

inductorDLL

1.0

Generated by Doxygen 1.8.11

Contents

1	File Index	1
1.1	File List	1
2	File Documentation	3
2.1	dll.h File Reference	3
2.1.1	Function Documentation	3
2.1.1.1	__declspec(dllexport) simuser(double t	3
2.1.2	Variable Documentation	4
2.1.2.1	delt	4
2.1.2.2	in	4
2.1.2.3	out	4
2.2	dllmain.cpp File Reference	4
2.2.1	Function Documentation	4
2.2.1.1	simuser(double t, double delt, double *in, double *out)	4
2.2.2	Variable Documentation	5
2.2.2.1	delt	5
2.2.2.2	in	5
2.2.2.3	out	5
2.2.2.4	t	5
	Index	7

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

dll.h	3
dllmain.cpp	4

Chapter 2

File Documentation

2.1 dll.h File Reference

```
#include <stdio.h>
```

Functions

- void __stdcall __declspec (dllexport) [simuser](#)(double [t](#)
DLL routine funtion.

Variables

- void __stdcall double [delt](#)
- void __stdcall double double * [in](#)
- void __stdcall double double double * [out](#)

2.1.1 Function Documentation

2.1.1.1 void __stdcall __declspec (dllexport)

DLL routine funtion.

Parameters

<i>t</i>	Time, passed from PSIM by value
<i>delt</i>	Time step, passed from PSIM by value
<i>in</i>	Input array, passed from PSIM by reference

Returns

output array, sent back to PSIM (Note: the values of out[*] can be modified in PSIM)

The maximum length of the input and output array "in" and "out" is 30.

PSIM calls the DLL routine at each simulation time step. However, when the inputs of the DLL block are connected to the output of one of these discrete elements (zero-order hold, unit delay, integrator, differentiator, z-domain transfer function block, and digital filters), the DLL block is considered as a discrete element. In this case, the DLL block is called only at the discrete times.

The DLL block receives the values from PSIM as the input, performs the calculation, and sends the output back to PSIM. The node assignments are: the input nodes are on the left, and the output nodes are on the right. The sequence is from the top to the bottom.

2.1.2 Variable Documentation

2.1.2.1 void __stdcall double delt

2.1.2.2 void __stdcall double double* in

2.1.2.3 void __stdcall double double double* out

2.2 dllmain.cpp File Reference

```
#include "dll.h"
#include <windows.h>
#include <math.h>
#include <stdio.h>
```

Functions

- __stdcall void [simuser](#) (double [t](#), double [delt](#), double *[in](#), double *[out](#))
DLL routine funtion.

Variables

- double [t](#)
Time, passed from PSIM by value.
- double [delt](#)
Time step, passed from PSIM by value.
- double * [in](#)
Input array, passed from PSIM by reference.
- double * [out](#)
Output array, sent back to PSIM (Note: the values of out[] can be modified in PSIM)*

2.2.1 Function Documentation

2.2.1.1 __stdcall void [simuser](#) (double [t](#), double [delt](#), double * [in](#), double * [out](#))

DLL routine funtion.

Parameters

<i>t</i>	Time, passed from PSIM by value
<i>delt</i>	Time step, passed from PSIM by value
<i>in</i>	Input array, passed from PSIM by reference

Returns

output array, sent back to PSIM (Note: the values of out[*] can be modified in PSIM)

The maximum length of the input and output array "in" and "out" is 30.

PSIM calls the DLL routine at each simulation time step. However, when the inputs of the DLL block are connected to the output of one of these discrete elements (zero-order hold, unit delay, integrator, differentiator, z-domain transfer function block, and digital filters), the DLL block is considered as a discrete element. In this case, the DLL block is called only at the discrete times.

The DLL block receives the values from PSIM as the input, performs the calculation, and sends the output back to PSIM. The node assignments are: the input nodes are on the left, and the output nodes are on the right. The sequence is from the top to the bottom. Define "sum" as "static" in order to retain its value.

< Variable to recieve voltage value

< Variable to recieve current value

Receive the voltage from PSIM

Integrate the voltage to obtain the flux linkage

Calculate the current

Send the current back to PSIM

This is a method that does so much that I must write an epic novel just to describe how much it truly does.

2.2.2 Variable Documentation**2.2.2.1 double delt**

Time step, passed from PSIM by value.

2.2.2.2 double* in

Input array, passed from PSIM by reference.

2.2.2.3 double* out

Output array, sent back to PSIM (Note: the values of out[*] can be modified in PSIM)

2.2.2.4 double t

Time, passed from PSIM by value.

Index

- `__declspec`
 - `dll.h`, [3](#)
- `delt`
 - `dll.h`, [4](#)
 - `dllmain.cpp`, [5](#)
- `dll.h`, [3](#)
 - `__declspec`, [3](#)
 - `delt`, [4](#)
 - `in`, [4](#)
 - `out`, [4](#)
- `dllmain.cpp`, [4](#)
 - `delt`, [5](#)
 - `in`, [5](#)
 - `out`, [5](#)
 - `simuser`, [4](#)
 - `t`, [5](#)
- `in`
 - `dll.h`, [4](#)
 - `dllmain.cpp`, [5](#)
- `out`
 - `dll.h`, [4](#)
 - `dllmain.cpp`, [5](#)
- `simuser`
 - `dllmain.cpp`, [4](#)
- `t`
 - `dllmain.cpp`, [5](#)