El estado actual del contador CW es 192 y considerando el signo “+” el valor total será: 192+65536= 65728

Cuando el encoder fue forzado a cero (ZE) la RIAC agrega un espaciador “ “ al valor de la cuenta CW y CCW, el espaciador aparecerá en la primer lectura (RE) y no volverá a aparecer en las lecturas sucesivas hasta tanto no ocurra una nueva puesta a cero.

ZE	Integer = ZeroEncoder (0)	público/privado

El comando ZE fuerza a cero los contadores CW y CCW

```
# r ZE 0 <CR>                    ´* Fuerza a cero CW y CCW
r, 0 <CR>

# 1 ZE 0 <CR>
1, 0

n = qx.ZeroEncoder (0)
```

CE	Integer = CloseEncoder (0)	público/privado

Este comando da cierre al encoder.

```
# r CE 0 <CR>                    ´*Cierre encoder
r, 0

# i CE 0 <CR>                    ´*Ejemplo cierre encoder.
r, 0

n = qx. CloseEncoder (0)
```


CODIFICADOR INCREMENTAL

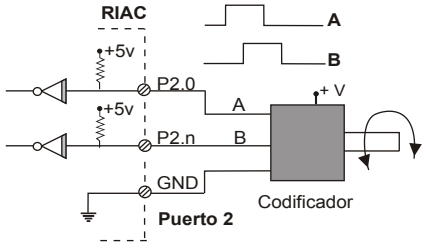
La presente nota describe las características y forma de uso que una placa RIAC ofrece para operar con un codificador incremental.

Características generales.

- Conexión de codificador incremental.
- Contadores de 16 bits independientes para CW y CCW.
- Amplio rango de niveles: TTL, CMOS, OC, lógica de 24V.
- Señales en cuadratura (código Gray).
- Máxima velocidad: 600 rpm.
- Reposición a cero de CW y CCW.
- Transporte aritmético para precisión múltiple o indicador de desborde.

Todos los módulos RIAC QFn y QMn permiten conectar un codificador incremental, (shaft enconder). Admiten la conexión de las señales digitales en cuadratura denominados canal A y canal B. Las conexiones se realizan sobre puerto el Puerto 2 y se indican en las siguientes figuras:

	RIAC QFn	RIAC QMn
Canal A	P2.0	P2.0
Canal B	P2.2	P2.1



Las placas RIAC operan con cualquier tipo de codificador que disponga de salidas a colector abierto (OC), o bien que trabaje con niveles TTL, CMOS o lógica de 24V. Estas alternativas están disponibles sin necesidad de ajustes o configuraciones previas.

La unidad RIAC dispone internamente de dos contadores, cada uno de 16 bits, denominados CW y CCW. El contador CW incrementa la cuenta por cada pulso en sentido horario proveniente del encoder. CCW incrementa la cuenta por cada pulso en sentido antihorario. La diferencia corresponde a la cantidad de pulsos netos.

PNETOS = CW-CCW

El beneficio de ofrecer CW y CCW reside en conocer la cantidad de pulsos por cada sentido de giro. Ejemplo: en mediciones de distancias, un contador determina el avance, el otro el retroceso y la diferencia es la distancia neta desde el inicio.
La cuenta tanto en CW como CCW progresa con la transición de alto a bajo proveniente del canal B

Comandos para el codificador

Los comandos disponibles para el codificador son:

- **OE, OpenEncoder.** El comando OE debe encabezar la secuencia, este pone a cero los contadores, CW y CCW, y se prepara para admitir y procesar los comandos ZE y RE.

- **RE, ReadEncoder.** El comando RE devuelve el estado interno de los contadores CW y CCW. Los contadores CW y CCW son de 16 bits, por lo tanto la cuenta estará comprendida entre 0 y 65535.

- **ZE, ZeroEncoder.** El comando ZE fuerza a cero ambos contadores.

- **CE, CloseEncoder.** CE da cierre a la operatoria del enconder.

Un programa simple consiste en una apertura del encoder, OE. Acontinuación se procede a realizar mediante RE todas las lecturas necesarias. El cierre del encoder del enconder, -CE- no resulta obligatorio, salvo que por motivos específicos se haga uso de CNT4 o bien de los terminales de P2.n para otra aplicación.

El Encoder comparte los terminales del Puerto 2 con otras aplicaciones de las RIAC, estas no pueden operar en forma simultanea. Ejemplo: si se opera el encoder no sera posible abrir el Contador 4. Siempre serán aceptados los comandos BitInput y ReadImput, así como BitSet y BitReset estos últimos en los modelos QF sobre P2.1 y P2.3 únicamente.

```
'*************************************************************************
*                  VBASIC. Ejemplo de uso del encoder                    *
'*************************************************************************
n = qx.OpenEncoder(0)    '* Abre contador

'* Lectura y tratamiento del encoder. Ejemplo dentro de un lazo .
Do
    .........
    d = qx.ReadEncoder(0, CountUP, CountDW)     '* Lectura del encoder

    c = Val(CountUP)      '* convierte CountUP a un valor numerico
    If Left$(CountUP, 1) = "+" Then
       '* Verifica si hay desborde (+) y procede
       n = (PulsosAvance \ 65536) + 1: PulsosAvance = n * 65536 + c
    ElseIf Left$(CountUP, 1) = " " Then
       '* Verifica si hay forzado a cero externo.
       PulsosAvance = 0:  c = 0
    Else
       PulsosAvance = PulsosAvance + Abs(c - PAanterior)  '* Acumula simple
    End If
    PAanterior = c    '* Conserva valor anterior

    c = Val(CountDW)      '* Sigue con el contador CountCW
    If Left$(CountDW, 1) = "+" Then
       n = (PulsosRetro \ 65536) + 1:  PulsosRetro = n * 65536 + c
    ElseIf Left$(CountDW, 1) = " " Then
       PulsosRetro = 0:  c = 0
    Else
       PulsosRetro = PulsosRetro + Abs(c - PRanterior)
    End If
    PRanterior = c    '* Conserva valor anterior

    PulsosNetos = PulsosAvance - PulsosRetro
    ...........
Loop

'* Cerrar el encoder en caso que desee salir de la aplicacion.
qx.CloseEncoder (0)     '* Cierra el Encoder
```


COMANDOS PARA EL ENCODER

Los comandos aparecen descriptos en modo nativo y en modo Qx. Se indica además si son públicos o privados

OE	Integer = OpenEncoder (0)	público/privado

El comando produce la apertura del encoder. Trás la emisión de OE, se aceptarán los comandos RE y ZE (Read Encoder yZeroEncoder).

El comando OE inicialmente pone en estado alto de todos los bornes de Puerto 2. A continuación fuerza a cero el estado de los contadores CW y CCW.

```
# 1 OE 0 <CR>                 ´* Apertura del encoder
1,0 <CR>                      ´* Okey

n = qx.OpenCounter (0)        ´* Apertura del encoder
```

RE	String = ReadEncoder (0)	privado

ReadEncoder lee el estado de los contadores internos CW y CCW. La respuesta contendrá el valor de los contadores en el momento de la recepción del comando. El comando es reconocido solo si previamente se ha emitido como encabezamiento el comando OpenEncoder. Tras CloseEncoder, el comado ReadEncoder no será reconocido.

```
# r RE 0 <CR>                 ´* Lectura del encoder
r, CW, CCW.<CR>               ´* Contadores CW y CCW

# 5 RE 0 <CR>
5, 387, 301 <CR>              ´* PNETOS= 387-301=86
```

El comando ReadEncoder retorna el estado de los contadores CW y CCW sobre las variables alfanuméricos CW y CCW. Los nombres utilizados son de índole general, el usuario podrá optar por otros nombres.

S = qx. ReadEncoder (0, CW, CCW)

Los contadores CW y CCW son de 16 bits y la cuenta de cada uno progresa de 0 a 65535. Cuando la cuenta supera el valor máximo (65535)el contador vuelca a cero y sigue la cuenta ascendente agregandose un signo “+” como un indicador de arrastre / desborde. El signo “+” se elimina en forma automática tras realizar la primer lectura (RE). Ejemplo:

```
#5 RE 0 <CR>                  ´* Lectura encoder
5, +192, 30481<CR>            ´* CW = +192, CCW = 30481
```