

PRÁCTICA

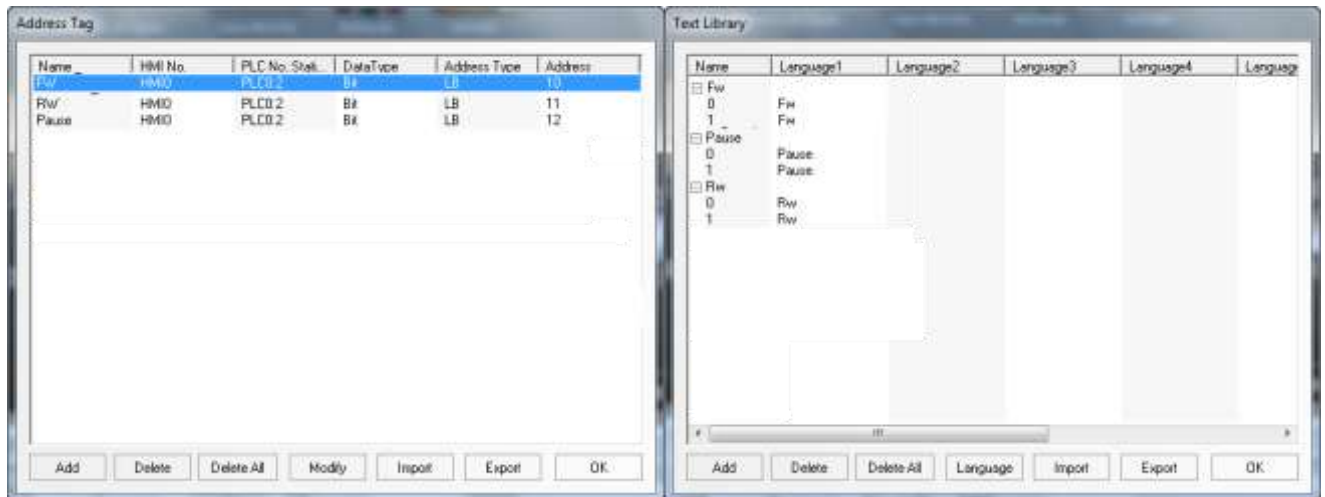
Proceso Principal

Curso Técnico de NBs
HMI Product Marketing
Diciembre 2012



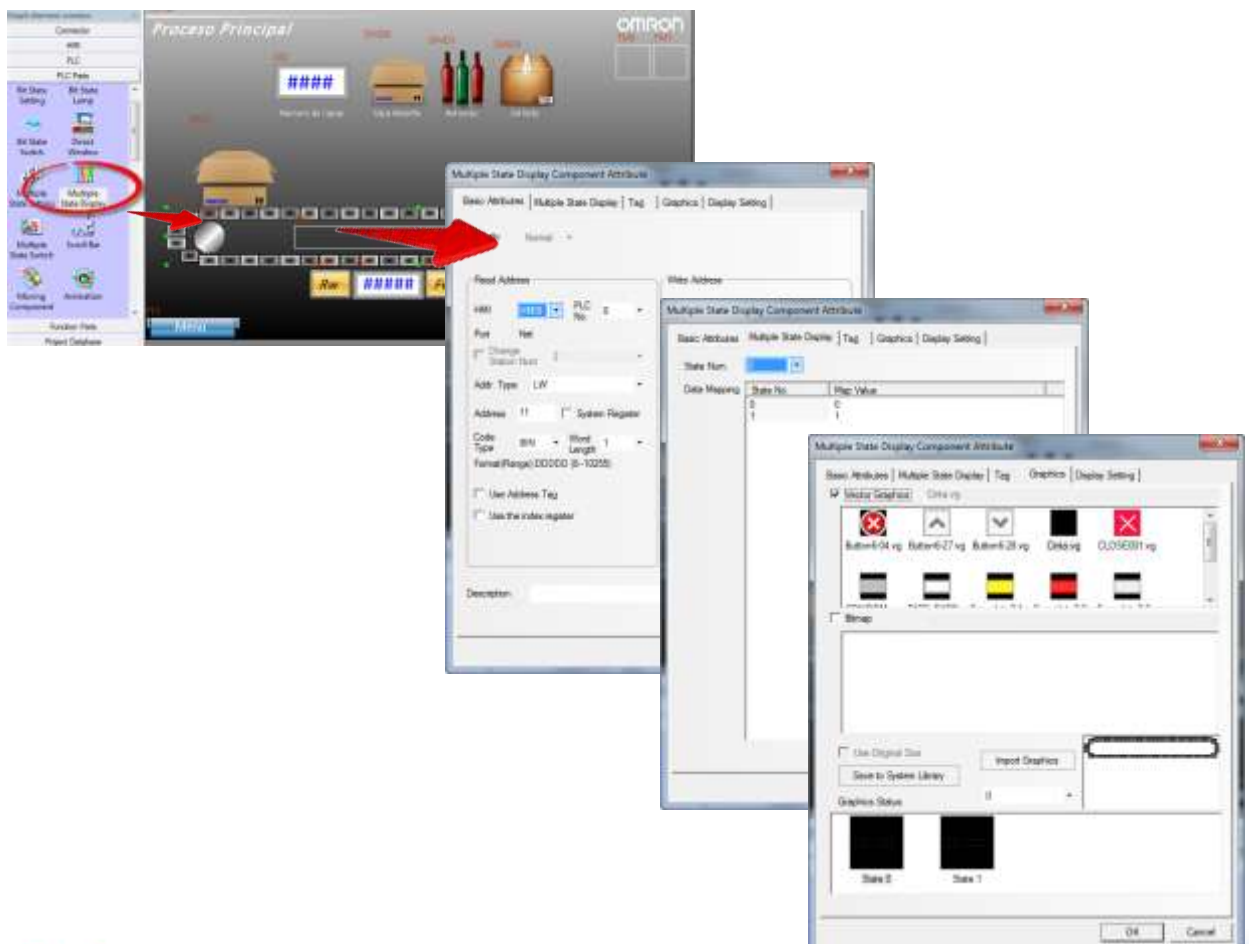
Paso 1.

Desde Project Database Tag Library y Text Library se crean los siguientes tags y textos que serán utilizados en el proceso principal



Paso 2.

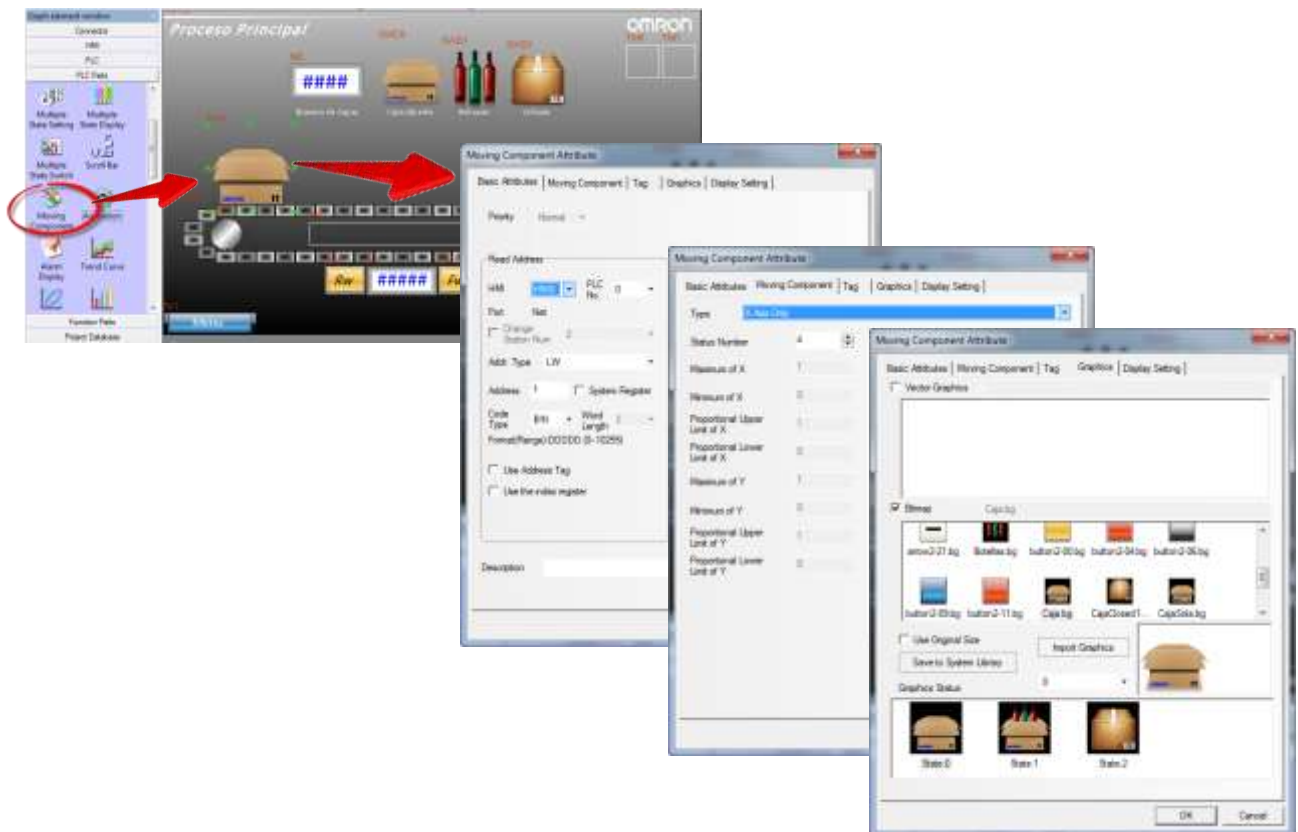
Ir a la Pantalla 1 e insertar desde PLC Parts el objeto Multiple State Display al cual se le asignará la imagen de la Cinta cargada anteriormente y cuyo estado será controlado por LW11



Realizar los mismos pasos para insertar el objeto rueda, direccionando su estado al área LW10

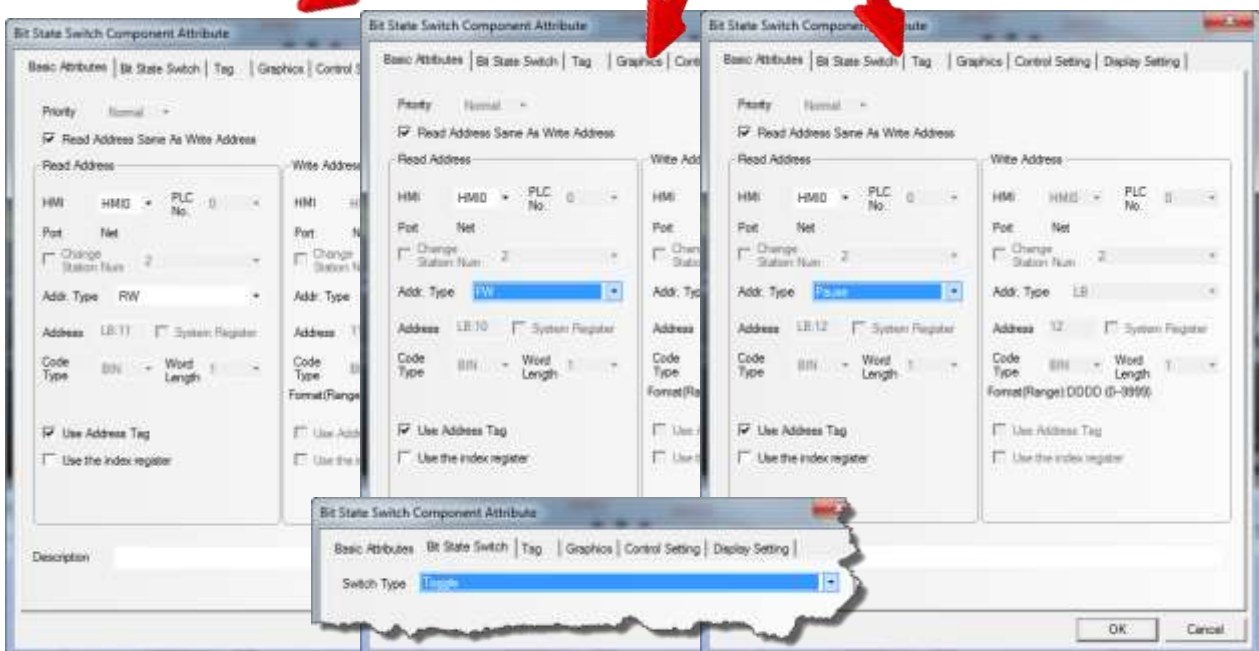
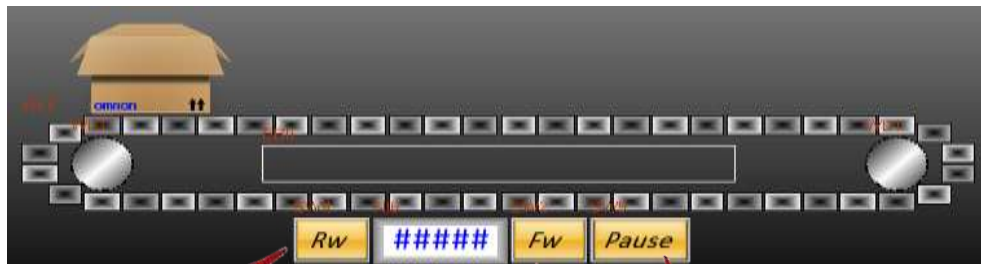
Paso 3.

En la misma página desde PLC Parts insertar Moving Components, en este caso se le asignará la imagen Caja y su estado será controlado por la dirección LW1



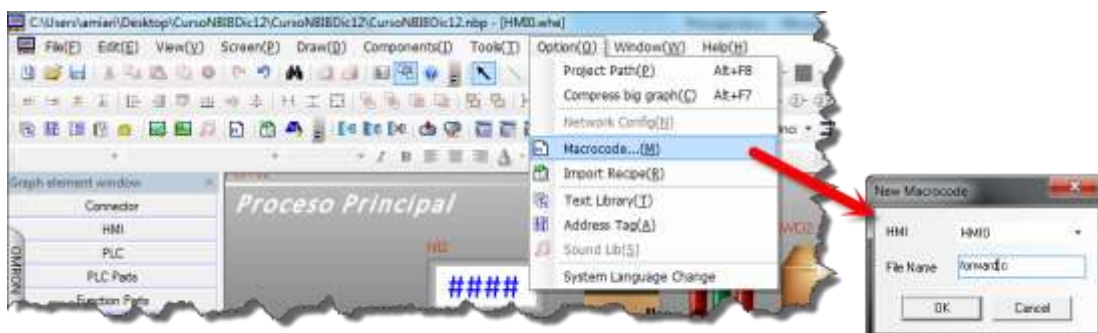
Paso 4.

Desde PLC Parts insertar tres Bit State Switch, direccionar dichos objetos a los Tags y Text Library creados con anterioridad



Paso 5.

Crear una macro que nos permitirá simular el movimiento hacia la derecha de la caja. Para ello se inserta una macro



Una vez creada la macro se definen las variables a utilizar

Parameters[forward.c]							
DataType	Param name	PLC No.	PLC Addres...	Address	Word...	OptMode	Array
signed short	iPosX		LW	2	1	Read/Write	No
bit	bPause		LB	12	1	Read/Write	No
bit	bEmergencia		LB	0	1	Read/Write	No
unsigned sh...	iMotor		LW	10	1	Read/Write	No
unsigned sh...	iCinta		LW	11	1	Read/Write	No
unsigned sh...	iNumCajas		LW	5	1	Read/Write	No

Y a continuación el código necesario para reproducir el movimiento

```

25  int MacroEntry()
26  = {
27    /*Movimiento circular caja. Comienza si no hay PAUSA y NO hay Alarma Emergencia*/
28    if ( (bPause == 0) && (bEmergencia == 0) )
29    = {
30      iPosX = iPosX + 10;      /*Movimiento de la caja FW*/
31      if (iPosX >= 540)      /*Si la posición de la caja es mayor que 540, que vaya al comienzo*/
32      = {
33        iPosX = 0;
34        iNumCajas++;
35      }
36      /*Animación cinta y motores*/
37      if (iMotor >= 3)
38      | iMotor = 0;
39      else
40      iMotor++;
41
42      if (iCinta >= 1)
43      | iCinta = 0;
44      else
45      iCinta++;
46    }
47    return 0;
48  }
49  }

```

Paso 6.

De igual manera que en el punto anterior se genera una macro "reverse" que permitirá el movimiento hacia la izquierda de la caja donde se declaran las mismas variables

Parameters[forward.c]							
DataType	Param name	PLC No.	PLC Addres...	Address	Word...	OptMode	Array
signed short	iPosX		LW	2	1	Read/Write	No
bit	bPause		LB	12	1	Read/Write	No
bit	bEmergencia		LB	0	1	Read/Write	No
unsigned sh...	iMotor		LW	10	1	Read/Write	No
unsigned sh...	iCinta		LW	11	1	Read/Write	No
unsigned sh...	iNumCajas		LW	5	1	Read/Write	No

Y cuyo código es muy similar al anterior.

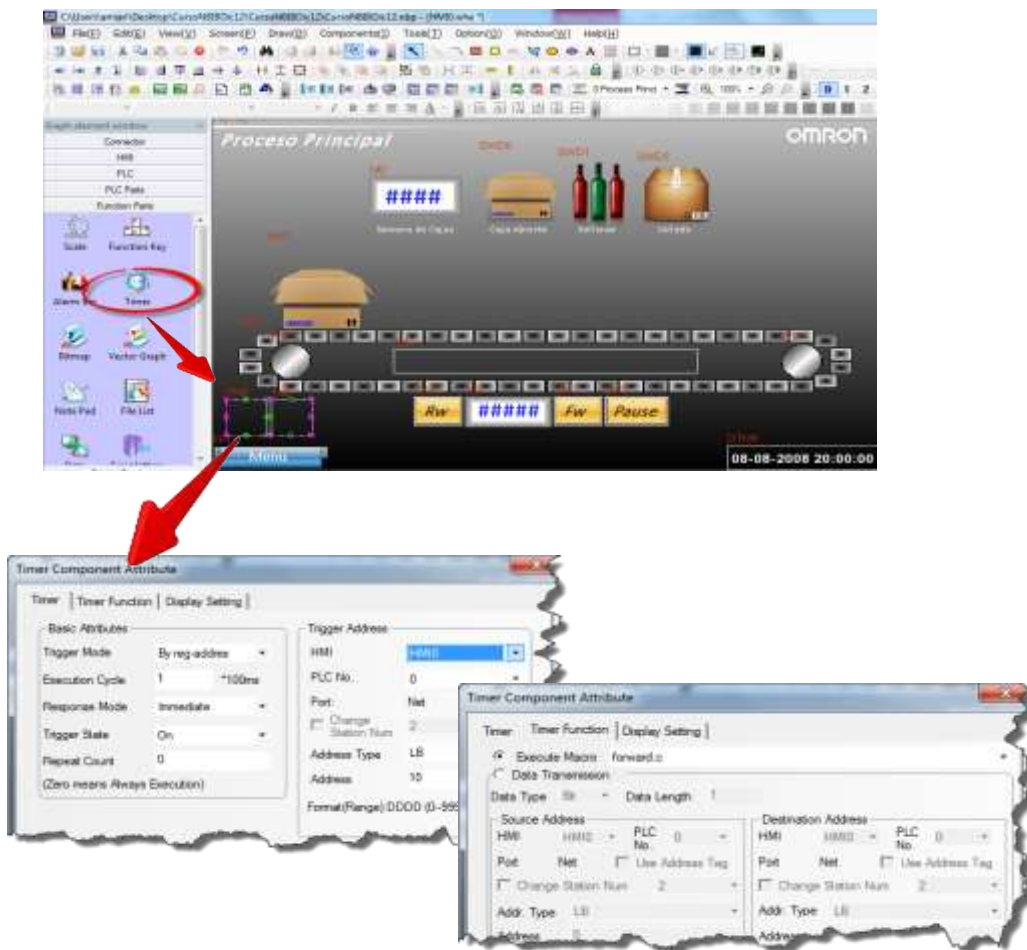
```

22 WriteLocal("LW", 202, 2, (void*)buf, 0);
23
24 int MacroEntry()
25 {
26     /*Animación caja reverso, más cinta y motores - sin no hay pausa y No alarma de emergencia*/
27     if ( (bPause == 0) && (bEmergencia == 0) )
28     {
29         iPosX = iPosX - 10;      /*Movimiento de la caja RW*/
30         if (iPosX <= 0)        /*Si posición menor que 0, que empiece desde el final*/
31         {
32             iPosX = 540;
33             iNumCajas++;
34         }
35         /*Animación cinta y motores*/
36         if (iMotor <= 0)
37         | iMotor = 3;
38         else
39         | iMotor--;
40         if (iCinta >= 1)
41         | iCinta = 0;
42         else
43         | iCinta++;
44     }
45     return 0;
46 }

```

Paso 7.

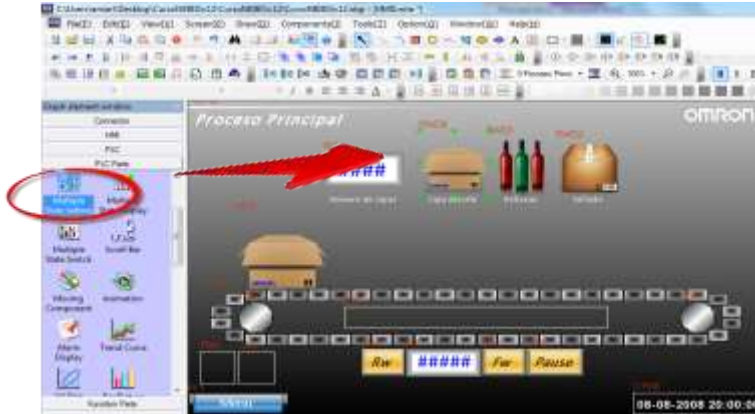
De vuelta en la página principal desde Function Parts se insertan dos objetos Timer cuya ejecución queda supedita a que se pulse sobre los botones Fw o Rv y que van a controlar el tiempo de lanzamiento de cada una de las macro y por tanto el ciclo de movimiento de la cinta.



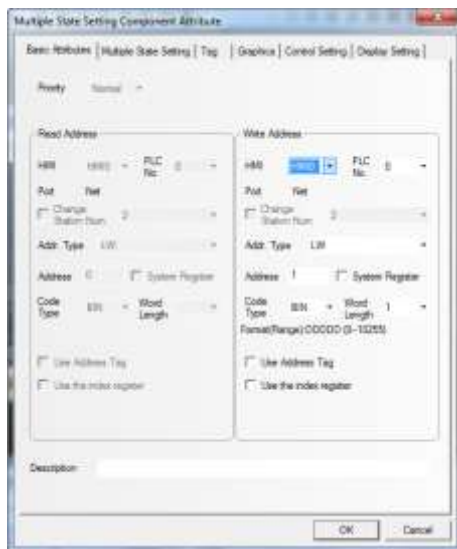
Repetir el mismo proceso para el Timer 2 asociándole el botón RV y la macro correspondiente.

Paso 8. Opcional

Desde PLC Part Insertar tres objetos Multiple State Setting,



Cada uno de ellos controlará el estado a mostrar de la caja, por ello todos deben tener asignada el área de memoria LW1



Y se cargará un valor diferente para cada botón

