

**INTERVALO DE MUESTREO, VELOCIDAD DE ADQUISICIÓN**  
**Adquisición con RIAC-Q utilizando la función RealTime**

Se describen cuestiones asociadas a la velocidad de adquisición y la determinación del intervalo de muestreo de señales analógicas y digitales. Un programa demostrativo en VBasic se dispone bajo el nombre RealTime\_Demo. La presente revisión corrige la rev.040608, que tiene un error en la expresión del tiempo (t).

**Introducción**

Una de las necesidades que se presenta durante la adquisición es que los datos se tomen a intervalos previamente establecidos y equidistantes. Por ejemplo pueden tomarse cada 15 minutos muestras de la temperatura de un recinto, ó bien tomar muestras para reconstruir y analizar una señal cuyo periodo se estima en 30Hz.

En el primer caso es posible aceptar cierta elasticidad en la determinación del periodo de muestreo, ello presumiendo que las variaciones de temperatura son lentas. En el segundo caso es necesario operar con márgenes de error menores al 1% a efectos de lograr una

reconstrucción de la señal con un mínimo de deformación.

**Adquisición mediante una PC.**

La determinación del tiempo de muestreo resulta simple cuando el programa de adquisición toma control absoluto del computador; tal hecho ocurre cuando se opera bajo DOS. La situación cambia cuando se opera con sistemas multi-tarea, como Windows NT, Windows XP, W95 y W98. Bajo estas plataformas el programa de adquisición debe compartir la máquina con el sistema operativo y con otro número de aplicaciones. Como resultado práctico al programa de adquisición le es difícil establecer un periodo regular de muestreo. El punto

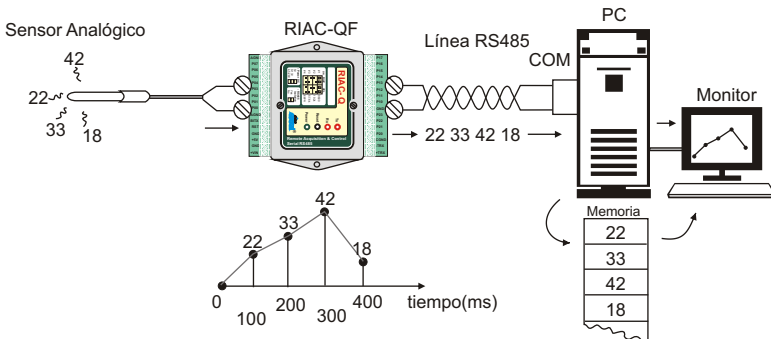


Figura 1: Placa de adquisición RIAC conectada al COM

próximo describe cómo resolver este problema haciendo uso de la placa RIAC-Q.

### Utilizando RIAC-Q

La figura 1 muestra la conexión de un módulo RIAC-QF, al puerto de comunicaciones serie de una PC.

Los módulos RIAC-Q disponen de la función denominada **RealTime**. Cuando esta función se activa, el módulo automáticamente comienza la adquisición de datos a intervalos regulares, como puede apreciarse en el gráfico de la figura 1, intervalos que el mismo módulo se encarga de administrar. Los datos luego son despachados con igual regularidad por la línea de comunicaciones hacia la PC. El proceso continúa en forma indefinida ó bien hasta que la función **RealTime** se desactive. Obsérvese que ahora es el módulo RIAC quien inicia y controla la adquisición.

¿Porqué este procedimiento garantiza la regularidad de las muestras? El motivo principal es que el puerto dispone de un **buffer de comunicaciones**, éste es totalmente autónomo de la actividad del procesador de la PC. En la figura 2 se muestra un gráfico más detallado que destaca la acción temporal.

En ① se da cuenta que los datos son convertidos de serie a paralelo e introducidos en el buffer de comunicaciones; esta acción es realizada por los circuitos de comunicaciones y es independiente del procesador de la PC.

En tanto el tiempo del procesador de la PC es dividido en lotes de tareas distribuido en el tiempo, las tareas son T1, hasta Tn, asignación realizada por el sistema operativo (ejemplo idealizado).

La tarea T5 toma los datos del buffer de comunicaciones y los transfiere al buffer de recepción a efectos que pueda disponerlos el programa de aplicación (pa). Este último tomará los datos en su “ranura de tiempo” y los procesará acorde a sus necesidades.

En otros términos, si a la llegada de un caracter, la PC está realizando alguna tarea prioritaria, los datos se acumulan en el buffer de comunicaciones y cuando la tarea en curso finaliza, los datos son trasladados del buffer de comunicaciones al de recepción (RAM). La presencia de los datos en el buffer de recepción indica la secuencia temporal ya que estos fueron enviados por la RIAC-Q a intervalos regulares. El programa de aplicación podrá tomarlos del buffer y proceder a graficarlos.

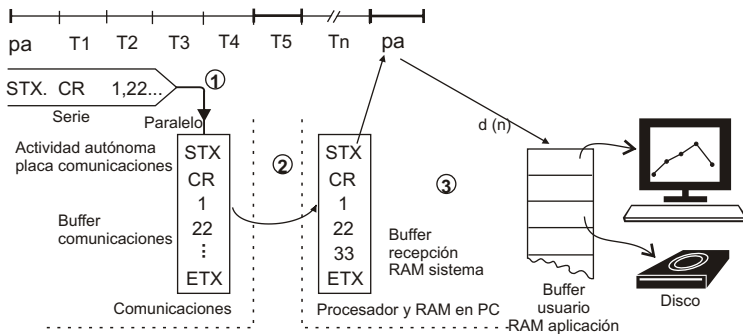
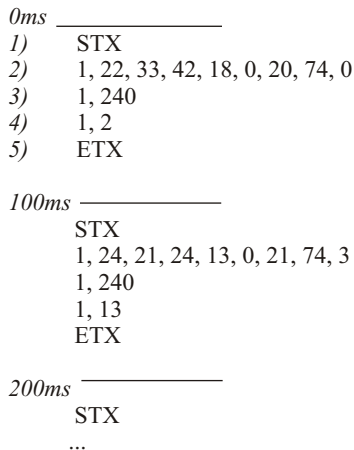


Figura 2: Comunicaciones procesadas y RAM en PC

Nota. El buffer de comunicaciones ha de ser habilitado por software. El autor observó que MSCOMM (5.0) no realizó esta tarea y el buffer queda así limitado a 2 bytes. Ensayó también realizar la habilitación directa sobre la UART, a tal efecto se utilizó las sentencias OUT, esta práctica fue exitosa y el buffer de comunicaciones operó en toda su amplitud (Win98).

### RIAC-Q. Trama de datos

En la figura 3 se observan bloques transmitidos en intervalos de 100ms, intervalo garantizado por el módulo RIAC-Q. Cada bloque inicia con un carácter de encabezamiento (STX) y termina con un cierre (ETX). Estos caracteres de control permiten que el programa receptor en la PC logre sincronismo de bloque. Entre cabeza y cola se hallan los datos propiamente dichos: la segunda línea corresponde a los datos de cada uno de los canales analógicos. La tercera línea corresponde al estado del Puerto1. Finalmente la cuarta línea devuelve el estado del Puerto2. El intervalo de muestreo en el presente trabajo,



**Figura 3:** Bloques de datos enviados por la RIAC. Por simplicidad se omitió <CR> como fin de línea.

hace referencia al tiempo para tomar las muestras de todos los datos de la RIAC.

### Desmenuzando los datos.

La trama debe ser “desmenuzada” desde el programa de aplicación para obtener los datos útiles. Una alternativa es extraer los datos con las sentencias MID\$(Vbasic) teniendo en cuenta que los datos se hallan separados por comas.

Además de este procedimiento, los módulos RIAC cuentan con el control activeX denominado RiacQX que hace todavía más simple la toma de datos. El programa que sigue para VBASIC muestra la sencillez y efectividad con que se realiza la adquisición en tiempo real.

```

1 n = RealTime (10,1)
.....
2 Sub Qx_RealTime ( )
3   For c = 0 to n
4     l(c) = qx.AnalogicChannel ( )
5     next = c
6     i1 = qx.InputPort (port1)
7     i2 = qx.InputPort (port2)
8   End Sub
.....
9 Sub qx_OnError
10   Lbl.caption = Qx.errnumber.
11 End Sub
.....
12 Sub qx_OverRealTime
13   LblExtempore = n Error
14 End Sub

```

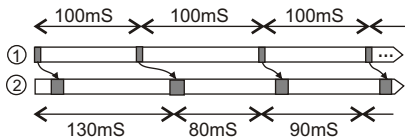
Para realizar un programa deben mantenerse estas consideraciones:

- a) Dar inicio a la adquisición en tiempo real. Consiste en emitir el comando RealTime (n,m), donde n y m determinan el periodo de muestreo según la expresión que sigue:

$$T = n \times m \times 10 \text{ [mS]}$$

La línea 1 constituye un ejemplo en donde el intervalo es de 100mSeg.

b) Se redactará el programa de adquisición propiamente dicho. En el ejemplo corresponde a las líneas 2 a 8. La subrutina QX\_RealTime se ejecutará automáticamente en intervalo aproximado a T, sin embargo los datos obtenidos serán equidistantes en exactamente T, como se aprecia en la fig. 4.



**Fig. 4.** (1) Envío regular de la RIAC-Q  
(2) Momento en que los toma el programa de usuario.

El envío de la RIAC esta garantizado cada 100mS (en este ejemplo). En tanto la subrutina qx\_RealTime los tomará dentro de un intervalo que en momentos se alarga (130mS) y en otros se acorta (80mS); a lo largo del tiempo el promedio será igual a T.

La línea (4), (6) y (7) contienen los comandos que permite tomar los datos que se hallan en el buffer de recepción.

c) Se redactarán las subrutinas qx\_OverRealTime y qx\_OnError, ambos son eventos que siguen la técnica tradicional de captura de errores. qx\_OnError se halla destinado a detectar envío de comandos no válido o respuestas no pertinentes. qx\_OverRealTime se activa cuando no se obtiene respuesta alguna, ello puede ocurrir por desconexión de la RIAC, fallas en el puerto, etc. Estos eventos solo se ejecutaran en forma automática si aparecen errores, cada aplicación deberá resolver las acciones a realizar.

## ¿Cuánto de rápido?

Como fuera antes mencionado, el comando RealTime inicia el muestreo en forma automática, en tanto el periodo de muestreo, queda determinado por la expresión  $T = n \times m \times 10 \text{ [ms]}$ ; donde n y m son enteros comprendidos entre 1 y 255. En la determinación del tiempo mínimo debe considerarse poder ingresar todos los caracteres de la trama de datos (fig.3). La expresión que sigue permite determinar el tiempo empleando en enviar toda la trama, en ella se toman en cuenta el número máximo de caracteres y la velocidad de comunicaciones. El periodo T de muestreo ( $n \times m \times 10 \text{ ms}$ ) deberá ser entonces mayor al tiempo t.

$$t \text{ [mSeg]} = 3 \text{ mSeg} + N_c \times \frac{10000}{V_{com}} \times F_c$$

Nc: Número de caracteres

Vcom: Velocidad de comunicación

Fc: Factor de corrección entre 1,05 y 1,43 depende de capacidad de procesamiento de la PC.

En el cuadro de la figura y a partir de la expresión dada se presentan a modo de ejemplos los tiempos mínimos de muestreo a diversas velocidades.

Velocidad [Baudios]	tiempo 't' de una trama	Periodo T, minimo posible
9600	87,90 mSeg	90 mSeg
19200	45,45 mSeg	50 mSeg
38400	24,22 mSeg	30 mSeg
57600	17,15 mSeg	20 mSeg
115200	10,07 mSeg	20 mSeg

Periodo mínimo en intervalos cuánticos de 10mSeg. ( $N_c = 57, F_c = 1.43$ ).

Finalmente la trama de la figura 3 corresponde al estándar del módulo RIAC-QF (10bits).

**Dpto. Técnico  
microAXIAL**