



Información sobre la configuración de una aplicación para transmitir datos ASCII por puerto serie, con pantallas terminales Magelis.

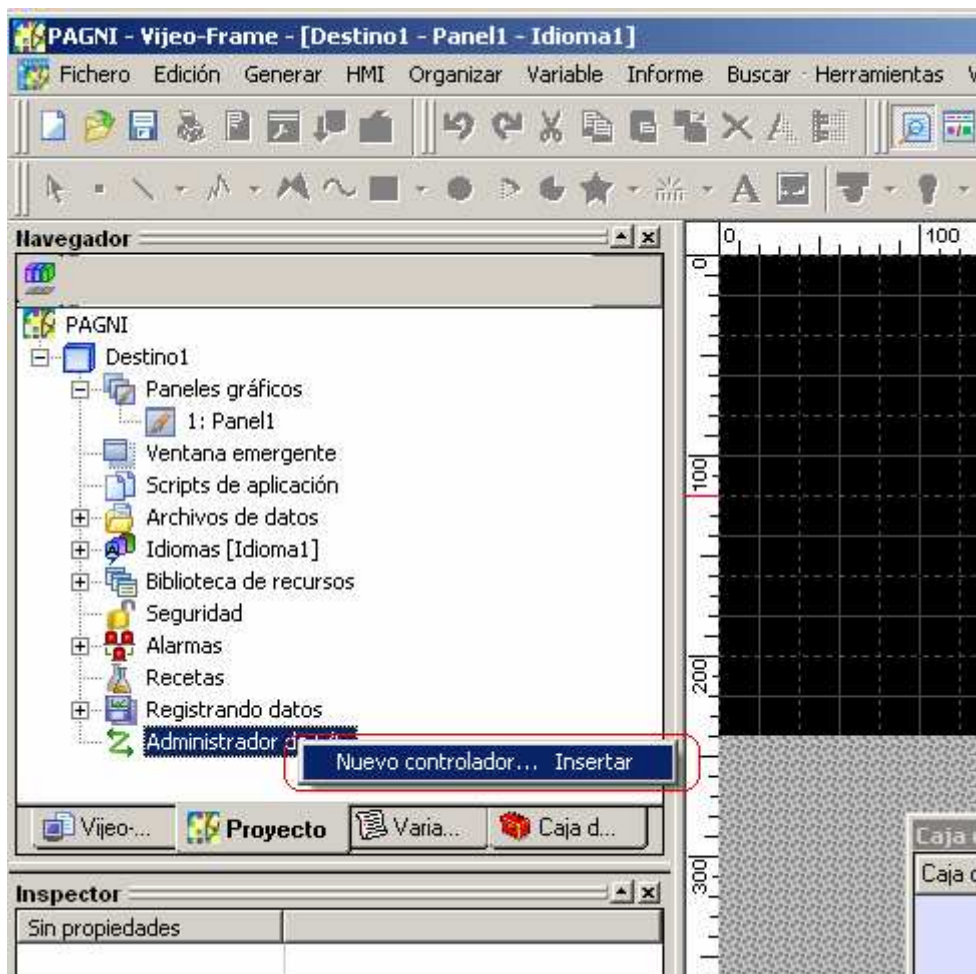
Materiales:

En la confección de la presente Nota de Aplicación se utilizó una pantalla Terminal **Magelis 2330**, cable de programación serie **XBTZ0935** y el software **Vijeo Designer**.

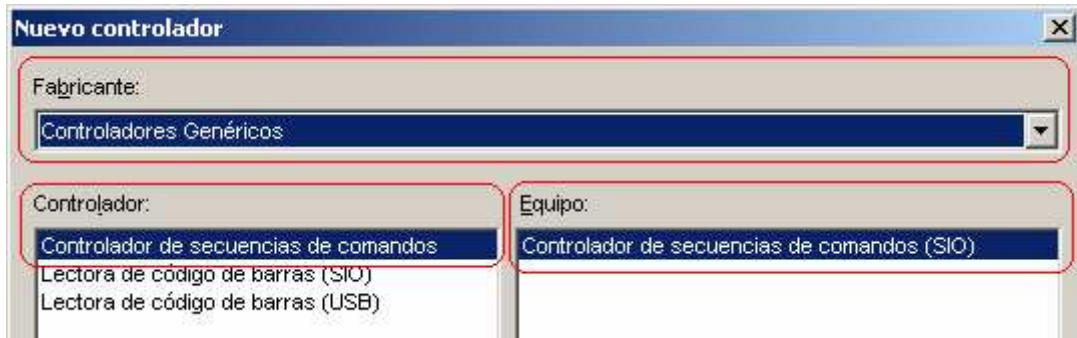
Configuración del controlador de scripts:

En la aplicación de **Vijeo-Designer**, agregue el controlador de scripts en el nodo de **Administrador de E/S**.

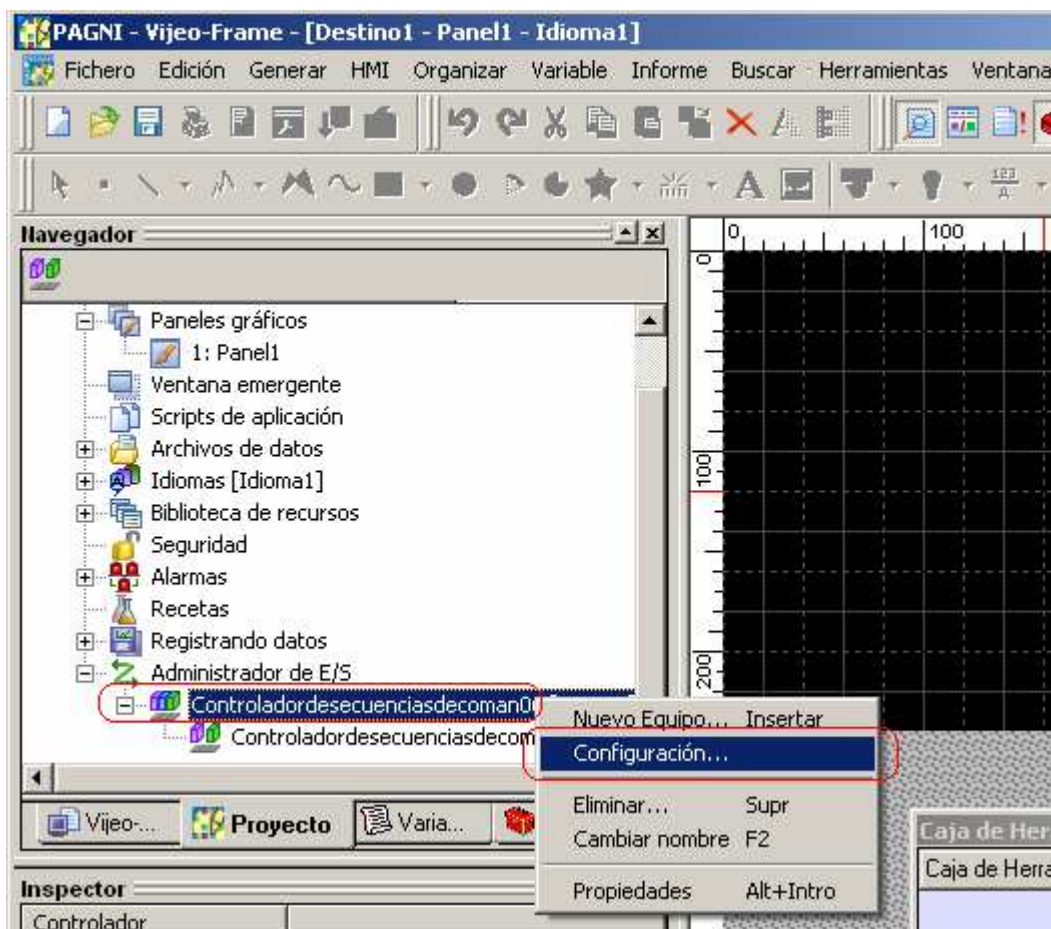
1. Haga clic con el botón derecho en el nodo de **Administrador de E/S** y, después, seleccione **Nuevo controlador**.



2. Seleccione **Controladores genéricos** como Fabricante, elija **Controlador de Secuencias de comandos**, y haga clic en **Aceptar**.



3. Haga clic con el botón derecho del ratón en el nuevo controlador agregado y seleccione **Configuración** para acceder al cuadro de diálogo **Configuración** del controlador.



4. En el cuadro de diálogo **Configuración del controlador**, defina los ajustes de comunicación del controlador.

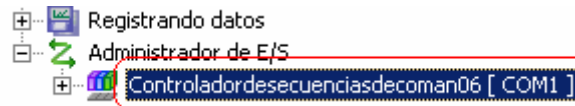


Asegúrese de que estos ajustes coinciden con los ajustes de la comunicación serie del equipo.

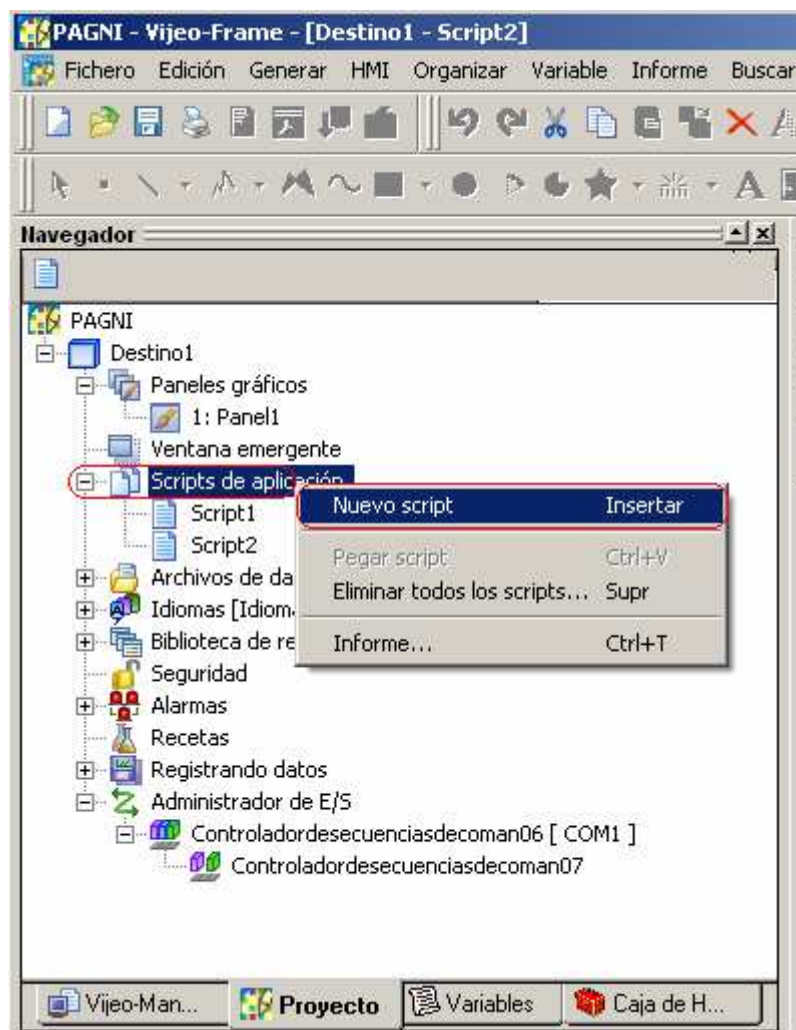
Campo	Descripción
Puerto COM	En la máquina de destino, seleccione el puerto COM para las comunicaciones serie.
Interfaz serie	<p>Seleccione la conexión serie utilizada en la máquina de destino: RS-232C, RS-422 (2 cables) o RS-422 (4 cables).</p> <p>Par obtener información acerca de la conexión del equipo serie, refiérase a la Sección 18.8.4, Conexiones de cables.</p>
Control de flujo	Si utiliza RS-232C, las señales de patilla utilizadas para el control de flujo son DTR(ER)/CTS. Si utiliza RS-422, no existe control de flujo.
Velocidad de transmisión	<p>Define la velocidad de comunicación, que corresponde a la velocidad en baudios del equipo.</p> <p>El controlador de scripts admite velocidades por encima de 2.400 bps y velocidades por debajo de 115.200 bps.</p>
Bit de paridad	Define el bit de paridad utilizado para validar paquetes de comunicación: Par, Impar o Ninguno.
Bit de parada	Define el bit de parada (1 ó 2) de los paquetes de comunicación.
Longitud de los datos	Define la longitud de datos (7 u 8 bits) de los paquetes de comunicación.



5. Haga clic en Aceptar para finalizar la configuración del controlador de *scripts*.
6. Cambie el nombre del controlador de forma apropiada.



Este nombre se usa en los *scripts* cuando se envía/recibe comunicaciones con los métodos de **write** y **read**.





Función `write`

Envía la cadena definida como flujo de bytes al puerto de comunicaciones definido.

Puede utilizar los métodos de *read/write* del controlador de *scripts* para ejecutar comunicaciones básicas entre la máquina de destino y un equipo. Dicho equipo puede ser un Autómata, una impresora u otra máquina de destino.

Formato:

```
boolean ScriptDriverName.write(  
    boolean clearBuffer,  
    int numOfBytes,  
    byte[] data );
```

Parámetros

Parámetro	Descripción
boolean clearBuffer	<p>Cuando es verdadero, elimina los datos en el búfer de recepción de la máquina de destino (<i>búfer</i> del controlador de <i>scripts</i>) antes de realizar la transmisión. .</p> <p>NOTA:</p> <ul style="list-style-type: none">• Utilice este parámetro cuando espere una respuesta desde el equipo serie. Configure este parámetro en "true" para eliminar todos los datos almacenados en el búfer de recepción, de forma que será más fácil analizar cualquier dato recibido.• Si el equipo serie conectado está iniciando el envío o recepción de los datos, entonces este parámetro debe ser "false" — los datos en el búfer del controlador de <i>scripts</i> no se eliminan. Es posible que hayan datos enviados por el equipo conectado que aún necesiten leerse.
int numOfBytes	<p>Define el número de <i>bytes</i> de la transmisión de datos. Se puede escribir hasta 1024 <i>bytes</i> de datos en una sola transmisión.</p> <p>IMPORTANTE:</p> <ul style="list-style-type: none">• Trate de configurar las aplicaciones para no tener que escribir grandes cantidades de datos. Todas las operaciones en la máquina de destino estarán bloqueadas hasta que se finalice la operación de escritura.



	<ul style="list-style-type: none">• Cuando escribe 1024 <i>bytes</i> con una velocidad de transmisión de 19200 bps, la máquina de destino deja de ejecutarse por 2 o 3 segundos.
byte[] data	Se ha enviado los datos al puerto de comunicación definido en el controlador de <i>scripts</i> .

Valor de retorno

Devuelve **true** cuando la operación de lectura se realiza correctamente. De lo contrario, devuelve **false**.

Comentarios:

El método de escritura impide la ejecución de otras operaciones mientras envía datos mediante el puerto serie.



Función read

Lee el flujo de bytes desde el puerto de comunicaciones definido.

Puede utilizar los métodos de lectura/escritura del controlador de scripts para ejecutar comunicaciones básicas entre la máquina de destino y un equipo. Dicho equipo puede ser un Autómata, una impresora u otra máquina de destino.

Formato:

```
int ScriptDriverName.read(  
    int timeout,  
    int numOfBytes,  
    byte[] data );
```

Parámetros

Parámetro	Descripción
int timeout	<p>Especifica el tiempo que el método de lectura esperará a los datos antes de detenerse. En la mayoría de los casos, el <i>timeout</i> debe ser cero.</p> <p>NOTA: Cuando no hay datos del puerto serie para leer, el controlador de <i>scripts</i> espera el <i>timeout</i> definido. Durante el <i>timeout</i>, el <i>script</i> impide la ejecución de otras operaciones del mismo tipo, lo cual podría causar que se ignoren las entradas del operador o que se retrasen los <i>scripts</i> periódicos. Sólo use el <i>timeout</i> si tiene una razón específica que requiera de un <i>timeout</i>.</p> <ul style="list-style-type: none">• No especifique un <i>timeout</i> largo si anticipa que los datos no estarán disponibles para lectura.• Si desea obtener información sobre los efectos secundarios del uso de un <i>timeout</i>.
int numOfBytes	<p>Define el número esperado de bytes en el flujo de bytes.</p> <p>NOTA:</p> <ul style="list-style-type: none">- La cantidad máxima de datos que se puede leer es 1024 <i>bytes</i>. Si desea leer más datos, ejecute el método de lectura otra vez.- El numOfBytes y el número de elementos en la matriz de bytes deben ser iguales. Si el numOfBytes es mayor, es posible que lea más datos de lo que se pueda almacenar.



byte[] data	<p>Área de almacenamiento del flujo de bytes. Cuando lee datos del puerto serie, se almacenan en esta matriz de bytes.</p> <p>NOTA: Cuando se refiera a los datos en la matriz, recuerde que la referencia del elemento de matriz empieza en 0. Por lo tanto, lea el byte octavo en el la secuencia de bytes, usar data[7].</p>
---------------------	---

Valor de retorno

Devuelve el número de bytes recibido. Un valor negativo (-1) indica que se ha producido un error.

NOTA:

- El valor de retorno máximo es **numOfBytes**. Aunque el búfer del controlador de scripts tenga mas datos, **numOfBytes** define el número máximo de bytes que se leen. Por ejemplo, si hay 110 bytes de datos en el búfer del controlador de scripts y el **numOfBytes** es 100:
 - lee 100 bytes
 - valor de retorno es 100
 - 10 bytes permanecen en el búfer del controlador de scripts.

Si hay 10 bytes de datos en el búfer del controlador de scripts y el **numOfBytes** es 10:

- lee 10 bytes
- valor de retorno es 10
- 0 bytes permanecen en el búfer del controlador de scripts.



Puntos importantes sobre el controlador de *scripts*

Leer tamaño del *búfer*

Cuando use el controlador de *scripts* para leer datos de equipos, puede especificar el número de bytes a leer, hasta un máximo de 1024 bytes. El número máximo es 1024 porque el búfer del controlador de *scripts* sólo almacena 1024 bytes.



Si desea leer más de 1024 bytes de datos, debe ejecutar operaciones múltiples.

- Hay dos búferes involucrados en las comunicaciones del controlador de *scripts*: el búfer del controlador de *scripts* y el búfer del puerto serie. El búfer del puerto serie almacena un máximo de 4096 bytes. Si el equipo envía más de 4096 bytes datos y no se leen, el búfer del puerto serie se desbordará y se perderán datos.
- La cantidad máxima de datos que puede escribir en un momento dado es 1024 bytes.

Operaciones de Lectura/Escritura requieren un tiempo de espera

Cuando se desea leer datos desde un equipo generalmente necesita enviar una solicitud. Use el método *write* para realizarlo. Después de enviar la solicitud habrá un retraso antes de que el equipo mande una respuesta. Si desea leer los datos enviados por el equipo, el *script* debería incluir un tiempo de espera que tome en cuenta este retraso.



Vea el ejemplo a continuación.

```
ScriptDriver.write(true, 8, SendPacket);  
  
ScriptDriver.read(100, 7, RecvPacket); //100ms tiempo de  
//espera
```

Después de enviar el comando de escritura, el script ejecuta el comando de lectura inmediatamente. Sin embargo, en este momento, el equipo no ha respondido y el comando de lectura espera por los datos.

Cuando por fin se reciben los datos, el comando de lectura se termina y continúa con el resto del *script*.

Si no se recibe ningún dato, el comando de lectura espera a que transcurra el tiempo de espera antes de continuar con el resto del *script*. Durante este período de espera (*timeout*), el *script* impide la ejecución de otras operaciones.

Bloqueo mientras se lee con un tiempo de espera

En la mayoría de los casos, cuando las operaciones son normales, el tiempo de espera no afectará a la aplicación. En las operaciones normales, el equipo responde de forma rápida, generalmente dentro de 100ms para las aplicaciones que trabajan con pocos datos (100 bytes o menos). El tiempo de espera se pone en funcionamiento en las operaciones anormales, por ejemplo, cuando hay un fallo en el equipo y no existe ningún dato en el puerto serie.

Durante el tiempo de espera, el *script* impide la ejecución de otras operaciones del mismo tipo. Las consecuencias del bloqueo son diferentes entre una y otro *script*.

Los efectos de usar un tiempo de espera

Lo siguiente describe las consecuencias no deseadas que ocurren en varios tipos de *scripts* que usan un tiempo de espera y mientras no hay ningún dato para leer desde el puerto serie. Use el tiempo de espera de manera controlada y limitada y pruebe todas las operaciones en *runtime*.

- **Scripts condicionales** (*Script* de aplicación o de panel): Disparador = Condicional) puede causar que el panel parezca estar congelado. El uso de un tiempo de espera con este tipo de *script* detiene la comunicación, la actualización de los valores de variables y detiene el dibujar y la animación.
- **Los scripts de Panel Al abrir/Al cerrar** causan el mismo problema que las *scripts* condicionales.
- **Los scripts de Alarma de Acción del disparador** causan el mismo problema que las *scripts* condicionales.
- **Scripts de pulsación** (*script* de aplicación: Disparador = Panel Al contacto, *script* de interruptor, *script* de Animación de contacto, *script* de Tecla de función, *script* de Acción de contacto de alarma) deje de ejecutar operaciones



de contacto durante el tiempo de espera. Las operaciones al contacto se almacenan en una cola que puede almacenar hasta cincuenta pulsaciones. Una vez transcurrido el tiempo de espera, se ejecutan las operaciones de contacto en cola. No afecta a otras operaciones tal como la comunicación y el dibujar.

IMPORTANTE

- El uso de tiempos de espera largos puede ser peligroso puesto que el operador del panel no puede responder a las alarmas y otros acontecimientos durante el tiempo de espera.
- El tiempo máximo recomendado para un tiempo de espera es un segundo (1000 ms).

Se puede usar tiempos de espera sin riesgo

Puede usar los tiempos de espera sin riesgo con los *scripts* de aplicación Al arranque y *scripts* periódicas con una Frecuencia única. Existen efectos secundarios cuando se usa un tiempo de espera con estos *scripts*, sin embargo, es posible que el impacto sea mínimo, según sean los requisitos de *runtime*.

- Los ***scripts* de Aplicación Al arranque** aumentan el tiempo requerido para iniciar y arrancar la aplicación de *runtime* en la máquina de destino y retrasar la comunicación con los equipos durante el tiempo de espera. Si un retraso durante el arranque no afecta a las operaciones, entonces puede usar el tiempo de espera sin incidente.
- ***Scripts* periódicas** (*Script* de aplicación o de panel): Disparador = Periódico) bloquee la configuración de otras *scripts* de aplicación o de panel periódicas *con la misma frecuencia*. Si usa un valor único para la frecuencia, este efecto puede ser insignificante.

Bloquear al escribir

Cuando use el método de escritura para enviar grandes cantidades de datos, todas las operaciones en la máquina de destino estarán bloqueadas hasta que se finalice la operación de escritura.

Si desea minimizar el impacto de las operaciones de escritura, se recomienda dividir los datos en partes más pequeñas, por ejemplo, 100 bytes, para cada operación de escritura.



IMPORTANTE

- El método de escritura del controlador de *scripts* bloquea TODAS operaciones en la máquina de destino hasta que se finalice la operación de escritura. Trate de configurar las aplicaciones para no tener que escribir grandes cantidades de datos de una vez.
- Cuando escribe el máximo de 1024 bytes con una velocidad de transmisión de 19200 bps, la máquina de destino no responde por 2 o 3 segundos.

CONSIDERACION ESPECIAL

Parte de cualquier ciclo de desarrollo incluye probar la aplicación del usuario antes de comenzar con la producción. Sin embargo, préstele mucha atención a cualquier aplicación de controlador de scripts que vaya a crear.