



AOX

# UN KIT PROGRAMMABILE IEC 61131-3

1 April 2011

## Premessa

Inizio da questo blog su **ElectroYou** un mio personale ed alternativo **corso sui PLC** rivolto a **studenti ed hobbisti di elettronica** che vogliono avvicinarsi a questi sistemi di controllo. Affronterò l'argomento da una direzione un po' insolita, partendo con la progettazione di un **PLC didattico** che potrà poi essere realizzato sottoforma di KIT con l'uso di un **semplice saldatore**. Una volta **assemblato il KIT**, utilizzerò questo PLC demo per continuare il corso con esempi di programmazione nel linguaggio standard **IEC61131-3**, illustrando **CoDeSys**, il diffuso ambiente di sviluppo software. Infine, dopo aver connesso il **PLC alla rete Internet**, mi soffermerò su alcuni argomenti relativi ai sistemi **Webserver** per la gestione remota del controllo mediante un comune **Browser**. Ho pensato di realizzare questo KIT appositamente per il mio corso per avere l'occasione di affrontare, pur sommariamente, sia gli aspetti **hardware** che **software** dei sistemi di controllo per l'automazione. Per questo motivo il **progetto del PLC** si svilupperà contemporaneamente alla pubblicazione dei pochi blogs nei quali intendo suddividere il corso e, solo al suo termine, il relativo materiale pubblicato potrà essere considerato definitivo.

## Le specifiche tecniche

Il punto di partenza del progetto dell'elettronica per il controllo di un'automazione è quello di fare un elenco delle cose che vorremmo mettere sulla scheda. L'elenco secco di queste cose è in genere uno dei primi input fornito ad un progetto di una scheda dedicata e spesso questo elenco lievita in continuazione, anche in corso d'opera, per la richiesta di funzioni sempre nuove e anche per lo scrupolo di riservarsi "la scorta della scorta". Un'altra cosa che allunga questa lista è la tentazione di pensare sempre al "non si sa mai che la posso usare anche per". Quest'ultimo elemento è il peggiore di tutti per un progettista, perchè lo costringe a mettere sulla scheda cose che sono ancora nello stato di nebbia e nella speranza che un giorno diventino indispensabili. Sono queste spesso le cose che rovinano la linearità, la pulizia, l'ottimizzazione ed anche l'estetica di una scheda. Poi il giorno che l'imprevedibile esigenza viene fuori si scopre che non è esattamente quella vagamente ipotizzata in partenza e nel frattempo magari il progetto ha anche già subito delle revisioni per altri motivi.

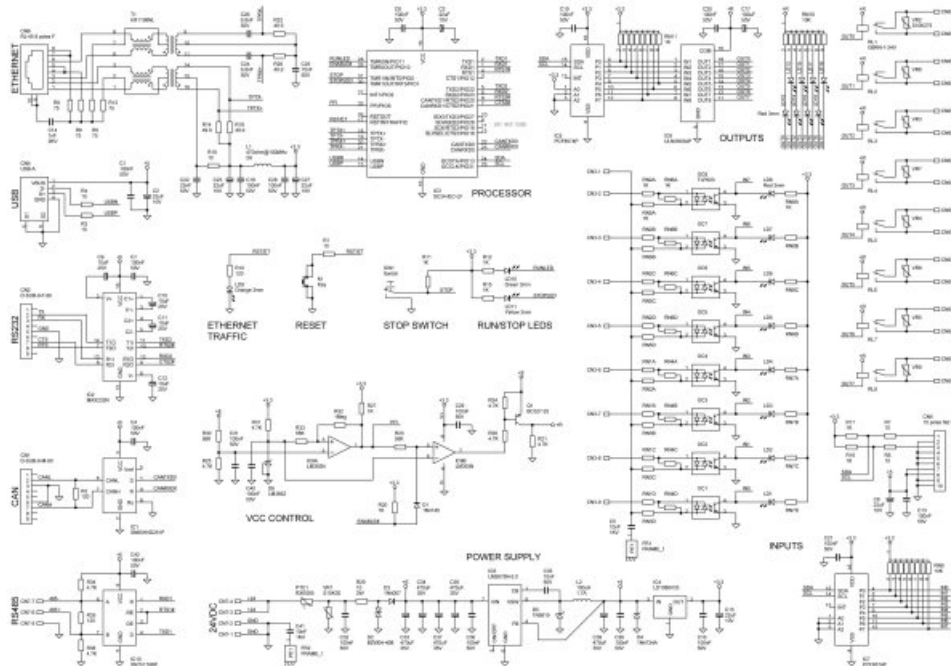
Questa "lista della spesa" va inoltre commisurata con tanti altri fattori molto importanti. Per esempio quanto abbiamo intenzione di investire nello sviluppo della scheda e quando deve costare la scheda. Qui le cose si complicano molto, in quanto manca spesso una pianificazione che tenga conto degli elementi nel loro insieme complessivo. Dedicare maggiori risorse sulla fase progettuale può sicuramente portare ad un prodotto più ottimizzato e quindi più economico, ma questo maggior costo di sviluppo va confrontato con il numero di schede da realizzare. Utilizzare e riciclare tecniche hardware e software, già messe a punto in precedenza, può sicuramente ridurre i costi di sviluppo, pur in presenza di una scheda ottimizzata, ma questo spesso costringe a forzature su quella lista iniziale di caratteristiche che vengono richieste. Non voglio comunque più annoiare nessuno con discorsi poco tecnici, ma questo era solo un estratto delle difficoltà relative alle scelte che si devono fare prima di realizzare una scheda. Alla fine, dopo aver combattuto tra dubbi e timori su quali siano le scelte da fare, ci si affida alle proprie sensazioni e si mette giù di getto la scheda:

- Dimensioni contenute il più possibile. Fissabile sia con 4 viti M4 che su un guscio di plastica con aggancio omega.
- Alimentazione 24V solo in DC per semplificare un po' di cose visto lo scopo didattico del progetto.
- 8 ingressi digitali a 24V sia PNP che NPN poichè un optoisolatore (per passare da 24V a logica) in AC ha un costo poco diverso da uno in DC.
- 8 relè da 8A con larghezza 1cm, molto utile per rimanere al passo con 2 poli di una morsettiera a vite.
- Interfaccia CAN perchè la gestione CANopen è nativa CoDeSys e non vogliamo negarcela per il solo costo del driver.
- Interfaccia RS485 perchè si presta all'espansione con protocolli standard ma anche custom e facilmente implementabili.
- Porta seriale RS232 perchè, anche se un po' obsoleta, soprattutto sui PC, fa sempre ancora molto comodo.
- Porta Ethernet: se ci rinunciamo togliamo le cose più interessanti che si possono poi fare nel seguito.
- Connettore USB per infilarci una chiavetta (magari quelle cortissime) con qualche GigaBytes di files dati e programmi. Molto utile per data-logger, appoggio pagine html ed immagini.
- Un connettore I2C per qualche esperimento di ampliamento a piacere (ovviamente a brevissima distanza).
- Programmazione CoDeSys nei 5 linguaggi della norma IEC61131-3 (ladder, lista, testo, blocchi, grafi) + schema elettrico a blocchi.
- Connessione in rete LAN ed Internet con pagine Web e trasferimenti in FTP.
- L'orologio invece lo risparmiamo perchè l'ora esatta l'andremo a chiedere ad un server NTP con l'Ethernet.

Queste sono le principali caratteristiche che mi sono proposto di ottenere dalla scheda tenendo conto delle cose migliori che offre il processore che adatterò e facendo particolare attenzione alla semplicità ma senza grosse rinunce. Diciamo che, allo stato attuale, questo mi sembra il miglior compromesso per lo scopo voluto.

## Lo schema della scheda

Lo schema complessivo della scheda è il seguente:



*Schema EY-CPU*

[Download](#)

[EY-CPU\\_Schema.pdf](#)

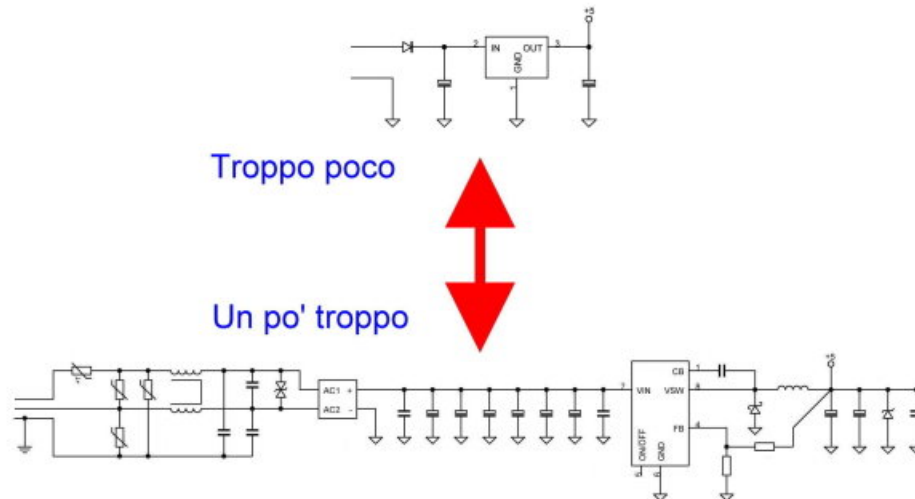
Avrei potuto frazionarlo nelle sue parti ma preferisco mostrarlo per intero in quanto abbastanza semplice e con etichette specifiche per le varie sezioni. A prima vista potrebbe sembrare complesso ma se lo si osserva con un po' di pazienza si nota subito che è l'insieme di tante cose singolarmente semplici. La stampa dello schema può essere ancora ben leggibile su un foglio A4, magari con una laser. Voglio ricordare che il progetto verrà realizzato appositamente per questo blog, che ne documenterà lo svolgimento in corso. Quindi ci potrebbero essere alcune cose da migliorare e per questo confido anche nelle capacità dei lettori.

Nel bordo sinistro del foglio ci sono le interfacce e le porte di comunicazione (ETHERNET, USB, RS232, CAN e RS485), mentre sulla destra dello schema abbiamo

le interfacce di ingressi ed uscite digitali, nonché il connettore per eventuali sperimentazioni (didattiche!) con il bus I2C. Nella parte centrale iniziamo dall'alto col processore e poi, scendendo, alcune parti di contorno come le segnalazioni a led dello stato del programma e dell'attività della rete Ethernet. Scendendo ancora troviamo una parte analogica, necessaria alla supervisione della tensione di alimentazione, principalmente con lo scopo di generare un interrupt (segnale PFI) per il salvataggio delle memorie ritentive al power-off e con lo scopo di tenere spenti i relè delle uscite durante il power-on-off. Esistono semplici integrati per lo scopo ma un po' di circuiti di vecchio stampo non guastano mai su un kit. Infine in basso, al centro, l'alimentatore di tutta la scheda con la creazione delle due tensioni necessarie agli integrati (+5V e +3.3V). Lo scopo didattico ed economico del progetto non consente di utilizzare nessun tipo di isolamento galvanico per le porte di comunicazione (ad esclusione, per sua natura, della porta Ethernet), né di prevedere particolari protezioni sulle relative interfacce.

### **Note sull'alimentatore**

L'alimentatore di una scheda è spesso quella sezione del progetto dove le scelte da fare sono le più difficili. Infatti questa parte richiede componentistica di un certo ingombro e costo, richiede una stima degli assorbimenti dei circuiti interni, è soggetta spesso a disturbi elettromagnetici della peggior razza, può avere problemi di temperatura e quindi di maggior mortalità e tante altre cose. Quindi le scelte da fare non sono semplici, anche perchè le condizioni che dovrà affrontare un alimentatore non sono sempre così note a priori. Vorrei utilizzare l'esempio dell'alimentatore per dire che un progetto, anche di una sola parte di una scheda, può essere fatto in mille modi, con costi e risultati anche molto diversi: il difficile è sceglierne uno adatto al caso. Nella foto seguente ho voluto raffigurare due casi limite per realizzare un alimentatore che porti la 24V (standard nell'automazione) alla +5V necessaria per i circuiti.



### Power-Supply

Questi due casi, piuttosto all'opposto tra loro, non sono poi così neanche dei limiti estremi. Esistono e si vedono spesso soluzioni ancora più semplici della prima e più complesse e sofisticate della seconda. Mi capita per esempio di vedere spesso schede la cui alimentazione a +5V è fornita direttamente tramite un morsetto dall'esterno magari affiancata anche alla 24V di un quadro. Lavorando su alimentatori switching, per i quali ho avuto sempre un particolare interesse, mi è capitato di dover realizzare alimentatori da quadro con uscita a +5V. Mi sono sempre chiesto, da curioso che sono, che uso ne facessero visto che la +5V non dovrebbe mai circolare dentro una canalina di un quadro. Dall'altra parte su PLC commerciali si trovano soluzioni estremamente interessanti e sofisticate. Per esempio una volta sull'ingresso di alimentazione a 24VDC di un PLC di fama mondiale ho trovato varistori da tensione di rete (470V diametro 10). Anche qui, da grande curioso, mi sono chiesto se avessero sbagliato o avessero, per semplicità di magazzino, voluto usare solo varistori di quella tensione. Per molto tempo sono rimasto con questo dubbio poi un giorno, facendo delle prove di burst EMI e combinando insieme alcuni elementi, mi sono dato una spiegazione o almeno penso che sia la spiegazione.

Comunque tornando all'alimentatore della scheda e sfumando in continuità tra questi due limiti indicati mi sono assestato su una situazione intermedia che dovrebbe garantire un corretto funzionamento anche al di là degli scopi prefissati. In ingresso all'alimentatore per prima cosa ho inserito un PTC, che ha la funzione di fusibile autoripristinante, per evitare di assorbire una corrente eccessiva in caso sfortunato di qualche guasto interno. Poi il varistore ed il condensatore da 1KV (tra negativo e vite di fissaggio all'eventuale frame metallico) possono costituire, in coppia, un'economica ma efficace prima barriera per eventuali disturbi EMI provenienti dall'esterno (sia di modo comune che differenziali). Un resistore da 10ohm, subito all'ingresso, potrebbe sprecare in calore al massimo un paio di volt di tensione, trascurabili nel passaggio da 24V a 5V, ma sicuramente è molto salutare per la

vita della scheda, nonché per il lavoro ingrato che deve svolgere il transil posto subito a valle. La scelta di ottenere la prima tensione di alimentazione a +5V con uno stabilizzatore switching è dovuta a motivi di riduzione del calore, che invece si avrebbe con uno stabilizzatore lineare, visto che il circuito intero potrebbe richiedere 2/3W di potenza. Inoltre uno stabilizzatore switching consente di ridurre gli ingombri, per il fatto che non necessita di dissipatore, ma anche di realizzare un'ulteriore barriera ai disturbi provenienti dall'esterno, grazie alla presenza del filtro LC del PWM. E' mia abitudine mettere uno zener da 5.6V 1W sull'uscita a +5V sia, per calmare eventuali transitori di tensione, ma soprattutto per evitare che guasti seri dell'alimentatore provochino a cascata il guasto dei circuiti interni. In riparazione ovviamente occorre sostituire tale diodo zener (quasi sempre in corto) ma evita di sostituire componenti più costosi e delicati. Per la tensione a +3.3V ho infine utilizzato uno stabilizzatore lineare di tipo low-dropout, ossia che non richiede grosse differenze di tensione tra ingresso ed uscita per fare correttamente il suo lavoro, visto che il salto da effettuare è di soli  $5-3.3=1.7V$ . Inoltre è importante utilizzare un modello di stabilizzatore con tolleranza di uscita abbastanza bassa (max 2%) in quanto il processore ha un circuito di reset tarato su una soglia di 3.1V.

## **Gli ingressi e le uscite digitali**

Ho voluto inserire in questa scheda un certo numero di ingressi ed uscite per consentire la scrittura ed il test dei propri programmi CoDeSys senza dover ricorrere a particolari schede di espansione.

La scelta di utilizzare optoisolatori per gli ingressi di una scheda per uso didattico può sembrare eccessiva ma alla fine è sempre la più conveniente comunque. Avremmo potuto mettere dei semplici partitori resistivi per abbassare la 24V alla 3.3V della logica ma sarebbe poi stato necessario proteggere gli integrati dai transitori. Avremmo potuto interporre dei transistors come translators di livello ma poi alla fine il circuito non sarebbe stato più semplice. Invece con optoisolatori con ingresso AC possiamo applicare tensioni polarizzate sia PNP che NPN, translare il livello di tensione da 24V a +3.3V ed isolarci galvanicamente il che non fa certo male dal punto di vista dei disturbi.

Per le uscite digitali ho utilizzato dei relè elettromeccanici mettendo a disposizione in morsettiera tutti i loro contatti NO e questo permette sicuramente una certa flessibilità di utilizzo. Ho predisposto su tali contatti un varistore per proteggerli da scintille di commutazione e ridurre le possibili influenze sulla scheda. I relè richiedono inoltre l'uso di un circuito di disabilitazione durante le fasi di accensione e spegnimento della scheda, per evitare eventuali sfarfallamenti degli stessi. Infine voglio evidenziare la presenza di condensatori posti in prossimità del pin 10 del driver a transistors dei relè (ULN2803) necessari per far lavorare bene i diodi di ricircolo incorporati in tale integrato. Infatti all'apertura del relè il diodo di ricircolo frena l'extratensione a 0.7V sopra la tensione presente sul pin 10 e se tale tensione non è ben vincolata al riferimento di massa questa può elevarsi a valori dannosi.

La parte di ingressi ed uscite digitali è realizzata attorno a circuiti integrati di espansione IO pilotati col bus I2C, lo stesso bus che mettiamo a disposizione su un connettore per chi volesse proseguire nell'implementazione, su una propria scheda, di altre periferiche. Tuttavia voglio sottolineare, fino alla noia, che il bus I2C non è un bus di campo e per la natura dei suoi segnali non può essere portato a "passeggio" tra varie schede ne tantomeno dentro un quadro o dentro le canaline dell'impianto elettrico di una casa. Quindi questa porta è esclusivamente per sperimentazioni da laboratorio.

## **Il processore**

Al processore utilizzato nel progetto si deve dare il merito di tutte le funzionalità e caratteristiche della scheda. Questo dispositivo non è un vero e proprio circuito integrato, ma un modulo molto compatto sul quale sono assemblati alcuni chips. Principalmente sul modulo è presente un microprocessore tipo Intel 186 a 16-bit, una memoria Flash ed una memoria Ram (ciascuna da 8Mbytes). Le funzioni software richieste dal sistema operativo Real Time, dal file-system, dalla gestione dei vari protocolli di comunicazione e da una programmazione CoDeSys richiedono una certa quantità di memoria e per questo il microprocessore, che già dispone di numerose periferiche, è stato supportato esternamente da questi dispositivi. Il tutto è realizzato su un unico componente in forma di classico circuito integrato standard in DIL32.

Le principali connessioni del processore sono relative alle porte di comunicazione. Dispone infatti di 3 porte seriali, 2x CAN, 1x Ethernet, 1x USB e di porte SPI ed I2C per l'espansione locale. Le porte seriali e CAN sono comunque parzialmente in alternativa tra loro e nei confronti dei comuni pins di IO del processore. Con le scelte effettuate in questo progetto, rimarrà libera la porta SPI normalmente utilizzata per schede di memoria SD-card e per altri dispositivi periferici. Il processore è dotato internamente anche di un Real Time Clock per l'ora di sistema ma, se non è supportato da un RTC esterno con pila di backup, non mantiene l'orario in mancanza di alimentazione. Per semplicità del progetto non sarà implementato questo dispositivo esterno, assumendo che l'orologio incorporato nel processore sia necessariamente resincronizzato con un server NTP ad ogni power-on (ammesso che per il nostro scopo sia necessario disporre della data/ora). Per quanto riguarda il salvataggio delle memorie Retain del programma IEC è necessario fornire un segnale di interrupt (PFI) al processore in mancanza della tensione di alimentazione. Per il loro salvataggio si utilizzerà una parte modesta degli 8Mbytes del memoria Flash. In alternativa il salvataggio può essere eseguito su memorie SPI esterne, per esempio di tipo FRAM per ottenere dei tempi di scrittura molto bassi (7ms per 8Kbytes). In applicazioni diverse dalla programmazione IEC potrebbe non essere indispensabile l'utilizzo di memorie Retain, in quanto, tramite il file system, si possono salvare configurazioni e parametri dell'applicativo su dei files.

## Porta Ethernet

Questa è la porta di comunicazione più interessante della scheda in quanto offre molte funzionalità e tutte utilizzabili sia in rete locale LAN che remota tramite una connessione Internet. Una prima applicazione di tale porta è sicuramente quella della programmazione e debug tramite CoDeSys. Inoltre l'applicativo IEC potrà inviare E-mail, anche con files allegati (molto utili nei data-logger), mediante l'apposito blocco funzione che utilizza il protocollo SMTP. Come già accennato si potranno inoltre effettuare delle interrogazioni dell'ora esatta mediante il protocollo NTP. Altrettanto importante è la funzionalità di Web-server che permette di accedere in modo remoto alle pagine Web memorizzate sulla scheda, interagendo con le variabili del programma. Queste pagine Web vengono disegnate graficamente mediante lo stesso tool CoDeSys con la Web Visualization. Per terminare questa sommaria descrizione delle possibilità offerte dalla porta Ethernet, si può dire che le operazioni di manutenzione del software possono essere realizzate anche mediante il trasferimento di files utilizzando la funzione di FTP server.

Tornando alle questioni hardware, l'interfaccia Ethernet è una delle periferiche incorporate nel single-chip per cui è compito della scheda adattare i segnali nei confronti del supporto di trasmissione fisica della rete. A tal scopo è utilizzato un trasformatore di segnale a due porte (una per la trasmissione ed una per la ricezione). I segnali che transitano su tali porte hanno frequenze molto elevate e probabilmente sono quelli che richiederanno la maggior attenzione possibile in fase di stesura del circuito stampato. Il cavo di connessione Ethernet trasporta i dati mediante due distinti doppini in una connessione di tipo punto-punto nei confronti di uno switch o di un router che si occuperà dello smistamento dei pacchetti dati agli altri dispositivi della rete. In entrambe le terminazioni di questi doppini, e quindi anche sulla nostra scheda, sono previste delle apposite resistenze di carico da 75ohm. La parte hardware di interfaccia Ethernet sollecita abbastanza la tensione di alimentazione a +3.3V per cui è necessario anche un filtro di tipo pi-greco per disaccoppiare l'alimentazione di questa interfaccia da quella degli altri circuiti. Infine, durante la comunicazione dati, un apposito pin di uscita del processore piloterà direttamente un led lampeggiante per indicare l'attività della porta.

## Porta seriale RS232

Su questa interfaccia c'è ben poco da dire in quanto realizzata con il solito MAX232. Questo componente, utilizzando 4 condensatori connessi ai suoi pins, riesce a creare delle tensioni doppie (o quasi) della +5V di alimentazione, una positiva ed una negativa. Infatti lo standard RS232 richiede che i dati siano inviati con livelli di tensione +/-10V riferiti alla massa (non sono quindi segnali differenziali e per questo sono poco adatti a realizzare distanze in ambienti rumorosi). I segnali a livello logico (nel nostro caso 0-3.3V) sono così trasformati, da questo circuito integrato, in livelli +/-10V. Nello schema della scheda sono stati previsti, oltre al TX ed RX, anche i



segnali di handshake hardware RTS e CTS. In particolare il segnale RTS può anche essere utilizzato come abilitazione della trasmissione nel caso si voglia inserire un adattatore esterno RS232-RS485 per ottenere un'ulteriore porta RS485.

La porta seriale RS232 non dispone, per semplicità del progetto didattico, di particolari protezioni sui segnali del relativo connettore. Tuttavia, per un uso normale, non dovremmo avere particolari problemi con questo vecchio ma consolidato componente.

## **Porta USB**

L'interfaccia della porta USB è la più semplice di tutte in quanto realizzata con solo 2 resistenze di basso valore connesse in serie. In realtà questa è la configurazione minima essenziale per tale tipo di porta in quanto il nostro scopo è principalmente quello di utilizzarla come host di una chiavetta di memoria (device). Tale porta potrebbe inoltre essere configurata anche in modalità device per connetterla ad un PC per la programmazione CoDeSys, tramite una COM virtuale. Tuttavia la programmazione tramite la porta Ethernet raggiunge prestazioni e funzionalità diversamente ottenibili con tutte le altre porte. Pensando quindi di utilizzarla per le chiavette di memoria USB, nello schema viene sempre applicata la tensione +5V di alimentazione per tale dispositivo esterno. Anche in tale interfaccia non sono state previste particolari protezioni.

## **Interfaccia CAN**

L'interfaccia CAN è gestita direttamente da CoDeSys utilizzando il protocollo CANopen. Per questa interfaccia occorrerà solo caricare nel tool i relativi files di configurazione .EDS delle espansioni connesse. Il protocollo CANopen può operare su questa scheda fino alla massima velocità di 1Mbit/s e la connessione al supporto fisico di trasmissione avviene tramite un apposito circuito integrato driver. Questo componente è già stato realizzato con adeguate protezioni a bordo per cui è normalmente richiesta solo la resistenza di terminazione da 120ohm (assumo per semplicità che questa scheda master sia sempre connessa ad un estremo della linea). In situazioni che esulano dallo scopo di questo progetto, un bus di campo richiede ulteriori accorgimenti per ottenere una maggior robustezza ed immunità ai disturbi come l'isolamento galvanico e l'inserimento di ulteriori componenti di protezione.

## **Interfaccia RS485**

Per questa interfaccia, essendo anch'essa di tipo differenziale, valgono le stesse considerazioni hardware fatte per la porta CAN. In questo caso sono state introdotte anche due resistenze di polarizzazione della linea differenziale nel caso di bus floating, ossia quando nessun nodo pilota la linea in trasmissione. Il segnale di abilitazione alla trasmissione è gestito tramite la linea RTS in modo automatico

dalla seriale del processore. Questo tipo di interfaccia consente di gestire una comunicazione multipunto utilizzando sia protocolli standard che realizzati in modo proprietario. Ovviamente, essendo una linea half-duplex, cioè nella quale un solo dispositivo alla volta può prendere il controllo della trasmissione, occorre gestire in modo opportuno il traffico dei pacchetti dati mediante il protocollo.

## **Circuito di controllo della tensione**

Questa parte è la più analogica di tutto lo schema ed ha il compito di monitorare la tensione di alimentazione a 24V a monte del primo stabilizzatore. Gli scopi di questo circuito sono due. Il primo è quello di generare un segnale attivo alto (PFI) quando la tensione è inferiore a circa 16V (con un'isteresi di un paio di volt). Il segnale PFI genera direttamente un interrupt al processore quando viene a mancare la tensione di alimentazione, permettendogli di salvare le memorie Retain del programma IEC in una porzione della Flash. Il secondo scopo è quello di fornire la tensione di alimentazione dei relè di uscita solo se la tensione è superiore alla soglia ed il processore ha completato le operazioni di inizializzazione, portando a massa il pin ENABLE. Per il riferimento di tensione del comparatore è stato utilizzato un reference da 1.24V costituito dall'integrato LM385. Il partitore di resistenze collegato alla tensione di alimentazione monitorata può essere modificato in caso di tensioni di alimentazione differenti.

## **Conclusione**

La prima parte del mio percorso termina qui. Ho definito gli obiettivi e le caratteristiche hardware della scheda, realizzato poi lo schema elettrico completo e data una breve descrizione delle sue parti. Ora il prossimo passo è quello di realizzare concretamente la scheda, progettandone il layout del circuito stampato. In realtà del circuito stampato esiste già una bozza approssimativa e con qualche pista tracciata che mi è servita in fase di schematico per capire cosa sarebbe stato conveniente metterci per avere un risultato ottimale allo scopo. A mio avviso, come in tutte le cose, è necessario valutare il lavoro nel suo insieme perchè per ottenere un risultato almeno discreto occorre procedere per passi successivi ma anche fare in modo che questi passi siano tra loro compatibili.

Estratto da "<http://www.electroyou.it/mediawiki/index.php?title=UsersPages:Aox:un-kit-programmabile-iec61131-3-2>"