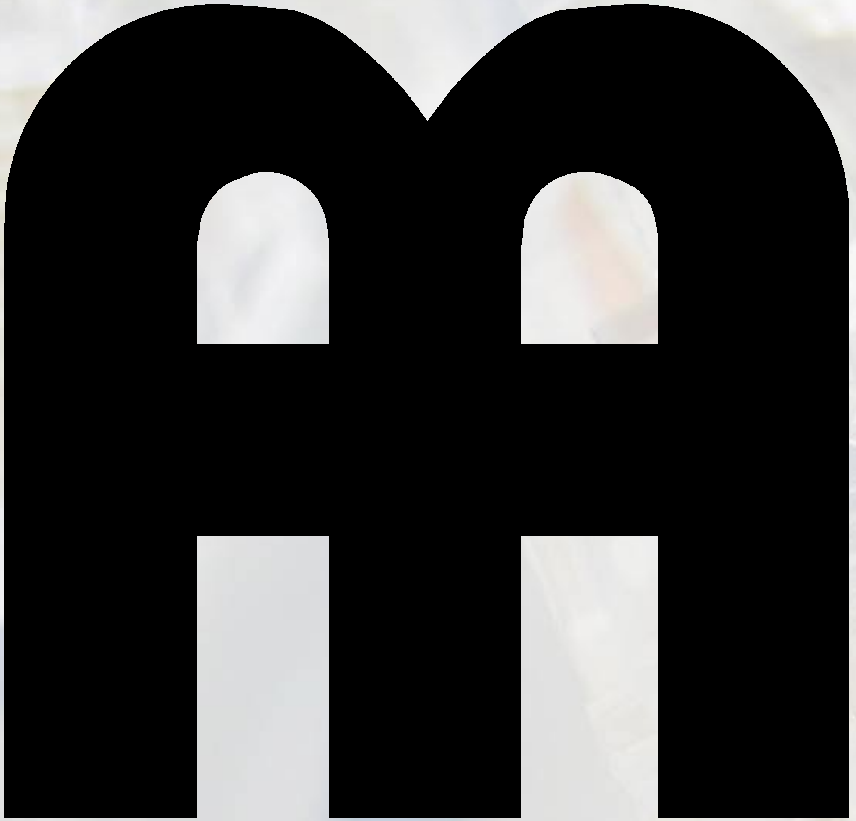
White paper



MambaNet™

Virtual reality controlled with knobs

Why MambaNet™?

In the audio industry for many years the studio installations where built out of many different equipment devices. Each piece of equipment has its own dedicated control because analog technology was used.
The integration of this equipment to achieve more functionality was done by system integrators.
In the last years we have seen technology has changed. Dedicated controls are less often used, more and more you can control the devices from software and/or remote control surfaces. The advantage is that we have many smart standalone products.
But when we need to integrate many of these smart products it will be more complex and it could be confusing for the system integrator and for the end-user as well.

"Technology should support the end-user"

The solution for simplifying the operation, integration and making profit of the available technology is, strange enough, a communication issue.
It is obvious that the end-users will profit most when they can use an open standard for communication control between their devices.
The available open standards are not strong enough to fulfill the high demanding communication requirements of modern studio equipment. This has resulted in our decision to develop an open standard for studio control communications, that is called:

MambaNet™

MambaNet™ vs. other protocols.

Investigations of management protocols that are often used in the IT industry shows they often use a client-server structure (a good example is SNMP). In the studio environment we require a more sophisticated solution that could have a multi-master principle.
This makes it possible for devices to freely send information, at any time, to other devices.

Secondly, the control protocol must be independent from the hardware layers used. This gives us the possibility to transport the control data over a variety of mediums. Currently we have already implementations on Ethernet (level 2), CAN, TCP/IP, RS232.

- Ethernet
  The default interconnection medium between devices will be Ethernet (level 2). This results in the fact that you can use all standard Ethernet equipment and security technology.
- TCP/IP
  Is used to transfer MambaNet™ over the World Wide Web.
- CAN and RS232
  These physical layers are used for efficient solutions within devices and are not expected to be a connection standard.

To make the protocol future proof and also for other manufactures it is important that the functionality of devices can be read from the devices.

MambaNet™ definitions

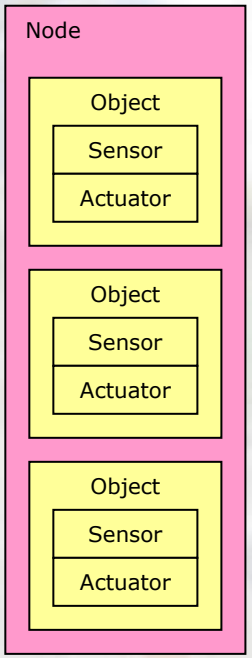
- Node
  An element that can be uniquely identified. For example a PCB or software process. A physical device may have multiple MambaNet™ nodes (e.g. for modularity).

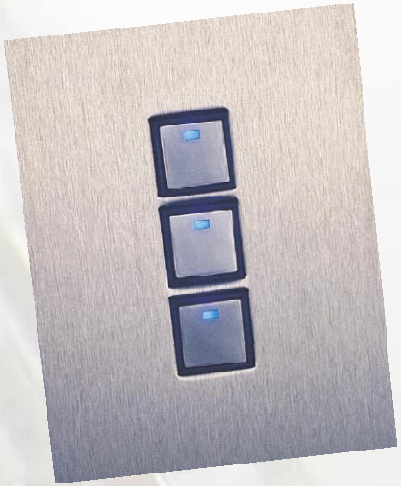
  A node will hold objects that can send and receive data.

- Object
  An object represents the smallest element that can be triggered and/or set. The object is divided in two parts:

  - Sensor
  This is the name for the information that is generated externally and needs to be sent to or read from the network (e.g. switch-position).

  - Actuator
  This is the name for the information that is sent from the network (e.g. switch-LED-indication).





Example:

- This node holds 3 objects
- Each object has a switch if sensor and a LED actuator.

MambaNet™

Virtual reality controlled with knobs

Sensor's and Actuators

We have standardized the data format for sensors and actuators in the most used primitives:

- No data
  A sensor or actuator is not implemented.
  e.g. a switch without a feedback LED.

- Unsigned integer
  e.g. a knob position

- Signed integer
  e.g. an encoder change

- State
  e.g. a sample rate selection with 2 states: 44.1kHz/48kHz

- Octet string
  e.g. display text or an array of bytes.

- Float
  e.g. dB level

- Bit string
  e.g. LED bar

- Object information
  holds all required information for exploring this object.

MambaNet™ implementation

If you have multiple devices that are standardized at the MambaNet™ network you can connect them in the same local area network (LAN) and an engine process (e.g. the D&R Axum) will discover the devices. This process will connect the objects to each other, if configured. The engine process will give the end-user functionality to all your objects.

For external developers it is possible to make a MambaNet™ design by defining the equipment or application with:
Nodes -> Objects -> Sensors/Actuators -> Data types.

When you have this information structured, it will be easy to make valid implementations of MambaNet™ compatible devices.

External manufacturers/developers will require a manufacture identifier that will be assigned by D&R, to make sure your device will be unique on the MambaNet™ Network.
We have also the MambaNet™ stacks and additional MambaNet™ datasheets available. This can be used to speed up your development and reduces the possibility for implementation faults.

Please contact us at info@d-r.nl if you like to join the MambaNet™ network community.

MambaNet™ specifications

- Multi master
- Medium/Transport Layer independent
- Object Oriented
- 65534 Manufacture IDs
- 4294836225 Unique numbers per Manufacturer
- 16384 message types
- Maximum op 98 bytes per message.
- 64512 Custom objects per node.
- Automatic identification/learning of nodes.

Headquarters

D&R Electronica B.V.
Rijnkade 15b
1382 GS WEESP
The Netherlands

Phone : +31 294 418014
Fax : +31 294 416987

Email : info@d-r.nl
Website : www.d-r.nl

Your dealer:

Version 1.0, March 9, 2011