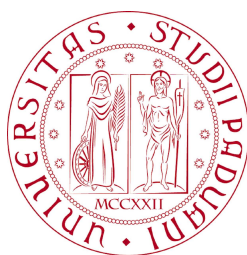


UNIVERSITÀ DEGLI STUDI DI PADOVA
FACOLTÀ DI INGEGNERIA



UNIVERSITÀ DEGLI STUDI DI PADOVA
FACOLTÀ DI INGEGNERIA

—
DIPARTIMENTO DI INNOVAZIONE MECCANICA E GESTIONALE

—
TESI DI LAUREA TRIENNALE IN INGEGNERIA
DELL'AUTOMAZIONE

CONTROLLO DI SERVOMOTORE ELETTRICO TRAMITE ETHERCAT

RELATORE: CH.MO PROF. ING. GIULIO ROSATI

LAUREANDO: ALESSANDRO PRETI

ANNO ACCADEMICO 2009-2010

ai miei genitori...

“ La teoria è quando si sa tutto e niente funziona. La pratica è quando tutto funziona e nessuno sa il perché. Noi abbiamo messo insieme la teoria e la pratica: non c'è niente che funzioni... e nessuno sa il perché ! ”

ALBERT EINSTEIN

Indice

Sommario	XI
Introduzione	XIII
1 EtherCAT	3
1.1 Introduzione	3
1.2 Trasferimento dati	4
1.3 Protocollo	5
1.4 Clock Distribuiti	7
1.5 Prestazioni	9
1.6 Topologia	10
1.7 Profili e protocolli supportati	11
1.8 Master Implementation	13
1.9 Slave Implementation	14
2 Controllo basato su PC	15
2.1 Introduzione	15
2.2 Beckhoff CX 1020	17
2.2.1 Moduli	17
2.2.2 Configurazione	20
2.2.3 Connessioni	22
2.3 Windows CE	23
2.3.1 Tecnologie e supporti	23
2.3.2 Architettura del sistema operativo	25
2.3.3 Application Development	26

2.3.4	Developer Workstation Requirements	28
3	Software utilizzato	29
3.1	TwinCAT	29
3.2	TwinCAT System Manager	32
3.3	TwinCat I/O	35
3.4	eMbedded Visual Studio C++ 4.0	36
4	Controllo di un Asse	39
4.1	Sistema di controllo	39
4.2	Driver Elmo	41
4.2.1	Connessioni	42
4.2.2	Elmo Application Studio (EAS)	48
4.3	Costruzione applicazione	49
4.3.1	Configurazione dispositivo	49
4.3.2	Task aggiuntive	50
4.3.3	Implementazione	52
4.3.4	Test di verifica I/O	54
	Conclusioni	57
A	Caratteristiche motore	59
A.1	Servo motore AC	59
A.2	Dimensioni motore	60
B	Glossario	61
	Bibliografia	65

Sommario

Negli ultimi anni si è diffusa moltissimo nell'automazione la tecnologia dei bus di campo basati su Ethernet per la comunicazione tra dispositivi. L'impiego di queste innovazioni ha garantito maggior velocità e robustezza nello scambio dei dati, aprendo la strada alla realizzazione di applicazioni di controllo basati su PC. In questo lavoro di tesi si tratterà in particolare del fieldbus EtherCAT che fornisce una rete real-time aperta, originariamente sviluppata da Beckhoff. Verranno descritte le caratteristiche di questa tecnologia di comunicazione e verrà introdotta la tematica del controllo basato su PC. L'obiettivo di questo lavoro è quello di utilizzare la rete EtherCAT per il controllo di un servo motore elettrico, utilizzando come controllore un embedded PC CX1020 prodotto da Beckhoff. Verrà quindi implementata un'applicazione che gestisca lo scambio di dati attraverso la base di campo e che effettui il controllo di movimento del servo motore elettrico.

Introduzione

Nel campo dell'automazione a partire dalla metà degli anni 80 la comunicazione ha segnato un punto di svolta con l'avvento dei bus di campo (fieldbus) con i quali si passa da segnali di tipo analogico a digitale. Tra le varie tecnologie, quella basata su Ethernet incominciò ad essere usata anche nell'ambito industriale per compiti di monitoraggio e diagnostica, sviluppandosi progressivamente anche sul campo, ovvero per la comunicazione tra dispositivi.

Nonostante le prestazioni delle CPU dei controllori aumentassero rapidamente, i convenzionali bus di campo tendono a formare un collo di bottiglia nei sistemi di controllo, e spinsero aziende come Beckhoff a sviluppare un nuovo protocollo di comunicazione, EtherCAT. Questo nuovo tipo di fieldbus, che sfruttava un trasferimento di dati "on-the-fly" portò dei vantaggi considerevoli, garantendo una banda maggiore, un know-how diffuso e una maggiore flessibilità e velocità della rete. Oltre ai vantaggi tecnologici l'utilizzo di interfacce standard portò un risparmio sostanziale dal punto di vista economico.

Il lavoro di tesi presenta e sviluppa queste tematiche e descrive la realizzazione del controllo di un servo motore elettrico tramite il fieldbus EtherCAT.

Nel *Primo Capitolo* viene riportata un'analisi approfondita del fieldbus EtherCAT, che descrive il protocollo di comunicazione utilizzato e la strategia di trasferimento dei dati. Vengono anche analizzate le caratteristiche peculiari di questa tecnologia come ad esempio la flessibilità di topologia e le alte prestazioni di utilizzo. Viene descritta brevemente l'implementazione dei dispositivi master e slave EtherCAT.

Nel *Secondo Capitolo* vengono trattate le problematiche e i vantaggi del con-

trollo basato su PC e viene descritto in modo accurato l'embedded PC CX1020 di Beckhoff utilizzato per l'applicazione di controllo. Viene anche analizzato il sistema operativo presente sul dispositivo indicandone le caratteristiche e la struttura.

Nel *Terzo Capitolo* vengono descritti i diversi software utilizzati sia per quanto riguarda la configurazione del controllore e dei dati scambiati sul fieldbus che per quanto riguarda l'ambiente di sviluppo dell'applicazione di controllo.

Nel *Quarto Capitolo* si descrive il lavoro sperimentale effettuato per la realizzazione del controllo di un asse. Viene introdotto il sistema di controllo ed in particolare viene descritto il driver Solo Gold Guitar utilizzato. Infine viene descritta l'implementazione dell'applicazione di controllo.

Capitolo 1

EtherCAT

1.1 Introduzione

Negli ultimi anni la tecnologia dei Fieldbus è divenuta parte integrante e fondamentale per la realizzazione di un sistema di automazione, aumentando in modo significativo la possibilità di integrare e di realizzare applicazioni di controllo basati su PC. Mentre le performance dei controllori CPU, in particolare IPC e embedded PC, sono aumentate rapidamente, i fieldbus tradizionali tendevano a rappresentare un limite di prestazione non facilmente superabile per la realizzazioni di impianti di automazioni [1] .

Un ulteriore fattore negativo era rappresentato dall'architettura del sistema di controllo che consisteva in una serie di sottosistemi subordinati (usualmente ciclici). Il passo fondamentale per superare tali problematiche è stato quello di utilizzare collegamenti Ethernet non solo per creare una rete tra i controllori, ma per sostituire i fieldbus nelle aree di applicazione dei driver e a livello I/O. Le caratteristiche principali per questo tipo di applicazione sono la capacità di supportare il “real-time” per piccole quantità di dati e la diminuzione dei costi di impianto.

Il protocollo EtherCAT, originariamente sviluppato da Beckhoff, riesce a soddisfare tali caratteristiche, mettendo a disposizione una rete “real-time” aperta che garantisce a livello I/O le tecnologie e le potenzialità delle reti Ethernet [2] .

1.2 Trasferimento dati

Il principio fondamentale che sta alla base del sistema di comunicazione EtherCAT è quello di utilizzare un “Pass-through telegram”, ovvero i dati da inviare vengono inseriti in un telegramma che passa tra i vari dispositivi inseriti nella rete e ritorna al controllore. Il telegramma viene modificato “on the fly” durante il passaggio da ogni dispositivo EtherCAT slave (come illustrato in figura 1.1), che legge l’indirizzo contenuto nelle informazioni di processo, e inserisce se necessario i dati di input. Le dimensioni delle informazioni di processo utilizzate da ogni slave sono quasi illimitate e vanno da un minimo di 1 bit al massimo di 60 kbyte (vengono usati più frame se necessario) [3] .

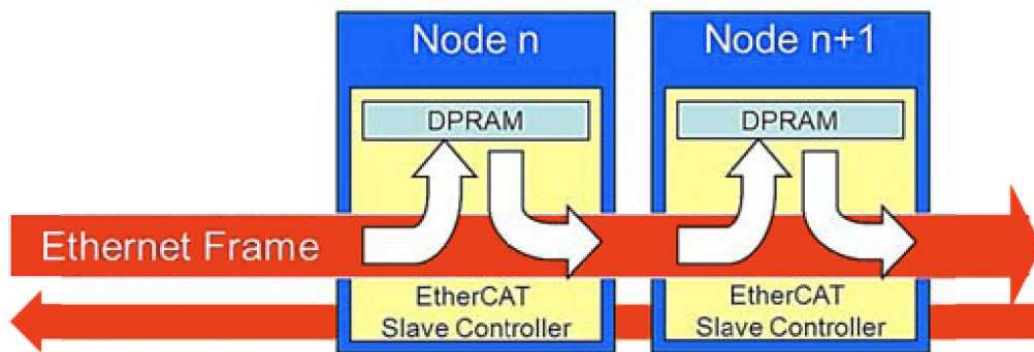


Figura 1.1: Telegramma modificato “on the fly” [4] .

La compilazione delle informazioni di processo viene modificata in modo sincrono ad ogni ciclo di controllo ed è possibile in aggiunta modificarla in modo asincrono tramite “event triggered communication” (comunicazione attivata da eventi). Ogni EtherCAT slave riceve il pacchetto Ethernet inviato dal master e lo ritrasmette automaticamente al dispositivo successivo. L’intero processo avviene sia in fase di lettura che in fase di scrittura con un leggero ritardo per ogni nodo non superiore ai 60 ns [1] .

Il telegramma spedito dal controllore master passa da un dispositivo all’altro fino ad arrivare all’ultimo della configurazione che rispedisce le informazioni modificate lungo il passaggio al master. Il fatto che il pacchetto venga trasmesso continuamente da uno slave all’altro, ne garantisce la continua presenza su tutti

i dispositivi. Con questa tecnica di comunicazione si riesce così ad ottimizzare la velocità di trasferimento e l'ampiezza di banda garantendo un certo grado di determinismo.

Questo tipo di strategia di trasferimento dati è possibile grazie all'aumento della miniaturizzazione dei componenti elettronici, in particolare al miglioramento dell'hardware di comunicazione EtherCAT e all'utilizzo di terminali dotati di ESC (EtherCAT Slave Controller). Questo ha permesso di sostituire le interfacce tipicamente presenti nell'IPC con interfacce intelligenti trasferite nei terminali del sistema EtherCAT. L'architettura degli slave è stata rivisitata e semplificata lasciando il compito di lettura e scrittura dei telegrammi "on the fly" all'ESC, con il vantaggio di operare direttamente sugli I/O digitali, in modo indipendente dalle performance dei μC installati e senza ritardi introdotti dai firmware locali.

1.3 Protocollo

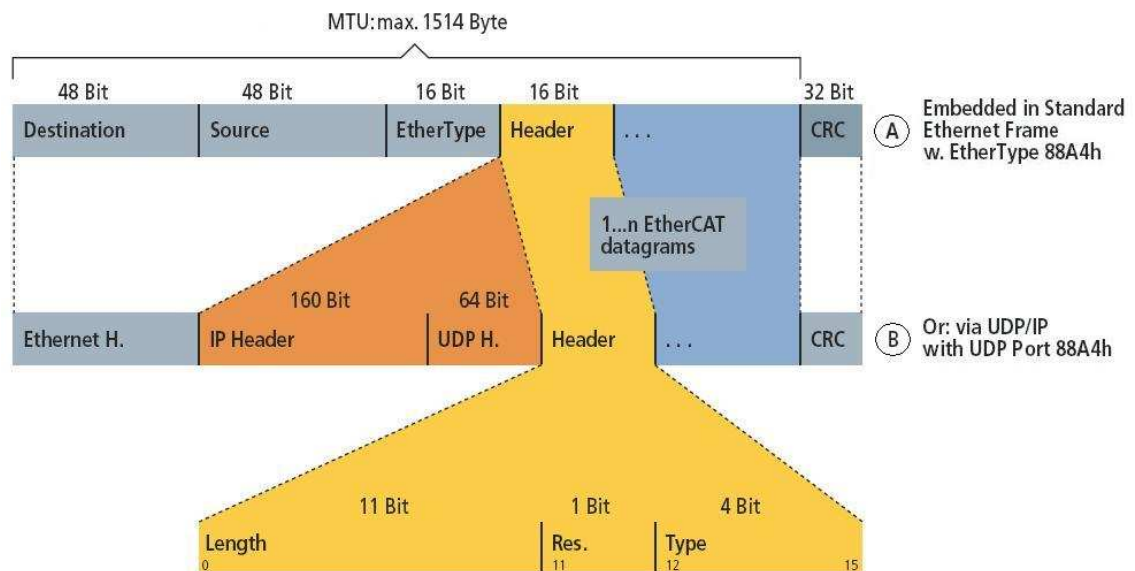


Figura 1.2: Protocollo di trasmissione EtherCAT : A) Via Ethernet (EtherType 0x88A4); B) Via UDP (User Datagram Protocol, UDP port 0x88A4); [1]

Il protocollo EtherCAT implementa un'architettura master/slave usufruendo di una connessione Ethernet in accordo con lo standard IEEE 802.3 che consente l'utilizzo di ogni tipo di controllore con interfaccia Ethernet (master), e di appli-

cazioni che supportano tale standard (es. qualsiasi tipo di monitoraggio disponibile in commercio) [1]. Il protocollo EtherCAT prevede due tipi di implementazioni. Il primo tipo è ottimizzato per processare dati e trasportarli direttamente all'interno del frame Ethernet grazie ad un speciale Ethertype (0x88A4h).

Esso consiste in diversi sotto-telegrammi, definiti Datagram, ognuno dei quali serve una particolare area di memoria dell'immagine processo logico che può arrivare fino a 4GB di grandezza. Ciascun Datagram ha un indirizzo verso uno o più slave; questa combinazione di "dati e indirizzi" (con l'aggiunta del conteggio di validazione WC) forma un telegramma EtherCAT. Il secondo tipo di implementazione consente la comunicazione via UDP/IP (tramite porta UDP 88A4) prevista quando si ricorre ad un router per interfacciarsi con subnet diverse. In questo caso al protocollo vengono aggiunti IP Header e UDP header. Le operazioni di codifica e la presenza di più subnet richiedono un tempo di risposta della rete maggiore con conseguente perdita di determinismo.

Entrambe le implementazioni prevedono la rilevazione dei bit errati durante il trasferimento attraverso la valutazione del checksum CRC: Il CRC a 32 bit ha un polinomio con una distanza di hamming minima di 4 [2]. Il protocollo EtherCAT garantisce una piena trasparenza per la comunicazione TCP/IP e rende disponibili tutte le tecnologie Internet (HTTP, FTP, Webserver ...) senza restrizioni alle prestazioni di real-time [3]. L'ordine dei dati è indipendente dalla posizione fisica dei terminali nella rete e l'inserimento dei dati di input da parte di un qualsiasi terminale viene fatto nel primo datagram vuoto. Tale procedura è garantita dalla Fieldbus Memory Management Unit (FMMU) presente nei slave EtherCAT della rete, con compito di trasferimento dei Datagram e la verifica degli indirizzi. I dati vengono copiati dai terminali I/O nell'allocazione desiderata direttamente senza aver bisogno di un'ulteriore mappatura. La memoria disponibile per gli indirizzi è molto grande e può arrivare fino ai 4 Gbytes.

EtherCAT può contenere più telegrammi, e spesso sono necessari più frames per gestire tutti i telegrammi in un ciclo di controllo. Il protocollo EtherCAT garantisce pure la comunicazione Broadcast, Multicast, slave-to-slave, master-to-master. Per quanto riguarda la comunicazione fra EtherCAT slave esistono due meccanismi utilizzati. Il primo dipende dalla topologia della rete e prevede che se

abbiamo due dispositivi disposti in serie uno prima dell'altro, la comunicazione possa avvenire tra il primo slave EtherCAT e il successivo nello stesso ciclo e molto velocemente. In alternativa la comunicazione può avvenire con uno scambio di informazioni tramite master, impiegando così almeno due cicli [1] .

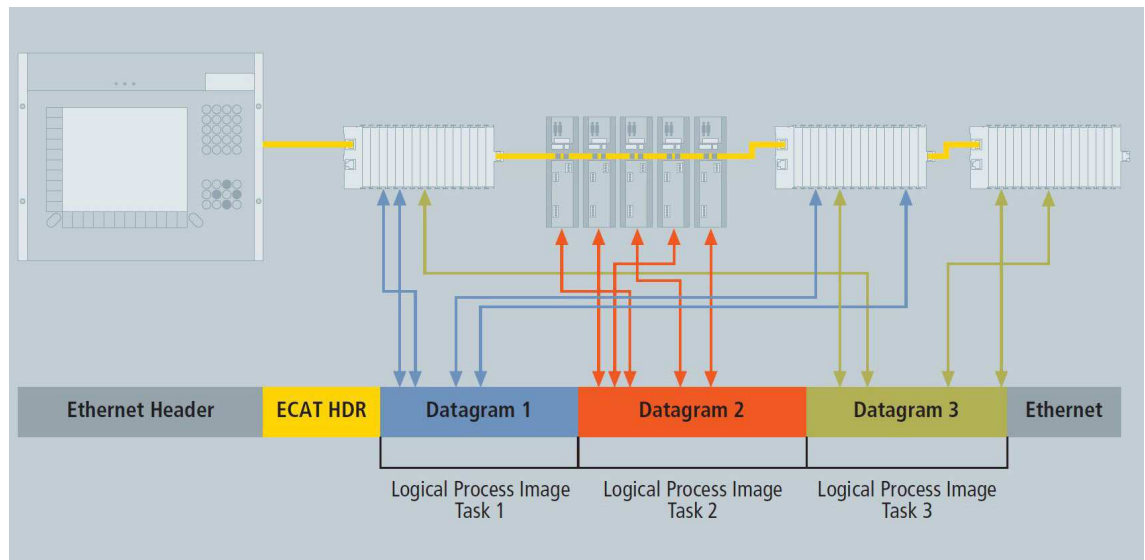


Figura 1.3: Le informazioni di processo sono inserite nei Datagram. [1]

1.4 Clock Distribuiti

Nel caso di processi spazialmente distribuiti, che richiedono azioni simultanee, una sincronizzazione accurata è particolarmente importante. L'approccio utilizzato dalla rete EtherCAT è l'allineamento accurato dei clock distribuiti, come descritto nello standard IEEE 1588, che prevede un alto grado di tolleranza rispetto a possibili ritardi dovuti ad errori di comunicazione.

La sincronizzazione dei dispositivi è completamente basata su una macchina hardware. Dal momento che la comunicazione si avvale di una struttura logica ad anello, ogni nodo della rete può misurare la differenza tra il passaggio di arrivo e quello di ritorno di un frame. Il clock principale, avvalendosi di tali misurazioni, può determinare il ritardo di propagazione di ogni singolo slave in modo semplice ed accurato. I clocks distribuiti sono corretti in base a questo valore, che significa avere un preciso tempo di base della rete con un jitter minore di $1 \mu\text{s}$ [1] .

Tuttavia, l'alta risoluzione dei clock distribuiti non viene solo usata per la sincronizzazione, ma risulta molto importante particolarmente nell'ambito della misurazione della posizione, consentendo l'aumento della velocità di calcolo del controllore, provvedendo ad un'accurata informazione circa il timing locale del dato da acquisire.

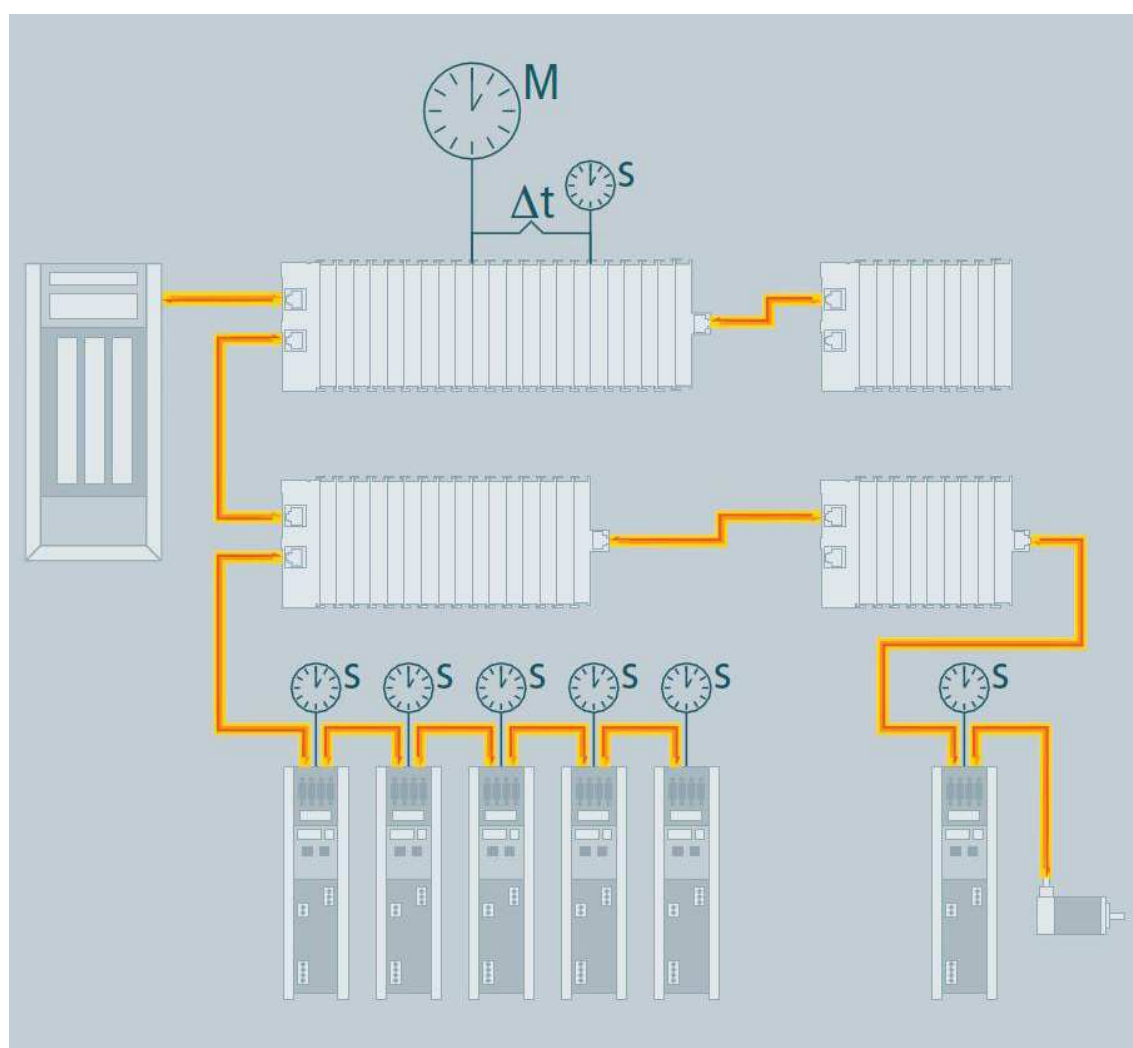


Figura 1.4: Distributed Clock :I clock di ogni singolo dispositivo vengono impostati sulla base dei tempi a livello di rete. [1]

1.5 Prestazioni

Il protocollo EtherCAT è progettato per raggiungere elevate prestazioni e per gestire un alto numero di canali in applicazioni di controllo. Poichè la fase di lettura e scrittura dello slave può avvenire nello stesso frame, la struttura del telegramma EtherCAT è ottimizzata per gestire sistemi di I/O decentralizzati. L'utilizzo di frame Ethernet, che comprende i dati di molte periferiche sia in ingresso che in uscita, fa aumentare il data rate utilizzabile di oltre il 90%. Sono pienamente utilizzate le caratteristiche full duplex di 100BaseTX così che l'effettivo data rate diventa maggiore di 100 Mbps ($> 90\%$ di 2×100 Mbps) [3].

Fino a 1486 byte di dati di processo possono essere scambiati con un unico frame. Inoltre, la completa elaborazione del protocollo avviene all'interno dell'hardware e pertanto è indipendente dalle prestazioni della CPU e da ogni componente software. Ad esempio, tramite l'accesso diretto alla memoria (DMA), è possibile trasferire i dati tra la scheda di rete e il processore master oppure gli I/O dello slave con coinvolgimento minimo della CPU. Gli slave, che dispongono di unità fieldbus per la gestione della memoria (FMMU) dedicati al trasferimento dei PDO e la verifica degli indirizzi, sono incaricati di mappare i telegrammi appropriati riducendo in questo modo il lavoro del master.

La tecnica dei clock distribuiti consente di sincronizzare gli assi con una deviazione minore di 1 microsecondo. Tali caratteristiche consentono di raggiungere determinate prestazioni come [1]:

- 256 I/O digitali in $11 \mu s$
- 1000 I/O digitali distribuiti su 100 nodi in $30 \mu s$
- 200 I/O analogici (16 bit) in $50 \mu s$, corrispondenti a $20 kHz$ di sampling rate
- 100 "Servo-axes" (con 8 Byte In + Out) ogni $100 \mu s$
- 12000 I/O digitali in $350 \mu s$

1.6 Topologia

EtherCAT supporta numerosi tipi di topologia di rete : linea, albero o stella (vedi figura 1.5). La struttura di linea viene riconosciuta dal fieldbus e diventa così disponibile anche per Ethernet, senza le limitazioni quantitative dovute alla presenza di switch e di hub [1] . Particolarmente utile per il cablaggio del sistema è la possibilità di avere una combinazione di linee e di rami : le interfacce richieste sono già predisposte su molti dispositivi (ad esempio sui moduli I/O), senza bisogno di ulteriori switches.

La flessibilità di collegamento e il basso costo di cablaggio sono ulteriormente ottimizzati attraverso l'utilizzo di cavi differenti : cavi industriali Ethernet standard per trasferire segnali in modalità 100BASE-TX (consente l'utilizzo di un cavo di lunghezza 100 *m* tra due dispositivi, fino a 65535 dispositivi possono essere collegati assieme e la dimensione della rete può superare i 500 *Km*); fibre ottiche 100BASE-FX o Plastic Optical Fibres per integrare il sistema con applicazioni speciali (consente l'utilizzo di un cavo di lunghezza variabile da 50 *m* a 2000 *m* tra due dispositivi).

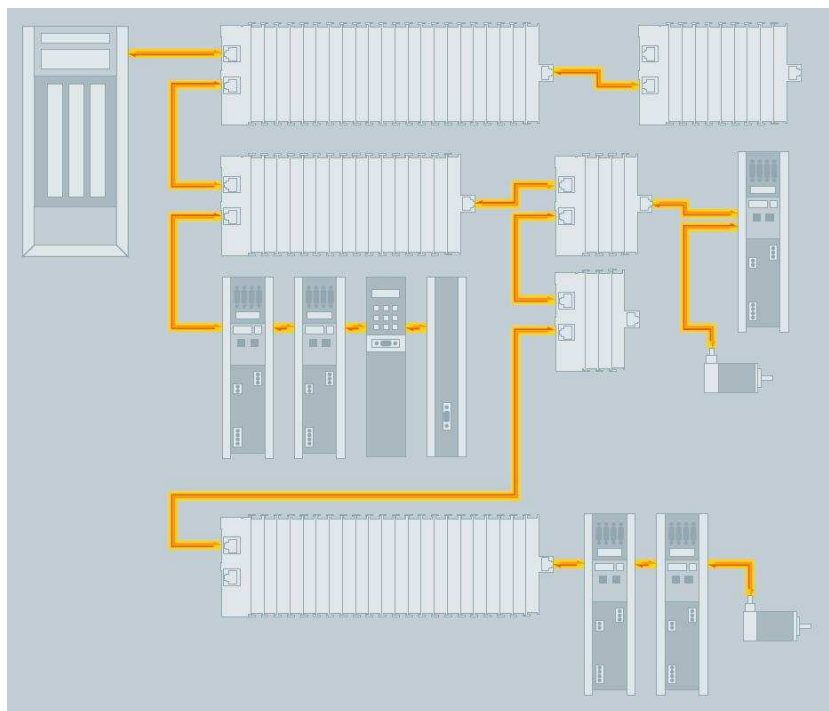


Figura 1.5: Topologia flessibile : linea, albero o stella [1] .

1.7 Profili e protocolli supportati

L'utilizzo di dispositivi con profili differenti o di altri protocolli viene garantito grazie a semplici interfacce che aiutano sia gli utenti che i produttori di dispositivi nel gestire il passaggio da bus di campo esistenti a EtherCAT (vedi figura 1.6). Allo stesso tempo, la specificazione EtherCAT si mantiene semplice perché tutti i protocolli sono opzionali. Il produttore del dispositivo deve solo implementare il protocollo che l'applicazione del dispositivo necessita. Diversi profili di dispositivi e protocolli possono essere supportati su rete EtherCAT :

- CAN application protocol over EtherCAT (CoE) : dispositivi CANopen e profili di applicazione sono disponibili (componenti di I/O, azionamenti, encoder, valvole proporzionali idraulici, per esempio). EtherCAT è in grado di fornire gli stessi meccanismi di comunicazione di CANopen (object Dictionary, PDO Processo Data Objects e SDO Service Data Objects).
- Servo drive profile in accordo con IEC 61800-7-204 (SERCOS over EtherCAT SoE) : l'interfaccia SERCOS è riconosciuto come un'interfaccia di comunicazione real-time ad alto rendimento. Il profilo SERCOS per servo unità e la tecnologia di comunicazione sono coperti dallo standard IEC 61800-7-204. Il canale di servizio, e quindi l'accesso a tutti i parametri e le funzioni, si basa sulla mailbox EtherCAT. I dati di processo, con SERCOS in forma di AT e dati MDT, vengono trasferiti utilizzando il meccanismo e il protocollo EtherCAT.
- Ethernet over EtherCAT (EoE) : la tecnologia EtherCAT non solo è compatibile con Ethernet, ma è caratterizzata anche da apertura particolare "by design". Il protocollo tollera infatti altri servizi e protocolli basati su Ethernet sulla stessa rete fisica (con minima perdita di prestazioni). Non vi è alcuna restrizione sul tipo di dispositivo Ethernet che può essere collegato all'interno della rete EtherCAT attraverso una switchport. I frame Ethernet sono inseriti tramite protocollo EtherCAT, rappresentando l'approccio standard per le applicazioni Internet (Ad esempio VPN, PPPoE, DSL, ecc.). La rete EtherCAT è completamente trasparente per i dispositivi Ethernet,

e le caratteristiche real-time non vengono compromesse. Tutte le tecnologie Internet possono dunque anche essere utilizzate in ambiente EtherCAT : web server integrato; e-mail; trasferimento FTP ecc.

- File Access over EtherCAT (FoE) : questo protocollo molto semplice, simile al TFTP, consente l'accesso a qualsiasi struttura dati del dispositivo. E' possibile quindi caricare sul dispositivo firmware standardizzati, a prescindere dalla presenza di supporto TCP / IP.

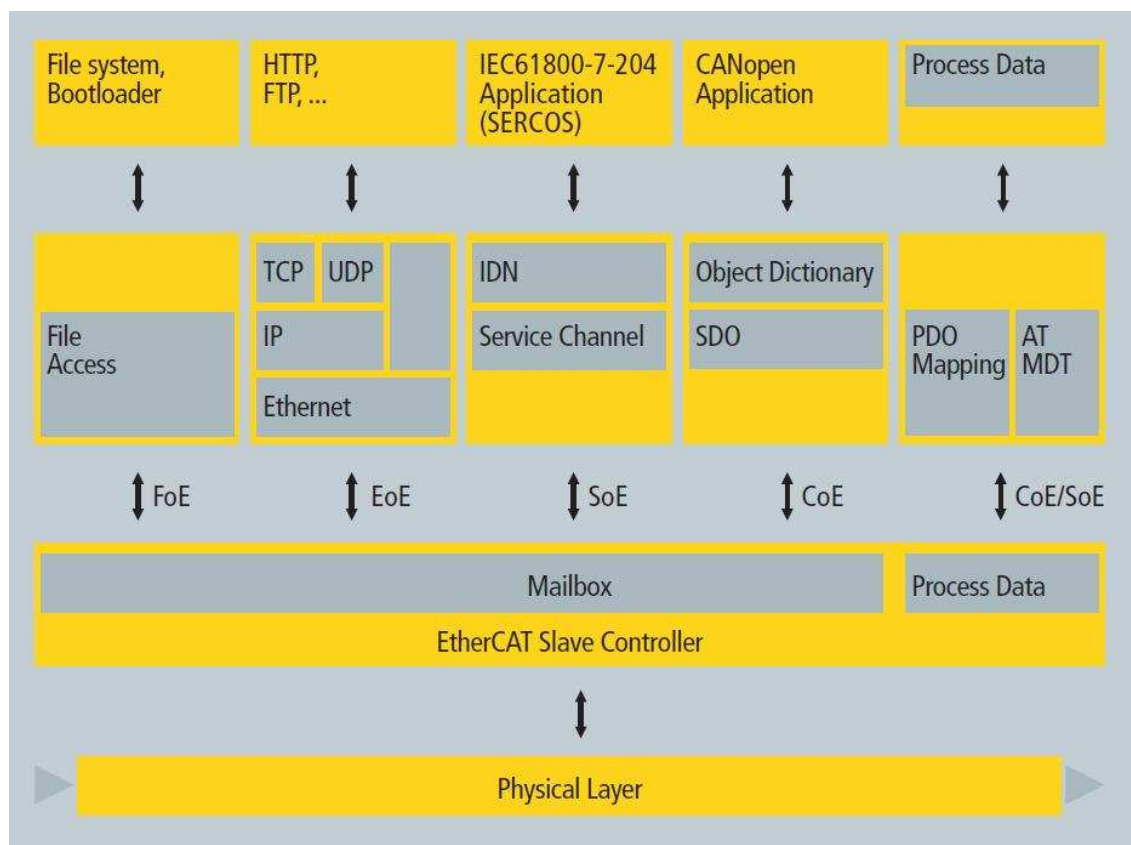


Figura 1.6: Diversi profili di dispositivi e protocolli possono coesistere fianco a fianco [1] .

1.8 Master Implementation

EtherCAT comunica un massimo di 1486 *byte* di dati di processo distribuiti con un solo frame Ethernet. Il sistema EtherCAT tipicamente ha bisogno solo di uno o due frame per ogni ciclo per la comunicazione con tutti i nodi. Pertanto il master EtherCAT non necessita di un apposito processore per la comunicazione. Il basso costo computazionale della comunicazione EtherCAT consente di sfruttare la CPU in maniera minimale per gestire questo compito lasciando maggior spazio per elaborare le applicazioni di controllo.

L'implementazione di un master EtherCAT può essere molto facile, in particolare per sistemi di controllo di piccole e medie dimensioni (ad esempio un PLC con una singola immagine di processo che non supera i 1486 byte). L'immagine di processo non viene mappata nel master EtherCAT ma negli slaves gravando ulteriormente meno la CPU. Gli EtherCAT masters possono essere implementati su una vasta gamma di RTOS (Real-Time Operating System), inclusi : eCos, INtime, MICROWARE OS-9, On Time RTOS-32, Proconos OS, Real-Time Java, RT Kernel, RT Linux, RTCX Quadros, RTAI Linux, PikeOS, Linux with RT-Preempt, QNX, VxWin + CeWin, VxWorks, Windows CE, Windows XP/XPE con CoDeSys SP RTE, Windows XP/XPE con TwinCAT RT e XENOMAI Linux [1] . Servizi di implementazione sono disponibili da una varietà di fornitori e per una varietà di piattaforme hardware. In un ambiente embedded l'implementazione del Master EtherCAT deve avere alcuni requisiti speciali : l'ambiente hardware comprende una variazione massima di micro controllori disponibili (8, 16 o 32 bit) ; l'utilizzo di diversi chip MAC per affrontare diversi metodi di accesso ai dati (accesso DMA, accesso IO, accesso Dual Port alla RAM); l'utilizzo di differenti sistemi operativi o, in alcuni casi, di nessun sistema operativo.

Il beneficio di EtherCAT in un ambiente embedded è la possibilità di avere un master scalabile utilizzando alcune caratteristiche o protocolli speciali per sfruttare in modo ottimale la memoria e la CPU. Questo è possibile grazie al fatto che un master non necessita di un hardware speciale ma può essere implementato in software.

1.9 Slave Implementation

Lo slave EtherCAT non ha bisogno di un microcontrollore integrato, in quanto la sua architettura prevede una semplice interfaccia I/O implementata solo con un ESC (EtherCAT Slave Controller), un PHY (Physical Layer) e i connettori RJ45. La process data interface (PDI) per l'applicazione di uno slave è un'interfaccia I/O 32 bit. Lo slave senza parametri configurabili non ha bisogno di software o di mailbox protocol. L'EtherCAT State Machine è gestito nel ESC che prende le informazioni di boot-up dalla EEPROM che supporta anche le informazioni di identità dello slave.

Gli slave più complessi, che sono configurabili, montano una CPU host a bordo, collegata al ESC con un interfaccia (8- 16 bit) in parallelo a o tramite una connessione seriale (SPI). Diversi fornitori offrono servizi di implementazione degli slave compresi un'integrazione hardware e software. Le varie EtherCAT Controller Slave sono supportate da corrispondenti evaluation kits, rendono tutte le interfacce del controllore facilmente accessibili. Dal momento che con EtherCAT non sono necessari potenti processori per la comunicazione, il kit sopra indicato contiene un microprocessore a 8 bit che può essere anche utilizzato come host CPU. Il kit viene fornito con uno slave host software, l'equivalente di uno stack di protocolli in codice sorgente, e un pacchetto di riferimento software per il master.

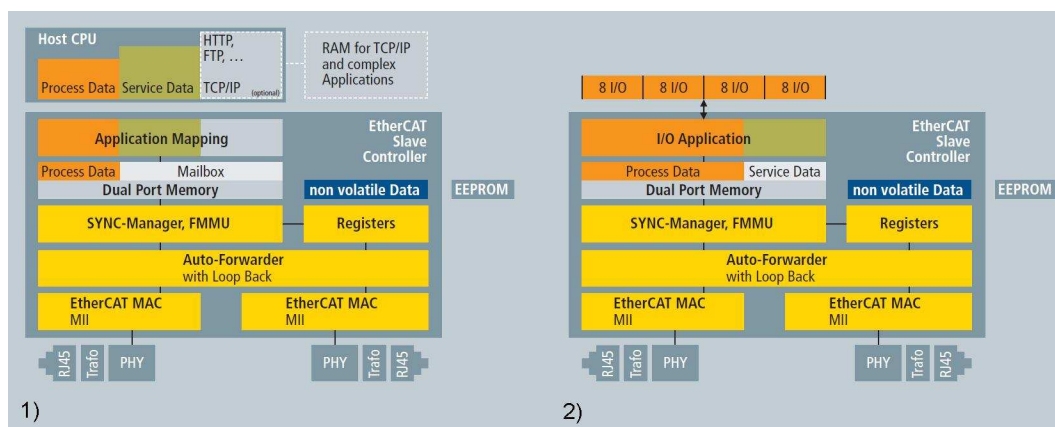


Figura 1.7: Implementazione Hardware dello slave : 1) con host CPU; 2) con interfaccia I/O diretta [1] .

Capitolo 2

Controllo basato su PC

2.1 Introduzione

Il personal computer negli ultimi decenni ha conosciuto una storia di successo senza precedenti ed è diventato una parte importante non solo della vita quotidiana, ma anche negli ambienti industriali. Le piattaforme hardware e software PC basati su tecnologie proprietarie, ma anche su sistemi standard e aperti, hanno rappresentato il fattore di crescita tecnologica e di integrazione digitale tra il mondo dell'information technology e quelli dell'automazione industriale [5]. Insieme con software associato, i PC hanno svolto una vasta gamma di differenti compiti di automazione: quali il controllo di macchine, processi o sistemi logistici; la messa in rete dei componenti del sistema; l'acquisizione di dati o di immagini. L'eccellente livello di flessibilità e di scalabilità offerto dalla tecnologia di controllo basata su PC nel soddisfare ai compiti di controllo classico, ha portato un maggior impiego di questi dispositivi rispetto all'hardware PLC.

I sistemi di controllo basati su IPC (Industrial PC) e su embedded PC hanno avuto un'innovazione e uno sviluppo continuo, principalmente ricercando un aumento di prestazioni di calcolo e una riduzione sostanziale dei costi. Grande sviluppo ha avuto l'utilizzo di embedded PC, soprattutto per compiti di controllo di medie prestazioni, offrendo ed unendo le potenzialità dei PC industriali e dell'hardware PLC in un unico dispositivo dedicato. I sistemi embedded dispongono di processore che può avere varie architetture come ARM, PowerPC, x86,

MIPS, ecc [6] . Alcuni dispositivi non dispongono di sistema operativo, mentre la maggior parte utilizzano un sistema operativo “real-time” (per esempio Windows CE).

L’ambiente in cui vengono utilizzati ha favorito la scelta di tali dispositivi, rispetto ai PC standard di tipo office o generale con cui condividono i componenti fondamentali, soprattutto per la robustezza e l’elevato grado di protezione in quanto utilizzati in applicazioni specifiche e in ambienti industriali caratterizzati da condizioni ambientali sfavorevoli per temperatura, umidità, sollecitazioni meccaniche, vibrazioni e interferenze elettromagnetiche [7] .

Altra caratteristica dei sistemi embedded è la disponibilità di un impiego a lungo termine in condizioni ottimali, in modo affidabile e garantendo un funzionamento in sicurezza sia rispetto a cose o persone che rispetto ai dati che devono essere protetti da manipolazioni. Con dispositivi dedicati la necessità di interventi di manutenzione viene ridotta al minimo perché spesso le applicazioni si trovano in siti poco accessibili con necessità di operazioni complesse e costose (basta pensare ai costi di un fermo linea in un impianto di produzione per grandi volumi oppure alle difficoltà di intervenire su sistemi di controllo installati in piattaforme “offshore”) [5] .

I componenti di sistemi embedded devono essere anche in grado di assicurare facilità di integrazione e installazione, ingombro minimo, bassi costi, basso consumo di energia, interconnessione con diverse interfacce uomo-macchina, compatibilità con i diversi bus di campo (Profibus, Profinet, DeviceNet, Ethernet IP, CANopen, EtherCAT, ControlNet, Modbus e Interbus) che collegano dispositivi come sensori e attuatori con controllori e PLC in una rete industriale.

Fondamentale infine, per applicazioni di controllo, il monitoraggio delle variabili e l’acquisizione in tempo reale dei dati dal campo. Per questo è necessario che software e firmware utilizzati per la programmazione della strategia di controllo siano a prestazioni eccellenti, modulari per adattarsi alle esigenze specifiche dell’applicazione, programmabili ad alto livello (ad esempio in C++), oltre a essere in grado di fornire tempi di risposta con la dinamica basata sul tempo o sugli eventi dei processi sotto controllo.

2.2 Beckhoff CX 1020

L'embedded PC di Beckhoff CX1020 fa parte della serie di dispositivi denominati CX che uniscono la tecnologia dei sistemi di controllo basati su PC al livello hardware dei terminali I/O per formare un'unità modulare da inserire in un qualsiasi pannello di controllo [8]. Il CX1020 estende la famiglia dei prodotti CX con una versione con performance della CPU più avanzate e permette la connessione diretta dei Bus Terminals e dei terminali EtherCAT.

Questo dispositivo è equipaggiato con un Intel Celeron M CPU con 600 MHz , che garantisce da una parte un'elevata prestazione di calcolo e dall'altra un minor dispendio di energia, grazie all'utilizzo di un "ultra-low core voltage" che garantisce una bassa dissipazione del calore termico di soli 7 Watt TDP (Thermal Design Power) [9]. Questo ha garantito soprattutto dei vantaggi nella progettazione del dispositivo, con una riduzione sostanziale delle dimensioni e con l'utilizzo di un dissipatore metallico fisso. Il CX1020 non monta quindi una ventola di raffreddamento sul dispositivo, e usando una scheda flash come boot e memoria, ha ridotto a zero l'utilizzo di oggetti rotanti sul controllore migliorando così il MTBF (Mean Time Between Failures) dell'intero sistema.

Il dispositivo è strutturato in maniera modulare, esso infatti è formato da diversi componenti che possono essere assemblati dall'utente in modo da ottenere ed utilizzare la configurazione più adatta ad ogni specifico impiego. Il CX1020 Embedded PC è stato sviluppato per supportare e rendere ottimale l'integrazione con EtherCAT consentendo però diverse opzioni di connessione con gli altri tipi di fieldbusses. E' installato sia il sistema operativo, Microsoft Windows CE, che il programma di Beckhoff TwinCAT, che svolge sia funzioni di configurazione che funzione più avanzate di PLC e Motion Control (quest'ultime necessitano di licenza specifica).

2.2.1 Moduli

Il CX 1020 è caratterizzato da una struttura modulare : infatti è costituito da vari moduli, ognuno con caratteristiche ben precise, con la possibilità di variare la configurazione dell'embedded PC a seconda delle esigenze di impiego. Il modulo

base è costituito dalla CPU, da una memoria flash e dalle varie interfacce disponibili a seconda della configurazione. A questo modulo possono essere aggiunti diversi altri tipi di moduli e terminali come :

- Accoppiatori di Bus
- Terminali Bus
- Terminali EtherCAT
- Power Supply Units
- Altro (schede, switch, accessori)

Power Supply Units L'unità di alimentazione CX1100-0004 è collegata al sistema tramite bus interno PC104 e dispone di caratteristiche aggiuntive che vanno oltre al semplice compito di unità di potenza. Essa infatti integra una NOVRAM che permette il salvataggio delle informazioni di processo qualora ci fosse un problema di alimentazione, un display LCD con due linee di 16 caratteri, entrambi usati per visualizzare messaggi.

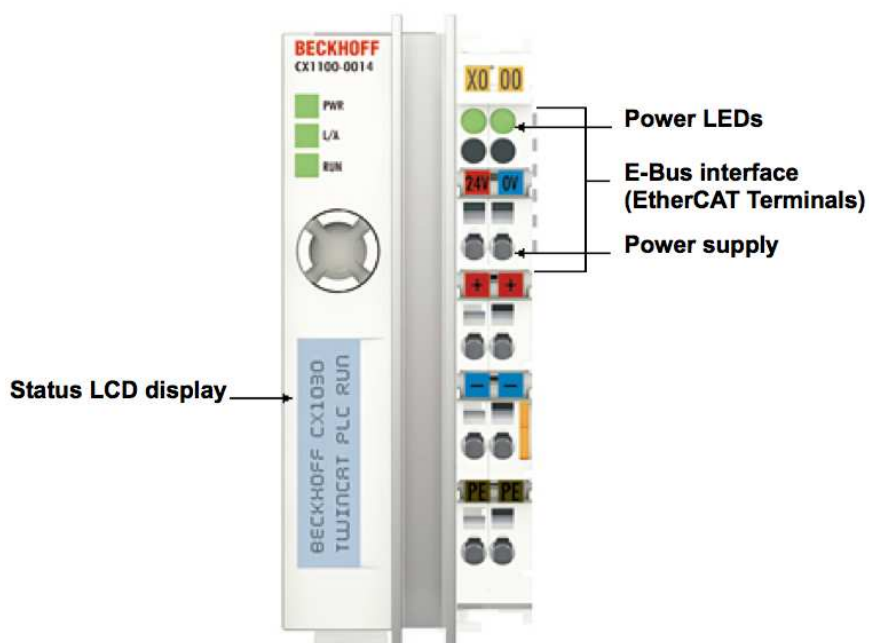


Figura 2.1: CX1100-0004 [9] .

CX1100-0004 power supply unit è equipaggiata con un'interfaccia I/O che permette di connettere i terminali EtherCAT in modo diretto al Embedded PC. Le informazioni provenienti dagli I/O vengono inserite direttamente nella memoria principale della CX1100-0004 CPU, senza il bisogno di una DPRAM [9]. La potenza fornita alimenta il sistema CX con un voltaggio di 24 V DC (-15% / $+20\%$) [9] e viene distribuita a tutti i terminali per mezzo dell'E-Bus.

Terminali EtherCAT Il CX1020 Embadded PC è stato sviluppato per supportare e rendere ottimale l'integrazione con EtherCAT. I terminali I/O EtherCAT sono concepiti in modo da integrarsi perfettamente con il dispositivo e ognuno di essi è fornito di guide di montaggio che permettono un facile collegamento. L'unità di alimentazione CX1100-0004 funge da accoppiatore tra i terminali EtherCAT e l'Embadded PC.

I collegamenti elettrici vengono realizzati automaticamente montando i vari componenti, in particolare per ogni terminale sono presenti sei contatti a molla che consentono il trasferimento dei dati attraverso E-Bus e l'approvvigionamento di potenza (fino a 24 Volt) o per tensioni superiori tramite morsetti di alimentazione. I terminali a nostra disposizione sono:

- EL1202 : Terminale di ingresso digitale a due canali, con tensione di ingresso a 24 Volt DC, Ton / Toff 1 microsecondo.
- EL1252 : Terminale di ingresso digitale a due canali con "time stamp" (tecnologia che permette di identificare, con una risoluzione di 1 ns, il momento del cambiamento di stato).
- EL1262 : Terminale di ingresso digitale a due canali con "oversampling" (I segnali vengono sovracampionato in modo regolabile con un multiplo intero del tempo di ciclo del bus).
- EL2202 : Terminale di uscita digitale a due canali, con tensione di uscita a 24 Volt DC, Ton / Toff 1 microsecondo e con "push-pull" (consente di collegare l'uscita attivamente a 24 Volt , 0 Volt o ad alta resistenza)

- EL2252 : Terminale di uscita digitale a due canali con “time stamp” (tecnologia che permette di identificare, con una risoluzione di 1 ns , il momento del cambiamento di stato).
- EL2262 : Terminale di uscita digitale a due canali con “oversampling” (i segnali vengono sovracampionato in modo regolabile con un multiplo intero del tempo di ciclo del bus).
- EL6751 : Terminale master CANopen che può funzionare sia da master che da slave e permette l’integrazione di tutti i dispositivi CANopen.
- EK1110 : Terminale di capolinea E-bus collegato alla fine del blocco dei terminali EtherCAT. Offre la possibilità di collegare un cavo Ethernet con connettore RJ-45, prolungando la rete EtherCAT per mezzo di un filo elettrico isolato fino a 100 m .

2.2.2 Configurazione

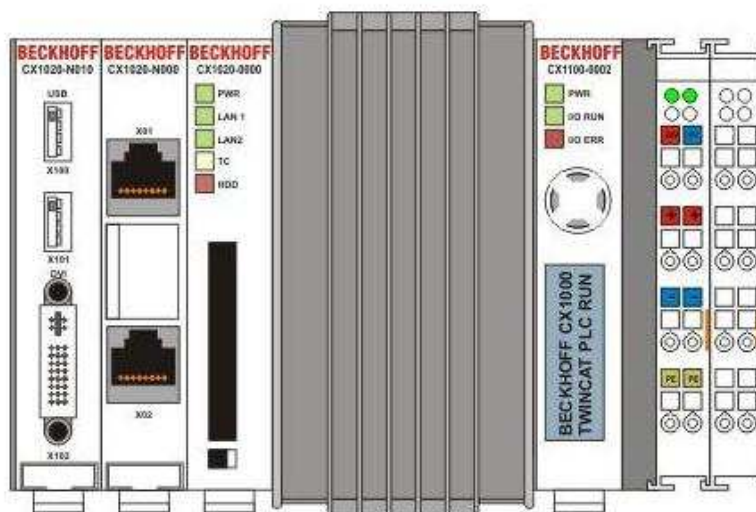


Figura 2.2: Configurazione CX1020-0111 [9] .

Il modulo di base del CX1020 è formato dalla componente hardware della CPU, da una memoria Flash da 64 MB e da due interfacce RJ-45 che sono connesse ad uno switch interno, offrendo una semplice opzione per creare una topologia di

linea senza il bisogno di aggiungere switch Ethernet. Gli altri componenti della serie CX possono essere connessi al modulo di base per mezzo dell'interfaccia PC104 che è disponibile su entrambi i lati del modulo di base.

La configurazione del CX1020 può essere identificata dal codice presente sul modulo base. Esso infatti presenta quattro cifre dopo la sigla del dispositivo, che descrivono sia i moduli che i software installati (come riportato in figura 2.3).

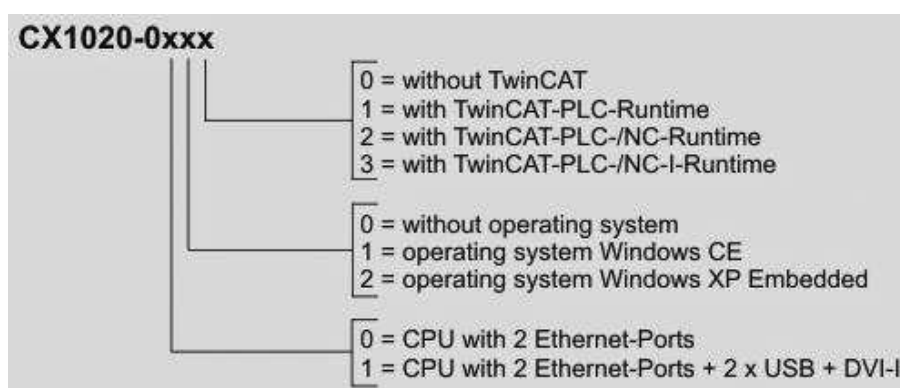


Figura 2.3: Configurazioni di base possibili [9] .

L'embedded PC utilizzato è identificato dal codice CX1020-0111, includendo in aggiunta al modulo base un'unità supplementare di potenza (CX1100-0004), un'interfaccia DVI e due USB-2.0, ed è fornito di sistema operativo Microsoft Windows CE e di software TwinCAT-PLC-Runtime. L'unità supplementare di potenza permette la connessione diretta al dispositivo dei terminali EtherCAT e dell'uscita EtherCAT (EK1100).

Il modulo CX1020-N010 DVI / USB, alimentato e connesso al modulo di base per mezzo del bus PC104, permette di utilizzare pannelli di controllo o monitor standard DVI o VGA attraverso le uscite DVI o USB. Come per un normale PC possono essere connessi, tramite porta USB, altri dispositivi, quali ad esempio stampante, scanner, mouse e tastiera.

2.2.3 Connessioni

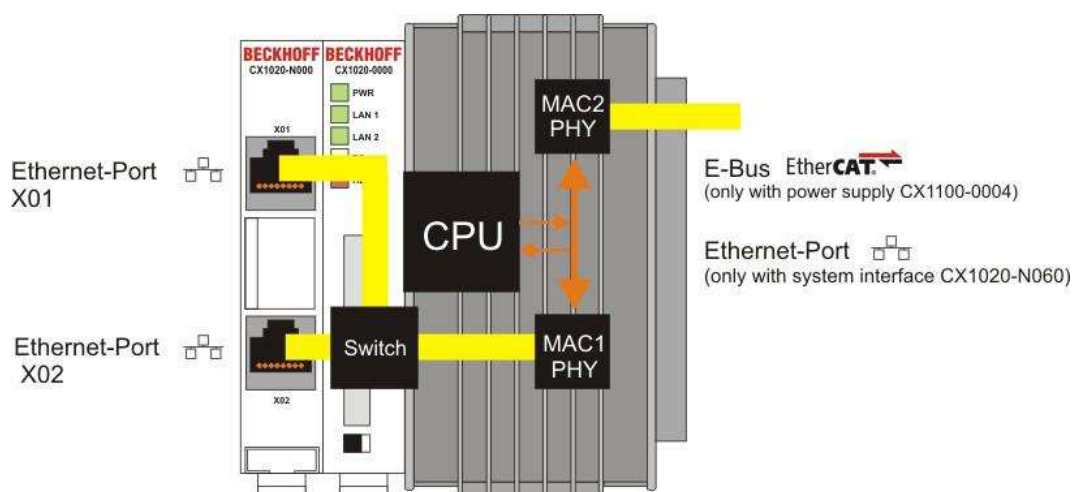


Figura 2.4: Struttura dei componenti di rete [9] .

Nella configurazione a nostra disposizione sono presenti due interfacce Ethernet RJ-45 (porta X001 e porta X002). L'embedded PC CX1020 è dotato di due blocchi MAC (Media Access Control). Il primo (MAC1) gestisce le interfacce di rete per le due porte Ethernet, che sono connesse per mezzo di uno switch, consentendo di variare le configurazioni della topologia della linea.

Il secondo blocco (MAC2) gestisce il bus PC104 che costituisce un'interfaccia Ethernet interna per la comunicazione EtherCAT [9] . Il suo livello fisico di segnale viene trasformato nell'E-bus tramite un sistema aggiuntivo, utilizzando per esempio l'unità di alimentazione CX1100-0004 che consente la connessione diretta dei terminali EtherCAT con l'embedded PC. L'E-bus attraversa ogni singolo terminale come un segnale LVDS (Low-Voltage Differential Signaling) fisico con un ritardo massimo di 10 ns per ogni terminale. La porta Ethernet X001 viene utilizzata per collegare l'embedded PC ad un desktop PC, tramite cavo cross o rete locale, mentre la porta X002 viene utilizzata per collegare il dispositivo con altri controllori della rete industriale.

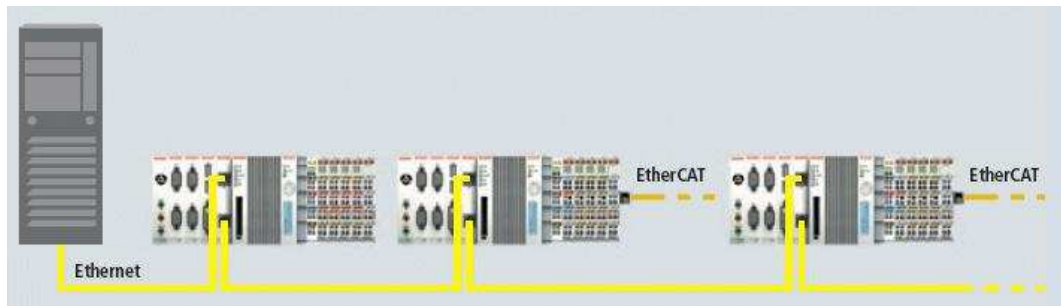


Figura 2.5: Connessioni di rete [9] .

2.3 Windows CE

Windows Embedded CE è un sistema operativo e una piattaforma di sviluppo che offre un sistema operativo a 32 bit, real-time e modulare, un kernel unificato e potenti strumenti di sviluppo embedded, progettato per dispositivi intelligenti, connessi e orientati ai servizi, da sistemi GPS portatili a controller industriali real-time [10] . In opposizione a molti altri sistemi operativi, Windows Embedded CE è stato creato in maniera tale da non essere legato ad una specifica architettura del processore o implementazione hardware. L'unica limitazione è stata l'utilizzo di un processore a 32 bit.

La versione più recente, Windows Embedded CE 6.0 supporta quattro architetture di processore (ARM, MIPS, SH4, e x86) e un numero considerevole di implementazioni offerte dai vari produttori di processori. Quest'ultima versione del sistema operativo Microsoft supporta vari sottoinsiemi di Microsoft Win32 API e diverse interfacce di programmazione. Come tutti i sistemi embedded che utilizzano OS Microsoft, anche il CX1020 di Beckhoff è pre-configurato con sistema operativo Windows CE 6.0 Immagine e dispone di licenza Windows CE 6.0.

2.3.1 Tecnologie e supporti

Windows Embedded CE 6.0 offre allo sviluppatore una vasta gamma di opportunità, supporti e di tecnologie.

In particolare garantisce sistemi rapidi per lo sviluppo di applicazioni tipo [10]

:

- AYGShell API, che assicura la compatibilità con le applicazioni Windows Mobile.
- .NET Compact Framework 2.0 e 3.5, tra cui Active Template Library (ATL), Microsoft Foundation Classes (MFC), Windows Template Library (WTL), Standard Template Library (STL), ActiveSync, Exchange Server Client, Global Positioning System (GPS) driver, Speech API 5.0, Windows Messenger, Pocket Outlook Object Model (POOM), Extensible Markup Language (XML) e Microsoft SQL Server Compact 3.5.
- Simple Network Management Protocol (SNMP).
- 3,9 milioni di righe di codice sorgente, il 100 per cento del codice sorgente del kernel.
- Production Quality OAL (PQOAL), un set di librerie e il codice sorgente per la creazione dell'OAL.

Dispone delle tecnologie di comunicazione [10] :

- Transmission Control Protocol/Internet Protocol (TCP/IP), IPv4, IPv6, Network Driver Interface Specification (NDIS) 5.1, Winsock 2.2, Internet Protocol security (IPsec) v4.
- Personal area network (PAN), local area network (LAN), wide area network (WAN), Bluetooth, 802.11.
- SOAP, Object EXchange (OBEX), Lightweight Directory Access Protocol (LDAP) client, Remote Desktop Protocol (RDP).
- VoIP, real-time communications (RTC), Session Initiation Protocol (SIP).
- Radio Interface Layer (RIL), support for Short Message Service (SMS), Wireless Application Protocol (WAP), support for Subscriber Identity Module (SIM) cards.

- Remote API (RAPI) and RAPI2, Point-to-Point Protocol over Ethernet (PPPoE), Telephony Application Programming Interface (TAPI), virtual private network (VPN).

Fornisce le tecnologie Server-side [10] :

- Telnet, File Transfer Protocol (FTP), server message block (SMB), Common Internet File System (CIFS), Microsoft Message Queuing (MSMQ), Remote Access Service (RAS), Point-to-Point Tunneling Protocol (PPTP), Universal Plug and Play (UPnP).
- Server Web con supporto per Active Server Pages (ASP).

Offre diverse applicazioni multimediali come [10] :

- DirectDraw, DirectShow, Direct3D.
- Windows Media Player, Windows Media Audio (WMA), MP3.
- Internet Explorer.
- interfaccia DVD.

2.3.2 Architettura del sistema operativo

Microsoft Windows Embedded CE 6.0 è un sistema operativo real-time (ogni processo deve terminare entro una scadenza temporale prefissata), che supporta il multitasking e gira su architetture di processori diversi, tra cui ARM, MIPS, x86 e SH4, operando nello spazio di indirizzamento virtuale di 4 gigabyte (GB) [10]. Il kernel di sistema utilizza i 2 GB di memoria virtuale superiori, mentre i restanti vengono impiegati per il processo del utente attivo.

Windows Embedded CE 6.0 supporta fino a 32000 processi utente, e tale numero effettivo viene limitato dalle risorse di sistema. I processi utente includono speciali sotto-processi che rendono l'interfaccia di programmazione (API) disponibile per le applicazioni utente. La shell del sistema rende l'interfaccia principale a disposizione del utente. Il nucleo del sistema operativo è il processo Nk.exe, nel quale sono state caricate le librerie dinamiche responsabili di varie funzionalità di

sistema. Il sistema API è disponibile per tutte le applicazioni attraverso la libreria `coredll.dll`, che è collegata a tutti i moduli eseguibili del sistema operativo.

Il kernel di Windows Embedded CE 6.0 interagisce con l'hardware attraverso la OAL, che nasconde l'implementazione del processore ed esegue l'inizializzazione del hardware. La costruzione del sistema operativo è basata sulla memoria virtuale, che viene fornita dall'OS in modo flessibile ed efficace per gestire le limitate risorse di memoria fisica. L'architettura della memoria virtuale è una mappatura degli indirizzi di memoria virtuale in indirizzi fisici.

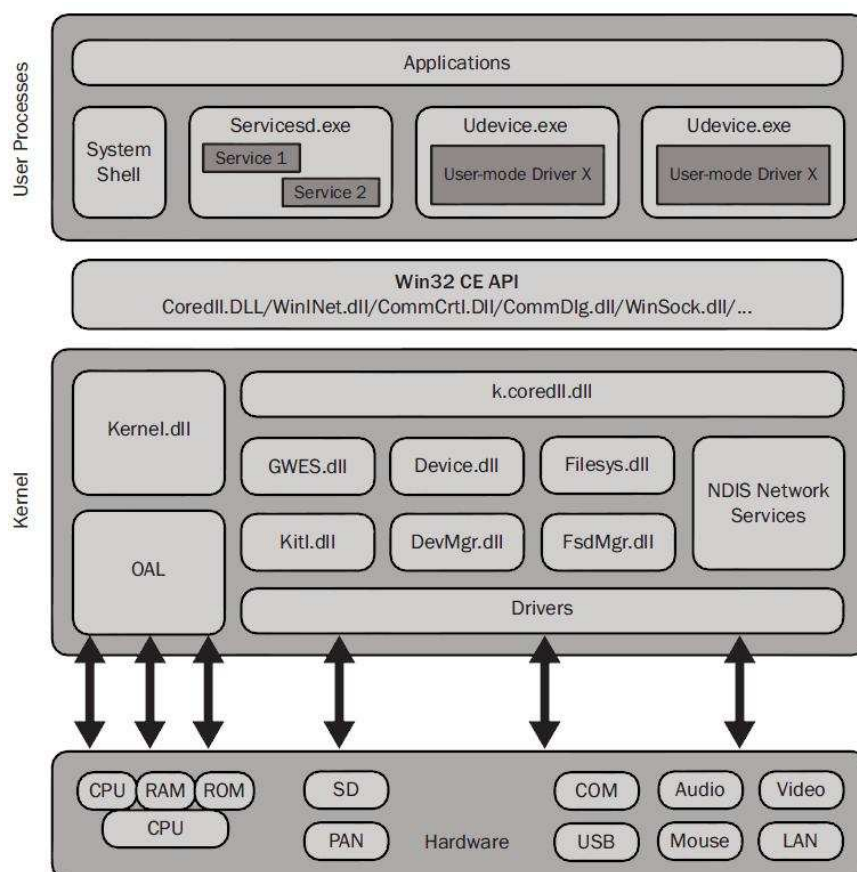


Figura 2.6: Architettura del sistema operativo [10].

2.3.3 Application Development

È possibile costruire applicazioni per Windows Embedded CE utilizzando due tipi di codice: "native code" oppure "managed code". Le applicazioni con "native code" possono essere costruite come sottoprogetti del progetto operativo, o

come singoli progetti e necessitano di un SDK. Le applicazioni con “managed code” possono essere costruite solo come applicazioni separate che non hanno bisogno di un SDK ma richiedono invece l’ambiente di esecuzione del dispositivo. Le principali differenze fra i due tipi di codice sono sintetizzate nella tabella :

Native Code	Managed Code
scritto in C/C++ o ASM	scritto in C#/VB.NET usando il .NET Compact Framework
Compilato in codice macchina su di una workstation di sviluppo.	Compilato in codice di linguaggio intermedio su di una workstation di sviluppo.
E’ necessario ricompilare il codice per supportare una diversa architettura della CPU.	Non è necessario ricompilare il codice per supportare una diversa architettura della CPU.
Non ha bisogno di strutture aggiuntive per girare sul dispositivo.	Ha bisogno di un Common Language Runtime sul dispositivo per girare.
Massimo accesso possibile alle system API e ai servizi.	Accesso ai servizi e alle API supportate da .NET Compact Framework. Usa P/Invoke per accedere alle piattaforme API e COM per interagire con gli oggetti COM.
Pieno supporto degli sviluppi COM e ActiveX	componenti “managed” possono essere esposti come componenti COM con alcune limitazioni.
Possono usare Microsoft Foundation Classes, Active Template Library, Windows Template Library, and Standard Template Library.	Si possono usare le Base Class Libraries e sono disponibili diverse third-party libraries.
Visual Studio 2005 Service Pack 1, Visual Studio 2008, eMbedded Visual C++ 4.0.	:Visual Studio 2005 Service Pack 1 with appropriate .NET Framework 2.0, and Visual Studio 2008 for .NET

Tabella 2.1: differenze fra “native code” e “managed code” [10] .

2.3.4 Developer Workstation Requirements

Per sviluppare delle applicazioni per Windows CE abbiamo bisogno di utilizzare un ambiente di sviluppo (ad esempio eMbedded Visual C++ 4.0) e di compilare il codice su una workstation con determinate caratteristiche [10] :

- Microsoft Windows 2000 Professional con Service Pack 4 oppure Windows XP Professional con Service Pack 2 o superiori.
- Processore minimo da 933 *MHz* (2 *GHz* raccomandato).
- RAM minimum da 512 *MB* (1 *GB* raccomandato).
- 18 *GB* di spazio libero sul disco per l'installazione.
- 1 *GB* di spazio libero sul disco di sistema.
- DVD-ROM drive.

Capitolo 3

Software utilizzato

3.1 TwinCAT

The Windows Control and Automation Technology è un software che permette di convertire ogni tipo di pc compatibile, in particolare PC industriali e PC embedded Beckhoff, in un controllore "real-time" [11]. L'architettura TwinCAT consiste in un sistema "run-time" che esegue programmi di controllo in "real-time" disponendo servizi di programmazione, diagnosi e configurazione, che consentono lo scambio di informazioni con programmi Microsoft per la visualizzazione dei dati o per l'esecuzione stessa dei comandi. Il pacchetto software TwinCAT è formato da vari sottoprogrammi specifici per varie applicazioni:

- TwinCAT PLC : permette la scrittura di programmi multi-PLC su PC in accordo con lo standard IEC 61131-3.
- TwinCat NC PTP : permette il posizionamento degli assi con tecnica "point-to-point".
- TwinCAT NC I : permette il posizionamento degli assi con tecnica di interpolazione in tre dimensioni.
- TwinCAT CNC : offre soluzioni CNC ad alte prestazioni per problemi complessi.
- TwinCAT I/O : connette gli ingressi e le uscite con i programmi windows.

- Altri supplementi per semplificare le funzioni di sistema, comunicazione, movimento, controllo e costruzione di sistema di automazione.

La versione base del programma è formata da applicazioni di configurazione (TwinCAT System Manager, TwinCAT System Control) e dal sottoprogramma TwinCAT I/O, che non necessitano di licenza Beckhoff. Tale licenza è indispensabile per usufruire degli altri software che consentono forniscono applicazioni di controllo più avanzate.

Installazione TwinCat è già installato sull'embedded PC CX1020 di Beckhoff con licenza PLC per il solo dispositivo, che prevede la possibilità di utilizzare sia TwinCAT PLC che TwinCAT I/O ma non consente l'utilizzo delle funzioni di configurazione direttamente da Windows CE. TwinCAT viene allora installato in versione base (senza licenza PLC) su una workstation di supporto, collegata al embedded PC tramite cavo cross o rete Ethernet (standard TCP/IP), da dove è consentita l'accesso alle applicazioni di configurazione e al sotto programma TwinCAT I/O.

Menù TwinCAT In seguito all'installazione del programma compare un'icona nella barra di comando in basso a destra che permette l'accesso alle varie applicazioni di sistema di TwinCAT per mezzo di un menù a finestra pop-up e permette di visualizzare in modo immediato lo stato attuale del sistema in base al colore dell'icona stessa :

- Rossa : TwinCAT stopped.
- Gialla: TwinCAT started.
- Verde : TwinCAT running.
- Blu : TwinCAT Config mode.

Cliccando con il tasto destro del mouse sull'icona è possibile accedere alla finestra del menù che dispone di varie voci che permettono il controllo del sistema come rappresentato in figura 3.1 . I vari comandi sono :

- "About TwinCAT" : da cui si accede ad una descrizione della versione del programma installato.
- "Log View" : programma per monitorare gli eventi nel sistema.
- "System Manager" : permette l'accesso all'applicazione di configurazione di TwinCAT.
- "System" : consente di cambiare istantaneamente lo stato del sistema.
- "IO" : ha funzione di reset per il sotto sistema TwinCAT I/O.
- "Router" fornisce le informazioni sulla memoria interna e sulla porta di comunicazione per le diagnosi.
- "Propieties" : attraverso questo comando si accede a "TwinCAT System Control" che è un applicazione "dialog-based" suddivisa per aree di sistema : general (informazioni sulla versione installata); System (configurazioni di base del sistema); AMS Router (viene indicatao l'indirizzo locale del pc nella rete TwinCAT); PLC (inerente a configurazioni PLC); Registration (per inserire licenze di utilizzo).



Figura 3.1: Menù TwinCAT .

3.2 TwinCAT System Manager

TwinCAT System Manager è un'applicazione fondamentale di TwinCAT per la configurazione del programma. Le informazioni derivanti dai dispositivi collegati al controllore vengono organizzate e collegate ai segnali di ingresso e di uscita dei task del programma di controllo, mappando gli indirizzi delle variabili in un'immagine di processo. Questa applicazione consente la comunicazione "task to task" in modo che "in and outputs" di un task possono essere modificati ciclicamente con "out and inputs" di un altro task conservando in modo ottimale i dati scambiati.

Le variabili possono essere individuali (bits, bytes, 16 bit data words, 32 bit data words etc.) oppure strutturate in arrays di dati. La più piccola unità che può essere indirizzata e collegata è una variabile booleana singola. TwinCAT System Manager produce assegnazioni individuali che contengono i collegamenti creati, in particolare tra task del programma di controllo e dispositivi sul fieldbus e tra vari task se hanno variabili collegate tra loro in comune. L'utilizzo di questa applicazione ci consente di gestire facilmente le variabili del programma di controllo senza conoscere il tipo di struttura e le proprietà di ogni fieldbus usato.

TwinCAT System Manager supporta tutti gli standard commerciali di fieldbus e anche altri standard di interfacce PC : Beckhoff Lightbus; Profibus DP, MC (DP-V2); Interbus; CANopen; SERCOS; DeviceNet; Ethernet; EtherCAT; Beckhoff Real-Time; Porta di stampa Ethernet PC (8 inputs e 8 outputs basati su TTL); Serial Bus Coupler BK8100 to COM; USB Bus Coupler BK9500 e USB Control Panel interface (CPx8xx); Dual-ported memory interface (DPRAM) per cards PC NOVRAM (Non-Volatile RAM) System Management Bus (SMB) per il monitoraggio del hardware PC come ventole, temperatura, e altro [12] .

Per accedere a TwinCAT System Manager basta entrare nel TwinCAT menù e selezionare "System Manager". La finestra TwinCAT System Manager viene mostrata in figura 3.2 . Nella barra principale viene scritto il nome del progetto e il nome del dispositivo collegato, mentre nelle barre sottostanti è presente il menù con le varie toolbar. Nello spazio centrale è presente una struttura ad albero che contiene i principali componenti di configurazione. Di particolare interesse sono :

- “Configurazione SYSTEM”
- “Configurazione I/O”.

Nella barra in fondo alla finestra viene indicato il nome del dispositivo collegato e lo stato attuale di TwinCAT:

- TwinCAT stopped con barra di colore rosso.
- TwinCAT started con barra di colore giallo. Il sotto programma TwinCAT I/O viene parametrizzato secondo la configurazione scelta.
- TwinCAT running con barra di colore verde ed indicando la percentuale di CPU utilizzata. In questa modalità è possibile verificare lo stato in linea delle variabili forzando il valore delle stesse.
- TwinCAT is in Config mode con barra di colore blu. In questa modalità è possibile configurare le variabili e collegarle ai vari dispositivi.

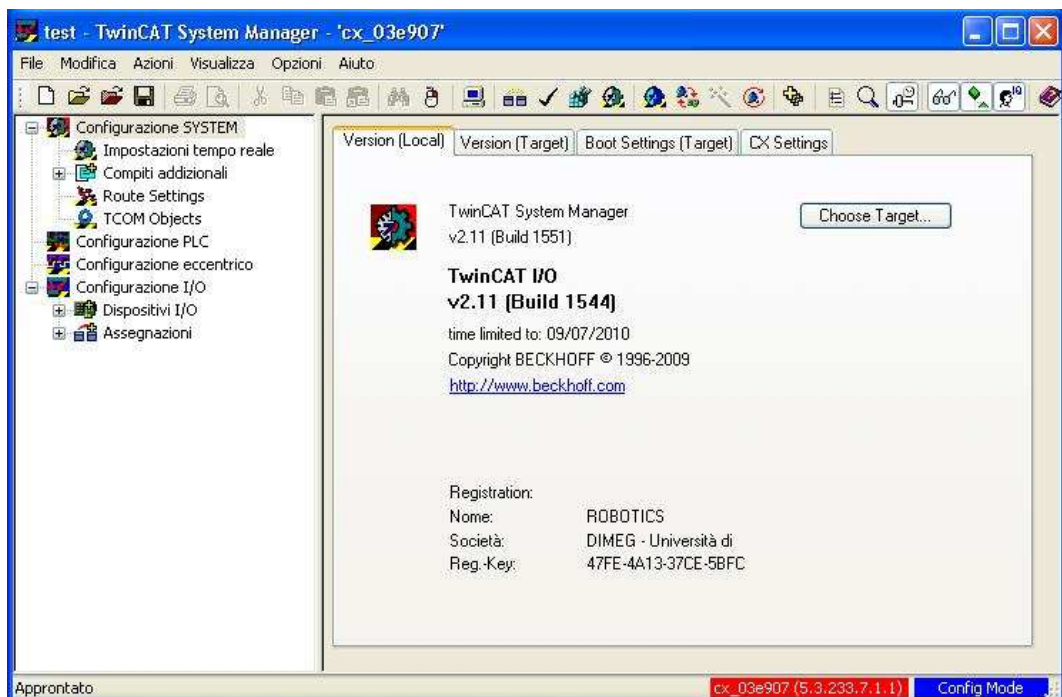


Figura 3.2: Finestra TwinCAT System Manager .

System Configuration Rappresenta l'applicazione di configurazione principale di TwinCAT System Manager la quale consente in primo luogo di creare un collegamento tra il pc remoto e l'embedded pc utilizzato per il controllo. Nel Pannello principale, nel quale compare la versione del programma, premendo il tasto "choose Target..." si accede ad una finestra che permette di ricercare la presenza di reti ethernet o di eventuali fildbus. Cliccando il tasto "Search (Ethernet)" compare la finestra "Add Route Dialog" che permette di ricercare i dispositivi collegati a reti ethernet. Premendo su "Broadcast Search" comparirà una lista nella finestra centrale e per aggiungere il proprio dispositivo alla rete TwinCAT, basterà selezionarlo nell'elenco e premere "Add Route". In questo modo il programma entra in comunicazione con il dispositivo e consente di impostare le configurazioni di sistema che sono suddivise in:

- Impostazioni tempo reale : consente di configurare il "Base Time" e la percentuale limite di utilizzo della CPU da parte del "real-time" di TwinCAT.
- Compiti Addizionali : in questa sezione si possono aggiungere i tasks addizionali che permettono, quando non viene utilizzata la funzione PLC, di creare variabili di ingresso e di uscita per creare delle applicazioni con diversi tipi di programmi (ad esempio embedded Visual studio C++). Le variabili create vengono collegate agli ingressi e alle uscite dei terminali presenti nel dispositivo, cliccando con il tasto destro del mouse e selezionando il comando "modifica collegamento...". Nella finestra "Task" si possono configurare il numero della porta AMS, il nome del compito addizionale, la priorità e il tempo di ciclo. In questo modo si va a configurare l'immagine di processo del compito addizionale creato ed è possibile esportare tale configurazione, cliccando con il tasto destro del mouse sul task creato e selezionando il comando "Export Header File". Il File così ottenuto contiene il numero di porta ADS del task, la dimensione delle immagini di processo degli input e degli output (in bytes) e le variabili dell'immagine di processo che vengono rappresentate mediante una struttura.
- Route Setting : consente le configurazioni dei routers e delle comunicazioni con i dispositivi trovati.

- TCOM Objects : descrive gli oggetti e le operazioni utilizzate.

I/O configuration Rappresenta il livello base di TwinCAT System Manager ed è sempre presente nello schema ad albero perchè non necessita di una licenza specifica (come per altro accade per le configurazioni di sistema). Essa è suddivisa in due diverse sezioni:

- Dispositivi I/O : vengono configurate gli ingressi e le uscite dei dispositivi collegati al embedded PC. Per aggiungere un dispositivo si può procedere con una scansione automatica , cliccando con il tasto destro del mouse e selezionando “cerca dispositivo”. Una volta terminata la scansione, vengono visualizzati dal programma i dispositivi trovati e gli eventuali terminali collegati. Le variabili di ingresso e di uscita dei vari terminali possono essere collegate alle variabili dei compiti addizionali ma necessitano dell’assegnazione ad una “Sync Units” (modulo che definisce un insieme di informazioni di processo che possono essere scambiati in modo sincrono tra il master e uno o più dispositivi slave EtherCAT) che va poi a collegarsi con il tempo di ciclo del compito addizionale.
- Assegnazioni : una volta creata e configurata l’immagine di processo vengono generate le assegnazioni che mappa gli indirizzi delle variabili dei vari dispositivi con le variabili dei tasks addizionali. Per creare le assegnazioni cliccare con il tasto destro su “Assegnazioni” e selezionare il comando “genera assegnazioni”.

3.3 TwinCat I/O

TwinCat I/O è un’applicazione di TwinCat che include un sistema “real-time” per gestire il funzionamento del fieldbus e un’interfaccia DLL per gestire i programmi e le applicazioni. TwinCat I/O si comporta da “real-time” driver per programmi Windows (nel nostro caso che sono in esecuzione con Windows CE), creando le relazioni corrette tra le variabili del programma, i dispositivi presenti nel sistema e i vari canali di connessione [12] . L’interfaccia TwinCat I/O permette invece

di creare applicazioni Windows CE e di accedere al sottosistema TwinCat I/O senza utilizzare la comunicazione tramite ADS. Per fare questo viene messa a disposizione una DLL con funzioni specifiche che consentono di comunicare con i dispositivi attraverso il fieldbus e di modificare le informazioni di processo.

Per i dispositivi Beckhoff basati su Windows CE viene messa a disposizione la funzionalità “TcTimer” che fornisce un timer deterministico che consente di eseguire programmi scritti in con codici di programmazione comuni (ad esempio C++). Tale funzione “TcTimer” di base può essere scalata fino $100 \mu s$ ed è derivata dal livello maggiore di priorità di Windows CE. Essa permette all’applicazione creata di accedere, per mezzo di un puntatore, al “IO-buffer”, di accedere in modo sincrono al fieldbus e di cambiare le informazioni dei vari task mappando le modifiche sul livello fisico degli I/O.

3.4 eMbedded Visual Studio C++ 4.0

Microsoft eMbedded Visual C++ 4.0 è uno programma che fornisce un completo ambiente di sviluppo per la creazione di applicazioni e componenti di sistema per Windows CE. Permette di costruire applicazioni con codice nativo Windows CE e utilizza come linguaggio di programmazione C e C++. Garantisce un massimo accesso alle API (Application Programming Interfaces) ed in particolare alle Win32 API e alle MFC (Microsoft Foundation Class Library). EMbedded Visual Studio C++ 4.0 mette a disposizione diversi tipi di Template Libraries in particolare : ATL (Active Template Library), WTL (Windows Template Library) e Standard Template Library (STL) [13] .

Dispone di diverse applicazioni che permettono il controllo remoto del dispositivo sia ereditate dalla versione precedente 3.0 (Remote Spy++, Remote Registry Editor, Remote Heap Walker, Remote Process Viewer, Remote Zoomin, Remote File Viewer) che nuove applicazioni come: Remote Kernel Tracker che permette il controllo del kernel; Remote Performance Monitor che permette la grafica display con monitoraggio continuo delle tracce (utilizzo CPU , threads, processi, memoria, traffico network, batteria, e altri valori); Remote System Information che mostra lo stato del sistema (versione OS version, CPU info, codice OEM, e altri

valori) [14]. Oltre all'ambiente di sviluppo viene fornito anche un emulatore del sistema operativo Microsoft Windows CE che rispecchia fedelmente l'immagine dell'OS del dispositivo.

Installazione L'ambiente di sviluppo viene installato su una Workstation di supporto sulla quale viene compilato il codice per creare le applicazioni per Windows CE.

I requisiti minimi di sistema della workstation sono:

- Sistema operativo supportato: Microsoft Windows® 2000 Professional SP2, Microsoft Windows 2000 Server SP2, or Microsoft Windows XP Professional.
- Processore Pentium-II r, 450 MHz o più veloce.
- 96 MB (128 MB raccomandati) di memoria per Windows 2000 Professional o Windows XP Professional. 192 MB (256 MB raccomandati) di memoria per Windows 2000 Server.
- 200 MB di spazio sul hard disk .
- CD-ROM drive.
- Risoluzione del monitor VGA o miglior.
- Mouse o dispositivo di puntamento compatibile.

Prima di installare l'ambiente di sviluppo bisogna modificare la stringa di comando di Start-up di Windows XP (procedura necessaria per workstation con più di due Gb di memoria). Per far questo basta cliccare con il tasto destro sull'icona delle "Risorse del Computer" e selezionare "Proprietà". Nella finestra che compare, bisogna aprire la sezione "Avanzate" e cliccare sul tasto "Impostazioni" presente nella sottosezione "Avvio e Ripristino". Si aprirà la finestra delle impostazioni di "Avvio e Ripristino" nella quale basterà modificare parte della stringa di "Sistema operativo predefinito" sostituendo la scritta `"/No execute = optin/` con `"/execute/`". Si può procedere quindi con l'installazione dell'ambiente di sviluppo e degli aggiornamenti necessari:

- eMbedded Visual C++ 4.0. Contiene l'ambiente di sviluppo e un emulatore del sistema operativo Windows CE. Supporta le piattaforme Windows CE (.NET, 4.0, 4.1, 4.2), Pocket PC 2003 e Smartphone 2003.
- eMbedded Visual C++ 4.0 Service Pack 4 (SP4) . Consente agli utenti di eVC4 di supportare la nuova piattaforma Windows CE 5.0 e Windows CE 6.0. SP4 ha carattere cumulativo e include il supporto per le piattaforme basate su Windows CE 4.0, 4.1 e 4.2. Questo programma deve essere installato successivamente a eMbedded Visual C++ 4.0 .
- Windows CE 5.0: Standard Software Development Kit (SDK). Esso contiene un sottoinsieme di comuni (OS) API che consentono ad un'applicazione scritta con standard SDK di girare su differenti piattaforme Windows CE che hanno una interfaccia utente grafica. Consente quindi ad un'applicazione per Windows CE, che si basa sullo Standard SDK, di funzionare su qualsiasi dispositivo che contiene i componenti SDK Standard.

Capitolo 4

Controllo di un Asse

Il controllo di un azionamento viene effettuato dal Embedded PC CX1020 di Beckhoff tramite EtherCAT e richiede di configurare diversi parametri e di generare alcuni file prima di essere operativo. Presenterò quindi come utilizzare il software di configurazione TwinCAT e l'ambiente di sviluppo Embedded Visual Studio C++ per predisporre il CX1020 al controllo di un dispositivo connesso al bus EtherCAT . L'azionamento in questione è il driver Gold Solo Guitar 6/200 prodotto dalla Elmo e presentato in 4.2.

4.1 Sistema di controllo

Il controllo dell'azionamento viene effettuato da un insieme di anelli di regolazione nidificati che sono : l'anello di corrente, di velocità e di posizione. Le funzioni di regolazione di questi anelli di controllo sono svolte dal driver mentre il controllore fornisce il riferimento. Lo scambio di informazioni tra master e slave viene effettuato tramite bus EtherCAT, garantendo così il determinismo necessario per controllare il sistema real-time, contenendo le spese di cablaggio e mettendo a disposizione funzioni di diagnostica. Il motore e il sensore, in questo tipo di configurazione, sono connessi direttamente al driver e non alla rete EtherCAT. Il nostro sistema è quindi formato da :

- Beckhoff CX1020 : controllore PC-based (vedi capitolo 2.2) sul quale viene fatta girare l'applicazione di controllo che fornisce il riferimento tramite

fieldbus EtherCAT.

- Gold Solo Guitar di Elmo : driver con funzione di regolazione per gli anelli di controllo che fornisce l'alimentazione al motore.
- Motore AC APM-SC05 : motore a corrente alternata prodotto da Mecapion (le caratteristiche sono riportate in appendice A), fornito di encoder incrementale, sensori di Hall e freno inseriti all'interno dell'azionamento stesso.
- Workstation : computer remoto collegato tramite Ethernet (comunicazione TCP/IP) all'embedded PC CX1020. Sulla Workstation vengono installati i software di configurazione (EAS e TwinCAT descritti rispettivamente nei capitoli 4.2.2 e 3.1) e l'ambiente di sviluppo (Visual Embedded C++ presentato nel capitolo 3.4).

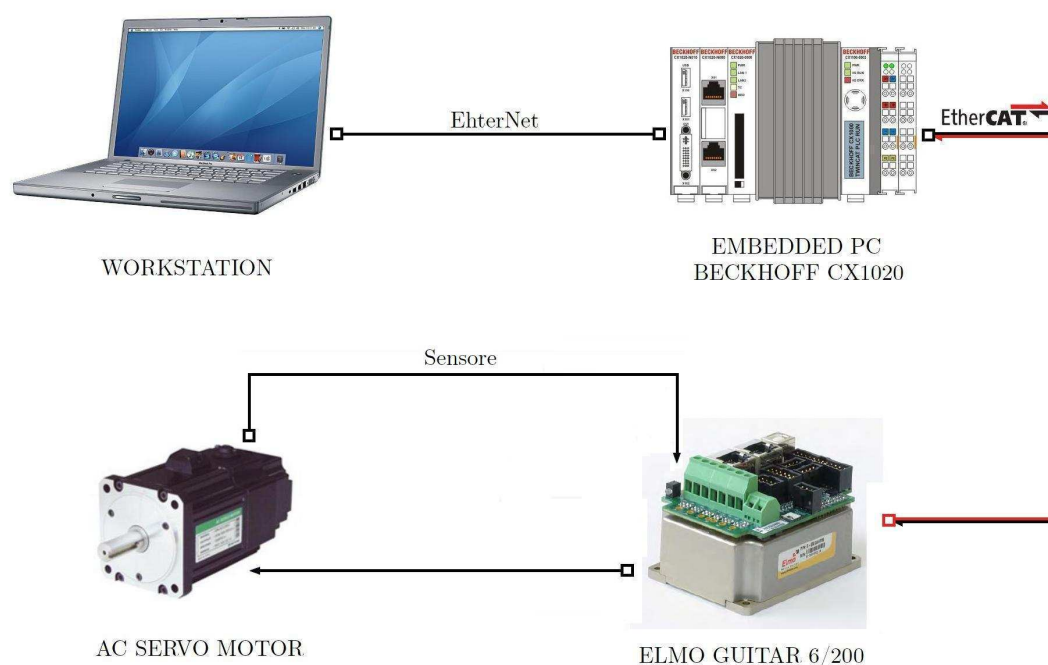


Figura 4.1: Sistema ad anello con fieldbus EtherCAT

4.2 Driver Elmo

Il Gold Solo Guitar 6/200 è un driver prodotto da Elmo che fa parte dei nuovi azionamenti intelligenti della linea Elmo Gold, che offre una soluzione eccellente ad alte prestazioni, un networking avanzato e alta densità di potenza, in combinazione con un controller di movimento completo e con un'intelligenza locale [15]. Il Gold Solo Guitar è una soluzione integrata progettata per collegare in modo semplice ed efficiente unità driver direttamente all'applicazione. La soluzione consiste in una comoda interfaccia di connessione che elimina o riduce i tempi di sviluppo e di progettazione di una scheda PCB (Printed Circuit Board) per l'applicazione.

Il driver dispone di varie opzioni di comunicazione (EtherCAT, CANopen, USB), che rendono il servocomando più veloce e più flessibile, rendendolo in grado di soddisfare qualsiasi applicazione che richieda alte prestazioni nel movimento ad alta precisione. L'azionamento può essere facilmente installato all'interno del motore, vicino alla motore, o in qualsiasi punto all'interno della macchina, garantendo un minimo ingombro ed un'alta resistenza alle condizioni ambientali di utilizzo.

La combinazione di queste caratteristiche unite ad una flessibilità di programmazione, ad una grande varietà di feedback per il funzionamento a ciclo chiuso e all'utilizzo dei bus di comunicazioni EtherCAT e CANopen, rende questo azionamento un'ideale soluzione a basso costo. Il Gold Guitar Solo è facilmente impostabile e regolato utilizzando il software Elmo Application Studio (EAS).

Configurazione Il Gold Solo Guitar è disponibile in varie configurazioni, che dipendono dalle varie opzioni di tensione e di potenza necessarie per l'applicazione, dalle diverse opzioni di comunicazione utilizzate e dalla varietà di feedback e I/O richiesti. Il Driver a nostra disposizione offre fino a 4,9 kW di potenza continua in un dispositivo di 258,6 cc ($80 \times 53 \times 61$ mm), fornendo una tensione massima DC di impiego di 200 Volt e una corrente continua di 6 A. Presenta diverse interfacce di comunicazione di rete : CANopen, USB, Ethernet, EtherCAT e RS-232. Dispone di due porte feedback input che sono flessibili e configurabili. La porta A supporta sensori come : encoder incrementali, Hall digitali o encoder as-

soluti, mentre la porta B supporta : encoder incrementali, encoder analogici, Hall analogici o resolver. Un interfaccio I/O è presente e gestisce segnali sia analogici che digitali.

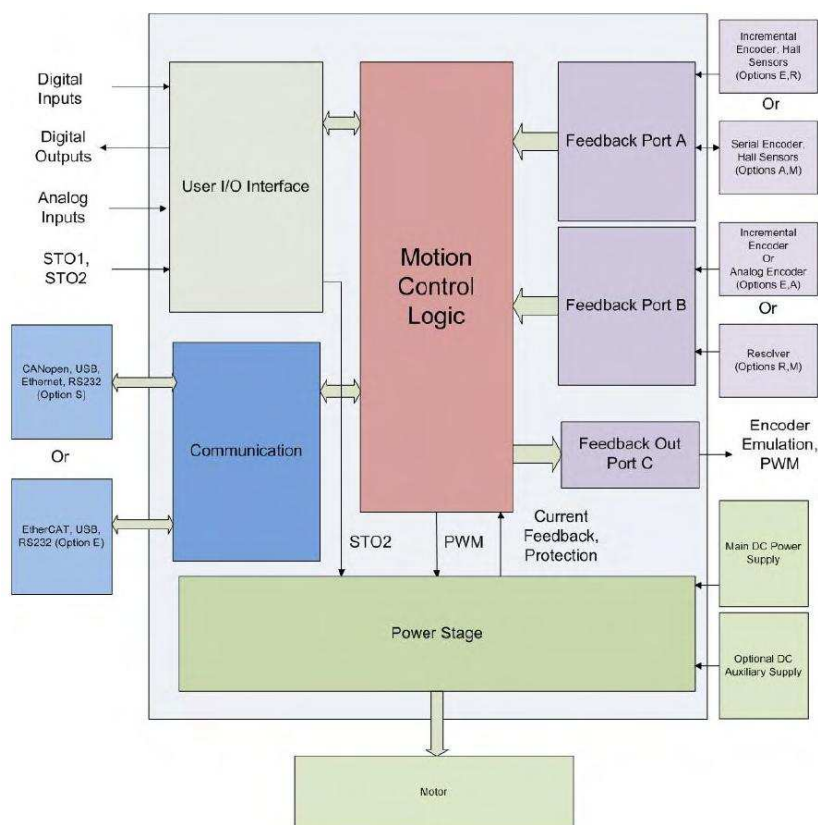


Figura 4.2: Diagramma a blocchi del sistema Gold Solo Guitar [15].

4.2.1 Connessioni

Il driver Gold Solo Guitar è fornito di una comoda interfaccia che facilita le connessioni direttamente con l'applicazione. Esso è equipaggiato con i seguenti tipi di connettori :

- Porta J12 : Alimentazione di potenza ausiliaria.
- Porta J10 : Alimentazione principale.
- Porta J11 : Alimentazione motore.
- Porta J4 : Feedback Porta A

- Porta J5 : Feedback Porta B
- Porta J6 : Input/Output e Feedback Porta C
- Porta J26 : STO (Safety Torque Off)
- Porta J7 : EtherCAT in (RJ-45)
- Porta J8 : EtherCAT out (RJ-45)
- Porta J9 : USB
- Porta J21 : RS232
- Porta J22 : CAN in
- Porta J23 : CAN out

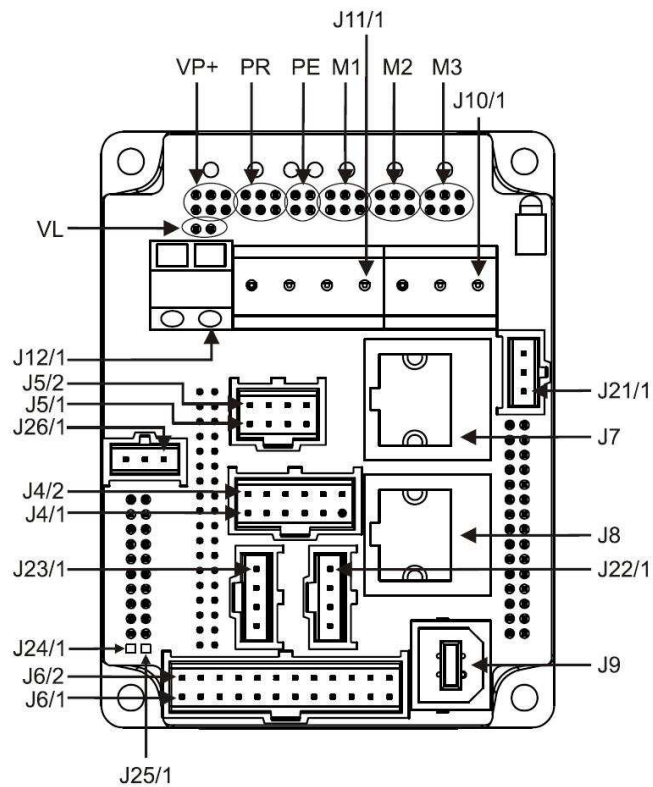


Figura 4.3: Tipi di connettori [15].

Alimentazione Motore La connessione tra driver e alimentazione del motore avviene collegando i morsetti M1, M2, M3 e PE del Gold Guitar Solo rispettivamente alle fasi del servo motore AC e al cavo di terra. La connessione delle fasi è arbitraria in quanto Elmo Application Studio (EAS) stabilirà automaticamente la commutazione corretta durante l'installazione.

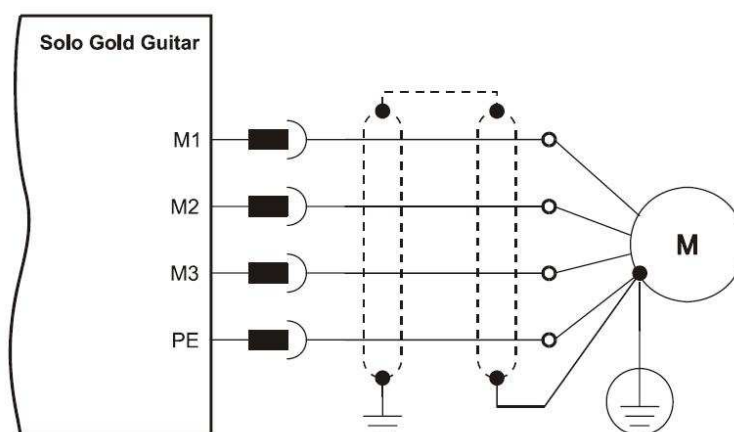


Figura 4.4: Connessione tra il driver e l'alimentazione del motore AC [15].

Alimentazione principale L'alimentazione principale per il Gold Solo Guitar è fornita da una sorgente di $14 - 195\text{ VDC}$. Un algoritmo di controllo gestisce l'alimentazione del Gold Guitar Solo in modo che funzioni solamente con l'alimentazione elettrica principale, senza bisogno di un'alimentazione ausiliaria di 24 Volt . Ci sono tensioni multiple per il Gold Guitar Solo (da 14 V a 195 V), pertanto è necessario utilizzare la tensione corretta in base alla tensione massima di funzionamento del driver.

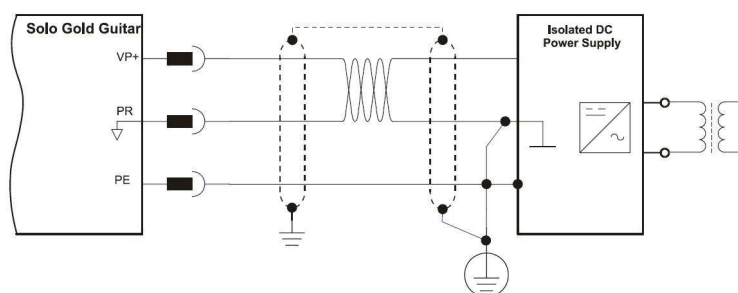


Figura 4.5: Diagramma connessione tra il driver e l'alimentazione principale [15].

Feedback motore Il driver a nostra disposizione è fornito di due porte configurabili per i segnali in ingresso dai sensori di movimento: Porta A (J4) e porta B (J5). Il software di configurazione designa ad ogni ingresso un ruolo, ad esempio, l'encoder incrementale sulla porta B è il feedback di posizione del motore, i sensori di Hall sulla porta A sono il feedback di commutazione. Nella nostra applicazione viene utilizzato un encoder incrementale e un sensore di Hall che vengono connessi entrambi alla porta A, come rappresentato nelle figure 4.6 e 4.7.

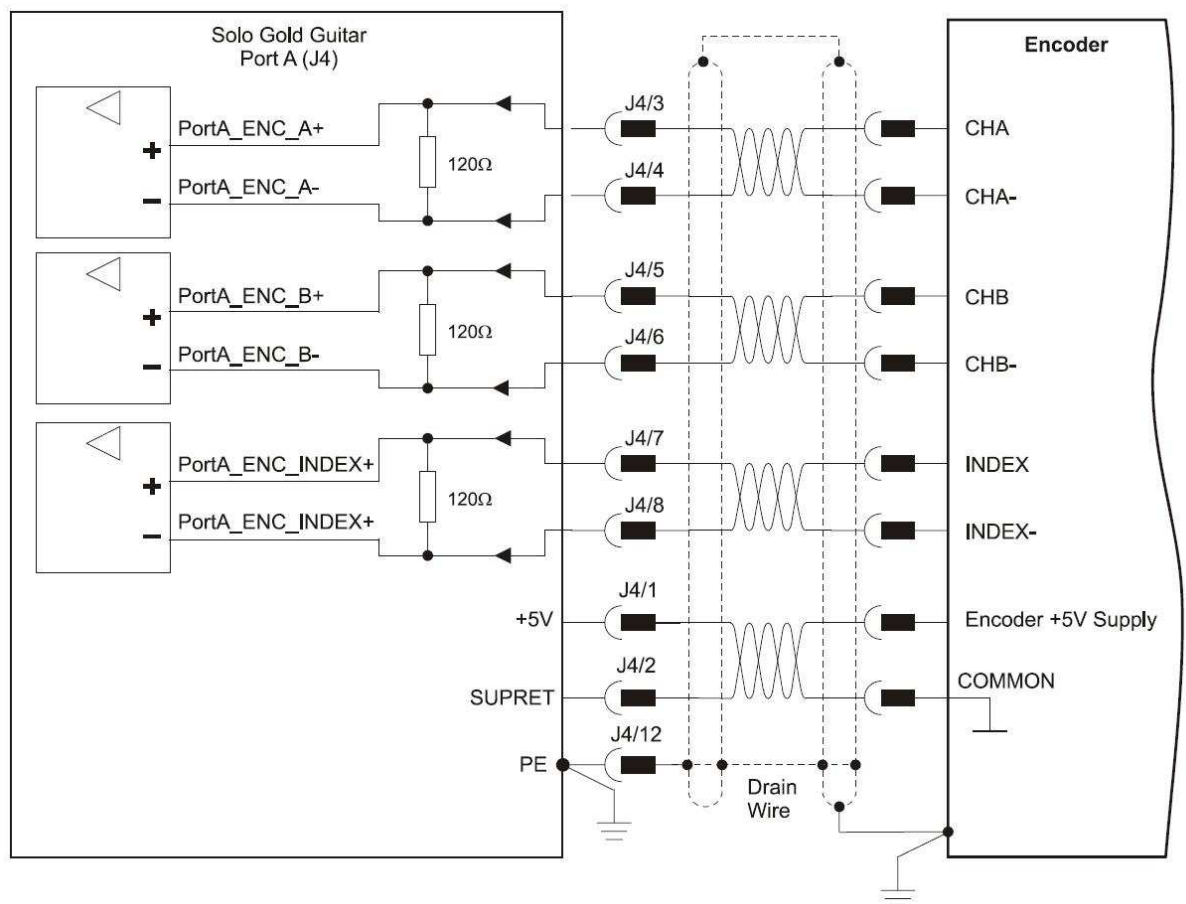


Figura 4.6: Porta A : Input dell'encoder incrementale [15].

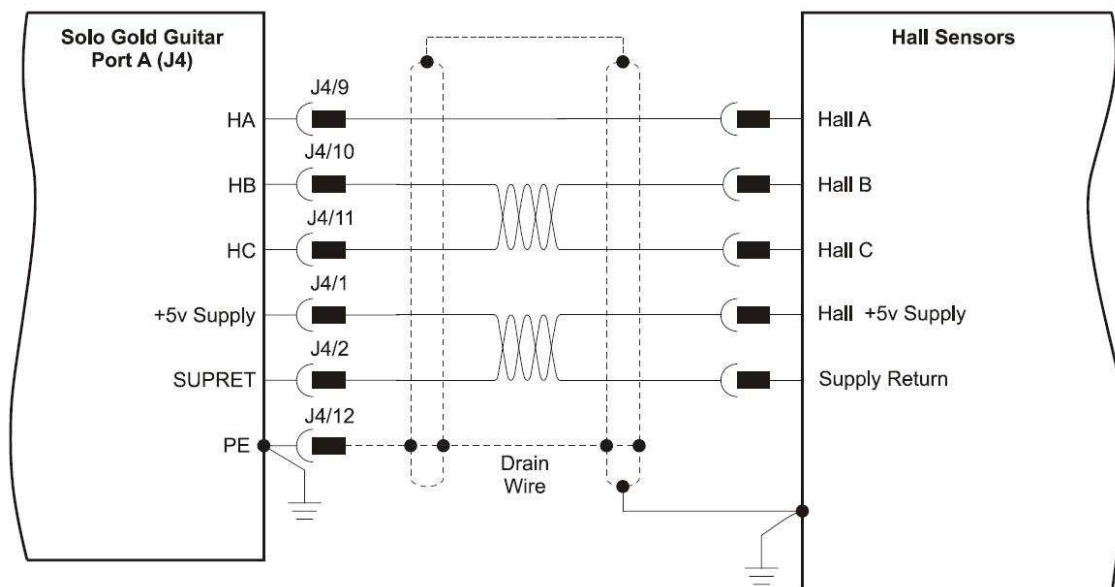


Figura 4.7: Diagramma di connessione sensori di Hall [15].

Input e output Il Gold Solo Guitar dispone di sei ingressi digitali programmabili, quattro uscite digitali e un ingresso analogico posizionati nella porta (J6). Nella nostra applicazione viene utilizzato un input digitale ed un output digitale per comandare il freno per il servo motore. Le connessioni utilizzate sono riportate in figura 4.8.

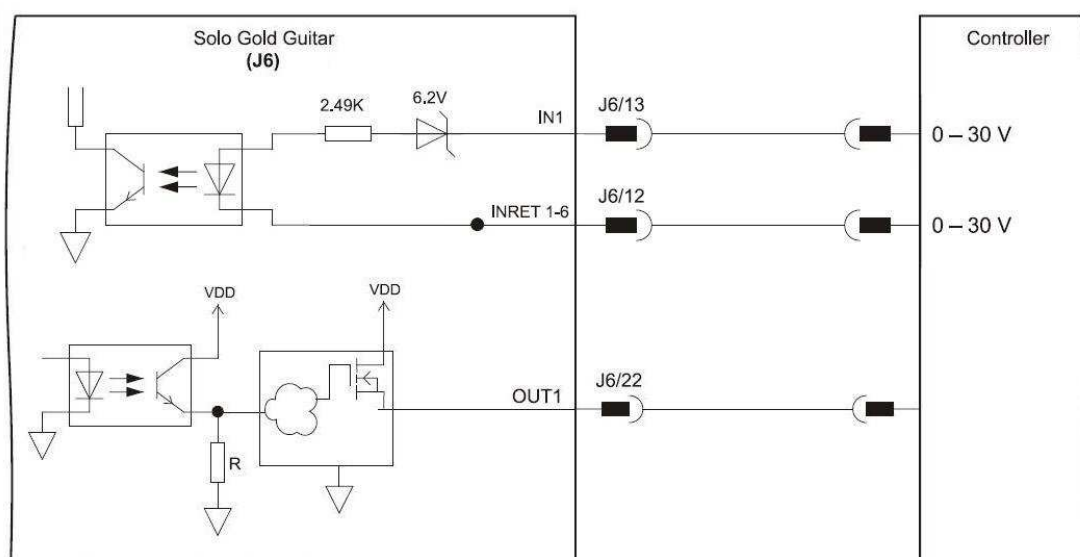


Figura 4.8: Diagramma di connessione Input/Output digitali [15].

EtherCAT Per la comunicazione con il controllore viene utilizzata la porta EtherCAT in (J7) e viene usato un cavo CAT5, con la configurazione delle connessioni come riportato in tabella 4.1 .

Pin	Signal	Function
J7/1	EtherCAT_IN_TX+	EtherCAT in transmit +
J7/2	EtherCAT_IN_TX-	EtherCAT in transmit -
J7/3	EtherCAT_IN_RX+	EtherCAT in receive +
J7/4	N/A	
J7/5	N/A	
J7/6	EtherCAT_IN_RX-	EtherCAT in receive -
J7/8	N/A	

Tabella 4.1: Connessioni cavo EtherCAT [15].

Quando si connettono vari dispositivi EtherCAT in una rete, il master EtherCAT deve sempre essere il primo dispositivo della rete. L'uscita di ogni dispositivo è collegato all'ingresso del dispositivo successivo. L'uscita dell'ultimo dispositivo può rimanere scollegata, mentre se è necessaria ridondanza, l'uscita dell'ultimo dispositivo deve essere collegata all'ingresso del master EtherCAT.

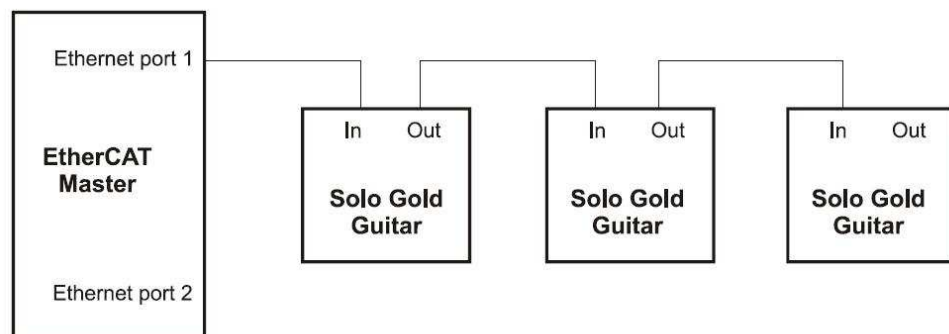


Figura 4.9: Rete EtherCAT senza Ridondanza [15].

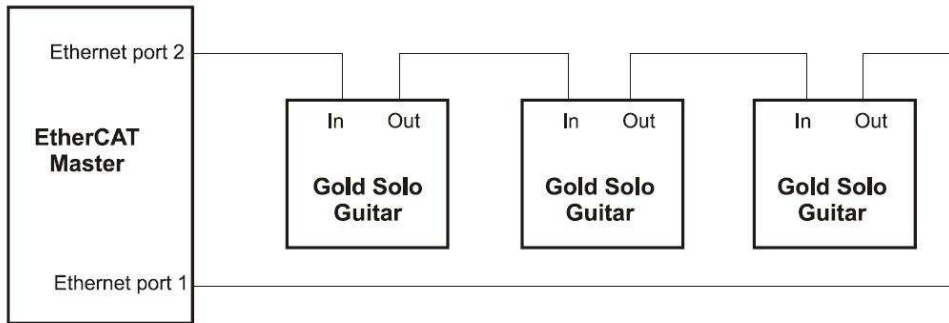


Figura 4.10: Rete EtherCAT con Ridondanza [15].

4.2.2 Elmo Application Studio (EAS)

L'Elmo Application Studio (EAS) è un'applicazione user-friendly, progettata per soddisfare la necessità di una semplice installazione, per la configurazione e il funzionamento del sistema di movimento e dei suoi componenti, ognuno dei quali avente differenti configurazioni delle modalità e dei parametri di funzionamento. Il software è progettato per utenti di diverso livelli : veloce, esperto e competente, fornendo un compromesso tra semplicità (Veloce) e ottimizzazione (Esperto).

EAS viene installato sulla workstation e fornisce un'interfaccia utente unificata e intuitiva per tutti i compiti che vengono effettuati durante lo sviluppo del sistema. Questo include la configurazione del sistema, l'installazione e la taratura dei driver, il comando e la programmazione multi-asse (se si utilizza un controllore multi-asse Elmo come il Gold Line Mastro) e il trattamento di tutti i file inerenti all'applicazione.

Nel nostro caso viene usato EAS per installare il driver Gold Solo Guitar nel nostro sistema di controllo. Durante l'installazione del driver viene effettuata e verificata la connessione delle fasi con i morsetti (M1, M2, M3) stabilendo la commutazione corretta. Il software di configurazione viene poi utilizzato per designare ad ogni porta (ingresso, uscita o feedback) un ruolo. Vengono quindi associati i vari compiti della nostra applicazione con le connessioni effettuate. In particolare l'encoder incrementale e i sensori di Hall, presenti all'interno del motore e connessi alla porta A, vengono associati al feedback di posizione del motore.

L'Elmo Application Studio viene anche utilizzato per scegliere il modo di utilizzo del driver. Il Gold Solo Guitar dispone di tre differenti modalità di funzionamento che sono : controllo di corrente/coppia, controllo di velocità e controllo di posizione.

Il design dell'interfaccia consente anche l'accesso a tutti i parametri di controllo e agli stati pertinenti, così come la registrazione dei dati e altri strumenti di diagnostica. EAS è fornito con template per applicazioni comuni, assicurando un rapido tempo di sviluppo e un'attuazione ottimale anche per le applicazioni più complesse.

4.3 Costruzione applicazione

L'applicazione di controllo viene costruita e compilata nella Workstation mediante l'utilizzo dell'ambiente di sviluppo Embedded Visual Studio C++ e necessita di alcune configurazioni ed impostazioni preliminari.

4.3.1 Configurazione dispositivo

Una volta avvenuta l'installazione e la configurazione del driver mediante EAS, il dispositivo deve essere connesso alla rete EtherCAT e riconosciuto dal controllore. Per fare questo viene quindi utilizzato il software di Beckhoff TwinCAT System Manager come descritto nel capitolo 3.2. Dopo la scansione automatica dei dispositivi viene riconosciuto il driver Elmo ed aggiunto tra i "Dispositivi I/O" in un "Box" inerente al sottoalbero denominato "Disositivo 3 (EtherCAT v2.10 only)".

Nel menù a finestre corrispondente (come riportato in figura 4.11) vengono riportate le informazioni generali del driver e le configurazioni della rete EtherCAT, come ad esempio il numero del nodo e l'indirizzo del dispositivo all'interno della rete. Di particolare importanza è la finestra relativa ai "Process Data" che contiene informazioni ed indirizzi relativi ai parametri e ai valori scambiati tra il driver e il controllore. Le variabili di input e output del dispositivo slave vengono visualizzate come CANopen Process Data Object (PDO).

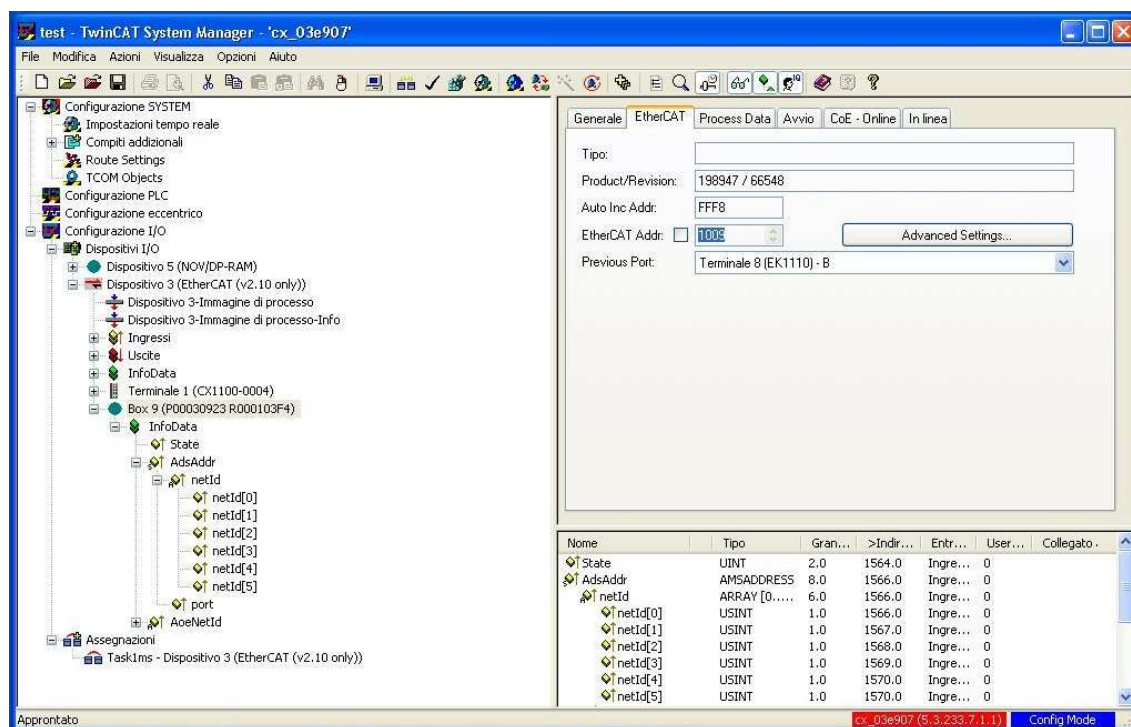


Figura 4.11: Dispositivo aggiunto alla rete EtherCAT.

4.3.2 Task addizionali

Per garantire all'applicazione di controllo l'accesso diretto alle variabili è necessario costituire un task addizionale, utilizzando il software di Beckhoff TwinCAT System Manager come descritto nel capitolo 3.2. Una volta creato un Additional Task si possono aggiungere le variabili che servono per l'applicazione di controllo. Tali variabili vengono successivamente collegate agli ingressi e alle uscite del dispositivo Gold Solo Guitar, inserito nella rete EtherCAT precedentemente. Ogni variabile del task addizionale necessita di una "Sync unit", che definisce un insieme di dati di processo, scambiati in modo sincrono tra il controllore e il driver. Nella nostra applicazione ogni oggetto PDO è assegnato alla stessa area di processo (Sync Unit) in modo tale che tutte le variabili siano scambiate e modificate con la stessa sincronizzazione. In questo modo si va a configurare l'immagine di processo del compito addizionale creato ed è possibile esportare tale configurazione in un file Header.

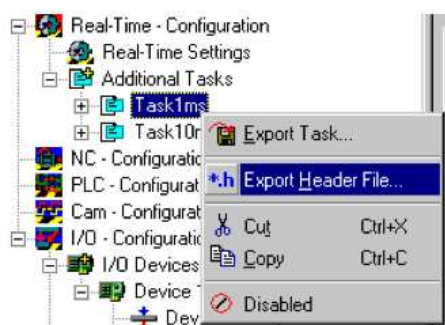


Figura 4.12: Export Header File

Il File così ottenuto contiene il numero di porta ADS del task, la dimensione delle immagini di processo degli input e degli output (in bytes) e le variabili dell'immagine di processo che vengono rappresentate mediante una struttura. La scelta dei parametri da collegare al task aggiuntivo viene effettuata in base al modo d'uso del driver e quindi a seconda che il controllo sia fatto in corrente, velocità o posizione. Il tempo di ciclo e quindi la velocità con cui avviene il controllo può essere configurata all'interno della finestra "Task" (come rappresentato in figura 4.13).

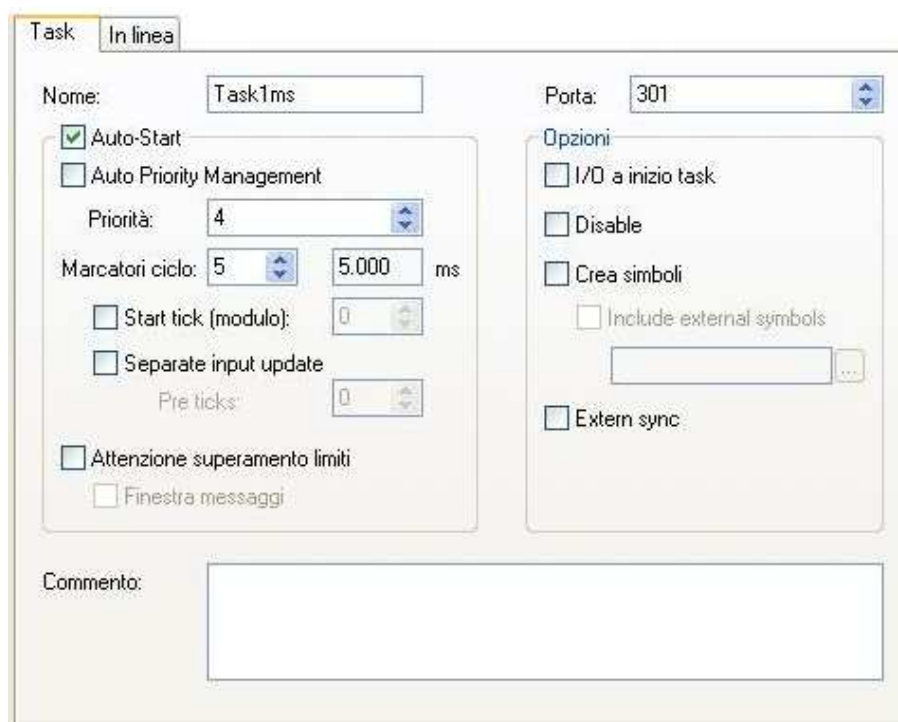


Figura 4.13: Finestra "Task" riguardante i gli Additional Task.

La configurazione tra variabili del task addizionale e ingressi ed uscite del dispositivo va quindi attivata cliccando sul tasto “apply configuration” presente nel tool bar di TwinCAT System Manager. Tale comando crea la mappatura delle variabili con i corrispondenti fieldbus I/O e genera le assegnazioni del compito addizionale. Come ultima cosa basterà riavviare il programma nella modalità “Run Mode”, premendo il tasto “start/restart” dopo aver salvato.

4.3.3 Implementazione

L’applicazione viene sviluppata utilizzando l’ambiente di sviluppo Embedded Visual C++ descritto nel capitolo 3.4. Una volta attivata la configurazione attraverso TwinCAT System Manager l’accesso da parte della nostra applicazione alle variabili e al sottoprogramma TwinCAT I/O viene garantito da una serie di DLL messe a disposizione da Beckhoff. Queste librerie offrono quindi all’utente una serie di funzioni e di interfacce per attivare il bus di campo e per scambiare i dati di processo I/O con il sottoprogramma TwinCAT I/O.

Avendo a disposizione il controllore CX1020 di Beckhoff, che presenta come RTOS Windows CE 6.0, è necessario includere le giuste librerie all’interno del nostro programma di controllo :

- TwinCAT I/O File : TCatIoW32Api.h e TCatIoDrvW32.lib, che servono per attivare la comunicazione sul bus.
- TwinCAT Timer file : TcTimerAPI.h eTcTimerWrap.lib , che fornisce un timer deterministico che consente di eseguire ad esempio codice C++ anziché implementare la logica in linguaggio PLC. Il TcTimer è scalabile fino a 100 μ s e deriva dal livello più alto di priorità di Windows CE, offrendo un accesso diretto tramite puntatori agli I/O del task addizionale di TwinCAT.
- TwinCAT e impostazioni Windows CE : bisogna attivare l’opzione ”Run as Device”, utilizzando ”CxConfig Tool”, messo a disposizione dal OS.

Oltre alle DLL per la comunicazione EtherCAT nel nostro programma viene incluso pure il file Header contenente le variabili del task addizionale collegate al nostro dispositivo che vengono poi utilizzate per il controllo.

La funzione principale alla nostra applicazione è chiamata `TimerProc1()`. All'interno di essa infatti viene inserito il codice necessario per effettuare il controllo del riferimento. Tale funzione svolge i calcoli necessari al controllo e attende il tempo necessario per lo scambio dei dati con il bus, come rappresentato nel codice riportato di seguito.

```
static DWORD WINAPI TimerProc1( LPVOID lpParam)
{
    static int i=0;
    long nError;
    while(!g_bExit)
    {
        WaitForSingleObject(g_hEvent1,INFINITE);
        // first try to get the inputs and test if input update succeeded
        if ( (nError = TCatIoInputUpdate( TASK1MS_PORTNUMBER )) == 0 )
        {
            // do your calculation and logic
            TCatIoOutputUpdate( TASK1MS_PORTNUMBER );
        }
        else
            printf( "TCatInputUpdate(%d) %d failed with 0x%x !\n",
                    TASK1MS_PORTNUMBER, i++, nError );
    }
    return 0;
}
```

Il programma presenta altre funzioni necessarie alla gestione della comunicazione, come ad esempio, quelle che calcolano il tempo di attesa e altre che creano e gestiscono eventi di trigger, sempre richiamando ed utilizzando le librerie TwinCAT incluse in precedenza.

L'applicazione viene infine compilata sulla workstation e successivamente caricata nel controllore, all'interno della cartella "ftp" presente in "Hard Disk". Per eseguire l'applicazione basta accedere alla shell di Windows CE ed eseguire il programma da linea di comando. Per mancanza del programma di configurazione del driver non è stato possibile testare l'applicazione di controllo del motore elettrico tramite bus EtherCAT, tuttavia il lavoro di questa tesi, in aggiunta al codice del programma e alle configurazioni di TwinCAT sono state predisposte per un loro utilizzo e per sviluppi futuri.

4.3.4 Test di verifica I/O

Per verificare le prestazioni del bus di campo EtherCAT e lo sviluppo di un'applicazione con l'ambiente di sviluppo embedded visual c++ è stato fatto un test di verifica dei terminali di ingresso e di uscita presenti nel CX1020 di Beckhoff.

Per questo tipo di applicazione sono stati scelti due dei moduli presenti nel dispositivo, in particolare i terminali:

- EL1202 : Terminale di ingresso digitale a due canali, con tensione di ingresso a 24 V olt DC, Ton / Toff 1 microsecondo
- EL2202 : Terminale di uscita digitale a due canali, con tensione di uscita a 24 V olt DC, Ton / Toff 1 microsecondo e con "push-pull" (consente di collegare l'uscita attivamente a 24 V olt , 0 V olt o ad alta resistenza)

Viene utilizzato un canale di uscita e uno di entrata collegandoli rispettivamente in modo che il terminale EL2202 di output digitale funzioni come alimentatore per il EL1202 di input digitale. La nostra applicazione ha quindi come scopo quello di attivare il modulo di uscita inviando un segnale che giungerà direttamente, tramite il collegamento effettuato, al modulo di ingresso. Una volta ricevuto il segnale in input il controllore farà apparire sul terminale di comando un messaggio di segnalazione.

Il primo passo è quello di configurare il controllore, procedendo con la ricerca di dispositivi tramite TwinCAT System Manager come visto in precedenza. Tra i dispositivi trovati, nella sezione riservata a EtherCAT, comparirà il Terminale1 (CX1100-0004) che rappresenta l'insieme dei terminali collegati al segmento EtherCAT presente nell'unità di alimentazione. Selezionando tale sezione compariranno i tutti i terminali EtherCAT collegati con essa, tra cui i due citati in precedenza, inerenti alla nostra applicazione.

A questo punto verrà creato un Task addizionale (vedi capitolo 4.3.2) e verranno aggiunte una variabile di ingresso, rinominata EL1202_in e una di uscita, rinominata a sua volta EL2202_out, entrambe della dimensione di un bit. Una volta create vengono successivamente collegate agli I/O dei terminali scelti e rispettivamente : EL1202_in viene collegata all'ingresso del primo canale del

terminale EL1202, mentre EL2202_out viene collegata al all'uscita del primo canale del terminale EL1202. Viene poi aggiunta al Task addizionale una nuova variabile relativa allo stato del dispositivo, rinominata WcState e collegata alla variabile corrispondente. Una volta terminati i collegamenti vengono generate le assegnazioni, esportato il file header del compito addizionale e viene attivata la configurazione attivando il controllore nella modalità "run mode".

Viene quindi implementato il codice del programma in modo analogo a come fatto nella sezione precedente. Vengono incluse le librerie per la comunicazione con il bus e il file header contenente le variabili da comandare. La funzione principale anche di questa applicazione è TimerProc1(). All'interno di essa viene inserito il codice necessario per attivare il primo canale del terminale di uscita digitale EL2202. Viene poi aggiunta una parte di codice che verifica se il segnale giunge in ingresso e che stampa sul terminale dell'OS il messaggio di segnalazione.

```
if ( (nError = TCatIoInputUpdate( TASK1MS_PORTNUMBER )) == 0 )
{
    if ( pT1msIn->WcState != 0 )
        printf( "I/O Device Error !\n" );
    else
    {
        pT1msOut->out_EL2202 = 1;
        if(pT1msIn->in_EL1202 == 1){
            printf("uscita attiva \n");
        }
        TCatIoOutputUpdate( TASK1MS_PORTNUMBER );
    }
}
```

Una volta completata l'implementazione il codice viene compilato e il file .exe generato viene inserito nella cartella "Hard Disk/ftp" del controllore e fatto eseguire da terminale.

Per verificare le prestazioni del bus viene ripetuto il test variando tramite software di configurazione TwinCAT System il tempo di ciclo del Task Addizionale. Vista la bassa complessità di calcolo e la ridotta dimensione delle variabili, entrambe della dimensione minima di un bit, si riesce a raggiungere la velocità massima di ciclo permessa dal sistema di 50 μs . Si riesce così a sfruttare l'intera banda messa a disposizione da EtherCAT e a testare il sistema dei terminali inseriti e comunicanti tramite bus.

Conclusioni

In questo lavoro di tesi è stata presentata come la base di campo EtherCAT risulti un'efficace soluzione per la comunicazione nell'automazione, capace di abbattere il problema del cablaggio, mantenendo comunque un livello di prestazione e di determinismo ottimale. L'implementazione di un'applicazione di controllo è stata supportata dall'utilizzo delle librerie per la comunicazione tramite il bus EtherCAT e dall'impiego del timer deterministico messo a disposizione per il RTOS installato sull'embedded PC.

Per mancanza del programma di configurazione del driver non è stato possibile eseguire il test di controllo del motore, tuttavia si è potuto verificare il lavoro svolto in questa tesi, testando il comando dei terminali I/O e predisponendo il sistema per utilizzi futuri.

L'uso di EtherCAT come bus di campo e del embedded PC CX1020 di Beckhoff è risultato un'ottima soluzione per lo scambio di informazioni a livello I/O in particolare per sistemi di controllo di piccola e media dimensione, con caratteristiche ideali per garantire ottime prestazioni nel campo dell'automazione industriale.


Appendice A

Caratteristiche motore

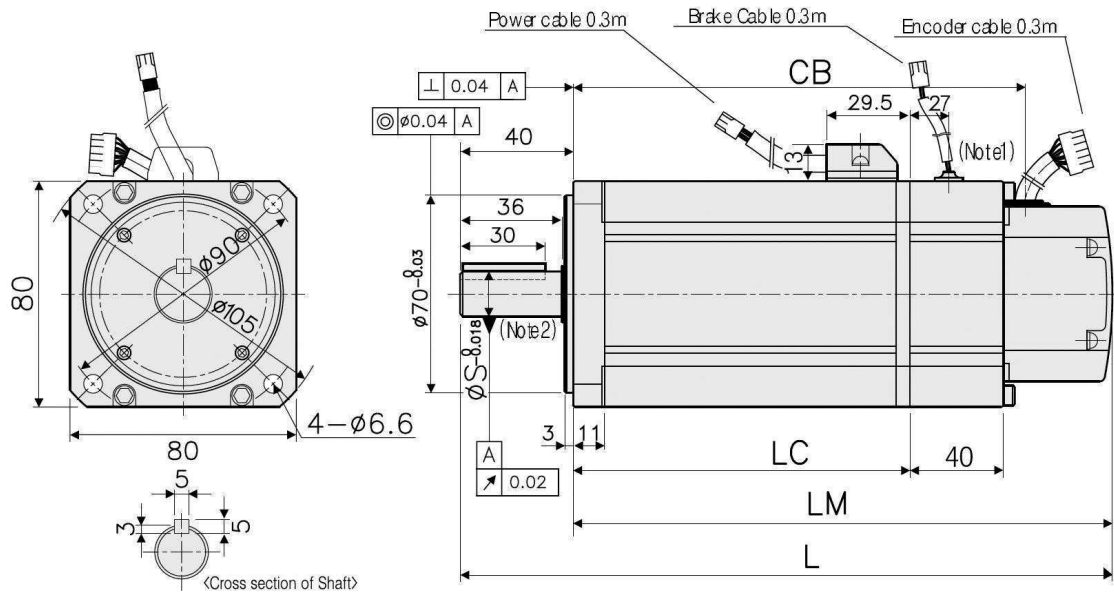
A.1 Servo motore AC

APM-SC05 di MECAPION :

Servo Motor Model (APM-□□□□□□)		SC03D	SC05D	SC06D	SC07D	SE 06D	SE 11D	SE 16D	SE 22D
Flange Size (□)			□80					□130	
Rated Power	[kW]	0.3	0.45	0.55	0.65	0.6	1.1	1.6	2.2
Rated Torque	[N · m]	1.43	2.15	2.63	3.09	2.86	5.25	7.63	10.5
	[kgf · cm]	14.6	21.9	26.8	31.6	29.2	53.6	77.9	107.1
Max. Instantaneous torque	[N · m]	4.29	6.44	7.88	9.29	8.59	15.75	22.92	31.51
	[kgf · cm]	43.8	65.7	80.4	94.8	87.7	160.7	233.8	321.4
Rated rpm	[r/min]		2,000						
Max. rpm	[r/min]		3,000						
Moment of inertia	[kg · m ² × 10 ⁻⁴]	0.674	1.092	1.509	1.927	6.569	11.999	17.339	22.67
	[gf · cm · s ²]	0.687	1.114	1.539	1.966	6.792	12.238	17.685	23.132
Allowable load inertia ratio		15 times of motor inertia		10 times of motor inertia		10 times of motor inertia			
Rated Power Rate	[kW/S]	30.36	42.19	43.68	47.90	12.31	22.97	33.63	48.61
Speed, Position Transducer	Standard(Notet)	Incremental 2500[P/R]				Incremental 3000[P/R]			
	Option	Absolute, 11/13bit Manchester communication				Absolute, 11/13bit Manchester communication			
Specification & Features	Protective Method	Totally enclosed, Non ventilated IP65(Excluding the shaft-through section and connectors)							
	Rated Time	Continuous							
	Ambient Temp.	Operating Temp. : 0~40[°C] · Storage Temp. : -20~80[°C]							
	Ambient Temp.	Lower than 90[%] (Avoid condensation)							
	Atmosphere	Avoid direct sunlight, no corrosive gas, inflammable gas, oil mist, or dust							
	E/V	Elevation/Vibration 49[m/s ²](5G)							
Weight	[kg]	1.85	2.52	3.18	3.90	5.5	7.54	9.68	11.78

 Note) Standard Encoder specification is 5[V] Line Driver.

A.2 Dimensioni motore



Model	External Dimension					Weight (kg)
	L	LM	LC	CB	S	
SC04A, SC03D	158(198)	118(158)	79	86.5(127)	14	1.88(2.92)
SC06A, SC05D	178(218)	138(178)	99	106.5(147)	16	2.52(3.56)
SC08A, SC06D	198(238)	158(198)	119	126.5(167)	16	3.18(4.22)
SC10A, SC07D	218(258)	178(218)	139	146.5(187)	16	3.90(4.94)

Appendice B

Glossario

ADS - Automation Device Specification

AMS - Automation Message Specification

API - Application Programming Interface

ARM - Acorn RISC Machine

ASM - Assembly

ASP - Active Server Pages

AT - Amplifier Telegram

ATL - Active Template Library

CAN - Controller Area Network

CIFS - Common Internet File System

CNC - Computer Numerical Control

CoE - CAN Over EtherCAT

COM - Component Object Model

CPU - Central Processing Unit

CRC - Cyclic Redundancy Check

CSMA/CD - Carrier Sense Multiple Access / Collision Detection

DC - Direct Current

DLL - Dynamic Link Library

DMA - Direct Memory Access

DPRAM - Dual-Ported RAM

DVD - Digital Versatile Disc

DVI - Digital Visual Interface

EAS - Elmo Application Studio

EEPROM - Electrically Erasable Programmable Read-Only Memory

EoE - Ethernet over EtherCAT

ESC - EtherCAT Slave Controller

EtherCAT - Ethernet for Control Automation Technology

FMMU - Fieldbus Memory Management Unit

FoE - File Access over EtherCAT

FTP - File Transfer Protocol

GB - Gigabyte

GPS - Global Positioning System

HMI - Human-Machine Interface

HTTP - HyperText Transfer Protocol

IEC - International Electrotechnical Commission

IEEE - Institute of Electrical and Electronics Engineers

IP - Internet Protocol

IPC - Industrial PC

I/O - Input/Output

ISO/OSI - International Organization for Standardization / Open Systems Interconnection

LAN - Local Area Network

LCD - Liquid Crystal Display

LDAP - Lightweight Directory Access Protocol

LVDS - Low-Voltage Differential Signaling

MAC - Media Access Control

MB - Megabyte

MDT - Master Data Telegram

MFC - Microsoft Foundation Classes

MIPS - Million Instructions Per Second

MP3 - Moving Picture expert group-1/2 audio layer 3

MSMQ - Microsoft Message Queuing

MTBF - Mean Time Between Failures

NDIS - Network Driver Interface Specification

NOVRAM - Non-Volatile Random Access Memory

OBEX - Object EXchange

OS - Operating System

PAN - Personal Area Network

PCB - Printed Circuit Board

PDI - Process Data Interface

PDO - Process Data Object

PDS - Power Drive System

PHY - Physical Layer

PLC - Programmable Logic Controller

POOM - Pocket Outlook Object Model

PPPoE - Point-to-Point Protocol over Ethernet

PPTP - Point-to-Point Tunneling Protocol

PQOAL - Production Quality OAL

PTP - Point-to-Point

RAM - Random Access Memory

RAS - Remote Access Service

RDP - Remote Desktop Protocol

RIL - Radio Interface Layer

RISC - Reduced Instruction Set Computer

RJ - Registered Jack

RS-232 - Recommended Standard 232

RT - Real-Time

RTC - Real-Time Communications

RTAI - Real-Time Application Interface

RTOS - Real-Time Operating System

SCADA - Supervisory Control And Data Acquisition

SDO - Service Data Object

SDK - Software Development Kit

SERCOS - Serial Real-time COmmunication System

SIM - Subscriber Identity Module

SIP - Session Initiation Protocol

SMB - Server Message Bus

SMS - Short Message Service

SNMP - Simple Network Management Protocol

SOAP - Simple Object Access Protocol

SoE - Sercos Over EtherCAT

SP - Service Pack

SPI - Serial Peripheral Interface

STL - Standard Template Library

STO - Safety Torque Off

SYNC - Oggetto di Sincronizzazione

TAPI - Telephony Application Programming Interface

TCP/IP - Transmission Control Protocol/Internet Protocol

TDP - Thermal Design Power

TFTP - Trivial File Transfer Protocol

TwinCAT - The Windows Control and Automation Technology

UDP - User Datagram Protocol

UPnP - Universal Plug and Play

USB - Universal Serial Bus

VB - Visual Basic

VGA - Video Graphics Array

VPN - Virtual Private Network

WAN - Wide Area Network

WAP - Wireless Application Protocol

WC - Working Counter

WMA - Windows Media Audio

WTL - Windows Template Library

XML - Extensible Markup Language

Bibliografia

- [1] *EtherCAT Brochure*, EtherCAT Technology Group, ETG Headquarters Ostendstraße 196 90482 Nuremberg Germany, Ottobre 2009.
- [2] D. Nicola, “Bus di campo per l’automazione industriale,” Ph.D. dissertation, Università degli studi di Padova - Facoltà di ingegneria, 2007-2008.
- [3] E. T. Group, “Ethercat the ethernet fieldbus.” ETG Headquarters Ostendstraße 196 90482 Nuremberg Germany: EtherCAT Technology Group, Maggio 2005.
- [4] M. Rostan and J. E. Stubbs, “Ethercat - enabled advanced control architecture.” ETG Headquarters Ostendstraße 196 90482 Nuremberg Germany: EtherCAT Technology Group, Luglio 2010.
- [5] S. Baldoni, “Dall’ipc all’automazione pc-based,” 2006.
- [6] L. Ma, F. Xia, and Z. Peng, “Integrated design and implementation of embedded control integrated design and implementation of embedded control systems with scilab,” 2008.
- [7] C. Paveri, “Componenti per l’automazione embedded.”
- [8] Beckhoff, “Addition to the family : Embedded pc cx1020,” 2005.
- [9] *Hardware documentation for CX1020 / CX1030 Embedded PC*, 1st ed., Beckhoff Automation GmbH, Eiserstr. 5 33415 Verl Germany, Giugno 2007.
- [10] S. Pavlov and P. Belevsky, *Windows® Embedded CE 6.0 Fundamentals*, Microsoft.

- [11] Beckhoff, *TwinCAT System Manager*, Beckhoff Automation GmbH, Eiserstr. 5 33415 Verl Germany.
- [12] —, *TwinCAT I/O*, Beckhoff Automation GmbH, Eiserstr. 5 33415 Verl Germany.
- [13] P. Yao, “Developing applications with evc++ 4.0.”
- [14] —, “Programming with embedded visual c++ 4.0,” 2004.
- [15] *Gold Solo digital servo drive installation guide*, 1st ed., ELMO Motion Control Ltd., 42 Technology Way Nashua, NH 03060 USA, Agosto 2010.

Ringraziamenti

Giunto al termine di questo lavoro desidero ringraziare ed esprimere la mia riconoscenza nei confronti di tutte le persone che, in modi diversi, mi sono state vicine e hanno permesso e incoraggiato sia i miei studi che la realizzazione e la stesura di questa tesi.

I miei più sentiti ringraziamenti vanno a chi mi ha seguito durante la redazione del lavoro di tesi:

- **Ch.mo Prof. Ing. GIULIO ROSATI** per avermi dato la possibilità di svolgere questo lavoro di tesi e per essere sempre stato presente e disponibile.
- **Ing. FABIO OSCARI** per la continua disponibilità e prontezza nei chiarimenti e suggerimenti, per la rilettura critica di tutti i capitoli della tesi e per avermi aiutato con i suoi consigli durante le varie fasi di questo percorso formativo.
- **Prof. Ing. GIOVANNI BOSCHETTI** per la disponibilità e i suggerimenti dati riguardo all'utilizzo del PC embedded e della sua programmazione.

Un ringraziamento va anche a tutti coloro che fanno parte della mia famiglia e a chi mi è sempre vicino nei momenti più belli ed importanti :

- Ai **miei Genitori** a cui dedico questo lavoro di tesi: senza il loro aiuto non avrei mai raggiunto questa meta. Sono davvero grato per il sostegno e per l'appoggio che mi hanno sempre garantito.
- A **Stefania** che mi ha sostenuto in ogni momento ed in particolare durante tutto il mio percorso di tesi. Grazie per avermi fatto dimenticare le ore passate in laboratorio davanti al PC e per esserti arrabbiata al posto mio con il Fornitore. Grazie per ogni momento passato con te e alla tua presenza costante all'interno della mia vita.

- A **mio Fratello** per avermi concesso di utilizzare la sua macchina nei lunghi viaggi verso Padova e per avermi restituito il portatile ad un mese dalla laurea.

Ultimo ma di certo non per importanza, un sentito ringraziamento spetta a tutti coloro che mi hanno accompagnato in questi stupendi tre anni universitari :

- Agli **amici della valle** che si sono sempre scomodati per venire a trovarmi e a passare serate memorabili in quel di Padova.
- Agli **amici da matricola** vedi Pancio Vic e Baldin con i quali ho vissuto i primi esami (dove solo io e Pancio passavamo ovviamente) ma anche le prime feste e i primi mercoledì in piazza.
- Agli **amici dell'università** : tutti ma proprio tutti coloro con i quali ho iniziato e trascorso i miei studi, con cui ho scambiato qualche pensiero, qualche idea e qualche risata. Potrei iniziare da Enri, compagno di avventure conosciuto una notte di carnevale a Venezia, dagli inquilini di via del Portello, dai compagni di calciotto, dai coinquilini della Ste (Alessia grazie ancora per la chiavetta e per internet), dai compagni di ferie sia estiva che invernale.
- Alla mensa **Forcellini** che mi ha sfamato con pasti deliziosi durante questi duri anni di studio.
- Ai **Coinquilini** passati e presenti e a tutti coloro che mi sono dimenticato di citare (il fornitore di Servotecnica non è stato dimenticato).