

# Cos'è AxesBrain™ ?

E' un pacchetto software che trasforma il PC in un sistema di controllo

Permette di realizzare un'UNITA' DI CONTROLLO per l'automazione con un'opportunità di scelta vasta in termini di:

- 1 Motion control
- 2 Input/Output
- 2 Prestazioni e programmabilità

Unendo tutte queste caratteristiche in un'unica soluzione, AxesBrain™ è un software dall'ARCHITETTURA APERTA e conforme agli standard di mercato

# Quali sono le sue origini?

Deriva da un linguaggio di programmazione sviluppato in Olivetti

Robotica Olivetti  
Anni 1974-85  
Linguaggio- MultiTask-  
Anticollisione

Sistemi Olivetti  
Anni 1984-89  
Transfert-Celle-  
FMS

PC Olivetti  
Anni 1981-89  
Sistema RealTime

CNC Olivetti  
Anni 1960-80  
Algoritmi GTL  
(CNC-FresaTornio)

AxesBrain™  
VisAlgo™

Ricerca Olivetti  
Anni 1984-87  
Sistema visione

# 1974 l'inizio....

Periodico di cultura economia tecnica  
Anno III - N. 4  
Dicembre 1973  
Lire 150  
Sped. in abb. post. 02/7374

# GO

informazioni

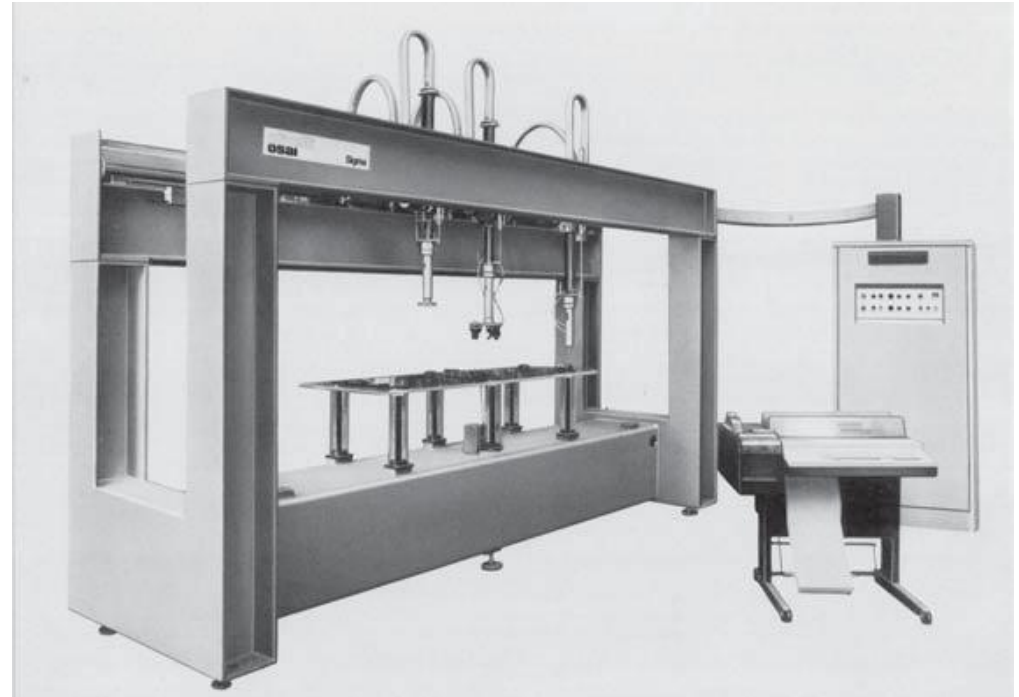
**Cronaca di un anno nero**  
I maggiori avvenimenti economici internazionali e italiani del 1973 visti e analizzati mese per mese: questo è il contenuto del «Taccuino economico» di Bruno Lomborghini che pubblichiamo a pag. 3

**Il gioco chimico della vita**  
Vasche di acquicoltura, cupole geodetiche, superserie in cui si producono pesci utilizzando «siente» altro che il sole e il vento: non è fantascienza, ma una realtà operante negli Stati Uniti (a pag.8)

A che punto siamo con gli automi  
**Ecco Sigma, robot intelligente se c'è un errore lo corregge**  
Automi sempre più sofisticati vengono utilizzati in alcune lavorazioni industriali negli Stati Uniti, in Giappone e in Europa. Uno dei più «intelligenti» lavora in Italia. Su questo tema pubblichiamo in questa pagina una nota del prof. Luciano Gallino e a pagina 4 un articolo del prof. Gian Federico Micheletti ed un servizio di Ermanno Franchetto



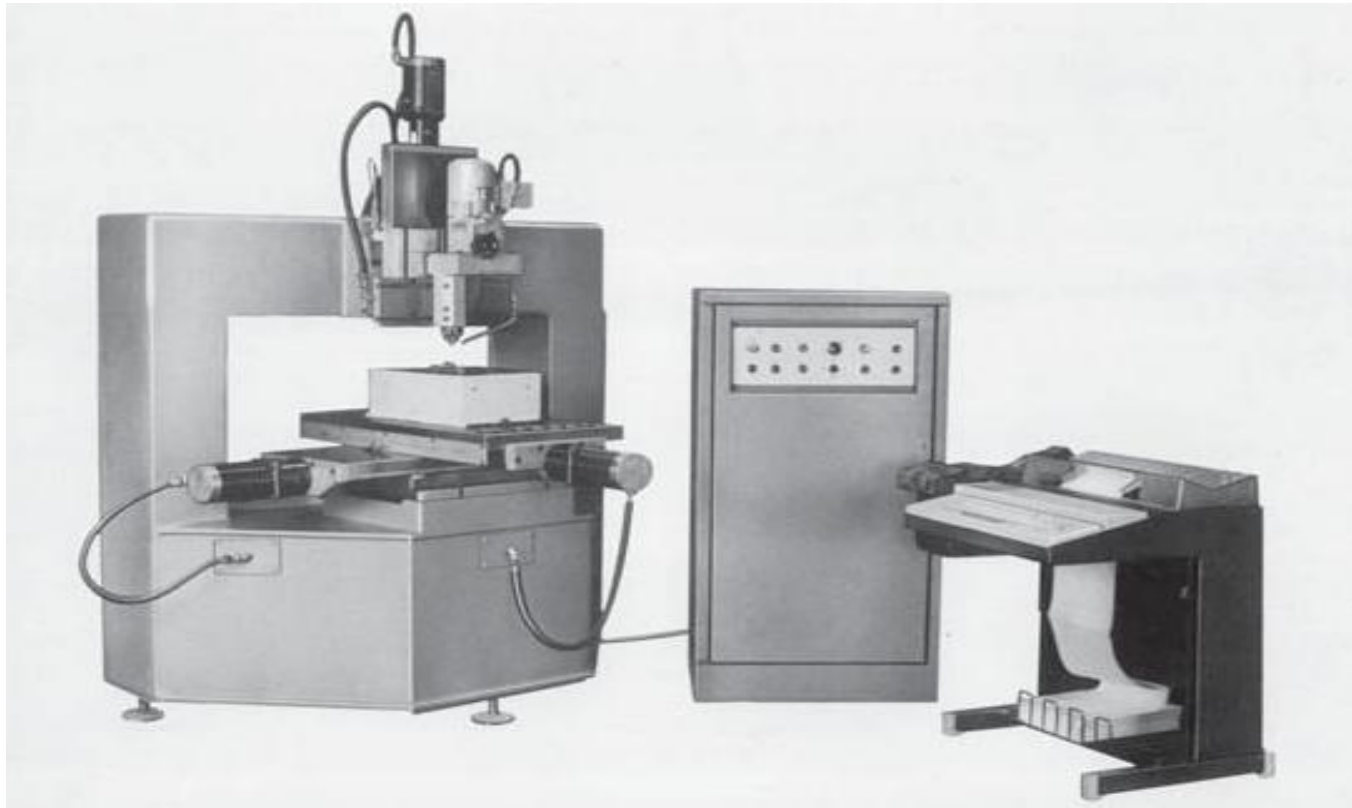
*I robot e la qualità del lavoro*



1974 robotica Olivetti

Multi Arms = Multi Tasks

# La prima microfresa....



1975 ISO G-Code per microfresatura

# Dove siamo

L'AB&T è ad IVREA

in provincia di Torino 50 Km, nei pressi di Milano 100 km, vicino a Genova 200 km

Famosa per l'Olivetti ed il carnevale ( la battaglia delle arance )



# Sensori, Attuatori e Risorse

I/O digitali , fisici e virtuali

I/O analogici

Assi fisici e virtuali

Mandrini

# I/O digitali fisici e virtuali

The screenshot displays the AxesBrainStation software interface. At the top, the title bar reads "AxesBrainStation" and the menu bar includes "File", "Tastiera", and "?". Below the menu bar is a status bar with the text "PLC running..." and a "Feed 100%" indicator. The main area is divided into two tables: "Input" and "Output".

Num...	Nome Input	Descrizione
114	I14	Input 14
115	I15	Input 15
116	I16	Input 16
117	I17	Input 17
118	I18	Input 18
119	I19	Input 19
120	I20	Input 20
121	I21	Input 21
122	I22	Input 22
123	I23	Input 23
124	I24	Input 24
125	I25	Input 25
126	I26	Input 26
127	I27	Input 27
128	I28	Input 28
129	I29	Input 29
130	I30	Input 30
131	I31	Input 31
132	I32	Input 32
401	OKMovAxi...	OK to move X1 axis
402	OKMovAxi...	OK to move Y1 axis
403	OKMovAxi...	OK to move Z1 axis
404	OKMovAxi...	OK to move X2 axis
405	OKMovAxi...	OK to move Y2 axis
406	OKMovAxi...	OK to move Z2 axis

Num...	Nome Output	Descrizione
117	O17	Output 17
118	O18	Output 18
119	O19	Output 19
120	O20	Output 20
121	O21	Output 21
122	O22	Output 22
123	O23	Output 23
124	O24	Output 24
125	O25	Output 25
126	O26	Output 26
127	O27	Output 27
128	O28	Output 28
129	O29	Output 29
130	O30	Output 30
131	O31	Output 31
132	O32	Output 32
301	M1.1	Merker M1
302	M1.2	Merker M1
303	M1.3	Merker M1
304	M1.4	Merker M1
305	M1.5	Merker M1
306	M1.6	Merker M1
307	M1.7	Merker M1
308	M1.8	Merker M1
309	M1.9	Merker M1

At the bottom of the interface, there is a row of function keys (F1-F12) with labels: F1 Input, F2 Output, F3, F4, F5, F6, F7, F8 Reset, F9 ServoON, F10 Emerg., F11, F12. Below this is a status bar with "Servo On", "Hold", "Run" buttons, and numerical data: "Speed = 100%", "94", "1.057", "0.047", "18/07/2011", "10:18:13". On the right side, there is a vertical toolbar with icons for "Sistema", "Manuale", "Assi e mandrini", "IO Digitali", "IO Analogici", "Apprendimento punti", "Taratura Assi", "Diagramma assi", "Diagramma", "Automazione", "CNC", and "Visione".



# I/O DIGITALI per il controllo degli assi

The screenshot displays the AxesBrainStation software interface, which is used for controlling a CNC system. The main window is titled "AxesBrainStation" and shows a "PLC running..." status. The interface is divided into several sections:

- System Status:** Shows "PLC running..." and "Feed 100%".
- Input Table:** A table listing digital inputs (I1-I7) and their descriptions (e.g., Home X1, Driver Ready Axis X1, PROBE X1).
- Output Table:** A table listing digital outputs (O1-O7) and their descriptions (e.g., Servo ON Axis X1, Servo ON Axis Y1, Servo ON Axis Z1, Servo ON Axis X2, Servo ON Axis Y2, Servo ON Axis Z2, Direction Axis Step 1-6, Output 1-7).
- Right Panel:** A vertical toolbar with icons for "Sistema", "Manuale", "Assi e mandrini", "IO Digitali", "IO Analogici", "Apprendimento punti", "Taratura Assi", "Diagramma assi", and "Diagramma".
- Bottom Panel:** A row of function keys (F1-F12) labeled "Input", "Output", "Reset", "ServoON", "Emerg.", and "Visione".
- Bottom Status Bar:** Shows "Servo On", "Hold", "Run", "Speed = 100%", "133", "1.090", "0.045", "18/07/2011", and "10:19:33".

Num...	Nome Input	Descrizione
1	HomeX1	Home X1
2	DrvOKX1	Driver Ready Axis X1
3	ProbeX1	PROBE X1
4	HomeY1	Home Y1
5	DrvOKY1	Driver Ready Axis Y1
6	ProbeY1	PROBE Y1
7	HomeZ1	Home Z1
8	DrvOKZ1	Driver Ready Axis Z1
9	ProbeZ1	PROBE Z1
10	HomeX2	Home X2
11	DrvOKX2	Driver Ready Axis X2
12	ProbeX2	PROBE X2
13	HomeY2	Home Y2
14	DrvOKY2	Driver Ready Axis Y2
15	ProbeY2	PROBE Y2
16	HomeZ2	Home Z2
17	DrvOKZ2	Driver Ready Axis Z2
18	ProbeZ2	PROBE Z2
101	I1	Input 1
102	I2	Input 2
103	I3	Input 3
104	I4	Input 4
105	I5	Input 5
106	I6	Input 6
107	I7	Input 7

Num...	Nome Output	Descrizione
1	EnableX1	Servo ON Axis X1
2	EnableY1	Servo ON Axis Y1
3	EnableZ1	Servo ON Axis Z1
4	EnableX2	Servo ON Axis X2
5	EnableY2	Servo ON Axis Y2
6	EnableZ2	Servo ON Axis Z2
25	EnableStep1	Servo ON Axis Step 1
26	EnableStep2	Servo ON Axis Step 2
27	EnableStep3	Servo ON Axis Step 3
28	EnableStep4	Servo ON Axis Step 4
29	EnableStep5	Servo ON Axis Step 5
30	EnableStep6	Servo ON Axis Step 6
31	DirStep1	Direction Axis Step 1
32	DirStep2	Direction Axis Step 2
33	DirStep3	Direction Axis Step 3
34	DirStep4	Direction Axis Step 4
35	DirStep5	Direction Axis Step 5
36	DirStep6	Direction Axis Step 6
101	O1	Output 1
102	O2	Output 2
103	O3	Output 3
104	O4	Output 4
105	O5	Output 5
106	O6	Output 6
107	O7	Output 7



# I/O digitali

AB&A  
Tecnologie Industriali

PLC running... Feed 100%

Num...	Nome Input	Descrizione
101	I1	Input 1
102	I2	Input 2
103	I3	Input 3
104	I4	Input 4
105	I5	Input 5
106	I6	Input 6
107	I7	Input 7
108	I8	Input 8
109	I9	Input 9
110	I10	Input 10
111	I11	Input 11
112	I12	Input 12
113	I13	Input 13
114	I14	Input 14
115	I15	Input 15
116	I16	Input 16
117	I17	Input 17
118	I18	Input 18
119	I19	Input 19
120	I20	Input 20
121	I21	Input 21
122	I22	Input 22
123	I23	Input 23
124	I24	Input 24
125	I25	Input 25

Num...	Nome Output	Descrizione
101	O1	Output 1
102	O2	Output 2
103	O3	Output 3
104	O4	Output 4
105	O5	Output 5
106	O6	Output 6
107	O7	Output 7
108	O8	Output 8
109	O9	Output 9
110	O10	Output 10
111	O11	Output 11
112	O12	Output 12
113	O13	Output 13
114	O14	Output 14
115	O15	Output 15
116	O16	Output 16
117	O17	Output 17
118	O18	Output 18
119	O19	Output 19
120	O20	Output 20
121	O21	Output 21
122	O22	Output 22
123	O23	Output 23
124	O24	Output 24
125	O25	Output 25

Sistema

- Manuale
- Assi e mandrini
- IO Digitali
- IO Analogici
- Apprendimento punti
- Taratura Assi
- Diagramma assi
- Diagramma

Automazione

- CNC
- Visione

F1 Input F2 Output F3 F4 F5 F6 F7 F8 Reset F9 ServoON F10 Emerg. F11 F12

Servo On Hold Run Speed = 100% 226 1.110 0.042 18/07/2011 10:20:57

# I/O digitali virtuali

The screenshot displays the AxesBrainStation software interface. The main window is titled "AxesBrainStation" and features a menu bar with "File" and "Tastiera ?". Below the menu bar is a logo for "AVATA" (Tecnologie Informatiche) and a "Feed 100%" indicator.

The interface is divided into two main panels, each with a table of I/O configurations:

Num...	Nome Input	Descrizione
401	<input type="radio"/> OKMovAxi...	OK to move X1 axis
402	<input type="radio"/> OKMovAxi...	OK to move Y1 axis
403	<input type="radio"/> OKMovAxi...	OK to move Z1 axis
404	<input type="radio"/> OKMovAxi...	OK to move X2 axis
405	<input type="radio"/> OKMovAxi...	OK to move Y2 axis
406	<input type="radio"/> OKMovAxi...	OK to move Z2 axis
600	<input type="radio"/> LAM_SK1_1	Menu 1 - Soft Key 1
601	<input type="radio"/> LAM_SK2_1	Menu 1 - Soft Key 2
602	<input type="radio"/> LAM_SK3_1	Menu 1 - Soft Key 3
603	<input type="radio"/> LAM_SK4_1	Menu 1 - Soft Key 4
604	<input type="radio"/> LAM_SK5_1	Menu 1 - Soft Key 5
605	<input type="radio"/> LAM_SK6_1	Menu 1 - Soft Key 6
606	<input type="radio"/> LAM_SK7_1	Menu 1 - Soft Key 7
607	<input type="radio"/> LAM_SK8_1	Menu 1 - Soft Key 8
608	<input type="radio"/> LAM_SK1_2	Menu 2 - Soft Key 1
609	<input type="radio"/> LAM_SK2_2	Menu 2 - Soft Key 2
610	<input type="radio"/> LAM_SK3_2	Menu 2 - Soft Key 3
611	<input type="radio"/> LAM_SK4_2	Menu 2 - Soft Key 4
612	<input type="radio"/> LAM_SK5_2	Menu 2 - Soft Key 5
613	<input type="radio"/> LAM_SK6_2	Menu 2 - Soft Key 6
614	<input type="radio"/> LAM_SK7_2	Menu 2 - Soft Key 7
615	<input type="radio"/> LAM_SK8_2	Menu 2 - Soft Key 8
616	<input type="radio"/> LAM_SK1_3	Menu 3 - Soft Key 1
617	<input type="radio"/> LAM_SK2_3	Menu 3 - Soft Key 2
618	<input type="radio"/> LAM_SK3_3	Menu 3 - Soft Key 3

Num...	Nome Output	Descrizione
301	<input type="radio"/> M1.1	Merker M1
302	<input type="radio"/> M1.2	Merker M1
303	<input type="radio"/> M1.3	Merker M1
304	<input type="radio"/> M1.4	Merker M1
305	<input type="radio"/> M1.5	Merker M1
306	<input type="radio"/> M1.6	Merker M1
307	<input type="radio"/> M1.7	Merker M1
308	<input type="radio"/> M1.8	Merker M1
309	<input type="radio"/> M1.9	Merker M1
310	<input type="radio"/> M1.10	Merker M1
311	<input type="radio"/> M1.11	Merker M1
312	<input type="radio"/> M1.12	Merker M1
313	<input type="radio"/> M1.13	Merker M1
314	<input type="radio"/> M1.14	Merker M1
315	<input type="radio"/> M1.15	Merker M1
316	<input type="radio"/> M1.16	Merker M1
317	<input type="radio"/> M2.1	Merker M2
318	<input type="radio"/> M2.2	Merker M2
319	<input type="radio"/> M2.3	Merker M2
320	<input type="radio"/> M2.4	Merker M2
321	<input type="radio"/> M2.5	Merker M2
322	<input type="radio"/> M2.6	Merker M2
323	<input type="radio"/> M2.7	Merker M2
324	<input type="radio"/> M2.8	Merker M2
325	<input type="radio"/> M2.9	Merker M2

At the bottom of the interface, there is a row of function keys: F1 (Input), F2 (Output), F3, F4, F5, F6, F7, F8 (Reset), F9 (ServoON), F10 (Emerg.), F11, and F12. Below these keys are status indicators for "Servo On", "Hold", and "Run".

The right sidebar contains a vertical menu with icons and labels: "Sistema", "Manuale", "Assi e mandrini", "IO Digitali", "IO Analogici", "Apprendimento punti", "Taratura Assi", "Diagramma assi", "Diagramma", "Automazione", "CNC", and "Visione".

The bottom status bar displays: "Speed = 100% 240 1.164 0.044 18/07/2011 10:21:53".

# I/O Analogici

The screenshot displays the AxesBrainStation software interface. At the top, the title bar reads "AxesBrainStation" and the menu bar includes "File", "Tastiera", and "?". Below the menu bar is a logo for "A&B&T" with the text "Tecnologie Integrate". To the right of the logo is a "Feed 100%" indicator with a green bar.

The main area contains two data tables. The left table, titled "Analog Input", has a light green background and lists three inputs:

Num...	Nome Input	Descrizione	Valore
1	ADC1	Analog Input ADC 1	0
2	ADC2	Analog Input ADC 2	0
3	ADC3	Analog Input ADC 3	0

The right table, titled "Analog Output", has a light red background and lists three outputs:

Num...	Nome Out...	Descrizione	Valore
1	DAC1	Analog Output DAC 1	0
2	DAC2	Analog Output DAC 2	0
3	DAC3	Analog Output DAC 3	0

On the right side of the interface is a vertical toolbar with icons for various functions: "Sistema", "Manuale", "Assi e mandrini", "IO Digitali", "IO Analogici", "Apprendimento punti", "Taratura Assi", "Diagramma assi", "Diagramma", and "Automazione". Below the toolbar are buttons for "CNC" and "Visione".

At the bottom of the interface is a control panel with function keys F1 through F12. F1 is labeled "Set Output", F8 is "Reset", F9 is "ServoON", and F10 is "Emerg.". Below these keys are three buttons: "Servo On", "Hold", and "Run". The status bar at the very bottom shows "Speed = 100%", "304", "1.272", "0.063", "18/07/2011", and "10:23:20".

# Assi fisici e mandrini

AxisBrainStation

File Tastiera ?

PLC running... Feed 100%

Num...	Nome Asse	Descrizione	Pos.Reale	Pos.Teorica	Volt	Velocità Max
1	X1	Axis X1	0.000	0.000	0.000	7700.000
2	Y1	Axis Y1	0.000	0.000	0.000	7700.000
3	Z1	Axis Z1	0.000	0.000	0.000	7700.000
4	X2	Axis X2	0.000	0.000	0.000	7700.000
5	Y2	Axis Y2	0.000	0.000	0.000	7700.000
6	Z2	Axis Z2	0.000	0.000	0.000	7700.000
12	Step1	Axis Step 1	0.000	0.000	0.000	500.000
13	Step2	Axis Step 2	0.000	0.000	0.000	500.000
14	Step3	Axis Step 3	0.000	0.000	0.000	500.000
15	Step4	Axis Step 4	0.000	0.000	0.000	500.000
16	Step5	Axis Step 5	0.000	0.000	0.000	500.000
20	Step6	Axis Step 6	0.000	0.000	0.000	500.000

Num...	Nome Mand...	Descrizione	RPM Reali	RPM Teorici	Volt	RPM Max
1	S1	Spindle 1	0.000	0.000	0.000	2400.000
2	S2	Spindle 2	0.000	0.000	0.000	2400.000

Sistema

Manuale

Assi e mandrini

IO Digitali

IO Analogici

Apprendimento punti

Taratura Assi

Diagramma assi

Diagramma

Automazione

CNC

Visione

F1 F2 F3 F4 F5 F6 F7 F8 Reset F9 ServoON F10 Emerg. F11 F12

Servo On Hold Run Speed = 100% 182 1.005 0.054 18/07/2011 10:30:23

# Assi fisici, virtuali e mandrini

The screenshot displays the AxesBrainStation software interface. At the top, there is a menu bar with 'File' and 'Tastiera'. Below the menu bar is the ABB logo and a 'Feed 100%' indicator. The main area contains two tables. The first table lists physical axes (X1, Y1, Z1, X2, Y2, Z2, Step1-6, Xs, Ys, Xp, Yp) with columns for 'Num...', 'Nome Asse', 'Descrizione', 'Pos.Reale', 'Pos.Teorica', 'Volt', and 'Velocità Max'. The second table lists spindles (S1, S2) with columns for 'Num...', 'Nome Mand...', 'Descrizione', 'RPM Reali', 'RPM Teorici', 'Volt', and 'RPM Max'. At the bottom, there is a control panel with function keys F1-F12, including 'Reset', 'ServoON', and 'Emerg.'. The status bar at the very bottom shows 'Servo On', 'Hold', 'Run', 'Speed = 100%', and various numerical values.

Num...	Nome Asse	Descrizione	Pos.Reale	Pos.Teorica	Volt	Velocità Max
1	X1	Axis X1	0.000	0.000	0.000	7700.000
2	Y1	Axis Y1	0.000	0.000	0.000	7700.000
3	Z1	Axis Z1	0.000	0.000	0.000	7700.000
4	X2	Axis X2	0.000	0.000	0.000	7700.000
5	Y2	Axis Y2	0.000	0.000	0.000	7700.000
6	Z2	Axis Z2	0.000	0.000	0.000	7700.000
12	Step1	Axis Step 1	0.000	0.000	0.000	500.000
13	Step2	Axis Step 2	0.000	0.000	0.000	500.000
14	Step3	Axis Step 3	0.000	0.000	0.000	500.000
15	Step4	Axis Step 4	0.000	0.000	0.000	500.000
16	Step5	Axis Step 5	0.000	0.000	0.000	500.000
20	Step6	Axis Step 6	0.000	0.000	0.000	500.000
13	Xs		0.000	0.000	0.000	500.000
14	Ys		0.000	0.000	0.000	500.000
15	Xp		0.000	0.000	0.000	500.000
16	Yp		0.000	0.000	0.000	500.000

Num...	Nome Mand...	Descrizione	RPM Reali	RPM Teorici	Volt	RPM Max
1	S1	Spindle 1	0.000	0.000	0.000	2400.000
2	S2	Spindle 2	0.000	0.000	0.000	2400.000

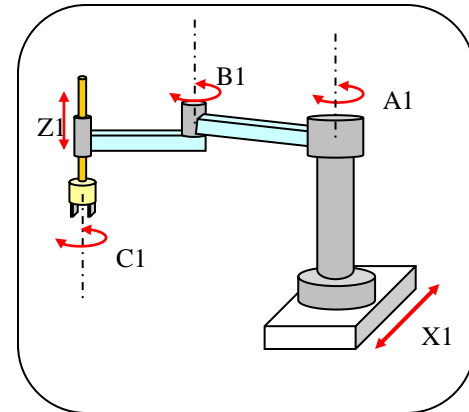
Speed = 100% 0 1.005 0.076 18/07/2011 10:32:40

# Asse virtuale SCARA

13	Xs		0.000	0.000	0.000	500.000
14	Ys		0.000	0.000	0.000	500.000

```

[Xs]
i=VIRTUAL_AXES_NAME( "Xs" , "Ys" )
b={
Description="SCARA"
Kind_Axis="ARM"
Kind_Axis_0=0x00010000           // Type SCARA Kind_Axis_1=0x00000000
Axis_name_ascisse="Xs"
Axis_name_ordinate="Ys"
Axis_name_1="Step1"           // A1 AXIS ANGLE ARM
Axis_name_2="Step2"           // B1 AXIS ANGLE FOREARM
Axis_name_3="Step3"           // C1 AXIS ROTARY (PULSE)
Axis_name_4=""                 // X1 ADDITIONAL AXIS MOTION
Lenght_arm_1=200.0
Lenght_arm_2=200.0
Abs_offset_A=100.0
Vel_max_axis_A=20000.0
Acc_max_axis_A=200.0
Dec_max_axis_A=200.0
Abs_offset_O=100.0
Vel_max_axis_O=20000.0
Acc_max_axis_O=1000.0
Dec_max_axis_O=1000.0
e=}
    
```



# Asse virtuale cilindrico

15	Xp		0.000	0.000	0.000	500.000
16	Yp		0.000	0.000	0.000	500.000

```

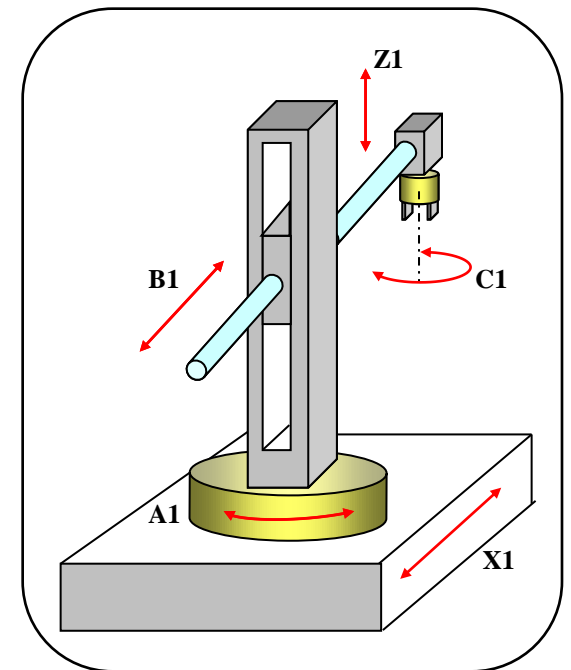
[Xp]
i=VIRTUAL_AXES_NAME( "Xp" , "Yp" )
b={
Kind_Axis="ARM"

Kind_Axis_0=0x00080000 // CYLINDRICAL SLEEVE
Kind_Axis_1=0x00000000
Axis_name_ascisse="Xc"
Axis_name_ordinate="Yc"

Axis_name_1="Step4" // A1 AXIS ANGLE ARM
Axis_name_2="Step5" // B1 AXIS LINEAR (SLEEVE)
Axis_name_3="Step6" // C1 AXIS ROTARY (PULSE)
Axis_name_4="" // X1 ADDITIONAL AXIS MOTION

Lenght_arm_1=000.0
Lenght_arm_2=200.0
Abs_offset_A=150.0
Vel_max_axis_A=50000.0
Acc_max_axis_A=1000.0
Dec_max_axis_A=1000.0
Abs_offset_O=100.0
Vel_max_axis_O=50000.0
Acc_max_axis_O=1000.0
Dec_max_axis_O=1000.0
e=}

```





# Sistema manuale movimentazione assi



# Taratura asse +/-10Volt PID

Il valore analogico +/-Volt è il riferimento di velocità, l'errore di posizionamento è utilizzato per la regolazione PID di retroazione.

- 1) Proporzionale errore
- 2) Integrativo errore
- 3) Derivativo errore

$$\text{Volt} = K_c * ( P * \text{errore} + I * \text{Somma degli errori} + D * \text{Variazione degli errori} )$$

# Taratura del PID

File Tastiera ?

PLC running... Feed 100%

Asse

X1

Offset

Vff 100.000 P 20.000

Aff 0.000 I 0.000 Er. 3.500  Mov

D 0.000

D2 0.000

Asse Riferimento

Y1  Abilita

Offset

Vff 100.000 P 20.000

Aff 0.000 I 0.000 Er. 3.500  Mov

D 0.000

D2 0.000

Errore  Errore 0.000

Tensione  Tensione 0.000

Campionatura

F1 F2 F3 Window clear F4 Attiva F5 Ripistina F6 Salva F7 Leggi F8 Reset F9 ServoON F10 Emerg F11 F12

Servo On Hold Run Speed = 100% 246 0.928 0.048 18/07/2011 10:39:50

Sistema

Manuale

Assi e mandrini

IO Digitali

IO Analogici

Apprendimento punti

Taratura Assi

Diagramma assi

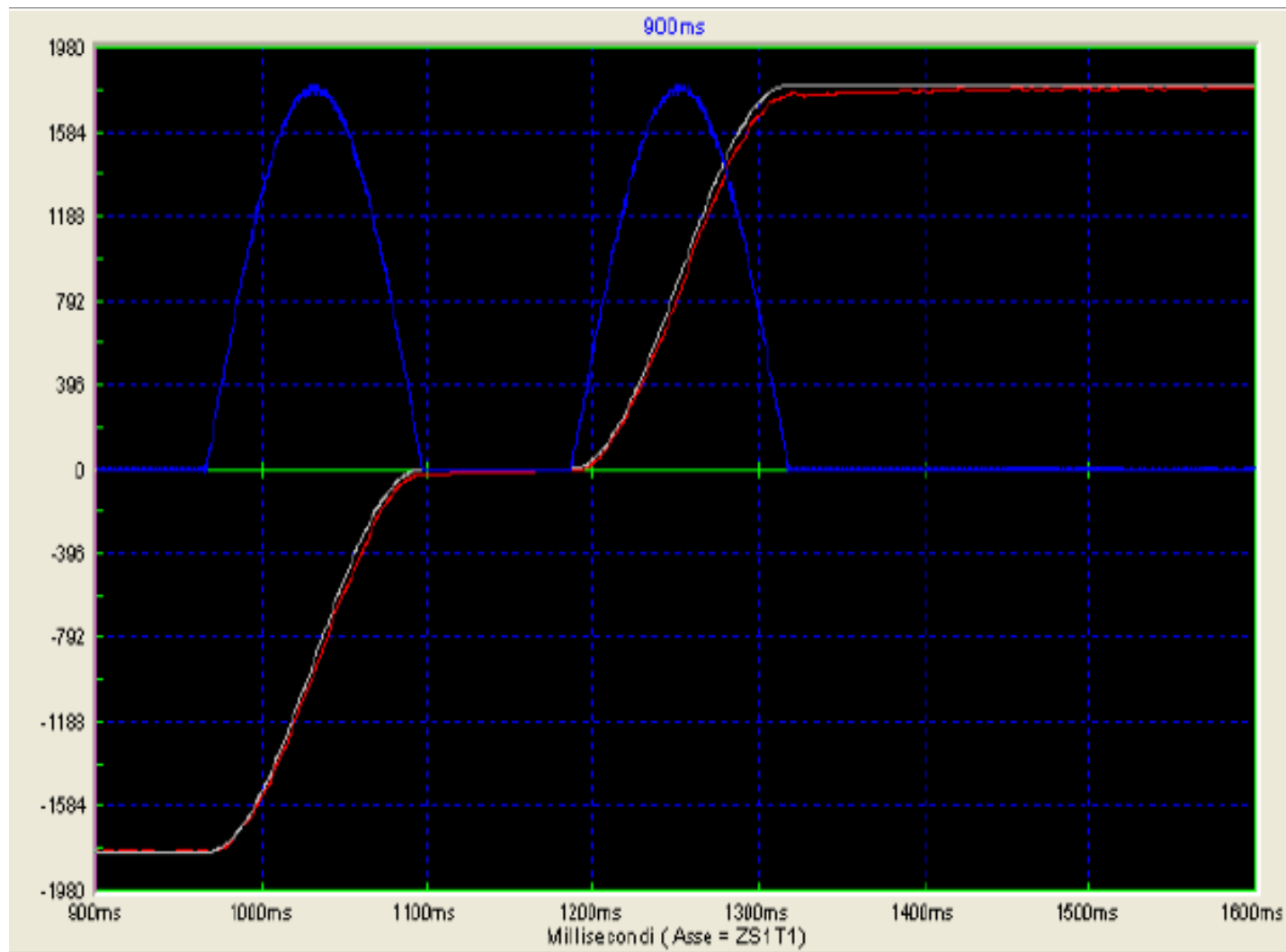
Diagramma

Automazione

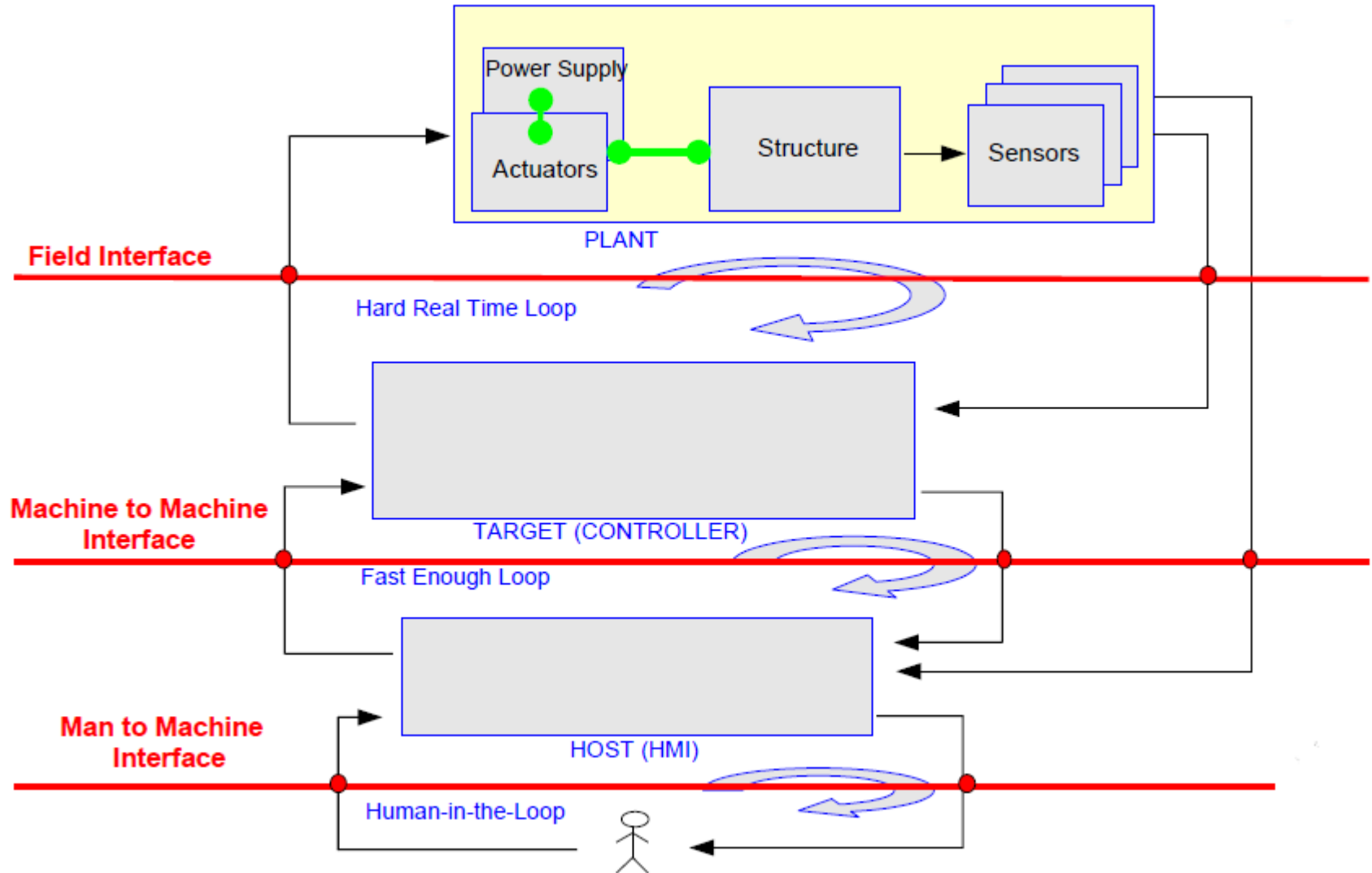
CNC

Visione

# Diagramma taratura asse



# Struttura meccatronica



# Tipologie di aperture

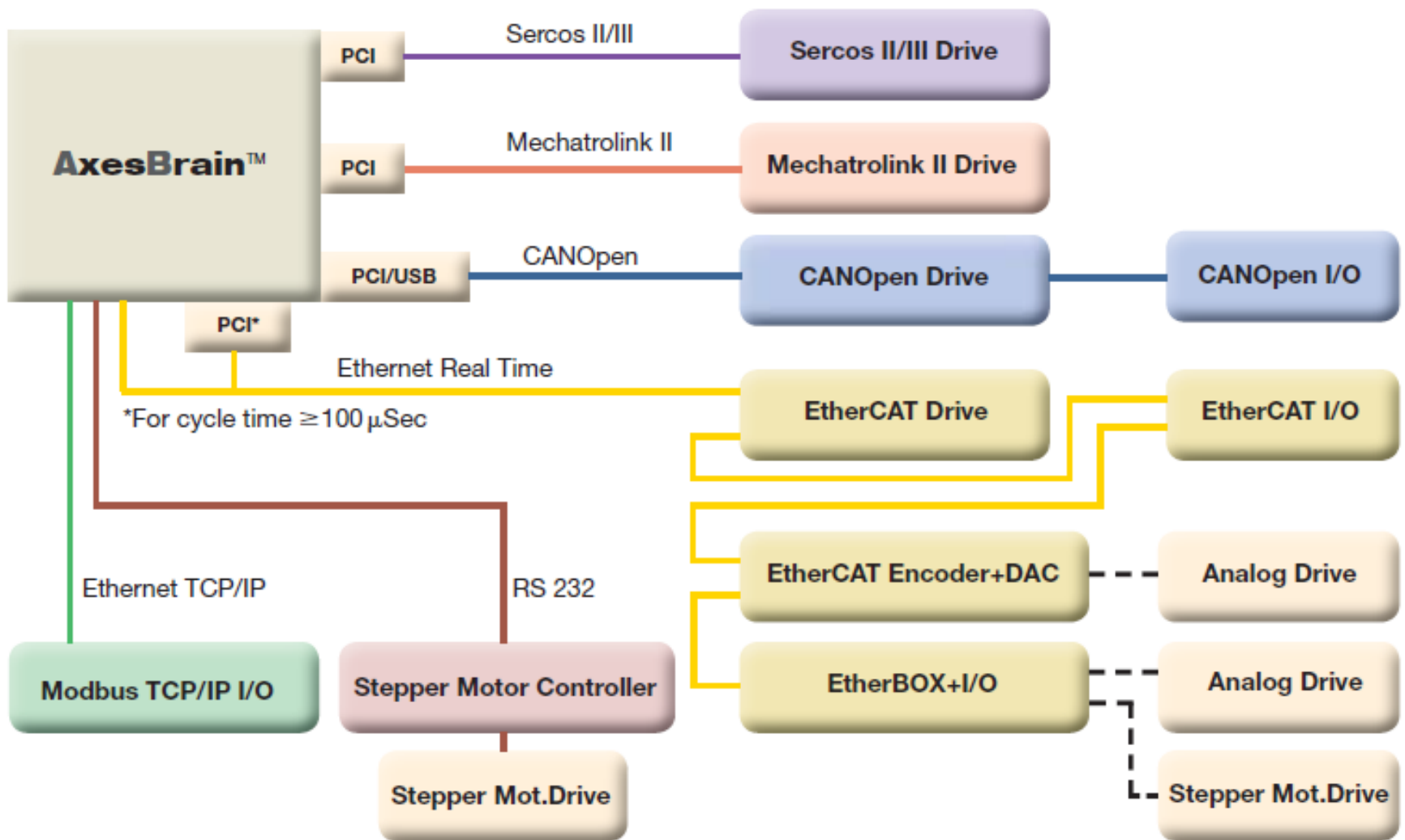
Apertura verso il basso:

Possibilità di utilizzare i più comuni Bus di campo e azionamenti sia analogici che digitali (Fieldbus interface)

Apertura verso l'alto:

possibilità di utilizzare linguaggi ad alto livello per crearsi la propria interfaccia utente (Man to machine interface)

# Aperture verso il basso





# Aperture verso l'alto

Il **Distributed Component Object Model** (noto con l'acronimo **DCOM**, inglese per *Modello ad oggetti per componenti distribuiti*) è una tecnologia informatica presentata nel 1996 da Microsoft.

DCOM permette di effettuare chiamate di procedure remote attraverso una rete, occupandosi di tutte le mediazioni necessarie, in maniera indipendente dal linguaggio, presenta dei servizi specifici all'applicativo in essere, che possono essere usati da applicazioni "Clienti" sullo stesso PC o su PC remoti.

Si possono quindi utilizzare linguaggi come VC++, VB, C#, .net, ecc... per crearsi la propria HMI

# AxesBrain™

## Una sola mente, tante attività simultanee

**AxesBrain™** è un componente software per il MotionControl, con funzionalità Multitask per l'automazione e Multiprocesso per il CNC

L'arma vincente di **AxesBrain™** è la sua capacità di gestire più processi, più assi e più attività in parallelo.

Si possono infatti eseguire fino a 32 processi CNC ISO simultanei e fino a 1024 attività GP-PLC contemporaneamente.

Queste sono funzionalità particolarmente utili per Transfer, macchine con carichi-scarichi automatici e macchine speciali in genere, dove sono richiesti processi di lavoro paralleli.

**AxesBrain™** è un controllo progettato per connettersi a tutti i maggiori bus di campo per la movimentazione degli assi e la gestione degli I/O.

Permette inoltre di collegarsi in modo *digitale*, tramite ethernet, ad azionamenti *analogici*.

Questa flessibilità offre una vasta possibilità di scelta che garantisce anche un abbassamento dei costi.

# AxesBrain™ Caratteristiche

- CNC iso ( fresa, tornio )
- 1024 Task GPL
- 32 Processi CNC iso
- Trasformazione assi nel piano
- 64 assi ( 16 assi interpolati )
- 4096 I/O
- Compensazione lineare, quadratura, matrice
- Gantry
- Anticollisione
- Camme Elettroniche

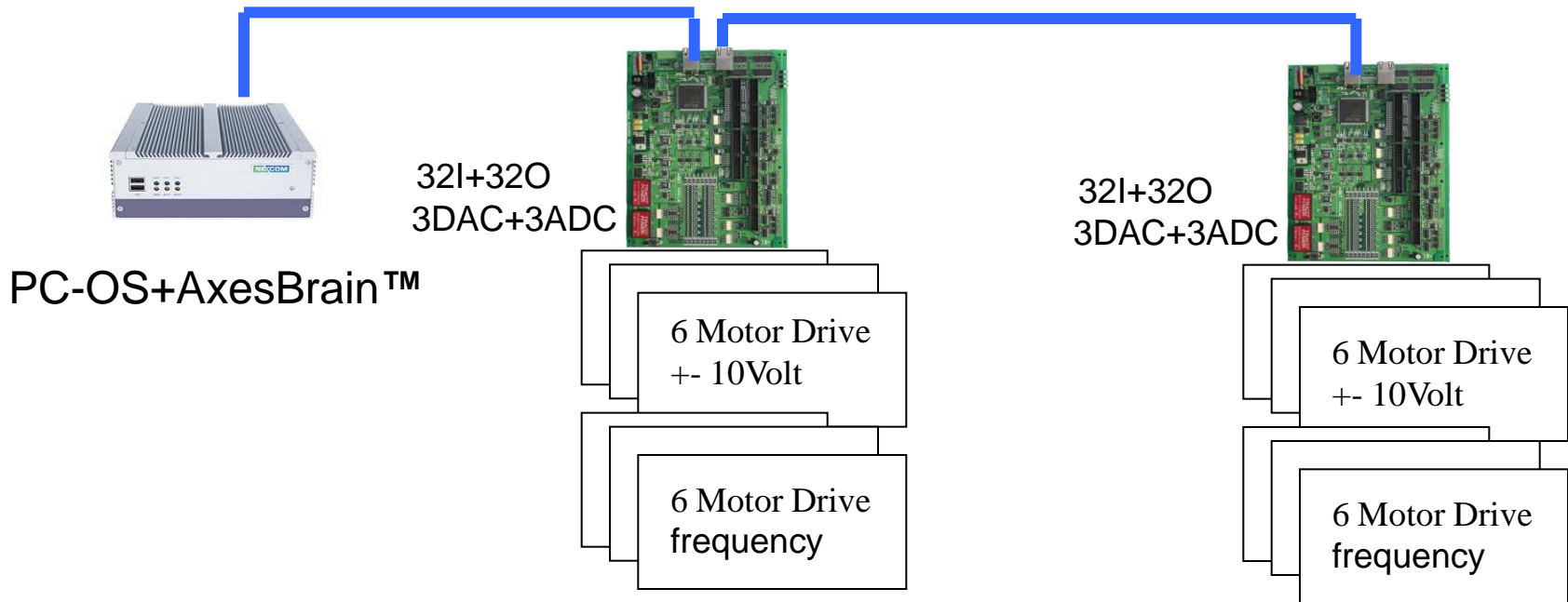
# La tecnologia Ethernet RealTime

La tecnologia si basa su alcuni punti:

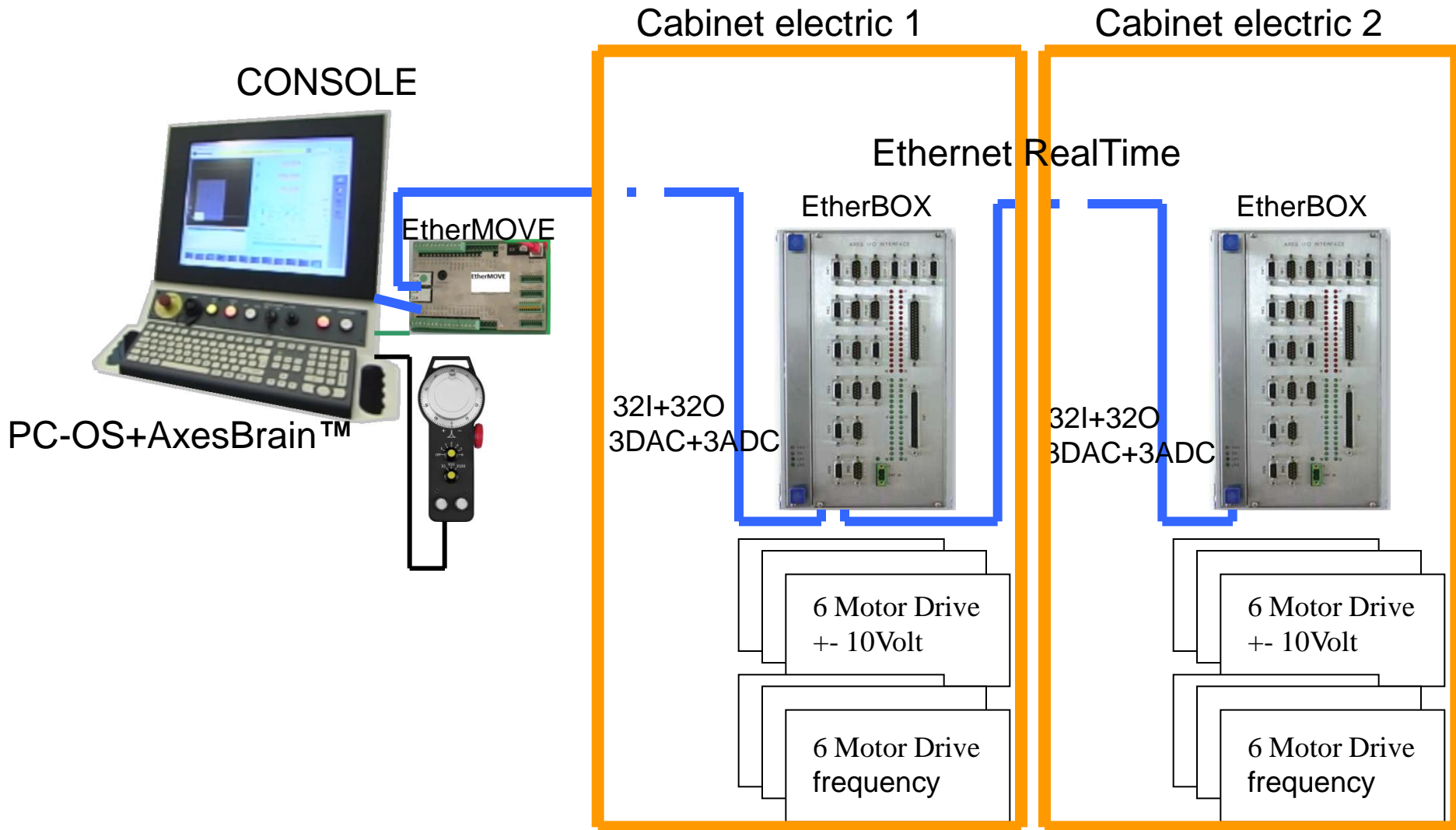
- 1 ) Per il cablaggio si usa il cavo Ethernet
- 2 ) Il Master ( normalmente un PC ) **trasmette e riceve** il FRAME utilizzando una porta Ethernet “standard” il concetto di FRAME è il medesimo di quando si trasmette e riceve delle informazione sulla rete internet tradizionale
- 3) I dispositivi Slave non ricevono e ritrasmettono il Frame allo Slave successivo, bensì lo vedono **transitare**.
- 4) Infine il FRAME ritorna al Master che lo riceve completo dei dati di tutti gli Slave.

# Come unire il mondo dell'automazione tradizionale al mondo Ethernet RealTime

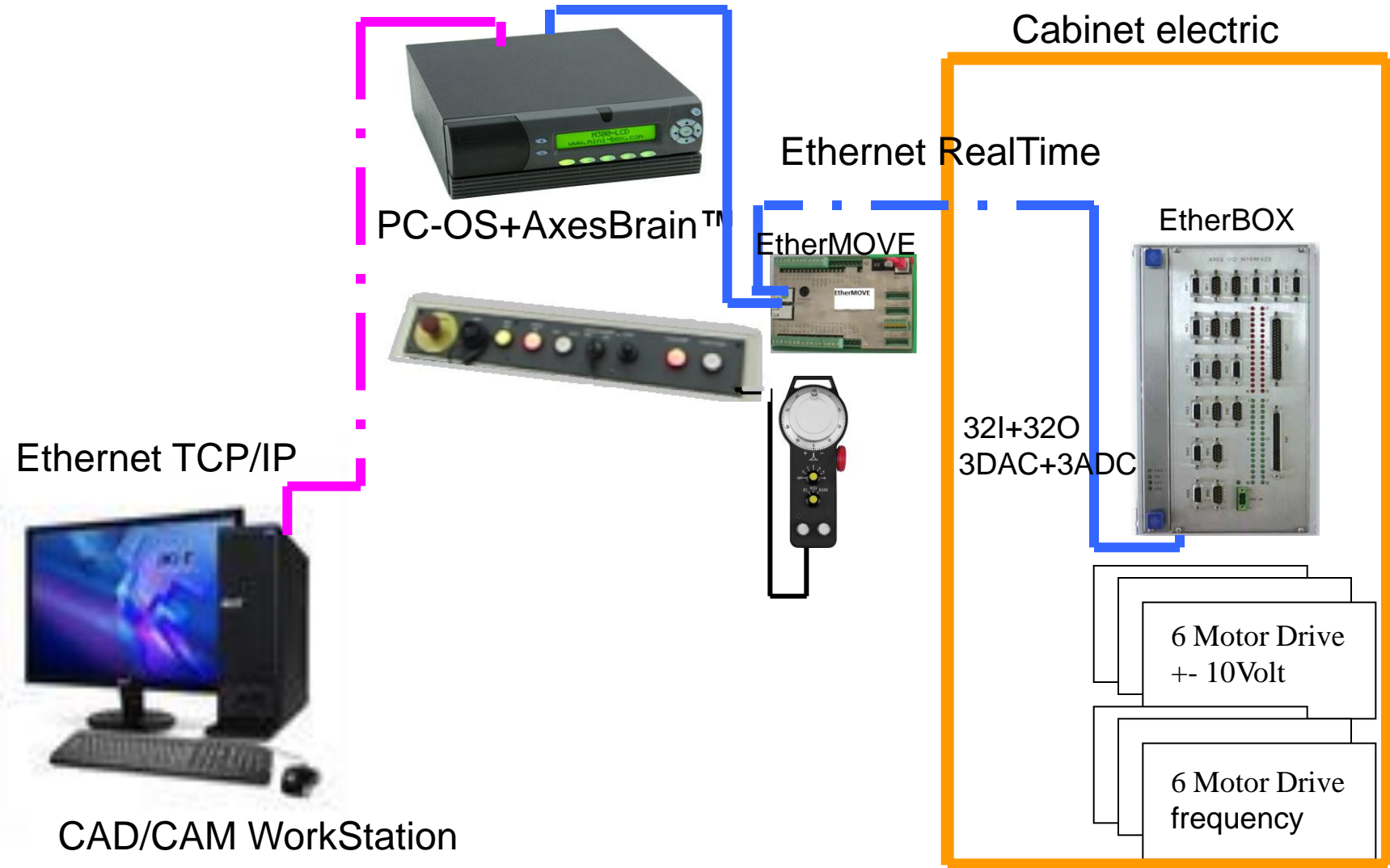
Ethernet RealTime  
with Board EtherBOX



# PanelPC + Ethernet RealTime

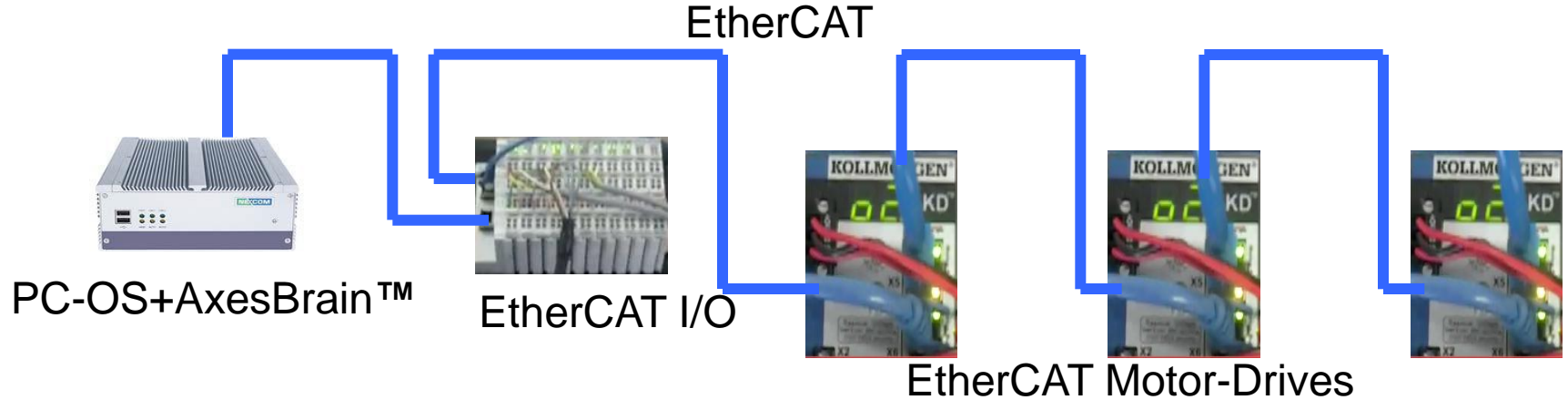


# WorkStation + AxesBrain™ + Ethernet RealTime

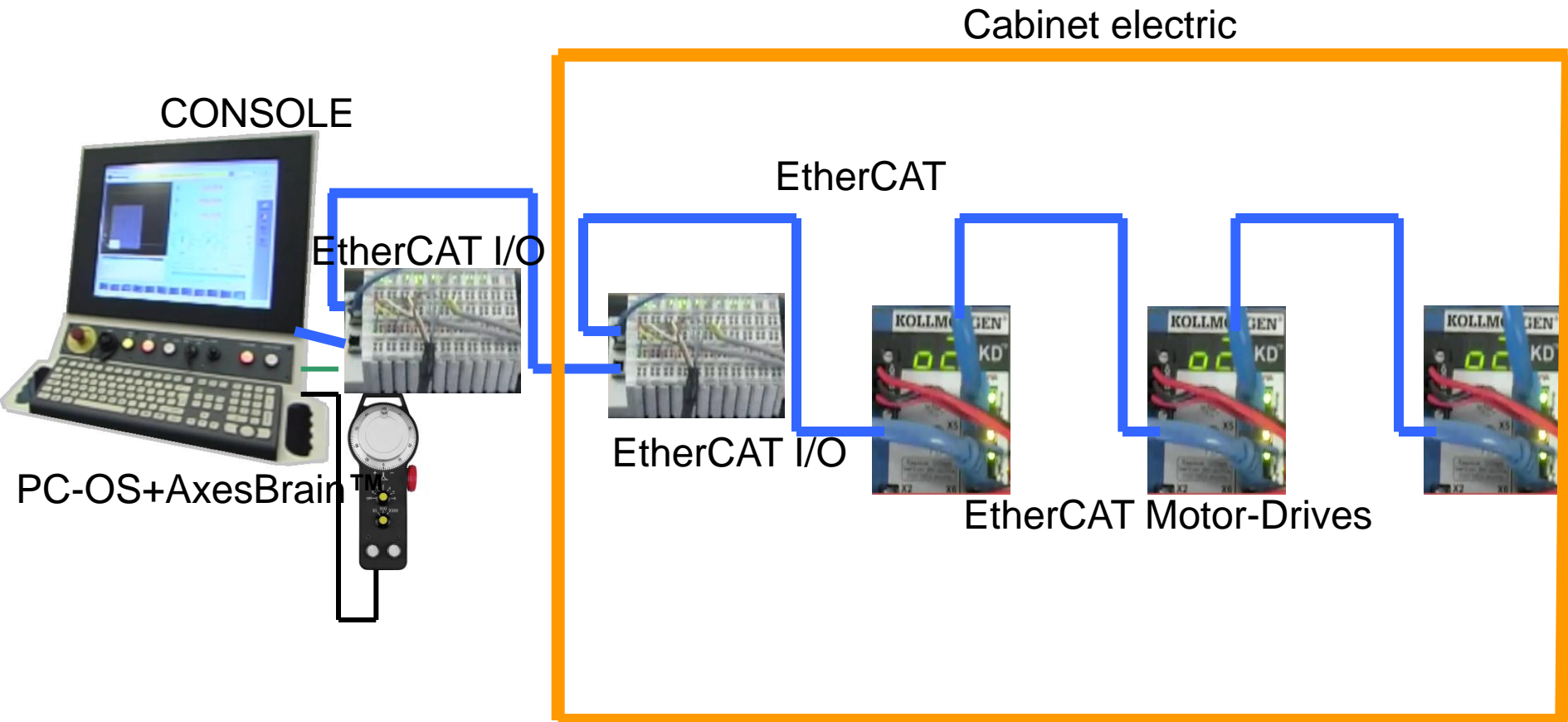




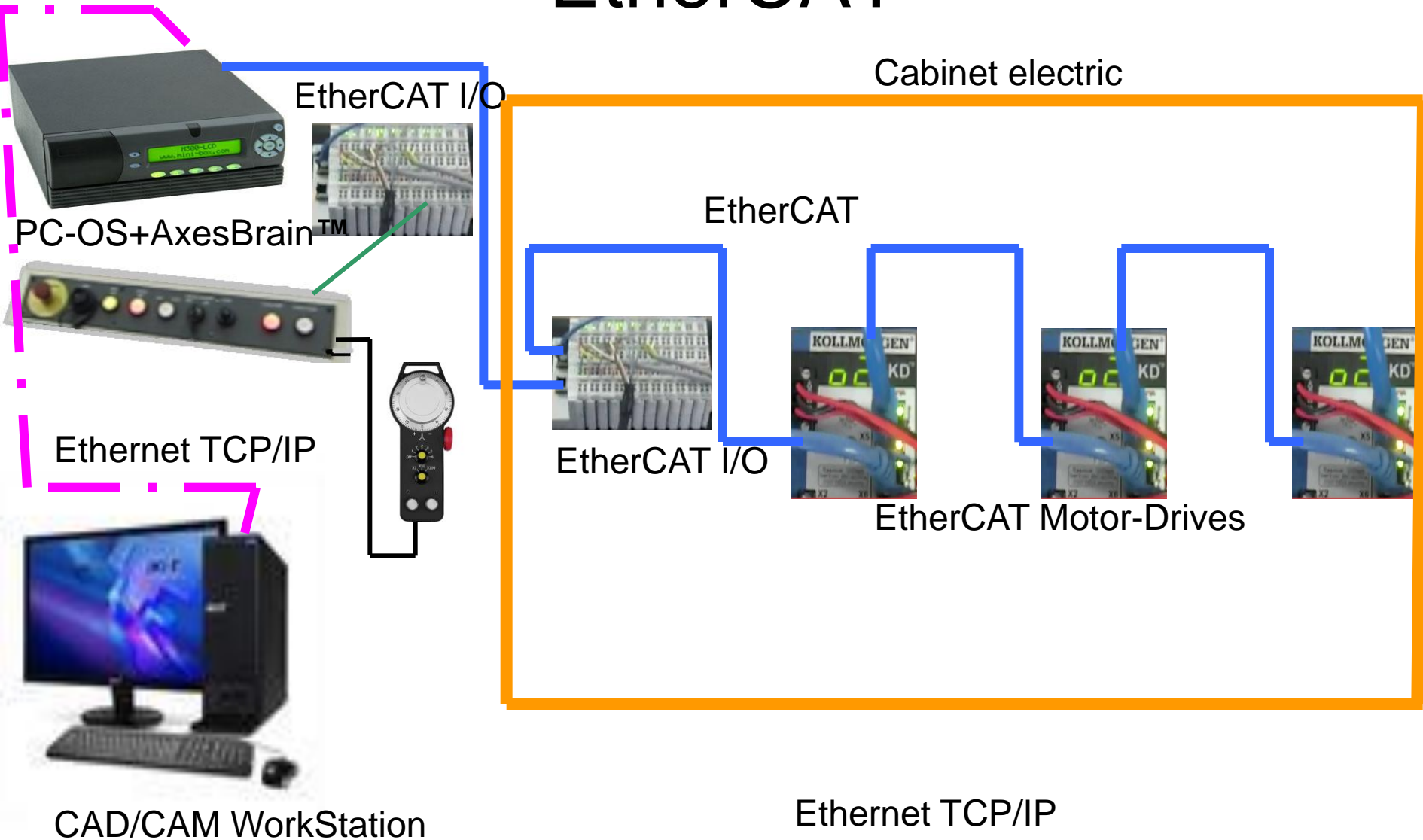
# EtherCAT comunicazione standard Ethernet RealTime



# PanelPC + EtherCAT



# WorkStation + AxesBrain™ + EtherCAT



# LINGUAGGI DI PROGRAMMAZIONE

Esistono **due tipologie di linguaggi**:

- Quelli orientati alla programmazione della macchina

PLC (Programmable Logic Controller) dei quali ne esistono di vari tipi, alcuni standard (IEC 61131-3), altri proprietari del costruttore, servono per poter programmare le funzionalità della macchina.

Questo linguaggio deve essere conosciuto del costruttore o dall'applicatore

- Quelli orientati alla programmazione della lavorazione del pezzo

ISO CNC anche detto G code ha una parte comune a tutti i controlli G0, G1, G2, G3, G4 + una parte specifica per ogni costruttore dove risiede la parametrica, le funzioni di test e subroutine, i cicli fissi, le macro, ecc...

Questo linguaggio deve essere conosciuto dall'operatore della macchina.

# Linguaggio GP-PLC

E un linguaggio proprietario di tipo IL (Instruction List), multitasking, orientato alla movimentazione degli assi.

Di tipo AWL per la gestione PLC degli I/O, possibilità di integrare le funzioni di visione di **VisAlgo™** .

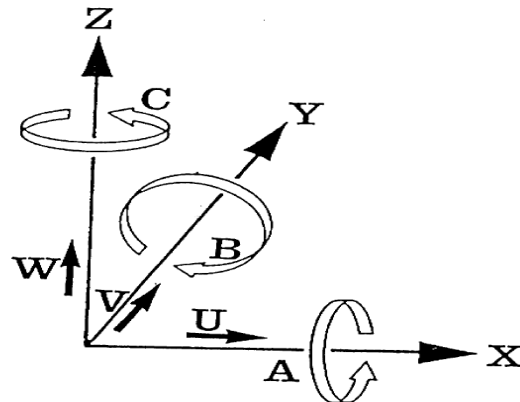
Ha la gestione degli eventi per la sincronizzazione dei task

Prevede la gestione di messaggi per l'interfaccia HMI

# Linguaggio ISO-CNC G-Code

ISO CNC anche detto G code ha una parte comune a tutti i controlli G0, G1, G2, G3, G4 + una parte specifica per ogni costruttore dove risiede la parametrica, le funzioni di test e subroutine, i cicli fissi, le macro, ecc... Questo linguaggio è utilizzato dall'operatore della macchina, per programmare la lavorazione del pezzo definendo l'utensile, il percorso, i cicli che la macchina deve compiere nelle varie fasi.

Il nome degli assi che identificano il percorso sono: X Y Z A B C U V W



# Videocamere

Le videocamere nel mondo dell'automazione rappresentano delle soluzioni ideali per alcune problematiche, vediamole in dettaglio.

## 1) Aumento della precisione della macchina attraverso due accorgimenti

L'accorgimento numero uno è di compensare il piano di lavoro attraverso una griglia conosciuta, rilevata tramite videocamera su un piano di lavoro campione

L'accorgimento numero due è di montare una o più videocamera su ogni macchina, per acquisire due o più riferimenti su pezzo posizionato sull'attrezzatura, in modo da conoscere la reale roto traslazione da effettuare sul programma di lavorazione.

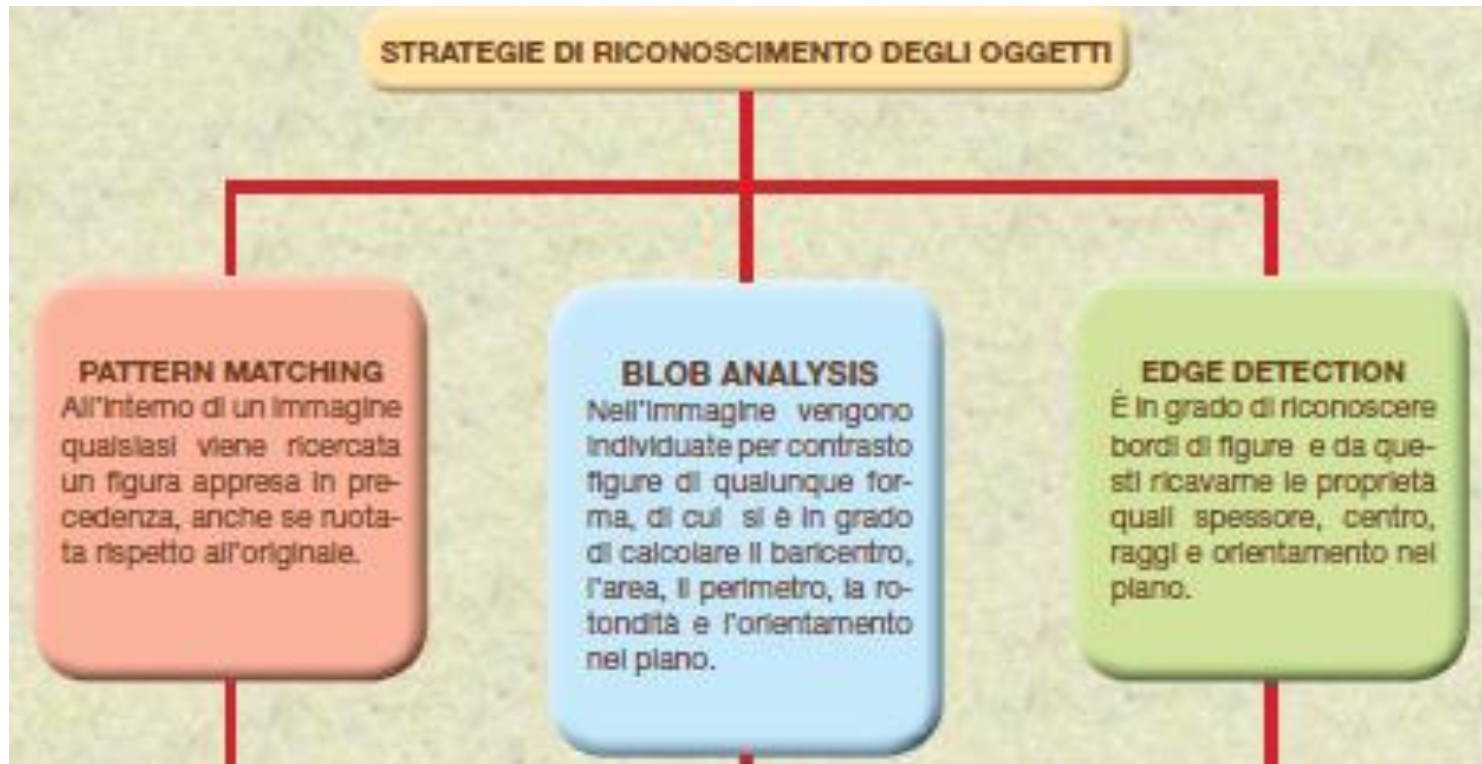
## 2) Controllo e riconoscimento a monte di parti prima della lavorazione

## 3) Controllo a valle delle parti dopo la lavorazione



# Sistema di visione VisAlgo™

**VisAlgo™** è una libreria di funzioni per il riconoscimento e l'elaborazione delle immagini



# Integrazione con la visione

Acquisizione dell'immagine

Rettangoli di pixels

640 x 480 768 x 574 1240 x 1000 ... ..

Definizione di una ROI

Determinare i BLOB

Estrarre le funzionalità geometriche

Estrarre le figure simili ad una campione (Pattern-matching)

Estrarre le caratteristiche di posizione nel piano e l'angolo di rotazione

Pixels / mm

La programmazione delle funzionalità di "VisAlgo" è realizzata con –ARI (GP-PLC) or con una libreria DLL

# Linguaggio di programmazione per l'automazione GP-PLC

*AxesBrain è un linguaggio di programmazione è indicato per la definizione dei cicli di lavoro per l'automazione.*

Per descrivere il ciclo di lavoro di un manipolatore o di una macchina automatica è necessario avere un linguaggio in grado di apprendere come evolvono le fasi.

Esistono numerosi linguaggi di programmazione, per lo più sono “proprietary” dell'azienda. Le specifiche che un linguaggio deve avere sono molteplici, programmazione in fasi parallele, sincronizzazione tra le attività programmate, alta integrazione con i dispositivi esterni (visione, laser, eccetera).

Nato dalla specifica del SIGLA ( SIGMA LANGUAGE) 1974 dell'Olivetti, come uno dei primi linguaggi di programmazione delle macchine speciali per l'assemblaggio di parti, è stato ampliato in modo da soddisfare le moderne esigenze d'integrazione e flessibilità. Riferimento a :“Robot technology at Olivetti: the sigma system” Olivetti, Milan 1976.

Una sua prerogativa oltre la semplicità di sintassi è la capacità di avere la multiprogrammazione dei singoli cicli, prerogativa indispensabile per poter effettuare delle operazioni di assemblaggio e di manipolazioni di parti.

# La struttura di GP-PLC

Linguaggio sequenziale  
( Newman )

Linguaggio logico  
( Bool )

Procedure ( CALL )  
e  
Process ( Task )

Local variables ( L )  
e  
Global variables ( G )

# Come viene eseguito il primo GP-PLC part program

Quando AxesBrainServer viene eseguito sul PC anche un GP-PLC program è caricato ed eseguito, il nome è trovato in "SISTEMA.TXT" in DAT directory.

Nel segmento: [ParametriAUTOMAZIONE]

e parametro : NomePartProgramLancio=

Esempioe

[ParametriAUTOMAZIONE]

StringaLancio=

NomePartProgramLancio=LogicaMacc.PP

In questo caso LogicaMacc.PP è caricato ed eseguito :

Loop           -TMM/100  
                  -NET/1  
                  -LDN/T1  
                  -TON/2,1000

-NET/2  
-LD/T2  
-TON/1,1000

-NET/3  
-LD/T2  
-PEX/-DIS/11,10:PLC running...  
-NOT  
-PEX/-DIS/11,10:  
-JMP/Loop  
-RET

# Architettura GP-PLC

*L'architettura del sotto sistema di automazione è quella di poter eseguire i cicli di lavoro in parallelo tra loro sincronizzandoli con degli eventi.*

Ogni programma ha al proprio interno un area proprietaria dove sono allocate le variabili LOCAL ,su cui tra l'altro vengono depositati i parametri di chiamata. Quando al programma viene inserito in esecuzione, da un comando esplicito o richiamato da un CALL o da un'istruzione TSK, il suo codice sorgente viene caricato in memoria ed automaticamente precompilato, in memoria vi rimarrà fino ad un comando esplicito di cancellazione o comando al sottosistema di RESET.

I programmi vengono abbinati ad un HANDLER di lavoro proprietario su cui si appenderanno tutte le attività del ciclo, possiamo aver un numero pressoché infinito di HANDLER, con questa modalità si possono avere configurazioni d'impianto estremamente flessibili, come linee di ROBOTS, macchine speciali multi testa, macchine con carichi scarichi integrati , macchine multifunzionali, etc.

# Sintassi del linguaggio GP-PLC

## Definizioni del linguaggio

*La sintassi del linguaggio è estremamente semplice un triletterale preceduto da un trattino “-“ rappresenta l’istruzione, uno “/” rappresenta il delimitatore dai parametri dell’istruzione che possono essere:*

- 1 Riferimenti diretti a variabili GLOBALI o LOCALI
- 2 Riferimenti indiretti a variabili GLOBALI
- 3 Espressioni numeriche con riferimenti alle variabili GLOBALI o LOCALI
- 4 Espressioni matematiche con riferimenti alle variabili GLOBALI o LOCALI
- 5 LABEL di salto
- 6 Nomi di risorse assi , mandrini, input e output

I parametri sono separati da virgole.

Le etichette o LABEL per i salti condizionati o incondizionati sono messe prima del separatore trattino “-“ dell’istruzione.

I commenti sono preceduti da carattere punto e virgola “;”

[ label] –XXX/[parametro 1],...[parametro n] [; questo è un commento]

Si possono commentare più righe usando “/\*” all’inizio dell’area commentata e “\*/” alla fine dell’area

# Esempio di sintassi GP-PLC

-LET/L1,500.089 ; Carica il valore 500.089 nella Locale L1  
-LET/G100,0 ; Carica il valore 0 nella Globale G100  
BeginCount- ; LABEL BeginCount  
-LET/G100,G100+11.23 ; Somma il valore 11.23 to the variabile G100  
-JLT/G100,L1,BeginCount ; Continua a sommare finchè il valore  
; della G100 non maggiore o uguale al valore  
; indicato in L1 (500.089)

; Usiamo ora un metodo indiretto per indicare la globale G 100

-LET/L2,100 ; Carica il valore 100 nella locale L2  
-LET/G(L2),0 ; Carica il valore 0 nella globale G100  
BeginCB- ; LABEL BeginCB  
-LET/G(L2),G(L2)+11.23 ; Somma il valore 11.23 to the variabile G100  
-JLT/G(L2),L1,BeginCB ; Continua a sommare finchè il valore  
; della G100 non maggiore o uguale al valore  
; indicato in L1 (500.089)



# Operatori matematici

- + somma
- sottrazione
- / divisione
- \* moltiplicazione
- ^ elevato
- ( parentesi aperta
- ) parentesi chiusa

# Funzioni delle espressioni matematiche 1 parte

abs	Assoluto di un numero
acos	Arco coseno
and	And tra due numeri
asin	Arco seno
atan	Arco tangente
atanw	Arco tangente di Y,X
ceil	Arrotondamento verso l'alto di un numero decimale in un numero intero
cos	Coseno
cosh	Coseno iperbolico
deg	Trasformazione in gradi di un angolo espresso in radianti
exp	Esponenziale
floor	Arrotondamento verso il basso di un numero decimale in un numero intero
logd	Logaritmo decimale
logn	Logaritmo naturale
lshift	Shift verso sinistra di un numero
max	Massimo fra enne espressioni
min	Minimo fra enne espressioni

# Funzioni delle espressioni matematiche 2 parte

mod	Modulo fra due numeri
not	Negazione booleana di un numero
pi	P greco
rad	Trasformazione in radianti di un angolo espresso in gradi
rshift	Shift verso destra di un numero
sin	Seno
sinh	Seno iperbolico
sqr	Radice quadrata
tan	Tangente
tanh	Tangente iperbolica
xor	Or esclusivo fra due numeri

# Esempio di espressione matematica

```
-LET/L1, max (sin (rad (G1 +12), cos (rad (+12 G1 * L1/56)))  
; The local variable is loaded the result of L1 expression:  
; max (sin (rad (G1 +12), cos (rad (+12 G1 * L1/56)))
```

## Nota

In numeri in rappresentazione esadecimale sono preceduti da 0x

## Esempio:

```
-LET/L1, 0x10 ; 0x10 = 16 decimale
```

L'utilizzo dell'operatore esadecimale "0x" è particolarmente utile nella mascheratura con le funzioni di "and" e "or", per poi usarli successivamente

## Esempio:

```
-LET/L1, and (L1, 0x8000)  
-JEQ/L1, 0x8000, Bit8000Uno
```

# Dimensioni massima dei parametri richiamabili

GLOBAL 32,767

LOCALI configurate nella voce NumeroLocali= del file di configurazione  
"SISTEMA.TXT" ( ampliabili da -DIM/Numero LOCALI )

WATCH 16

DRT 6

DIS 17 linee

Columns DIS 128

NETwork 128 per istanza

Timer (T) 128 per istanza

Counter (C) 128 per istanza

# Variabili globali e locali

*Per poter effettuare delle operazioni logiche, leggere valori numerici sono necessarie le variabili.*

Il sotto sistema AXESBRAIN prevede due tipi di variabili :

LOCALI

GLOBALI

Ogni programma all'atto dell'entrata in funzione si alloca un numero di variabili pari a quello configurate nel sistema, vengono tutte azzerate e sono a disposizione delle istruzioni di quel programma, le prime variabili vengono impostate con i parametri di chiamata del comando di esecuzione, le variabile rimarranno in memoria a disposizione per le operazioni di interrogazioni o visualizzazione.

# Locali

Il numero di LOCALI può essere cambiato dal quello previsto nel “**SISTEMA.TXT**” alla voce “**NumeroLocali=**”, utilizzando l’istruzione – **DIM**/numero locali, che rimane definito solo per quella istanza di task. Un caso particolare sono i programmi richiamati dalle istruzioni di “-CAL” e di “-TSK” con parametri, in questi casi le prime variabili LOCALI sono caricate con i parametri di chiamate nello stesso ordine posizionale, se esistono delle LOCALI nei parametri di chiamate verranno caricate con il nuovo valore al ritorno del programma chiamato.

# Globali

Oltre le variabili LOCALI nel sotto sistema di automazione sono previste **32767** variabili **GLOBALI** che vengono richiamate con la lettera “G” ed il numero della stessa. Tutte le **GLOBALI** vengono salvate su HardDisk, che quindi possono essere usate per memorizzare dati persistenti.

Le Globali sono salvate su HardDisk all’uscita del sistema o nella procedura di “SHUT DOWN” è comunque possibile effettuare l’operazione con l’istruzione **SGL**.

L’OPERAZIONE di scrittura delle Globali viene effettuata con una modalità di COMMIT per assicurare l’integrità con l’ultimo salvataggio.

Per indicizzare Globali o Locali si può usare la parentesi seguito dalla Globale o Locale utilizzate come indice ( l’uso della “I” è rimasta per compatibilità ).

Esempio

:

;questo è un esempio di programmazione indicizzata -LET/ L1, 1

-LET/ G (L1), 0

here -TMM/50

-JNE / G (L1) 1, here ; expects the value of global variable G1 is set to 1



# MultiTask

*Una caratteristica importante nell'automazione è il poter effettuare più operazioni insieme, coordinate tra loro o meno, quindi abbiamo la necessità di avere la funzionalità di "MULTITASK".*

Un ciclo di attività può essere eseguito con un comando esplicito, oppure da una istruzione "-TSK", il ciclo o programma viene abbinato ad un HANDLER di lavoro proprietario su cui si appenderanno tutte le attività del ciclo, possiamo aver un numero pressoché infinito di HANDLER.

Un "TASK" può essere cancellato da parte di un altro task o da se stesso con l'istruzione "-TKM" oppure quando viene eseguito il comando di RESET per il sotto sistema di automazione.

Il concetto di HANDLER è utile inoltre vederlo come un canale su cui vengono effettuate operazioni di inizio della movimentazione in continuo , aggregazioni di movimento , attesa che i movimenti siano completati.

# Anticollisione tra gli assi

Con questa architettura noi siamo in grado di vedere il sistema di movimentazione come una apparecchiatura con numerosi “Bracci” che lavorano insieme coordinati o meno, aggregando dinamicamente gruppi di assi,.

Un esempio interessante può essere immaginare lo riempimento di un vassoio di bicchieri, in una fase iniziale abbiamo due “Bracci” che separatamente riempiono i calici, il sistema gestisce l’anticollisione e abbiamo così due gruppi di assi che lavorano separatamente, quando i calici sono stati riempiti il sistema, raggruppando i due bracci come farebbe un cameriere, porta il vassoio in zona di scarico.

Come prima evidenziato durante il riempimenti dei calici, i due bracci incidendo su uno stesso asse fisico X, grazie alla gestione dell’anticollisione è possibile programmare due cicli indipendenti di riempimento, sincronizzare i due a completamento delle loro rispettive fasi, quindi programmare un unico ciclo di scarico del vassoio con un raggruppamento unico dei due bracci.

# Istruzioni matematiche

- .1-LET(SET) Imposta ad una Variabile il valore dell'espressione ( LET )
- .2-ADD Somma ad una Variabile il valore dell'espressione ( ADDed )
- .3-MUL Moltiplica una Variabile con il valore dell'espressione ( MULtipliy )
- .4-DIV Dividi una Variabile con il valore dell'espressione ( DIVided )
- .5-NEG Nega il valore di una Variabile ( NEGation )
- .6-LBF Imposta ad un vettore di Variabili il valore dell'espressione( Load BuFfer )

# Istruzioni di controllo 1 parte

- .1-JMP Salta incondizionatamente ad una Label ( JuMP )
- .2-JEQ Salta ad una Label se le due espressioni sono uguali ( Jump if Equal )
- .3-JNE Salta ad una Label se le due espressioni non sono uguali ( Jump in Not Equal )
- .4-JLT Salta ad una Label se il valore del primo parametro è minore del secondo ( Jump if Less Then )
- .5-JLE Salta ad una Label se il valore del primo parametro è minore o uguale del secondo ( Jump if Less then and Equal )
- .6-JGT Salta ad una Label se il valore del primo parametro è maggiore del secondo ( Jump if Great Then )
- .7-JGE Salta ad una Label se il valore del primo parametro è maggiore o uguale del secondo ( Jump if Great then and Equal )
- .8-JRN Salta se il valore del parametro è compreso nel range ( Jump if RaNge )
- .9-JNR Salta se il valore del parametro è fuori del range( Jump if Not Range )

# Istruzioni di controllo 2 parte

- .10-JOS Salta se almeno un bit del valore del parametro è uno ( Jump Or if bit Set )
- .11-JOC Salta se almeno un bit del valore del parametro è zero ( Jump Or if bit Clear )
- .12-JAS Salta se tutti i bit del valore del parametro sono uno ( Jump And if bit Set )
- .13-JAC Salta se tutti i bit del valore del parametro sono zero ( Jump And if bit Clear )
- .14-CAL Chiama un part program passandogli dei parametri ( CALI )
- .15-RET Ritorna al chiamante del programma ( RETurn )
- .16-END Fine processo ( END )
- .17-TSK Esegue in parallelo un ciclo di lavoro ( TaSK )
- .18-TKM Sospende, ripristina e cancella un ciclo di lavoro ( Task Manegement )
- .19-DIM Dimensiona il numero di variabile L di un part pogram

# Istruzioni di movimentazione 1

## parte

- .1-HOM(OMO) Origine di un asse ( HOMing )
- .2-MOV Movimento interpolato linearmente di un gruppo di assi ( MOVE )
- .3-CIR Movimento interpolato circolare o ellittico in senso orario di un gruppo di assi ( Circular Right )
- .4-CIL Movimento interpolato circolare o ellittico in senso antiorario di un gruppo di assi ( Circular Left )
- .5-CRR Movimento interpolato circolare o ellittico in senso orario di un gruppo di assi con raggio noto ( Circular Radius Right )
- .6-CRL Movimento interpolato circolare o ellittico in senso antiorario di un gruppo di assi con raggio noto ( Circular Radius Left )
- .7-STC Inizio di una movimentazione in continuo con definizione di percorso ( STart Continuous )
- .8-HLC Attesa del completamento della movimentazione in continuo ( HaLtContinuous )
- .9-ABC Cancellazione del movimento in continuo ( Abort Continuous )
- .10-CAP Cambia i parametri asse ( Change Axis Parameter )
- .11-HMS Gestione Master Slave ( Handling Master Slave )

# Istruzioni di movimentazione 2 parte

- .12-HEC Gestione delle camme ( Handling Electronic Cam )
- .13-GEI Legge le informazioni della camma ( Get Electronic Cam Information)
- .14-CFR Cambia i parametri dinamici di un asse ( Change Feed Rate )
- .15-CPL Cambia il loop di posizione ( Change Position Loop )
- .16-PRD Legge le posizioni dell'asse ( Position ReaD )
- .17-RAV Legge i parametri dell'asse ( Read Axis Value )
- .18-RSV Legge la velocità di un mandrino ( Read Speed Value )
- .19-SFP Imposta la velocità del profilo di movimentazione ( Set Feed Profile )
- .20-SPD Imposta la velocità di rotazione di un mandrino ( SPeeD )
- .21-TCH Movimento con tostatura ( TouCH )
- .22-TMT Movimento con ricerca valore di segnale analogico( Test Movement Transducer )

# Istruzioni di movimentazione 3 parte

.23-TMS	Movimento con ricerca valore di sensore digitale ( Test Movement Sensor )
.24-TPE	Abilita il tastatore ( Touch Probe Enable)
.25-SZP	Definisci la posizione di un set zeri macchina ( Set Zero Point )
.26-LZP	Attiva un set di zeri macchina ( Load Zero Point )
.27-PIN(INQ)	Flag di incrementale su un asse ( Position INcremental )
.28-PAB(ABS)	Flag di assoluto su un asse ( Position ABSolute )
.29-MMA	Muove un asse con un movimento manuale (Move Manual Axis)
.30-OPT	Apri un file di punti (Open PoinT file)
.31-MOR	Movimento interpolato linearmente di un gruppo di assi con anticipo ( MOv Re )
.32-DCT	Movimento con profondità controllata da tastatore ( Deep Control Touch)
.33-DCS	Movimento con profondità controllata da ingresso digitale ( Deep Control Sensor)
.34-GRM	Comandi agli assi e mandrini raggruppati ( GRoup Management )



# Istruzioni I/O 1 parte

- .1-WDI (WIN) Attende che un segnale d'ingresso digitale si porti ad un determinato stato ( Wait Digital Input)
- .2-WAI Attende che un segnale d'ingresso analogico si porti ad un determinato stato ( Wait Analog Input )
- .3-AIN Attendi per il case uno degli input mettendo il numero sulla G o L ( Attesa INput)
- .4-TDI(TES) Effettua un'operazione di test su un segnale d'ingresso digitale ( Test Digital Input )
- .5-TDO Effettua un'operazione di test su un segnale d'uscita digitale ( Test Digital Output )
- .6-IDI Effettua un'operazione di test su un segnale d'ingresso digitale ( If Digital Input )
- .7-IDO Effettua un'operazione di test su un segnale d'uscita digitale ( If Digital Output )
- .8-TAI Effettua un'operazione di test su un segnale d'ingresso analogico( Test Analog Input )
- .9-SDO(SAX) Setta o resetta dei segnali d'uscita digitale ( Set Digital Output )
- .10-EDO Setta o resetta dei segnali d'uscita digitale in base ad un test ( Enhanced Digital Output )

# Istruzioni I/O 2 parte

- .11-SAO(SAC) Scrive il valore di un segnale analogico d'uscita ( Set Analog Output )
- .12-GDI(RBI) Legge il valore di un segnale digitale d'ingresso ( Get Digital Input )
- .13-GDO Legge il valore di un segnale digitale d'uscita ( Get Digital Output )
- .14-GAI(RAI) Legge il valore di un segnale analogico d'ingresso ( Get Analog Input )
- .15-WBD(BPO) Scrive un blocco di segnali digitali in uscita (Write Buffer Digital input )
- .16-RBD(BPI) Legge un blocco di segnali digitali in ingresso ( Read Buffer Digital input )
- .17-CPI Attende che uno dei segnali d'ingresso digitale subisca una variazione
- .18-CDI Al cambiamento del segnale d'ingresso digitale attiva un task o processo
- .19-CDO Al cambiamento del segnale d'ingresso digitale attiva un task o processo
- .20-RDI Effettua un'operazione di test su un segnale d'ingresso digitale attivando un Task se il test è positivo ( Run Digital Input )
- .21-RDO Effettua un'operazione di test su un segnale d'uscita digitale attivando un Task se il test è positivo ( Run Digital Output )

# Istruzioni di sincronizzazione

- .1-EVS    Setta degli eventi di sincronizzazione ( Event Set )
- .2-EVC    Resetta degli eventi di sincronizzazione ( Event Clear )
- .3-EVW    Attende alcuni eventi di sincronizzazione ( Event Wait )
- .4-EVG    Legge lo stato degli eventi ( Event Get )
- .5-CSA    Crea una sincronizzazione per l'uso di un asse tra più task ( Create Syncro Axes )
- .6-WSA    Attende la sincronizzazione per l'uso di un asse tra più task ( Wait Syncro Axes )
- .7-DSA    Elimina la sincronizzazione per l'uso di un asse tra più task ( Delete Syncro Axes )

# Istruzioni di servizio 1 parte

- .1-FOC(AZZ)      Azzera il contenuto un file o la crea se non esiste ( File Open Create )
- .2-FWR            (SCR)    Scrive un record su file ( File WRite )
- .3-FWA            Scrive un record su file ( File Write Ascii)
- .4-FRD            (LEG)    Legge un record da file ( File ReaD )
- .5-TIM            Temporizzatore in secondi ( TiME )
- .6-TMM            Temporizzatore in millesimi di secondo ( TiME Millisecond )
- .7-SWA            Inizializza un orologio ( Start WAtch )
- .8-RWA            Legge un orologio ( Read WAtch )
- .9-HWA            Ferma un orologio ( Halt WAtch )
- .10-CWA           Continua un orologio ( Continue WAtch )
- .11-KYB           Attende un'operazione da tastiera ( KeYBoard )
- .12-DRT           Visualizza continuamente i valori di assi, globali, segnali ( Display Real Time )
- .13-DIS           Visualizza una riga di messaggio ( DISplay )

# Istruzioni di servizio 2 parte

.14-HLD	Manda il sistema nello stato di Cycle Stop ( HoLD )
.15-PWO	Manda il sistema nello stato di Power ON ( PoWerOn )
.16-EMC	Reset del sistema ( ReSeT )
.17-LCK	Lock il task ed eventualmente segnala con una SEC
.18-ULK	Rilascia tutti i task in stato di LOCK
.19-RST	Reset del sistema ( ReSeT )
.20-SDW	Shut down del sistema ( ShutDoWn )
.21-SOR	Ordinamento di una sequenza di valori ( SORT )
.22-GTK	Rileva le informazioni inerenti ad un TASK o processo ( Get Task information )
.23-MDI	Esegue un'istruzione ISO - GCODE ( ISO )
.24-OTC	Modifica le tabelle O,T,C - GCODE ( ISO )
.25-ISO	Esegue un programma ISO - GCODE ( ISO )
.26-WND	Attende una segnalazione di stato di errore delle risorse assi o mandrino (Wait Notify Detected )
.27-WKY	Attende la premuta di un tasto (Wait KeYboard)
.28-NHL	No Hold
.29-YHL	Yes Hold

# Istruzioni di servizio 3 parte

.30-GDT	Get Date and Time
.31-GAT	Get Absolute Data e Time
.32-GLN	Get Numero locali
.33-GMI	Get Motion Information
.34-RTC	Read Timer o Counter
.35-SGL	Save Globali
.36-SHL	Shell di applicativi o procedure
.37-G80	Fine ciclo fisso (G80)
.38-G81..89	G81-G89 Attiva il ciclo fisso specificato
.39-ESE	Esegue delle sequenze esterne ( Exec Sequence) <b>sistema ETEL</b>
.40-ERR	Leggi registri esterni ( External Read Registry) <b>sistema ETEL</b>
.41-EWR	Scrivi registri esterni( External Write Registry) <b>sistema ETEL</b>
.42-ECM	Esegue un comando esterno (External CoMmand) <b>sistema ETEL</b>
.43-EWS	Attende una segnalazione (External Wait Signal) <b>sistema ETEL</b>
.44-CLM	Comando da Logica di Macchina <b>sistema ETEL</b>
.45-SND	Effettua l'emissione di un file Wav sull' uscita audio del PC

# Istruzioni per l'integrazione con gli altri ambienti

- .1-ARI Richiesta di esecuzione di una istruzione per l'ambiente specificato nel primo parametro e attesa della risposta ( ritorno effettuato con la funzione di "WriteServiceParametersAndContinue" al sottosistema di automazione AXESBRAIN ) ( Ambient Request Instruction )
- .2-SEC Set event client

# Istruzioni di comunicazione 1 parte

- .1-CSO Connessione ad un socket TCP/IP ( Connect SOcket )
- .2-LSO Listen ad un socket TCP/IP ( Listen SOcket )
- .3-RSO Leggi i dati che transitano su un socket TCP/IP ( Read SOcket )
- .4-TSO Scrive su un socket TCP/IP ( Write SOcket )
- .5-DSO Cancella una connessione a un socket TCP/IP ( Destroy SOcket )
- .6-FSO Azzera i dati eventualmente ricevuti su un socket TCP/IP ( Free SOcket )
- .7-GSO Acquisisce le informazioni di un socket TCP/IP ( Get information SOcket )
- .8-OSL Apre una linea seriale ( Open Serial Line)
- .9-RXL Riceve dati da una linea seriale( Receive Serial Line)
- .10-TXL Trasmette dati su una linea seriale( Trasmit Serial Line)
- .11-CSL Chiude la linea seriale( Close Serial Line)
- .12-FSL Azzera i dati eventualmente ricevuti da una linea seriale(Free Serial Line)
- .13-RFB Leggi dati da FieldBus
- .14-WFB Scrivi dati su FieldBus



# Istruzioni di comunicazione 2 parte

.15-RGS	Reset linea GSM/GPRS
.16-SMS	Send SMS su GSM/GPRS
.17-WMS	Wait message SMS da GSM/GPRS
.18-CGS	Effettua una chiamata su GSM/GPRS
.19-WRG	Attende una chiamata da GSM/GPRS
.20-CTL	Effettua chiamata su linea telefonica
.21-WTL	Attende una chiamata da linea telefonica
.22-STL	Chiude la linea telefonica
.23-GTL	Acquisisce un numero da linea telefonica
.24-PTL	Invia un file registrato su linea telefonica
.25-EML	Invia un E-Mail

# AWL GP-PLC Operazioni speciali a contatti 1 parte

Il **Contatto normalmente aperto** è chiuso (on) se il valore del bit interrogato dell'indirizzo n è 1.

In AWL il contatto normalmente aperto è rappresentato dalle operazioni del tipo: **Carica operazione, Combina il valore di bit tramite And, Combina il valore di bit tramite OR**. Queste operazioni, rispettivamente, caricano il valore di bit dall'indirizzo n nel valore superiore dello stack logico, o combinano tramite And o OR il valore bit dell'indirizzo n con il valore superiore dello stack logico.

**Contatto normalmente chiuso** è chiuso (on) se il valore del bit interrogato dell'indirizzo n è 0

In AWL il contatto normalmente chiuso è rappresentato dalle operazioni del tipo: **Carica il valore di bit negato, Combina il valore di bit negato tramite And, Combina il valore di bit negato tramite OR**.

Queste operazioni caricano il valore di bit dall'indirizzo n nel valore superiore dello stack logico, o combinano tramite And o Or il valore bit dell'indirizzo n con il valore superiore dello stack logico.

# AWL GP-PLC Operazioni speciali a contatti 2 parte

## **Contatti diretti**

Il **Contatto diretto normalmente aperto** è chiuso (on) se il valore del bit dell'ingresso fisico indirizzato n è 1.

In AWL il contatto diretto normalmente aperto è rappresentato dalle operazioni del tipo **Carica il valore di bit direttamente, Combina bit direttamente tramite And, e Combina bit direttamente tramite OR.**

Queste operazioni, rispettivamente, caricano direttamente il valore di bit dall'indirizzo n al valore superiore dello stack logico, o combinano direttamente tramite And o OR il valore bit dell'ingresso fisico indirizzato n con il valore superiore dello stack logico.

Il **Contatto diretto normalmente chiuso** è chiuso (on) se il valore del bit dell'ingresso fisico indirizzato n è 0.

In AWL il contatto diretto normalmente chiuso è rappresentato dalle operazioni del tipo **Carica il valore di bit negato direttamente, Combina direttamente il valore di bit negato tramite And, e Combina direttamente il valore di bit negato tramite OR.**

Queste operazioni, rispettivamente, caricano direttamente il valore di bit negato dall'indirizzo n nel valore superiore dello stack logico, o combinano direttamente tramite And o OR il valore bit negato dell'indirizzo n con il valore superiore dello stack logico.

# AWL GP-PLC Operazioni speciali a contatti 3 parte

## **Contatto Not**

Il contatto NOT modifica lo stato dei segnali. Se il flusso di corrente raggiunge il contatto Not, questo viene bloccato. Se il flusso non raggiunge il contatto Not, questo genera flusso di corrente.

In AWL l'operazione **Negazione del valore superiore** modifica il valore superiore dello stack da 0 a 1, o da 1 a 0.

## **Normative semantiche**

Il carattere **#** indica che il valore dell'espressione numerica è interpretata come valore 0 oppure 1.

## **Esempio 1:**

-NET/1

-LD/#1

; Viene caricato il valore 1 nello STACK

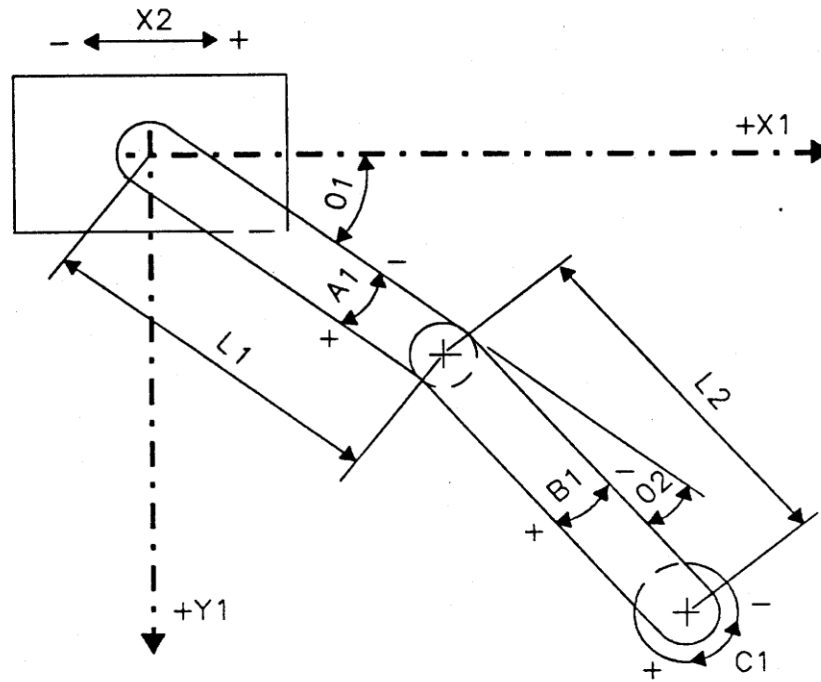
# Istruzioni AWL PLC 1 parte

.1-NET	PLC Network
.2-LD	PLC Carica operazione
.3-LDN	PLC Carica operazione negata
.4-A	PLC Combina il valore di bit tramite AND
.5-AN	PLC Combina il valore di bit negato tramite AND
.6-O	PLC Combina il valore di bit tramite OR
.7-ON	PLC Combina il valore di bit negato tramite OR
.8-EU	PLC Rilevamento di fronte positivo
.9-ED	PLC Rilevamento di fronte negativo
.10-EQU	PLC Assegna, copia nel parametro specificato il valore superiore dello stack
.11-S	PLC Imposta ad 1 il numero di punti specificato se lo stack è 1
.12-R	PLC Imposta ad 0 il numero di punti specificato se lo stack è 1
.13-LPP	PLC Prelevamento logico
.14-LPS	PLC Duplicazione logica
.15-LRD	PLC Copiatura logica
.16-ALD	PLC Combina il primo e il secondo elemento tramite AND

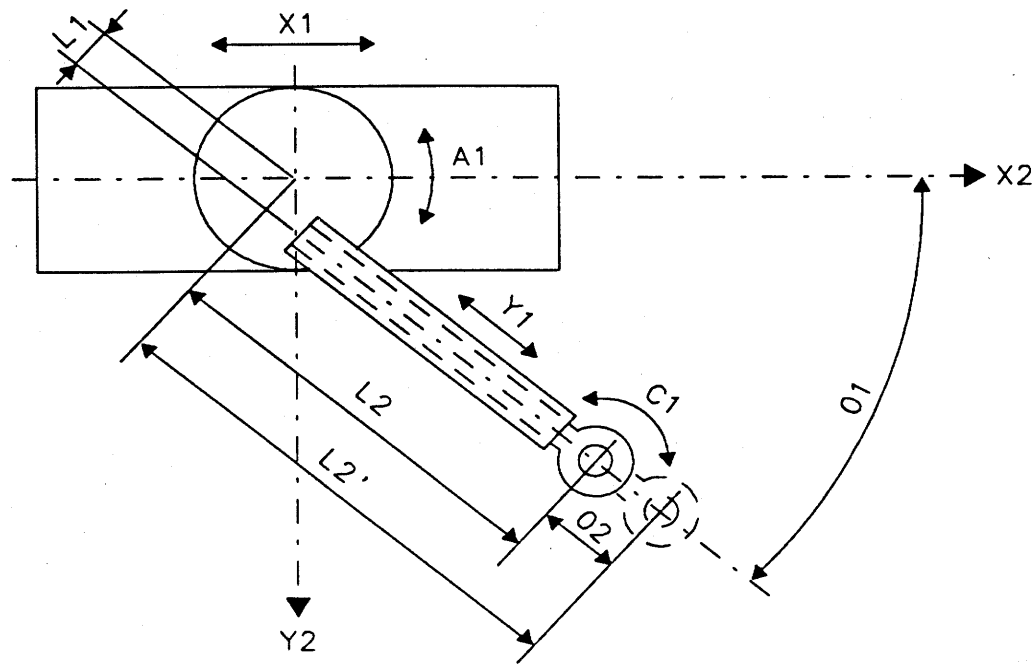
# Istruzioni AWL PLC 2 parte

.17-OLD	PLC Combina il primo e il secondo elemento tramite OR
.18-NOT	PLC Modifica del valore superiore
.19-LEQ	PLC Confronta due valori se uguali carica con lo stack 1 se no 0
.20-LGE	PLC Confronta due valori se uguali o maggiore carica con lo stack 1 se no 0
.21-LLE	PLC Confronta due valori se uguali o minore carica con lo stack 1 se no 0
.22-AEQ	PLC Confronta due valori se uguali fa l'AND di 1 con lo stack
.23-AGE	PLC Confronta due valori se uguali o maggiore fa l'AND di 1 con lo stack
.24-ALE	PLC Confronta due valori se uguali o minore fa l'AND di 1 con lo stack
.25-OEQ	PLC Confronta due valori se uguali fa l'OR di 1 con lo stack
.26-OGE	PLC Confronta due valori se uguali o maggiore fa l'OR di 1 con lo stack
.27-OLE	PLC Confronta due valori se uguali o minore fa l'OR di 1 con lo stack
.28-PEX	PLC Esegui su 1 in stack l'istruzione AXESBRAIN
.29-TON	PLC Timer senza ritenzione
.30-TOR	PLC Timer con ritenzione
.31-CTU	PLC Counter Up
.32-CUD	PLC Counter Up e Down

# Asse virtuale SCARA



# Asse virtuale cilindrico





# Controllo asse con volantino

*Il posizionamento di un asse in manuale può essere abbinato ad un dispositivo chiamato volantino che viene visto dal sistema come un asse di sola lettura .*

Il valore di posizione letto dal volantino modifica la posizione dell'asse abbinato, è così possibile dare degli incrementi micrometrici all' asse stesso.

Il volantino è visto come un asse di sola lettura, che tramite opportuni comando viene agganciato ad un asse che ne rimarrà controllato.

Nel linguaggio di automazione AXESBRAIN, l'istruzione “-HMS” permette l'inserimento e il disinserimento di un asse volantino “master” ad un asse di posizionamento “slave”.

Per gestire l'abbinamento asse volantino con asse di posizionamento in ambiente “DCOM”, deve essere utilizzato il servizio “WriteAxesRegister” per entrambi gli assi.

# Gantry

*L'asse a portale( Gantry)è una struttura meccanicamente rigida ( normalmente è una struttura a portale) e corrisponde pertanto ad un asse unico, ma è trattata dal controllo come se fosse costituita da una coppia di assi ( asse master e asse slave, ciascuno con i propri sistemi di conteggio ed il proprio azionamento.*

*Una delle funzioni del controllo è quella di mantenere la posizione dell'asse "slave" più prossima possibile a quella dell'asse "master".*

I movimenti richiesti sugli assi Master-Slave, vengono eseguiti nel modo seguente:  
La richiesta di movimento può essere effettuata tramite tutte le istruzioni di movimento.

Nel caso di movimento relativo al solo asse Master, l'asse Slave lo segue o rimane fermo rispettivamente in caso di associazione o disassociazione.

Durante il movimento, l'asse Slave segue il suo Master in tempo reale

# Camme elettroniche

*La camma elettronica permette di abbinare la posizione di un gruppo di assi ad un asse “master” ed una tabella di posizioni multiple.*

E' così possibile simulare elettronicamente il comportamento delle camme, sostituirne il funzionamento meccanico con un sistema analogo formato da un gruppo di assi asservito ad un asse “master” che può essere di sola lettura. Le leggi del moto dei cedenti è definita come una tabella di vettori, che ne defisse le posizioni rispetto al moto della camma.

Nel linguaggio di automazione AXESBRAIN, l'istruzione “-HEC” permette la gestione delle camme elettroniche.

Per gestire le camme elettriche in ambiente “DCOM”, deve essere utilizzato il servizio “WriteAxesRegister” per tutti gli assi .

# Automazione HMI Generica

The screenshot displays the AxesBrainStation HMI software interface. At the top, the title bar reads "AxesBrainStation". Below it is a menu bar with "File" and "Keyboard" options. A status bar shows the "Feed 100%" indicator. The main display area features six digital readout (DRO) fields labeled "Val: 1" through "Val: 6", all currently showing "0". Below these fields is a large, empty grey area, likely reserved for a 2D or 3D graphical representation of the workpiece. At the bottom of the main area, a file list on the left shows "3GLOBALI .pp" selected, with a scrollable list of sub-files including "3GLOBALI\_PP", "BlobBaricenter.pp", "BlobEllipse.pp", "CIL.PP", "CIR.PP", "Delay.pp", "EdgeCircle.PP", "EdgeEllipse.pp", "EdgeFinder.PP", "elica.pp", "ErroreAsse.pp", "GestSoftKey.PP", "LogicaMacc.pp", and "PatternMatching.PP". The main text area displays the G-code for the selected file:

```
3GLOBALI.pp
Activate visualization  Display part program in job

3GLOBALI_PP
BlobBaricenter.pp      -LET/L1,0
BlobEllipse.pp         pip -LET/G(L1+1),2.5
CIL.PP                 -TMM/33
CIR.PP
Delay.pp
EdgeCircle.PP          -LET/G(L1+2),2.5
EdgeEllipse.pp         -TMM/33
EdgeFinder.PP          -LET/G(L1+3),2.5
elica.pp               -TMM/33
ErroreAsse.pp          -LET/G(L1+1),-2.5
GestSoftKey.PP
LogicaMacc.pp
PatternMatching.PP
```

At the bottom of the interface is a row of function keys (F1-F12) with labels: F1 Select, F2 Run, F3 MDI, F4 Kill, F5 Start, F6 COMM, F7 Shut Down, F8 Reset, F9 ServoON, F10 Emerg, F11, and F12. Below the keys are status indicators for "Servo On", "Hold", and "Run". The bottom status bar shows "Speed = 100%", "48", "1.018", "0.120", "14/07/2011", and "18:18:30". On the right side, a vertical toolbar contains icons for "System", "Automation", "Editor", "Graphics 2D", "Graphics 3D", "Debug", and "Global".

# Automazione HMI Editor

The screenshot displays the AxesBrainStation software interface. The main window is titled "AxesBrainStation" and contains a menu bar with "File" and "Keyboard ?". Below the menu bar is a status bar showing "PLC running..." and a "Feed 100%" indicator. The central area is a code editor displaying the following PLC program:

```
-PWO
; ***** ELICA SU X1 Y1 Z1 *****
-DIS/1:-----
;
Attesa servo on
a
-SET/L1,0
-SFP/1000,100,100
-MOV/X1,10,Y1,0,Z1,0
-CAP/0:Z1,128
-STC/0:-100
PIPPO
-CIR/X1,10,X1,0,Y1,0,Y1,0,Z1,10
-ADD/L1,1
-JLE/L1,10,PIPPO
-HLC
-CAP/0:Z1,-128
-JMP/a
```

On the left side, there is a "Edit Part Program" section with a list of files. The file "elica.pp" is selected. Below this list are buttons for "Ln 1, Col 1", "Import <--", and "Export -->". At the bottom of the interface, there is a toolbar with function keys F1 through F12, each with a corresponding action: F1 Select, F2 Restore, F3 Save, F4 Rename, F5 Copy, F6 New, F7 Cancel, F8 Select char, F9 Goto line, F10 Change, and F12 Search. The bottom status bar shows "Servo On", "Hold", "Run", "Speed = 100%", "232", "1.006", "0.040", "14/07/2011", and "18:22:18". On the right side, there is a vertical toolbar with icons for "System", "Automation", "Editor", "Graphics 2D", "Graphics 3D", "Debug", "Global", "CNC", and "Vision".

# Automazione HMI 2D Grafica

The screenshot displays the AxesBrainStation HMI software interface. The main window is titled "Macchina OK.." and features a central 2D coordinate system with a black background and green axes. A yellow wireframe drawing of a scorpion is centered on the origin. The text "<-190,150 mm" is visible in the top-left and bottom-left corners of the coordinate system. The interface includes a menu bar (File, Tastiera, ?), a toolbar with the AxesBrain logo, and a right-hand sidebar with various system and automation functions. The bottom of the screen shows a row of function keys (F1-F12) and a status bar with real-time data.

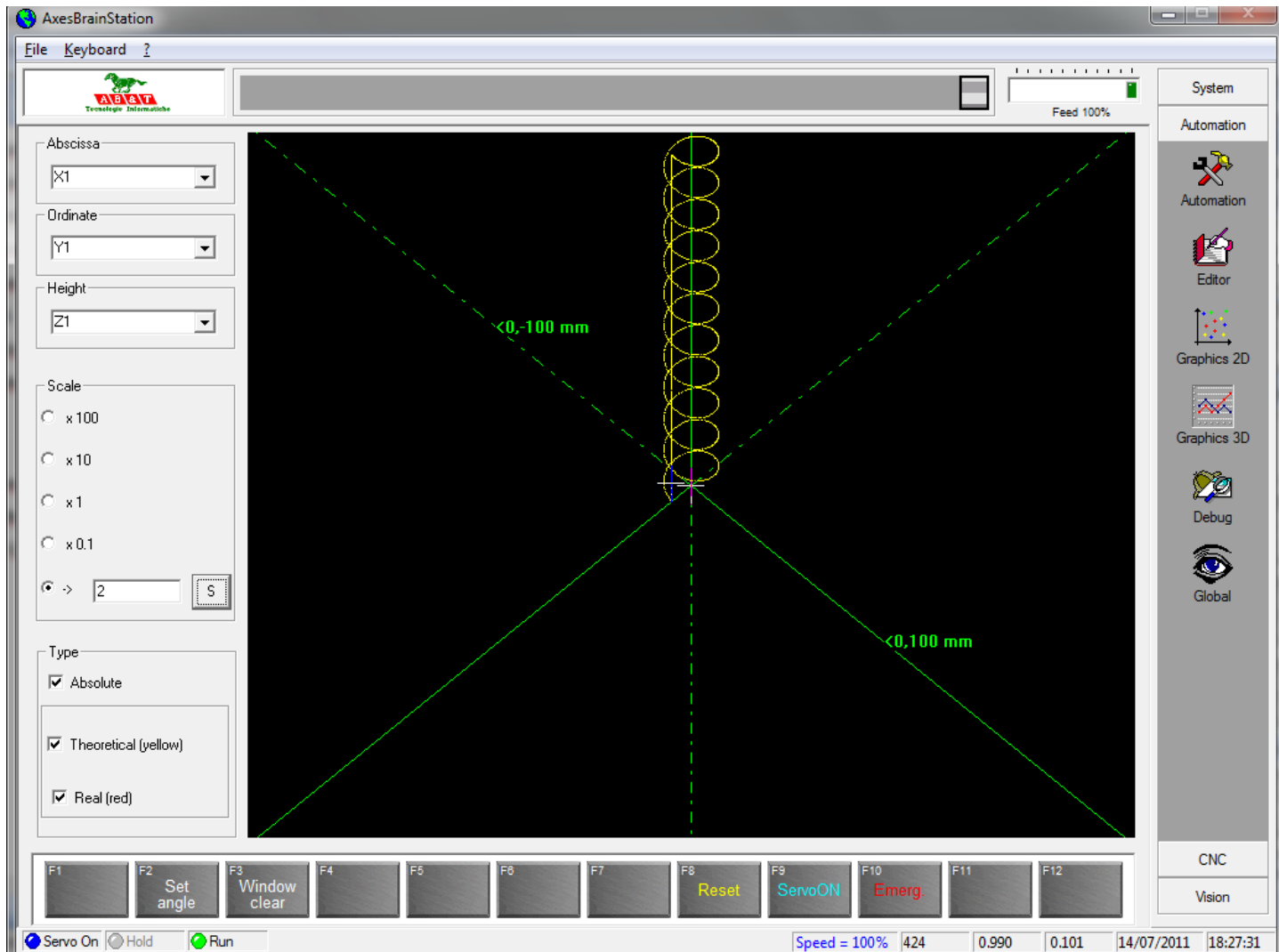
**System and Automation Panel (Right Sidebar):**

- Sistema
- Automazione
  - CNC
  - CNC
  - Editor
  - Grafica a punti 2D
  - Grafica CNC
  - Setup
  - Convertitore DXF->ISO
- Visione
- Transfer
- Gestore cella
- Tomio

**Control Panel (Bottom):**

- F1, F2, F3 (Window clear), F4, F5, F6, F7, F8 (Reset), F9 (ServoON), F10 (Emerg), F11, F12
- Speed = 100% | 33 | 1.008 | 0.215 | 17/10/2006 | 13:03:55

# Automazione HMI 3D Grafica



# Automazione HMI Debug

The screenshot displays the AxesBrainStation software interface. The main window shows a ladder logic program with the following code:

```

-DIM/100
-LET/G3,1
-DIS/1:
-DIS/2:
-DIS/3:
-DIS/4:
-LET/L1,1 // iNumberCameraActive
-ARI/1,12,VisAlgo,SetActiveCamera,L1,L1,L21,L21

pippo -SWA/1
-LET/L1,0 // iTipoPalette,
-LET/L2,1 // iNumeroImagini,
-ARI/1,12,VisionBase,ReadImage,L1,L1,L21,L21 ;,300,pippo

-LET/L21,0 // dXCenterPoint
-LET/L22,0 // dYCenterPoint
-LET/L23,110 // dRadiusMax
-LET/L24,70 // dRadiusMin
-LET/L25,150 // dDegreeStartFirstArc,
-LET/L26,210 // dDegreeEndFirstArc,
-LET/L27,330 // dDegreeStartSecondArc
-LET/L28,390 // dDegreeEndSecondArc
-LET/L29,150 // dDifferenziale
-LET/L30,2 // dDifferenziale
    
```

The line `-LET/L25,150 // dDegreeStartFirstArc,` is highlighted in red. The interface also shows a file explorer on the left, a system status bar at the top right, and a control panel at the bottom with various function keys (F1-F12) and status indicators.

**Actual Stack << 00000000**

**Global / Local / Timer / Counters**

G	Value	L	Value	T	Value
1	2.500				
2	2.500				

**Input / Output / Events / NET**

Input	Value	Output	Value	Events	Value

**Trace Window**

**System**

Automation

Editor

Graphics 2D

Graphics 3D

Debug

Global

**CNC**

Vision

Servo On Hold Run Speed = 100% 572 0.990 0.053 14/07/2011 18:30:07



# Automazione HMI Globals View

The screenshot displays the 'Globals View' in the AxesBrainStation software. The interface includes a menu bar (File, Keyboard), a logo for 'ABCA' (Tecnologie - Informatiche), and a 'Feed 100%' indicator. The central table lists global variables with their current values. The right sidebar provides navigation options for different system views. The bottom status bar features function keys and real-time operational data.

Number	Name	Description	Value
10		Incremento velocità	0
11		Attesa in millisecondi	0
12		Velocità massima	0
13		Velocità minima	0
14		Numero cicli per pressione	0
15		Numero cicli globali	0
16		Sensore di velocità presente (0 = Assente)	0

System  
Automation  
Automation  
Editor  
Graphics 2D  
Graphics 3D  
Debug  
Global  
CNC  
Vision

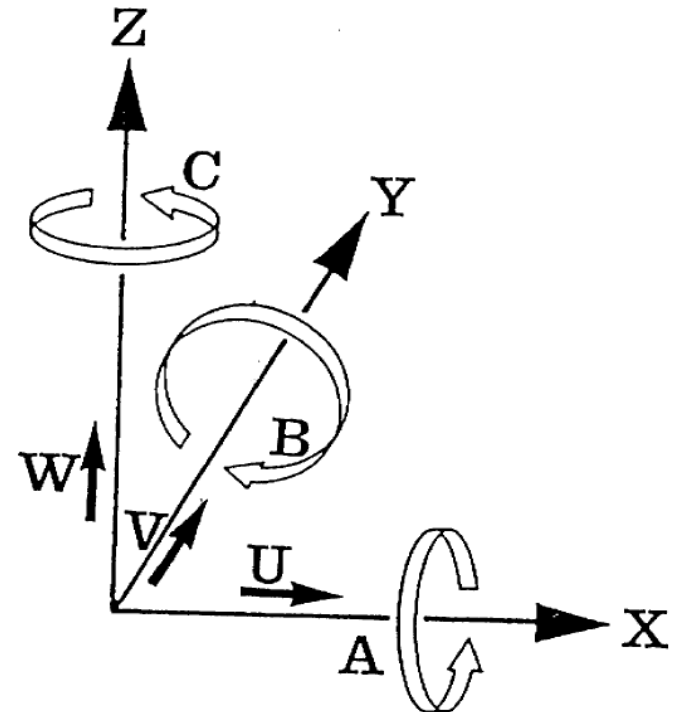
F1 Open F2 Save F3 Cancel grid F4 F5 F6 F7 F8 F9 F10 F11 F12

Servo On Hold Run Speed = 100% 622 0.996 0.037 14/07/2011 18:31:42

# ISO G-CODE

Si possono avere fino a 32 processi simultanei ISO CNC

- 1-G funzioni
- 2-M funzioni
- 3-Origins Tabella origini
- 4-Tools Tabella utensili
- 5-Parameters Tabella parametri
- 5-G M and Funzioni speciali
- 7-Cicli fissi
- 8-G89 Ciclo fisso
- 9-Funzioni di controllo programma



# Funzioni G 1 parte

G00	posizionamento rapido degli assi
G01	interpolazione lineare
G02	interpolazione circolare o elicoidale senso orario
G03	interpolazione circolare o elicoidale senso antiorario
G04	pausa temporizzata, tempo di pausa programmato.
G08	decelerazione alla fine del blocco che la contiene
G09	attivazione modalità "Guardare Avanti"
G17	specifica XY come piano lavoro e Z asse perpendicolare
G18	specifica ZX come piano lavoro e Y asse perpendicolare
G19	specifica YZ come piano lavoro e X asse perpendicolare
G30	decelerazione alla fine del blocco che la contiene e ripartenza in continuo
G32	fine subroutine
G40	annulla G41 e G42
G41	attivazione correzione raggio, utensile a sinistra del profilo
G42	attivazione correzione raggio, utensile a destra del profilo
G49	dichiarazione valore raggio
G50	fine rototraslazione
G51	rototraslazione

# Funzioni G 2 parte

G52	shift delle origini uguale G92
G54	mirror X
G55	mirror Y
G56	mirror Z
G57	mirror X e Y
G58	mirror Z e X
G59	mirror Y e Z
G60	fine fattore di scala
G61	fattore di scala
G62	centro del cerchio K1 assoluto K2 Incrementale
G70	programmazione in pollici
G71	programmazione in millimetri
G75	programmazione cartesiana
G76	programmazione polare
G78	impostazione tangenziale al percorso 2D
G79	fine impostazione tangenziale al percorso 2D

# Funzioni G 3 parte

G80	annullamento cicli fissi
G81	ciclo fisso per foratura
G82	ciclo fisso per lamatura
G83	ciclo fisso per foratura profonda
G84	ciclo fisso per maschiatura
G85	ciclo fisso per alesatura
G86	ciclo fisso per barenatura
G89	ciclo fisso con richiamo di un part program AxesBrain
G90	programmazione assoluta
G91	programmazione incrementale
G92	shift delle origini
G100 - G1999	funzioni AxesBrain

# Funzioni M

M00	arresto programmato
M03	rotazione oraria del mandrino
M04	rotazione antioraria del mandrino
M05	arresto del mandrino
M06	cambio utensile
M07	attivazione refrigerante secondario
M08	attivazione refrigerante primario
M09	disattivazione refrigerante
M10	attivazione bloccaggio assi
M11	disattivazione bloccaggio assi
M13	rotazione oraria mandrino e attivazione refrigerante
M14	rotazione antioraria mandrino e attivazione refrigerante
M19	orientamento mandrino
M30	fine programma, azzera le funzioni ausiliarie attive
M31 - M1999	funzioni AxesBrain

# Origini

La funzione O0 disabilita le origini e le coordinate sono riferite allo zero macchina. La funzione O-1 richiama, dopo la programmazione di O0, l'ultima origine programmata.

Specifica il numero della origine della tabella delle ORIGINI i cui valori vengono utilizzati per attivare o disattivare (O0) le origini pezzo durante la lavorazione, ed è collocata nel file delle origini.

Ogni processo ISO lavora su una tabella propria di **Origini**, tramite la sessione **[AbbinamentoOrigini]** del file "sistema.txt" è possibile specificare l'intero percorso ed il nome del file contenente la tabella.

# Utensili

La funzione T serve per il cambio utensile, manuale o automatico. Le cifre che seguono la funzione T definiscono il numero dell'utensile da richiamare.

Oltre all'utensile la funzione T richiama tutti i parametri memorizzati nella tabella dei dati utensili (correttori lunghezza, raggio, ecc.)

Il modo di programmare il cambio utensile (manuale, automatico sequenziale o no, con o senza braccio scambiatore) dipende da come è stato realizzato dall'utente e da come è stata la tabella degli utensili ( TOOL ) viene utilizzata per attivare o disattivare gli utensili durante la lavorazione, ed è collocata nel file dei Tool.

Ogni processo ISO lavora su una tabella propria di **Utensili (TOOL)**, tramite la sessione **[AbbinamentoTOOL]** del file "sistema.txt" è possibile specificare l'intero percorso ed il nome del file contenente la tabella.



# Parametri

Il sistema permette per ogni istanza di part program ISO 256 parametri definiti dalla lettera P.

I parametri P possono essere usati al posto delle dichiarazioni numeriche di posizione od altro come esempio :

P34=10

P35=500

XP34Y0

G4 P35

Inoltre possono essere usato per controllare dei rami differenti di lavoro con la seguente sintassi

{Pm=Pn} Li

{Pm>Pn} Li

{Pm<Pn} Li

{Pm<=Pn} Li

{Pm>=Pn} Li

{Pm<>Pn} Li

## **Esempio:**

L=loop

P3=P3+1

{P3<10}loop

# Funzioni M e G speciali

Le funzioni G or M or H chiamano un part program GP-PLC con le variabili locali da L1 a L18 con questi valori:

L1 = Numero asse Z rispetto al Piano di lavoro

L2 = Numero mandrino S

L3 = Numero AsseMandrino AS oppure 0

L4 = Numero della testa definita in "Base\_NumeroTesta"  
oppure 0

L5 = Speed mandrino ( valore di speed attivo blocco  
compreso )

L6 = 1 se M, 2 se G

L7 = Numero della G o M chiamante

L8 = 1 se il comando in MDI

L11 = K

L12 = Q

L13 = J

L14 = I

L15 = H

L16 = Numero Origine

L17 = Numero Utensile

L18 = Numero correzione

# Cicli fissi

La programmazione delle procedure di foratura può essere semplificata con i cicli di foratura. La selezione disponibile dei cicli di foratura copre i casi standard più importanti. Il programmatore deve solo definire qualche parametro, per adattare i cicli di foratura alla sua applicazione particolare.

Un ciclo fisso di foratura in un programma CN parziale è sempre programmato nelle seguenti fasi:

Assegnare i parametri

Selezionare il ciclo fisso desiderato

Spostarsi alla posizione di lavoro in X e Y (una volta o ripetutamente)

Chiamare automaticamente e eseguire il ciclo fisso selezionato dopo aver raggiunto la posizione di lavoro

Deselezionare il ciclo fisso

# Ciclo Fisso G89

La funzione Ciclo Fisso G89 se dichiarata nel file “sistema.txt” nella voce PartProgramG89 della sezione ParametriGenerali, attiva il part program di automazione.

## **[ParametriGenerali]**

### **PartProgramG89= [ nome part program di automazione]**

Ad ogni movimento se seguono la Funzione G89, viene richiamato il part program di automazione definito nel file “sistema.txt”, con il passaggio di parametri:

L1 = Numero asse Z rispetto al Piano di lavoro

L2 = Numero mandrino S

L3 = Numero AsseMandrino AS oppure 0

L4 = Numero della testa definita in “Base\_NumeroTesta” oppure 0

L5 = Speed mandrino ( valore di speed attivo blocco compreso )

L6= PosZ\_Inizio

L7 = PosZ\_Foratura

L8 = PosZ\_Ritorno

L9 = VelZ\_Foratura o Passo

L10 = Tempo in Millisec oppure Percentuale stiramento

L11 = Primo incremento da PosZ\_Inizio

L12 = Incremento successivo (L11 – 10%di L11 in ISO)

L13 = Incremento Sicurezza

# Funzioni di controllo programma

## **Ripetizioni di parte di programma**

Utilizzando i codici L è possibile ripetere n volte un programma o parte di esso. Il numero massimo di ripetizioni è 32767.

La parte di programma che si vuole ripetere è racchiusa fra una definizione di riferimento “label” e l’istruzione di salto alla label seguita dal numero di ripetizioni.

Il numero di ripetizioni può essere un numero o un parametro.

## **Sottoprogrammi interni al programma**

Si intende per sottoprogramma una sequenza di blocchi che possono essere richiamati da punti diversi del programma principale (ad esempio la successione dei vari punti su cui applicare i diversi cicli fissi, foratura, carenatura, alesatura, ecc.) o un profilo da richiamare più volte in punti diversi o con correttori raggio diversi.

Il sottoprogramma è richiamato programmando la funzione L seguita dal numero del sottoprogramma.

I sottoprogrammi interni al programma principale vanno programmati alla fine dello stesso, dopo la funzione M30.

# ISO G\_CODE HMI Tabelle Fresa

The screenshot displays the AxesBrainStation HMI interface. At the top, the title bar reads "AxesBrainStation" and the menu bar includes "File" and "Keyboard ?". Below the menu bar, there is a status bar with the text "PLC running..." and a "Feed 100%" indicator. The main area is divided into several sections:


- Process:** A dropdown menu showing "Mill".
- Origins:** A tabbed interface with "Origins" selected, showing a table with columns for N., Description, X, Y, Z, A, B, C, U, V, and W. The table is currently empty, with a small rectangular highlight in the cell for row 9, column C.
- Tools:** A tabbed interface with "Tools" selected.
- Parameters:** A tabbed interface with "Parameters" selected.
- Correctors:** A tabbed interface with "Correctors" selected.

On the right side, there is a vertical toolbar with icons for various functions: System, Automation, CNC, Editor, Graphics 2D, Graphic CNC, Setup, Converter DXF->ISO, and Vision.

At the bottom, there is a control panel with buttons for "Save Table" (F1), "Restore Table" (F2), "Export Table" (F3), "Import Table" (F4), and "Selection process" (F10). The status bar at the very bottom shows "Speed = 100%", "0", "0.993", "0.037", "15/07/2011", and "11:53:08".

# ISO G\_CODE HMI Tabelle Tornio

File Tastiera ?


Macchina OK..
Feed 100%

Processo

Origini (O) | Utensili (T) | Parametri (L) | Dati utensili (D)

N.	Descrizione	X	Y	Z	A	B	C	U	V	W
1		-24.9296	2210.0000	40.0000	0	0	0	0	0	0
2		0	0	0	0	0	0	0	0	0
3										
4										
5										
6										
7										
8										
9										
10										
11										

**Asse**
0.000
Reale
Teorica
7.5
15
diámetro
raggio
pollici
mm
Errore
0
Posizionamento

X
Seleziona asse
X-
X+
STOP
0
Valore quota
Azzeramento quota

0
 posizionamento a passo costante
 Relativo all'azz.

Posizione attuale
Classica
Inizializzazione (Homing)

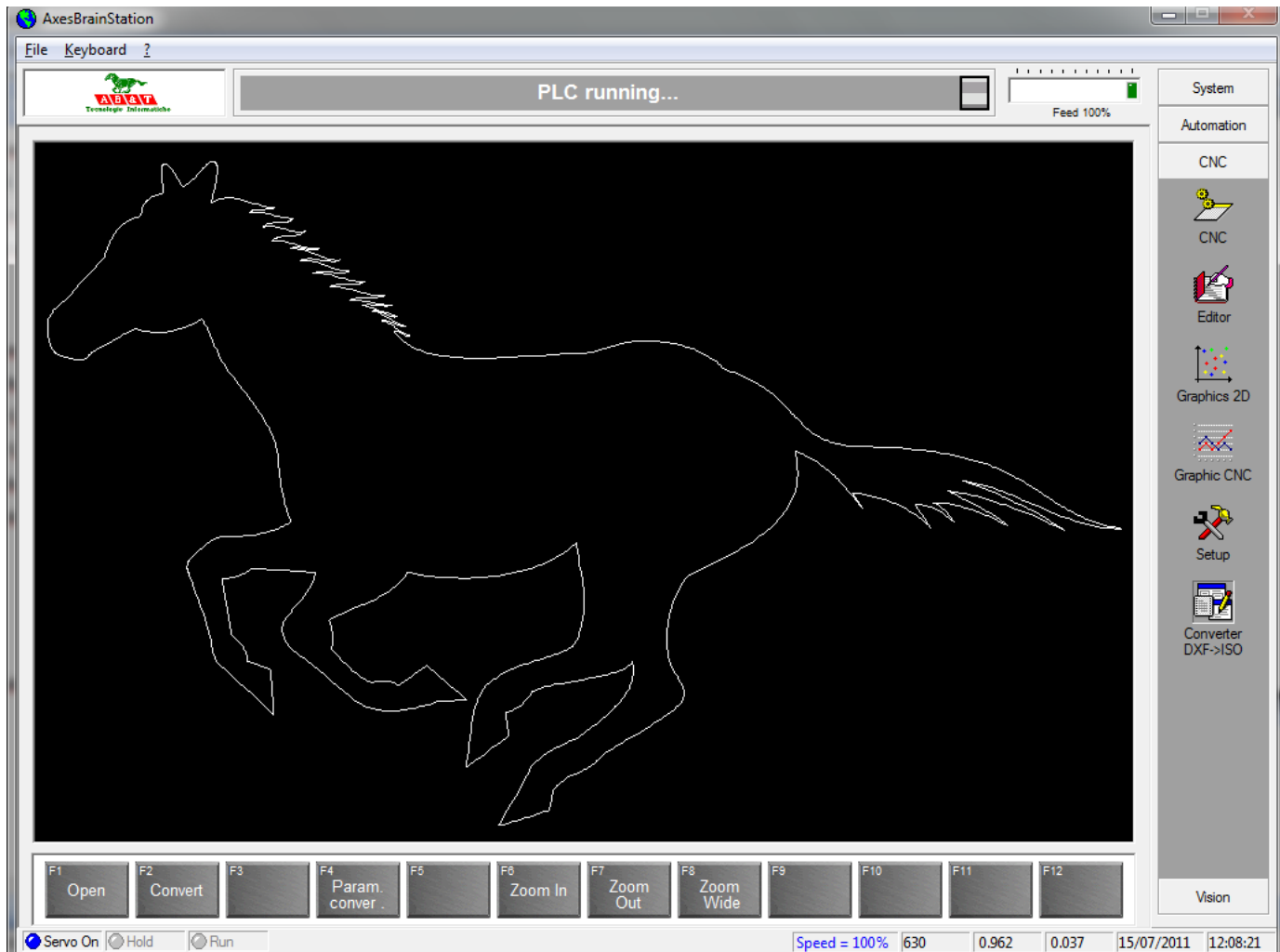
**Mandrino**
0
Velocità Reale
Velocità Teorica
Rotazione Oraria
Rotazione Antioraria
Arresto rotazione
M03
M04
M05
0
360
720
0
0
Posizionamento in gradi
M19

F1 Salva Tabella
F2 Ripristina Tabella
F3 Esporta Tabella
F4 Importa Tabella
F5
F6
F7
F8
F9
F10
F11
F12

Servo On
Hold
Run
Speed = 100%
5
1.018
0.195
16/10/2006
17:06:01

Sistema  
 Automazione  
 CNC  
 Visione  
 Transfer  
 Gestore cella  
 Tornio  
 Automatico  
 Grafica Edit  
 Setup  
 Lavorazione lenti

# ISO G\_CODE HMI DXF->ISO



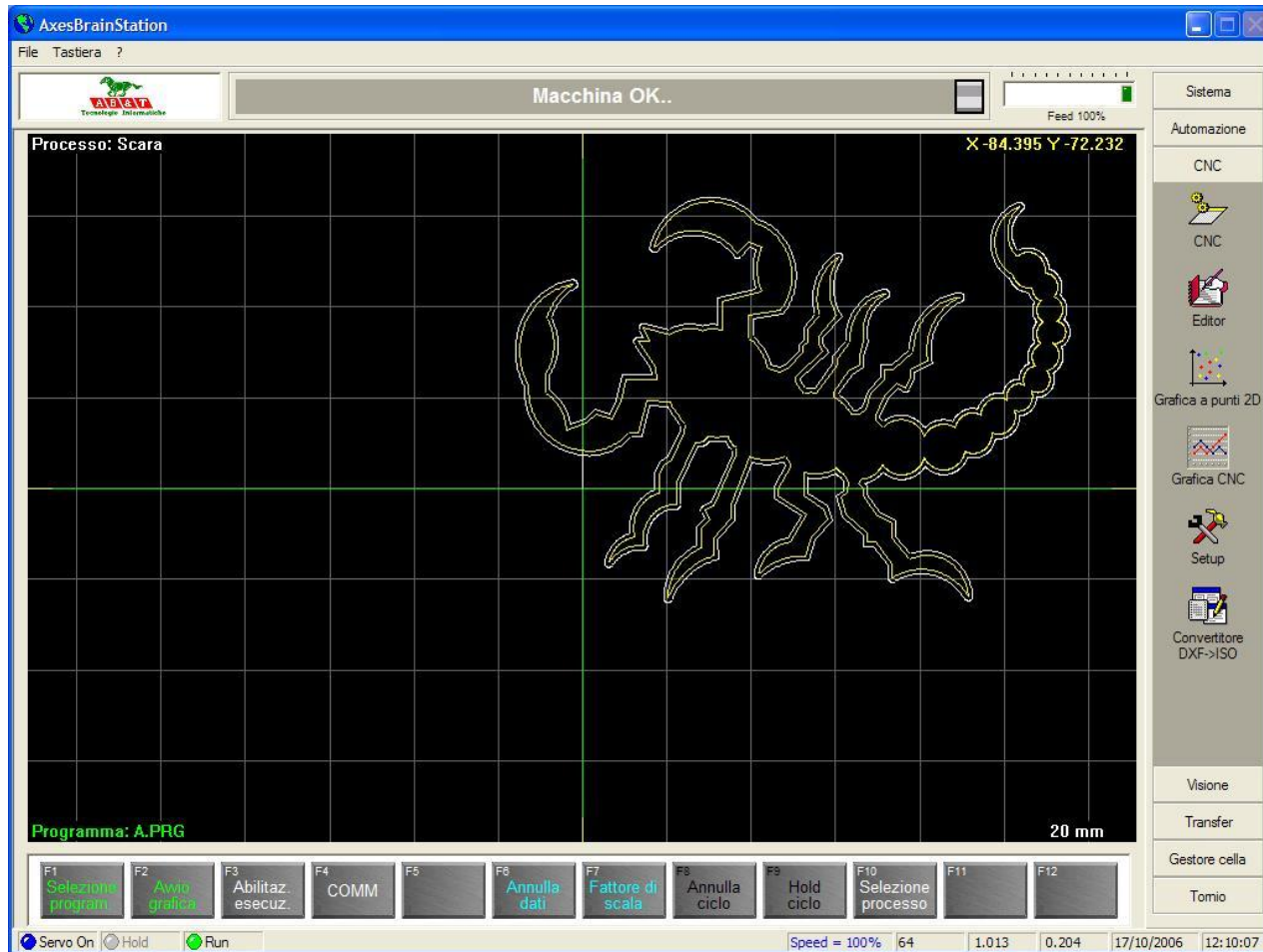


# ISO G\_CODE HMI CNC Fresa

The screenshot displays the AxesBrainStation HMI interface for a CNC machine. The main window is titled "AxesBrainStation" and shows the following information:

- System Status:** PLC running... (indicated by a green bar and "Feed 100%")
- Current position (mm):**
  - X: 0.000
  - Y: 0.000
  - Z: 0.000
- Final distance (mm):**
  - X: 0.000
  - Y: 0.000
  - Z: 0.000
- Name process:** Laser
- Feed rate (F):** 0 mm/min (0%)
- Spindle speed (S):** 0 RPM (0%)
- Couple (%):** 0%
- Program:** CAVALLOGrande.PRG (Start bl. 0)
- Execution:** Automatic, Operation: CNC
- Time:** (00:02:08) / 00:00:00
- Parameters:**
  - T: 0, D: 0
  - O: 0, Working: X-Y
  - CL: 0.000, CR: 0.000
  - F (mm/min): 5000, S (RPM): 1000
- G and modal M:**
  - G 17: G01 G71 G17 G80 G40 G90 G50 G08 G79
  - M: M05 M09
- Variable P:** 0, **Variable G:** 0
- 2D Coordinate System:** X -49.639, Y 56.859, 1.00 50 mm
- Control Panel:** F1 Selection program, F2 Start cycle, F3 MDI, F4 COMM, F5 Manual, F6 Set piece, F7 Modality cycle, F8 Cancel cycle, F9 Hold cycle, F10 Selection process, F11, F12
- Bottom Bar:** Servo On, Hold, Run, Speed = 100%, 839, 1.008, 0.043, 15/07/2011, 12:13:23

# ISO G\_CODE HMI Grafica Fresa



# ISO G\_CODE HMI manuale CNC

The screenshot displays the AxesBrainStation CNC HMI software interface. The main window shows the machine status as 'Macchina OK..' and the workpiece name 'FPZ500 A'. The current position is 0.000 mm on the X, Y, and Z axes. The feed rate is 100%. The 'Impostazioni pezzo' dialog box is open, allowing the user to configure workpiece settings. The dialog includes fields for tool length offsets (T and D), tool numbers, and workpiece origins (O). The 'Piano di lavoro' (Workplane) is set to X-Y (G17). The 'Esecuzione' (Execution) mode is set to 'AUTOMATICA' and the 'Funzionamento' (Operation) mode is 'AUTOMATICA'. The 'G e M modali' (Modal G and M codes) are G71, G17, G80, G40, G90, G50, G79, G75, M05, and M09. The 'Variabile P120' and 'Variabile G' are both set to 0. The 'Grafica a punti 2D' (2D Point Graphics) window shows the workpiece geometry. The 'Grafica CNC' (CNC Graphics) window shows the current tool path. The 'Setup' window shows the workpiece origin. The 'Convertitore DXF->ISO' (DXF to ISO Converter) window is also visible. The 'Visione' (Vision) window is active, showing the workpiece position. The 'Transfer' window is active, showing the workpiece transfer status. The 'Gestore cella' (Cell Manager) window is active, showing the cell status. The 'Tomio' (Tool) window is active, showing the tool status. The 'Servo On' button is active, indicating the machine is ready for operation. The 'Hold' button is inactive. The 'Run' button is active, indicating the machine is running. The 'Speed = 100%' indicator shows the current feed rate. The '48' indicator shows the current spindle speed. The '1.015' and '0.203' indicators show the current X and Y coordinates. The '17/10/2006' and '12:08:13' indicators show the current date and time.

**Impostazioni pezzo**

Correttore lunghezza utensile T secondo l'asse di lavoro : Z

Numero T  offset aggiuntivo  Memorizza Lunghezza

Attiva T  incremento  Memorizza Incremento

Correttore lunghezza utensile D secondo l'asse di lavoro : Z

Numero D  offset aggiuntivo  Memorizza Lunghezza

Attiva D  incremento  Memorizza Incremento

Origine O

Numero O  offset aggiuntivo  Memorizza Quota

Attiva O

Piano di lavoro

Esecuzione: AUTOMATICA Funzionamento: AUTOMATICA

N bl. Tempo esec. (00:00:00)

T 0 D 0

O 0 Piano

CL 0.000 CR

F (mm/min) 5000 S (RPM)

G e M modali

G 17 G71 G17 G80 G40 G90 G50 G79 G75

M M05 M09

Variabile P120 0 Variabile G 0

N100 G02 Z160 X=Z0 I=Z0 KU ;358.568

N110 G01 Z160 X-100 ;358.568

N120 G01 Z0 X-100 ;270

N130 G01 Z0 X0 ;180

;90

Speed = 100% 48 1.015 0.203 17/10/2006 12:08:13

# ISO G\_CODE HMI Editor Fresca

The screenshot displays the AxesBrainStation software interface for editing ISO G-code. The main window shows the following G-code:

```
P3=12  
F25000  
  
M102  
X100Y50C180Z0R  
  
M101  
L=YY  
  
G2 X100Y50 I50J50 C-360 Z10I  
  
{P3>1}YY
```

The interface includes a menu bar (File, Tastiera, ?), a status bar (Macchina OK.., Feed 100%), and a sidebar with various tool icons. The sidebar contains the following items:

- Sistema
- Automazione
- CNC
- CNC
- Editor
- Grafica a punti 2D
- Grafica CNC
- Setup
- Convertitore DXF->ISO
- Visione
- Transfer
- Gestore cella
- Tomio

At the bottom, there is a row of function keys (F1-F12) and a status bar with the following information:

- Speed = 100%
- 55
- 1.010
- 0.201
- 17/10/2006
- 12:09:06

# ISO G\_CODE HMI CNC Tornio

**Programma: Asferica.PRG**

Posizione attuale (mm): X **-93.54360**, Z **-51.70340**

Distanza finale (mm): X 0.00000, Z 0.00000

N bl.	Tempo exec. (00:03:05)	00:03:05	F	0
T 0	D 0			
O 0	Piano	Z-X		
CL 0.000	CR	0.000	S	0
F (mm/min) 12	S (RPM) 6000			

G e M modali

G 92	G71 G18 G80 G40 G90 G50 G08 G79 G75
M 30	M05 M09 M30

MAN Disattivo HOLD Variabile L0 0 Variabile E0 0

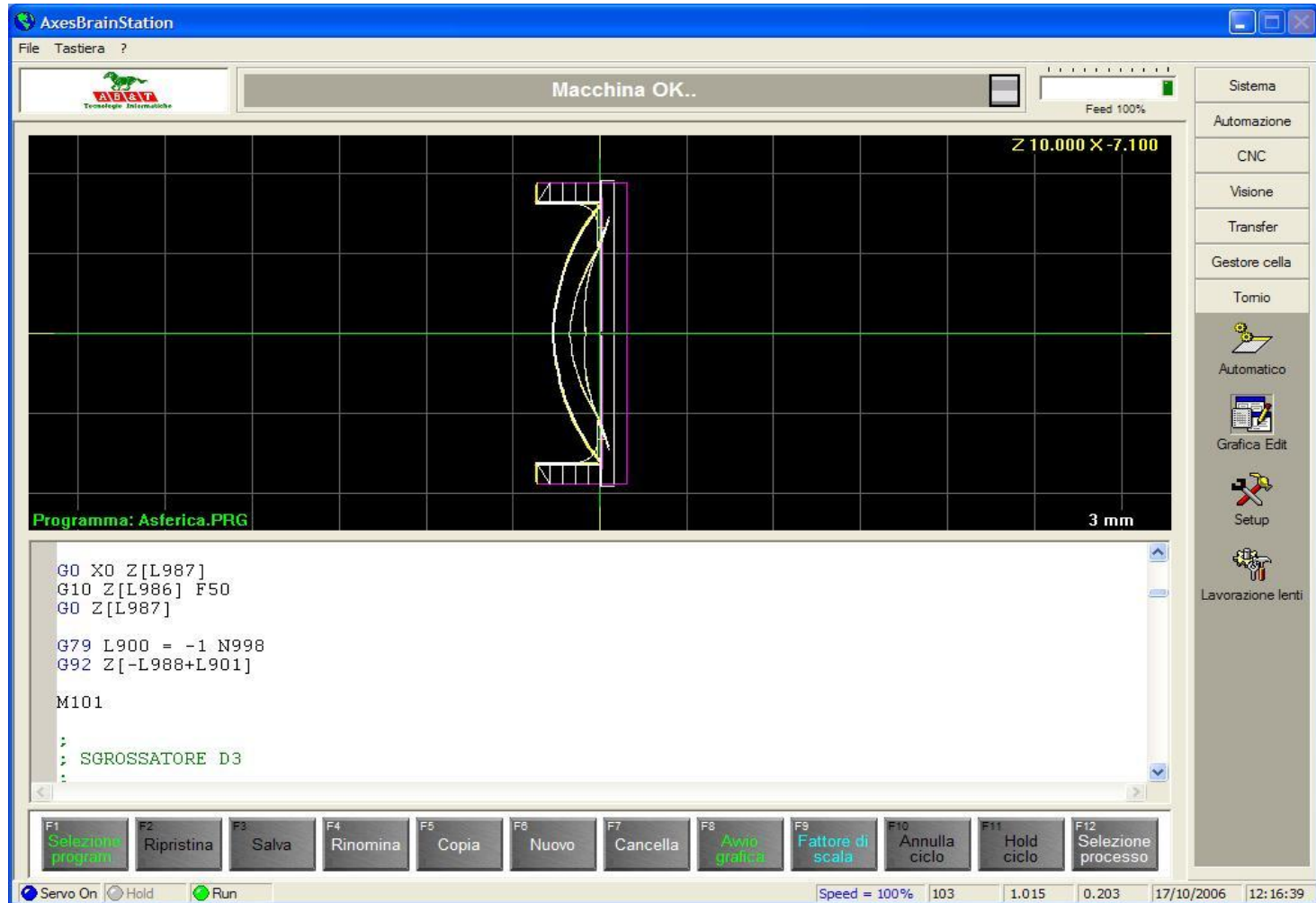
Coppia (%)

Esecuzione: AUTOMATICA Funzionamento: CNC Programma: ASFERICA.PRG Inizio bl. 0 Nome processo: A2

F1 Selezione program F2 Avvio ciclo F3 MDI F4 COMM F5 Manuale F6 Imposta pezzo F7 Modalità ciclo F8 Annulla ciclo F9 Hold ciclo F10 BREAK F11 Fattore di scala F12 Visualizza Edit

Servo On Hold Run Speed = 100% 97 1.012 0.206 17/10/2006 12:15:52

# ISO G\_CODE HMI Editor Tornio



# Motori passo e pilotaggio in frequenza

Il pilotaggio in frequenza del sistema di azionamento motore è ormai uno standard e consiste in due ingressi per l'azionamento a 5Volt, un segnale chiamato DIR rappresenta la direzione di rotazione del motore se 5Volt (uno logico) rotazione oraria se 0Volt (zero logico) rotazione antioraria, un secondo segnale chiamato PULSE che ad ogni cambiamento di stato da 0 a 5Volt provoca un scatto di una frazione di passo del motore (sottopasso), modalità chiamata **pilotaggio in frequenza**.

Nella personalizzazione dell'azionamento, si dichiara in quanti sottopassi viene diviso il passo (STEP), si arriva fino a 256 sottopassi in quasi tutti gli azionamenti, normalmente per effettuare un giro di motore sono necessari circa 200 passi, abbiamo così che con un divisore 256 il giro motore è suddiviso in 51200 parti.

Se ad giro motore abbiniamo un rapporto 10mm, abbiamo un sottopasso (un pulse) pari a 0.0002 mm, il tutto con una tecnologia estremamente economica, costo di un azionamento-passo a partire da 120€ e di un motore da 0.5 Nm a partire da 40€ con un costo cavi azionamento motore nemmeno da paragonare rispetto alle altre soluzioni, arrivando così a meno di 200€, l'equivalente di azionamento/motore brushless e cavi si arriva a circa 800€, dove però le informazioni della posizione sono disponibili (encoder o resolver sul motore).



# Micropasso per motori stepper

Per mantenere l'economicità dell'applicazione **non si abbina un** "encoder" o una scala per leggere la posizione, si perde di conseguenza un riferimento di precisione come il "marker", ma si ci si deve accontentare della precisione e la ripetibilità di un segnale dato da micro interruttore meccanico od elettronico, questa carenza unita alla possibile perdita di passo dovuta ad un eventuale duro meccanico fa sì che non può essere usato come asse di lavorazione soprattutto ad alta velocità, ma solo come asse di servizio.

Un'altra caratteristica degli azionamenti dei motori passo da tener conto è la frequenza massima del segnale di PULSE, attualmente un buon azionamento arriva come frequenza massima di 150Khz, vediamo cosa significa nel nostro esempio 150000 impulsi in un secondo significano 3 giri/secondo quindi nel nostro esempio di 30mm al secondo, oppure 1.8 m/min che è una velocità molto bassa, per alzarla bisogna usare un divisore del passo molto minore diciamo di 16 invece di 256, il che ci permette di arrivare ad una velocità ragionevole di 28.8 m/minuto però la precisione del posizionamento è passata da 0.0002mm a 0.0015mm.

Se passiamo un ottavo di passo avremo 57.6 m/minuto mm a 0.003 mm che rimane un ottimo compromesso, l'unica pecca è che man mano si scende con la suddivisione del passo si perde nella prestazione del motore.



# Pilotaggio in frequenza per gli azionamenti dei motori

Il pilotaggio dei motori utilizzando una frequenza d'impulsi anziché un riferimento analogico  $\pm 10\text{V}$  è sempre stato adottato dai costruttori giapponesi già dagli anni 60, questo permette **di non avere** sul CONTROLLO il controllo di posizione PID, semplificandolo di molto.

Oggi moltissimi costruttori danno questa possibilità di pilotaggio in alternativa al riferimento in tensione, allargando così l'utilizzo della movimentazione assi anche ai PLC di fascia bassa.

Il controllo in frequenza in fondo è anche lo standard universale per dire quante unità di posizionamento fare al azionamento ( controllo in posizione anziché in velocità ). Quindi il pilotaggio in frequenza, più direzione, ha uniformato i motori tradizionali ai motori passo e permesso il controllo del movimento anche ai PLC di fascia bassa. Rimangono due problematiche in questa modalità di pilotaggio:

1) La tensione di  $5\text{V}$  dei due segnali non permette distanze superiori di 1 o 2 metri tra controllo ed azionamento.

2) Il segnale PULSE con frequenze che possono arrivare a  $250\text{KHz}$  è facilmente disturbabile

da fonti esterne, con una drammatica influenza sul posizionamento finale.

# Motion Control 1 parte

Il motion control deve poter operare in multi-task , su agglomerati di assi definiti da “handler” dinamici a cui si fa riferimento con dei comandi di movimentazione.

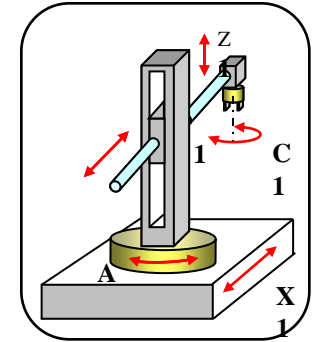
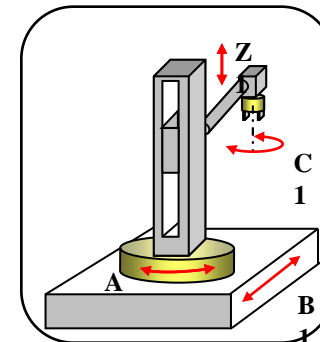
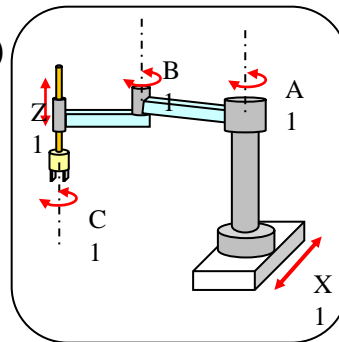
Un comando di movimentazione viene espletato nelle seguenti fasi:

a) **Interpolazione comando di movimento** in cui intervengono sugli assi interessati gli algoritmi di:

- a.1) Interpolazione lineare, circolare, elicoidale, “spline”, “surf”
- a.2) Anticollisione tra gli assi.
- a.3) “Gantry.
- a.4) Camme elettroniche

b) **Trasformate nel piano di lavoro** per cui possiamo avere diverse tipologie di configurazioni o robot:

- b.1) SCARA
- b.2) SCARA PANTOGRAFO
- b.3) Polare
- b.4) Polare canotto



# Motion Control 2 parte

## c) **Calcolo posizione teorica istantanea**

Nel calcolo della posizione istantanea bisogna rispettare le leggi del moto accelerando, mantenere la velocità programmata ed infine decelerare.

Le accelerazioni e decelerazioni possono essere:

c.1 ) Lineare

c.2) "S\_Curve"

c.3) Sinusoidale

d) **Correzione della posizione teorica** con la matrice, o i vettori di compensazione lineare, quadratura, rollio e beccheggio.

e) **Controllo della posizione** naturalmente questa funzionalità è necessaria se il comando che viene dato all'azionamento è di velocità o coppia.

Normalmente questa funzionalità è realizzata con il metodo di retroazione PID ( Proporzionale Integrativo Derivativo

# DCOM

*Il prodotto “AxesBrain” è stato sviluppato per offrire dei servizi di movimentazione agli applicativi utente, utilizzando la tecnologia “DCOM”, sarà così possibile avere l’accesso a questi servizi non solo dallo stesso PC, ma anche da PC integrati in rete locale o collegati tramite Internet.*

Per utilizzare i servizi di movimentazione possono essere utilizzate essenzialmente due strade o utilizzando le funzioni dirette alle risorse assi, mandrini e i segnali di ingresso uscita oppure avvalendosi di cicli programmati. Tramite i linguaggi di programmazione vengono definite delle procedure o cicli di percorsi degli assi e di operazioni di manipolazione e lavorazione, nel nostro caso abbiamo tre linguaggi disponibili: AxesBrainL, AxesBrainISO e AxesBrainAWL che l’utente può utilizzare a seconda del tipo di applicazione che si viene a presentare, è inoltre possibile integrare come libreria , delle funzionalità o addirittura dei processi per la gestione di cicli di lavorazione personalizzati ad esigenze specifiche, completamente sviluppate dall’utente.

I servizi di movimentazione sono:

- 1 FUNZIONI dirette alle risorse come assi, I/O digitali ed analogici
- 2 CICLI di lavorazione e manipolazione con programmazione tramite i due linguaggi GP-PLC e CNC ISO G\_CODE