Working with Giano

Richard Neil Pittman

Microsoft Research

Procedures

- Installing Giano
- Running Simulations in Giano from Visio
- Running Simulations from the Command Line
- Connecting to Simulations using Serplexd
- Simulating EB63 with HostFS
- Creating/modifying a configuration
- Creating a new Giano module

Procedure #1

- **Installing Giano**
- Running Simulations in Giano from Visio
- Running Simulations from the Command Line
- Connecting to Simulations using Serplexd
- Simulating EB63 with HostFS
- Creating/modifying a configuration
- Creating a new Giano module

Installing Giano

1. Install Giano from MSI file (V2), in a writeable directory.
2. Rebuild Giano if necessary. Look at the bin\recompile* scripts for debug/release differences.

Rebuilding Giano

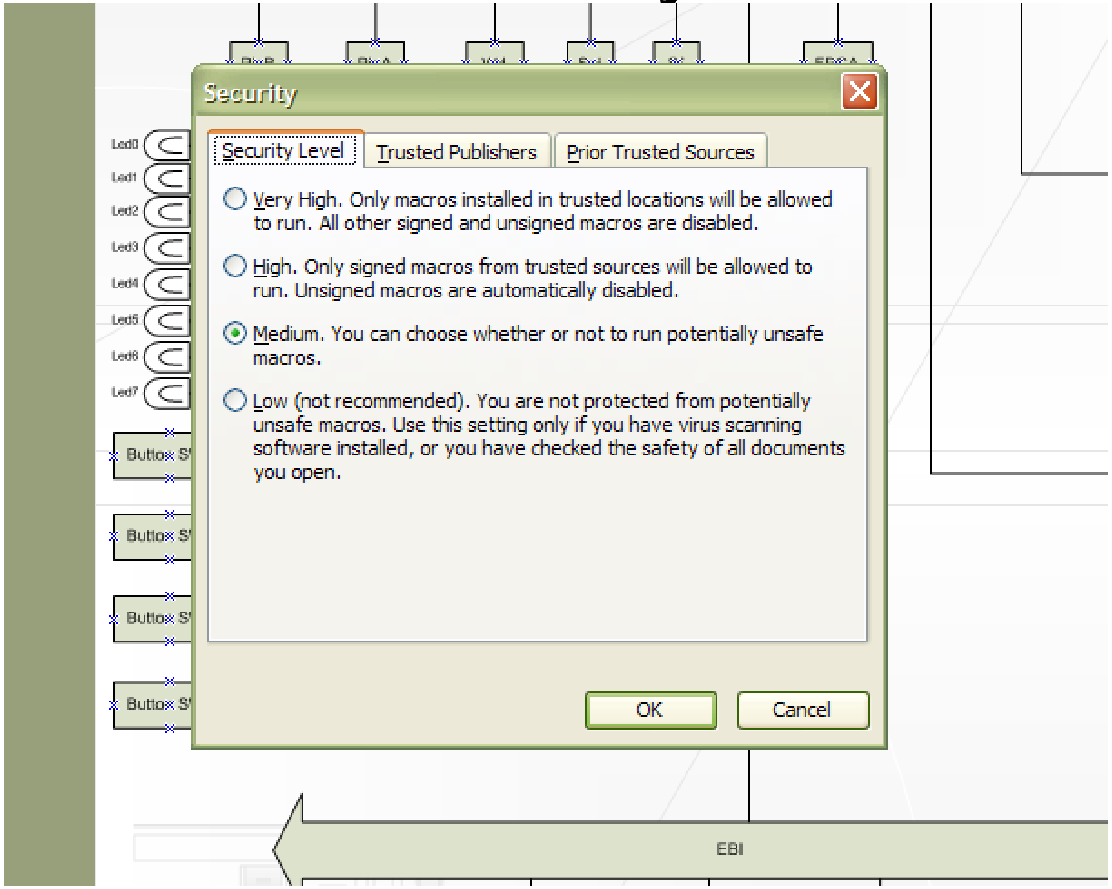
1. Have Microsoft Visual Studio installed on the Computer (VC6,7 and 8 are all ok)
2. Delete all *.manifest files from the subtree
3. Do not delete the following files from the ‘bin’ folder:

   - bbuniq.exe
   - bbw.dll
   - cbw32.dll
   - mcc.dll
   - msvcr71.dll
   - np.dll
   - vpi2g.dll
   - vssver.scc
   - vvp.dll
   - bbcat.exe
   - bbdump.exe
   - bbfind.exe
   - bbmatch.exe
   - bbmerge.exe
   - bbrewrite.exe
   - bbs.dll
   - bbselect.exe
   - bbsort.exe

4. Open a MVS Command Prompt and call ‘nmake clean all’ or “nmake clean debug”

Visio and Macros

- You must enable Visio macros, set Tools::Macros::Security to Medium or Low



○ Very High. Only macros installed in trusted locations will be allowed to run. All other signed and unsigned macros are disabled.

○ High. Only signed macros from trusted sources will be allowed to run. Unsigned macros are automatically disabled.

● Medium. You can choose whether or not to run potentially unsafe macros.

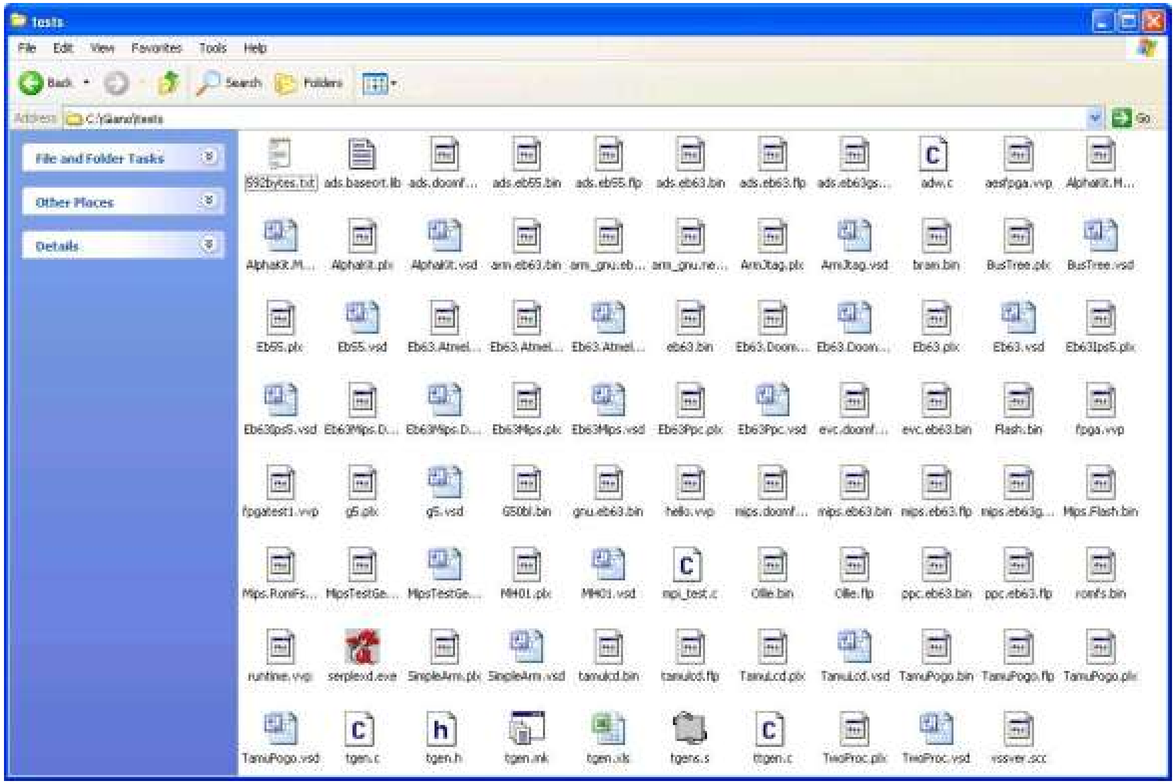
○ Low (not recommended). You are not protected from potentially unsafe macros. Use this setting only if you have virus scanning software installed, or you have checked the safety of all documents you open.

Procedure #2

- Installing Giano
- **Running Simulations in Giano from Visio**
- Running Simulations from the Command Line
- Connecting to Simulations using Serplexd
- Simulating EB63 with HostFS
- Creating/modifying a configuration
- Creating a new Giano module

Running Simulations in Giano from Microsoft Visio

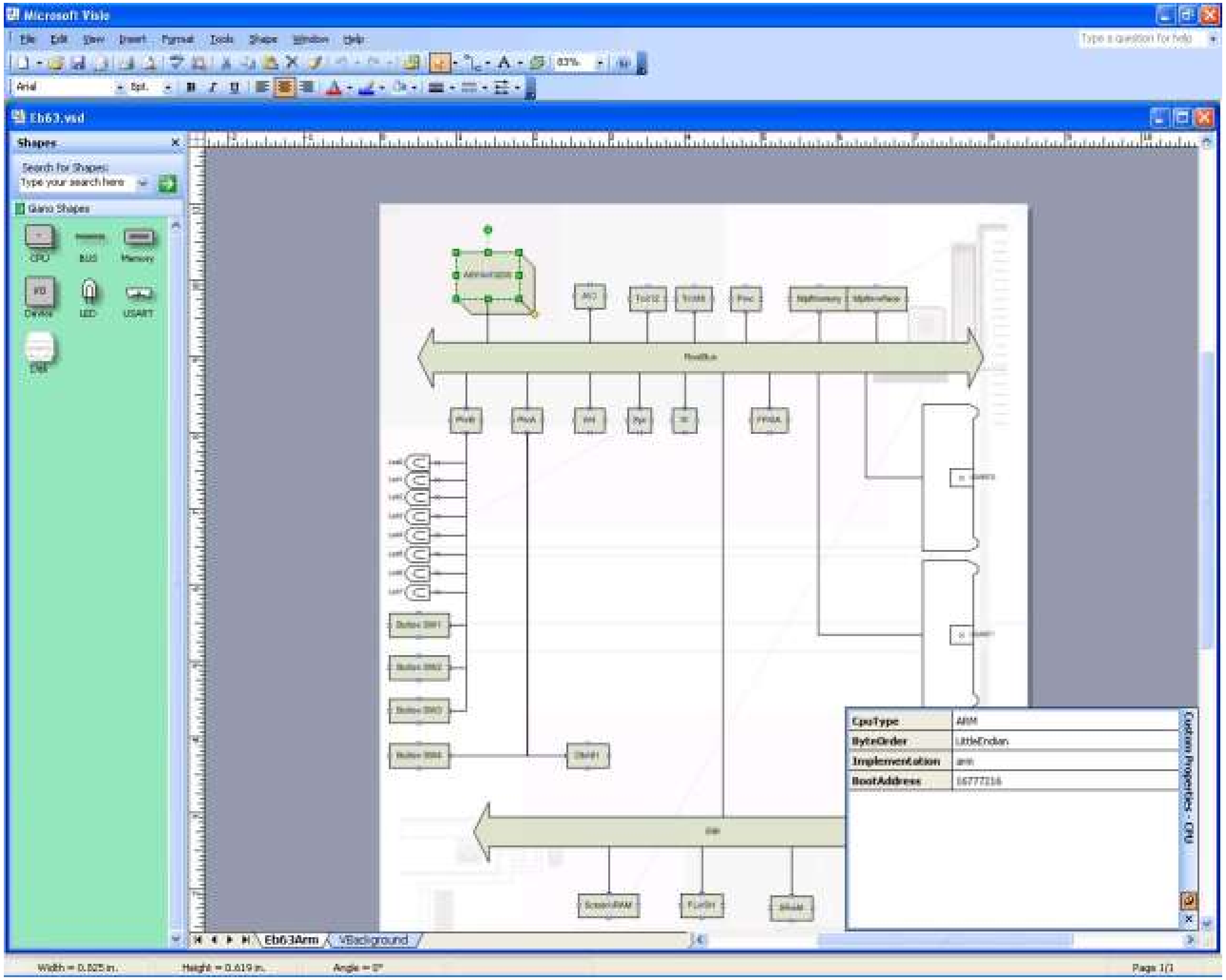
1. Open the *.vsd of the configuration you wish to run.



Running Simulations in Giano from Microsoft Visio

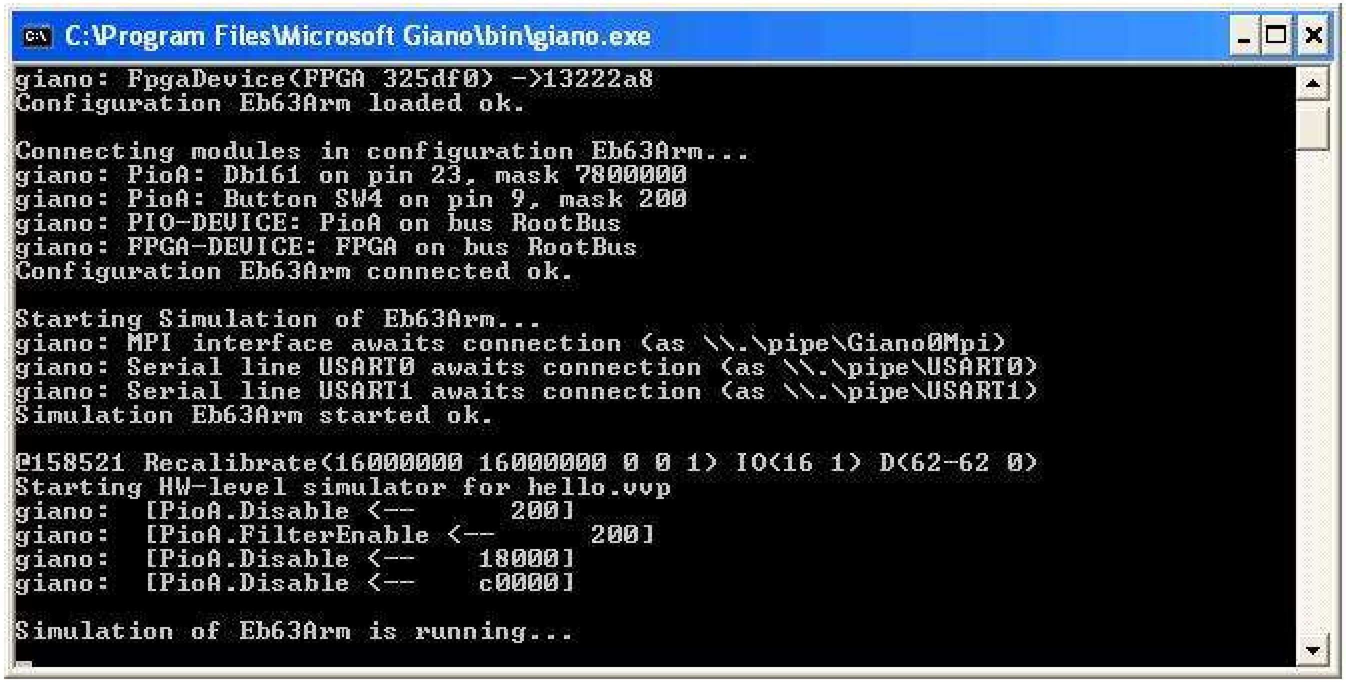
2. Right Click on CPU model and select ‘Create the Simulation Configuration File’. This creates the *.plx configuration file based on the model drawn in Microsoft Visio
3. Right Click on the CPU model and select ‘Start Simulation’. The simulation will start in a command window.

Running Simulations in Giano from Microsoft Visio



CpuType	ARM
ByteOrder	LittleEndian
Implementation	arm
BootAddress	16777216

Running Simulations in Giano from Microsoft Visio



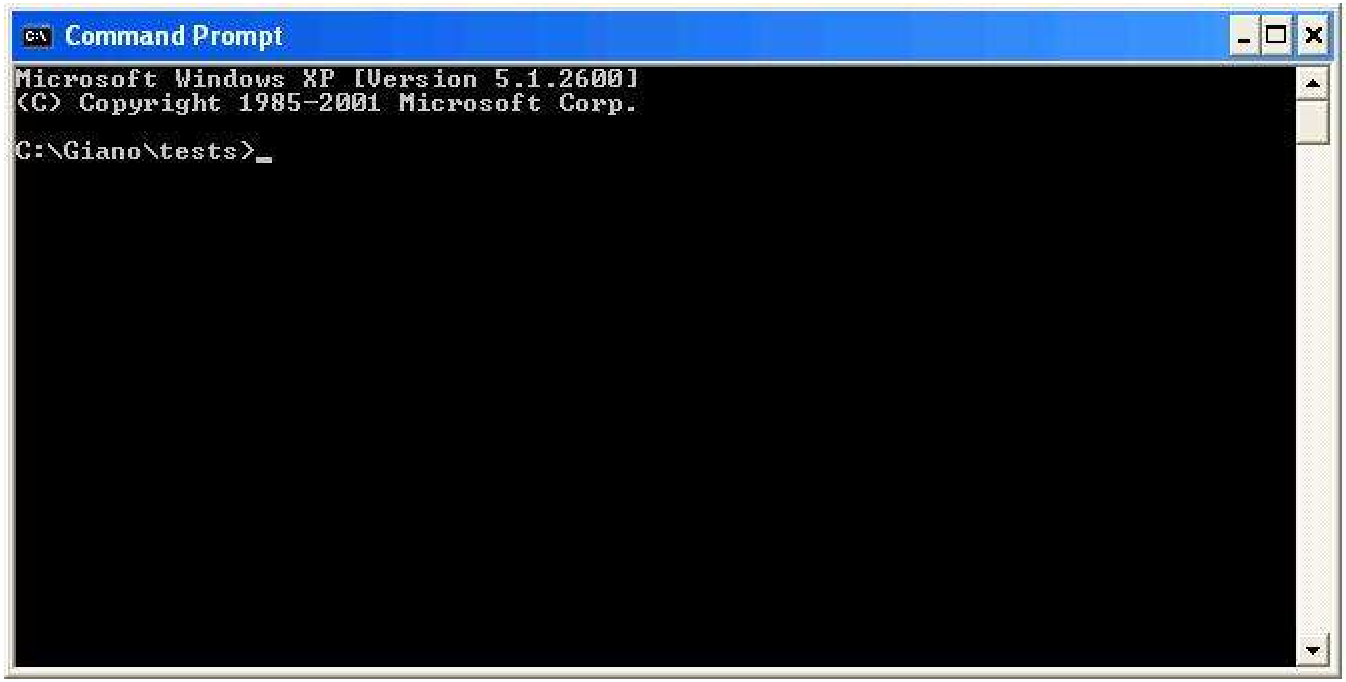
```
giano: FpgaDevice(FPGA 325df0) ->13222a8
Configuration Eb63Arm loaded ok.

Connecting modules in configuration Eb63Arm...
giano: PioA: Db161 on pin 23, mask 7800000
giano: PioA: Button SW4 on pin 9, mask 200
giano: PIO-DEVICE: PioA on bus RootBus
giano: FPGA-DEVICE: FPGA on bus RootBus
Configuration Eb63Arm connected ok.

Starting Simulation of Eb63Arm...
giano: MPI interface awaits connection (as \\.\pipe\Giano0Mpi)
giano: Serial line USART0 awaits connection (as \\.\pipe\USART0)
giano: Serial line USART1 awaits connection (as \\.\pipe\USART1)
Simulation Eb63Arm started ok.

@158521 Recalibrate(16000000 16000000 0 0 1) IO(16 1) D(62-62 0)
Starting HW-level simulator for hello.vvp
giano:  [PioA.Disable <--      200]
giano:  [PioA.FilterEnable <--      200]
giano:  [PioA.Disable <--    18000]
giano:  [PioA.Disable <--    c0000]

Simulation of Eb63Arm is running...
```

Procedure #3

- Installing Giano
- Running Simulations in Giano from Visio
- **Running Simulations from the Command Line**
- Connecting to Simulations using Serplexd
- Simulating EB63 with HostFS
- Creating/modifying a configuration
- Creating a new Giano module

Running Simulations from the Command Line

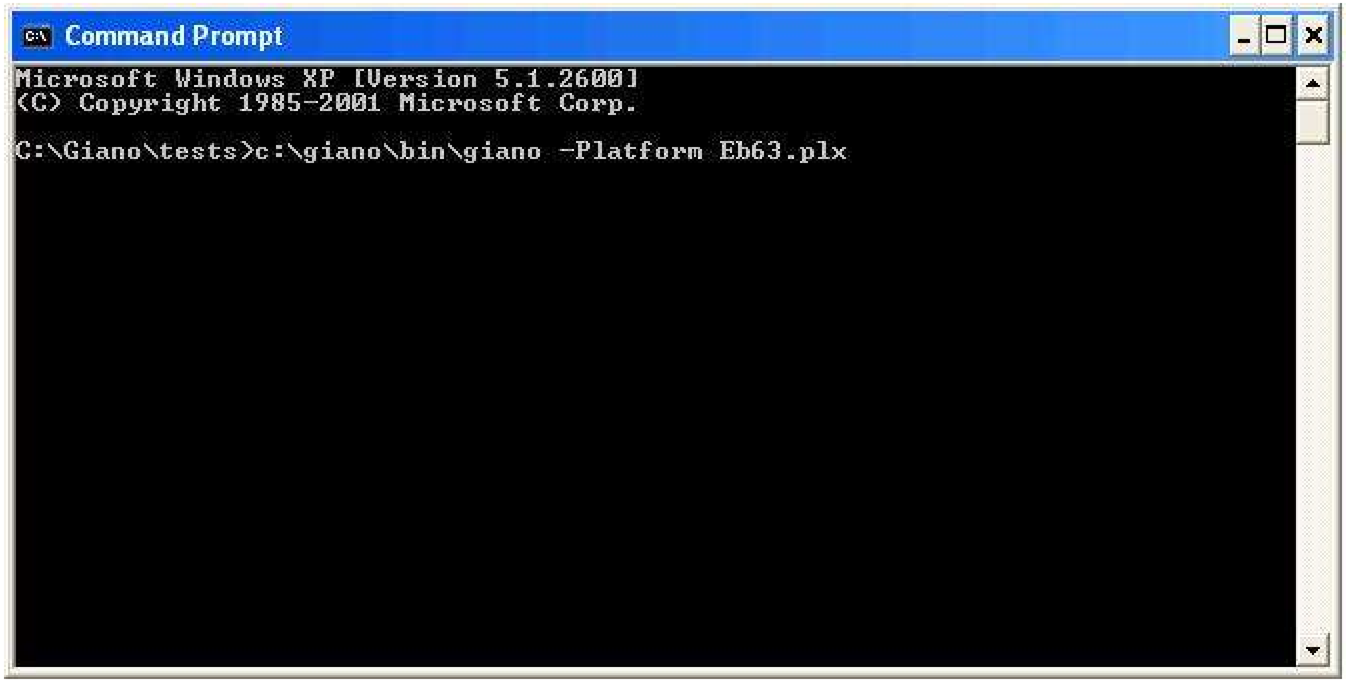
1. Open a Command Prompt to the directory where your *.plx configuration file is located.



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Giano\tests>_
```

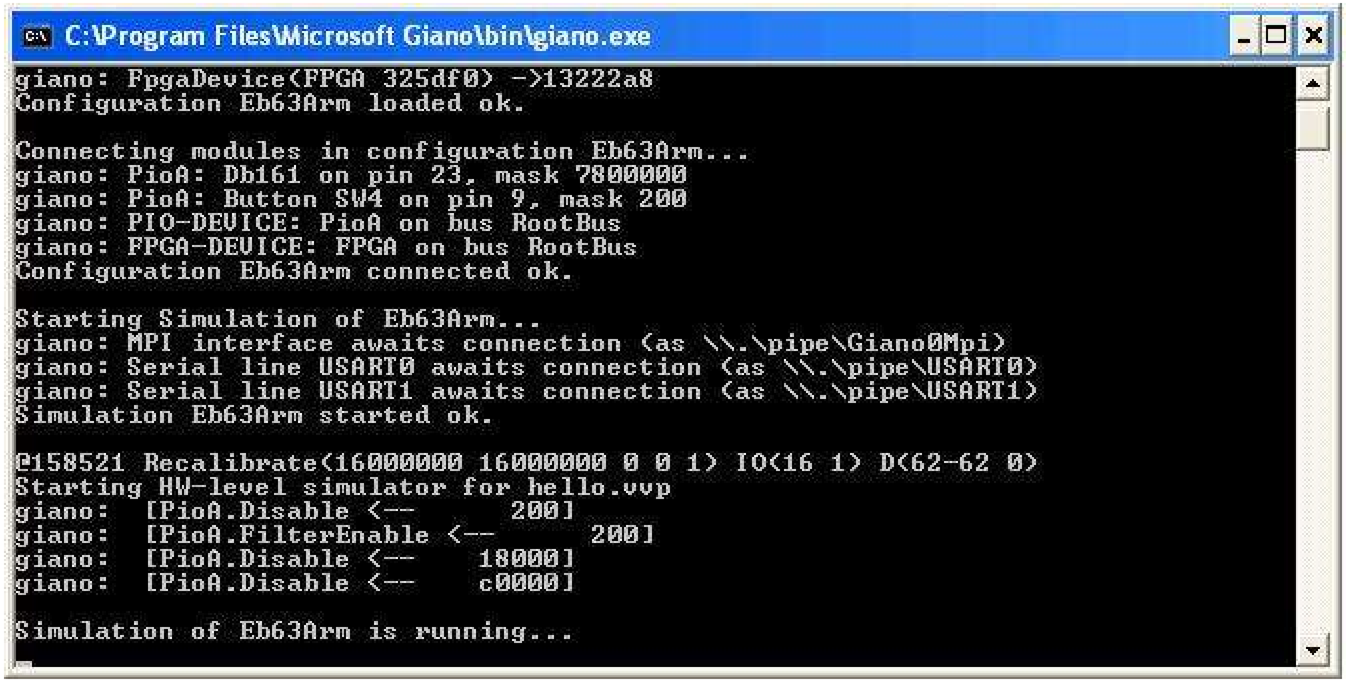
Running Simulations from the Command Line

2. Type “giano.exe –Platform <name of *.plx file>” and press ‘Enter’. The simulation will begin running in this Command Window.



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Giano\tests>c:\giano\bin\giano -Platform Eb63.plx
```

Running Simulations from the Command Line



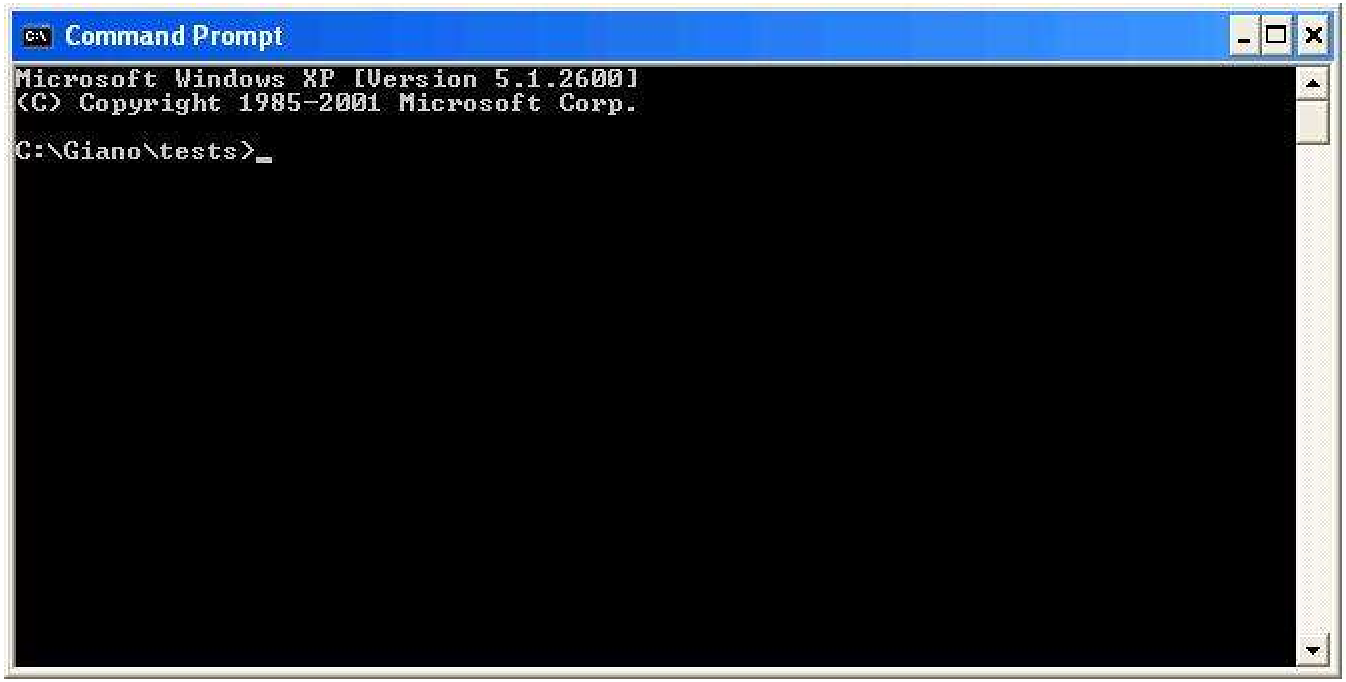
```
giano: FpgaDevice(FPGA 325df0) ->13222a8
Configuration Eb63Arm loaded ok.

Connecting modules in configuration Eb63Arm...
giano: PioA: Db161 on pin 23, mask 7800000
giano: PioA: Button SW4 on pin 9, mask 200
giano: PIO-DEVICE: PioA on bus RootBus
giano: FPGA-DEVICE: FPGA on bus RootBus
Configuration Eb63Arm connected ok.

Starting Simulation of Eb63Arm...
giano: MPI interface awaits connection (as \\.\pipe\Giano0Mpi)
giano: Serial line USART0 awaits connection (as \\.\pipe\USART0)
giano: Serial line USART1 awaits connection (as \\.\pipe\USART1)
Simulation Eb63Arm started ok.

@158521 Recalibrate(16000000 16000000 0 0 1) IO(16 1) D(62-62 0)
Starting HW-level simulator for hello.vvp
giano:  [PioA.Disable <--       200]
giano:  [PioA.FilterEnable <--       200]
giano:  [PioA.Disable <--    18000]
giano:  [PioA.Disable <--    c0000]

Simulation of Eb63Arm is running...
_
```

Procedure #4

- Installing Giano
- Running Simulations in Giano from Visio
- Running Simulations from the Command Line
- **Connecting to Simulations using Serplexd**
- Simulating EB63 with HostFS
- Creating/modifying a configuration
- Creating a new Giano module

Connecting to Simulation using Serplexd

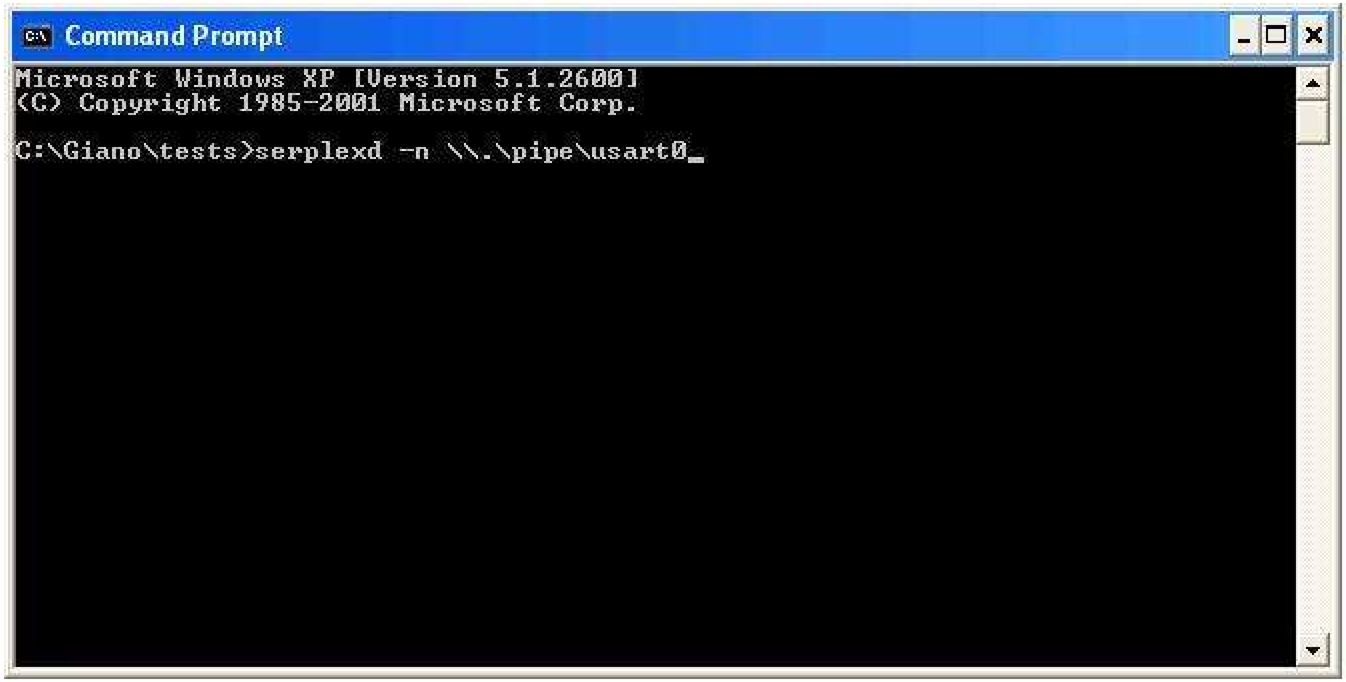
1. Open a Command Prompt to the directory where serplexd.exe is located, e.g. Giano\tests.



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Giano\tests>_
```

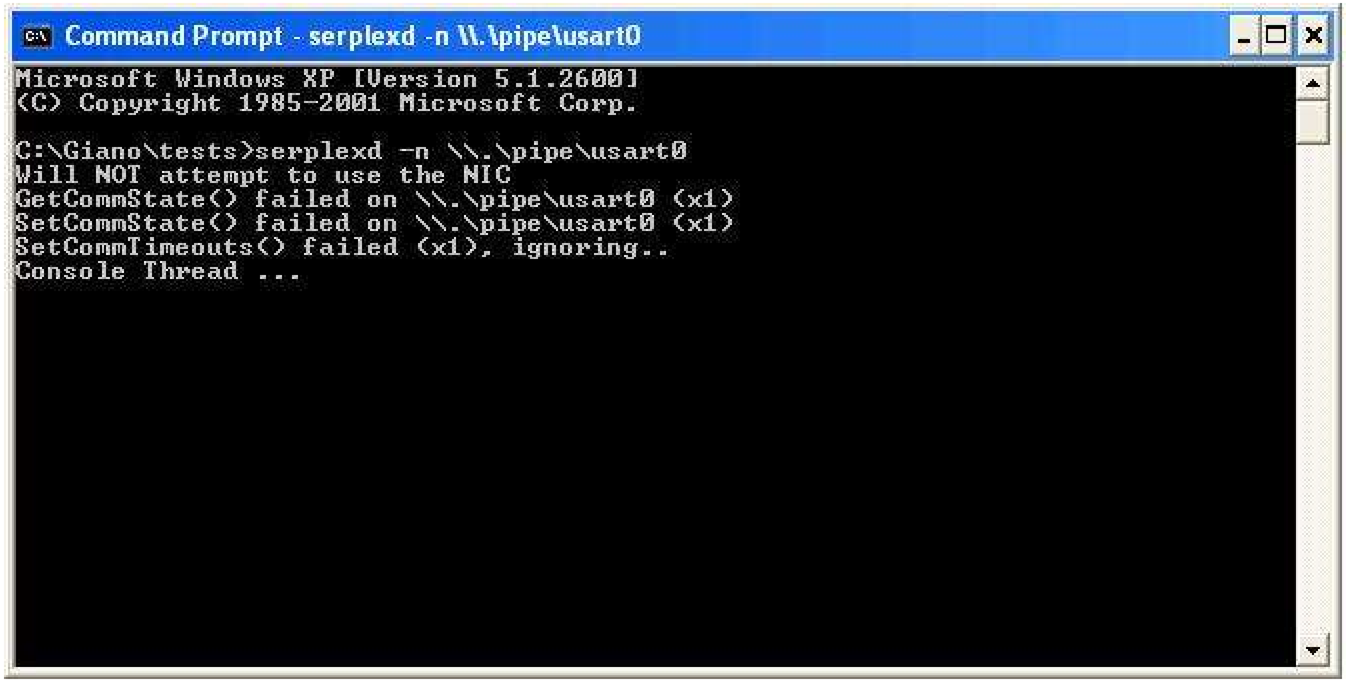
Connecting to Simulation using Serplexd

2. Type ‘serplexd -n \\.\pipe\usart0’ and press ‘Enter’. Serplexd will begin running in this Command Window.



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Giano\tests>serplexd -n \\.\pipe\usart0_
```

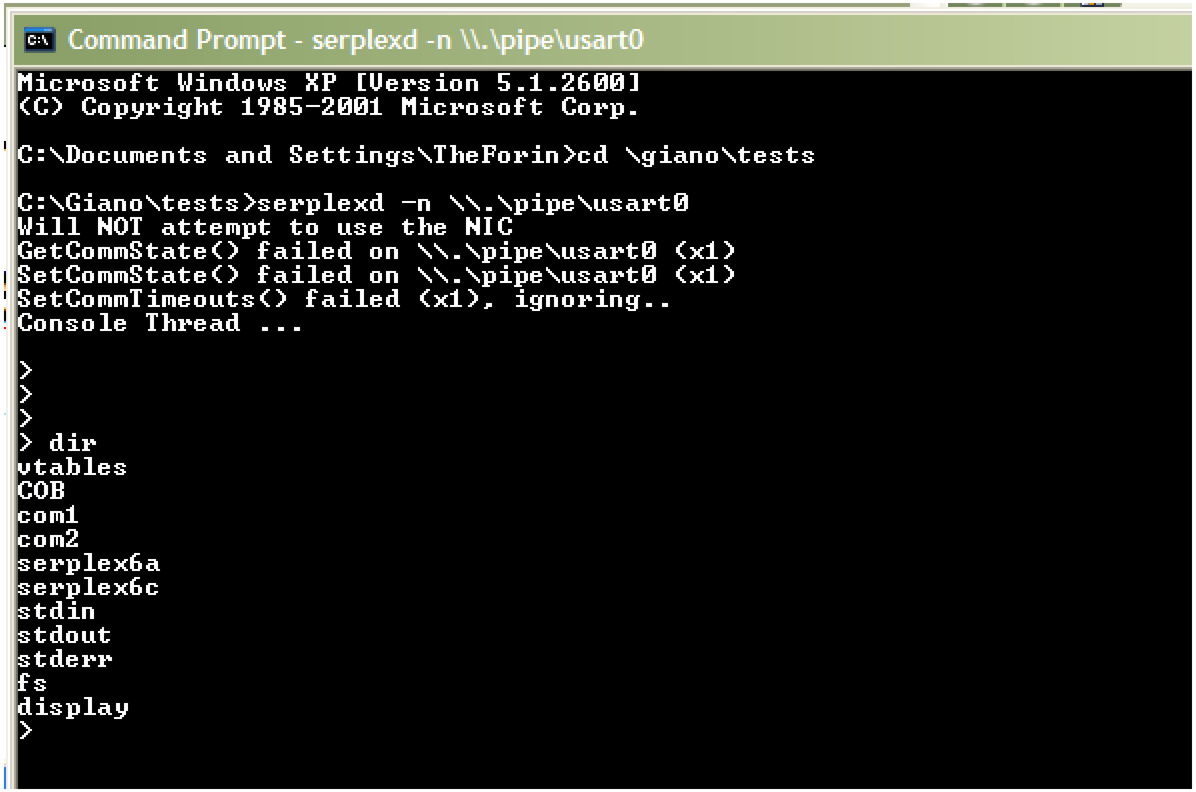
Connecting to Simulation using Serplexd



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Giano\tests>serplexd -n \\.\pipe\usart0
Will NOT attempt to use the NIC
GetCommState() failed on \\.\pipe\usart0 (x1)
SetCommState() failed on \\.\pipe\usart0 (x1)
SetCommTimeouts() failed (x1), ignoring..
Console Thread ...
```

Connecting to Simulation using Serplexd

3. Type something, like “dir”



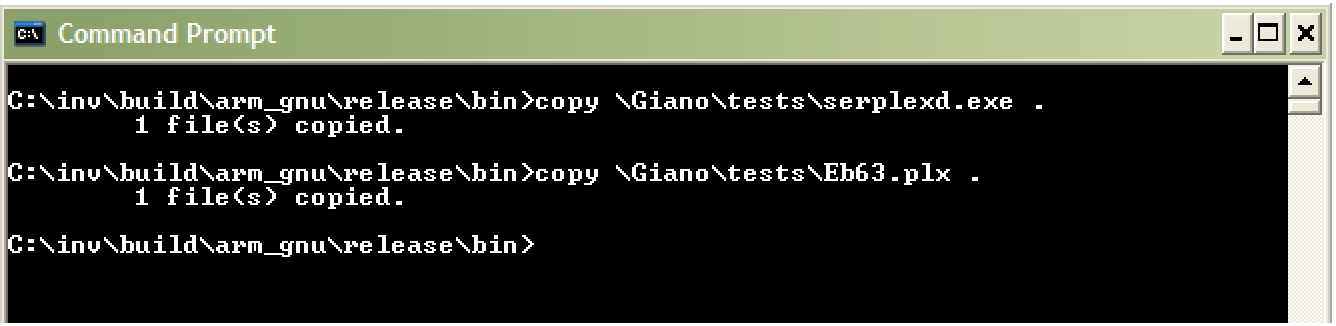
```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\TheForin>cd \giano\tests

C:\Giano\tests>serplexd -n \\.\pipe\usart0
Will NOT attempt to use the NIC
GetCommState() failed on \\.\pipe\usart0 (x1)
SetCommState() failed on \\.\pipe\usart0 (x1)
SetCommTimeouts() failed (x1), ignoring..
Console Thread ...

>
>
>
> dir
vtables
COB
com1
com2
serplex6a
serplex6c
stdin
stdout
stderr
fs
display
>
```

Procedure #5

- Installing Giano
- Running Simulations in Giano from Visio
- Running Simulations from the Command Line
- Connecting to Simulations using Serplexd
- **Simulating EB63 with HostFS**
- Creating/modifying a configuration
- Creating a new Giano module

Simulating EB63 with HostFS

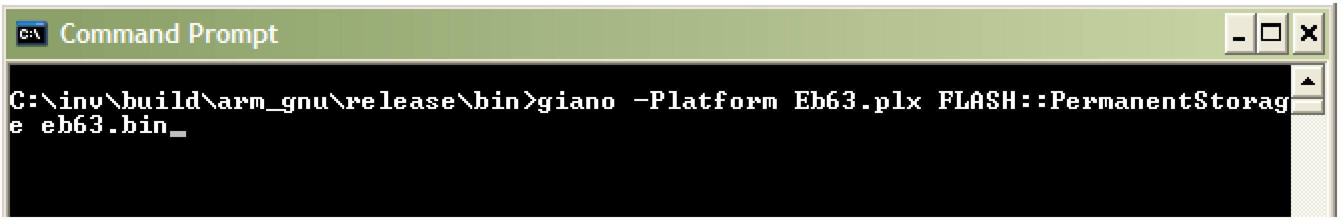
1. Build MMlite (compile source code)
2. Copy serplexd to MMlite build directory bin of the desired compiler (EX. C:\inv\build\arm_gnu\release\bin)
3. Copy the *.plx configuration file to the MMlite build directory



```
C:\inv\build\arm_gnu\release\bin>copy \Giano\tests\serplexd.exe .
        1 file(s) copied.

C:\inv\build\arm_gnu\release\bin>copy \Giano\tests\Eb63.plx .
        1 file(s) copied.

C:\inv\build\arm_gnu\release\bin>
```

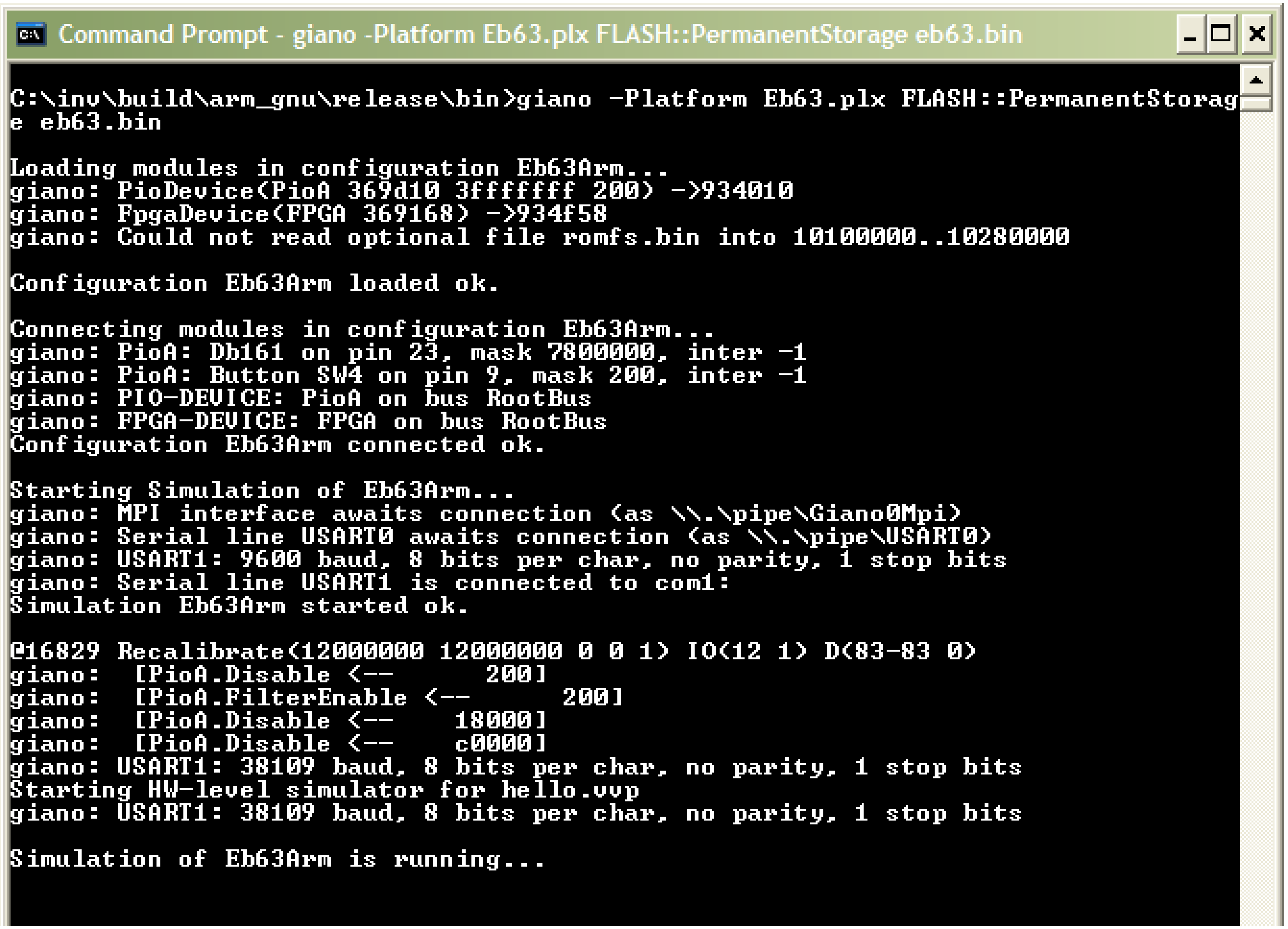
Simulating EB63 with HostFS

4. Run Microsoft Giano in the Command Line with the following in the location of the Eb63.plx file: giano.exe –Platform Eb63.plx FLASH::PermanentStorage eb63.bin



```
C:\inv\build\arm_gnu\release\bin>giano -Platform Eb63.plx FLASH::PermanentStorag
e eb63.bin_
```

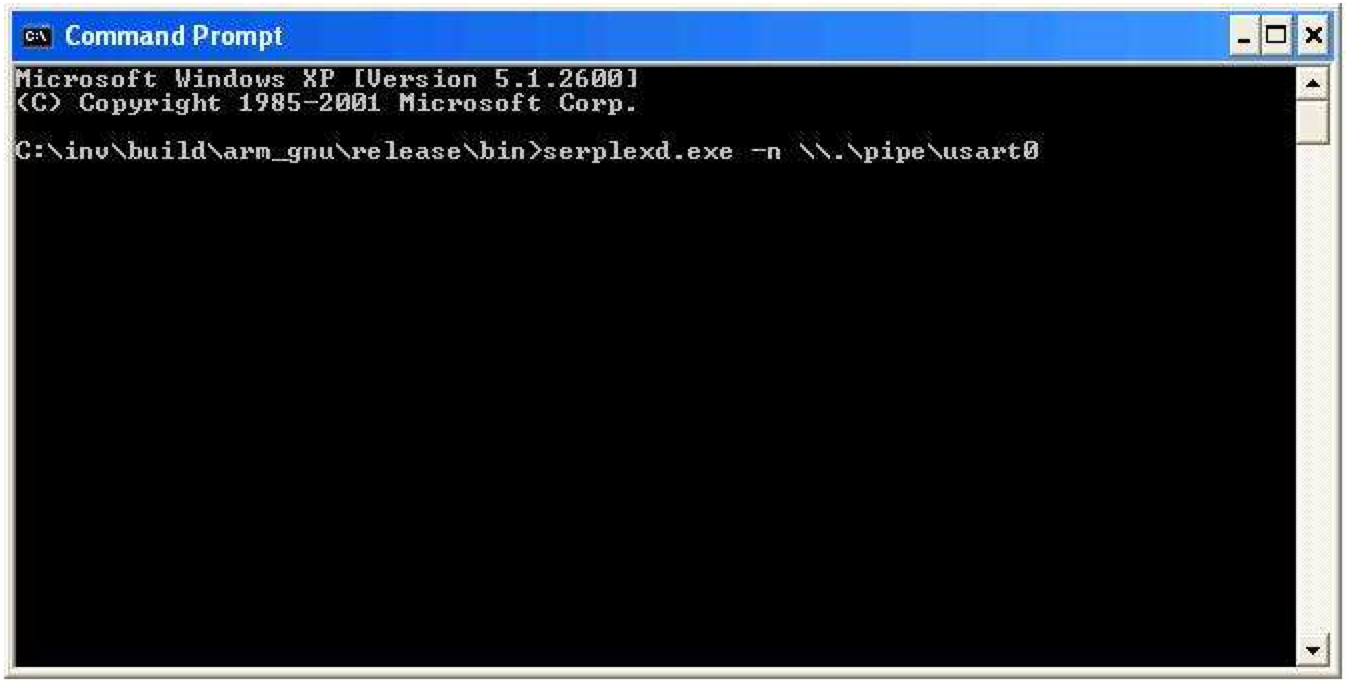
Simulating EB63 with HostFS


Command Prompt - giano -Platform Eb63.plx FLASH::PermanentStorage eb63.bin

```
C:\inv\build\arm_gnu\release\bin>giano -Platform Eb63.plx FLASH::PermanentStorag
e eb63.bin

Loading modules in configuration Eb63Arm...
giano: PioDevice(PioA 369d10 3ffffff 200) ->934010
giano: FpgaDevice(FPGA 369168) ->934f58
giano: Could not read optional file romfs.bin into 10100000..10280000

Configuration Eb63Arm loaded ok.

Connecting modules in configuration Eb63Arm...
giano: PioA: Db161 on pin 23, mask 7800000, inter -1
giano: PioA: Button SW4 on pin 9, mask 200, inter -1
giano: PIO-DEVICE: PioA on bus RootBus
giano: FPGA-DEVICE: FPGA on bus RootBus
Configuration Eb63Arm connected ok.

Starting Simulation of Eb63Arm...
giano: MPI interface awaits connection (as \\.\pipe\Giano0Mpi)
giano: Serial line USART0 awaits connection (as \\.\pipe\USART0)
giano: USART1: 9600 baud, 8 bits per char, no parity, 1 stop bits
giano: Serial line USART1 is connected to com1:
Simulation Eb63Arm started ok.

@16829 Recalibrate(12000000 12000000 0 0 1) IO(12 1) D(83-83 0)
giano:  [PioA.Disable <--      200]
giano:  [PioA.FilterEnable <--      200]
giano:  [PioA.Disable <--    18000]
giano:  [PioA.Disable <--    c0000]
giano: USART1: 38109 baud, 8 bits per char, no parity, 1 stop bits
Starting HW-level simulator for hello.vvp
giano: USART1: 38109 baud, 8 bits per char, no parity, 1 stop bits

Simulation of Eb63Arm is running...
```

Simulating EB63 with HostFS

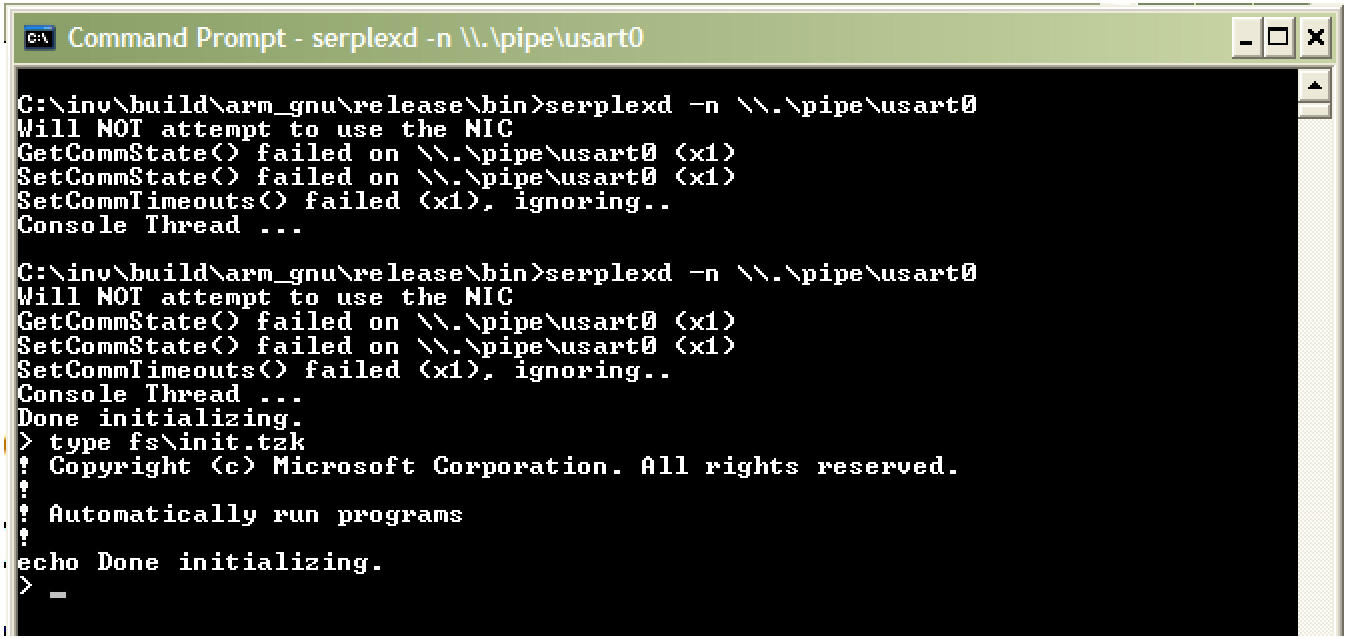
5. Run serplexd.exe from the MMlite build directory bin.



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\inv\build\arm_gnu\release\bin>serplexd.exe -n \\.\pipe\usart0
```

Simulating EB63 with HostFS

6. After a short time you should see the ‘Done Prompt’ to inform you that it has completed loading the first program from HostFS



```
C:\inv\build\arm_gnu\release\bin>serplexd -n \\.\pipe\usart0
Will NOT attempt to use the NIC
GetCommState() failed on \\.\pipe\usart0 (x1)
SetCommState() failed on \\.\pipe\usart0 (x1)
SetCommTimeouts() failed (x1), ignoring..
Console Thread ...

C:\inv\build\arm_gnu\release\bin>serplexd -n \\.\pipe\usart0
Will NOT attempt to use the NIC
GetCommState() failed on \\.\pipe\usart0 (x1)
SetCommState() failed on \\.\pipe\usart0 (x1)
SetCommTimeouts() failed (x1), ignoring..
Console Thread ...
Done initializing.
> type fs\init.tzk
! Copyright (c) Microsoft Corporation. All rights reserved.
!
! Automatically run programs
!
echo Done initializing.
> _
```

Procedure #6

- Installing Giano
- Running Simulations in Giano from Visio
- Running Simulations from the Command Line
- Connecting to Simulations using Serplexd
- Simulating EB63 with HostFS
- **Creating/modifying a configuration**
- Creating a new Giano module

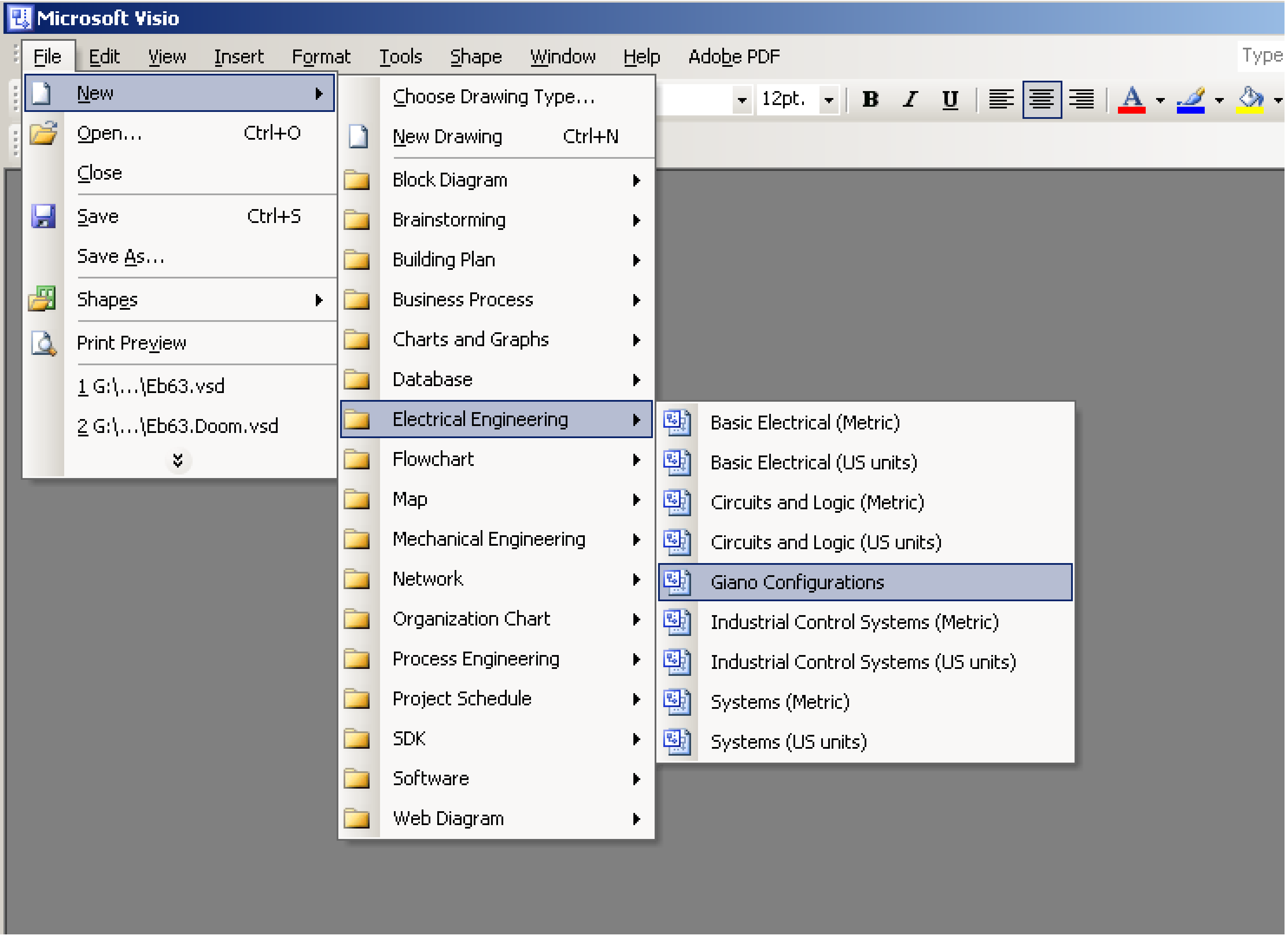
Creating/modifying a Configuration

1- Start Visio and select File::New::Electrical Engineering::Giano Configurations or..

1a- Open an existing .VSD configuration

Note: the .plx file is just XML, you can edit it directly but Visio is much easier

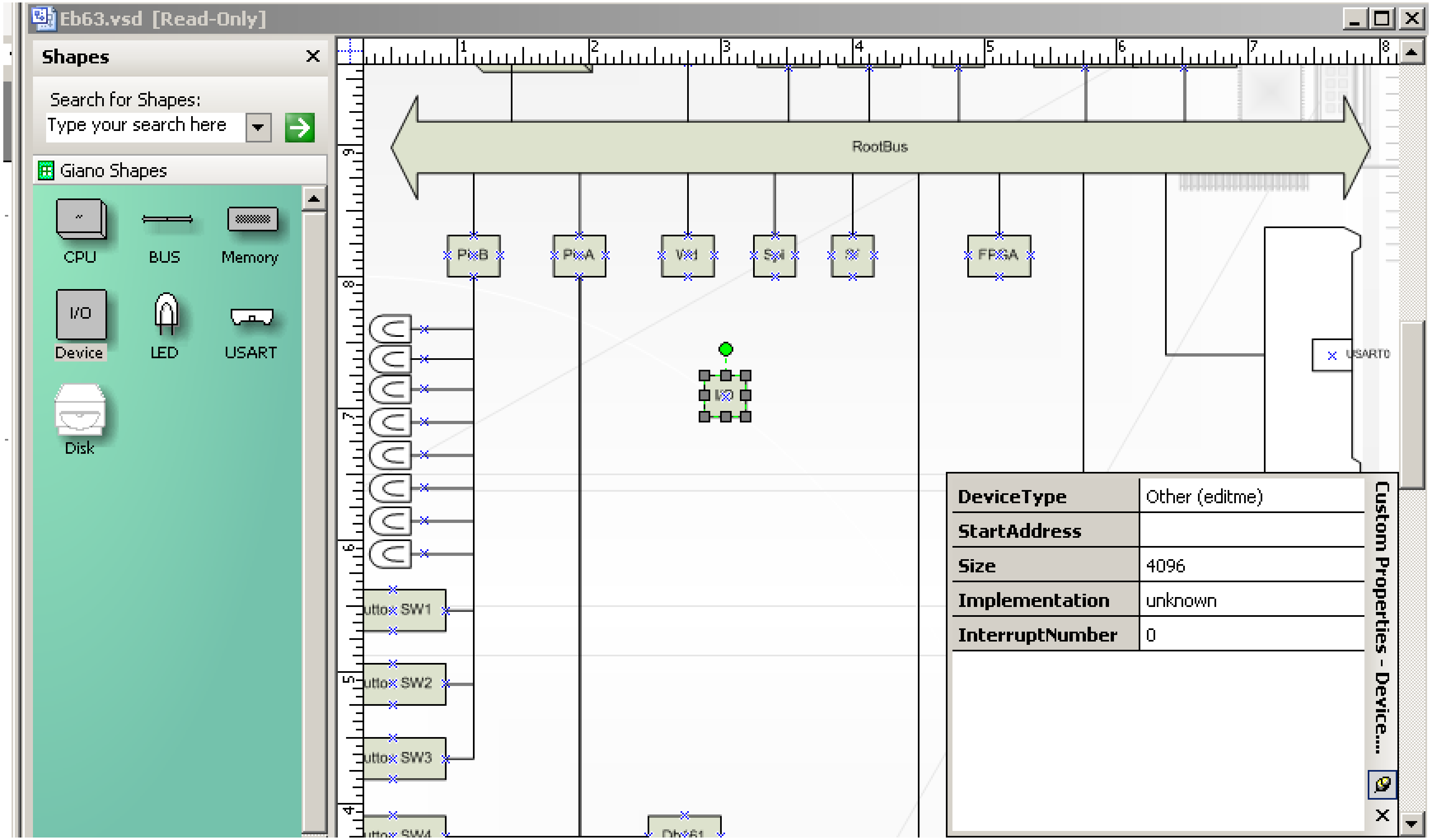
Creating/modifying a Configuration



Creating/modifying a Configuration

2- Drag&drop new objects from “Giano Shapes” to your configuration

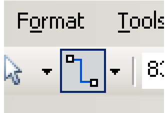
Creating/modifying a Configuration



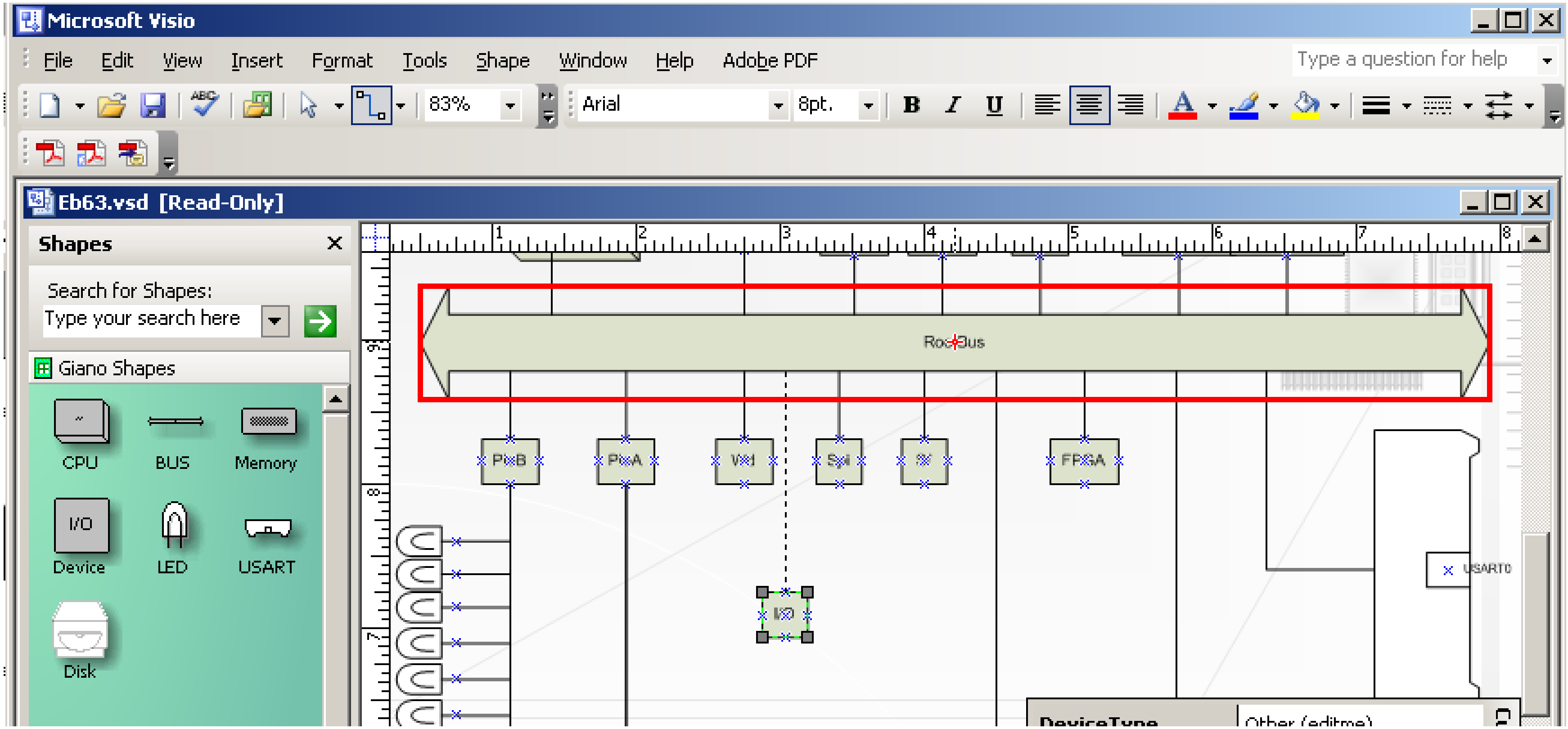
DeviceType	Other (editme)
StartAddress	
Size	4096
Implementation	unknown
InterruptNumber	0

Creating/modifying a Configuration

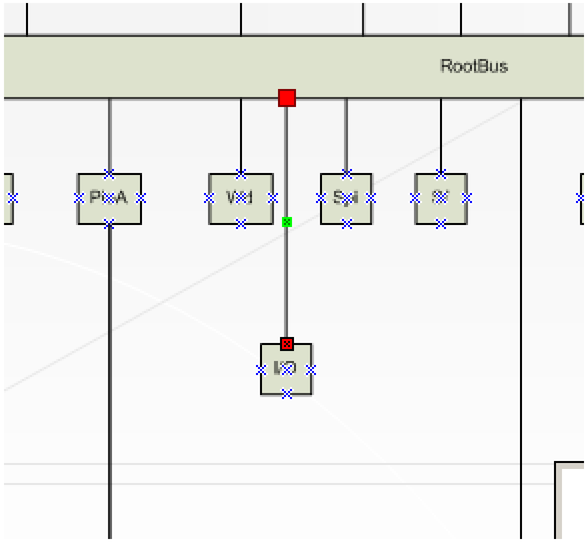
3- Connect objects with the “Connector Tool”:



Creating/modifying a Configuration



Creating/modifying a Configuration

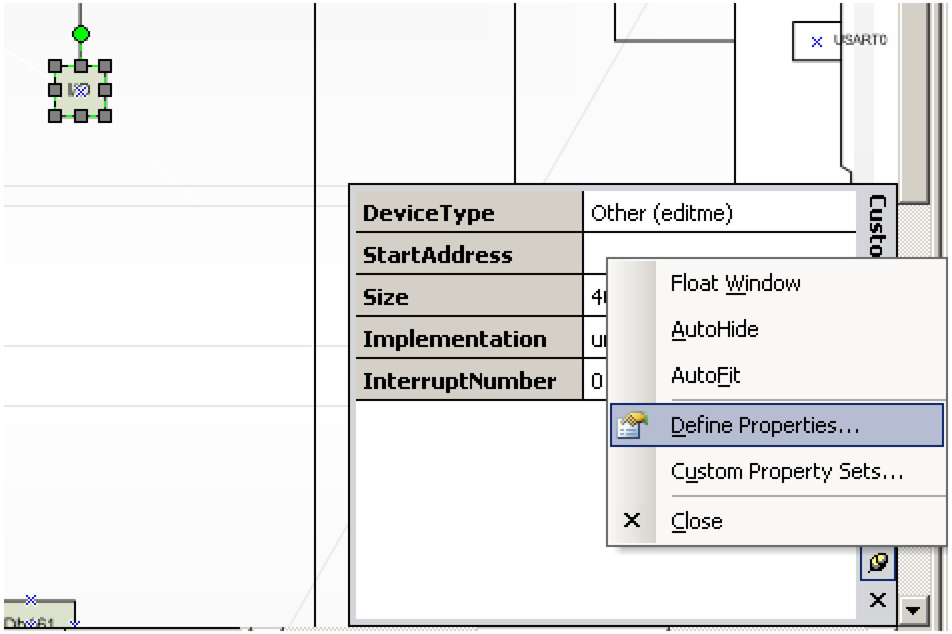


Creating/modifying a Configuration

4- Edit an object properties by selecting it and working on the “Custom Properties”

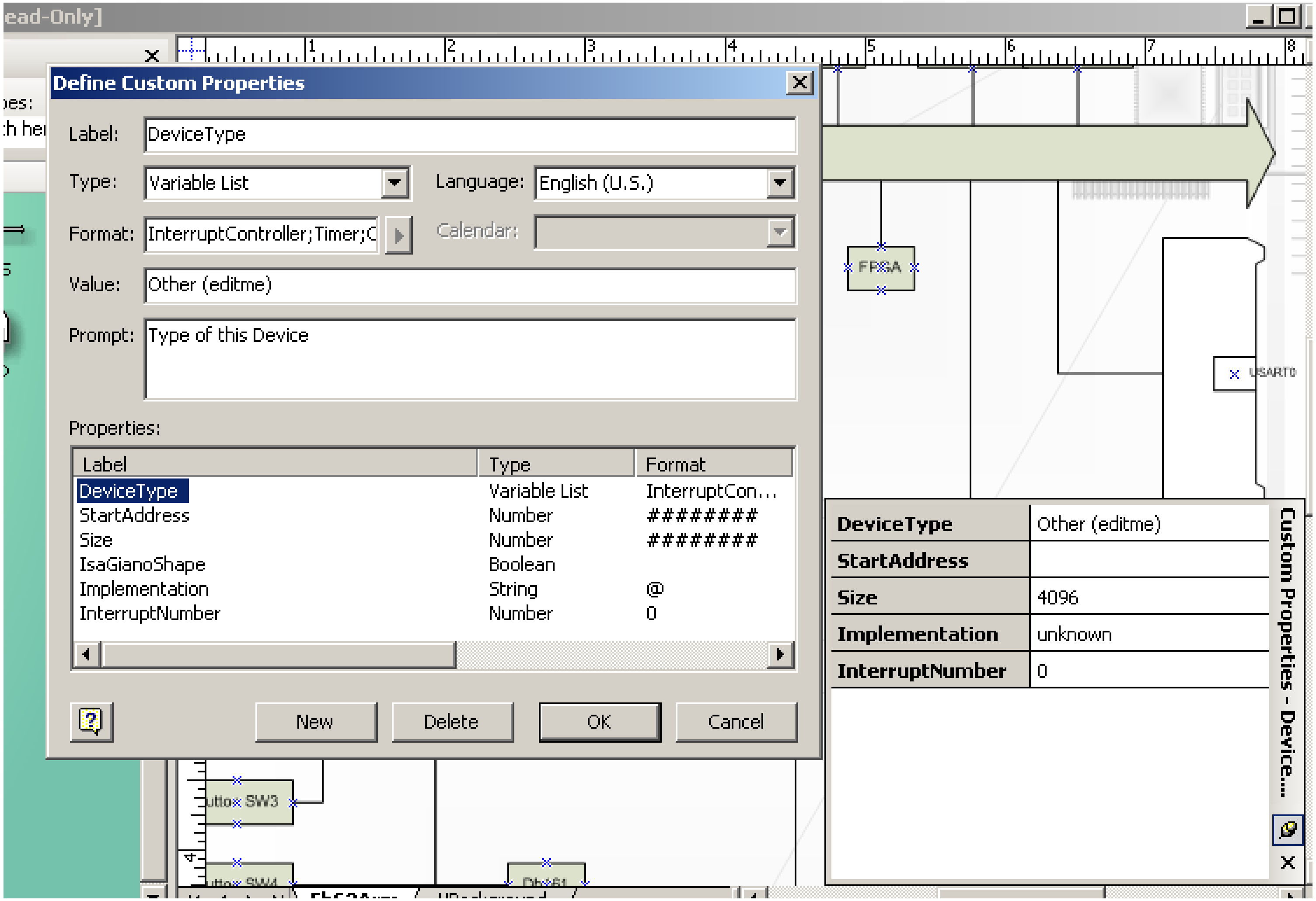
5- Add/Remove properties by right-clicking on the Custom Properties and selecting “Define Properties”

Creating/modifying a Configuration



DeviceType	Other (editme)
StartAddress	
Size	4(
Implementation	ui
InterruptNumber	0

Creating/modifying a Configuration



Label: DeviceType

Type: Variable List

Language: English (U.S.)

Format: InterruptController;Timer;C

Calendar:

Value: Other (editme)

Prompt: Type of this Device

Properties:

Label	Type	Format
DeviceType	Variable List	InterruptCon...
StartAddress	Number	########
Size	Number	########
IsaGianoShape	Boolean	
Implementation	String	@
InterruptNumber	Number	0

DeviceType	Other (editme)
StartAddress	
Size	4096
Implementation	unknown
InterruptNumber	0

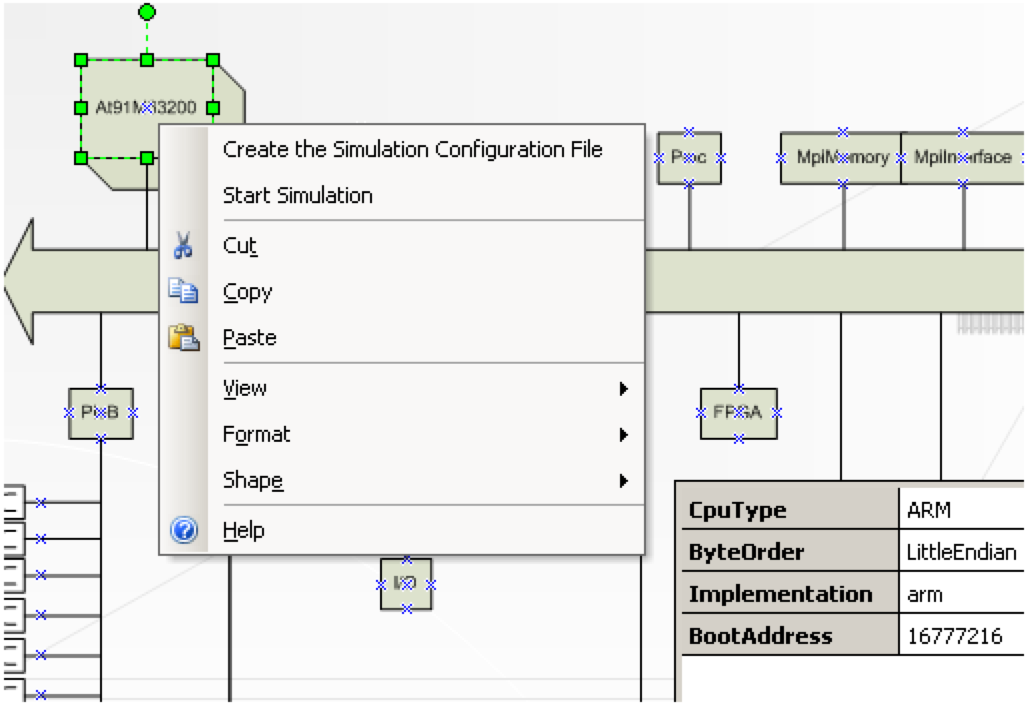
Creating/modifying a Configuration

6- Save your new configuration with File::SaveAs

7- Create your .PLX file by right-clicking on a CPU module and selecting “Create the Simulation Configuration File”

8- Run your new configuration

Creating/modifying a Configuration



CpuType	ARM
ByteOrder	LittleEndian
Implementation	arm
BootAddress	16777216

Procedure #7

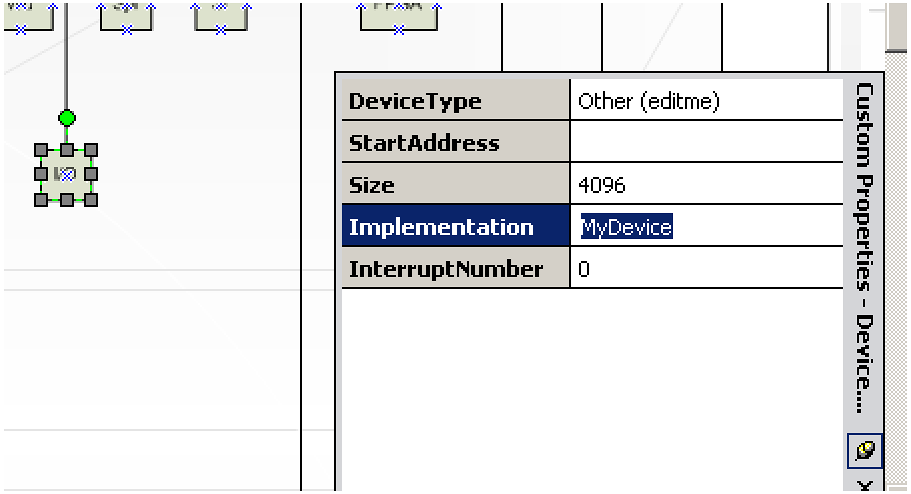
- Installing Giano
- Running Simulations in Giano from Visio
- Running Simulations from the Command Line
- Connecting to Simulations using Serplexd
- Simulating EB63 with HostFS
- Creating/modifying a configuration
- **Creating a new Giano module**

Creating a new Giano module

1- Create a suitable new object in your configuration and change the “Implementation” property to the name of your new device, such as “MyDevice”

Note: We will create a new Device object. Busses, CPUs and memories are similar

Creating a new Giano module



DeviceType	Other (editme)
StartAddress	
Size	4096
Implementation	MyDevice
InterruptNumber	0

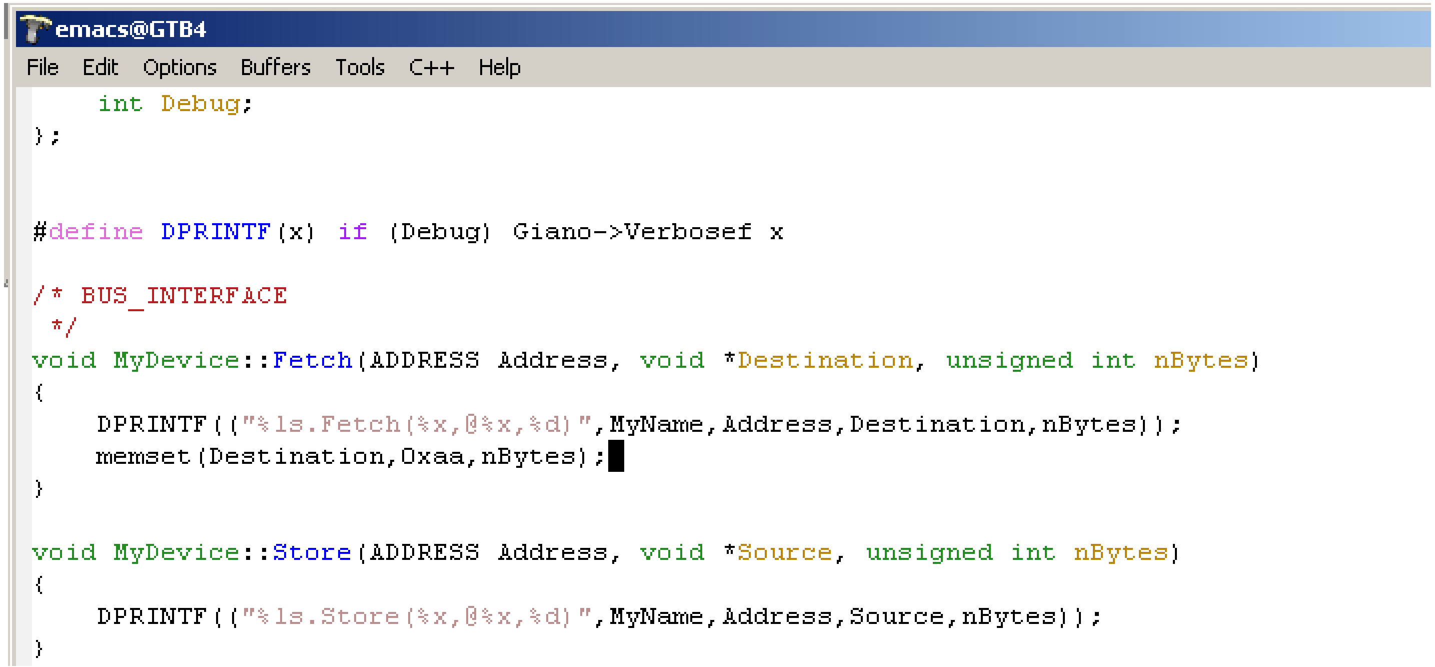
Creating a new Giano module

2- Go to the Peripherals folder and copy the template file Unknown.cpp to MyDevice.cpp

3- Edit the file as needed to implement your functionality

Note: To debug you should build Giano for debugging with “nmake clean debug”

Creating a new Giano module

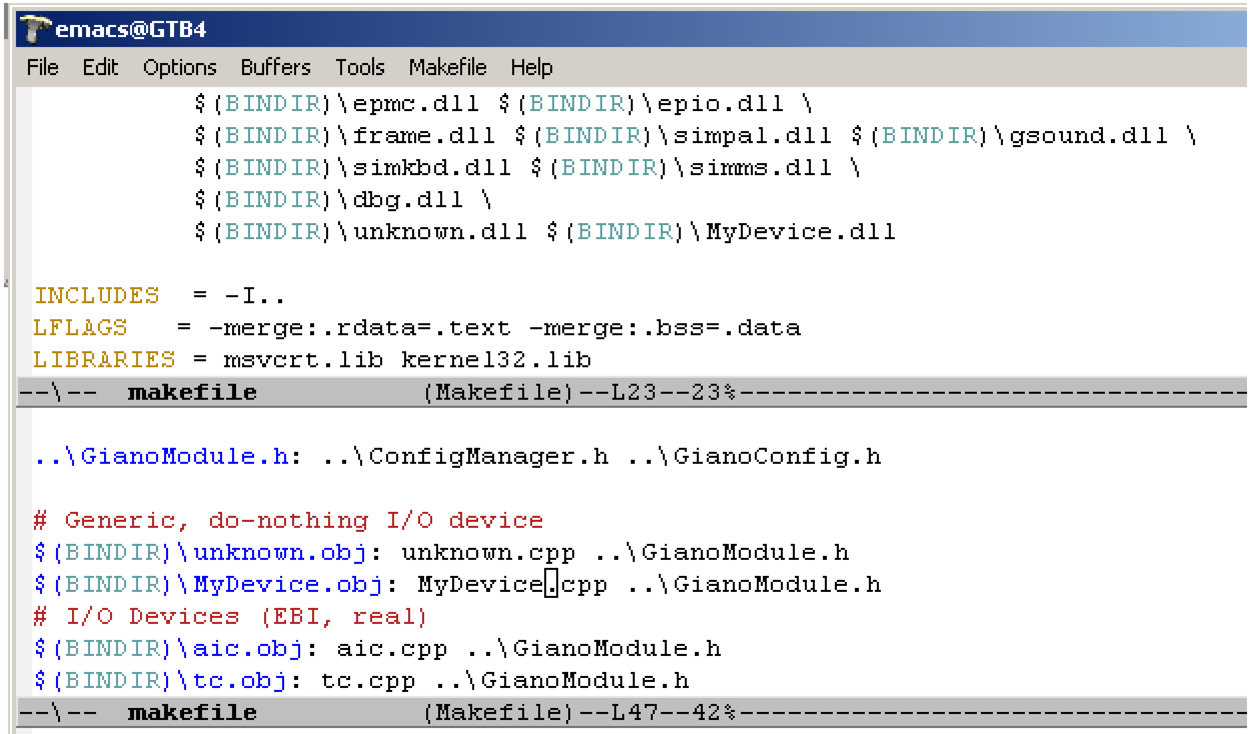


```
    int Debug;
};


#define DPRINTF(x) if (Debug) Giano->Verbosef x

/* BUS_INTERFACE
 */
void MyDevice::Fetch(ADDRESS Address, void *Destination, unsigned int nBytes)
{
    DPRINTF(("%ls.Fetch(%x,@%x,%d)",MyName,Address,Destination,nBytes));
    memset(Destination,0xaa,nBytes);
}

void MyDevice::Store(ADDRESS Address, void *Source, unsigned int nBytes)
{
    DPRINTF(("%ls.Store(%x,@%x,%d)",MyName,Address,Source,nBytes));
}
```

Creating a new Giano module

4- Edit the Peripherals\makefile to build your module along with the others

Creating a new Giano module

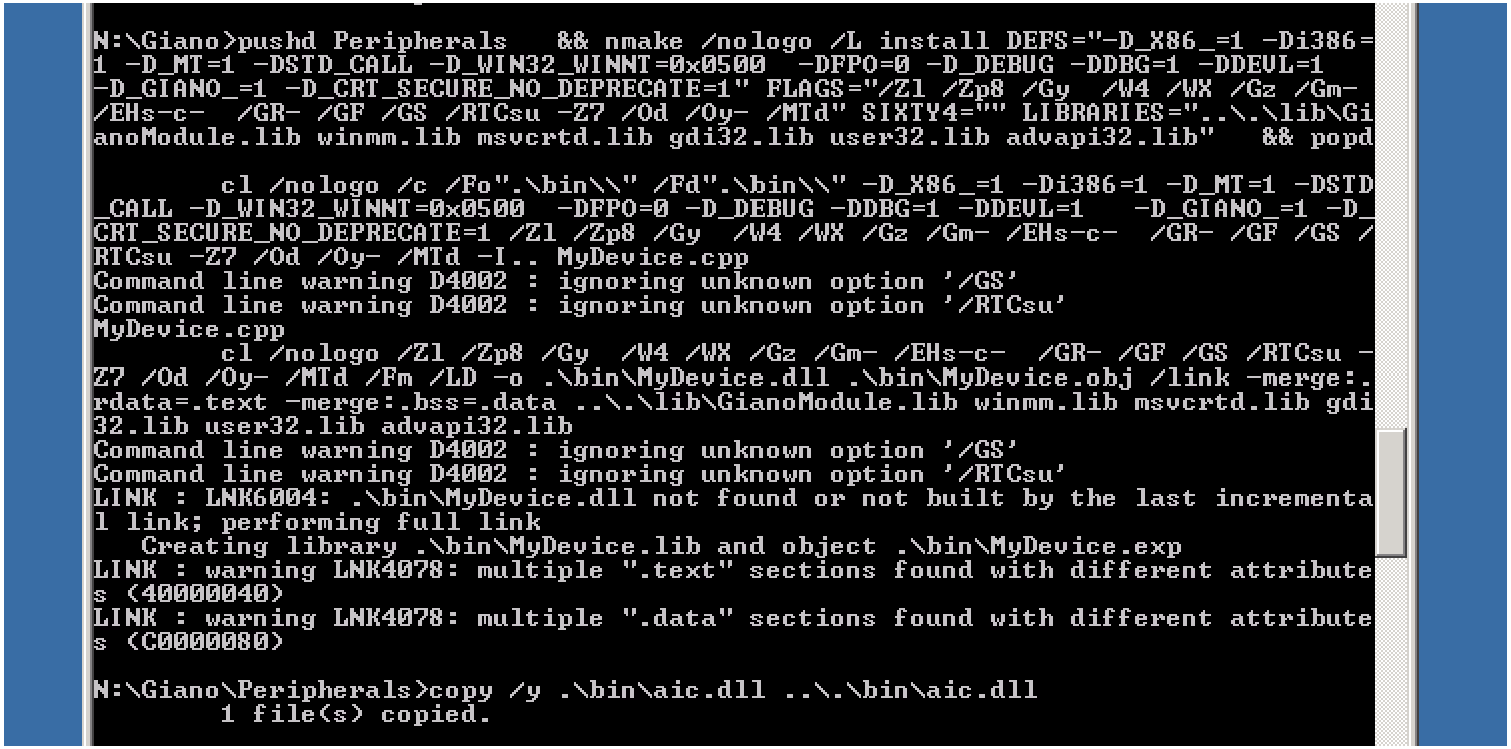


```
          $(BINDIR)\epmc.dll $(BINDIR)\epio.dll \
          $(BINDIR)\frame.dll $(BINDIR)\simpal.dll $(BINDIR)\gsound.dll \
          $(BINDIR)\simkbd.dll $(BINDIR)\simms.dll \
          $(BINDIR)\dbg.dll \
          $(BINDIR)\unknown.dll $(BINDIR)\MyDevice.dll

INCLUDES  = -I..
LFLAGS    = -merge:.rdata=.text -merge:.bss=.data
LIBRARIES = msvcrt.lib kernel32.lib
--\--  makefile          (Makefile)--L23--23%----------------------------------

..\GianoModule.h: ..\ConfigManager.h ..\GianoConfig.h

# Generic, do-nothing I/O device
$(BINDIR)\unknown.obj: unknown.cpp ..\GianoModule.h
$(BINDIR)\MyDevice.obj: MyDevice.cpp ..\GianoModule.h
# I/O Devices (EBI, real)
$(BINDIR)\aic.obj: aic.cpp ..\GianoModule.h
$(BINDIR)\tc.obj: tc.cpp ..\GianoModule.h
--\--  makefile          (Makefile)--L47--42%----------------------------------
```

Creating a new Giano module

5- Rebuild from the top with “nmake debug”

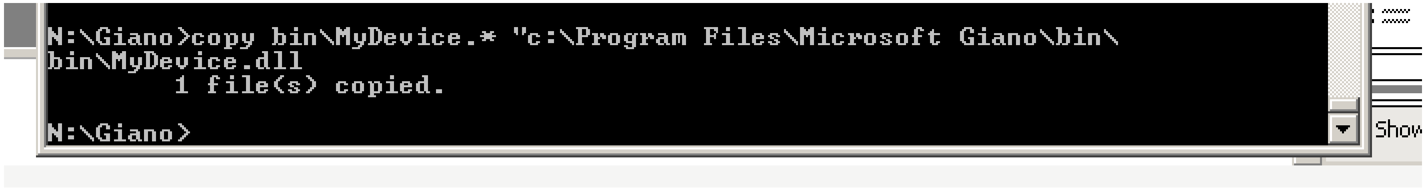
Creating a new Giano module



```
N:\Giano>pushd Peripherals   && nmake /nologo /L install DEFS="-D_X86_=1 -Di386=
1 -D_MT=1 -DSTD_CALL -D_WIN32_WINNT=0x0500  -DFPO=0 -D_DEBUG -DDBG=1 -DDEVL=1
-D_GIANO_=1 -D_CRT_SECURE_NO_DEPRECATE=1" FLAGS="/Zl /Zp8 /Gy  /W4 /WX /Gz /Gm-
/EHs-c-  /GR- /GF /GS /RTCsu -Z7 /Od /Oy- /MTd" SIXTY4="" LIBRARIES="..\.\lib\Gi
anoModule.lib winmm.lib msvcrtd.lib gdi32.lib user32.lib advapi32.lib"   && popd

        cl /nologo /c /Fo".\bin\\" /Fd".\bin\\" -D_X86_=1 -Di386=1 -D_MT=1 -DSTD
_CALL -D_WIN32_WINNT=0x0500  -DFPO=0 -D_DEBUG -DDBG=1 -DDEVL=1   -D_GIANO_=1 -D_
CRT_SECURE_NO_DEPRECATE=1 /Zl /Zp8 /Gy  /W4 /WX /Gz /Gm- /EHs-c-  /GR- /GF /GS /
RTCsu -Z7 /Od /Oy- /MTd -I.. MyDevice.cpp
Command line warning D4002 : ignoring unknown option '/GS'
Command line warning D4002 : ignoring unknown option '/RTCsu'
MyDevice.cpp
        cl /nologo /Zl /Zp8 /Gy  /W4 /WX /Gz /Gm- /EHs-c-  /GR- /GF /GS /RTCsu -
Z7 /Od /Oy- /MTd /Fm /LD -o .\bin\MyDevice.dll .\bin\MyDevice.obj /link -merge:.
rdata=.text -merge:.bss=.data ..\.\lib\GianoModule.lib winmm.lib msvcrtd.lib gdi
32.lib user32.lib advapi32.lib
Command line warning D4002 : ignoring unknown option '/GS'
Command line warning D4002 : ignoring unknown option '/RTCsu'
LINK : LNK6004: .\bin\MyDevice.dll not found or not built by the last incrementa
l link; performing full link
   Creating library .\bin\MyDevice.lib and object .\bin\MyDevice.exp
LINK : warning LNK4078: multiple ".text" sections found with different attribute
s (40000040)
LINK : warning LNK4078: multiple ".data" sections found with different attribute
s (C0000080)

N:\Giano\Peripherals>copy /y .\bin\aic.dll ..\.\bin\aic.dll
        1 file(s) copied.
```

Creating a new Giano module

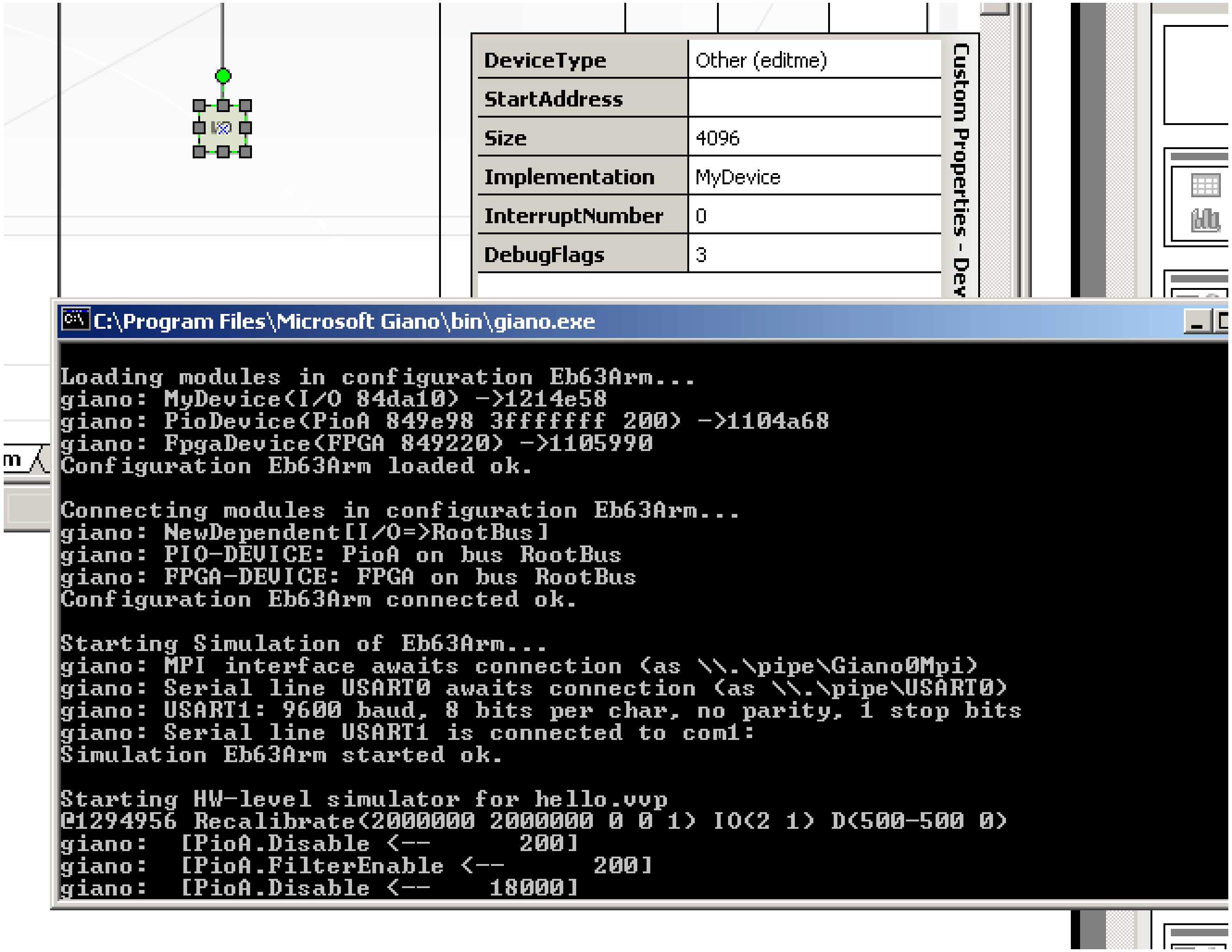
6- Copy your module in the installation directory



```
N:\Giano>copy bin\MyDevice.* "c:\Program Files\Microsoft Giano\bin\
bin\MyDevice.dll
        1 file(s) copied.

N:\Giano>
```

7- Add the “DebugFlags” property to your device and try out your configuration

Creating a new Giano module



DeviceType	Other (editme)
StartAddress	
Size	4096
Implementation	MyDevice
InterruptNumber	0
DebugFlags	3

C:\Program Files\Microsoft Giano\bin\giano.exe

```
Loading modules in configuration Eb63Arm...
giano: MyDevice(I/O 84da10) ->1214e58
giano: PioDevice(PioA 849e98 3fffffff 200) ->1104a68
giano: FpgaDevice(FPGA 849220) ->1105990
Configuration Eb63Arm loaded ok.

Connecting modules in configuration Eb63Arm...
giano: NewDependent[I/O=>RootBus]
giano: PIO-DEVICE: PioA on bus RootBus
giano: FPGA-DEVICE: FPGA on bus RootBus
Configuration Eb63Arm connected ok.

Starting Simulation of Eb63Arm...
giano: MPI interface awaits connection (as \\.\pipe\Giano0Mpi)
giano: Serial line USART0 awaits connection (as \\.\pipe\USART0)
giano: USART1: 9600 baud, 8 bits per char, no parity, 1 stop bits
giano: Serial line USART1 is connected to com1:
Simulation Eb63Arm started ok.

Starting HW-level simulator for hello.vvp
@1294956 Recalibrate(2000000 2000000 0 0 1) IO(2 1) D(500-500 0)
giano:  [PioA.Disable <--      200]
giano:  [PioA.FilterEnable <--      200]
giano:  [PioA.Disable <--    18000]
```