



Ergotech

Green Drive

High torque plastic strain wave gear

ErgotechGreenDrive is a strain wave powered gearbox

- Fully integrated with frameless motor and control system
- High torque density, brushless PMSM
- High resolution positioning sensors
- RS 485 or one wire driver board
- Low weight due to technopolymers
- Post-industrials materials



Sommario

Comunicazione tra motore e PC via USB	3
Applicazioni per comunicare con il motoriduttore.....	3
Il modulo TCOM.....	4
Comunicazione tra motore e PC con schede commerciali.....	5
Comunicazione diretta tra motore e PLC in RS485.....	6
Protocollo di comunicazione	7
Parametri di comunicazione.....	8
TMOT1 - Communication parameters (read).....	8
TMOT1 - Communication parameters (write).....	9
MEANINGS.....	10
NOTES.....	10
Modulo di comunicazione TSIO	11
Registri motoriduttore e scheda TSIO.....	12
Schema di collegamento nel caso si utilizzi il solo PLC	13
Schema di collegamento nel caso si utilizzi un PC.....	14
Interfaccia di collegamento.....	15

Comunicazione tra motore e PC via USB

Questo è il metodo di comunicazione più semplice che vi permette di utilizzare tutte le applicazioni open source studiate per facilitare l'utilizzo del motoriduttore e i test sul sistema in cui il motoriduttore viene integrato (vedi paragrafo applicazioni).

Nei motori EGD1 con l'attuale firmware (versione 2.3), la comunicazione è fissata a 1 Mega Baud.

Applicazioni per comunicare con il motoriduttore

Nel caso abbiate scelto di comunicare con il motoriduttore tramite il PC, potrete utilizzare una serie di applicazioni open source e "Portable" (che non richiedono installazione) progettate per dialogare con il motoriduttore e monitorarne il funzionamento. Una volta scaricato il file .zip delle applicazioni, è necessario scompattarlo e salvarlo in una qualunque cartella del pc; poi si lancia l'eseguibile Automation.exe che a sua volta avvia tutte le altre applicazioni.

Il file .zip con tutte le applicazioni può essere scaricato a questo indirizzo: www.ergotech.it

E' anche possibile scaricare le singole applicazioni e i singoli file di documentazione (che potrebbero essere più aggiornati) dal sito www.theremino.com, per mezzo dei seguenti link:

Automation: per programmare tutto il sistema che comprende più motoriduttori e sensori, dialogare con le altre applicazioni e interagire con l'utente, anche per mezzo di messaggi sonori e vocali.

Motors: gestisce il protocollo di comunicazione con il motoriduttore e eventuali altri moduli come lo TSIO che si occupa di InOut in alternativa all'HAL.

HAL: si occupa degli input/output (ADC, InOut digitali, PWM, contatori, sensori, ecc.).

SlotViewer: visualizza gli Slot che contengono i valori numerici e le stringhe di testo che si utilizzano per comunicare tra le applicazioni del sistema.

SignalScope: visualizza i segnali (posizioni dei motori, temperature, velocità, ecc.)

Logger: aggiorna in tempo reale un file di log, con i segnali desiderati (temperature, velocità, ecc.)

Graphs: visualizza in tempo reale i file di log

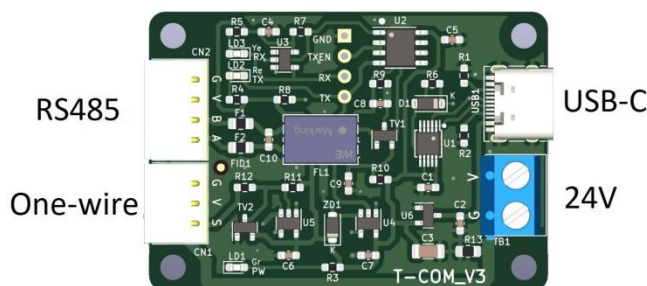
Programmare in Automation per mezzo della intelligenza artificiale.

Se progettate programmi e sistemi con le nostre applicazioni [leggete questa pagina!](#)

Il notebook LM cercherà quello che vi serve tra centinaia di pagine di documentazione e potete anche farvi scrivere interi programmi, basta chiedere!

Il modulo TCOM

Per consentire la comunicazione tra motoriduttore e PC è possibile utilizzare il modulo TCOM.



A destra in alto c'è il connettore USB-C da collegare a una porta USB del pc.

Subito sotto c'è il connettore di alimentazione a 24 V.

A sinistra in alto c'è il connettore seriale 485, che è il protocollo di comunicazione utilizzato su tutti i modelli Ergotech Green Drive 1. A sinistra in basso c'è il connettore seriale one-wire utilizzato per alcuni modelli di motoriduttore, predisposti per comunicare con questo protocollo di comunicazione alternativo.

I vantaggi della scheda TCOM sono due:

- La possibilità di comunicare sia in RS485 che in one-wire
- La presenza di connettori su cui sono riportati sia i segnali che l'alimentazione.

Connettori per TCOM

Per collegare l'Ergotech Green Drive 1 al pc tramite la scheda TCOM con protocollo di comunicazione RS485 è necessario dotarsi di terminali e connettore, disponibili in commercio su tutti i principali siti di e-commerce:

- Terminali del tipo [Molex 0008701039](#)
- Connettore del tipo [Molex 50375043](#), a 4 fili

Nel collegamento dei 4 fili al connettore è necessario rispettare l'ordine specificato sulla scheda TCOM e riportato anche sull'etichetta, in cui sono specificate le posizioni del ground (G), dell'alimentazione (V), del segnale RS485_B (B) e del segnale RS485_A (A).

Black: ground	WARNING!
Red: +24±2 V	Do not
Blu: RS485_B	reverse
Yellow: RS485_A	polarity

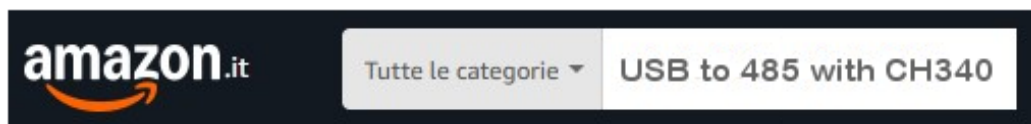


Etichetta EGD1

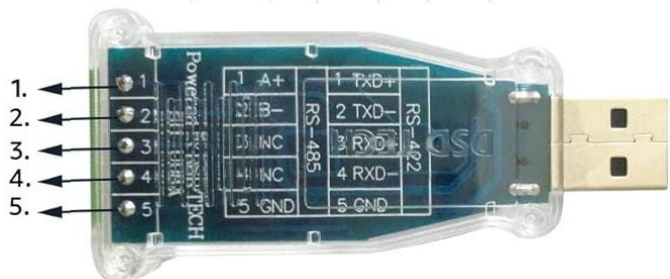
Comunicazione tra motore e PC con schede commerciali

Per la comunicazione tra motore e PC tramite USB è possibile utilizzare anche schede di comunicazione, diverse dalla scheda TCOM.

Queste schede sono disponibili in commercio e reperibili sui principali siti e-commerce cercando “usb to 485 with CH340” come mostrato di seguito.



1	A+	RS-485	1	TXD+	RS-232
2	B-		2	TXD-	
3	NC		3	RXD+	
4	NC		4	RXD-	
5	GND		5	GND	



In questa immagine si vede una scheda commerciale che abbiamo provato e che funziona bene.

Questo particolare modello contiene il chip CH340E che comunica più velocemente di altri.

In ogni caso quando fate le ricerche assicuratevi che il chip sia un CH340 e non un FTDI, che funzionerebbe ugualmente ma con prestazioni inferiori e maggiori difficoltà di configurazione.

Con queste schede si potrà comunicare solo in RS485 e non in One-wire come è invece sarebbe possibile utilizzando il modulo TCOM e motoriduttori speciali che comunicano in One-wire.

Inoltre, non disponendo del morsetto di alimentazione del modulo TCOM, si dovrà collegare il 24 V direttamente tra GND e +V del cavo a 4 fili che va alla catena di motori.

Comunicazione diretta tra motore e PLC in RS485

Per consentire la comunicazione tra motoriduttore e PLC è necessario utilizzare un modulo adattatore RS485 per PLC. Per esempio il modulo Siemens xxx

Per comunicare direttamente con il motore dovrete scrivere un programma nel PLC seguendo il protocollo e i parametri di comunicazione indicati nelle prossime pagine.

Si possono progettare programmi per controllare i motori in modi diversi e numerosi parametri non si utilizzano per muovere il motore. Ad esempio leggere la temperatura e impostare limiti di accelerazione e velocità, i parametri del PID, l'identificatore del motore e la password.



Ad esempio una importante operazione che dovrebbe fare il programma sul PLC è di impostare l'identificatore dei motori anche quando essi sono già tutti collegati alla linea seriale. Tutti i prodotti concorrenti per impostare l'ID dei motori devono per forza scollegarli uno per uno dalla linea e collegarli a un programmatore e questo in alcuni casi potrebbe essere molto scomodo.

Per cui abbiamo progettato i motori per poterli identificare manualmente avvicinando un magnete ad essi. Il programma all'avvio della procedura di identificazione imposta i motori per rispondere un messaggio quando il sensore supera un certo livello e poi li identifica uno dopo l'altro.

Programmare i PLC per compiti così complessi è piuttosto difficile, per cui un buon consiglio potrebbe essere di utilizzare prima le nostre applicazioni su un PC e familiarizzarsi con le operazioni principali di controllo dei motori.

Potrete così controllare la linea di comunicazione, identificare i motori e provarli e solo in un secondo tempo scollegare il PC e usare il solo PLC.

Eventualmente in seguito potreste anche leggere le procedure usate nel codice della nostra applicazione Motors. E potrete copiarle nel PLC senza problemi dato che sono tutte Open Source e libere da copyright.

Protocollo di comunicazione

Nel caso si voglia comunicare direttamente con il motoriduttore dal PLC, è necessario rispettare il protocollo descritto di seguito.

La comunicazione avviene in seriale (8N1 – 8 bit, no parity, 1 bit di stop) con pacchetti strutturati secondo il seguente schema.

Pacchetto di trasmissione (Instruction packet):

Header1	Header2	Packet ID	Length	Instruction	Param 1	...	Param N	Checksum
0xFF	0xFF	Packet ID	Length	Instruction	Param 1	...	Param N	CHKSUM

Pacchetto di ricezione (Return packet):

Header1	Header2	Packet ID	Length	Error	Param 1	...	Param N	Checksum
0xFF	0xFF	ID	Length	Error	Param 1	...	Param N	CHKSUM

Per tutti i dettagli del protocollo (ad esempio la composizione del checksum e la lista dei bit di errore) fare riferimento al paragrafo “Protocolli di comunicazione” del seguente documento:

https://www.theremino.com/wp-content/uploads/files/Theremino_Motors_Help_ENG.pdf

Parametri di comunicazione

Le prossime pagine illustreranno i parametri di comunicazione per la lettura e la scrittura dei registri del motoriduttore.

TMOT1 - Communication parameters (read)

Adr.	Memory	Bytes	Default	Min	Max	Unit	Prot.	R/W
000	Serial Number	3		0	2 ²⁴			RO
003	Device Version	1		0	250			R
004	Device Sub-Version	1		0	250			R
005	Firmware Version	1		0	250			R
006	Firmware Sub-version	1		0	250			R
007	Boot loader version	1		0	250			R
010	Actual Position (Note 1)	4		-2 ³¹	+2 ³¹	Step		R/W
014	Actual Torque (Note 2)	2		0	32k	nm		R
016	Actual Acceleration (Note 3)	2		0	65000	Rpm/S		R
018	Actual Velocity (Note 4)	2		-32k	+32k	Rpm		R
020	Actual Current	2		0		1 mA		R
022	Actual Voltage	2		0		1 V		R
024	Actual Temperature	1		0	250	1 °C		R
025	Moving Status (Note 5)	1		0	250			R
026	Hardware Error Status	1		0	250			R
027	ID Setting HallSensor value	2	2000	0	4095			R
030	MagSens1 A	2		0	5760			R
032	MagSens1 X	2		0	65535			R
034	MagSens1 Y	2		0	65535			R
036	MagSens2 A	2		0	5760			R
038	MagSens2 X	2		0	65535			R
040	MagSens2 Y	2		0	65535			R

TMOT1 - Communication parameters (write)

Adr.	Memory	Bytes	Default	Min	Max	Unit	Prot.	R/W
050	Write protection (Note 6)	1	1	0	1			W
051	Working Mode (Note 7)	1	1	0	250		PRT	W
052	Reply Mode (Note 8)	1	0	0	1			W
053	Broadcast Code	1	254	200	250			W
054	Baud Rate Index	1	4	0	7			W
055	ID (device identifier) (Note 9)	1	1	0	199		PRT	W
056	ID Setting Tolerance	1	10	1	250			W
058	Goal Position (Note 1)	4	0	-2 ³¹	+2 ³¹	Step		W
062	Goal Torque (Note 2)	2	1000	0	32k	nm		W
064	Goal Acceleration (Note 3)	2	10000	0	65000	Rpm/S		W
066	Goal Velocity (Note 4)	2	0	-32k	32k	Rpm		W
068	Position P Gain	1	20	0	250			W
069	Position I Gain	1	20	0	250			W
070	Position D Gain	1	20	0	250			W
071	Dead Band	1	0	0	250	Step		W
072	Zero Speed Current	1	0	0	255			W
080	MagSens1 GainAxis	1	0					W
081	MagSens1 Gain	1	0	0	255			W
082	MagSens1 OffsetX	1	0	-127	+127			W
083	MagSens1 OffsetY	1	0	-127	+127			W
084	MagSens1 GainAxis	1	0					W
085	MagSens1 Gain	1	0	0	255			W
086	MagSens1 OffsetX	1	0	-127	+127			W
087	MagSens1 OffsetY	1	0	-127	+127	Step		W
088	MagSens_WorkMode	1	0	0	255		PRT	W
100	Serial And Password (Note 10)	16					PRT	WO

MEANINGS

- PRT** = Write protected
- R** = Read (and eventually write)
- W** = Write (and eventually read)
- R/W** = Read and Write
- RO** = Read Only
- WO** = Write Only

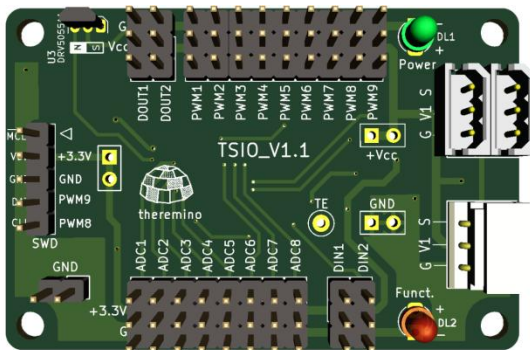
NOTES

- (1) POSITION** - Measure unit is "step". Before the harmonic drive the steps are $384 * 60$ (23040) per turn. After the harmonic drive the steps are $384 * 60 * 20$ (460800) per turn. Writing zero to the Present Position the actual motor position becomes the zero point. But you could also assign a non zero value that will be the actual motor position.
- (2) TORQUE** - Measure unit = milli-newton on the output shaft (EGD1 max = 10 nm)
- (3) ACCELERATION** - Measure unit = RPM per sec.
- (4) VELOCITY** - Measure unit = RPM
- (5) MOVING STATUS** - 0 = Not moving
- (6) PROTECTION** - Some parameters may be written also in flash to retain the value between power cycles. <Write protection> needs to be turned Off (0) before modifying a value to write it also in flash. See the column "Prot."
- (7) WORKING MODE** - 0 = nothing / 1 = position mode / 2 = constant speed mode / 200 = StartSettingID / 201 = StopSettingID / 253 = RESET MOTOR
To modify this value the WriteProtect must be unlocked with a zero.
- (8) REPLY MODE** - 0 = Reply only for read instructions / 1 = Reply for all instructions
- (9) DEVICE ID** - This value is preserved between the power cycles.
To modify this value the WriteProtect must be unlocked with a zero.
The max value is 199 because the broadcast codes are from 200 to 254
- (10) SERIAL NUMBER** (3 byte) + PASSWORD (12 byte). After boot-loader programming the serial number is 0 and the password is not set (all bytes are 0). If you write a serial number with password not set (all bytes are 0) the serial number will be programmed; this write operation may be repeated with different serial numbers until the password is not set. If you write a serial number with password set (not all bytes are 0) the serial number and the associated password will be programmed and subsequent serial number write operations will succeed only if associated with the same password. If the password is wrong the write operations will fail. To reset the password and the serial number the boot loader must be programmed.

Modulo di comunicazione TSIO

Questo modulo può leggere e scrivere valori analogici e digitali con il vantaggio che si trova sulla stessa linea di comunicazione dei motoriduttori, senza la necessità di ulteriori collegamenti dal PC o PLC.

Attualmente la versione 1.1 del modulo TSIO può comunicare solo in One-wire per cui lo potete utilizzare solo con la scheda TCOM e motoriduttori in versione One-wire. Esiste anche una versione 1.2, utilizzabile sui sistemi che comunicano in RS485 ma verrà prodotta solo su richiesta.



Tutti i dettagli per comunicare con lo TSIO sono riassunti a questo link:

https://www.theremino.com/wp-content/uploads/files/Theremino_Motors_Help_ENG.pdf

Registri motoriduttore e scheda TSIO

Le tabelle seguenti mostrano i parametri da usare per il modulo di input e output TSIO che è compatibile con la linea di comunicazione del motoriduttore.

TSIO v.xx		EGD1 v.xx	
SerialNumber,	000-003	SerialNumber,	000-003
DeviceVer,	003-001	DeviceVer,	003-001
DeviceSubVer,	004-001	DeviceSubVer,	004-001
FirmwareVer,	005-001	FirmwareVer,	005-001
FirmwareSubver,	006-001	FirmwareSubver,	006-001
BootLoaderVer,	007-001	BootLoaderVer,	007-001
ADC1,	010-002	ActualPosition,	010-004
ADC2,	012-002		
ADC3,	014-002	ActualTorque,	014-002
ADC4,	016-002	ActualAccel,	016-002
ADC5,	018-002	ActualVelocity,	018-002
ADC6,	020-002	ActualCurrent,	020-002
ADC7,	022-002	ActualVoltage,	022-002
ADC8,	024-002	ActualTemp,	024-001
		MovingStatus,	025-001
DIN1,	026-002	HwErrorStatus,	026-001
		IdSetHallSensor,	027-002
DIN2,	028-002	HallPhase (1..6),	029-001
ActualTemp,	030-002	MagSens1_A,	030-002
IdSetHallSensor,	032-002	MagSens1_X,	032-002
		MagSens1_Y,	034-002
		MagSens2_A,	036-002
		MagSens2_X,	038-002
		MagSens2_Y,	040-002
.....			
WriteProtect,	050-001	WriteProtect,	050-001
WorkingMode,	051-001	WorkingMode,	051-001
ReplayMode,	052-001	ReplayMode,	052-001
BaudRateIndex,	054-001	BaudRateIndex,	054-001
DeviceId,	055-001	DeviceId,	055-001
IdTolerance,	056-001	IdTolerance,	056-001
PWM1,	058-002	GoalPosition,	058-004
PWM2,	060-002		
PWM3,	062-002	GoalTorque,	062-002
PWM4,	064-002	GoalAccel,	064-002
PWM5,	066-002	GoalVelocity,	066-002
PWM6,	068-002	PositionPGain,	068-001
		PositionIGain,	069-001
PWM7,	070-002	PositionDGain,	070-001
		DeadBand,	071-001
PWM8,	072-002	ZeroSpeedCurrent,	072-001
PWM9,	074-002		
DOUT1,	076-002		
DOUT2,	078-002		
		MagSens1_GainAxis,	080-001
		MagSens1_Gain,	081-001
		MagSens1_OffsetX,	082-001
		MagSens1_OffsetY,	083-001
		MagSens2_GainAxis,	084-001
		MagSens2_Gain,	085-001
		MagSens2_OffsetX,	086-001
		MagSens2_OffsetY,	087-001
		MagSens_WorkMode,	088-001
SerialPassword,	100-016	SerialPassword,	100-016

Schema di collegamento nel caso si utilizzi il solo PLC



Questo schema è elettricamente il più semplice, ma richiederà molto lavoro di programmazione sul PLC

Il programma sul PLC dovrà:

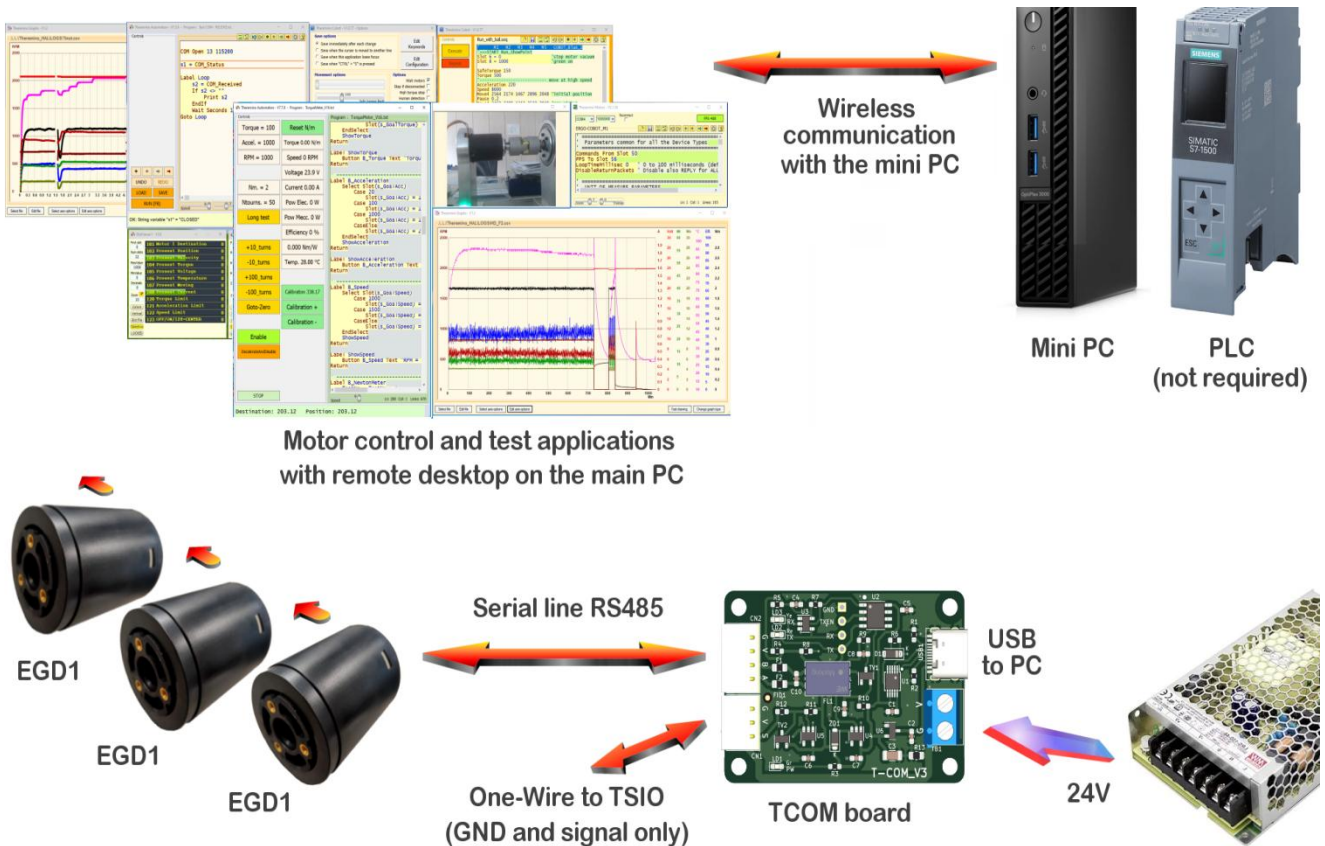
- Implementare il protocollo di comunicazione.
- Implementare i comandi per leggere e scrivere i registri dei motori.
- Implementare le funzioni di controllo, ad esempio per dare gli ID ai motori.
- Gestire la comunicazione seriale con i motori.
- Inviare la destinazione ai motori, traducendola da gradi a unità del motore.
- Leggere la posizione raggiunta e tradurla in gradi.
- Leggere la temperatura e gli altri parametri di ogni motore e tenerne conto.
- Gestire gli errori, eseguire le azioni necessarie e eventualmente avvertire gli operatori.
- Tenere un LOG delle operazioni effettuate.

Le informazioni per scrivere il programma sono tutte disponibili in questo documento e in caso di necessità potreste anche utilizzare parti dei codici OpenSource che abbiamo scritto per le applicazioni sul PC.

Ad esempio, invece che riscrivere la funzione che crea il CRC basterebbe copiarla e adattarla al linguaggio usato nel PLC. E lo stesso si potrebbe fare con tutte le altre funzioni.

In ogni caso scrivere e collaudare questo programma potrebbe impegnare un programmatore esperto per un minimo di alcuni mesi. E questo tempo potrebbe allungarsi anche di molto per progetti complessi.

Schema di collegamento nel caso si utilizzi un PC



Consigliamo di utilizzare un PC nella sala di controllo e collegarlo in wireless al Mini PC, che non avrà quindi bisogno di tastiera e schermo.

Il PLC che si vede nella immagine non è realmente richiesto ma potrebbe essere utile a chi è abituato a utilizzarli per controllare i segnali di Input Output (microswitch, relè ecc..). Il PLC potrà poi comunicare con il PC in vari modi, ad esempio con stringhe di testo trasmesse via seriale e gestite dalla nostra applicazione Automation, oppure con altri metodi che dipendono dal particolare tipo di PLC usato, ad esempio API predisposte per controllarlo.

Se utilizzerete il modulo TSIO V1.1 fate attenzione a non inviargli il 24 volt. Collegate alla scheda TCOM solo il GND e il segnale. La alimentazione dello TSIO dovrà essere fornita a parte con un alimentatore da 5 a 12 V, oppure con uno StepDown che abbassa il 24 volt a 5 volt.

In tutti i casi, cioè anche con versioni di TSIO che sopportano il 24V, utilizzare uno StepDown da 5V permetterebbe di fornire più corrente (anche 1 o 2 ampere) alle uscite che si controllano, mentre con il solo regolatore interno allo TSIO la corrente massima è di circa 100 mA totali.

Interfaccia di collegamento

