

MÓDEM-GSM

INDICE

- 1. INTRODUCCIÓN**
- 2. CONFIGURACIÓN PUERTO SERIE CPU**
- 3. FUNCIONAMIENTO DE LA FUNCIONES TXD Y RXD**
- 4. EJEMPLO DE ENVÍO DE SMS DESDE EL PLC**
- 5. EJEMPLO DE RECEPCIÓN DE SMS EN EL PLC**
- 6. UTILIZACIÓN DEL MENSAJE RECIBIDO PARA ACTIVAR UN BIT**

1. INTRODUCCIÓN

WES-GDW11 es un módem para aplicaciones industriales.

Este módem permitirá que el usuario pague solamente por los datos que transmite y no por el tiempo de conexión. WES-GDW11 permite establecer conexiones remotas a través de la red móvil GSM (Global Service Mobile) y aprovechar todos los servicios que ofrecen los distintos operadores: envío de mensajes SMS, conexiones telefónicas, conexión de datos a través de GPRS (General Packet Radio Service), conexiones a Internet, etc.

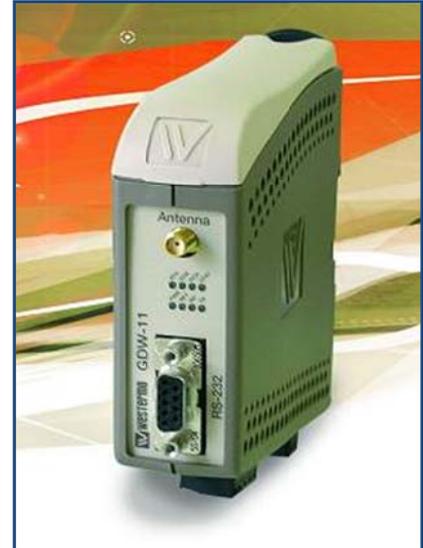
La conexión a dispositivos inteligentes (PLC's) es muy sencilla, es suficiente con conectar el modem al puerto serie RS232C del autómatas programable. La comunicación se realiza fácilmente mediante comandos AT o comandos HAYES y resulta transparente y de gran fiabilidad.

El WES-GDW11 es un modem GSM dual capaz de operar en las dos bandas de frecuencia europeas 900/1800 MHz. Esta capacidad aumenta su robustez y fiabilidad ya que puede conmutar, de forma automática, entre las dos bandas en el caso de dificultades en las comunicaciones.

Incorpora Stack TCP/IP para establecer comunicaciones GPRS (Clase B, Clase 10). Estas categorías ponen de manifiesto, por un lado, que el modem se puede registrar y activar simultáneamente en GSM y GPRS sin soportar la simultaneidad de servicio, y por el otro, que ofrece una capacidad multislot de 4/2 slots para descarga y envío de datos. Según esto, la conexión GPRS es mucho más potente y fiable, pudiendo alcanzar velocidades de 85,6 Kbit/s.

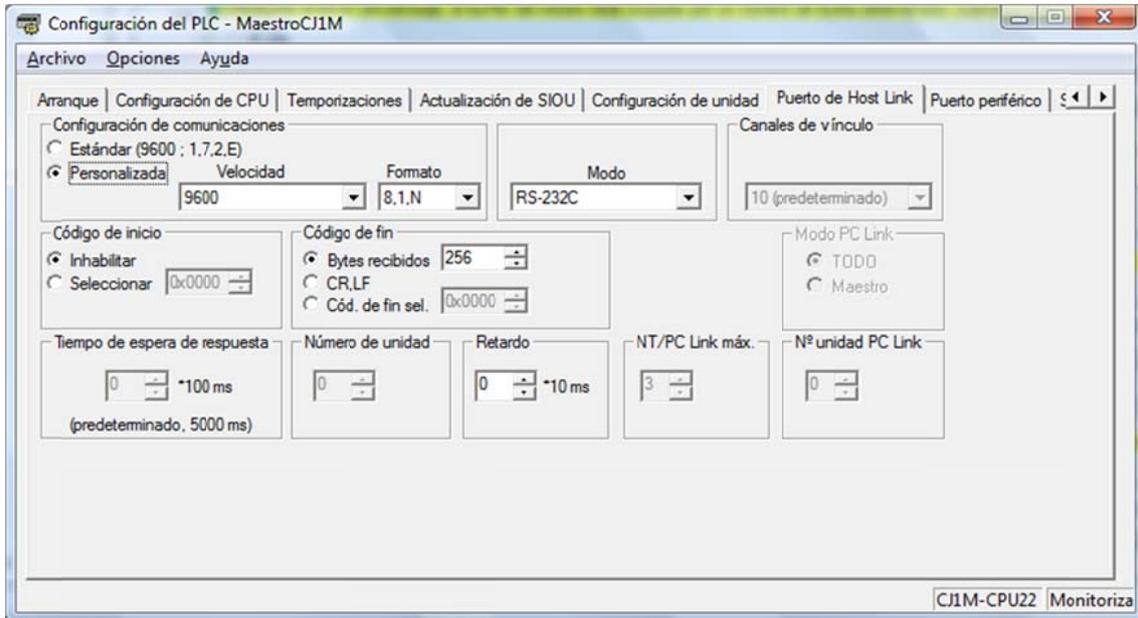
Por otro lado, dado que el servicio GPRS permite la transmisión de información a un coste asociado a la cantidad de datos y no al tiempo de conexión, el usuario puede mantener las comunicaciones abiertas durante 24 h, ahorrando tiempo al no tener que activar y desactivar la comunicación cada vez que necesite hacer un envío de información.

WES-GDW11 es un módem muy versátil para el desarrollo de aplicaciones. Ofrece una inmejorable respuesta operando junto a autómatas programables Omron, incrementando la eficacia de las comunicaciones. Sin embargo, también se pueden generar llamadas y/o envío de mensajes SMS mediante la señal digital DTR (sin necesidad de conexión a un dispositivo inteligente) por ejemplo, mediante el uso de sensores



2. CONFIGURACIÓN PUERTO SERIE CPU

Dado que el enlace entre el PLC y el módem se realiza a través del puerto serie de la CPU, es necesario realizar en dicho puerto la configuración de la siguiente figura:



Configuración Puerto de Host Link, para conectar con modem GDW-11 :

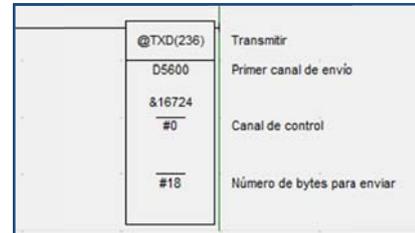
En código de fin, con la opción seleccionada se escribirá <CR> y <LF> al final de los códigos (ver apartado 3)

Seleccionando la opción de “código de fin” a CR,LF no será necesario escribir dicho código al final de los mensajes (No se ha configurado así ya que en el ejemplo utilizado, al recibir SMS, necesita de esta configuración).

3. FUNCIONAMIENTO DE LA FUNCIONES TXD Y RXD

TRANSMITIR TXD

Esta instrucción lee en número de bytes en bruto indicado en el tercer dato (#18), comenzando de la zona de memoria indicada en el primer dato (D5600) y los envía por el puerto especificado en el canal de control (#0 corresponde al puerto RS232C de la CPU).



El contenido del *Nº* Bytes enviados (tercer dato) que están guardados en DM5600 (primer dato), consta de tres partes:

Código de envío (ver manual del módem)

Texto del envío (texto, número de teléfono, etc.)

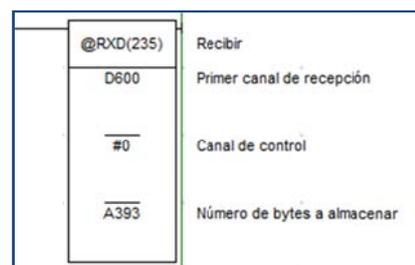
Códigos de final de envío.

Para enviar un mensaje de texto se necesita utilizar dos instrucciones TXD, la primera realiza la llamada al número indicado a partir de D5600 y la segunda envía el texto y el código de final de envío a partir del número de DM indicado en la instrucción.

Los datos sólo pueden ser enviados cuando el indicador de puerto listo, A392.05 está a ON. Terminado el envío, se pone a ON el bit A392.06

RECIBIR RXD

RXD lee los datos que han sido recibidos en modo sin protocolo en el puerto RS-232C integrado en la CPU y almacena el número de bytes recibidos, a partir del DM especificado en el primer canal de recepción. En número de bytes a almacenar se especifica en el tercer dato de la instrucción, se suele utilizar el A393 que pertenece al sistema e indica los bytes recibidos.

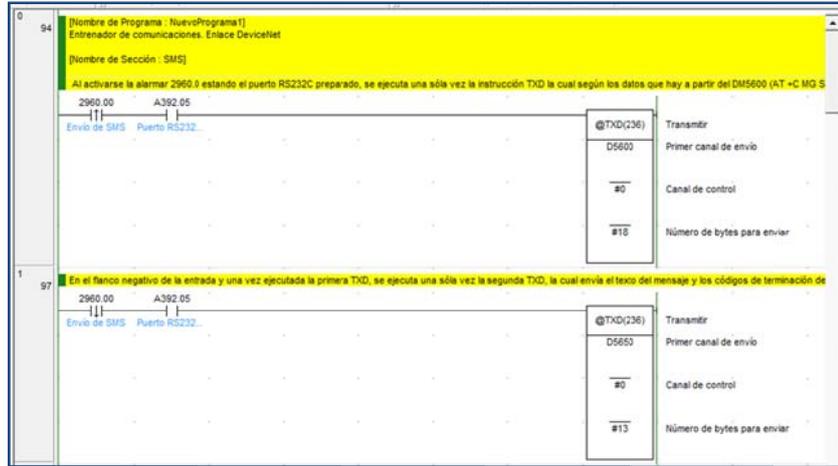


Los datos solamente pueden recibirse cuando el indicador de listo para recepción del puerto está en ON. El indicador de listo para recepción es A392.06

Trabajando con el módem GDW-11, ambas instrucciones pueden realizar multitud de funciones en base a los códigos escritos en el primer dato, para ampliar información sobre dichos códigos, consultar el manual del módem.

4. EJEMPLO DE ENVÍO DE SMS DESDE EL PLC

Para enviar el mensaje deberemos de utilizar dos instrucciones TXD, una para indicar el nº de teléfono al que se envía el mensaje, y la otra con el texto del mensaje. Con el fin de no utilizar más de una señal de



“alarma”, al relé que activa dicha alarma y dado que en un mismo scan no se pueden ejecutar dos instrucciones TXD, le configuraremos el flanco de subida para ejecutar una TXD y el flanco de bajada para la siguiente (ver imagen).

Explicación de la 1ª instrucción TXD.

Se utiliza para indicar a que nº de móvil vamos a enviar el mensaje.

D05600	4154	2B43	4D47	533D	222B	3334	36				
D05610	3222	0D0A	0000	0000	0000	0000	0000	0000	0000	0000	0000
D05620	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

A partir del DM5600 se escribe: **AT +C MG S= “+ 34 número teléfono” 0D0A** que en hexadecimal corresponde a los códigos de la figura (ver tabla de códigos Ascii).

- Código de envío de mensaje (ver manual del módem).
- Entre comillas, número de teléfono al que se envía el mensaje (incluir +34 para teléfonos de España).
- Final de envío (Retorno de carro 0D, final de línea 0A).

La información que contiene es la encargada de realizar la llamada.

El dato de Control es un #0 ya que se está utilizando el puerto serie de la CPU (para otros puertos consultar el manual).

El nº de bytes a enviar en este ejemplo es #18 en hexadecimal que corresponden a 24 en decimal. Son los Bytes utilizados para escribir el primer dato.

Explicación de la 2ª instrucción TXD.

Esta contiene el texto del mensaje.

D05640	
D05650	VA LV UL A 5 RO TA
D05660	

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
D05650	5641	4C56	554C	4120	3520	524F	5441	0000	0D0A	1A00

A partir del DM5650 se escribe el texto del mensaje a enviar. Se termina con retorno de carro y final de línea (como la TXD anterior) y además se incluye el código de control <ctrl+z> (1A en hexadecimal)

En la ventana del nº de bytes para enviar, el valor que coloquemos (en hex.) indicará la longitud máxima del texto, incluyendo los comandos finales.

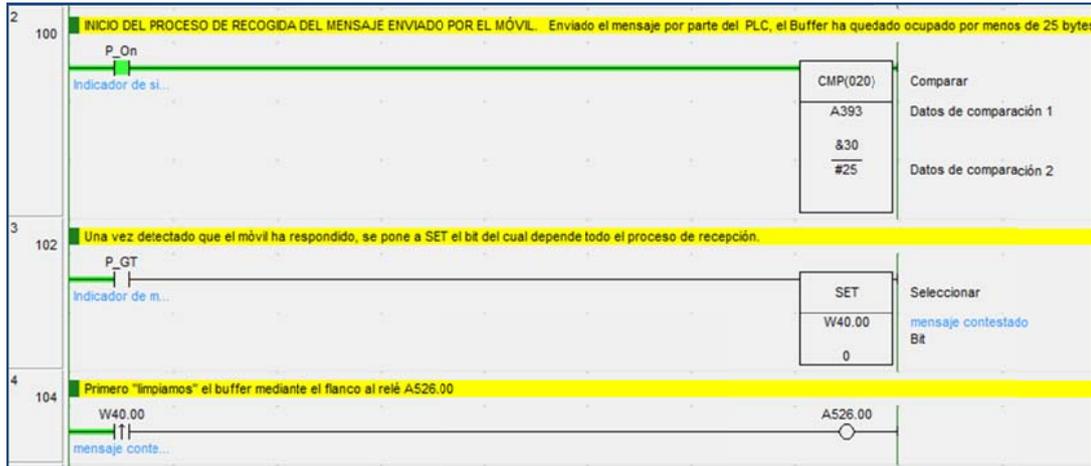
El nº de bytes máximo a enviar en este ejemplo es #13 (Hex.) que son 19 bytes (16 el texto + <CR><LF> <Ctrl+z>).

TABLA DE EQUIVALENCIAS DEL CÓDIGO ASCII a HEXADECIMAL

ASCII	Hex	Símbolo	ASCII	Hex	Símbolo	ASCII	Hex	Símbolo	ASCII	Hex	Símbolo
0	0	NUL	16	10	DLE	32	20	(espacio)	48	30	0
1	1	SOH	17	11	DC1	33	21	!	49	31	1
2	2	STX	18	12	DC2	34	22	"	50	32	2
3	3	ETX	19	13	DC3	35	23	#	51	33	3
4	4	EOT	20	14	DC4	36	24	\$	52	34	4
5	5	ENQ	21	15	NAK	37	25	%	53	35	5
6	6	ACK	22	16	SYN	38	26	&	54	36	6
7	7	BEL	23	17	ETB	39	27	'	55	37	7
8	8	BS	24	18	CAN	40	28	(56	38	8
9	9	TAB	25	19	EM	41	29)	57	39	9
10	A	LF	26	1A	SUB	42	2A	*	58	3A	:
11	B	VT	27	1B	ESC	43	2B	+	59	3B	;
12	C	FF	28	1C	FS	44	2C	,	60	3C	<
13	D	CR	29	1D	GS	45	2D	-	61	3D	=
14	E	SO	30	1E	RS	46	2E	.	62	3E	>
15	F	SI	31	1F	US	47	2F	/	63	3F	?
ASCII	Hex	Símbolo	ASCII	Hex	Símbolo	ASCII	Hex	Símbolo	ASCII	Hex	Símbolo
64	40	@	80	50	P	96	60	`	112	70	p
65	41	A	81	51	Q	97	61	a	113	71	q
66	42	B	82	52	R	98	62	b	114	72	r
67	43	C	83	53	S	99	63	c	115	73	s
68	44	D	84	54	T	100	64	d	116	74	t
69	45	E	85	55	U	101	65	e	117	75	u
70	46	F	86	56	V	102	66	f	118	76	v
71	47	G	87	57	W	103	67	g	119	77	w
72	48	H	88	58	X	104	68	h	120	78	x
73	49	I	89	59	Y	105	69	i	121	79	y
74	4A	J	90	5A	Z	106	6A	j	122	7A	z
75	4B	K	91	5B	[107	6B	k	123	7B	{
76	4C	L	92	5C	\	108	6C	l	124	7C	
77	4D	M	93	5D]	109	6D	m	125	7D	}
78	4E	N	94	5E	^	110	6E	n	126	7E	~
79	4F	O	95	5F	_	111	6F	o	127	7F	

5. EJEMPLO DE RECEPCIÓN DE SMS EN EL PLC

Terminado el envío del SMS el canal indicador de bytes recibidos A393 guarda el valor que corresponde al mensaje mandado por el módem indicando que el envío ha sido correcto, dicho valor es inferior a 25 (en hexadecimal). Cuando desde el móvil se responde con un mensaje de texto y éste es recibido por el módem, ese valor aumenta y pasa a ser mayor que 25 (el valor final depende del mensaje enviado), con lo que se inicia el proceso de recepción del mensaje por el PLC.



Cuando el resultado de la comparación es superior a 25 (se ha contestado al mensaje), se activa el relé W40.00.

Para lograr que el texto del mensaje se posicione siempre en los mismos DMs, es necesario que la "cabecera" del mensaje sea siempre igual, para ello se necesita eliminar toda la información residual que esté en el búffer del módem. Lo primero que se hace es poner el contador de bytes recibidos a cero, reiniciando para ello el puerto RS232C de la CPU, lo cual se consigue mandando un impulso al bit A526.00 que el sistema tiene configurado para esa función.

Reiniciado el puerto RS232C de la CPU comienza el proceso de recepción del mensaje. Hay que ejecutar una primera RXD (Siempre con la instrucción diferencial @ para que sólo se ejecute una vez) en la que se solicita el número de mensaje a recepcionar. El número de mensaje hace referencia a los mensajes almacenados en el módem. Siempre será el número uno ya que como se verá, al final del proceso se borran todos los mensajes almacenados en el módem.



La instrucción RXD funciona básicamente como la TXD, hay que configurar tres datos: en el primero se escriben los códigos que utiliza el módem según el comando a utilizar. El segundo hace referencia al puerto a utilizar y el tercero al número de Bytes a recibir.

En el primer dato, a partir del DM300 el módem escribirá en hexadecimal una serie de códigos indicando que todo se ha recibido correctamente.

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
D00300	..	OK	>	..	>	..	+C	M
D00310	S:	1	59	OK	+C	MT
D00320	t:	"	S,	08	:3	1:	03	+0	8"	..
D00330	do	s.	..	.O	K

En el segundo dato se indica el puerto de la CPU por el cual se recibirá el mensaje.

En el tercer dato se pone el A393 que indica el número de bytes recibidos. No se pone un dato fijo ya que el número de bytes a recibir depende de la información que se ha solicitado.

A la vez que se ejecuta la RXD, se pone a funcionar un temporizador, el cual dará la siguiente orden de recepción.

La RXD anterior ha respondido que el mensaje a recibir es el número 1. Hay que ejecutar un TXD en la que se solicita la recepción del mensaje número 1 que está almacenado en el módem. Funciona igual que las TXD anteriores, lo único que



cambian son los códigos que se escriben a

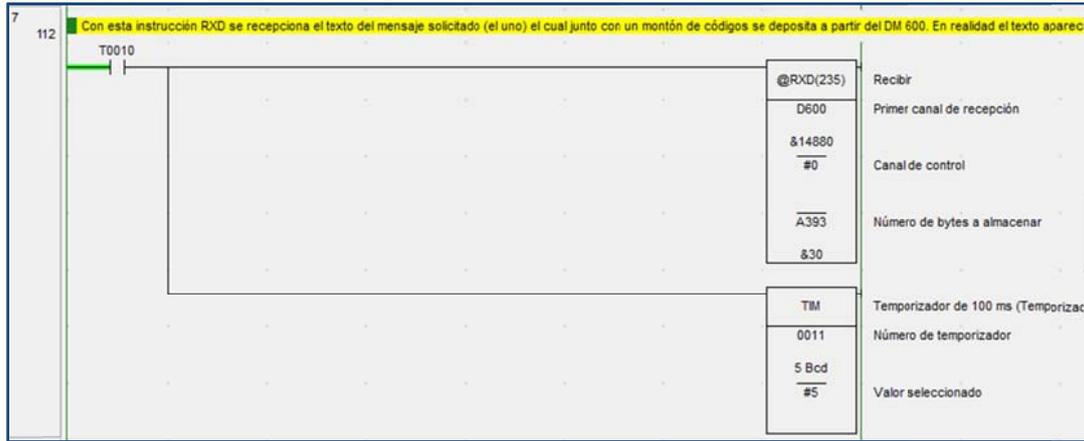
D00450	AT	+C	MG	R=	1.
--------	----	----	----	----	----	----	----	----	----	----

partir del D450. En hexadecimal se escribirá como en la figura (se incluyen los códigos de <CR> y <LR>).

D00450	4154	2B43	4D47	523D	310D	0A00	0000	0
--------	------	------	------	------	------	------	------	---

Un nuevo temporizador es puesto en marcha para ejecutar la siguiente RXD.

La nueva RXD recoge el mensaje número 1 que se solicitó anteriormente, el mensaje junto con una cantidad importante de códigos se deposita a partir del D600. Como inicialmente se reinició el puerto RS232C mediante el bit A526.00, el verdadero texto del mensaje se depositará siempre a partir del D637, que será el que utilizaremos para posterior uso del programa para conectar/desconectar un bit.

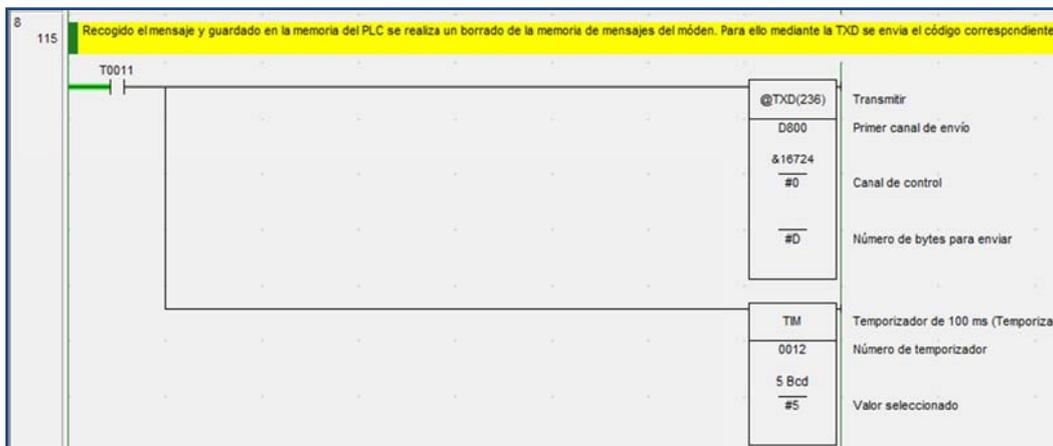


Es importante tener en cuenta que la longitud del mensaje de texto tiene que ser siempre la misma para poder realizar las posteriores comparaciones. Por ejemplo caso de existir dos posibles respuestas (marcha) y (paro), en ambos casos el texto constará de 6 bytes, como "paro" sólo tiene cuatro, se añadirán dos espacios (para borrar las dos últimas letras del mensaje anterior (si este había sido marcha).

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
D00550
D00560
D00570
D00580
D00590
D00600	AT	+C	M	R=	1.	+C	M	R:
D00610	"	RE	C	UN	RE	AD	",	"	+	34 63
D00620	02	89	58	2"	..	"0	9/	05	/1	1,
D00630	11	:1	7:	35	+0	8"	..	ma	rc	ha
D00640	OK
D00650

Los dos últimos datos de esta instrucción RXD indican los mismo que la primera.

Un nuevo temporizador se encargará de ejecutar la orden de borrar el mensaje del módem.



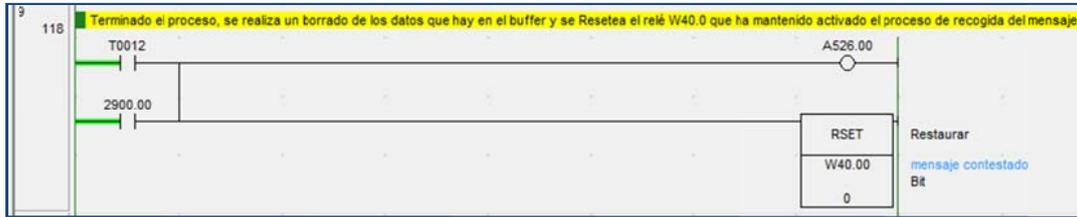
Para realizar el borrado, se utiliza una nueva

instrucción TXD con la cual enviamos a partir del DM 800, el comando de borrar mensaje como siempre en hexadecimal y terminando con los comandos <CR> <LF>

D00800	AT	+C	M	D=	1,	4.
--------	----	----	---	----	----	----	----	----	----	----

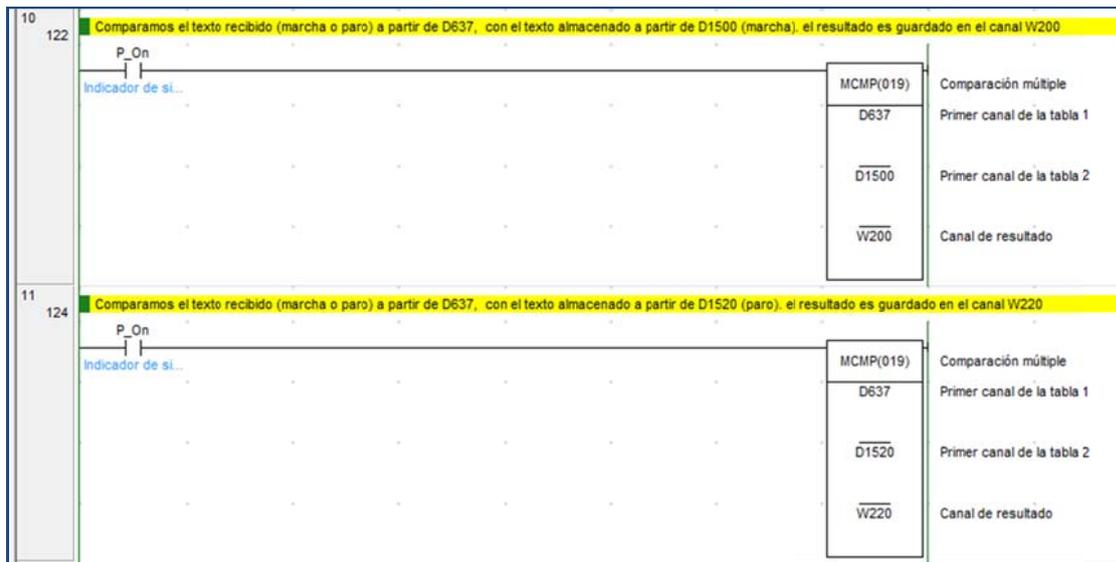
D00800	4154	2B43	4D47	443D	312C	340D	0A00
--------	------	------	------	------	------	------	------

Un nuevo temporizador reiniciará el puerto y desconectará el relé W40.0 que ha permanecido activado durante todo el proceso de recepción del mensaje.



6. UTILIZACIÓN DEL MENSAJE RECIBIDO PARA ACTIVAR UN BIT

Supongamos que el mensaje recibido se utiliza para conectar/desconectar un bit dependiendo de que este sea “marcha” o “paro__”. Los siguientes segmentos del programa se utilizan para realizar las comparaciones necesarias para ello.

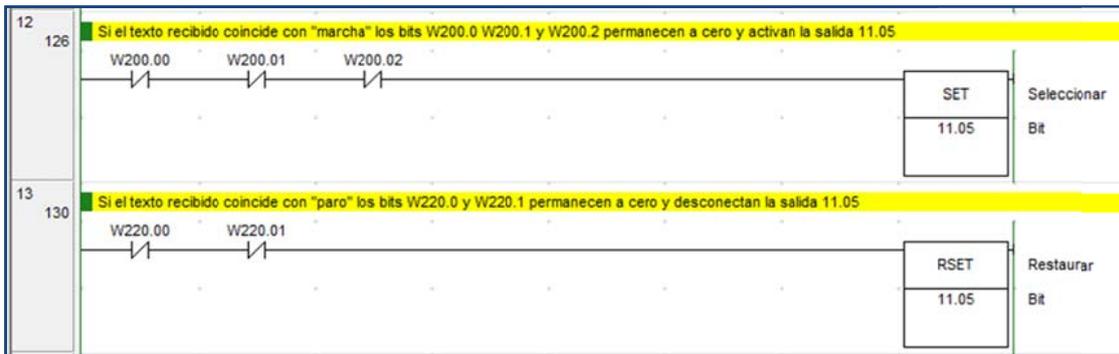


Mediante la instrucción MCMP se realiza una comparación múltiple de 16 canales, los datos almacenados en los canales D637 a D652 son comparados canal a canal con los fijos que se han escrito en los canales D1500 a D1515, si la comparación de cada canal es igual, el bit correspondiente del canal de resultado se pone a cero.

SI	D637	=	D1500	ENTONCES	W200.00	=	0
SI	D638	=	D1501	ENTONCES	W200.01	=	0
SI	D639	=	D1502	ENTONCES	W200.02	=	0
SI	D640	≠	D1503	ENTONCES	W200.03	=	1
SI	D641	≠	D1504	ENTONCES	W200.04	=	1
..
SI	D652	=	D1515	ENTONCES	W200.15	=	0

Los resultados obtenidos al ejecutar las dos instrucciones MCMP son utilizados para conectar/desconectar un bit (11.05). La instrucción SET 11.05 sólo se ejecutará cuando el mensaje recibido sea "marcha" ya que los datos escritos a partir del D637 serán iguales que los que tenemos a partir del D1500. Igualmente cuando se reciba el mensaje "paro", como coincide con los datos que tenemos a partir del D1520, se ejecutará la instrucción RSET11.05.

Caso de no coincidir el mensaje recibido con ninguno de los dos que tenemos almacenados, no se ejecuta ninguna de las dos instrucciones.



En la imagen, datos guardados en los DMs

