

# **traffiController<sup>4SpW</sup>**

## **Technical Presentation**

**14th SpW Working Group Meeting,  
February 23rd, 2010.**

**Stéphane Davy, Matthieu Salanave**  
[spacewire@skylab-corporate.com](mailto:spacewire@skylab-corporate.com)

**Dr Slim Ben Saoud**  
[www.asic-international.com](http://www.asic-international.com)



# Outline

- 1 Introduction
- 2 TrafficController<sup>4SpW</sup> DVS
- 3 TrafficController<sup>4SpW</sup> API
- 4 TrafficController<sup>4SpW</sup> GUI
- 5 Conclusion and questions



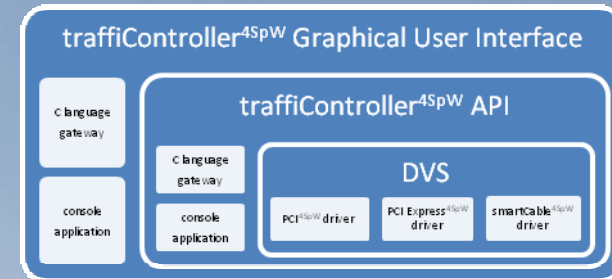
# Outline

- 1 Introduction
- 2 TrafficController<sup>4SpW</sup> DVS
- 3 TrafficController<sup>4SpW</sup> API
- 4 TrafficController<sup>4SpW</sup> GUI
- 5 Conclusion and questions



### 3 Java software layers, with a gateway to C language

- Devices Virtualization Service
- Application Programming Interface
- Graphical User Interface



### Multiple functions in one single software package

- SKYLAB test hardware management, data communication and control
- Integration and test, simulation and network dimensioning thanks to:
  - virtual routers,
  - RMAP support,
  - IP/Ethernet tunnels,
  - real, emulated or virtual nodes,
  - software implementation of SpW link analyzers,
  - communication statistics,
  - etc...

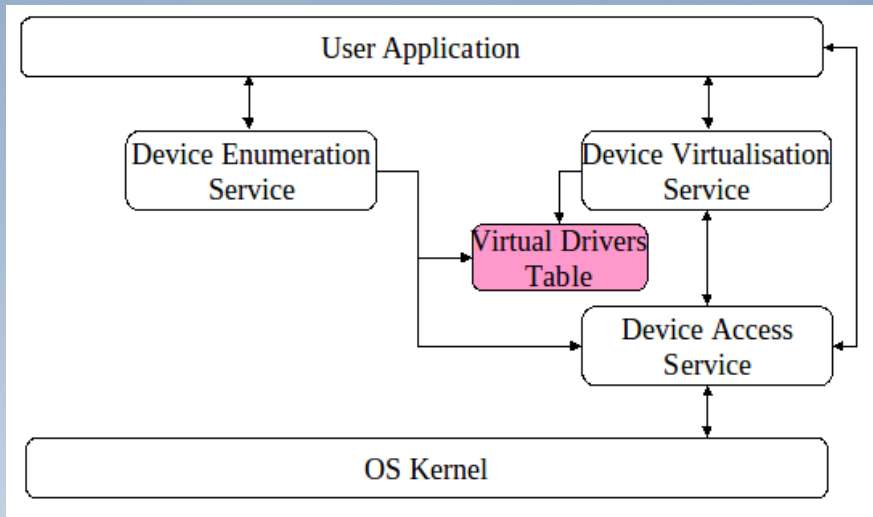
# Outline

- 1 Introduction
- 2 **TraffiController<sup>4SpW</sup> DVS**
- 3 **TraffiController<sup>4SpW</sup> API**
- 4 **TraffiController<sup>4SpW</sup> GUI**
- 5 Conclusion and questions



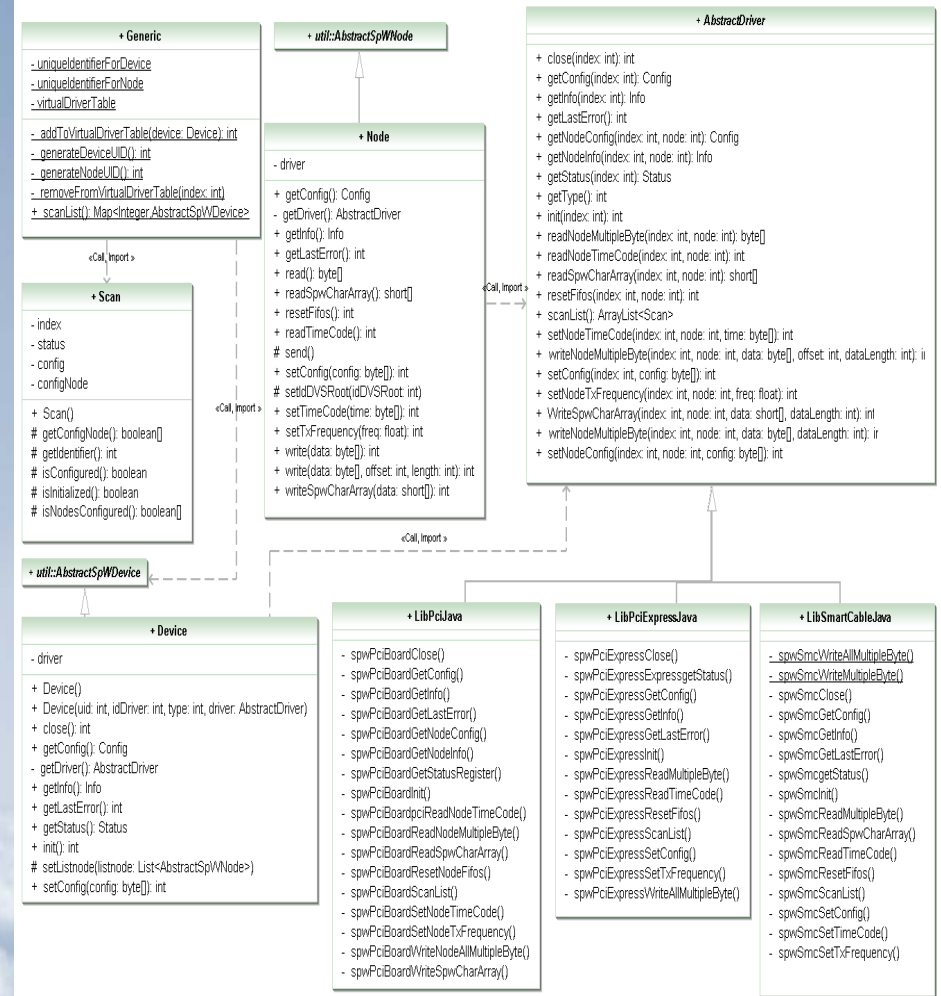


# DES/DVS/DAS (Devices Virtualization Service) overview



## DVS main functions

- Enumeration and update of hardware
- Management of SpW devices, nodes
- Driver(s) routines access service
- *ID\_DVS* listing



DVS object-oriented architecture

### DVS list of routines

DVS\_4SpW\_scanlist,  
DVS\_4SpW\_get\_device,  
DVS\_4SpW\_init,  
DVS\_4SpW\_close,  
DVS\_4SpW\_set\_config,  
DVS\_4SpW\_get\_Info,  
DVS\_4SpW\_get\_config,  
DVS\_4SpW\_get\_status\_register,

DVS\_4SpW\_get\_node, DVS\_4SpW\_set\_node\_config,  
DVS\_4SpW\_set\_node\_tx\_frequency,  
DVS\_4SpW\_set\_node\_time\_code,  
DVS\_4SpW\_reset\_node\_fifo,  
DVS\_4SpW\_get\_node\_Info,  
DVS\_4SpW\_read\_node\_time\_code,  
DVS\_4SpW\_read\_SIP, DVS\_4SpW\_read\_SpW,  
DVS\_4SpW\_write\_SIP, DVS\_4SpW\_write\_SpW

### DVS scan example

```
import util.AbstractSpWDevice;
import util.AbstractSpWNode;
import dvs.Generic;
public class Test {
    public static Map<Integer, AbstractSpWDevice> listDevice = new TreeMap<Integer, AbstractSpWDevice>();
    public static void main(String[] args) {
        listDevice = Generic.scanList();
        boolean exist = false;
        System.out.println("|");
        System.out.println("|__[List of SMARTCABLE devices]");
        for (AbstractSpWDevice device : listDevice.values())
            if (device.getType() == 1) {
                exist = true;
                System.out.printf("|    |__SMC %d : IDDVS [0x%04x] : Initialized [%b] : Configured [%b]\n",
                    device.getIdDriver(), device.getUID(), device.isInitialized(), device.isConfigured());
                if (device.getListnode() != null)
                    for (AbstractSpWNode node : device.getListnode())
                        System.out.printf("|    |    |__Node %d : IDDVS [0x%04x] : Initialized [%b] : Configured [%b]\n",
                            node.getNumNode(), node.getUIdDvs(), node.isInitialized(), node.isConfigured());
            }
    }
}
```

## 2 – traffiController<sup>4</sup>SpW

### DVS console

```
[Test Device]
!__(e) Enumerate Devices
(i) Init
(c) Config
(g) Get Info
(d) Command Devices
(x) Exit
Enter your choice :
```

```
__[List of SMARTCABLE devices]
_SMC 0 : IDBUS [0x0000] : Initialized [false] : Configured [false]
!_Node 0 : IDBUS [0x0000] : Initialized [false] : Configured [false]
_SMC 1 : IDBUS [0x0001] : Initialized [false] : Configured [false]
!_Node 0 : IDBUS [0x0001] : Initialized [false] : Configured [false]
_SMC 2 : IDBUS [0x0002] : Initialized [false] : Configured [false]
!_Node 0 : IDBUS [0x0002] : Initialized [false] : Configured [false]
_SMC 3 : IDBUS [0x0003] : Initialized [false] : Configured [false]
!_Node 0 : IDBUS [0x0003] : Initialized [false] : Configured [false]
_SMC 4 : IDBUS [0x0004] : Initialized [false] : Configured [false]
!_Node 0 : IDBUS [0x0004] : Initialized [false] : Configured [false]
_SMC 5 : IDBUS [0x0005] : Initialized [false] : Configured [false]
!_Node 0 : IDBUS [0x0005] : Initialized [false] : Configured [false]

__[List of PCI devices]
_PCI 0 : IDBUS [0x0006] : Initialized [false] : Configured [false]
!_Node 0 : IDBUS [0x0006] : Initialized [false] : Configured [false]
!_Node 1 : IDBUS [0x0007] : Initialized [false] : Configured [false]
!_Node 2 : IDBUS [0x0008] : Initialized [false] : Configured [false]
!_Node 3 : IDBUS [0x0009] : Initialized [false] : Configured [false]
_PCI 1 : IDBUS [0x0007] : Initialized [false] : Configured [false]
!_Node 0 : IDBUS [0x000a] : Initialized [false] : Configured [false]
!_Node 1 : IDBUS [0x000b] : Initialized [false] : Configured [false]
!_Node 2 : IDBUS [0x000c] : Initialized [false] : Configured [false]
!_Node 3 : IDBUS [0x000d] : Initialized [false] : Configured [false]
_PCI 2 : IDBUS [0x0008] : Initialized [false] : Configured [false]
!_Node 0 : IDBUS [0x000e] : Initialized [false] : Configured [false]
!_Node 1 : IDBUS [0x000f] : Initialized [false] : Configured [false]
!_Node 2 : IDBUS [0x0010] : Initialized [false] : Configured [false]
!_Node 3 : IDBUS [0x0011] : Initialized [false] : Configured [false]
_PCI 3 : IDBUS [0x0009] : Initialized [false] : Configured [false]
!_Node 0 : IDBUS [0x0012] : Initialized [false] : Configured [false]
!_Node 1 : IDBUS [0x0013] : Initialized [false] : Configured [false]
!_Node 2 : IDBUS [0x0014] : Initialized [false] : Configured [false]
!_Node 3 : IDBUS [0x0015] : Initialized [false] : Configured [false]
_PCI 4 : IDBUS [0x000a] : Initialized [false] : Configured [false]
!_Node 0 : IDBUS [0x0016] : Initialized [false] : Configured [false]
!_Node 1 : IDBUS [0x0017] : Initialized [false] : Configured [false]
!_Node 2 : IDBUS [0x0018] : Initialized [false] : Configured [false]
!_Node 3 : IDBUS [0x0019] : Initialized [false] : Configured [false]

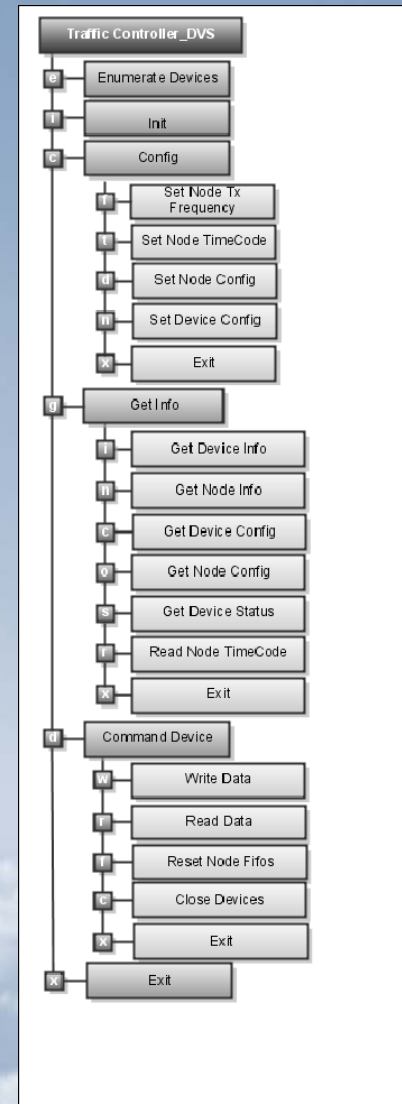
__[List of PCI Express devices]
!_NONE
```

```
[Test Device]
!__[Config]
!__(f) Set Node Tx Frequency
(t) Set Node TimeCode
(n) Set Node Config
(d) Set Device Config
(x) Exit
Enter your choice : f

For which node do you want to set Frequency?[node ID --> IDBUS]
12

===== Result =====
Enter SpW CODEC transmission frequency value between 2Mhz and 200Mhz for node 0x0012 :
150

Setup SpW node 0 frequency of PCI Board 3 to 150.0MHz.
NO ERROR
=====
```





# Outline

- 1 Introduction
- 2 TrafficController<sup>4SpW</sup> DVS
- 3 TrafficController<sup>4SpW</sup> API
- 4 TrafficController<sup>4SpW</sup> GUI
- 5 Conclusion and questions



## Application Programming Interface (API) overview

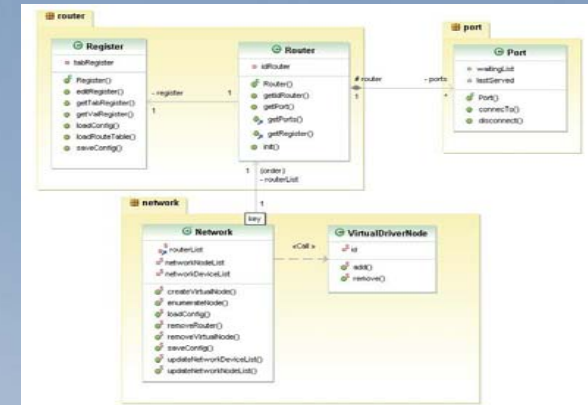
- SpW networking emulation and command control based on the following objects:
  - Router, node, ports objects
  - RMAP engine, multiple instances
  - *Ethernet communication for API dialogues* engine, multiple instances
  - Link streams for Virtual protocol analyzer engine, multiple instances

### API main functions

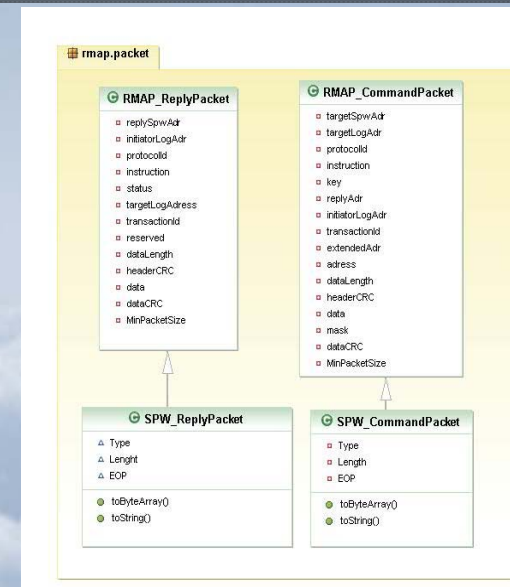
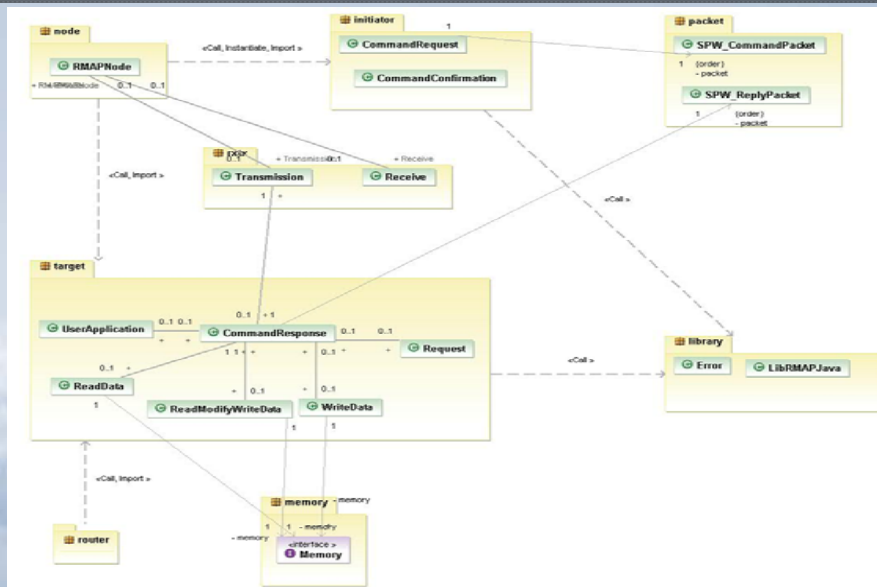
- SpW networking using virtual routers, and virtual, real or emulated nodes
- RMAP support (ECSS-E-ST-50-11C)
- IP/Ethernet tunneling
- Virtual protocol analyzer

### API Router

- 2 to 31 ports Wormhole routing,
- configurable via RMAP port 0
- path and logical addressing using routing table
- 3 arbitration modes:
  - packet priority, last served, lowest port number
- implementation inspired from SpW-10x router



### API RMAP module

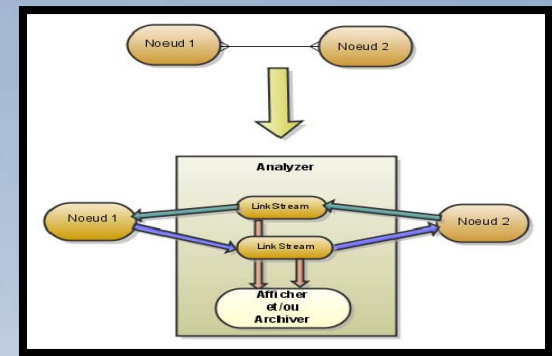




### API: IP/Ethernet tunneling

- API interconnexion through IP/Ethernet
- local PC / local area / intranet or internet networking modes
- using Java RMI. Tested with VPN and direct client/server connexion.

### API: Protocol analyzer



- between real or emulated nodes (PCI/PCI Express, smartCable), router ports and virtual nodes
- activable (on/off) for each direction of any link
- 4 configuration types: debug, blocking buffered, unblocking buffered, continuous
- 2 outputs: display or archive
- simple `setMode()` and `getData()` routines for easy mode config and data retrieves



## 2 - traffiController<sup>4</sup>SpW

### API console

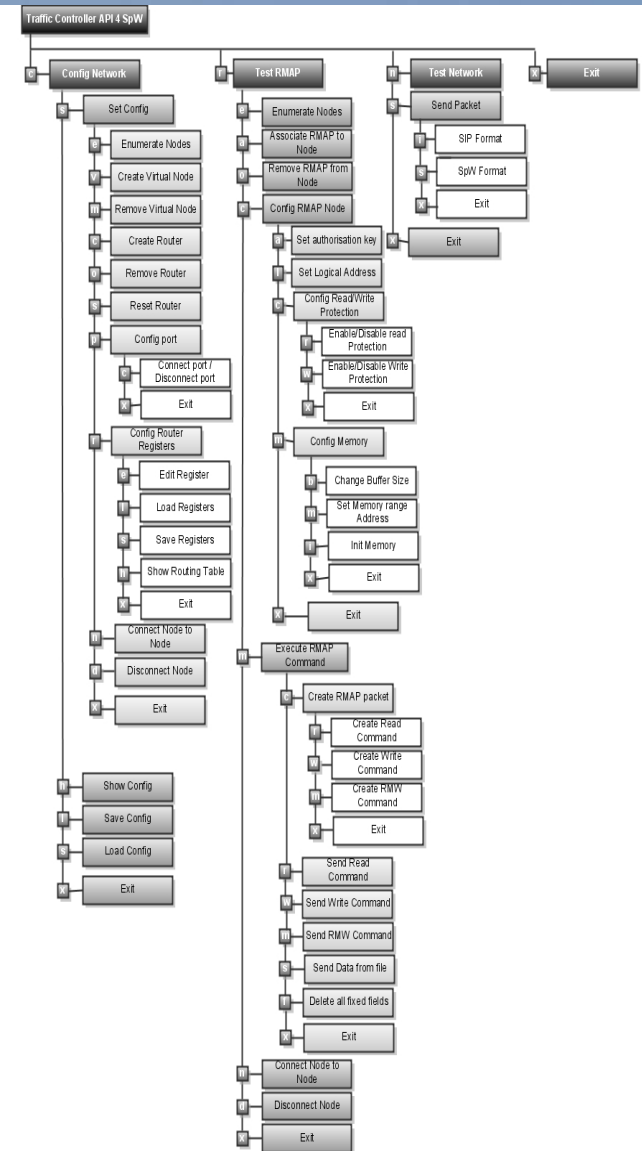
```
~~~~~
Traffic Controller 4 SpW
~~~~~
=====Main Menu=====
(c) Config Network
(r) Test RMAP
(n) Test Network
(x) Exit
Enter your choice : _
```

```
[Config Network]
!_ (s) Set Config
!_ (h) Show Config
!_ (a) Save Config
!_ (l) Load Config
!_ (x) Exit
Enter your choice : _
```

```
[Config Network]
!_ [Set Config]
!_ (e) Enumerate Nodes
!_ (v) Create Virtual Node
!_ (m) Remove Virtual Node
!_ (c) Create Router
!_ (o) Remove Router
!_ (s) Reset Router
!_ (p) Config port
!_ (r) Config Router Registers
!_ (n) Connect Node to Node
!_ (d) Disconnect node from node
!_ (x) Exit
Enter your choice : _
```

```
[Config Network]
!_ [Set Config]
!_ (e) Enumerate Nodes
!_ (v) Create Virtual Node
!_ (m) Remove Virtual Node
!_ (c) Create Router
!_ (o) Remove Router
!_ (s) Reset Router
!_ (p) Config port
!_ (r) Config Router Registers
!_ (n) Connect Node to Node
!_ (d) Disconnect node from node
!_ (x) Exit
Enter your choice : p
number of router in network [2]
!_ Router Id: [0x00]
!_ Router Id: [0x01]
Enter the Id of router
0
Router Id [0x00]
!_ number of port in this router: 5
!_ Port 1 [IdNode:0x0000] : connected with virtual Node 0 [IdNode:0x0008]
!_ Port 2 [IdNode:0x0001] : Not connected
!_ Port 3 [IdNode:0x0002] : connected with virtual Node 1 [IdNode:0x0009]
!_ Port 4 [IdNode:0x0003] : Not connected
!_ Port 5 [IdNode:0x0004] : Not connected
Enter port number
```

```
[Config Network]
!_ [Set Config]
!_ [Config Router Registers]
!_ (e) Edit Register
!_ (l) Load Registers
!_ (s) Save Registers
!_ (h) Show Routing Table
!_ (x) Exit
Enter your choice : h
!_ Routing Table
Address      Priority    Delete Header    Ports
32          0          0                2
33          0          0                3
34          0          0                4
35          0          0                5
36          0          0                6
37          0          0                7
38          0          0                8
39          0          0                9
40          0          0                1
41          0          0                2
Press ENTER to continue or X to Exit
```



# Outline

- 1 Introduction
- 2 traffiController<sup>4SpW</sup> DVS
- 3 traffiController<sup>4SpW</sup> API
- 4 traffiController GUI
- 5 Conclusion and questions



## Graphical User Interface (overview)

Rich Menu  
Toolbar

Local API  
Hardware  
ressources

log

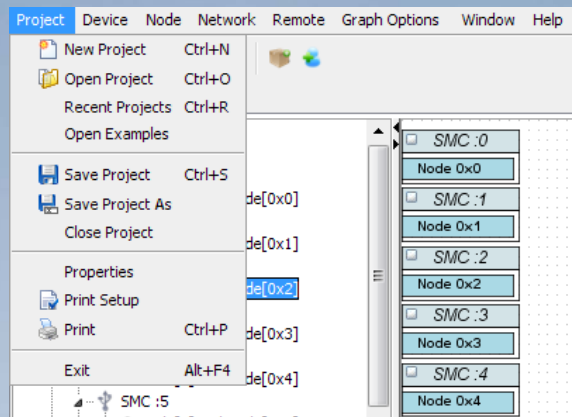
Work area  
dimensionning

Traffic statistics  
Eye diagrams

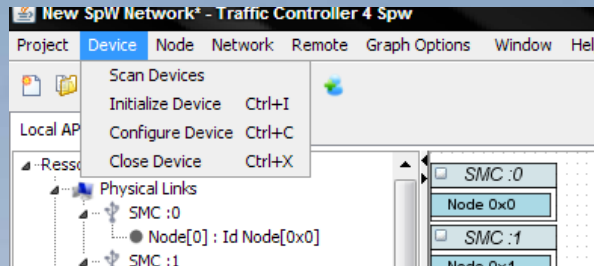
The screenshot displays the traffiController<sup>4</sup>SpW graphical user interface. The main window is titled "D:\ASIC\SMC\projct\_smartcable\projct\_lots1\Lot4\l1vivable\_sky\l1b\lot4\_25\_11\_09\Lot4\_App\TC\_4SpW\_GUI\Executable\Windows\Examples\example2.tic - Traffic Controller 4 Spw". The interface includes a menu bar (Project, Device, Node, Network, Remote, Window, Help) and a toolbar. On the left, there is a "Local API" panel showing a tree view of hardware resources, including PCI devices, Virtual Modules, User links, and Routers (Router 0 and Router 1) with their respective ports. Below this is an "Information" log panel. The central workspace is a large grid where a network diagram is being constructed, featuring two routers (Router:0x0 and Router:0x1) connected to various switches (SMC:0 to SMC:5) and PCI devices (PCI:0 to PCI:4). On the right side, there are panels for "Work area dimensionning" (showing a small schematic) and "Traffic statistics" (with tabs for Statistics, Graph, and Spy). At the bottom right, there are "Options" panels (Options 1, Options 2, Options 3). The Windows taskbar at the bottom shows the system clock as 07:20 and several open applications.

Main working area panel with intuitive functions acces  
(right click or keyboard shortcuts)

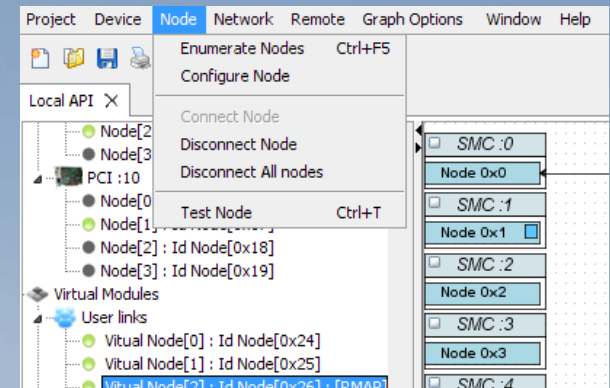
# Graphical User Interface (basic functions)



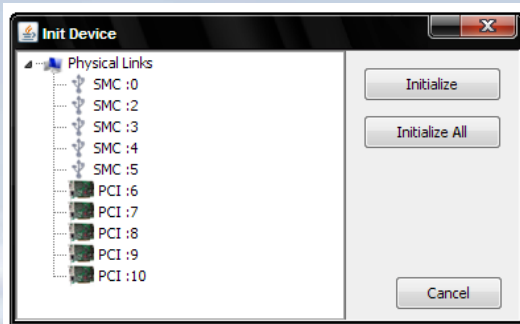
Project management



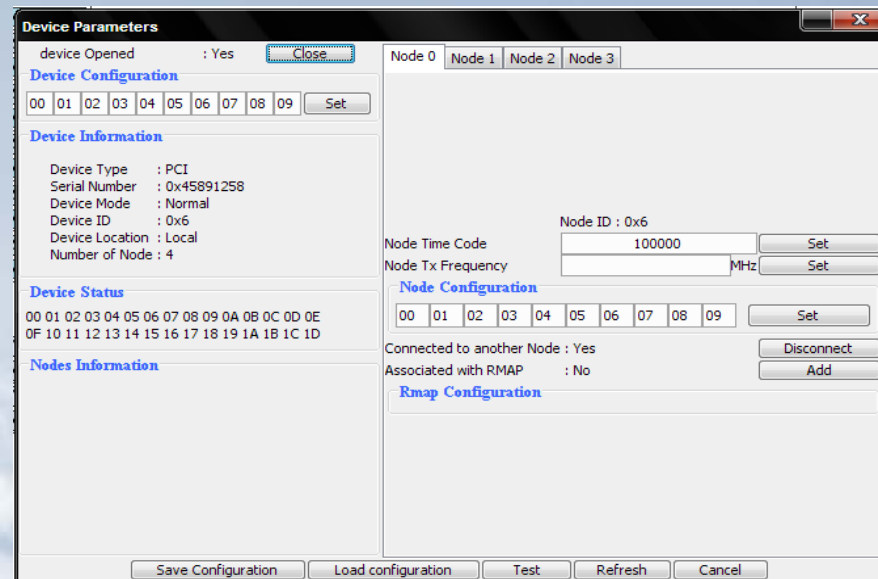
Device access



Node management



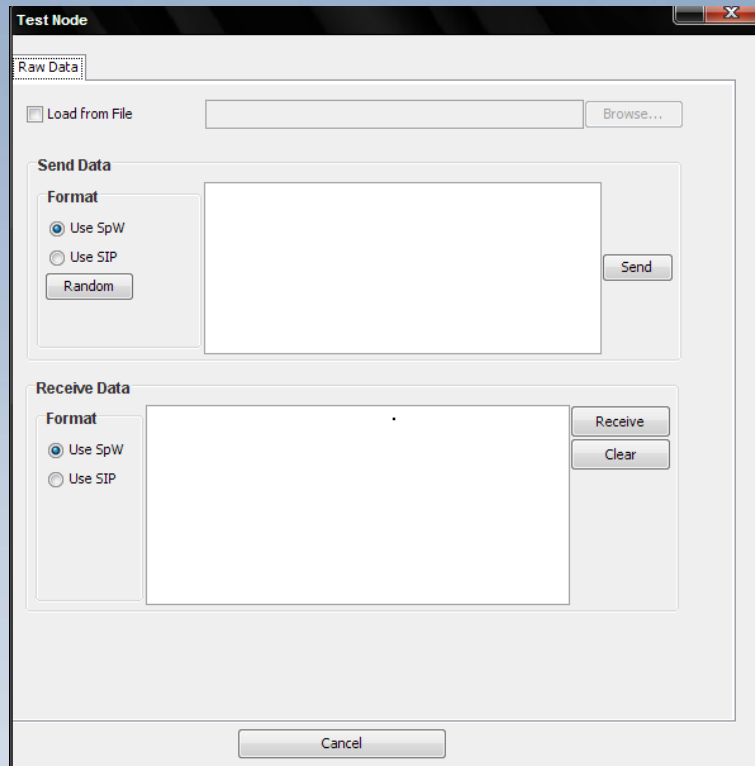
Peripheral initialization



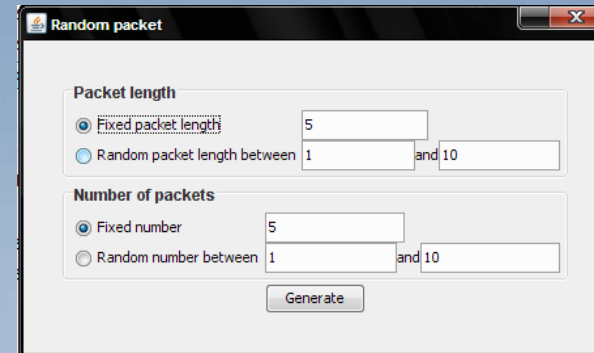
Peripheral configuration (ex: PCI<sup>4</sup>SpW)



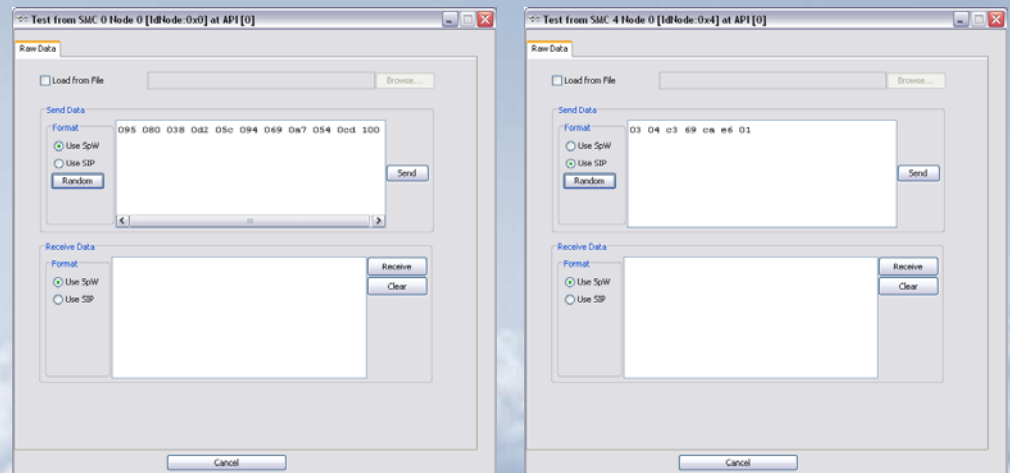
## Graphical User Interface (SpW read/write functions)



SpW or SIP\* Send/Receive



Packet generator



\*SIP is SKYLAB SpaceWire Interpreted Protocol, a 32bit 'large cargos' efficient processing format.

# Graphical User Interface (RMAP Write function)

**Device Parameters**

device Opened : Yes Close

**Device Configuration**

00 01 02 03 04 05 06 07 08 09 Set

**Device Information**

Device Type : PCI  
 Serial Number : 0x45891258  
 Device Mode : Normal  
 Device ID : 0x6  
 Device Location : Local  
 Number of Node : 4

**Device Status**

00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E  
 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D

**Nodes Information**

Node 0 Node 1 Node 2 Node 3

Node ID : 0x6

Node Time Code  Set

Node Tx Frequency  MHz Set

**Node Configuration**

00 01 02 03 04 05 06 07 08 09 Set

Connected to another Node : Yes Disconnect

Associated with RMAP : Yes Remove

**Rmap Configuration**

Logical Address ( Hex)  Set

Authorization Key (Hex)  Set

Buffer Size (decimal)  Set

Extended Address range  ~  Set

Memory Address range  ~  Set

☐ Write Protection Set

☐ Read Protection Set

Save Configuration Load configuration Test Refresh Cancel

**Test from SMC 0 Node 0 [IdNode:0x0] at API[0]**

Raw Data RMAP Command

target SpaceWire Address

Target Logical Address

Initiator Logical Address

Reply Address

Extended Address

Memory Address

Authorization Key

Transaction Id

Data length

RMW Data

RMW Mask

**Select Command**

☐ Read Command

☒ Write Command

☐ RMW Command

**Reply Address length**

☐ 0 byte

☒ 4 bytes

☐ 8 bytes

☐ 12 bytes

**Flags**

☒ Ack

☒ Verify

☐ Increment

**Send Data**

Send Command

Cancel

# Graphical User Interface (RMAP Read function)

**Device Parameters**

device Opened : Yes Close

**Device Configuration**

00 01 02 03 04 05 06 07 08 09 Set

**Device Information**

Device Type : PCI  
 Serial Number : 0x45891258  
 Device Mode : Normal  
 Device ID : 0x6  
 Device Location : Local  
 Number of Node : 4

**Device Status**

00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E  
 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D

**Nodes Information**

Node 0 Node 1 Node 2 Node 3

Node ID : 0x6

Node Time Code  Set

Node Tx Frequency  MHz Set

**Node Configuration**

00 01 02 03 04 05 06 07 08 09 Set

Connected to another Node : Yes Disconnect

Associated with RMAP : Yes Remove

**Rmap Configuration**

Logical Address ( Hex)  Set

Authorization Key (Hex)  Set

Buffer Size (decimal)  Set

Extended Address range  ~  Set

Memory Address range  ~  Set

☐ Write Protection Set

☐ Read Protection Set

Save Configuration Load configuration Test Refresh Cancel

**Test from SMC 0 Node 0 [IdNode:0x0] at API [0]**

Raw Data RMAP Command

target SpaceWire Address

Target Logical Address

Initiator Logical Address

**Reply Address**

**Extended Address**

Memory Address

Authorization Key

Transaction Id

Data length

RMW Data

RMW Mask

**Send Data**

Send Command

Cancel

**Select Command**

☒ Read Command  
☐ Write Command  
☐ RMW Command

**Reply Address length**

☐ 0 byte  
☐ 4 bytes  
☒ 8 bytes  
☐ 12 bytes

**Flags**

☒ Ack  
☐ Verify  
☒ Increment

# Graphical User Interface (RMAP RMW function)

**Device Parameters**

device Opened : Yes Close

**Device Configuration**

00 01 02 03 04 05 06 07 08 09 Set

**Device Information**

Device Type : PCI  
 Serial Number : 0x45891258  
 Device Mode : Normal  
 Device ID : 0x6  
 Device Location : Local  
 Number of Node : 4

**Device Status**

00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E  
 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D

**Nodes Information**

Node 0 Node 1 Node 2 Node 3

Node ID : 0x6

Node Time Code  Set

Node Tx Frequency  MHz Set

**Node Configuration**

00 01 02 03 04 05 06 07 08 09 Set

Connected to another Node : Yes Disconnect

Associated with RMAP : Yes Remove

**Rmap Configuration**

Logical Address ( Hex)  Set

Authorization Key (Hex)  Set

Buffer Size (decimal)  Set

Extended Address range  ~  Set

Memory Address range  ~  Set

☐ Write Protection Set

☐ Read Protection Set

Save Configuration Load configuration Test Refresh Cancel

**Test from SMC 0 Node 0 [IdNode:0x0] at API [0]**

Raw Data RMAP Command

target SpaceWire Address

Target Logical Address

Initiator Logical Address

Reply Address

Extended Address

Memory Address

Authorization Key

Transaction Id

Data length

RMW Data

RMW Mask

**Select Command**

☐ Read Command

☐ Write Command

☒ RMW Command

**Reply Address length**

☐ 0 byte

☒ 4 bytes

☐ 8 bytes

☐ 12 bytes

**Flags**

☒ Ack

☐ Verify

☐ Increment

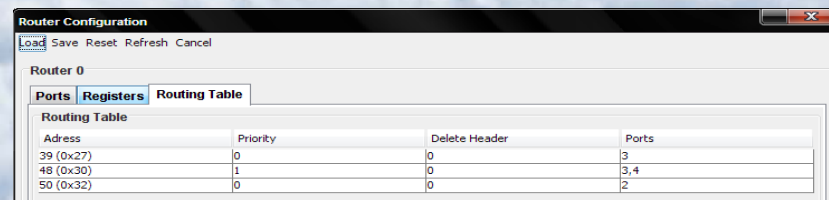
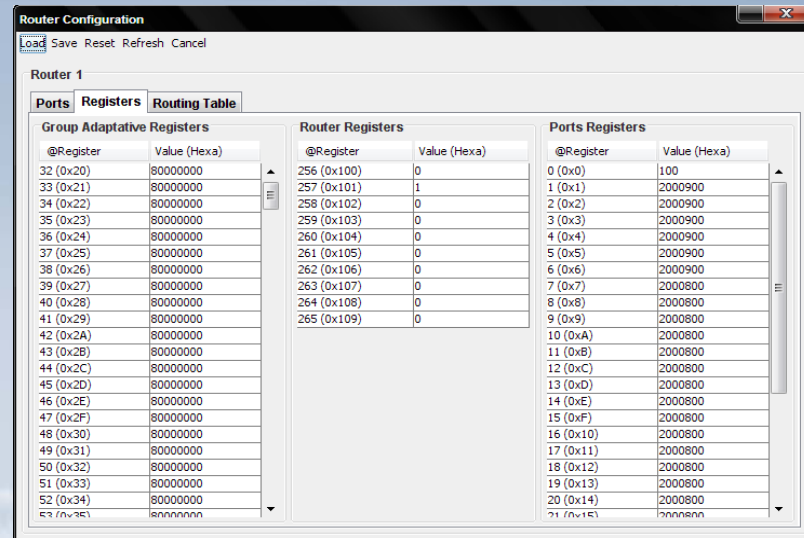
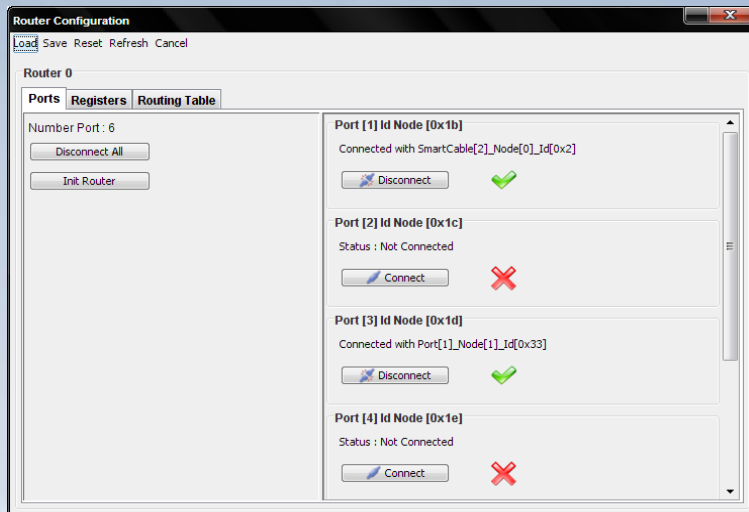
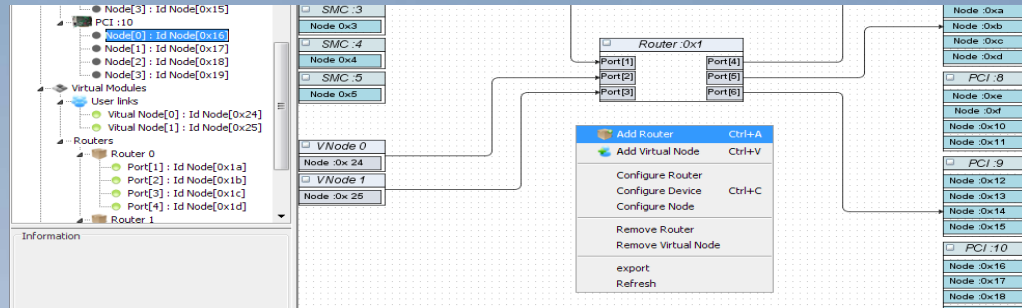
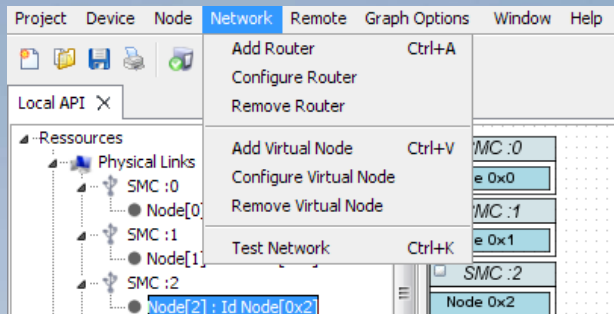
**Send Data**

Send Command

Cancel



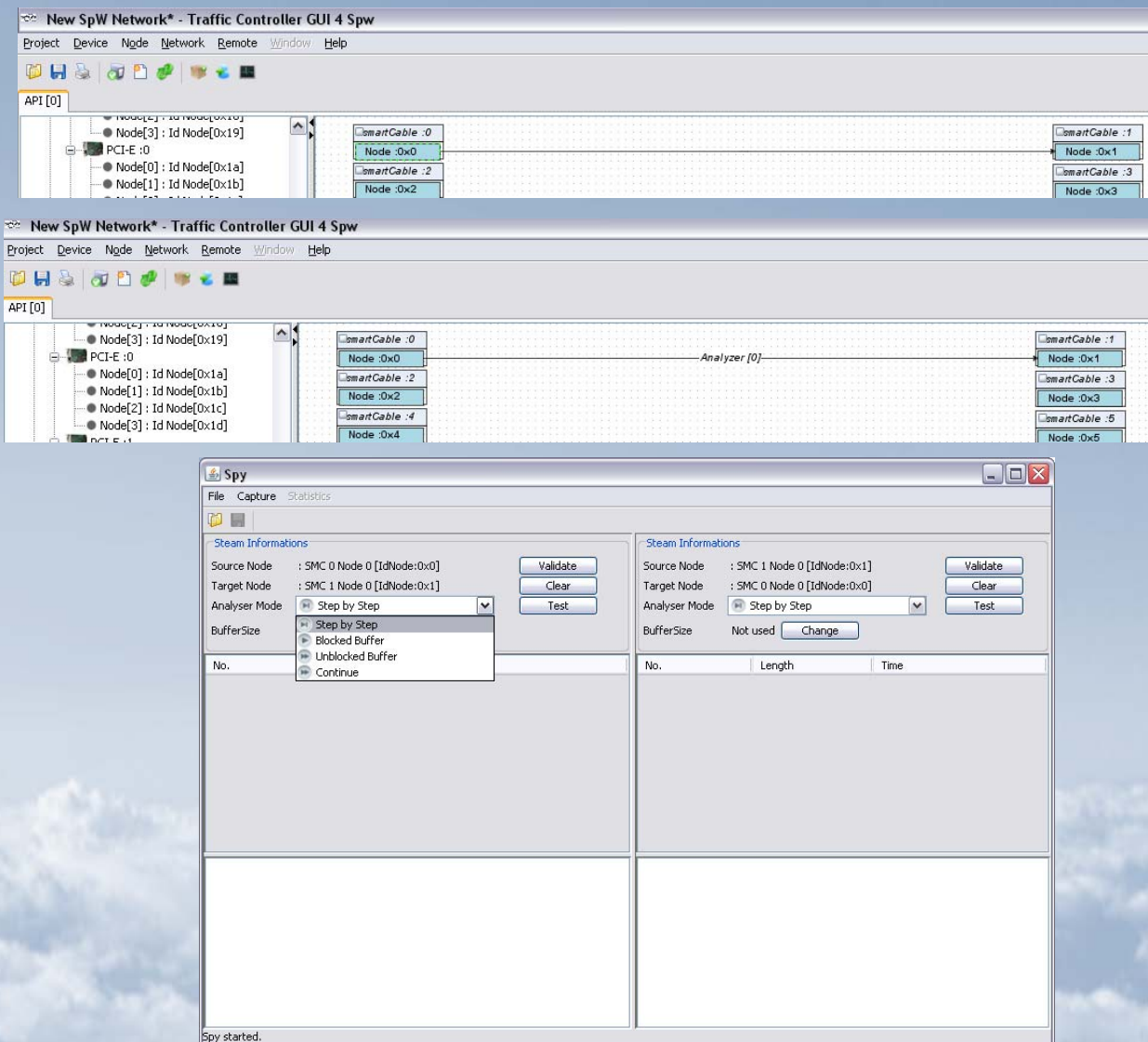
# Graphical User Interface (Router functions)





## 2 – traffiController<sup>4</sup>SpW

# Graphical User Interface (virtual analyzer function)



# Outline

- 1 Introduction
- 2 TrafficController<sup>4SpW</sup> DVS
- 3 TrafficController<sup>4SpW</sup> API
- 4 TrafficController<sup>4SpW</sup> GUI
- 5 Conclusion and questions



## 5 – Conclusion and questions

### **traffiController<sup>4SpW</sup>**

- DVS
- API
- GUI

are the 3 levels of **traffiController** architecture, which implements extensive software features useful for network emulation, integration and test.

It is provided with user manual, javadoc, test use cases

TraffiController is included all hardware equipment commercialized by SKYLAB

It is also available in stand alone (without hardware, using emulated nodes)

There is a Skylab4SpW Googlegroup community for user feedback and support

SKYLAB plans to open the platform to competitive hardwares if demand strong enough.

We would like to hear about your potential additionnal requirements to be implemented.

**Questions?**

**Thank you for your attention !**

## **SKYLAB Industries SAS**

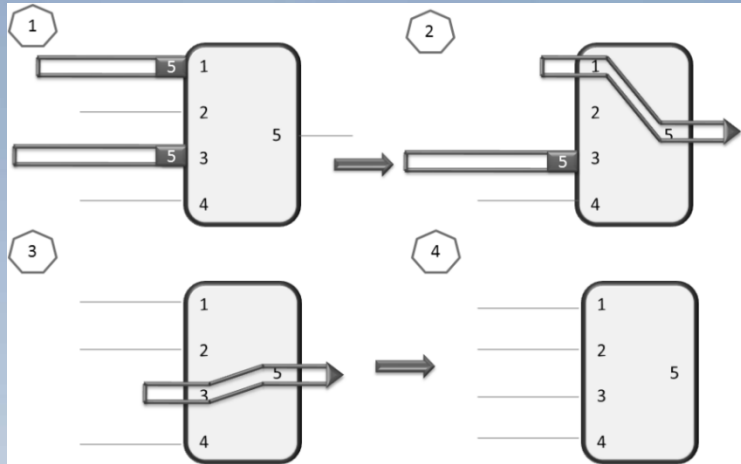
**42, av. du general de Croutte  
31100 Toulouse - France**

**Tel. +33 (0) 5.61.41.77.03  
Fax +33 (0) 5.61.41.63.56**

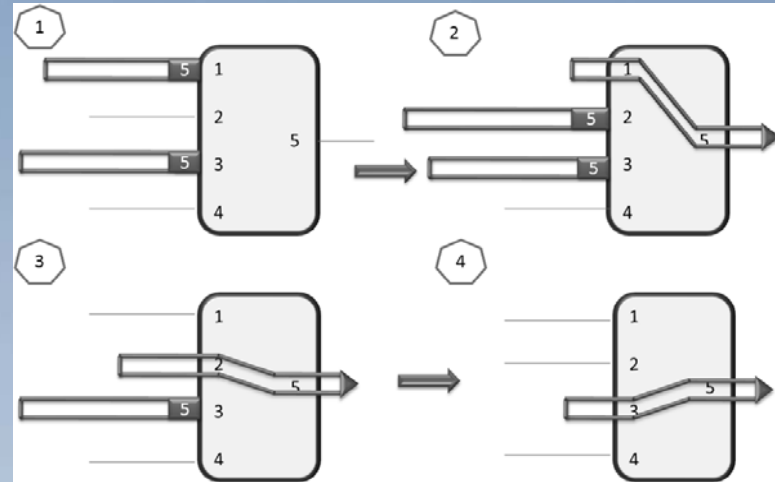
**[spacewire@skylab-corporate.com](mailto:spacewire@skylab-corporate.com)  
[www.skylab-corporate.com](http://www.skylab-corporate.com)**



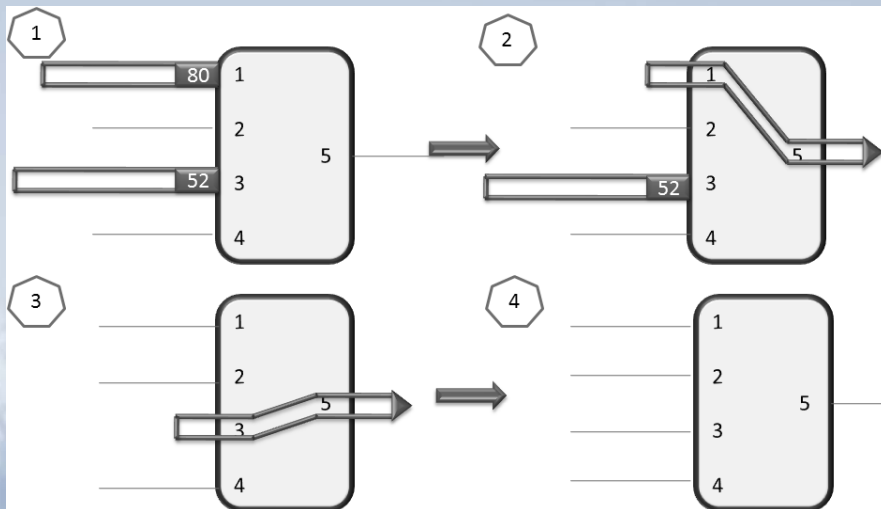
# Spare Slide 1



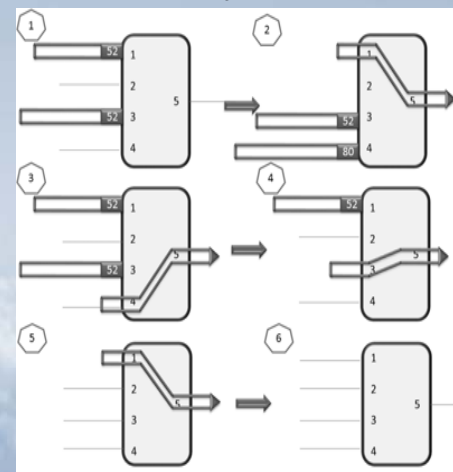
Same priority arbitration



Same priority arbitration



Different priorities arbitration



Different priorities arbitration