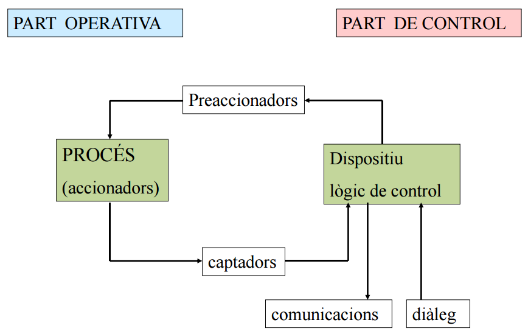
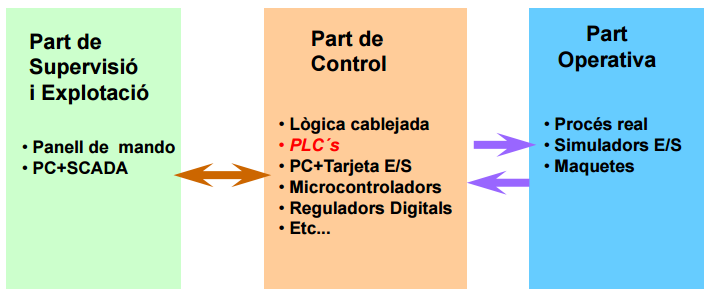
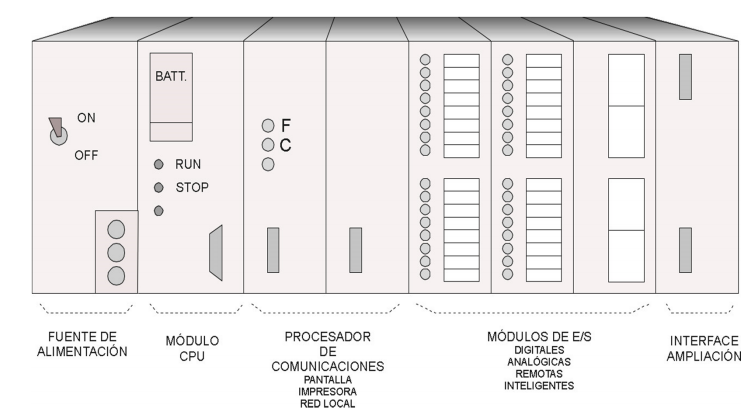
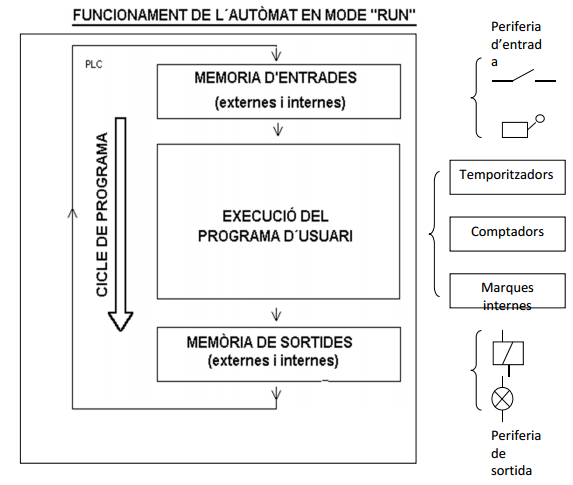
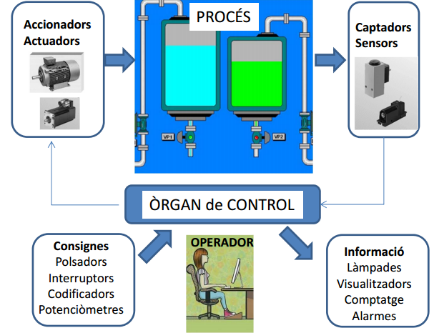
**Autòmats Programables –Entorn i aplicacions-.**

1. Sistema d’automatització components habituals



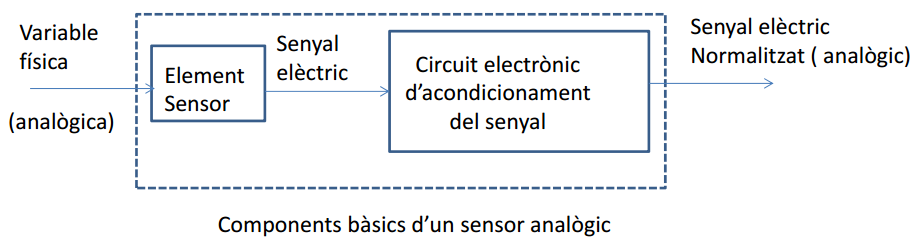
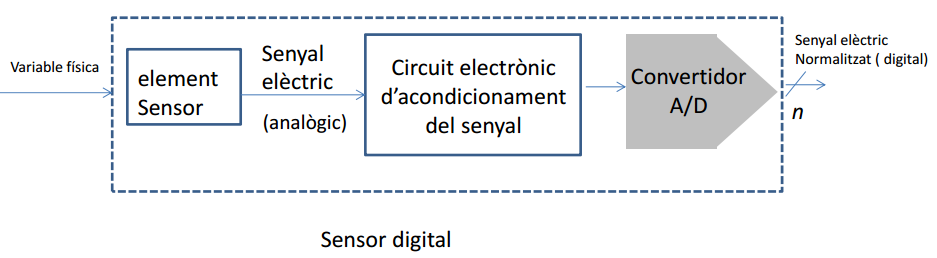
1. Arquitectura d’un autòmat Com treballa un autòmat?
2. Sortides i entrades analògiques

En un procés d’automatització es pretén un pas d’informació en 2 sentits:

* Transferir a l’autòmat la informació del procés
* Aplicar les variables de sortida de l’autòmat cap al procés a través del corresponents actuadors.

Tant els senyals elèctrics d’entrada com els de sortida poden ser: Analògiques o On-Off.

* 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.

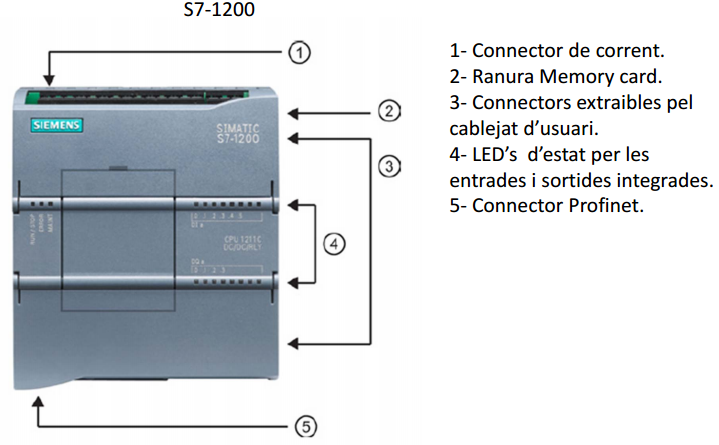


**Autòmat Siemens**

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 programa per 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:**



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

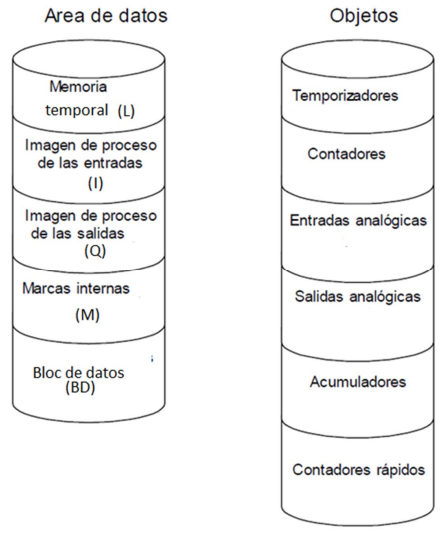
AC/DC/Rly: alimentació amb alterna/sortida contínua DC/les sortides tenen connexió de relé o contacte lliure de potencial RLY.

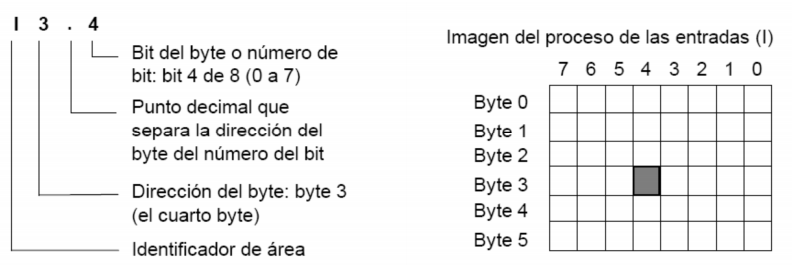
LED’s d’estat: Donen informació de l’estat de funcionament de la CPU, de l’estat de les entrades i sortides i de les fallades del sistema.

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.

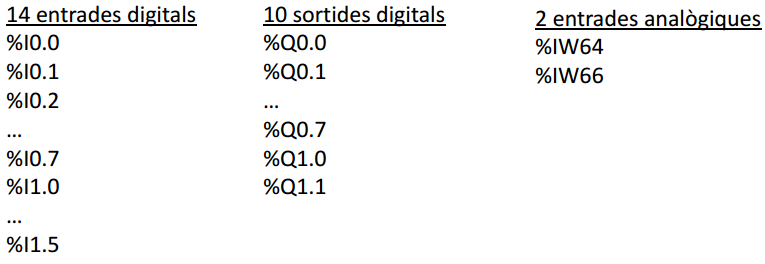
- Tots els LED’s fan pampallugues si es detecta defeca en el fireware.

Memòria de dades:



Distribució de la memòria

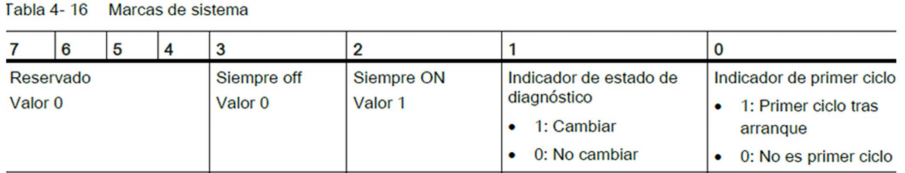
* Memòria de programa: Conté les operacions d’esquema de contactes (KOP), de la llista d’assignació (AWL) o altres, que executa l’autòmat programable per a l’aplicació desitjada.
* Memòria de paràmetres: Permet emmagatzemar determinats paràmetres configurables, tal com contrasenyes, direccions d’estacions 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:

Marques de sistema:

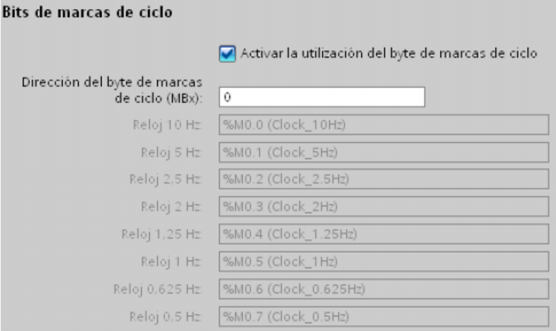
Una marca de sistema és una marca amb valors definits. En la parametrització de la marxa de sistema es determinada el byte de marques de la CPU que es convertirà en el byte de marques de sistema. Aquestes marques s’habiliten en les propietats de la CPU.





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

Marques de cicle:

* Tot bit del Byte de marques de cicle genera un impuls d’ona quadrada
* El byte de marques de cicle 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 cicle és asíncrona respecte al cicle de la CPU, l’estat de la marca del cicle pot canviar varies vegades durant el cicle llarg.

Adreçament de les entrades analògiques:

* Els valors de les entrades analògiques es transformen en paraules (2bytes), és a dir, és un conversor AD 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 aposició que ocupen respecte l0anterior 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 d0entrades 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.

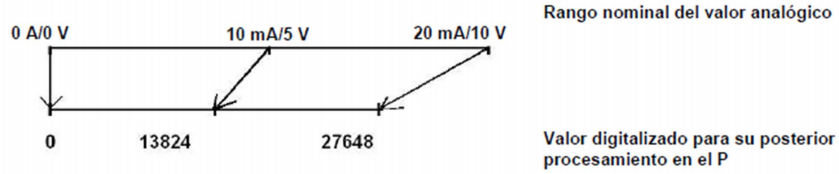
1. Sortides digitals:

* 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.

1. 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 projecta 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.

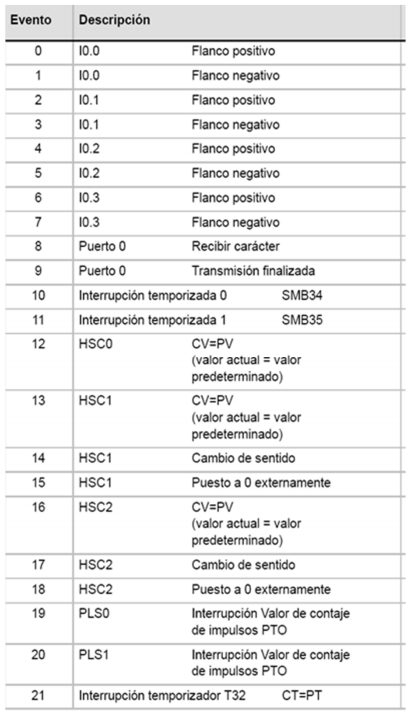
**Rutines d’interrupció:**

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

L’operació Associar interrupció serveix per a assignar l’event d’interrupció a una part del programa.

També és possible associar varis events d’interrupció a una única rutina d’interrupció. Pel contrari, no es pot associar un sol event a vàries rutines.

Quan s’associa un event a una rutina d’interrupció, s’habilita automàticament l’event. Si s’inhibeixen tots els events d’interrupció, 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’interrupció.

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

Regles per a la utilització correcte 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 temp. 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 s0arranca la rutina (és un múltiple sencer de cicle de base d’1ms).
* Desfase: temps de desfase del temps d’arranc respecte al cicle base.

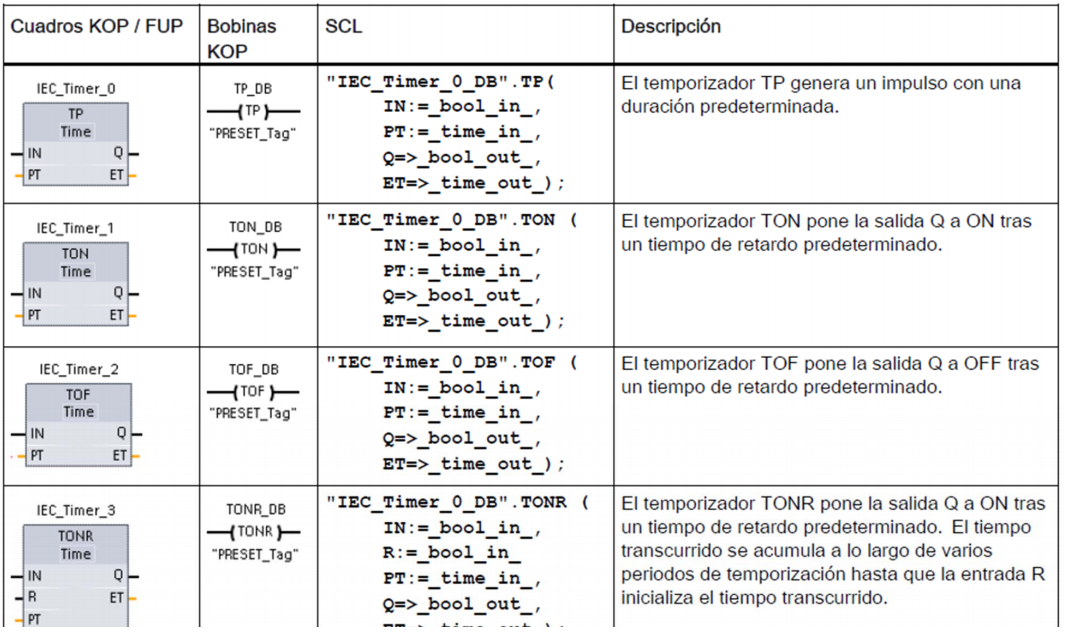
Temps de cicle ajustat ha d’incloure:

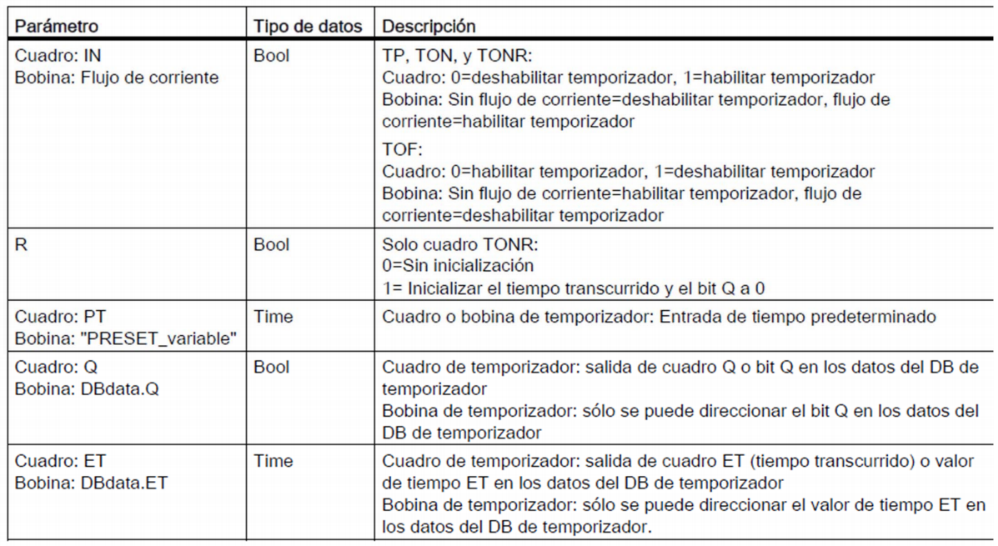
* Temps de processament del OB1 i actualització de les imatges.
* Temps de processament d’interrupcions d’alta prioritat.
* Processos de comunicació del sistema operatiu.
* Accés a la CPU des d’un dispositiu de programació.

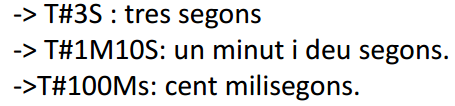
**Bloc PID:**

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.

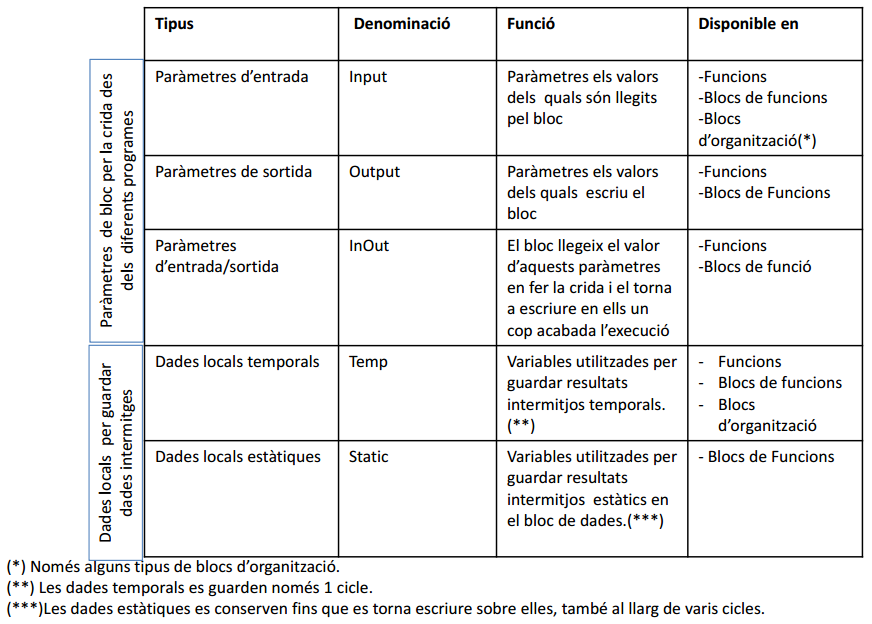


Els valors PT (temps predeterminat) i ET (temps transcorregut) s’emmagatzemen en les dades de DB IEC\_TIMER com sencers dobles amb signe.

**TIA 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.



**Vijeo-CITECT 7.20**

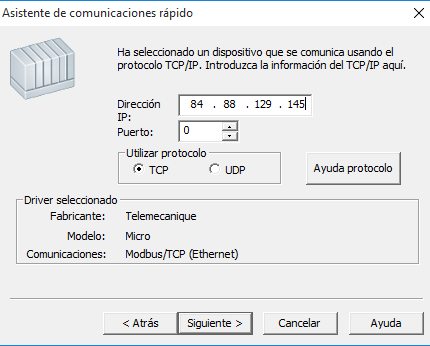
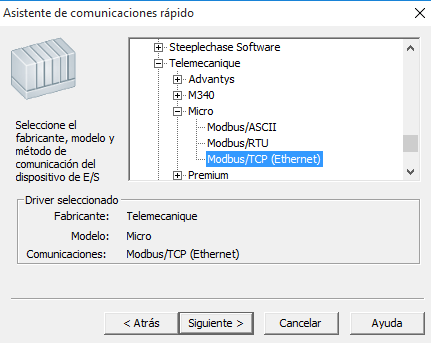
Errades típiques:

- No posar contrasenya.

- No definir usuari.

- Crear mes d’un servidor.

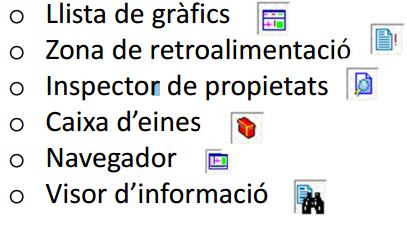
Configuració rapida d’E/S amb un PLC: Crear un nuevo servidor🡪 crear un nuevo dispositiva de E/S (nom PLC) 🡪 seleccionar el tipus de dispositiu de E/S (si es vol amb PLC és externo) 🡪 introduir el TCP/IP🡪 següent, següent i finalitza.

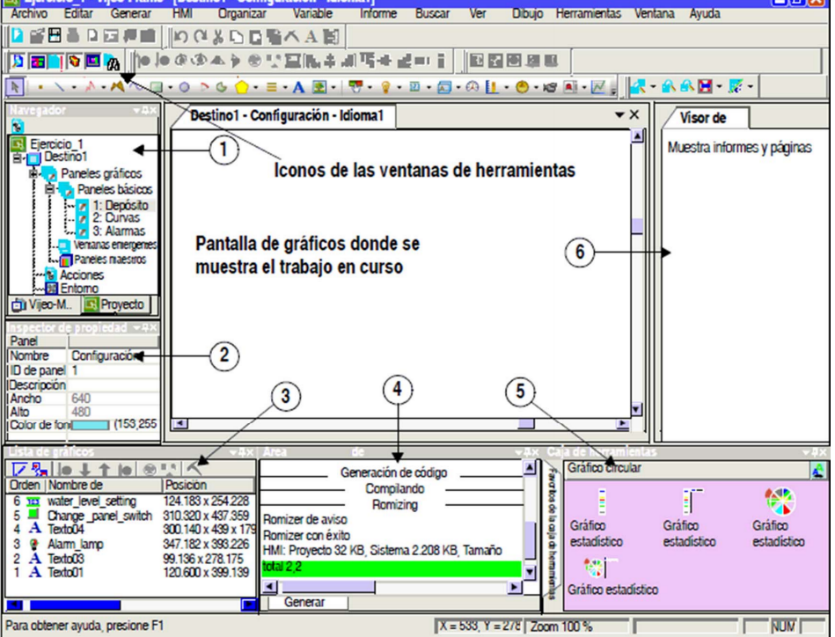


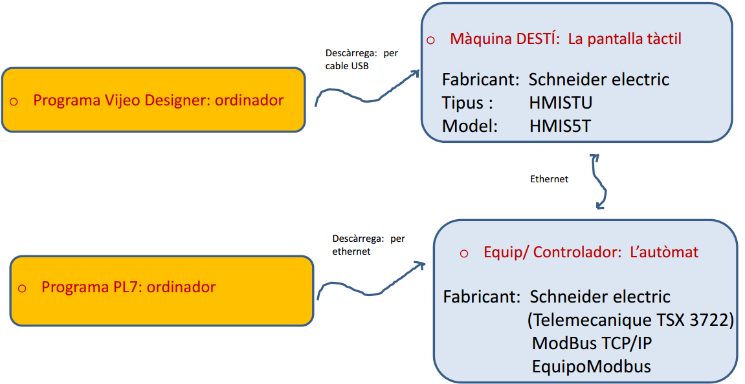
**PANTALLES TÀCTILS VIJEO DESIGNER**

* Permeten la creació de panells d’operador.
* Permeten la configuració dels paràmetres operatius per dispositius d’Interfície Home – Màquina.
* Interessa treballar amb una plicació 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’usurari 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:



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:**

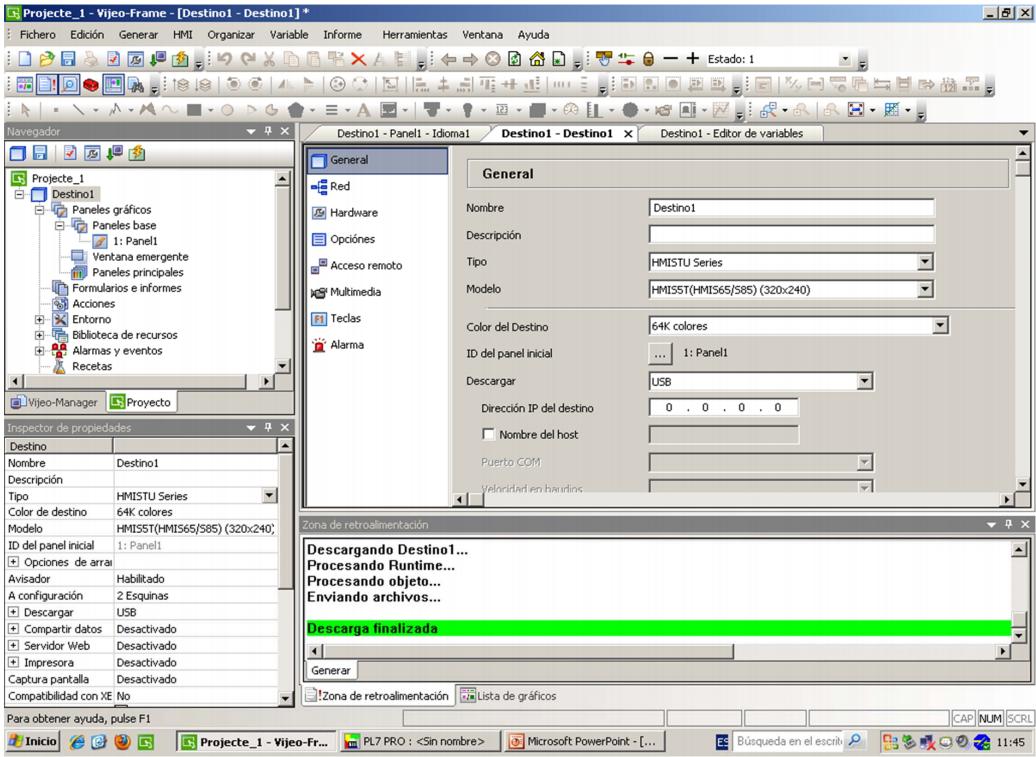
Com construir un projecte amb Vijeo-Designer:

Etapa inicial 1: sense connexió amb l’autòmat.

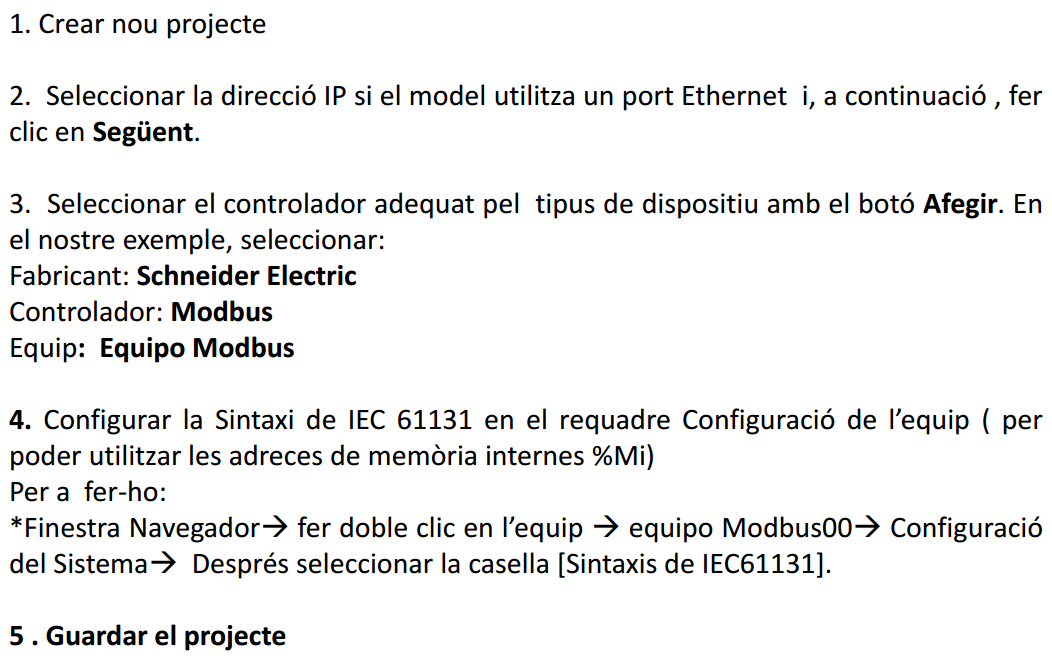
Etapa inicial 2: amb connexió amb l’autòmat.

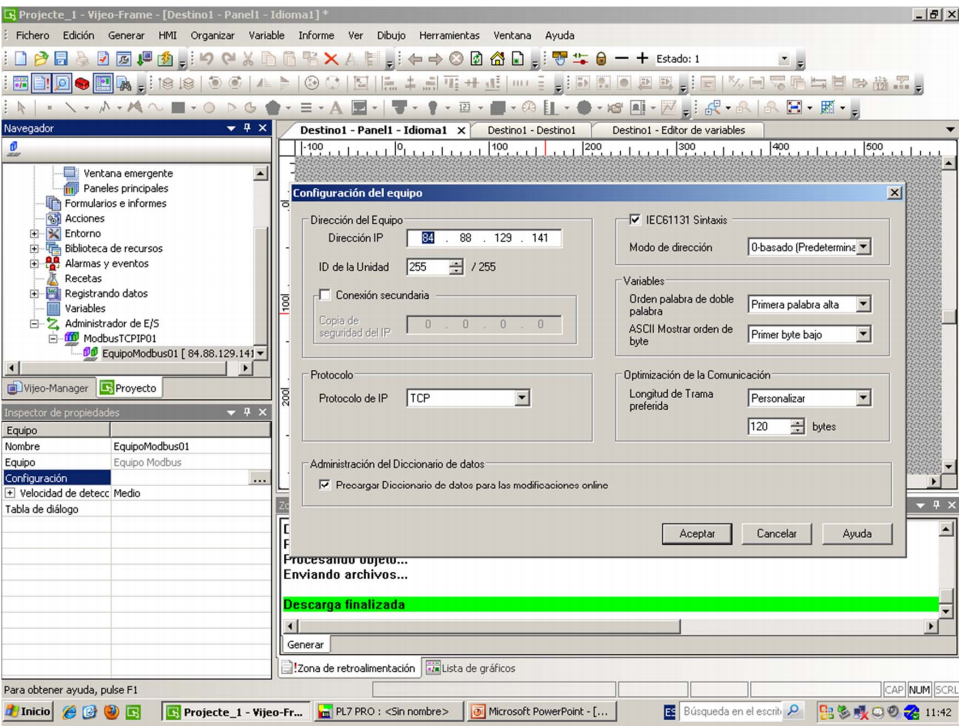
Etapa de desenvolupament de l’aplicació : creació del projecte.

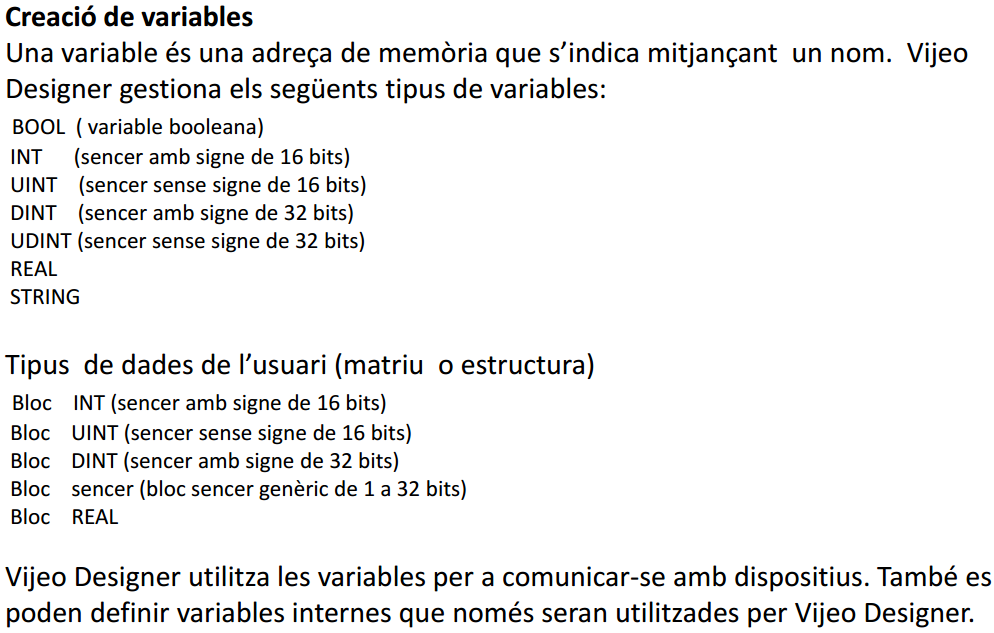
1. Configuració de la pantalla tàctil.



1. Configuració de l’autòmat – controlador-





1. Creació de variables.

**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 “Nevagador”, 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. Antecedents
  2. Objecte
  3. Especificacions i abast

1. 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

1. Plec de condicions
2. Estat d’amidaments
3. Pressupost