





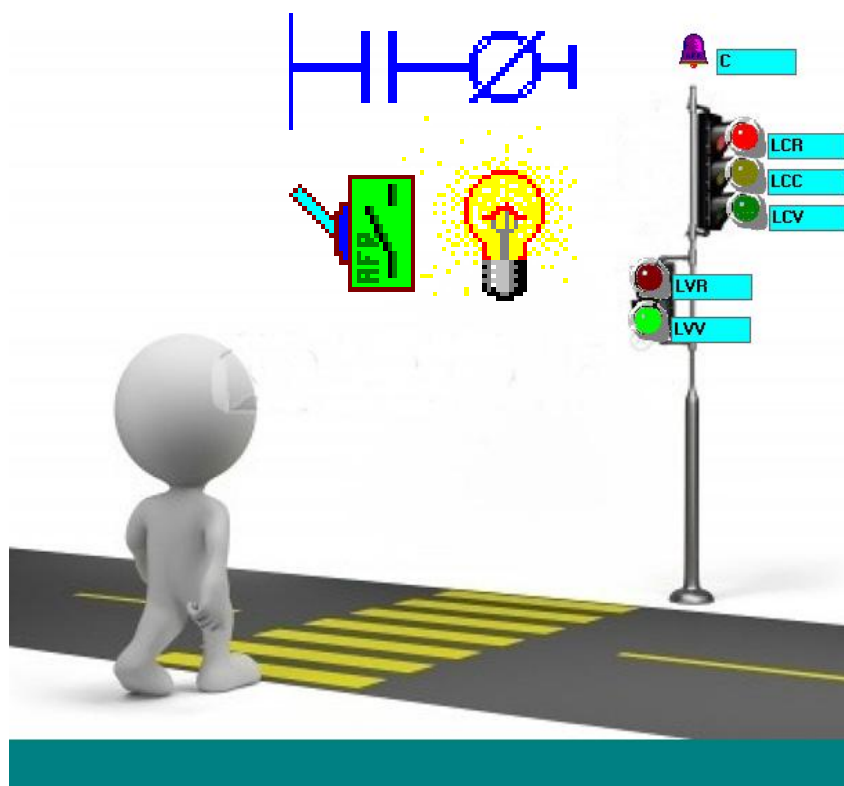


		Traductor
 Euskera	Gabe. Egin nahi duzu?	aferrer8 @xtec.cat
 Français	Manque. Volez vous le faire?	aferrer8 @xtec.cat
 Mexicano	<u>Manual de VirPLC</u>	Emiliano Zapata
 English	Pending. Do you want do it?	aferrer8 @xtec.cat
 Italiano	Manca ancora. Vuoi farlo tu?	aferrer8 @xtec.cat
 Català	<u>Manual de VirPLC</u>	Toni Ferrer



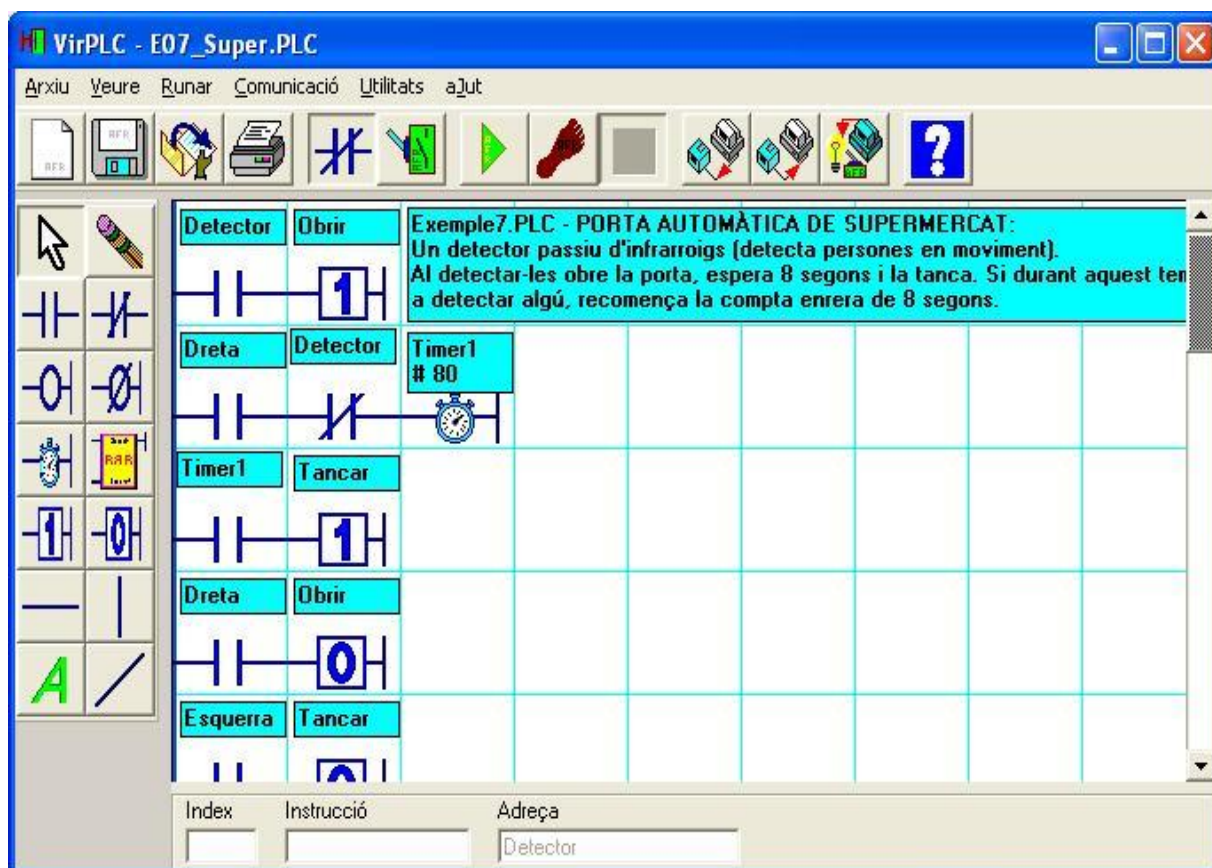
VirPLC és un simulador d'Autòmats Programables o PLC's, creat exclusivament amb finalitats didàctiques, per tal de permetre aprendre a programar PLC mitjançant diagrames de contactes (ladder), sense necessitat d'adquirir el PLC real.

Un PLC és un equip dotat d'un microcontrolador; una sèrie d'entrades i sortides; i una memòria on es carrega un programa que defineix relació entre el comportament del sistema i l'estat dels elements d'entrada i de sortida. Un PLC fa funcions de control de tipus lògic i seqüencial.

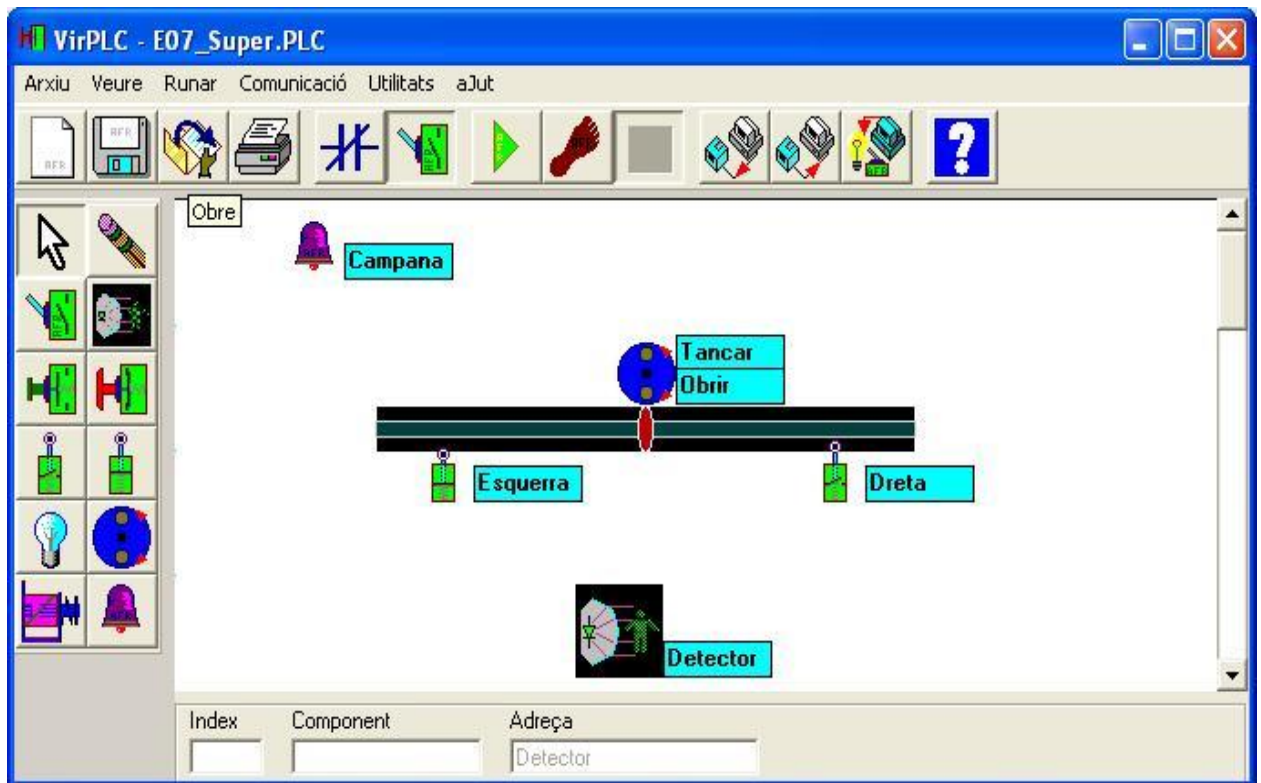
Avui dia, els autòmats programables han desplaçat completament aquells enormes quadres elèctrics de control industrial formats per relés, temporitzadors i un munt de cablatge. Els seus avantatges (**fiabilitat**, **flexibilitat** d'operació, dimensions i preu) han suposat una revolució en el camp del control de màquines i sistemes industrials i en algunes aplicacions d'àmbit domèstic: ascensors, portes automàtiques de garatges, ...

**VirPLC** és virtual perquè té dos entorns de treball:




- Veure - Soft: Un entorn de Software format per una pantalla on s'hi poden col·locar una sèrie d'instruccions de tipus gràfic a mode de diagrames de contactes.



- b) Veure - Hard: Un entorn de Hardware on es poden disposar dels elements d'entrada i de sortida més comuns: interruptors, finals de cursa, sensors, bombetes, electroimants, motors, sirenes.... Tots aquests elements són dinàmics i molt entenedors en temps d'execució del programa. A més: mai fallen.



Aquests 2 entorns de treball (soft i hard), es poden combinar amb 3 modalitats d'execució:

 Runar Stop	En temps de disseny del programa i del hardware
 Runar Pas a Pas	En temps de depuració (1 Intro = 1 pas)
 Runar Go	En temps d'execució i test. A velocitat ajustable. Si executem runa amb el programa maximitzat, i la resolució de pantalla ho permet, a partir de la versió 3.0 podem veure soft i hard en execució simultània.

El programa, a més, s'acompanya de molts exemples expressament no depurats, per poder entendre millor el seu funcionament. Les possibilitats de crear-ne d'altres depèn només del límit de la imaginació de cadascú. Aquests exemples serveixen també per aclarir estratègies com ara la col·locació correcta dels finals de cursa respecta el conjunt motor/corredissa.


L'entorn de treball del software és molt més flexible i dinàmic que el subministrat pel desenvolupament de programes en altres autòmats. Per exemple, el programa no està condicionat a la partició en blocs típica dels autòmats comercials, cosa que dóna molta més llibertat i possibilitats a l'usuari programador del PLC.

## Funcionament i programació dels PLC's

El microprocessador del PLC, en temps d'execució del programa fa els següent:








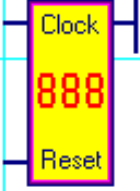
- 1.- Llegeix l'estat de les **entrades**, de les **sortides**, dels **timers** i dels **flags** (bits de memòria)
- 2.- Interpreta i executa totes les instruccions de dalt a baix
- 3.- Escriu totes les sortides i flags.
- 4.- Torna a anar a 1

Aquest cicle continu és anomenat Scan, i la seva durada: temps d'scan. Aquest temps es pretén que sigui el més petit possible, de manera que respongui el més immediatament millor a qualsevol eventualitat. No obstant aquest temps és directament proporcional a la llargada del programa i inversament a la potència del PLC. VirPLC permet augmentar el temps d'scan a efectes de prova i depuració del programa.

 **Important:** Per escriure o llegir un bit de memòria interna, cal associar-li un flag a la pantalla de hard, amb els seu nom corresponent.

## FUNCIONS – INTRUCCIONS amb VirPLC

VirPLC només té 8 instruccions. Totes actuen sobre la seva adreça de la següent forma:

<b>P1</b>		<b>LOAD (LLEGEIX)</b> Llegeix l'estat d'una entrada, d'una sortida, d'un bit intern, d'un timer o d'un comptador. Torna 1 si l'entrada està tancada, la sortida està activa, el timer ha acabat de descomptar, o el comptador està a zero. Torna 0 en cas contrari.
<b>P3</b>		<b>LOAD NOT</b> Llegeix i, després, inverteix el resultat. (respon a la pregunta: el interruptor està obert?, la làmpada està apagada?... 1 => Si)
<b>L1</b>		<b>OUT (ESCRIU)</b> Escriu el resultat a una sortida o a un bit intern. (Lògicament, no es poden escriure entrades)
<b>L3</b>		<b>OUT NOT</b> Inverteix el que té i després ho escriu.
<b>B6</b>		<b>SET (POSA A 1)</b> Posa a 1 una sortida o un bit intern si la seva "Condicció d'execució" (el que té) és 1.
<b>B4</b>		<b>RESET, RSET o CLR</b> Posa a 0 una sortida o un bit intern si la seva "Condicció d'execució" és 1.
<b>Timer1 #40</b>		<b>TIMER o TIM (TEMPORITZADOR)</b> Si la seva "condició d'execució" és 1, descompte el valor de temps programat en dècimes de segon. Si la seva condició d'execució és zero, Restableix el temps al valor original. Recordem que es pot llegir, i torna 1 si el timer s'ha descomtat.
<b>Compta1 #8</b>		<b>CMPT (DESCOMPTADOR)</b> Descompte el valor original amb els impulsos que li entren. La seva entrada reset restableix aquest valor.

*S'entén per "Condicció d'execució" la sèrie de LOADs i LOAD's\_NOT disposats, a l'esquerra de la sortida, en forma d'operacions lògiques més o menys complexes en disposició AND (sèrie) o OR (paral·lel).*

## CURS DE PLC: Proposta d'exercicis amb VirPLC

Molts d'aquest exemples es poden descarregar des de la pàgina WEB de VirPLC. Cal tenir en compte que, a efectes de pràctiques, alguns estan expressament incomplets o erronis

### PRÀCTIQUES LD, LDNOT, OUT, OUTNOT...

- 1) P1 Interruptor tancat => Bombeta encesa i brunzidor apagat. Interruptor obert => Al revés
- 2) P2-ALARMA: 1 interruptor-clau; 2 sensors passius infraroigs NT (P0, P1), 1 polsador sota el taulell
  - a) R: alarma sona si Int-Clau = OFF i detecta algú o si premem Puls NT sota el taulell
- 3) P3-CAMIONER:
  - a) R: si una de les 4 rodes rebentades ^ led indicador;
  - b) OK: si les 2 rodes de l'esquerra o les 2 de la dreta rebentades sona sirena.
- 4) P4-COMBINACIONS
  - a) R: 1 de 2 int tancats encendre una bombeta
  - b) OK: 1 de 3 int tancats encendre una bombeta
  - c) OK+: 1 de 4 int oberts apagar una bombeta
  - d) OK++ 2 de 4 int tancats encendre una bombeta
- 5) P5-FUNCIÓ: Representació d'una funció lògica complexa  $F = [ [ ( a b + (a+b)e ) \cdot c ] + f ] \cdot d$
- 6) P6-GRUA SEGURA: mitjançant 2 comandaments de 2 int x pujar i 2 x Baixar
  - a) R: P1, P2 = PUJA i B1, B2 = BAIXA
  - b) OK: cada comandament incorpora un Int. Clau (si no està tancat no va)
  - c) OK+: La grua te Finals de cursa NT
  - d) OK++: si un dels dos comandaments està tancat i l'altre no, funcionen individualment

### PRACTIQUES AMB .... + SET, RSET

- 7) P7-HOSPITAL: 4 habitacions amb 4 polsador i 4 bombetes d'avís al passadís. A infermeria 1 bombeta, 1 brunzidor, 1 Puls i 1 Int obert=> dia; tancat => nit
  - a) R: En prémer Puls de pacient s'encén la seva bombeta i la de consergeria. En prémer a consergeria s'apaguen totes les bombetes.
  - b) OK: a més s'encén el brunzidor si és de dia.
  - c) OK+ els pacients tenen un altre polsador "ME'N ARREPENTEIXO" només apaga la seva bombeta i si és la única, també apaga els avisos d'infermeria.
- 8) P8-CONCURS TV3: 3 concursants amb 3 polsadors.
  - a) R: S'encén la bombeta de qui prem primer. El presentador les ha d'apagar totes amb PulsOFF abans de la propera pregunta
  - b) OK: El presentador te una bombeta verda quan els concursants poden contestar, Per encendre-la prem PulsON i només s'encén si totes les bombetes dels concursants estan apagades.
- 9) P9-MOVIMENT CONTINUU
  - a) R: Porta tancant i obrint continuament amb 2 finals de cursa NO i amb un polsador de marxa NO i un d'aturada NT
  - b) OK:... Idem però que si quan el parem, girava a la dreta, que en tornar a engegar continuï girant a la dreta (utilitzant un FLAGS)
  - c) OK+: ... Idem amb un sol polsador que engega i para (utilitzant un altreFLAG)
- 10) P10-CAIXA FORTA amb 10 polsadors i motor amb finals de cursa NT
  - a) R: Codi clau de 4 xifres per obrir una caixa forta (3 FLAGS)

- b) OK Microrruptor porta tancada: motor obre pestells si codi=ok + microrruptor off. I tanca els pestells si es tanca la porta.

### **PRÀCTIQUES AMB ..... + TIM**

- 11) P11-SEMÀFOR: (INCLOURE IMATGE)
- a) R: Semàfor Kamikage: Control semàfor de vianants amb so. Sol·licitud per polsador
  - b) OK: Control 1 semàfor de vianants i 1 de cotxes amb so.
  - c) OK+: Control 2 semàfor de vianants i 2 de cotxes amb so.(USANT OUT)
- 12) P12-INTERMITENTS:
- a) R: control dels 2 intermitents del cotxe amb 2 polsadors
  - b) Ok: Incloure les altres 2 bombetes i un 3r Int de Perill
- 13) P13-RULETA GIRATORIA:
- a) R: Roda a temps mínim fins que es deixa anar un polsador (queda amb la nova posició)
  - b) OK: es para a cap d'un temps de deixar anar el polsador
  - c) OK+: es para desaccelerat
- 14) P14-PORTA DE SUPERMERCAT – preguntar que ha de fer i què fa falta de hard???
- a) R: amb el sensor obre, es para 4 segons i tanca quan és de dies
  - b) OK: idem però que si detecta algú mentre la porta està oberta, reposi el temps de descompte i que torni a obrir.
  - c) OK+: Quan l'encarregat amb la cau posa Int a nits, es tanca si està obert
  - d) OK++ Les 3 caixeres tenen un polsador SOS sota cada caixa. Es tanca la porta; sona una alarma; i fins que l'encarregat no obra i tanca la clau de l'Int de dies, no torna a obrir-se la porta
- 15) P15-ASCENSOR – preguntar que ha de fer i què fa falta de hard???
- a) R: Control d'un ascensor de tres plantes amb 3 Puls, 3 LEDs; 3 FC\_NT;
  - b) OK: amb polsadors de cabina: P0, P1, P2 i STOP
  - c) OK+: amb microrruptor de porta de la cabina tancada i si s'ha premut stop, només van els de la cabina.
  - d) OK++: amb memòria
- 16) P16-GARATGE – preguntar que ha de fer i què fa falta de hard??? ( 2 finals de cursa, 1 detector reflectiu de objectes sota la porta i un polsador P1 accionable interiorment, exteriorment amb clau, o de dins del cotxe amb un comandament.
- a) R: Sistema de funcionament particular. En prémer un únic polsador "P1" fa el següent: Si la porta està tancada, obre; si la porta està oberta, la tanca; si està obrint, l'atura; si està tancat l'atura i obra. Si està aturada, la tanca.
  - b) OK: funcionament comunitari: 3 polsadors P1, Pstop; P2. Si la porta esta tancada o esta parada o esta tancant, en prémer P1 s'obre. Quan la porta esta oberta, passats 3min o prement P2 es tanca. En prémer Pstop amb la porta oberta, la porta obrint o tancat, para i deixa de comptar els 3 min per tancar automàticament (només tancarà prement P2)
  - c) OK++: els 2 sistemes funcionen a la vegada i depenen d'un selector lsel = ON => Particular
- 17) P17: BARRERA DE PÀRQUING
- 18) 3 EN LINIA: 9 PULSADORS, 9 LEDS VERMELLS NOU LEDS VERDS; 2 LEDS INDICANT EN TORN; SI GUANYA EL VERD, TOTS ELS LEDS VERD PARPADEIXEN 5 SEGONS I TORNA A COMENÇAR

### **PRÀCTIQUES AMB ..... + CMPT**

- 19) P18:COMPTA
- a) R: Al prémer 8 cops el polsador CK s'activa la bombeta. Al polsar Reset el comptador s'inicialitza altre cop a 8 i s'apaga la bombeta.
  - b) OK: En prémer P es realitza un moviment esquerra-dreta-esquerra-dreta 8 cicles i es para.

## PROPOSTA DE PROJECTES

Objectiu: es pretén el disseny i simulació d'un sistema de control habitual amb el que sovint ens hem trobat davant dels nassos i segurament no li hem donat cap importància.

Es proposen els següents 4 exemples però es pot optar per un altre de diferent...

<b>1.- PORTA DE GARATGE:</b>		<b>2.- ASCENSOR 2 PLANTES</b>	
<b>Sortides a controlar</b>	<b>Codi</b>	<b>Sortides a controlar</b>	<b>Codi</b>
Motor gira tancant	MT	Motor gira baixant	MB
Motor gira obrint	MO	Motor gira pujant	MP
<b>Entrades a controlar</b>	<b>Codi</b>	<b>Entrades a controlar</b>	<b>Codi</b>
Final de carrera superior	FCS	Final de carrera Pl. 1a.	FCP1
Final de carrera inferior	FCI	Final de carrera Pl. Baixa	FCP0
Polsador de pujar (obrir)	PO	Polsador Pl. 1a (cridar-lo)	P1
Polsador d'aturada	PA	Polsador Pl. Baixa (cridar-lo)	P0
Polsador de baixar (tancar)	PT		

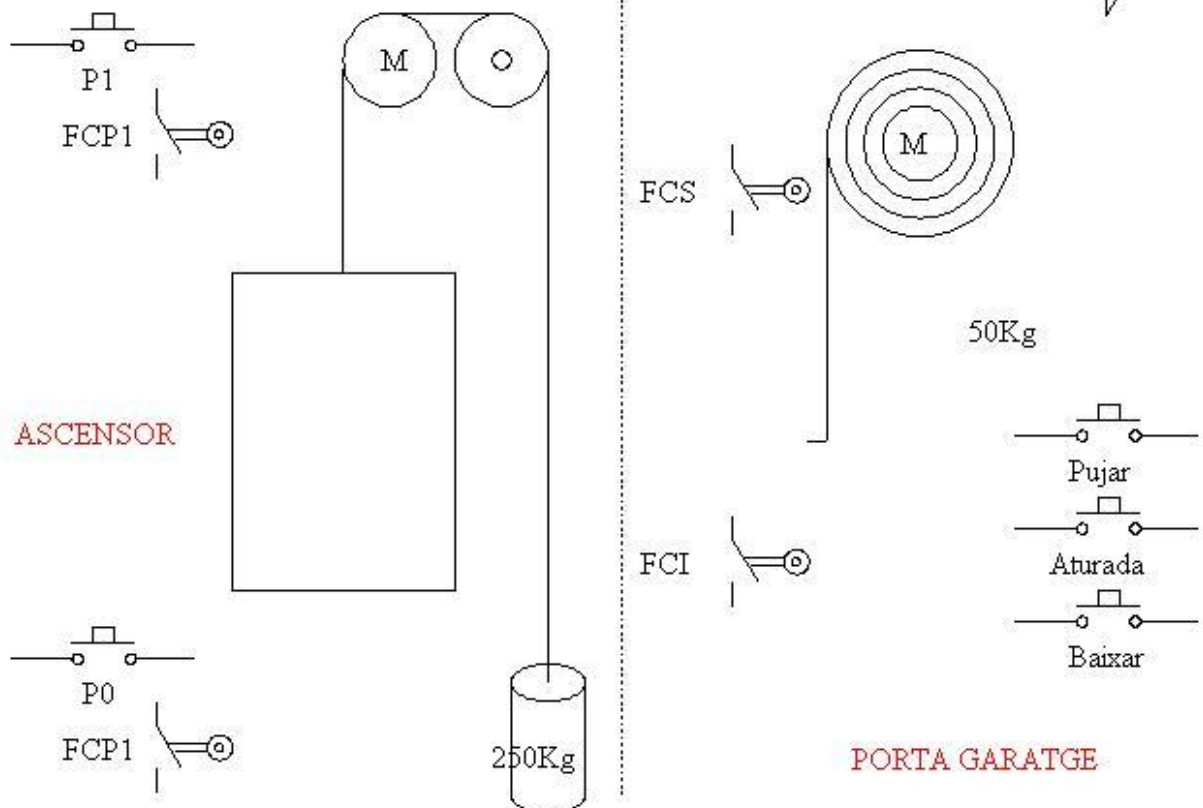
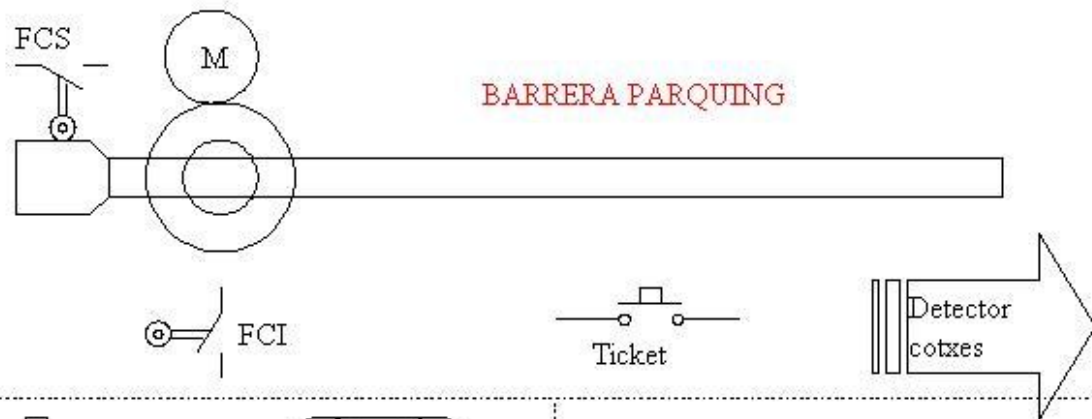
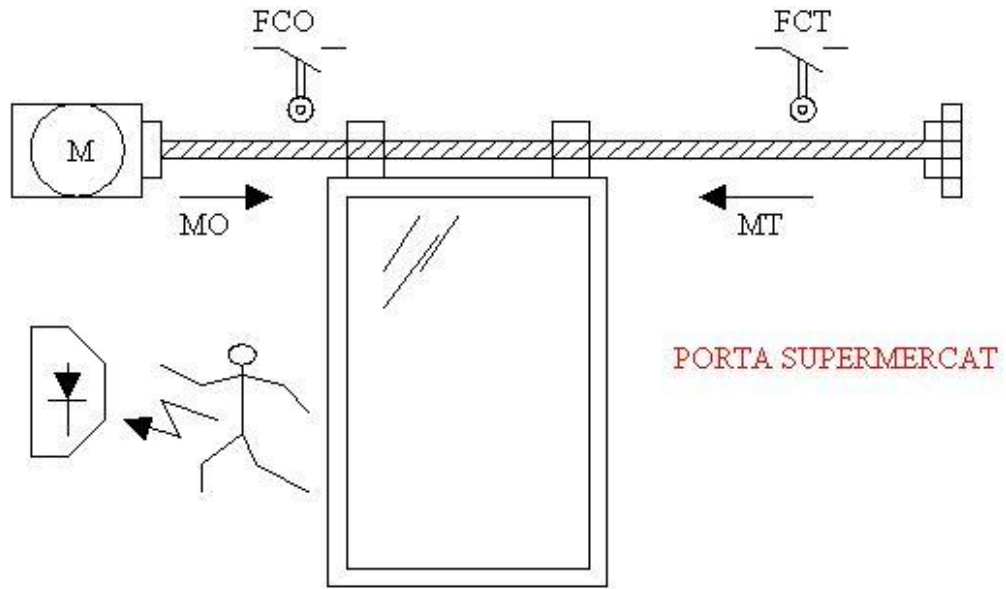
<b>3.- BARRERA DE PARQUING</b>		<b>4.- PORTA SUPERMERCAT</b>	
<b>Sortides a controlar</b>	<b>Codi</b>	<b>Sortides a controlar</b>	<b>Codi</b>
Motor gira tancant	MT	Motor gira tancant	MT
Motor gira obrint	MO	Motor gira obrint	MO
<b>Entrades a controlar</b>	<b>Codi</b>	<b>Entrades a controlar</b>	<b>Codi</b>
Final de carrera superior	FCS	Final de carrera porta oberta	FCO
Final de carrera inferior	FCI	Final de carrera porta tancada	FCT
Polsador sol·licitud ticket	PT	Detector de persones	DPI
Detecció de cotxe sota barrera	DC	Detector d'encallades tancant	DET

### Ampliació porta de garatge:

La porta de garatge pot tenir dos sistemes de funcionament commutables per un Switch:

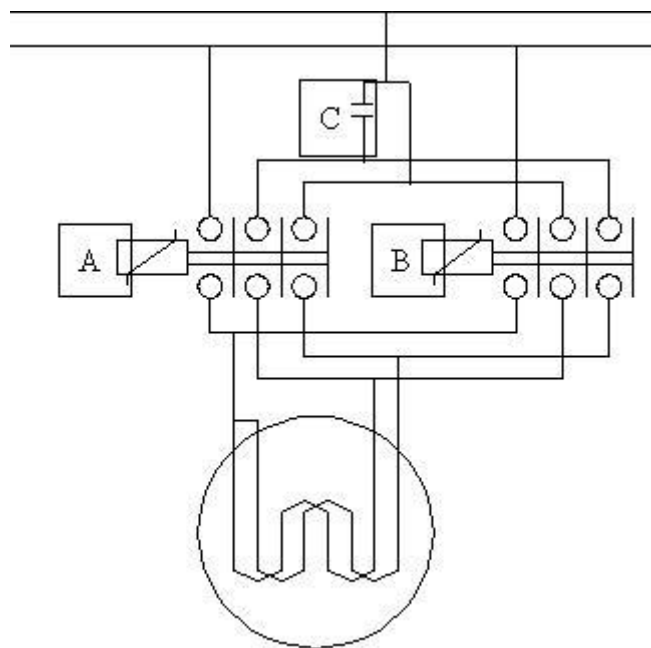
- Porta baixa per time out i té 3 polsadors interns (Pujar, Stop, Baixar)
- La porta només té un polsador intern i un extern. No baixa fins que es prem un d'ells.





AMPLIACIÓ: CANVI DE DIRECCIÓ EN MOTORS ASINCRONS MONOFÀSICS DE CONDENSADOR

A	B	Funcionament
0	0	Motor aturat
0	1	Motor activat girant en un sentit
1	0	Motor activat girant en l'altre sentit
1	1	Curtcircuit



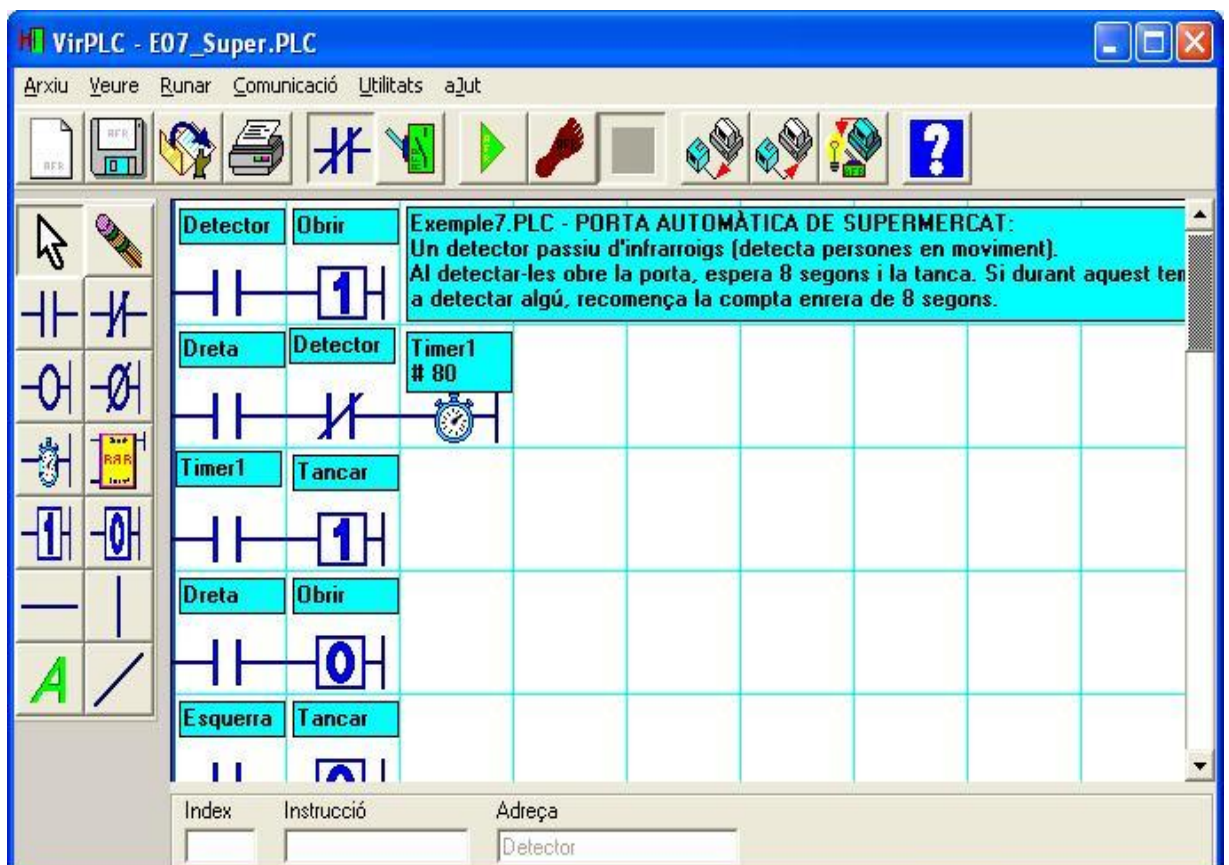
**VirPLC** es un simulador de Autómatas Programables o PLC, creado exclusivamente con fines didácticos, para permitir aprender a programar PLC mediante diagramas de contactos (ladder), sin necesidad de adquirir el PLC real.

Un PLC es un equipo dotado de un microcontrolador; una serie de entradas y salidas; y una memoria donde se carga un programa que define relación entre el comportamiento del sistema y el estado de los elementos de entrada y de salida. Un PLC hace funciones de control de tipo lógico y secuencial.

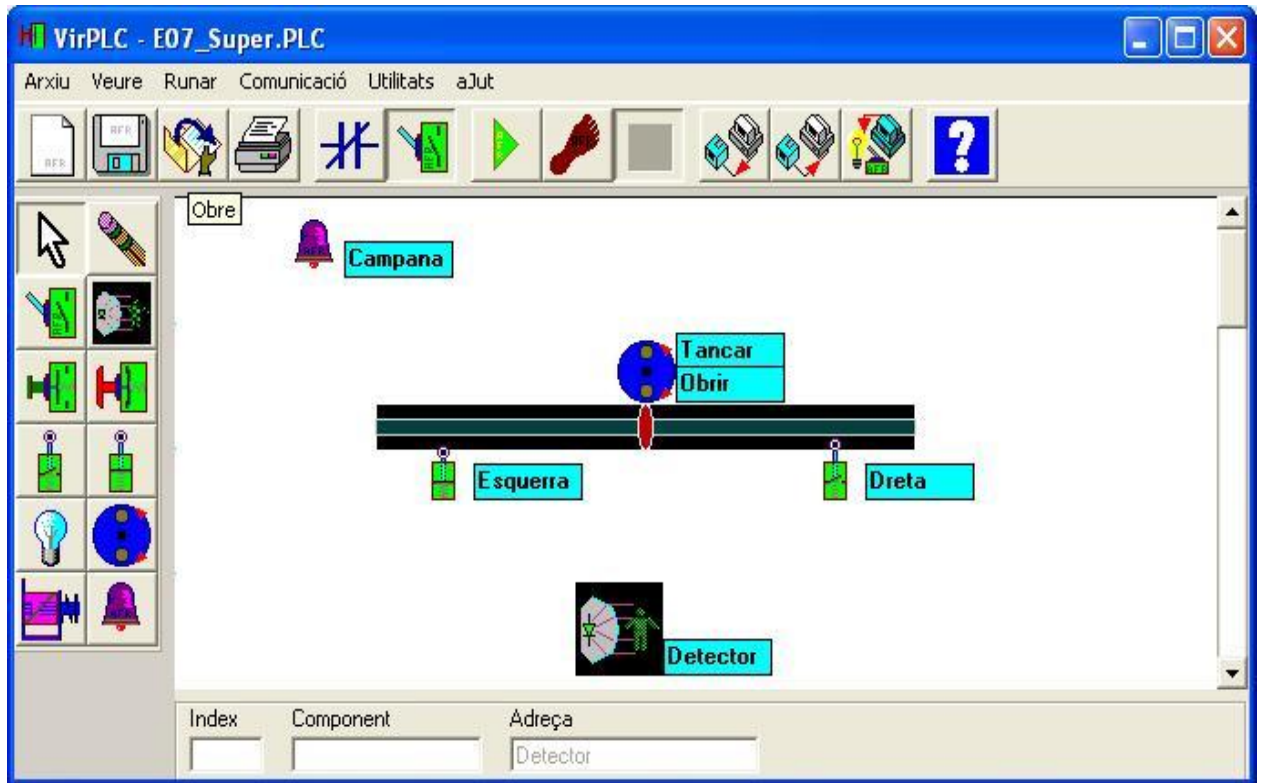
Hoy día, los autómatas programables han desplazado completamente aquellos enormes cuadros eléctricos de control industrial formados por relés, temporizadores y un montón de cableado. Sus ventajas (fiabilidad, flexibilidad de operación, dimensiones y precio) han supuesto una revolución en el campo del control de máquinas y sistemas industriales y en algunas aplicaciones de ámbito doméstico: ascensores, puertas automáticas de garajes,...

**VirPLC** es virtual porque tiene dos entornos de trabajo:




a) **Ver - Soft**: Un entorno de Software formado por una pantalla donde se pueden poner una serie de instrucciones de tipo gráfico a modo de diagramas de contactos.



- c) **Ver - Hard:** Un entorno de Hardware donde se pueden disponer de los elementos de entrada y de salida más comunes: interruptores, finales de carrera, sensores, bombillas, electroimanes, motores, sirenas .... Todos estos elementos son dinámicos y muy comprensibles en tiempo de ejecución del programa. Además: nunca fallan.



Estos 2 entornos de trabajo (soft y hard), se pueden combinar con 3 modalidades de ejecución:

 Runar Stop	En tiempo de diseño del programa y del hardware.
 Runar Pas a Pas	En tiempo de depuración (1 Intro = 1 paso)
 Runar Go	En tiempo de ejecución y test. A velocidad ajustable. Si ejecutamos "Runar" con el programa maximizado, y la resolución de pantalla lo permite, a partir de la versión 3.0 podemos ver soft y hard en ejecución simultánea.

El programa, además, se acompaña de muchos ejemplos expresamente no depurados, para poder entender mejor su funcionamiento. Las posibilidades de crear otras, depende sólo del límite de la imaginación de cada uno. Estos ejemplos sirven también para aclarar estrategias tales como la colocación correcta de los finales de carrera respecto al conjunto motor / corredera.


El entorno de trabajo del software es mucho más flexible y dinámico que el suministrado por el desarrollo de programas en otros autómatas. Por ejemplo, el programa no está condicionado a la partición en bloques típica de los autómatas comerciales, lo que da mucha más libertad y posibilidades al usuario programador del PLC.

## Funcionamiento y programación de PLC's

El microprocesador del PLC, en tiempo de ejecución del programa hace lo siguiente:








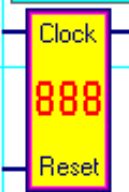
- 1.- Lee el estado de las entradas, de las salidas, los timers y los flags (bits de memoria)
- 2.- Interpreta y ejecuta todas las instrucciones de arriba abajo
- 3.- Escribe todas las salidas y flags.
- 4.- Vuelve a ir a 1

Este ciclo continuo es llamado Scan, y su duración: tiempo de scan. Este tiempo se pretende que sea lo más pequeño posible, de forma que responda lo más inmediatamente posible a cualquier eventualidad. Sin este tiempo es directamente proporcional a la longitud del programa e inversamente a la potencia del PLC. VirPLC permite aumentar el tiempo de scan a efectos de prueba y depuración del programa.

 **Importante:** Para escribir o leer un bit de memoria interna, hay que asociarle un flag en la pantalla de hard, con su nombre correspondiente.

## FUNCIONES – INSTRUCCIONES con VirPLC

VirPLC sólo tiene 8 instrucciones. Todas actúan sobre su dirección de la siguiente forma:

<b>P1</b>	<b>LOAD (LEER)</b> Lee el estado de una entrada, de una salida, d'un bit interno, de un timer o de un contador. Retorna 1 si la entrada está cerrada, la salida está activa, el timer ha finalizado el descuento, o el contador está a cero. Devuelve 0 en caso contrario.
	
<b>P3</b>	<b>LOAD NOT</b> Lee y luego invierte el resultado. (responde a la pregunta: el interruptor está abierto? la luz está apagada?... 1 => Si)
	
<b>L1</b>	<b>OUT (ESCRIBE)</b> Escribe el resultado a una salida o a un bit interno. (Lógicamente, no es posible escribir en entradas)
	
<b>L3</b>	<b>OUT NOT</b> Invierte y luego escribe.
	
<b>B6</b>	<b>SET (PONER A 1)</b> Pone a 1 una salida o un bit interno. Pero solo lo hace si su "Condición de ejecución" es 1.
	
<b>B4</b>	<b>RESET, RSET o CLR</b> Pone a 0 una salida un bit interno si su "Condición de ejecución" es 1.
	
<b>Timer1 #40</b>	<b>TIMER o TIM (TEMPORIZADOR)</b> Si su "condición de ejecución" es 1, descuenta el valor en tiempo real a partir del programado en decimas de segundo. Si su condición de ejecución es 0, Restablece el tiempo al valor original.
	
<b>Compta1 #8</b>	<b>CMPT (DESCONTADOR)</b> Descuenta el valor original con los impulsos que le llegan. Su entrada Reset, restablece este valor.
	

Se entiende por "Condición de ejecución" la serie de LOAD y LOAD's\_NOT dispuestos, a la izquierda de la salida, en forma de operaciones lógicas más o menos complejas en disposición AND (serie) o OR (paralelo).

## CURSO DE PLC: Propuesta d'ejercicios con VirPLC

*Muchos de estos ejemplos pueden descargarse desde la página WEB de VirPLC. Hay que tener en cuenta que, a efectos de prácticas, algunos están expresamente incompletos o erróneos*

### PRÁCTICAS LD, LDNOT, OUT, OUTNOT ...

- 1) P1 Interruptor cerrado => Bombilla encendida y zumbador apagado. Interruptor abierto, al revés
- 2) P2-ALARMA: 1 interruptor-llave; 2 sensores pasivos infrarrojos NT (P0, P1), 1 pulsador de SOS
  - a) R: alarma suena si Int-Clave = OFF y detecta a alguien o si pulsamos Puls NT bajo el mostrador
- 3) P3-camiones:
  - a) R: si una de las 4 ruedas reventadas ^ led indicador;
  - b) OK: si las 2 ruedas de la izquierda o las 2 de la derecha reventones suena sirena.
- 4) P4-COMBINACIONES
  - a) R: 1 de 2 int cerrados encender una bombilla
  - b) OK: 1 de 3 int cerrados encender una bombilla
  - c) OK +: 1 de 4 int abiertos apagar una bombilla
  - d) OK ++ 2 de 4 int cerrados encender una bombilla
- 5) P5-FUNCIÓN: Representación de una función lógica compleja  $F = [[(ab + (a + b) e) \cdot c] + f] \cdot d$
- 6) P6-GRÚA SEGURA: mediante 2 mandos de 2 int x subir y 2 x Bajar
  - a) R: P1, P2 = SUBE y B1, B2 = BAJA
  - b) OK: cada mando incorpora un Int. Clave (si no está cerrado no)
  - c) OK +: La grúa tiene Finales de carrera NT
  - d) OK ++: si uno de los dos mandos está cerrado y el otro no, funcionan individualmente

### PRÁCTICAS con .... + SET, RSET

- 7) P7-HOSPITAL: 4 habitaciones con 4 pulsador y 4 bombillas de aviso al pasillo. A enfermería 1 bombilla, 1 zumbador, 1 Puls y 1 Int abierto => día; cerrado => noche
  - a) R: Al pulsar Puls de paciente enciende su bombilla y la de conserjería. Al pulsar en conserjería apagan todas las bombillas.
  - b) OK: además enciende el zumbador si es de día.
  - c) OK + los pacientes tienen otro pulsador "ME arrepentido" sólo apaga su bombilla y si es la única, también apaga los avisos de enfermería.
- 8) P8-CONCURSO TV3: 3 concursantes con 3 pulsadores.
  - a) R: Se enciende la bombilla de quien pulsa primero. El presentador las tiene que apagar todas con PulsOFF antes de la próxima pregunta
  - b) OK: El presentador tiene una bombilla verde cuando los concursantes pueden contestar, Para encenderla pulsa PulsON y sólo se enciende si todas las bombillas de los concursantes están apagadas.
- 9) P9-MOVIMIENTO continuo
  - a) R: Puerta cerrando y abriendo continuamente con 2 finales de carrera NO y con un pulsador de marcha NO y un de parada NT
  - b) OK: ... Idem pero que si cuando el paramos, giraba a la derecha, que al volver a poner en marcha continúe girando a la derecha (utilizando un flag)
  - c) OK +: ... Idem con un solo pulsador que pone en marcha y para (utilizando un altreFLAG)
- 10) P10-CAJA FUERTE con 10 pulsadores y motor con finales de carrera NT
  - a) R: Código clave de 4 cifras para abrir una caja fuerte (3 Flag)
  - b) OK Microrruptores puerta cerrada: motor abre pestillos si código = ok + microrruptor off. Y cierra los pestillos si se cierra la puerta.

### **PRÁCTICAS con ..... + TIM**

#### 11) P11-SEMÁFORO: (INCLUIR IMAGEN)

- a) R: Semáforo Kamikage: Control semáforo de peatones con sonido. Sol • litud por pulsador
- b) OK: Control 1 semáforo de peatones y 1 de coches con sonido.
- c) OK +: Control 2 semáforo de peatones y 2 coches con sonido. (USANDO OUT)

#### 12) P12-INTERMITENTES:

- a) R: control de los 2 intermitentes del coche con 2 pulsadores
- b) Ok: Incluir las otras 2 bombillas y un 3º Int de Peligro

#### 13) P13-RULETA GIRATORIA:

- a) R: Rueda a tiempo mínimo hasta que se suelta un pulsador (queda con la nueva posición)
- b) OK: se para a cabo de un tiempo de soltar el pulsador
- c) OK +: se para desacelerado

#### 14) P14-PUERTA DE SUPERMERCADO - preguntar qué debe hacer y qué hace falta de hard ???

- a) R: con el sensor abre, se para 4 segundos y cierra cuando es de días
- b) OK: idem pero que si detecta alguien mientras la puerta está abierta, reponga el tiempo de descuento y que vuelva a abrir.
- c) OK +: Cuando el encargado con la llave pone Int a noches, se cierra si está abierto
- d) OK ++ Las 3 cajas tienen un pulsador SOS bajo cada caja. Se cierra la puerta; suena una alarma; y hasta que el encargado no obra y cierra la llave del Int de día, no vuelve a abrirse la puerta

#### 15) P15-ASCENSOR - preguntar qué debe hacer y qué hace falta en el hard ???

- a) R: Control de un ascensor de tres plantas con 3 Puls, 3 LEDs; 3 FC\_NT;
- b) OK: con pulsadores de cabina: P0, P1, P2 y STOP
- c) OK +: con microrruptor de puerta de la cabina cerrada y si se ha pulsado stop, sólo los de la cabina.
- d) OK ++: con memoria

#### 16) P16-GARAJE - preguntar qué debe hacer y qué hace falta de hard ??? (2 finales de carrera, 1 detector reflectivo de objetos bajo la puerta y un pulsador P1 accionable interiormente, exteriormente con llave, o de dentro del coche con un mando.

- a) R: Sistema de funcionamiento particular. Al pulsar un único pulsador "P1" hace lo siguiente :. Si la puerta está cerrada, abre; si la puerta está abierta, la valla; si está abriendo, la detiene; si está cerrado, para y abre. Si está parada, la valla.
- b) OK: funcionamiento comunitario: 3 pulsadores P1, Ptops; P2. Si la puerta está cerrada o esta parada o está cerrando, al pulsar P1 abre. Cuando la puerta está abierta, pasados 3min o pulsando P2 se cierra. Al pulsar Pstops con la puerta abierta, la puerta abriendo o cerrado, para y deja de contar los 3 min para cerrar automáticamente (sólo cerrará pulsando P2)
- c) Aceptar ++: los 2 sistemas funcionan a la vez y dependen de un selector Isel = ON => Particular

#### 17) P17: BARRERA DE APARCAMIENTO

#### 18) P18: 3 EN LINEA: 9 PULSADORES, 9 LEDS ROJOS NUEVO LED VERDES; 2 LEDS INDICANDO EN TURNO; SI GANA EL VERDE, TODOS LOS LEDS VERDE parpado 5 SEGUNDOS Y VUELVE A EMPEZAR

### **PRÁCTICAS CON ..... + Contador**

#### 19) P19: CONTADOR

- a) R: Al pulsar 8 veces el pulsador CK se activa la bombilla. Al pulsar Reset el contador se inicializa de nuevo a 8 y se apaga la bombilla.
- b) OK: Al pulsar P se realiza un movimiento izquierda-derecha-izquierda... de 8 ciclos y se para.