

FC1 - <offline>

"Gestion_IO"

Nombre: Familia:
Autor: Versión: 0.1
Hora y fecha Código: Versión del bloque: 2
 26/07/2019 17:02:35
Interface: 27/06/2019 07:44:27
Longitud (bloque / código / datos): 00356 00200 00002

Nombre	Tipo de datos	Dirección	Comentario
IN		0.0	
W_Dog	Int	0.0	
NumNodo	Int	2.0	
TX_Word	Word	4.0	
OUT		0.0	
RX_Word	Word	6.0	
Conect	Bool	8.0	
Error	Bool	8.1	
IN_OUT		0.0	
TEMP		0.0	
Iteraciones	Int	0.0	
RETURN		0.0	
RET_VAL		0.0	

Bloque: FC1

Segm.: 1 Prepara El DB Asociado a la aplicación

```
AUF DB 1 //Abre el DB asociado a nuestra aplicación
```

Segm.: 2 Verifica numero de Nodo comprendido entre 1-25

```

L #NumNodo // Cargamos número de Nodo #NumNodo
L 26
<I
SPB Met1 //Nodo menor que 26
SPA Err //Error número de nodo
Met1: L #NumNodo //Cargamos número de Nodo #NumNodo
L 0
>I
SPB Met2 //Nodo mayor que 0
SPA Err //Error número de nodo
Met2: SET //Nodo comprendido entre 1-25
R #Error //Resetea bit de ERROR #Error

```

Segm.: 3 Calcular la dirección inicial del nodo Actual

```

LAR1 P#0.0 //Inicializa posición Nodo 0
L #NumNodo // Cargamos tantas veces como numero de No #NumNodo
do
NEXT: T #Iteraciones // Lo transferimos a la variable auxiliar #Iteraciones
+AR1 P#10.0 //Incrementamos puntero 10 unidades por cada vuelta.
L #Iteraciones // Leemos el número de vueltas que llevamos #Iteraciones
s y lo cargamos en el ACU1
LOOP NEXT // saltamos a la meta NEXT si el resultado no es cero.
// El AR1 contiene la dirección del nodo Actual por ejemplo: Nodo5 AR1-> P#60.0

```

Segm.: 4 Control de un Flanco P

```
//Control de un Flanco, crea una única ejecución por cada flanco de subida
UN DBX [AR1,P#8.2] //Buffer_Entrada
SPB Time //Salta a Temporizar si no hay datos Buffer_Entrada

SET
R DBX [AR1,P#8.2] //Reset Flag del Buffer_Entrada
L 0
T DBB [AR1,P#1.0] //Reset valor Temporiza

//Carga la word a enviar en área de memoria de envío al NODO
L #TX_Word #TX_Word
T DBW [AR1,P#4.0] //Tx_Word
//Carga el área de memoria de recepción del NODO en la word de recepción.
L DBW [AR1,P#6.0] //Rx_Word
T #RX_Word #RX_Word

SPA Fin
```

Segm.: 5 Control del timer de exceso de tiempo sin recibir datos

```
Time: NOP 0
//Se ejecuta con el CLK del byte de marcas de ciclo de la CPU.
SET
U "CLK-0_1seg" //Marca Oscilador de la CPU M0.0
FP DBX [AR1,P#8.3] //Flanco
SPBN Tfin //Final temporizador

L #W_Dog // DBB [AR1,P#0.0] //SP_Tie #W_Dog
mpo
L DBB [AR1,P#1.0] //Temporiza
<=I
SPB TKO //Tiempo excedido vivo=0

SET
S DBX [AR1,P#8.1] //Vivo

// Incrementa una unidad el temporizador
L DBB [AR1,P#1.0] //Temporiza
L 1
+I
T DBB [AR1,P#1.0] //Temporiza
SPA Tfin //Final temporizador

TKO: NOP 0 //Time out vivo=0
SET
R DBX [AR1,P#8.1] //Vivo

Tfin: SPA Fin //Final temporizador
```

Segm.: 6 Salida Rutina

```
Err: NOP 0 //Salida con ERROR
SET
S #Error #Error
R DBX [AR1,P#8.1] //Vivo
Fin: NOP 0
U DBX [AR1,P#8.1] //Vivo
= #Conect //Señal NODO conectado #Conect
```