**BLOQUE GENERICO DE MEMORIA COMPARTIDA**

****

**Introduccion**

Este bloque permite la comunicacion de Matlab Simulink con programas externos a traves de memoria compartida.

**Compatibilidad**

Este bloque es compatible con Windows y Linux de 32 y 64 bits. Asegúrese de tener el archivo compilado ".mex" adecuado para su sistema operativo y versión de Matlab en la carpeta de la librería o bien en la carpeta de trabajo.

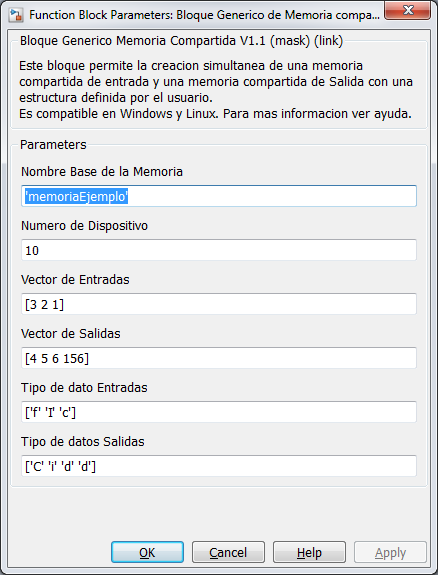
Este bloque es compatible con cualquier versión de Matlab que soporte s-functions. Sin embargo se recomienda Matlab R2014b x64 o posteriores.

**Funcionamiento General**.

Al principio de la ejecución el bloque lee los parámetros entregados, a partir de ellos se genera una memoria compartida de entrada y otra memoria compartida de salida con el tamaño y tipo especificado. A estas memorias se les asigna una Id y almacena en un contenedor que permite la gestión interna de las mismas, esto posibilita varias instancias del bloque a la vez.

En cada paso de simulación el bloque lee y escribe las memorias de entrada y de salida. El bloque no cuenta con ningún tipo de sincronización por lo que el usuario debe prever no escribir en el mismo espacio de memoria en el que escribe el bloque.

**Descripción de parámetros (Disponible en Help del Bloque)**

****

**Nombre Base de Memoria (nombre):** Es el string base de la memoria compartida ej 'robot'

**Numero de Dispositivo (dispositivo):** entero que identifica el dispositivo entre varios en uso

Se crean dos memorias compartidas con el nombre:

"%name\_entrada\_%dispositivo" para la entrada al dispositivo (escritura) .

"%name\_salida\_%dispositivo" para la salida del dispositivo (lectura)

*El usuario debe asegurarse de no iniciar dos bloques con el mismo nombre y numero de dispositivo porque se generará un error.*

**Vector de Entradas (vecE):** arreglo que especifica la cantidad de entradas y su dimensión

**Vector de Salidas (vecS):** arreglo que especifica la cantidad de entradas y su dimensión

**Tipo Entradas(tipoE):** arreglo de caracteres que indica el tipo de cada entrada

**Tipo Salidas (tipoS):** arreglo de caracteres que indica el tipo de cada salida

Los tipos pueden ser de la siguiente forma:

('f'=float, 'd'=double, 'i'=int32, 'c'=char, 'C'= unsigned char, 'I'=unsigned int)

Ejemplo.

Suponer los siguientes parametros:

nombre='memoriaEjemplo'

dispositivo=10

vecE=[3 2 1]

vecS=[6 5 7 8]

tipoE=['f' 'I' 'c']

tipoS=['C' 'i' 'd' 'd']

Crean dos memorias compartidas denominadas :

"memoriaEjemplo\_entrada\_10" (entrada al bloque)

"memoriaEjemplo\_salida\_10" (salida del bloque)

Se crean 3 entradas:

1° Dimension 3 y tipo "unsigned char"

2° Dimension 2 y tipo "int"

3° Dimension 1 y tipo "char"

Se crean 4 salidas:

1° Dimension 6 y tipo "float"

2° Dimension 5 y tipo "unsigned int"

3° Dimension 7 y tipo "double"

3° Dimension 8 y tipo "double"

Esto es el equivalente a una estructura de entrada en C/C++

typedef struct ENTRADA {

unsigned char entrada1[3]

int entrada2[3]

char entrada3[3]

} memoriaEjemplo\_entrada\_10

typedef struct SALIDA {

float salida1[6]

unsigned int salida2[5]

double salida3[7]

double salida4[8]

} memoriaEjemplo\_salida\_10