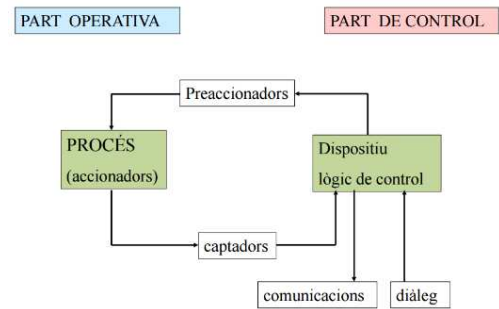
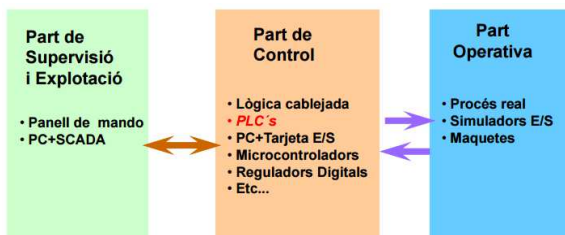
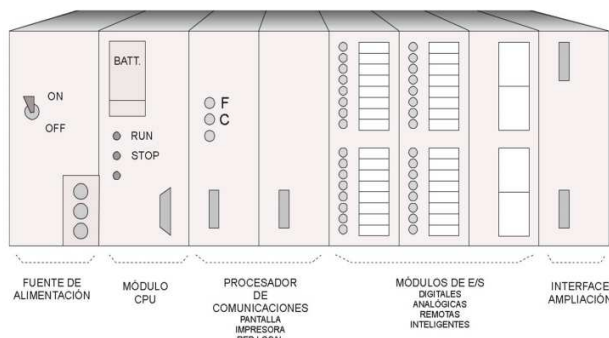


**Autòmats Programables –Entorn i aplicacions–.**

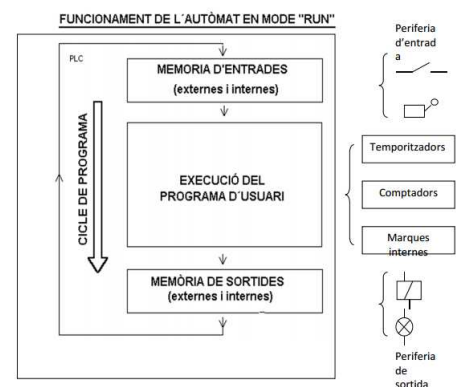
## 1. Sistema d'automatització components habituals



## 2. Arquitectura d'un autòmat



## Com treballa un autòmat?

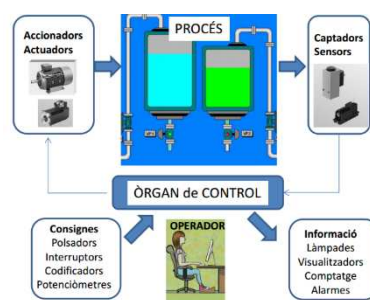


## 3. Sortides i entrades analògiques

En un procés en 2 sentits:

- Transferir a
- Aplicar les
- través del

Tant els senyals Analògiques o On-Off.

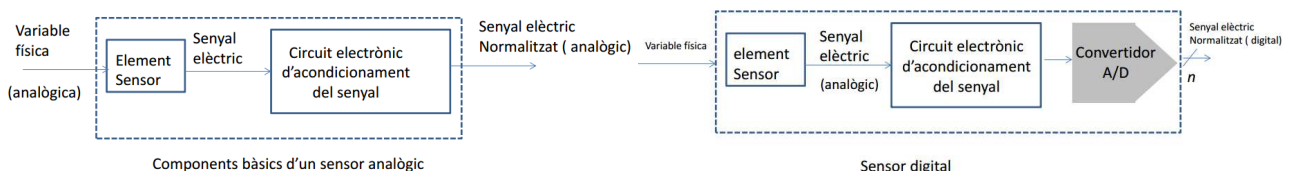


d'automatització es pretén un pas d'informació

l'autòmat la informació del procés variables de sortida de l'autòmat cap al procés a corresponents actuadors.

elèctrics d'entrada com els de sortida poden ser:

- Els processos físics en general són analògics → la majoria de sensors proporcionen senyals analògics.
- Problema: presència de soroll, interferències i distorsió.
- Solució: Circuit d'acondicionament del senyal.

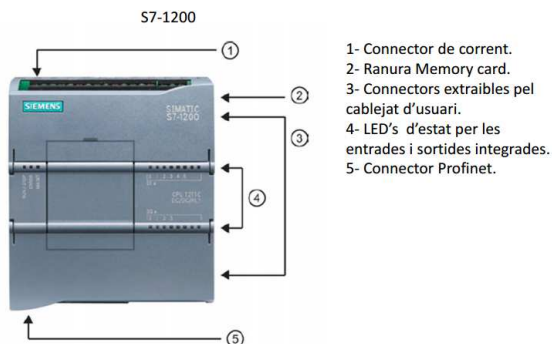


## Autòmat Siemens

Siemens disposa de 4 famílies de PLC's (entre d'altres, com el LOGOm el ET200, S7,...), anomenades Simatic.

- **S7-200:** Per aplicacions petites. Format compacte. Ampliable amb mòduls. Connexió RS485. El programaper treballar, en "Micro/WIN" és fàcil de treballar i amb possibilitat de programació amb diferents llenguatges.
- **S7-300:** Per aplicacions mitjanes.
- **S7-400:** Per aplicacions grans o complexes.
- **S7-1200:** Gamma alta
- **S7-1500:** Gamma alta (control lògic i de moviment) –Motion Control-.

### Autòmat Simenns S7-1200:



SIMATIC S7-1200 with the connected bus systems PROFIBUS, AS-Interface and PROFINET by Siemens

Existeixen mòduls addicionals per a la comunicació en xarxes PROFIBUS, GPRS, RS485 o RS232.

La CPU S7-1200 està formada per: un microprocessador, una font d'alimentació integrada, circuits d'entrada i sortida, PROFINET integrat, E/S de control de moviment d'alta velocitat i entrades analògiques incorporades.

Els models de CPU: 1211C 1212C **1214C** 1215C 1217C

AC/DC/Rly: alimentació amb alterna/sortida contínua DC/les sortides tenen connexió de relé o contacte lliure de potencial RLY.

LED's d'estat: Donen informació de l'estat de funcionament de la CPU, de l'estat de les entrades i sortides i de les fallades del sistema.

1. RUN: l'autòmat executa cíclicament les instruccions del programa d'usuari. (verd).
2. MAINT: LED de manteniment (fa pampallugues, ex: quan hi ha problema amb memory card)
3. STOP: l'autòmat està engegat però el programa d'usuari no s'executa. S'hi ubiquen també els potenciòmetres analògics i la connexió d'ampliació.(groc).
4. ERR: - Llum vermella intermitent indica un error de la cpu o memory card o de configuració.
  - Llum vermella permanent indica que hi ha un error de hardware.
  - Tots els LED's fan pampallugues si es detecta defeca en el fireware.

Memòria de dades:



### Distribució de la memòria

- Memòria de programa: Conté les operacions d'esquema de contactes (KOP), de la llista d'assignació (AWL) o altres, que executa l'autòmat programable per a l'aplicació desitjada.
- Memòria de paràmetres: Permet emmagatzemar determinats paràmetres configurables, tal com contrasenyes, direccions d'estacions i informacions sobre les àrees romanents.
- Memòria de dades: La memòria de dades és l'àrea de treball a la que accedeix el programa d'aplicació (també denominat programa d'usuari).

### Adreces de memòria en els PLC'S del laboratori:

<u>14 entrades digitals</u>	<u>10 sortides digitals</u>	<u>2 entrades analògiques</u>
%I0.0	%Q0.0	%IW64
%I0.1	%Q0.1	%IW66
%I0.2	...	
...	%Q0.7	
%I0.7	%Q1.0	
%I1.0	%Q1.1	
...		
%I1.5		

### Marques de sistema:

Una marca de sistema és una marca amb valors definits. En la parametrització de la marxa de sistema es determina el byte de marques de la CPU que es convertirà en el byte de marques de sistema. Aquestes marques s'habiliten en les propietats de la CPU.

Tabla 4- 16 Marcas de sistema

7	6	5	4	3	2	1	0
Reservado Valor 0				Siempre off Valor 0	Siempre ON Valor 1	Indicador de estado de diagnóstico • 1: Cambiar • 0: No cambiar	Indicador de primer ciclo • 1: Primer ciclo tras arranque • 0: No es primer ciclo

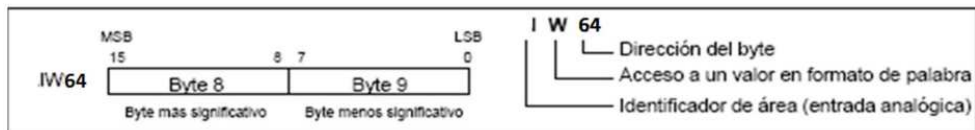
El byte de marques seleccionat no pot utilitzar-se per guardar dades en la memòria immediata.

### Marques de cycle:

- Tot bit del Byte de marques de cycle genera un impuls d'ona quadrada
- El byte de marques de cycle permet 8 freqüències diferents compreses entre 0.5Hz i 10Hz
- Aquest bits poden ser utilitzats com a bits de control per a disparar accions cícliques en el programa d'usuari (és habitual combinar-les amb instruccions de detecció de flanc).
- Ja que la marca de cycle és asíncrona respecte al cycle de la CPU, l'estat de la marca del cycle pot canviar varies vegades durant el cycle llarg.

Adreçament de les entrades analògiques:

- Els valors de les entrades analògiques es transformen en paraules (2bytes), és a dir, és un conversorAD de 16 bits.
- Donat que cada entrada analògica ocupa dos bytes aquestes es numeren per nombres parells (IW64, IW66).
- Les entrades analògiques sempre s'accedeixen en format word.

Cicle de tractament dels senyals d'entrada/sortida:

- Abans de l'execució del programa d'usuari, la CPU consulta els estats de les entrades físiques i carrega amb ells la memòria imatge d'entrades.
- Durant l'execució del programa d'usuari, la CPU realitza els càlculs a partir de les dades de la memòria i relés interns. El resultat d'aquests càlculs queda dipositat en la memòria imatge de sortides.
- Finalitzada l'execució, la CPU transfereix a les interfícies de sortida els estats de les senyals contingudes en la memòria imatge de sortida quedant el sistema preparat per a començar un nou cicle.

E/S integrades i mòduls d'expansió:

- Les CPU'S disposen d'un cert nombre d'E/S integrades, l'adreçament d'aquestes és fixa i no es pot modificar.
- A partir de la CPU (mòdul 0) es poden afegir mòduls addicionals (mòdul 1,2,...N). L'adreçament d'aquest mòduls addicionals és variable i depèn de l'aposiició que ocupen respecte l'0anterior mòdul del mateix tipus i els seu tipus (entrades, sortides, digitals, analògiques...)

Assignació d'adreces als mòduls:

## 1. Entrades digitals:

- L'adreça inicial depèn només de l'últim mòdul d'entrades digitals.
- Si una adreça no té entrada física en el mòdul, aquesta adreça es perd i no pot ser utilitzada en l'àrea de programa.
- Les adreces no assignades a cap mòdul poden ser utilitzades com a marques internes.

## 2. Sortides digitals:

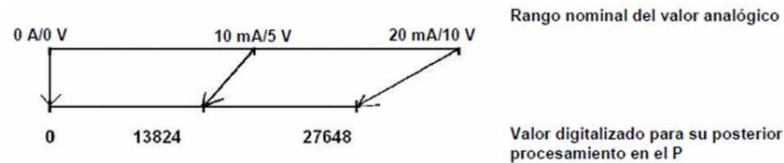
- L'adreça inicial depèn de l'últim mòdul de sortides digitals.
- Tota adreça no assignada (ja sigui per falta d'algun punt físic en els mòduls o per falta del mòdul sencer) es pot utilitzar coma marca del programa.

## 3. Entrades/Sortides analògiques:

- L'adreça inicial depèn de l'últim mòdul del mateix tipus.
- Les adreces no assignades es perden completament.

Normalització dels valors analògics:

Si el valor analògic es troba en forma digitalitzada, normalment es requereix una normalització per tal que els valor numèrics coincideixin amb les magnituds físiques en el procés. Per la normalització es fan operacions aritmètiques, així doncs caldrà treballar en dades del tipus REAL.



### Subrutines

- Les subrutines s'utilitzen per a estructurar o dividir el programa en blocs més petits, més fàcils de gestionar.
- La CPU també es pot utilitzar més eficientment, invocant al bloc només quan es necessiti, enlloc d'executar tots els bloc en cada cicle.
- Les subrutines es poden transportar si es fa referència únicament als seus paràmetres i a la seva memòria local.
- Per a que una subrutina es pugui transportar, s'ha d'evitar la utilització de variables /símbols globals (direccions absolutes en les àrees de memòria I,Q,M,SM,AI,AQ,V,T,C,S,AC).
- Si la subrutina no té paràmetres de crida (IN,OUT ó IN\_OUT), o si utilitza únicament variables locals en la memòria L, la subrutina es pot exportar i importar d'un projecte diferent.

### Tipus de subrutines:

1. OB: Defineix l'estructura del programa (temps d'execució per defecte es de 150ms-ajustambe 1ms-6s CPU config).
2. FB i FC: Contenen el codi de programa corresponents a tasques específiques o combinacions de paràmetres. Proporcionen paràmetres d'IN/OUT intercanviant dades amb el bloc invocant.
3. DB: Emmagatzemen dades que poden ser utilitzades pels blocs del programa.

### Rutines d'interruptió:

Abans de poder cridar a una rutina d'interruptió és necessari establir un enllaç entre l'event d'interruptió i l'apart del programa que es desitgi executar quan es presenti l'event.

L'operació Associar interruptió serveix per a assignar l'event d'interruptió a una part del programa.

També és possible associar varis events d'interruptió a una única rutina d'interruptió. Pel contrari, no es pot associar un sol event a varies rutines.

Quan s'associa un event a una rutina d'interruptió, s'habilita automàticament l'event. Si s'inhibeixen tots els events d'interruptió, es posarà en cua d'espera fins que les interrupcions s'habilitin de nou, utilitzant per a tal la operació Habilitar tots els events d'interruptió.

També és possible inhibir certs events d'interruptió eliminant l'associació entre l'event i la corresponent rutina. Aquesta operació retorna la interrupció a un estat inactiu o ignorat.

Evento	Descripción
0	I0.0 Flanco positivo
1	I0.0 Flanco negativo
2	I0.1 Flanco positivo
3	I0.1 Flanco negativo
4	I0.2 Flanco positivo
5	I0.2 Flanco negativo
6	I0.3 Flanco positivo
7	I0.3 Flanco negativo
8	Puerto 0 Recibir carácter
9	Puerto 0 Transmisión finalizada
10	Interrupción temporizada 0 SMB34
11	Interrupción temporizada 1 SMB35
12	HSC0 CV=PV (valor actual = valor predeterminado)
13	HSC1 CV=PV (valor actual = valor predeterminado)
14	HSC1 Cambio de sentido
15	HSC1 Puesto a 0 externamente
16	HSC2 CV=PV

Regles per a la utilització correcta de les interrupcions:

1. El processament d'interrupcions permet reaccionar ràpidament davant determinats events interns o externs. Les rutines d'interrupció s'han d'estructurar de forma que, un cop executades determinades tasques, tornin el control al programa principal.
2. Per això és convenient crear rutines d'interrupció curtes amb indicacions precises, de manera que es puguin executar ràpidament sense interrompre altres processos durant períodes massa llargs.
3. Si no es tenen en compte aquestes mesures, és possible que es produeixin estats imprevistos que puguin afectar a la instal·lació controlada pel programa principal.

#### **Tipus de rutines d'interrupció:**

- Interrupcions de comunicació: El port sèries. La comunicació a través d'aquest port es denomina mode Freeport (comunicació programable per usuari). En mode Freeport, el programa defineix la velocitat de transferència, els bits per caràcter, la paritat i el protocol.  
Les interrupcions de transmissió i recepció permeten controlar la comunicació mitjançant el programa.
- Interrupcions E/S: cobreixen interrupcions en flancs positius i negatius, interrupcions dels comptadors ràpids, així com interrupcions de la sortida d'impulsos.
- Interrupcions temporitzades: S'utilitzen per a indicar àrees que s'hagin d'executar cíclicament (inclou els temps T32/T96).

S'utilitzen per:

1. Controlar el mostreig de les entrades analògiques
2. Executar un bucle PID en intervals regulars.

Associar un event d'interrupció temporitzat a una rutina d'interrupció, s'habilita l'event i immediatament es comença a temporitzar.

Per a poder modificar el temps de cicle s'haurà de canviar el valor del mateix i reassociar la rutina d'interrupció a l'event de la interrupció temporitzada. En reassociar-se la rutina d'interrupció, la funció borra el temps acumulats de l'associació anterior, amb el que es trona a temporitzar a partir del nou valor .

Un cop habilitada, la interrupció funciona de forma continua executant la rutina associada cada cop que transcorri l'interval de temps indicat.

La interrupció temporitzada s'inhibeix sortint del mode RUN o desassociant-la de la rutina corresponent

#### **Prioritats de la rutina d'interrupció:**

Interrupcions de comunicació (prioritat alta), interrupcions E/S, interrupcions temporitzades (més baixa). La CPU processa les interrupcions segons la seva prioritat i després en l'orde en que apareixen. Només s'executa una ordre d'interrupció en cada cas. Les noves interrupcions que apareguin mentre s'executa una altre interrupció es posen en cua d'espera pera a ser processades posteriorment.

#### **Alarma cíclica:**



El temps d'execució d'un OB d'alarma cíclica ha de ser molt inferior al seu període. Si no hagués acabat encara i li toqués tornar-se a executar s'arrencaria la OB d'error de temps. (Si no existeix la CPU passa a mode STOP).

Les OB's d'alarma cíclica serveixen per iniciar programes en intervals periòdics, independentment de l'execució cíclica dels programa. El temps d'arranc d'una OB d'alarma cíclica s'indiquen mitjançant el període i el desfase.

- Període: defineix l'interval en el que s'arranca la rutina (és un múltiple sencer de cicle de base d'1ms).
- Desfase: temps de desfase del temps d'arranc respecte al cicle base.

Temps de cicle ajustat ha d'incloure:

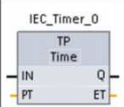



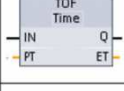



- Temps de processament del OB1 i actualització de les imatges.
- Temps de processament d'interrupcions d'alta prioritat.
- Processos de comunicació del sistema operatiu.
- Accés a la CPU des d'un dispositiu de programació.

### Bloc PID:

La necessitat de regular un procés per tal que funcioni segons unes especificacions pot fer-se de forma analògica o de forma digital.

### Temporitzadors:

- Les instruccions amb temporitzadors s'utilitzen per crear retard programats.
- Cada temporitzador utilitza una estructura de DB del tipus de dades IEC-Timer de 16 bytes per guardar la informació del temporitzador especificada a sobre de la instrucció de quadre o bobina.
- Step crea automàticament el DB en introduir la instrucció.

Cuadros KOP / FUP	Bobinas KOP	SCL	Descripción
		<pre>"IEC_Timer_0_DB".TP (   IN:=_bool_in_,   PT:=_time_in_,   Q=&gt;_bool_out_,   ET=&gt;_time_out_);</pre>	El temporizador TP genera un impulso con una duración predeterminada.
		<pre>"IEC_Timer_0_DB".TON (   IN:=_bool_in_,   PT:=_time_in_,   Q=&gt;_bool_out_,   ET=&gt;_time_out_);</pre>	El temporizador TON pone la salida Q a ON tras un tiempo de retardo predeterminado.
		<pre>"IEC_Timer_0_DB".TOF (   IN:=_bool_in_,   PT:=_time_in_,   Q=&gt;_bool_out_,   ET=&gt;_time_out_);</pre>	El temporizador TOF pone la salida Q a OFF tras un tiempo de retardo predeterminado.
		<pre>"IEC_Timer_0_DB".TONR (   IN:=_bool_in_,   R:=_bool_in_,   PT:=_time_in_,   Q=&gt;_bool_out_,   ET=&gt;_time_out_);</pre>	El temporizador TONR pone la salida Q a ON tras un tiempo de retardo predeterminado. El tiempo transcurrido se acumula a lo largo de varios periodos de temporización hasta que la entrada R inicializa el tiempo transcurrido.

Parámetro	Tipo de datos	Descripción
Cuadro: IN Bobina: Flujo de corriente	Bool	TP, TON, y TONR: Cuadro: 0=deshabilitar temporizador, 1=habilitar temporizador Bobina: Sin flujo de corriente=deshabilitar temporizador, flujo de corriente=habilitar temporizador TOF: Cuadro: 0=habilitar temporizador, 1=deshabilitar temporizador Bobina: Sin flujo de corriente=habilitar temporizador, flujo de corriente=deshabilitar temporizador
R	Bool	Solo cuadro TONR: 0=Sin inicialización 1= Inicializar el tiempo transcurrido y el bit Q a 0
Cuadro: PT Bobina: "PRESET_variable"	Time	Cuadro o bobina de temporizador: Entrada de tiempo predeterminado
Cuadro: Q Bobina: DBdata.Q	Bool	Cuadro de temporizador: salida de cuadro Q o bit Q en los datos del DB de temporizador Bobina de temporizador: sólo se puede direccionar el bit Q en los datos del DB de temporizador
Cuadro: ET Bobina: DBdata.ET	Time	Cuadro de temporizador: salida de cuadro ET (tiempo transcurrido) o valor de tiempo ET en los datos del DB de temporizador Bobina de temporizador: sólo se puede direccionar el valor de tiempo ET en los datos del DB de temporizador.

Els valors PT (temps predeterminat) i ET (temps transcorregut) s'emmagatzemen en les dades de DB IEC\_TIMER com sencers dobles amb signe.

- > T#3S : tres segons
- > T#1M10S: un minut i deu segons
- > T#100Ms: cent milisegons.

#### NA PORTAL

Creació d'un projecte: crear nou projecte → configurar dispositiu → afegir dispositiu → configuració de l'adreça IP → Programació.

Creació d'una subrutina: Declaració de les variables locals a aquesta funció → escriure el programa en el llenguatge triat (KOP-llenguatge de contactes) Es fan servir les variables referint-nos a elles pel símbol # → Crida de la rutina FC! Des del Main. Arrosseguem el bloc FC! Cap al codi de la rutina Mani → En el main (OB1) cal interconnectar les variables que entren i surten de la funció FC1 amb les variables del PLC → descarregar el programa cap el PLC → execució, autòmat en mode Run.

	Tipus	Denominació	Funció	Disponible en
Paràmetres de bloc per la crida des dels diferents programes	Paràmetres d'entrada	Input	Paràmetres els valors dels quals són llegits pel bloc	-Funcions -Blocs de funcions -Blocs d'organització(*)
	Paràmetres de sortida	Output	Paràmetres els valors dels quals escriu el bloc	-Funcions -Blocs de Funcions
	Paràmetres d'entrada/sortida	InOut	El bloc llegeix el valor d'aquests paràmetres en fer la crida i el torna a escriure en ells un cop acabada l'execució	-Funcions -Blocs de funció
Dades locals per guardar dades intermitges	Dades locals temporals	Temp	Variables utilitzades per guardar resultats intermitjos temporals. (**)	- Funcions - Blocs de funcions - Blocs d'organització
	Dades locals estàtiques	Static	Variables utilitzades per guardar resultats intermitjos estàtics en el bloc de dades.(***)	- Blocs de Funcions

(\*) Només alguns tipus de blocs d'organització.

(\*\*) Les dades temporals es guarden només 1 cicle.

(\*\*\*)Les dades estàtiques es conserven fins que es torna escriure sobre elles, també al llarg de varis cicles.

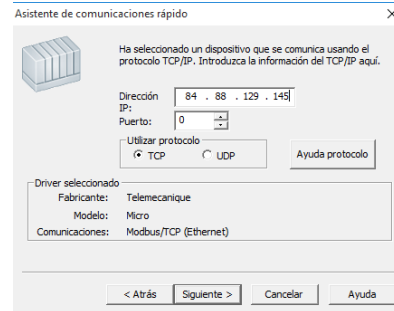
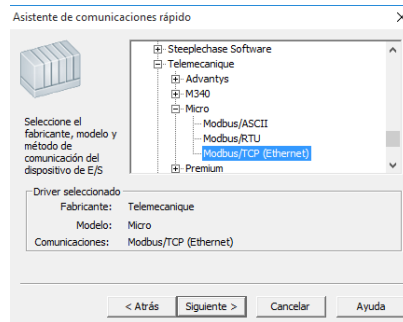
## Vijeo-CITECT 7.20

Errades típiques:

- No posar contrasenya.
- No definir usuari.
- Crear mes d'un servidor.

Configuració rapida d'E/S amb un PLC: Crear un nuevo servidor → crear un nuevo dispositiva de E/S (nom PLC) → seleccionar el tipus de dispositiu de E/S (si es vol amb PLC és externo) → introduir el TCP/IP → següent, següent i finalitza.





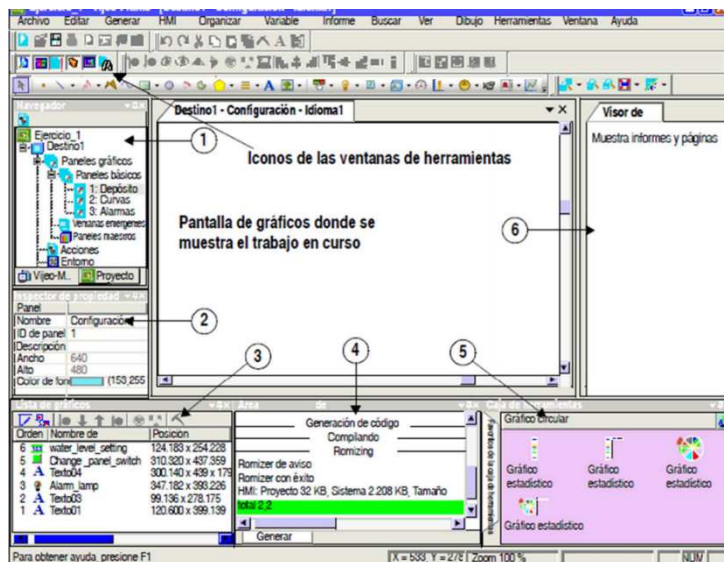
### PANTALLAS TÀCTILS VIJEO DESIGNER

- Permeten la creació de panells d'operador.
- Permeten la configuració dels paràmetres operatius per dispositius d'Interfície Home – Màquina.
- Interessa treballar amb una aplicació Software que proporcioni les eines necessàries pel disseny d'un projecte HMI. Cal que aquest Software permeti des de l'adquisició de dades fins a la creació i la visualització de sinopsis animades.

Amb Vijeo Designer, l'usuari pot configurar un panell HMI per a comunicar-se de forma simultània amb varis dispositius de Schneider Electric diferents i també amb dispositius d'altres fabricants. La finestra "Inspector" de propietats, es poden configurar o modificar les variables i característiques dels objectes.

Els tipus que permeten interactuar amb Vijeo Designer:

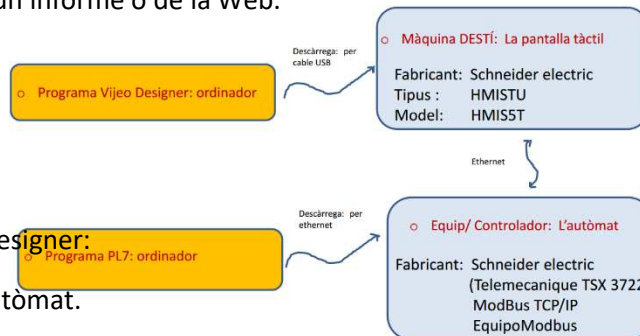
- Llista de gràfics
- Zona de retroalimentació
- Inspector de propietats
- Caixa d'eines
- Navegador
- Visor d'informació



- 1. Navegador:** Utilitzat per a crear aplicacions. La informació relativa a cada projecte s'ordena de forma jeràrquica en un explorador de documents.
- 2. Inspector de propietats:** Mostra els paràmetres de l'objecte seleccionat (si es selecciona més d'un objecte, només es mostraran els paràmetres comuns a tots els objectes seleccionats).
- 3. Llista de gràfics:** Ofereix una llista amb tots els objectes que apareixen en el dibuix i indica:

4. **Zona de retroalimentació:** Mostra la progressió i resultat de la programació dels errors, de la compilació i de la càrrega.
5. **Caixa d'eines:** Biblioteca de components que subministra el fabricant o que s'han creat amb anterioritat per l'usuari.
6. **InfoViewer:** Mostra el contingut d'un informe o de la Web.

### Pantalla tàctil del laboratori:



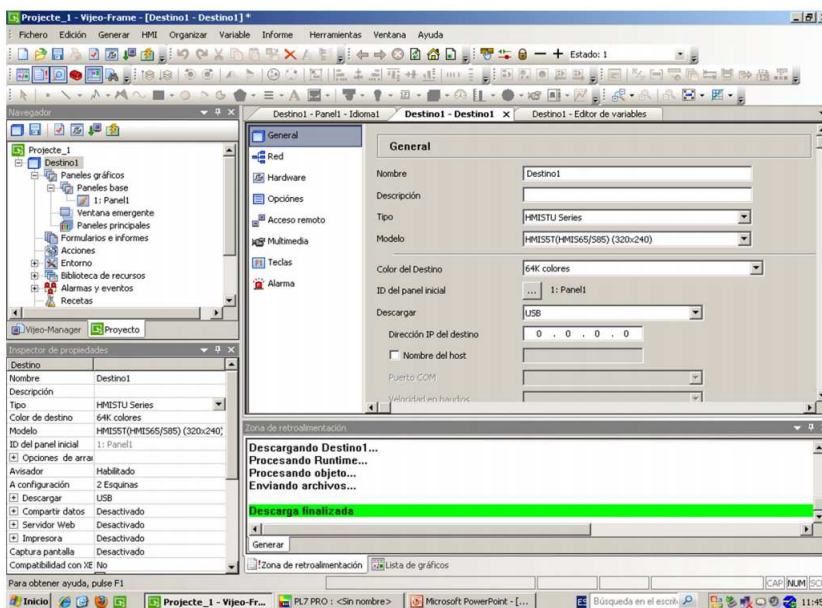
Com construir un projecte amb Vijeo-Designer:

Etapa inicial 1: sense connexió amb l'autòmat.

Etapa inicial 2: amb connexió amb l'autòmat.

Etapa de desenvolupament de l'aplicació : creació del projecte.

#### 1. Configuració de la pantalla tàctil.



#### 2. Configuració de l'autòmat – controlador-

1. Crear nou projecte
2. Seleccionar la direcció IP si el model utilitza un port Ethernet i, a continuació, fer clic en **Següent**.
3. Seleccionar el controlador adequat pel tipus de dispositiu amb el botó **Afegir**. En el nostre exemple, seleccionar:  
Fabricant: **Schneider Electric**  
Controlador: **Modbus**  
Equip: **Equipo Modbus**
4. Configurar la Sintaxis de IEC 61131 en el quadre Configuració de l'equip (per poder utilitzar les adreces de memòria internes %Mi)  
Per a fer-ho:  
\*Finestra Navegador → fer doble clic en l'equip → equipo Modbus00 → Configuració del Sistema → Després seleccionar la casella [Sintaxis de IEC61131].

#### 5. Guardar el projecte



### 3. Creació de variables.

#### Creació de variables

Una variable és una adreça de memòria que s'indica mitjançant un nom. Vijeo Designer gestiona els següents tipus de variables:

BOOL (variable booleana)  
 INT (sencer amb signe de 16 bits)  
 UINT (sencer sense signe de 16 bits)  
 DINT (sencer amb signe de 32 bits)  
 UDINT (sencer sense signe de 32 bits)  
 REAL  
 STRING

Tipus de dades de l'usuari (matriu o estructura)

Bloc INT (sencer amb signe de 16 bits)  
 Bloc UINT (sencer sense signe de 16 bits)  
 Bloc DINT (sencer amb signe de 32 bits)  
 Bloc sencer (bloc sencer genèric de 1 a 32 bits)  
 Bloc REAL

Vijeo Designer utilitza les variables per a comunicar-se amb dispositius. També es poden definir variables internes que només seran utilitzades per Vijeo Designer.

#### Com crear una variable quan volem relacionar amb variables de l'autòmat.

1. Fer clic amb el botó dret del mouse en el node "Variables" en la finestra "Navegador", seleccionar "Nova variable" i fer clic en "BOOL" (o el tipus que vulguem).
2. En l'Inspector de Propietats de les Variables, canviar el nom de la variable tipus Booleà, "BOOL01" a "High\_level". Especificar l'origen de la variable (extern en aquest cas). En la propietat "Dirección del dispositivo", escriure %M0.
3. Per a poder escriure %M0 com la sintaxi de l'adreça, cal haver configurat la propietat [Sintaxis de IEC61131] en el quadre Configuració de l'autòmat.

### PAUTES DE REALITZACIÓ DEL PROJECTE/TREBALL DE FI DE CARRERA

Un Projecte/ Treball fi de carrera s'estructura en 5 documents principals. Aquest documents s'ordenen i enumeren seguint la següent estructura:

1. Memòria
  1. Introducció
    - 1.1 Antecedents
    - 1.2 Objecte
    - 1.3 Especificacions i abast
2. Plànols
 

.....

n

n+1. Resum del pressupost

n+2. Conclusions

n+3. Relació de documents

n+4. Bibliografia

n+5. Glossari.

Annexos a la memòria

3. Plec de condicions

4. Estat d'amidaments

5. Pressupost