



MANUAL INTRODUCTORIO PLC KDN-K3 Serie 304/306

Instrucciones y Software

Carga de un programa al PLC KDN.

Ejecutaremos el EasyPog. Dentro de **(File, Open Project)** seleccionaremos el proyecto a ejecutar.

Una vez cargado el archivo en el editor, conectaremos el PLC con el cable de programación.

En **(Tools, Communications)** realizaremos un Search para detectar el equipo, una vez encontrado el PLC, daremos Ok. Es importante no estar ejecutando otros programas a la vez.

Ahora si podemos realizar la carga al equipo, **(PLC, Download)**.

Nos preguntara que para realizar la carga, el PLC interrumpira la ejecucion del programa en curso, diremos que si, y en el cuadro de texto inferior nos anunciara si la carga fue exitosa.

Tipos de datos:

TIPO	DESCRIPCION	TAMANO EN BITS	RANGO DE VALORES	VALOR INICIAL
BOOL	booleano	1	True/false	Falso
BYTE	Cadena de 8bits	8	0~255	0
WORD	Cadena de 16bits	16	0~65535	0
DWORD	Cadena de 32bits	32	0~4.294.967.295	0
INT	Entero con signo	16	-2e15~(2e15-1)	0
DINT	Doble entero con signo	32	-2e31~(2e31-1)	0
REAL	Numero Punto flotante	32	1.8e-38~3.40e38 -3.40e38~1.18e-38	0.0

Tipo de memoria, características:

I (entradas digitales)

TIPO DE ACCESO	Por bit (solo leemos una entrada digital por vez) %Ix.y x:byte de direccion de la variable y:bit del byte de la variable (entre 0~7) Ej: %I0.0, %I0.7, %I1.0
Aqui se describe solo el acceso por bit, pudiendose hacer tambien por byte, word, doble word.	
TIPO DE DATO	BOOL
ACCESO CORRECTO	Solo lectura

Q (salidas digitales)	
TIPO DE ACCESO	Por bit (solo leemos/escibimos una salida digital por vez) %Qx.y x:byte de direccion de la variable y:bit del byte de la variable (entre 0~7) Ej: %Q0.0, %Q0.7, %Q1.0
Aqui se describe solo el acceso por bit, pudiendose hacer tambien por byte, word, doble word.	
TIPO DE DATO	BOOL
ACCESO CORRECTO	Lectura/escritura

AI (entradas analogicas)	
TIPO DE ACCESO	Por word (leemos una entrada analogica) %AIWx x:byte de direccion de la variable Ej: %AIW0, %AIW2, %AIW12
TIPO DE DATO	INT
ACCESO CORRECTO	Lectura/escritura

AQ (salidas analogicas)	
TIPO DE ACCESO	Por word (leemos/escibimos una entrada analogica) %AQWx x:byte de direccion de la variable Ej: %AQW0, %AQW2, %AQW12
TIPO DE DATO	INT
ACCESO CORRECTO	Lectura/escritura

M (Area de memoria interna, propicia para la operacion con bits)

TIPO DE ACCESO	Por bit (leemos/ escribimos un bit) %Mx.y x:byte de direccion de la variable y:bit del byte de la variable (entre 0~7) Ej: %M0.0, %M0.7, %M1.0
TIPO DE DATO	BOOL
ACCESO CORRECTO	Lectura/escritura

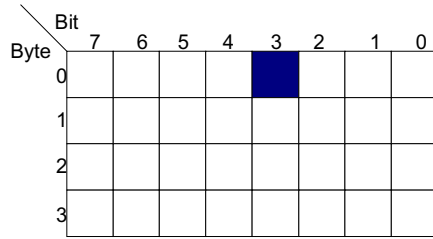
V (Area de memoria interna, usada para almacenar gran cantidad de datos)	
TIPO DE ACCESO	<p>Por bit (leemos/ escribimos un bit) %Vx.y x:byte de direccion de la variable y:bit del byte de direccion de la variable(entre 0~7) Ej: %V0.0, %V0.1, %V4.6</p> <p>Por byte (leemos/ escribimos un byte) %VBx x:byte de direccion de la variable Ej: %VB0, %VB1, %VB14</p> <p>Por word (leemos/ escribimos una word) %VWx x:byte de comienzade direccion de la variable (debe ser numero par) Ej: %VW0, %VW2, %VW14</p> <p>Por dobleWORD, dobleINT (leemos/ escribimos una dWord o dINT) %VWx x:byte de comienzo de direccion de la variable (debe ser par y ocupa 4 bytes) Ej: %VD0, %VD4, %VD14</p>
TIPO DE DATO	DWORD, DINT: REAL (x debe ser en un rango especifico ver nota sobre numeros reales)
ACCESO CORRECTO	Lectura/escritura

HC (Area de memoria de contadores rapidos)

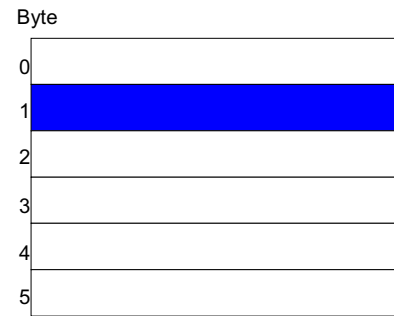
TIPO DE ACCESO	Por DWORD (leemos el valor actual de un contador rapido) %HCx x:numero de contador rapido a leer Ej: %HC0, %HC1
TIPO DE DATO	DWORD
ACCESO CORRECTO	Lectura

SM (Area registros internos)	
TIPO DE ACCESO	<p>Por bit (leemos/escribimos un bit) %SMx.y x:byte de direccion de la variable y:bit del byte de direccion de la variable(entre 0~7) Ej: %SM0.0, %SM0.1, %SM4.6</p> <p>Por byte (leemos/escribimos un byte) %SMBx x:byte de direccion de la variable Ej: %SMB0, %SMB1, %SMB14</p> <p>Por word (leemos/escribimos una word) %SMWx x:byte de comienzade direccion de la variable (debe ser numero par) Ej: %SM0, %SM2, %SM14</p> <p>Por dobleWORD, dobleINT (leemos/escribimos una dWord o dINT) %SMDx x:byte de comienzo de direccion de la variable (debe ser par y ocupa 4 bytes) Ej: %SM0, %SM4, %SM14</p>
TIPO DE DATO	DWORD, DINT: REAL (x debe ser en un rango especifico ver nota sobre numeros reales)
ACCESO CORRECTO	Lectura/escritura

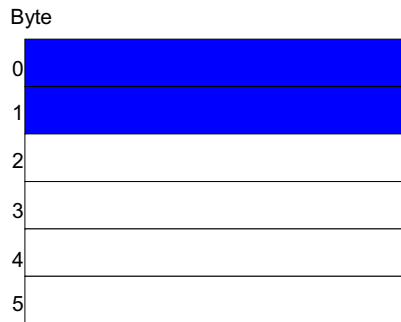
Selección de un bit
Ej en memoria M: %M0.3



Selección de un byte
Ej en memoria V: %VB1

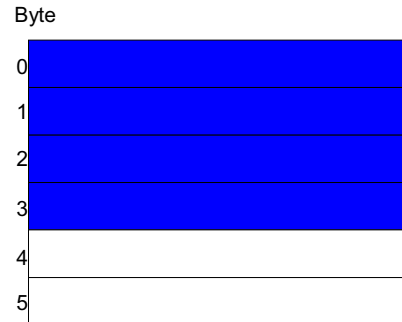


Selección de una Word
Ej en memoria V: %VW0



Nota: una word ocupa 2 bytes y siempre se direcciona en numeros pares.

Selección de DWord/DInt/Real
Ej en memoria V: %VD0



Nota: una Dword ocupa 4bytes y siempre se direcciona en numeros pares.

Componentes de Proyecto:

Programa	Initial Data	Podemos asignar valores numericos iniciales, ya sean Bit, Byte, Word, Int, Dword, Dint o Real
	Main Program	Programa principal, la CPU, lo ejecuta una vez por vuelta de programa y solo uno, puede existir en el proyecto.
	Subroutines	Se ejecutan al ser invocadas desde el programa principal. Nos ayudan a tener una mejor estructura de programa.
	Interrupt	No se ejecutan al ser invocadas desde el programa principal, sino, que lo hacen en casos particulares. Ver pag...
Configuraciones	Hardware	Aqui se pueden configurar el modelo KDN-K3 a utilizar, con sus modulos de expansion correspondientes. Tambien el seteo del puerto de comunicacion, area de variables con retencion temporal.
	Global Variables	Aqui se definen las variables a utilizar en forma global, es decir tanto en el programa principal, subrutinas o interrupciones.

TEMPORIZADORES Y CONTADORES

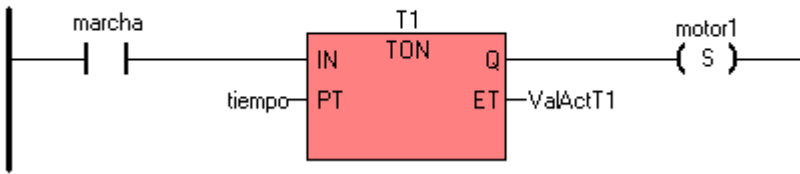
		CPU 304	CPU 306	
Temporizadores	Cantidades	64	128	
	Nombres	T0--T63	T0--T127	
	Base de Tiempo	T0--T3	1ms	T0--T3 1ms
		T4--T19	10ms	T4--T19 10ms
		T20--T63	100ms	T20--T127 100ms
Tiempo Maximo	32767	32767		
Contadores	Cantidades	64	128	
	Nombres	C0--C63	C0--C127	
	Cuenta Maxima	32767	32767	

Temporizador:

Los hay de tres tipos TON, TOFF, TP.

El mismo empezara a temporizar a partir de un flanco ascendente por la entrada IN; cuando ET (valor actual) sea igual o mayor que PT (tiempo prefijado), tendremos un 1 (True) por la salida Q.

(* Network 0 *)
 (* tiempo1(T1) *)



Symbol	Address	Data Type	Comment
1	marcha	%I0.0	BOOL
2	motor	%Q0.0	BOOL
3	tiempo	%VW0	INT
4	ValActT1	%VW2	INT
5			
6			

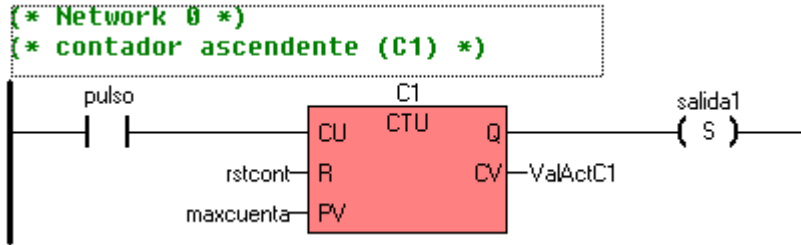
Con la siguiente tabla, tendremos el tipo de variables aceptadas en cada una de las entradas o salidas del bloque.

Operands	Input/Output	Data Type	Acceptable Memory Areas
<i>Tx</i>	-	Timer instance	T
<i>IN</i>	Input	BOOL	Power flow
<i>PT</i>	Input	INT	I, AI, AQ, M, V, L, SM, constant
<i>Q</i>	Output	BOOL	Power flow
<i>ET</i>	Output	INT	Q, M, V, L, SM, AQ

Contador:

Tendremos CTU(contador ascendente) y CTD (contador descendente)

Por cada flanco ascendente por la entrada CU, el contador incrementara (si endo CTU), o decrementara (al ser CTD). Cuando el valor actual(CV) sea igual o mayor a PV, la salida Q tomara valor (True)1.



MAIN * VAR_GLOBAL		Address	Data Type	Comment
1	pulso	%I0.0	BOOL	
2	rstcont	%I0.1	BOOL	
3	salida1	%Q0.0	BOOL	
4	maxcuenta	%VW0	INT	
5	ValActC1	%VW2	INT	
6				

Con la siguiente tabla, tendremos el tipo de variables aceptadas en cada una de las entradas o salidas del bloque.

Operands	Input/Output	Data Type	Acceptable Memory Areas
<i>Cx</i>	-	Counter instance	C
<i>CU</i>	Input	BOOL	Power flow
<i>R</i>	Input	BOOL	I, Q, M, V, L, SM, T, C
<i>CD</i>	Input	BOOL	Power flow
<i>LD</i>	Input	BOOL	I, Q, M, V, L, SM, T, C
<i>PV</i>	Input	INT	I, AI, AQ, M, V, L, SM, constant
<i>Q</i>	Output	BOOL	Power flow
<i>CV</i>	Output	INT	Q, M, V, L, SM, AQ

DESCRIPCION DE REGISTROS INTERNOS

SMB0:

SM Bit	Description
SM0.0	Always ON
SM0.1	ON during the first scan cycle only. Usually used for some initializations.
SM0.2	If the data in RAM is lost, this bit is ON during the first scan cycle, and later cleared to FALSE.
SM0.3	Provide a pulse train (50% duty cycle) with a cycle time of 1s.
SM0.4	Provide a pulse train (50% duty cycle) with a cycle time of 2s.
SM0.5	Provide a pulse train (50% duty cycle) with a cycle time of 3s.
SM0.6	Provide a pulse train (50% duty cycle) with a cycle time of 60s.
SM0.7	Reserved

SMW26 –SMW28:

Estas direcciones estan asociadas a los dos potenciometros en la CPU del equipo.

SMB31-SMW32

Control para almacenar los datos en FRAM

TIPOS DE MEMORIA CON RETENCION

En el PLC KDN-K3, existen dos maneras de guarda datos, una de forma temporal y otra permanente.

-Temporal: Los datos son retenidos por 72 horas gracias a un Super Capacitor. En la solapa 'Retentive Range', dentro de los seteos de Hardware, se pedran seleccionr los registros a retener.

-Permanente: Los datos se almacenan en un a memoria del tipo FRAM, la misma asegura la informacion de manera permanente, el manejo de la misma se detalla a continuacion

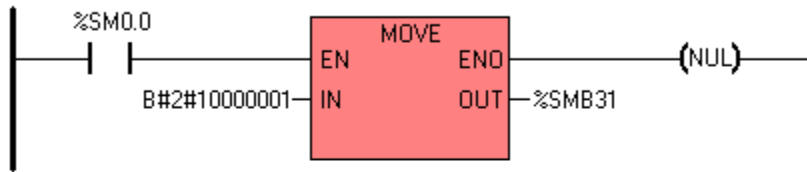
Salvando en FRAM

Las variables a salvar deben estar dentro de estos rangos

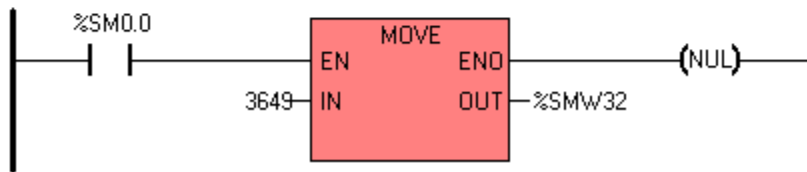
	CPU304	CPU306
Length	128 bytes	255 bytes
Range	VB1648~VB1775	VB3648~VB3902

Ejemplos

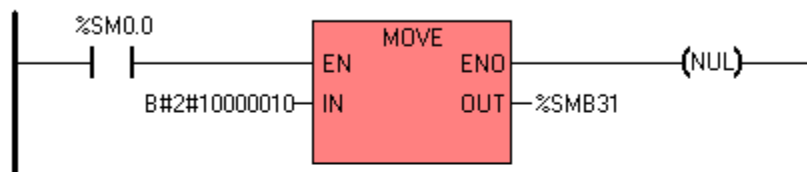
(* Network 3 *)
(* Definiendo salvar 1 Byte *)



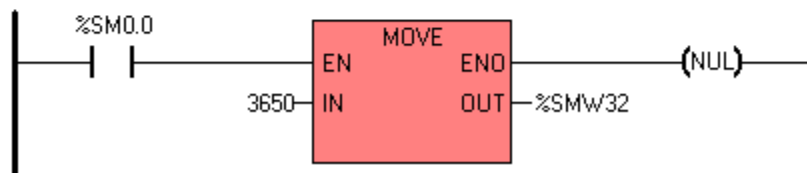
(* Network 4 *)
(* Salvado UB3649 *)



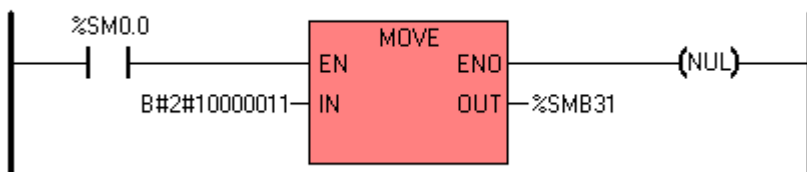
(* Network 3 *)
(* Definiendo salvar 2 Byte *)



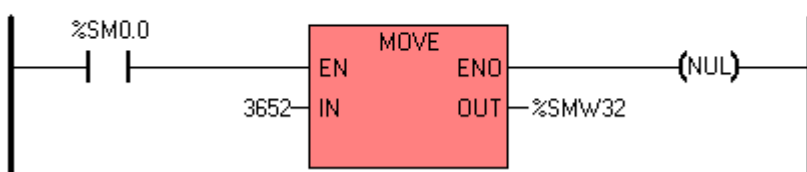
(* Network 4 *)
(* Salvado UW3650 *)



(* Network 3 *)
(* Definiendo salvar 4 Byte *)



(* Network 4 *)
(* Salvado UD3652 *)



CONTADORES RAPIDOS

Feature	CPU304	CPU306
High-speed counters	2 counters (HSC0 and HSC1)	6 counters (HSC0 to HSC5)
Single phase	2 at 20KHz	6 at 30KHz
Two phase	2 at 10KHz	4 at 20KHz.

Modos de operacion:

Para elegir el correcto contador y su modo de operacion, debemos saber, si daremos de forma externa la senal de START, RESET, DIRECCION, y si conectaremos las dos fases del encoder, para tener cuenta ascendente y desendente, y la CPU con la que estamos trabajando.

Contador	Modos de operacion
HSC0	7
HSC1	11
HSC2	11
HSC3	1
HSC4	7
HSC5	1

HSC 0				
Mode	Description	I0.1	I0.0	I0.5
0	Single-phase up/down counter with internal direction control	Clock		
1			Reset	
2			Reset	Start
3	with external direction control			Direction
4			Reset	Direction
6	Two-phase counter with up/down clock inputs	Clock Up	Clock Down	
9	A/B phase quadrature counter	Clock B	Clock A	

HSC 1					
Mode	Description	I0.3	I0.7	I1.2	I1.3
0				Clock	
1		Reset			
2		Reset	Start		
3	Single-phase up/down counter				Direction
4		Reset			Direction
5		Reset	Start		Direction
6	Two-phase counter with up/down clock inputs			Clock Down	Clock Up
7		Reset			
8		Reset	Start		
9	Free counter			Clock B	Clock A
10		Reset			
11		Reset	Start		

HSC 2					
Mode	Description	I0.6	I1.1	I1.4	I1.5
0	Single-phase up/down counter with external direction control			Clock	
1		Reset			
2		Reset	Start		
3	Single-phase up/down counter with external direction control				Direction
4		Reset			Direction
5		Reset	Start		Direction
6	with up/down clock inputs			Clock Down	Clock Up
7		Reset			
8		Reset	Start		
9				Clock B	Clock A
10		Reset			
11		Reset	Start		

HSC 3		
Mode	Description	I0.0
0	Single-phase up/down counter with internal direction control	Clock

HSC 4				
Mode	Description	I0.2	I1.0	I1.1
0	Single-phase up/down counter	Clock		
1			Reset	
2			Reset	Start
3	Single-phase up/down counter with external direction control	Clock		Direction
4			Reset	Direction
6	Two-phase counter with up/down clock inputs	Clock Down	Clock Up	
9	A/B phase quadrature counter	Clock B	Clock A	

HSC 5		
Mode	Description	I0.3
0	Single-phase up/down counter with internal direction control	Clock

HCS0	HCS1	HCS2	HCS4	DESCRIPCION
SM37.0	SM47.0	SM57.0	SM147.0	RESET 0=Activado en alto 1=Activado en bajo
SM37.1	SM47.1	SM57.1	SM147.1	START 0=Activado en alto 1=Activado en bajo
SM37.2	SM47.2	SM57.2	SM147.2	Factor de multiplicacion 0=multiplco x4 1=multiplco x1

Graficando todos los modos;

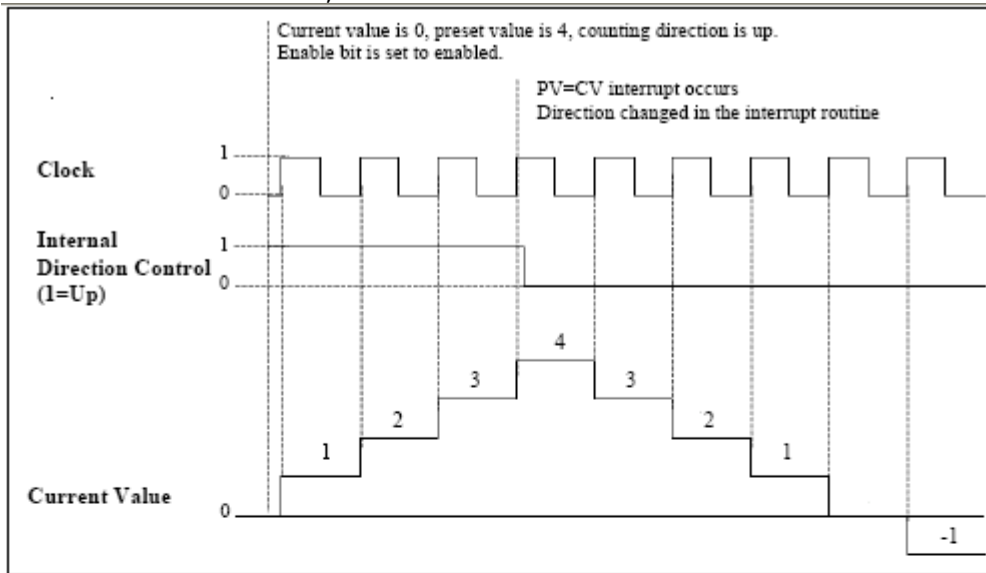


Figure 6-4 Time Sequence of Mode 0, 1 or 2

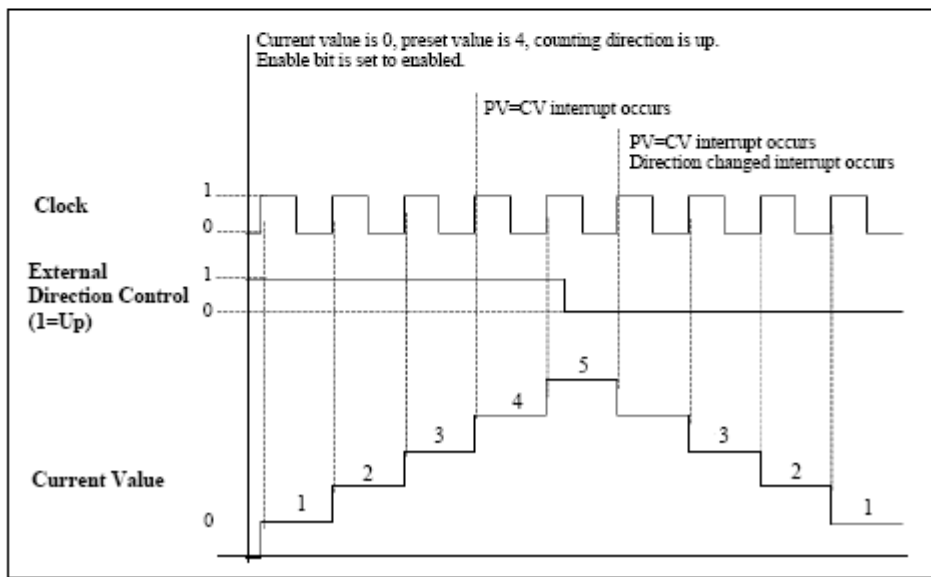


Figure 6-5 Time Sequence of Mode 3, 4 or 5

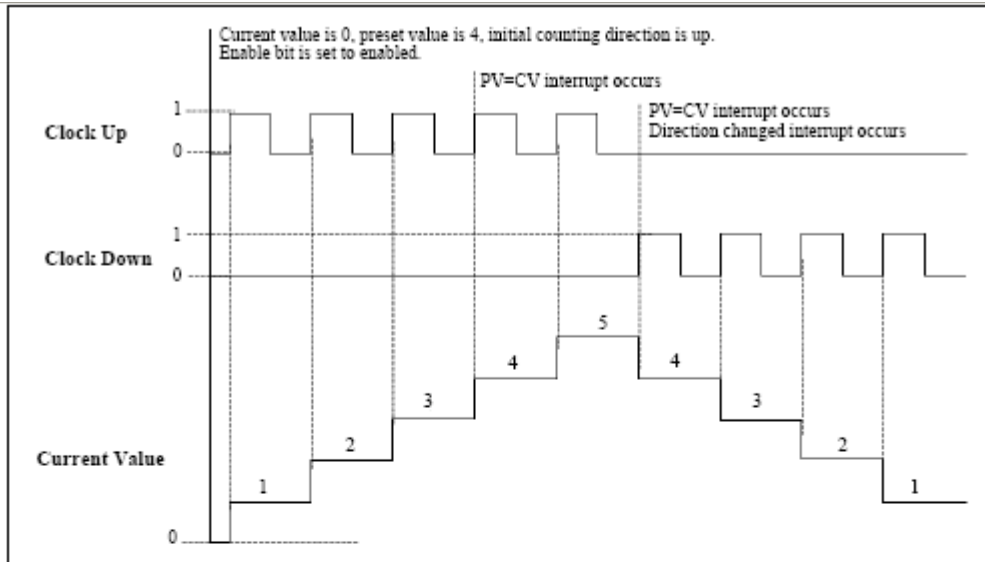


Figure 6-6 Time Sequence of Mode 6, 7 or 8

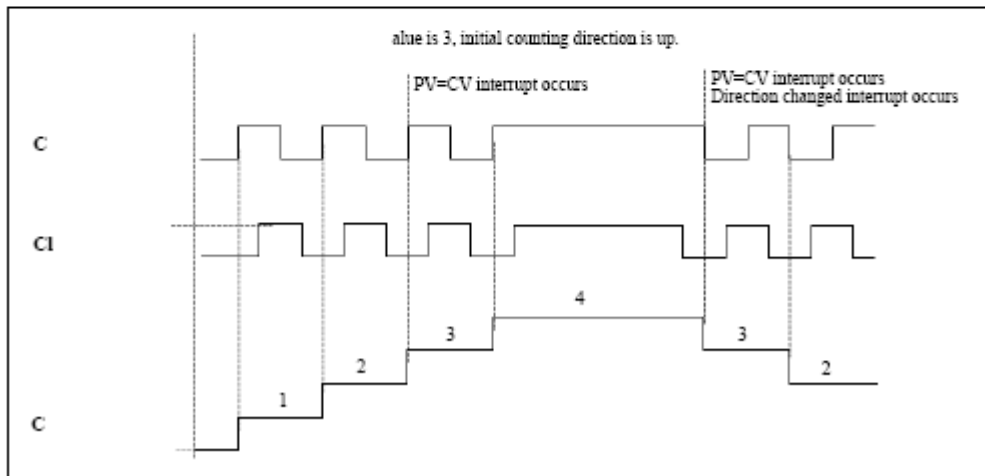


Figure 6-7 Time Sequence of Mode 9, 10 or 11 (Quadrature, 1x rate)

HCS0	HCS1	HCS2	HCS4	DESCRIPCION
SM37.0	SM47.0	SM57.0	SM147.0	RESET 0=Activado en alto 1=Activado en bajo
SM37.1	SM47.1	SM57.1	SM147.1	START 0=Activado en alto 1=Activado en bajo
SM37.2	SM47.2	SM57.2	SM147.2	Factor de multiplicacion 0=multiplco x4 1=multiplco x1

HSC0	HSC1	HSC2	HSC3	HSC4	HSC5	DESCRIPCION
SM37.3	SM47.3	SM57.3	SM137.3	SM147.3	SM157.3	Direccion de conteo 1=up; 0=down
SM37.4	SM47.4	SM57.4	SM137.4	SM147.4	SM157.4	Permite escribir la direccion de conteo 0=NO; 1=SI
SM37.5	SM47.5	SM57.5	SM137.5	SM147.5	SM157.5	Permite escribir el nuevo valor de preset 0=NO; 1=SI
SM37.6	SM47.6	SM57.6	SM137.6	SM147.6	SM157.6	Permite escribir el nuevo valor actual de cuenta 0=NO; 1=SI
SM37.7	SM47.7	SM57.7	SM137.7	SM147.7	SM157.7	Habilitar el HCS 0=Deshab; 1=Habilita

	HSC0	HSC1	HSC2	HSC3	HSC4	HSC5
Nuevo valor actual	SMD38	SMD48	SMD58	SMD138	SMD148	SMD158
Nuevo valor de preset	SMD42	SMD52	SMD62	SMD142	SMD152	SMD162

Teniendo en claro que contador usar y su modo de operacion, pasaremos a programarlo.

_Programando un contador rapido

- Definir registro (SMB37 en HSC0 ,SMB47 en HSC1 , SMB57 en HSC2 , SMB137 en HSC3, SMB147 en HSC4 y SMB157 para HSC5)
- Definir el contador y su modo usando la instruccion HDEF.
- Asignando un valor actual y un preset.
- Dando inicio al contador rapido con la instruccion HSC.

La estas definiciones las ejecutaremos solo una vez cuando la CPU empiece a ejecutar el programa, para eso usaremos SM0.1, que solo se pondra en 1 en el primer scan de programa.

Por ejemplo:

Para HSC2
SMB57= b#16#F8

F8 en binario 11111000

SMB57.0=0 Reset activo, en estado alto
SMB57.1=0 Start activo, en estado alto
SMB57.2=0 Multiplico x4
SMB57.3=1 Conteo ascendente
SMB57.4=1 Permite cambiar el sentido de conteo
SMB57.5=1 Permite escribir un nuevo valor de preset

SMB57.6=1 Permite escribir un nuevo valor de preset
 SMB57.7=1 Habilita el contador HSC2

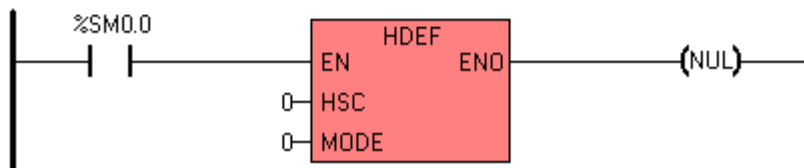
Registros internos de solo lectura, en relacion a los contadores rapidos

HSC0	HSC1	HSC2	HSC3	HSC4	HSC5	Description
SM36.0	SM46.0	SM56.0	SM136.0	SM146.0	SM156.0	Reserved
SM36.1	SM46.1	SM56.1	SM136.1	SM146.1	SM156.1	Reserved
SM36.2	SM46.2	SM56.2	SM136.2	SM146.2	SM156.2	Reserved
SM36.3	SM46.3	SM56.3	SM136.3	SM146.3	SM156.3	Reserved
SM36.4	SM46.4	SM56.4	SM136.4	SM146.4	SM156.4	Reserved
SM36.5	SM46.5	SM56.5	SM136.5	SM146.5	SM156.5	Current counting direction: 0 = Down; 1 = Up
SM36.6	SM46.6	SM56.6	SM136.6	SM146.6	SM156.6	Current value equal to preset value: 0 = Not equal; 1 = Equal
SM36.7	SM46.7	SM56.7	SM136.7	SM146.7	SM156.7	Current value greater than preset value: 0 = Not greater than; 1 = Greater than

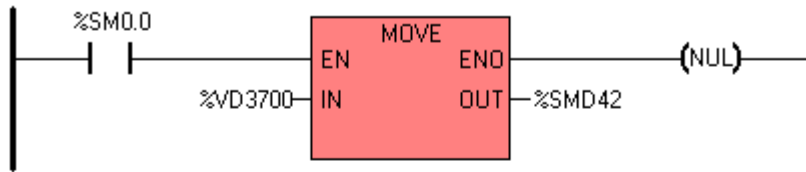
(* Network 0 *)
 (* Inicializa contador rapido x1 *)



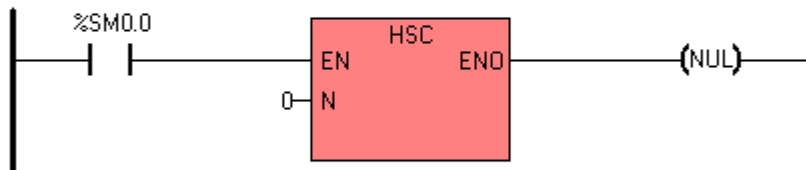
(* Network 1 *)
 (* Modo 0 - Single Phase *)



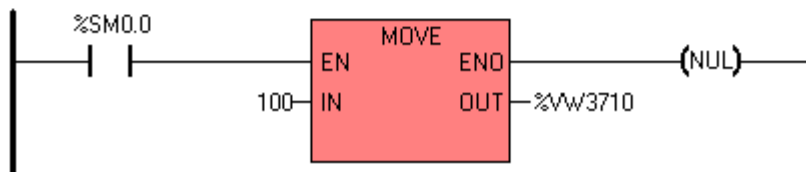
(* Network 2 *)



(* Network 3 *)
(* Contador Rapido 0 *)



(* Network 4 *)



SALIDA RAPIDA DE PULSOS

El KDN-K3 provee de dos generadores de pulsos PTO/PWM a una frecuencia maxima de 20KHz. Las salidas asignadas seran la Q0.0 llamada PTO0 o PWM0 y la Q0.1 llamada PTO1 o PWM1. Q0.0 y Q0.1 no deben ser salidas a rele.

PWM_ modulación por ancho de pulso

Tendremos una salida continua de pulso, pudiendo cambiar, el ciclo de tiempo, y el ancho del pulso. Si el ancho del pulso es mayor que el ciclo de tiempo, tendremos un TON al 100%, teniendo una salida continua. Si el ancho del pulso es 0, tendremos un TON al 0%, y la salida permanecera apagada.

- Actualizacion Sincronica : las variaciones de PWM las haremos sobre el ancho de pulso (TON), permaneciendo constante la frecuencia.
- Actualizacion Asincronica: si queremos modificar la base de tiempo tendremos una actualizacion asincronica.

PTO_salida de tren de pulsos

PTO provee de una onda cuadrada con un TON del 50%

Usando PTO podemos generar variaciones en la salida de pulsos divididos en diferentes segmentos.

Multi segmentos

La CPU lee automaticamente la configuracion cada tren de pulsos de la Tabla Profile

Tabla de configuracion de registros PTO / PWM

Q0.0	Q0.1	BITS DE CONTROL
SM67.0	SM77.0	(PTO/PWM) Actualizar el tiempo de ciclo 0=no actualizar 1= actualizar
SM67.1	SM77.1	(PWM) Actualizar el ancho de pulso 0=no actualizar 1= actualizar
SM67.2	SM77.2	(PTO) Actualizar la cantidad de pulsos 0=no actualizar 1= actualizar
SM67.3	SM77.3	(PTO/PWM) Actualizar la base de tiempo 0=1 μ s 1= 1ms
SM67.4	SM77.4	(PWM) Modo de actualizacion 0=asincronico 1= sincronico
SM67.5	SM77.5	(PTO) Simple u operacion multi segmento 0=simple 1= multi segmento
SM67.6	SM77.6	Seleccion entre modo PTO o PWM 0=PTO 1= PWM
SM67.7	SM77.7	(PTO/PWM) Habilitacion 0=deshabilitado 1= habilitado
Q0.0	Q0.1	OTROS BITS DE CONTROL
SMW68	SMW78	(PTO/PWM) valor del ciclo de tiempo, entre 2 y 65535
SMW70	SMW80	(PWM) valor del ancho del pulso, entre 0 y 65535
SMD72	SMD82	(PTO) cantidad de pulsos, entre 1 y 4.294.967.295

Configuracion Multi segmentos.

Byte offset	Largo	Descripcion	Segmento
0	16-bit	La cantidad de segmentos	1
1	16-bit	Tiempo de ciclo inicial (2 a 65535 tiempos de la base de tiempo)	
3	16-bit	Incrementos de tiempo en cada pulso (-32768 a 32768 tiempos de la base de tiempo)	
5	32-bit	Cant de pulsos (1 a 4.294.967,295)	2
9	16-bit	Tiempo de ciclo inicial (2 a 65535 tiempos de la base de tiempo)	
11	16-bit	Incrementos de tiempo en cada pulso (-32768 a 32768 tiempos de la base de tiempo)	
13	32-bit	Cant de pulsos (1 a 4.294.967,295)	

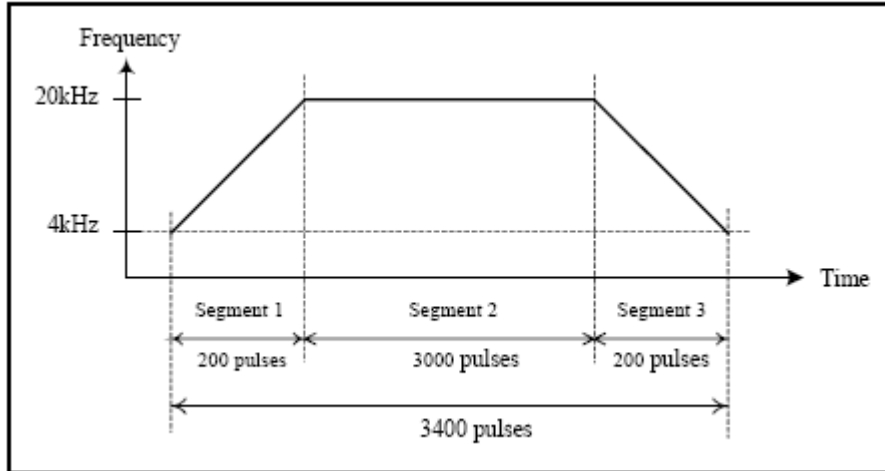
Los Byte offset, son relativos a la posicion de la tabla profile.

SMB166	SMB176	El numero de los segmentos en progreso.
SMW168	SMW178	La locacion de inicio de la tabla profile.

La siguiente tabla, describe el estado de los bits.

Q0.0	Q0.1	Estado del bit
SM66.4	SM76.4	PTO
SM66.5	SM76.5	
SM66.6	SM76.6	
SM66.7	SM76.7	PTO terminado 0=en progreso 1=terminado

Ejemplo Multi segmento



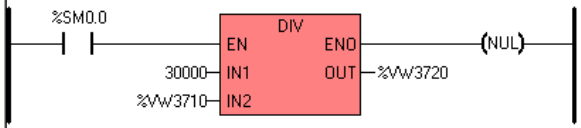
Ejemplo de una tabla profile:

Byte Offset	Value	Comment	
VB701	3	The number of segments	
VW702	250	Initial cycle time	Segment 1
VW704	-1	Cycle time increment	
VD706	200	The number of pulses	
VW710	50	Initial cycle time	Segment 2
VW712	0	Cycle time increment	
VD714	3000	The number of pulses	
VW718	50	Initial cycle time	Segment 3
VW720	1	Cycle time increment	
VD722	200	The number of pulses	

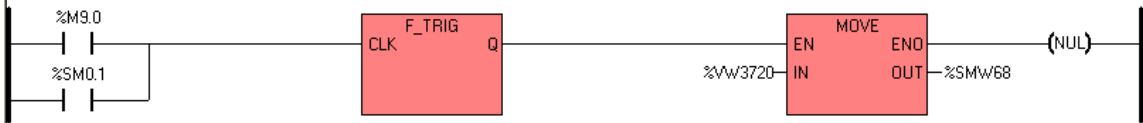
(* Network 0 *)
(* Inicializa salida de pulsos *)



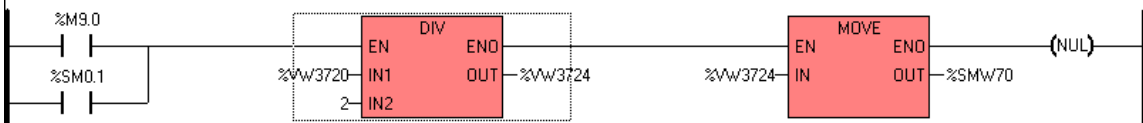
(* Network 1 *)
(* Calcula tiempo de ciclo en microseg <50 a 65535> *)



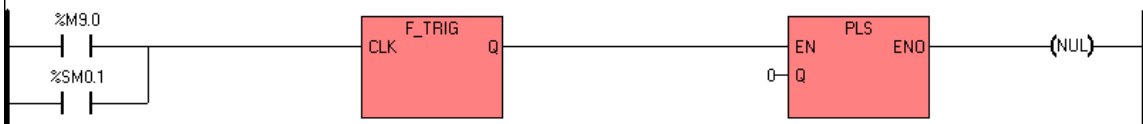
(* Network 2 *)
(* Carga Tiempo de Ciclo *)




(* Network 3 *)
(* Ancho de Pulso *)



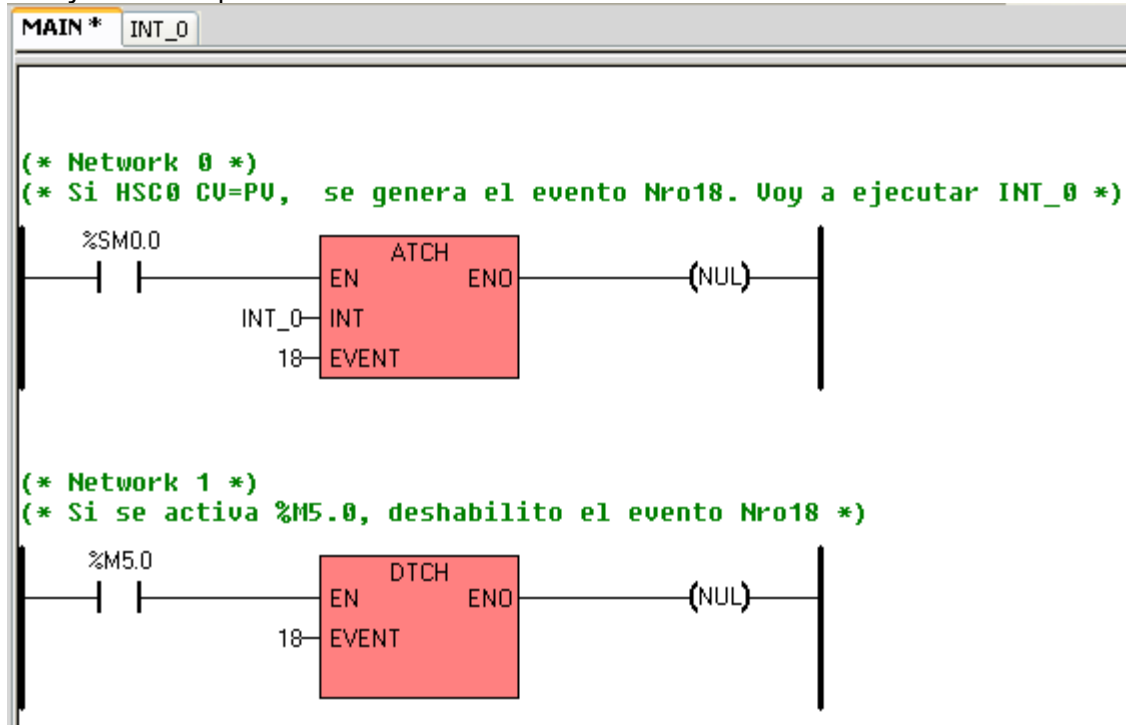
(* Network 4 *)
(* Arranca Salida de Pulsos *)



INTERRUPCIONES

Event No.	Description	Type	Priority
32	PORT 1: XMT complete	Communication Port Interrupts	Highest
31	PORT 1: RCV complete		
30	PORT 0: XMT complete		
29	PORT 0: RCV complete		
28	PTO 0 complete	I/O Interrupts	
27	PTO 1 complete		
26	I0.0, Falling edge		
25	I0.0, Rising edge		
24	I0.1, Falling edge		
23	I0.1, Rising edge		
22	I0.2, Falling edge		
21	I0.2, Rising edge		
20	I0.3, Falling edge		
19	I0.3, Rising edge		
18	HSC0 CV=PV		
17	HSC0 direction changed		
16	HSC0 external reset		
15	HSC1 CV=PV		
14	HSC1 direction changed		
13	HSC1 external reset		
12	HSC2 CV=PV		
11	HSC2 direction changed		
10	HSC2 external reset		
9	HSC3 CV=PV		
8	HSC4 CV=PV		
7	HSC4 direction changed		
6	HSC4 external reset		
5	HSC5 CV=PV		
4	Timed interrupt 1. Its period is specified in SMW24, unit: ms, range: 1~65535ms.	Time Interrupts	
3	Timed interrupt 0. Its period is specified in SMW22, unit: ms, range: 1~65535ms.		
2	Timer T3 ET=PT		
1	Timer T2 ET=PT		Lowest

Manejo de Interrupciones



COMUNICACION CON PROTOCOLO MODBUS RTU

Area	Range.	Type	Corresponding Modbus Registers
I	I0.0 --- I7.7	DI	0 --- 63 ⁽¹⁾
Q	Q0.0 --- Q7.7	DO	0 --- 63 ⁽¹⁾
M	M0.0 --- M31.7	DI/DO	64 -- 319 ⁽¹⁾
AI	AIW0 --- AIW30	AI	0 --- 15 ⁽¹⁾
AQ	AQW0 --- AQW30	AO	0 --- 15 ⁽¹⁾
V	VW0 --- VW4094	AI/AO	16 -- 2063 ⁽¹⁾

Al registro de Modbus se le debe sumar 1

0x Escritura/Lectura de Bit
 1x Lectura de Bit
 3x Lectura de Word
 4x Lectura/Escritura de Word

Ejemplos:

BITS

Lectura de %I0.0>>>>> **1x1**

Lectura de %I0.1>>>>> **1x2**

Escritura de %Q0.0>>>> **0x1**
Lectura de %Q0.1>>>> **0x2**

Lectura de %M0.0>>>> **1x65**
Lectura de %M0.1>>>> **1x66**

Escritura de %M0.0>>>> **0x65**
Escritura de %M0.1>>>> **0x66**

Ejemplos:
WORD

Lectura de %VW0>>>> **1x17**
Lectura de %VW2>>>> **1x18**

Escritura de %VW0>>>> **0x17**
Escritura de %VW2>>>> **0x18**

Lectura de %VW500>>>> **1x267**
Lectura de %VW600>>>> **1x307**

Escritura de %VW500>>>> **0x267**
Escritura de %VW600>>>> **0x307**