Layer (DLL) IEC 61158-3/-4 Type 12
1	Physical Layer (PhL) IEC 61158-2 Type 12

* "empty" means that the layer behaviour exists, but is not shown explicitly.

5.1.3 Topology

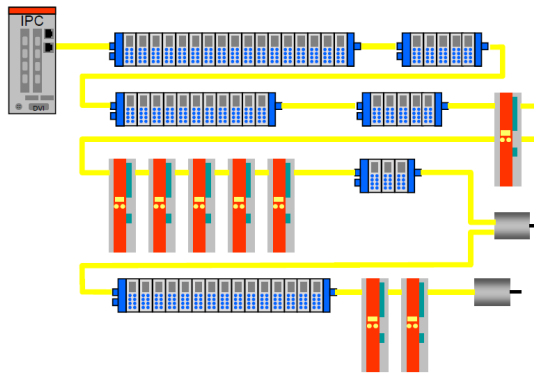
Several topologies of connection are supported by EtherCAT networks: line, tree, daisy-chain, star, ...). Furthermore EtherCAT networks can be configured in almost any topology in the same structure. The maximum length of the cable between two slaves is 100 m; standard EtherCAT cables commercially available can be used.

The choice of the topology depends on the structural characteristics of the plant and it is made in order to reduce the complexity and time for cabling.

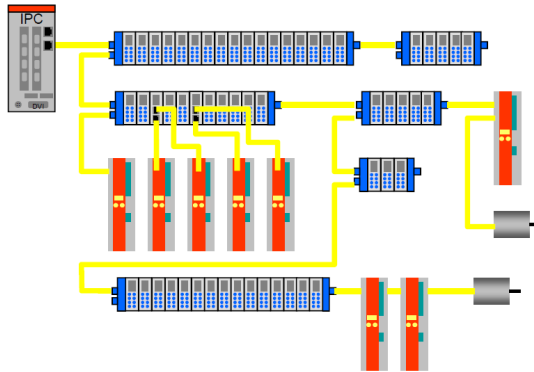
Inside an EtherCAT network up to 65535 devices can be connected.

Some topology examples are shown in the Figures below:

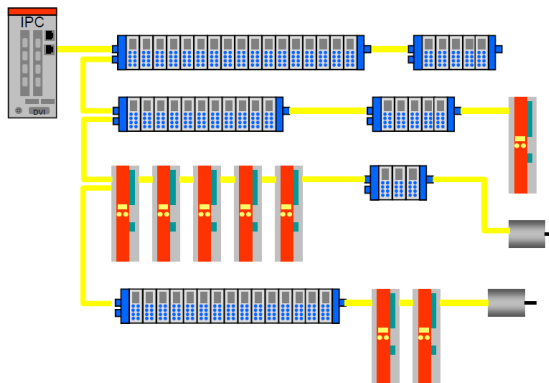
LINE topology:



TREE topology:



DAISY CHAIN with drop lines topology:



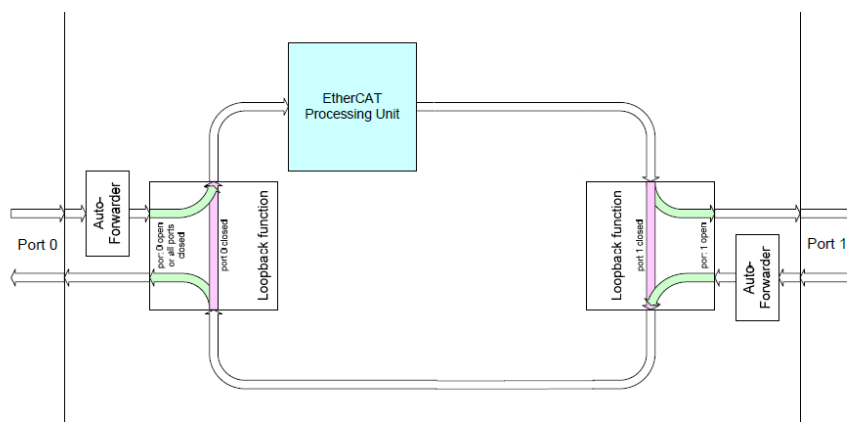
5.1.4 Line Termination

EtherCAT network needs no line termination because the line is terminated automatically; in fact every slave is able to detect the presence of downstream slaves.

An EtherCAT slave is able to detect the presence of the signal in the outgoing line (Port 0) or in the return line (Port 1).

If the slave is not able to detect the signal in its return line then it closes the communication ring by short-circuiting the TX signal of its outgoing line with the RX signal of its return line; in this way a telegram received through the outgoing line is processed and sent back through the TX of the return line.

The slave sends a "carrier signal" or a telegram on TX of the outgoing line continuously and, once the next slave is connected again, a signal on RX of the return line is detected again; so the short circuit is removed and the telegrams are sent on TX of the outgoing line.



5.1.5 Addressing

It is not necessary to assign a physical address to the device (for instance using a dip-switch) because the addressing of the slave is automatic at power on during the initial scanning of hardware configuration.

8 Bit	8 Bit	32 Bit		11 Bit	2	1	1	1	16 Bit
Cmd	Idx	Address		Len	R	C	R	M	IRQ
APxx		16 Bit	16 Bit						
		Position	Offset						← Auto Increment Addressing (Position addressing)
FPxx		Address	Offset						← Fixed Physical Addressing (Node addressing)
Lxx		Logical Address							← Logical Addressing

The field for addressing is 32 bits long; there are three kinds of addressing:

- Auto Increment Addressing = Position Addressing: 16 bits indicates the physical position of the slave inside the network while 16 bits are scheduled for local memory addressing; when the slave receives the frame then it increments the position address and the slave receiving address 0 is the addressed device;
- Fixed Addressing = 16 bits indicates the physical address of the slave inside the network while 16 bits are scheduled for addressing the local memory;

- Logical Address = the slave is not provided with its own individual address, but it can read and write data in a section of the total memory space available (4 Gigabytes).

5.1.6 Communication mode

Lika encoders with EtherCAT interface support the following operating modes:

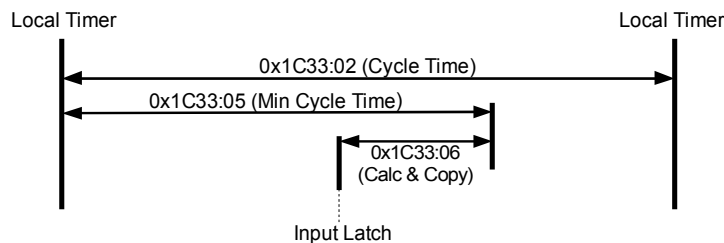
- FreeRun: asynchronous mode;
- SM3 event: synchronous mode;
- DC: distributed clock synchronization mode (synchronous mode).

For a system that requires high performances in real time (closed-loop applications) we suggest using DC mode; if real time requirements are not so mandatory SM3 or Freerun modes can be used instead.

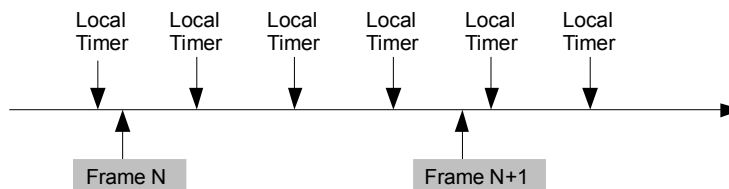
A reference parameter is the "Jitter": it represents the temporal fluctuation of the instant data sampling; in other words data sampled by the micro-controller is available in ECAT DPRAM memory after a certain time and the measure of the variability over time is the "jitter".

FreeRun

Asynchronous mode; encoder position is sampled directly from EtherCAT frame sent by the master; position update is performed by an internal timer of the controller every 100 microseconds.



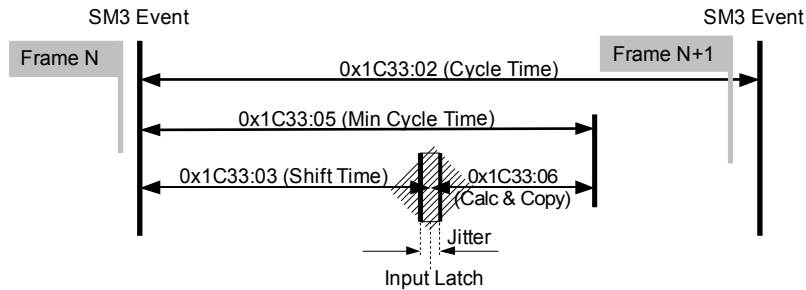
This operating mode has a high sampling jitter (up to 100 microseconds) and can be chosen only when cycle times are quite longer than the jitter if we want to ensure a sufficient real-time system performance.



Description	Min	Typ	Max	
Jitter	0		100	μsec
Cycle Time	1000		64000	μsec

Synchronous with SM3

In this mode data is sampled and then copied into Sync Manager buffer as soon as previous data was read from the master (SM event); in this way new sampled data is synchronous with master readings.



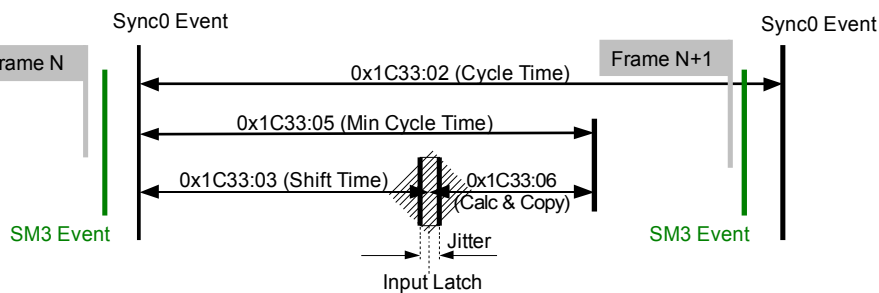
New data will be read by the master at next cycle (following SM3 event), so if cycle time is too long, data could be relatively old for a real-time system. The main advantage is that data is updated exactly when master is reading (synchronous mode).

Description	Min	Typ	Max	
Jitter	0		7,2	μsec
Cycle Time	62,5		64000	μsec

Synchronous with DC

In this operating mode data is sampled and then copied into Sync Manager buffer simultaneously at SYNC0 event generated by the ESC capture/compare unit.

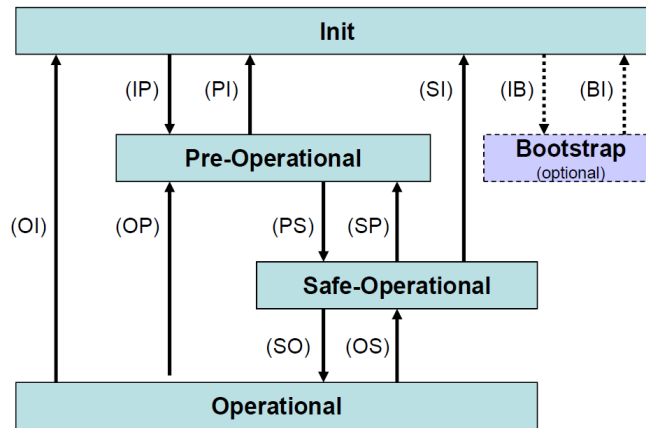
Time required for accomplishing these operations is set in object 1C33hex; in particular object "Shift Time" (1C33hex, sub3) and object "Calc & Copy time" (1C33hex, sub6).



In this operating mode "Jitter" is a fundamental parameter in the sampling of two consecutive data. The main advantage of this mode is that there is a direct relation between the sampling instant and the absolute time of the system; in this way, if we know the shift times of the slaves, we can have an exact image of the system at a given moment (with a tolerance equal to jitter).

Description	Min	Typ	Max	
Jitter	0	100	200	μsec
Cycle Time	62,5		64000	μsec

5.1.7 EtherCAT State Machine (ESM)



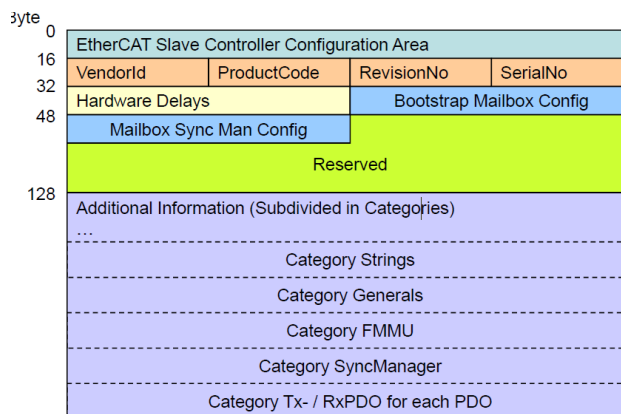
EtherCAT slave is a state machine; communication and operating characteristics depend on the current state of the device:

- INIT: it is the default state after power on; in this state there is not direct communication between master and slave on the Application Layer; some configuration registers are initialized and the Sync Managers are configured;
- PRE-OPERATIONAL: in this state mailbox is active; both master and slave can use the mailbox and its protocols for exchanging specific initialization parameters of the application. Exchange of Process Data (PDO) is forbidden;
- SAFE-OPERATIONAL: in this state master and slave can issue only input process data, while output process data are still in safe-state;
- OPERATIONAL: in this state master and slave are enabled to send both input process data and output process data.

5.1.8 Slave configuration

The configuration of the slave communication characteristics (Sync Manager configuration, addresses, synchronization modes, PDO mapping,...) can be made both using the XML file (EtherCat Slave Information - ESI) or loading data directly from EEPROM (Slave Information Interface SII).

EEPROM content (SII):



5.1.9 Timing and Synchronization

The main feature of EtherCAT is its almost ideal representation of a real-time system.

Hence the master has to synchronize all the slaves at the same time in order to build a system where all nodes have the same reference time; this goal can be achieved by using "distributed clocks".

Master downloads its clock into one of the slaves (customarily the first slave) which becomes the reference clock for all the slaves in the network; so it has the task of synchronizing the other slaves. Master controller periodically sends a special synchronization-telegram where the reference slave writes its own "current time". This telegram is then sent to all the other slaves that, in this way, provide for a new re-synchronization of their own clock in order to avoid possible drifts.

This synchronization of the reference time is very important in order to have a snapshot of the system and accordingly to take simultaneous actions in high sensitive applications such as the coordination in axis control operations.

Besides, EtherCAT Slave Controller (ESC) is fitted with a capture/compare unit that provides accurate synchronization signals (SYNCO or interrupts): they are sent to the local micro-controller so that it is able to synchronize its own clock to slaves clock.

Sync Manager

Sync Manager has the task of synchronizing data transfer between master and slave and prevents the same memory area from being written by different events.

There are two synchronization modes:

- 3-Buffer Mode;
- 1-Buffer Mode.

Synchronisation mode is initialized through the XML file or by loading data directly from EEPROM (SII).

Buffered Mode (3-Buffer Mode)

In this mode new data can be accessed at any time by both master EtherCAT and ESC controllers; no timing restrictions are imposed.

Three buffers are necessary (three consecutive memory areas); one buffer is always available to ESC controller for writing and one buffer always contains updated data to be read by the master.

Customarily this is used for process data communication.

Mailbox Mode (1-Buffer Mode)

In this mode a "handshake" between master and slave must be used; in fact one only memory buffer is available to both master and slave for writing and reading; the master (or the slave) is enabled to write only when the buffer is empty, that is when the slave (or the master) has finished reading the data buffer. And vice versa: the master (or the slave) is enabled to read only when the buffer is empty, that is when the slave (or the master) has finished writing the data buffer.

5.2 CANopen Over EtherCAT (CoE)

Lika encoders are slave devices and support "CanOpen Over EtherCAT" (COE) mode for data transfer. In particular, they support the "Device profile for encoders", Class 2.

For any omitted specification on CANopen® protocol, please refer to "CiA Draft Standard 301" and "CiA Draft Standard 406" documents available at the address www.can-cia.org.

For any omitted specification on EtherCAT® protocol, please refer to "ETG.1000 EtherCAT Specification" document available at the address www.ethercat.org.

5.2.1 File XML

EtherCAT® encoders are supplied with their own XML file **Lika_Hx58_EC_Vx.xml** (see enclosed documentation or click www.lika.biz > **PRODUCTS > ROTACOD > Hx58x EC**).

XML file has to be installed on EtherCAT® master device.

5.2.2 Communication messages

EtherCAT Datagram of CoE mode has the following structure:

Mbx Header	CoE Cmd			Cmd specific data
type = 3	Number	res	Type	
6 byte	9 bit	3 bit	3 bit	0...1478 byte

Mbx Header = 3 CoE mode

Number = 0 in case of SDO messages
 ≠ 0 in case of PDO messages, it defines the type of service

res reserved bits

Type = 0 reserved
 = 1 Emergency message
 = 2 SDO request
 = 3 SDO response
 = 4 Transmitted PDO (TxPDO)
 = 5 Received PDO (RxPDO)
 = 6 Remote transmission request of TxPDO
 = 7 Remote transmission request of RxPDO
 = 8 SDO information
 = 9...15 reserved

Cmd specific data PDO messages: are the process data, e.g. position value
 SDO messages: standard CANopen frame

Transmit (tx) or receive (rx) "Type" is viewed from slave.

5.2.3 Process Data Objects (PDO)

PDO messages are used for transmitting or receiving process data in real time; data to be transmitted or received is defined in PDO Mapping and managed by Sync Manager PDO Mapping.

5.2.4 Service Data Objects (SDO)

SDO messages are issued via Mailbox (low priority data); Segmented SDO Service and SDO Complete Access are not supported (transfer of low size data and one sub-index at a time).

"CoE Cmd type" = 2 or 3

Structure of "Cmd specific data":

Cmd specific data				
SDO control	Index	Sub index	Data	Data optional
8 bit	16 bit	8 bit	32 bit	1...1470 byte

SDO control standard CANopen SDO Service

Index parameter index

Sub index parameter sub-index

Data parameter value

Data optional optionally, more then 4 bytes of data can be sent in one frame.
Full mailbox size usable.

Index and sub-index values are described in the "Object dictionary".

5.2.5 Object dictionary

In the following pages the objects implemented are listed and described as follows:

Index-subindex Object name [data types, attribute]

- Index and sub-index are expressed in hexadecimal notation.
- Attribute:
 - ro = read only access
 - rw = read and write access

Unsigned16 data type:

Process data bytes	
byte 4	byte 5
LSByte	MSByte

Unsigned32 data type:

Process data bytes			
byte 4	byte 5	byte 6	byte 7
LSByte	MSByte

Standard objects

1000-00 Device type [Unsigned32, ro]

Default = 0001 0196h = single turn encoder, DS 406
0002 0196h = multi turn encoder, DS 406

1008-00 Manufacturer Device Name [String, ro]

Default = "HS58xx" = single turn encoder
"HM58xx" = multi turn encoder

1009-00 Hardware version [String, ro]

Contains the hardware version of the device.

100A-00 Software version [String, ro]

Contains the software version of the device.

1010-01 Store parameters [Unsigned32, rw]

Use this object to save all parameters on non-volatile memory.

Write "save" in the data bytes:

Master → Encoder

Cmd specific data							
Cmd	Index		Sub	Data			
23	10	10	01	73	61	76	65

Encoder → Master (confirmation)

Cmd specific data							
Cmd	Index		Sub	Data			
60	10	10	01	00	00	00	00

1011-01 Restore default parameters [Unsigned32, rw]

This object allows you to restore all parameters to default values.

Write "load" in the data bytes:

Master → Encoder

Cmd specific data							
Cmd	Index		Sub	Data			
23	11	10	01	6C	6F	61	64

Encoder → Master (confirmation)

Cmd specific data							
Cmd	Index		Sub	Data			
60	11	10	01	00	00	00	00



NOTE

To save default values execute the "Store parameters" function (see object 1010h). When the power is turned off, parameters not saved are lost.

1018 Identification object

- **01 Vendor number** [Unsigned32, ro]
- **02 Product number** [Unsigned32, ro]
- **03 Revision number** [Unsigned32, ro]
- **04 Serial number** [Unsigned32, ro]

1A00-01 PDO mapping parameter [Unsigned32, rw]

This object contains the mapped position value of the encoder according to DS406 device profile.

Default = 6004 0020h

1C00 Sync Manager Communication Type

- **01** SM MailBox Receive (SM0) [Unsigned8, ro] Default = 1
- **02** SM MailBox Send (SM1) [Unsigned8, ro] Default = 2
- **03** SM PDO output (SM2) [Unsigned8, ro] Default = 3
- **04** SM PDO input (SM3) [Unsigned8, ro] Default = 4

1C12-00 Sync Manager RxPDO Assigned [Unsigned8, ro]

This device does not support receive PDO messages.

Default = 0

1C13-01 Sync Manager TxPDO Assigned [Unsigned32, ro]

This device uses TxPDO messages to send the position value.

Default = 0000 1A00hex

1C33 Sync Manager input parameter

Some sub-indexes of object 1C33hex are calculated dynamically and depend on the encoder configuration (programmed resolution, counting direction, ...) and synchronization mode (SM or DC).

- **01 Sync Type** [Unsigned16, rw] Default = 1
 - 0: FreeRun
 - 1: Synchronized with SM3 Event
 - 2: DC mode synchronized with SYNC0 event
- **02 Cycle time** [Unsigned32, ro]
 - this parameter depends on the Sync Type selected:
 - if "FreeRun": interval between two position samplings (internal timer);
 - if "Synchronized with SM3": min. interval between two SM3 events;
 - if "Synchronized with DC Sync0": Sync0 cycle time.
- **03 Shift Time** [Unsigned32, ro]
 - Interval between the synchronization event and the moment of inputs latching from hardware. This parameter is calculated dynamically.
- **04 Sync modes supported** [Unsigned16, ro] Default = 7
 - bit 0: FreeRun (supported)
 - bit 1: Synchronized with SM3 (supported)
 - bit 2: Synchronized with DC Sync0 (supported)
- **05 Minimum cycle time** [Unsigned32, ro]
 - Max. duration of the encoder internal cycle time. This parameter is calculated dynamically and depends on the operating parameters and the position value.
- **06 Calc and Copy time** [Unsigned32, ro]
 - Time the internal micro-controller (DSP) needs to make calculations on latched optical reading of position and then copy updated data from local memory to ESC memory (Sync Manager) before they are available to EtherCAT. This parameter is calculated dynamically and depends on the operating parameters and the position value.

Device profile objects

6000-00 Operating parameters [Unsigned16, rw]

Bit	Function	bit = 0	bit = 1
0	Code sequence	cw (clockwise)	ccw (counter clockwise)
1	not used		
2	Scaling function	disabled	enabled
3...15	not used		

Default = 0000h

- "Code sequence" sets the rotation direction of the encoder shaft and consequently defines whether the position value output by the encoder increases when the encoder shaft rotates clockwise (CW) or counterclockwise (CCW). CW and CCW rotations are viewed from shaft.
- "Scaling function": if disabled the device uses its own physical resolution (see objects 6501h and 6502h); if enabled it uses the resolution set in objects 6001h and 6002h with the following relation:

$$\text{posTx} = \frac{\text{obj_6001}}{\text{obj_6501}} \cdot \text{RealPos} \leq \text{obj_6002}$$

6001-00 Counts per revolution [Unsigned32, rw]

This object sets the number of distinguishable steps per revolution. Enabled if bit 2 of object 6000h = "1".

To avoid counting errors, check that $\frac{\text{obj_6501}}{\text{obj_6001}}$ is an integer value.

Allowed values are equal to or lower than "Hardware counts per revolution" (see encoder label).

6002-00 Total resolution [Unsigned32, rw]

This object sets the number of distinguishable steps over the total measuring range.

Enabled if bit 2 of object 6000h = "1".

Allowed values are equal to or lower than "Total hardware resolution" (see encoder label).

Example

Multi turn encoder HM5816/16384EC-6.

Resolution is:

- Hardware counts per revolution: "obj_6501" = 65536 (2^{16})
- Hardware number of turns: "obj_6502" = 16384 giri (2^{14})
- Total hardware resolution: = 1073741824 (2^{30})

2048 counts/rev. * 1024 turns are required:

- Enable scaling function: "obj_6000", bit 2 = "1"
- Counts per revolution: "obj_6001" = 2048 (0000 0800h)
- Total resolution: "obj_6002" = 2048 * 1024 = 2097152 (0020 0000h)



NOTE

To avoid counting errors we recommend values which are power of 2 (2^n : 2, 4, ..., 2048, 4096, 8192,...) to be set in objects 6001h and 6002h.

If "Counts per revolution" and/or "Total resolution" values change, the "Preset value" should be adjusted to the new resolution. A new setting of the "Preset value" is also required.

6003-00 Preset value [Unsigned32, rw]

This object allows to set the encoder position to a Preset value. Preset function is meant to assign a certain value to a desired physical position of the encoder shaft.



NOTE

- If "Scaling function" is disabled (see obj_6000), "Preset value" must be lower than "Total hardware resolution".
- If "Scaling function" is enabled (see obj_6000), "Preset value" must be lower than "Total resolution" (object 6002).

6004-00 Position value [Unsigned32, ro]

This object contains the position value (also considering the scaling function).

6500-00 Operating status [Unsigned16, ro]

Bit	Function	bit = 0	bit = 1
0	Code sequence	Clockwise	CCW
1	not used		
2	Scaling function	Disable	Enable
3...15	not used		

- "Code sequence" shows the rotation direction of the encoder shaft and consequently whether the position value output by the encoder increases when the encoder shaft rotates clockwise (CW) or counterclockwise (CCW). CW and CCW rotations are viewed from shaft.
- "Scaling function": if disabled the device uses its own physical resolution (see objects 6501h and 6502h); if enabled it uses the resolution set in objects 6001h and 6002h.

6501-00 Hardware counts per revolution [Unsigned32, ro]

This object defines the number of distinguishable steps per each turn given by the hardware.

To set a different resolution see object 6001h.

6502-00 Hardware number of turns [Unsigned32, ro]

This object defines the number of distinguishable turns given by the hardware.

"Total hardware resolution" = "obj_6501"*"obj_6502".

To set a different number of turns see objects 6001h and 6002h.

6503-00 Errors [Unsigned16, ro]

The corresponding bits of supported errors are set (see object 6504h).

6504-00 Supported errors [Unsigned16, ro]

Default = 0000h (Errors not supported).

6505-00 Warning [Unsigned16, ro]

Bits of supported warnings are set (see object 6506h).

6506-00 Supported warning [Unsigned16, ro]

bit 12: wrong parameters loaded from flash memory at power on.

Default = 1000h

6509-00 Offset value [Unsigned32, ro]

This object contains the Offset value. This value is the difference between the physical position of the encoder and position relating to the Preset value.



NOTE

To save new parameters execute the "Store parameters" function (see object 1010h).

When the power is turned off, parameters not saved are lost.

5.2.6 SDO Abort codes

SDO transfer could be unsuccessful; causes of error are listed and described in SDO Abort Codes (see ETG1000.6 par. 5.6.2.7.2 table 40).

5.2.7 Emergency Error Codes

Emergency Service is used by server for transmitting diagnostic messages to client using MailBox; Error Codes are listed and described in ETG1000.6 par. 5.6.4.2 table 50.

Error Code		Error Register	Diagnostic Data				
Byte (0)	Byte (1)	Byte (2)	Byte (3)	Byte (4)	Byte (5)	Byte (6)	Byte (7)

- Error Code State Transition Errors of state machine:
 (for detailed description see ETG1000.6 par. 5.6.4.3)
 A000hex: transition error from PRE-OP to SAFE-OP
 A001hex: transition error from SAFE-OP to OP
 Encoder errors:
 5000hex: hardware error
 5001hex: diagnostic data (wrong parameters loaded from flash memory)
- Error Register EtherCAT state machine current status (ESM)
- Diagnostic Data information about possible error causes (see ETG1000.6 par. 5.6.4.3.2-5).

5.2.8 AL Status Error Codes

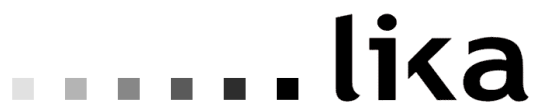
If the state transition requested by the master through the "AL Control Register" is unsuccessful, slave sets to 1 the "Error Indicator Bit" in "AL Status Register" and writes the cause of the error in "AL Status Code Register". Values and descriptions of "AL Status Code" are available in ETG1000.6 par.5.3.2 Table 11.

5.3 File Over EtherCAT (FoE)

Lika encoders are devices that allow the firmware update using the protocol "File over EtherCAT (FoE)".

For any specification on FoE protocol, please refer to "ETG.1000 EtherCAT Specification" document available at the address www.ethercat.org.

Document release	Description
1.0	1st issue
1.1	General review
1.2	Add chapter 5.3



LIKA Electronic
Via S. Lorenzo, 25 - 36010 Carrè (VI) - Italy

Tel. +39 0445 806600
Fax +39 0445 806699

Italy: eMail info@lika.it - www.lika.it
World: eMail info@lika.biz - www.lika.biz