



Uso de los CGI's del proyecto CRR

Los Scripts CGI necesarios para la comunicación con el puerto paralelo de formar remota, están escritos en PERL, y se instalan por defecto en la carpeta:

C:\Archivos de programa\EasyPHP1-8\cgi-bin\crr

Importante: todos los CGI del proyecto CRR depende de la librería "entorno2.lib". Por tanto, dicho archivo debe estar en la misma carpeta que los CGI`s.

Conecta.cgi

Descripción:

Permite conectar una salida de la controladora, activando los bits correspondientes de los registros de Datos y de Control del puerto paralelo.

Parámetros:

- salida: Solamente dispone de un parámetro denominado "salida", en el que se le debe indicar con un número el bit de la salida a activar.

Bits de registro de datos: 0 al 7.

Bits del registro de control: 16 al 19

Ejemplos:

.....cgi-bin/crr/conecta.cgi?salida=1

.....cgi-bin/crr/conecta.cgi?salida=16

Apaga.cgi

Descripción:

Permite desconectar una salida de la controladora, que haya sido previamente activada. Pone a 0 el bit correspondiente del registro de control o de datos del puerto paralelo.

Parámetros:

- salida: Solamente dispone de un parámetro denominado "salida", en el que se le debe indicar con un número el bit de la salida a desactivar.

Bits de registro de datos: 0 al 7.

Bits del registro de control: 16

Ejemplos:

.....cgi-bin/crr/apaga.cgi?salida=1

.....cgi-bin/crr/apaga.cgi?salida=16

Conectatodo.cgi

Descripción:

Permite conectar a la vez las 12 salidas de la controladora. Pone a 1 lógico, todos los bits de salidas de los registros de datos y control del puerto paralelo.

Parámetros:

No tienen.

Ejemplo:

.....cgi-bin/crr/conectatodo.cgi



Apagatodo.cgi

Descripción:

Permite desconectar a la vez las 12 salidas de la controladora. Pone a 0 lógico, todos los bits de salidas de los registros de datos y control del puerto paralelo.

Parámetros:

No tienen.

Ejemplo:

.....cgi-bin/crr/apagatodo.cgi

Crr.cgi

Descripción:

Permite enviar datos a los registros del puerto.

Parámetros

- valor_registro: Elige el registro que va a recibir el dato. 0 – Registro de datos; 1 – Registro de estado ; 2- Registro de control
- valor_byte: dato entre 0 y 255 a enviar a registro seleccionado.

Ejemplos:

Envía el dato 255 al registro de datos (0).

http://...../cgi-bin/crr/crr.cgi?valor_registro=0&valor_byte=255

Envía el dato 15 al registro de control (2).

http://...../cgi-bin/crr/crr.cgi?valor_registro=2&valor_byte=15

Set_data.cgi

Descripción

Envía un dato al registro de datos del puerto paralelo.

Parámetros

- valor_datos: dato a enviar al registro de dato (entre 0 y 255)

Ejemplos:

http://...../cgi-bin/crr/set_data.cgi?valor_datos=12

Set_control.cgi

Descripción

Envía un dato al registro de control del puerto paralelo.

Parámetros

- valor_control: dato a enviar al registro de control (entre 0 y 15)

Ejemplos:

http://...../cgi-bin/crr/set_control.cgi?valor_control=12

pap1.cgi

Descripción

Trabaja con los cuatro bits de mayor peso del registro de datos, para mover el motor PaP número 1 en un sentido de giro.

A través de los parámetros se le puede establecer el número de pasos y la cadencia entre ellos.

IMPORTANTE: El motor 1 se para en este sentido de giro si la señal de la entrada 1 (bit 3 del registro de estado) es 0 (debe usarse un contacto normalmente cerrado)

Parámetros



- motor1: pone en marcha el motor
- temp: tiempo en ms entre pasos. A menor tiempo mayor velocidad de giro.
- pasos2: número de pasos que el motor debe avanzar.

Ejemplo:

El motor 1 arranca, hasta hacer 20 pasos, con un intervalo entre pasos de 5 ms.

<http://...../cgi-bin/crr/pap1.cgi?motor1=On&temp=50&pasos2=20>

[pap1_inv.cgi](#)

Descripción

Trabaja con los cuatro bits de mayor peso del registro de datos, para mover el motor PaP número 1 en sentido contrario a pap1.cgi.

A través de los parámetros se le puede establecer el número de pasos y la cadencia entre ellos.

Parámetros

- motor1_inv: pone en marcha el motor
- temp1_inv: tiempo en ms entre pasos. A menor tiempo mayor velocidad de giro.
- Pasos1: número de pasos que el motor debe avanzar.

Ejemplo:

El motor 1 arranca, en sentido contrario al ejemplo anterior, hasta hacer 20 pasos, con un intervalo entre pasos de 5 ms.

http://...../cgi-bin/crr/pap1_inv.cgi?motor1_inv=On&temp1_inv=50&pasos1=20

[parar.cgi](#)

Descripción

Envía una señal de parada para el motor PaP número 1 en cualquier sentido de giro.

Parámetros

- motor1: envía Off al CGI para parar el motor.

Ejemplo:

<http://...../cgi-bin/crr/parar.cgi?motor1 =Off>

[pap2.cgi](#)

Descripción

Trabaja con los cuatro bits de menor peso del registro de datos, para mover el motor PaP número 2 en un sentido de giro.

Parámetros

- motor2: pone en marcha el motor
- temp2: tiempo en ms entre pasos. A menor tiempo mayor velocidad de giro.

Ejemplo:

El motor 2 arranca en un sentido de giro, con un intervalo entre pasos de 50 ms.

<http://...../cgi-bin/crr/pap2.cgi?motor2=On&temp2=50>

[pap2_inv.cgi](#)

Descripción

Trabaja con los cuatro bits de menor peso del registro de datos, para mover el motor PaP número 2 en sentido contrario a pap2.cgi.



IMPORTANTE: A este motor no se le ha previsto que se pare con ninguna señal de entrada.

Parámetros

- motor2_inv: pone en marcha el motor
- temp2_inv: tiempo en ms entre pasos. A menor tiempo mayor velocidad de giro.

Ejemplo:

El motor 2 arranca, en sentido contrario al ejemplo anterior con un intervalo entre pasos de 5 ms.

http://...../cgi-bin/crr/pap2_inv.cgi?motor2_inv=On&temp2_inv=50

[parar2.cgi](#)

Descripción

Envía una señal de parada para el motor PaP número 2 en cualquier sentido de giro.

Parámetros

- motor2: envía Off al CGI para parar el motor.

Ejemplo:

<http://...../cgi-bin/crr/parar2.cgi?motor2 =Off>

[pap3.cgi](#)

Descripción

Trabaja con los cuatro bits de menor peso del registro de control, para mover el motor PaP número 3 en un sentido de giro.

IMPORTANTE: : El motor 3 se para en este sentido de giro si la señal de la entrada 1 (bit 3 del registro de estado) es 0 (debe usarse un contacto normalmente cerrado)

Parámetros

- motor3: pone en marcha el motor
- temp3: tiempo en ms entre pasos. A menor tiempo mayor velocidad de giro.
- pasos4: número de pasos que el motor debe avanzar.

Ejemplo:

El motor 3 se mueve en un sentido de giro 20 pasos, con un intervalo entre pasos de 50 ms.

<http://...../cgi-bin/crr/pap3.cgi?motor3=On&temp3=50&pasos4=20>

[pap3_inv.cgi](#)

Descripción

Trabaja con los cuatro bits de menor peso del registro de control, para mover el motor PaP número 3 en sentido contrario a pap3.cgi.

Parámetros

- motor3_inv: pone en marcha el motor
- temp3_inv: tiempo en ms entre pasos. A menor tiempo mayor velocidad de giro.
- Pasos3: número de pasos que el motor debe avanzar.

Ejemplo:

El motor 2 se mueve 30 pasos en sentido contrario al ejemplo anterior, con un intervalo entre pasos de 5 ms.

http://...../cgi-bin/crr/pap3_inv.cgi?motor3_inv=On&temp3_inv=5&pasos3=30



parar3.cgi

Descripción

Envía una señal de parada para el motor PaP número 3 en cualquier sentido de giro.

Parámetros

- motor2: envía Off al CGI para parar el motor.

Ejemplo:

`http://...../cgi-bin/crr/parar3.cgi?motor3 =Off`

motor_pap1.cgi

Todo igual que para pap1.cgi excepto que su funcionamiento no está condicionado a ninguna entrada de la controladora.

motor_pap2.cgi

Todo igual que para pap2.cgi excepto que su funcionamiento no está condicionado a ninguna entrada de la controladora.

motor_pap3.cgi

Todo igual que para pap3.cgi excepto que su funcionamiento no está condicionado a ninguna entrada de la controladora.

entradas.cgi

Descripción

Permite leer las entradas de la controladora de forma remota.

Este archivo debe estar en la carpeta `...\\cgi-bin\\crr\\` del servidor web.

El Script tiene 3 posibilidades de mostrar los resultados:

1- A través de un archivo de texto llamado "entradas.txt". En este caso, cada línea del archivo muestra el valor de una entrada. Por tanto, desde el lenguaje de programación se debe leer una línea para obtener el valor de una entrada. La primera línea es la entrada 1, la segunda línea la entrada 2, etc.

La ejecución de comandos a seguir por el lenguaje de programación es la siguiente:

- * Ejecutar el CGI
- * Descargar el archivo "entradas.txt" al ordenador cliente.
- * Abrir el archivo con la aplicación cliente y leer las líneas de forma individual, asociando cada una de ellas a una variable, que sería el estado de la entrada correspondiente.

2- Este es similar al anterior. En este caso, el resultado se registra en un archivo llamado "entradas2.txt" en una sola línea, para que el dato se pueda leer como un número binario (Mayor peso a la izquierda y menor peso a la derecha).

La ejecución de comandos es idéntica al método número 1.

3- Extraer el resultado devuelto por el CGI en una variable de la aplicación cliente. En este caso, el valor almacenado en la variable será algo así: 10010.



Ahora aquí se pueden utilizar varias formas para leer el estado de las entradas. Sin embargo, la forma más inmediata puede ser extrayendo individualmente cada carácter de la cadena y asociar cada uno de ellos a una variable.

De esta forma se obtiene el valor de cada entrada en una variable diferente y así se puede trabajar con ellas de forma individual.

También se puede leer de dato registrado en el archivo como un número binario.

Parámetros

- No tiene.

Ejemplo:

Para que esto funcione se debe hacer lo siguiente:

Dentro de EasyPhp

El CGI (entradas.cgi) debe ser copiado a la carpeta:

../cgi-bin/crr/

Los archivos de texto se crean automáticamente, al menos así debería ser, en:

.../www/

por tanto las rutas para ejecutar el CGI debe ser del tipo:

<http://localhost/cgi-bin/crr/entradas.cgi>

y para leer los archivos de texto:

<http://localhost/entradas.txt>

<http://localhost/entradas2.txt>

Con la tercera opción, no es necesario utilizar el archivo de texto. En teoría esta opción debería ser la más rápida en ejecutarse en Internet.

Ejemplos WEB

Una vez instalada la Aplicación del Entorno CRR, en la carpeta

C:\Archivos de programa\EasyPHP1-8\www\panel_crr

se puede localizar un proyecto para el acceso a los CGI's mediante vínculos y formularios WEB

Esta aplicación se puede lanzar desde el navegador con:

http://localhost/panel_crr/

y se puede comprobar su funcionamiento mediante el simulador CRR.

