

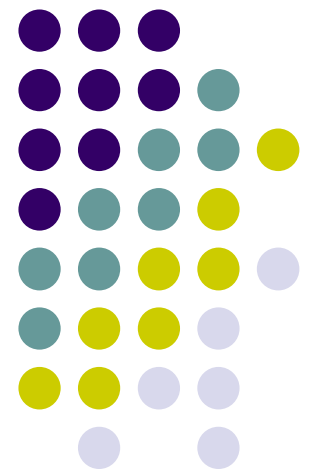
# Integra OSC protocol

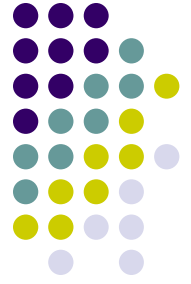
---

Concept proposal

Henrik Sundt

NOTAM





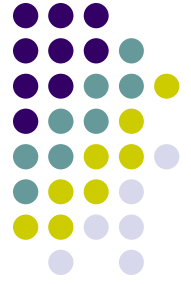
# Integra technology goals

- User friendly tool for live electronics, (“Integra Simple-GUI”).
- Methods for preservation of works.
- Encourage inter-program communication, and contribute to a modular environment for live electronics.



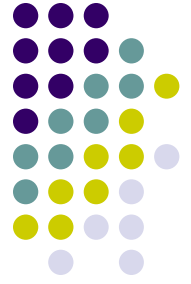
## NOTAM's current focus

- Proposal for a **protocol** as a solution to the goals, the "Integra OSC protocol".
- Transfer of the work "Physis" by Asbjørn Schaathun.
- Make a Simple-GUI example in Max.



# Our main ideas

- To obtain a user friendly tool: improve functionality of an existing, commonly used tool, rather than develop a new tool for live electronics.
- A **technology-independent** protocol for live electronics can be a basis for preserving works, as specific technology is becoming obsolete.
- The protocol can be implemented in different software systems and computer platforms, and in the user friendly tool.
- Achieve popularity of the protocol through usefulness and developer participation.

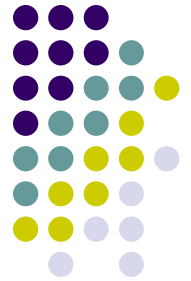


# Integra OSC protocol

- Based on Open Sound Control (OSC), a messaging system for audio and control purposes already commonly in use.
- OSC allows inter-program communication within a computer, and across networks.
- The Integra OSC protocol should be in constant development, with backwards compatibility.

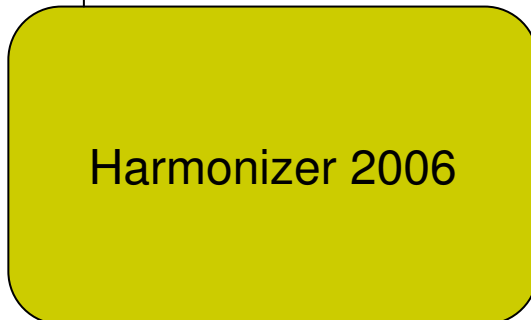
# IntegraOSC concept example

## - Future compability

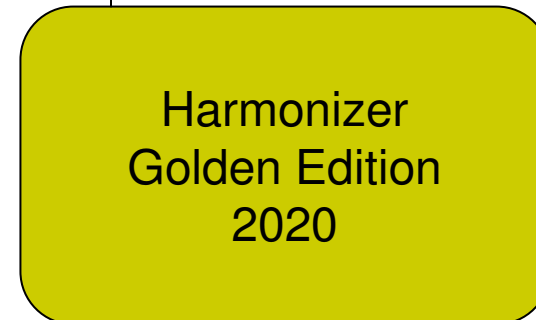


**Input OSC message:**

"/transpose 100 midicent"

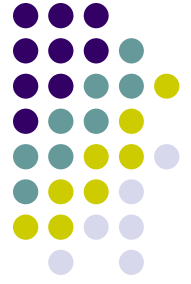


"/transpose 100 midicent"



# IntegraOSC concept example

## - Functional scalability

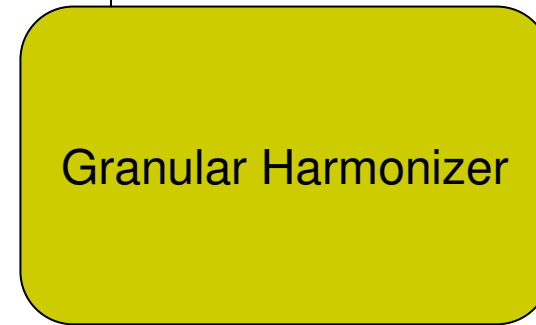


Input OSC message:

"/transpose 100 midicent"



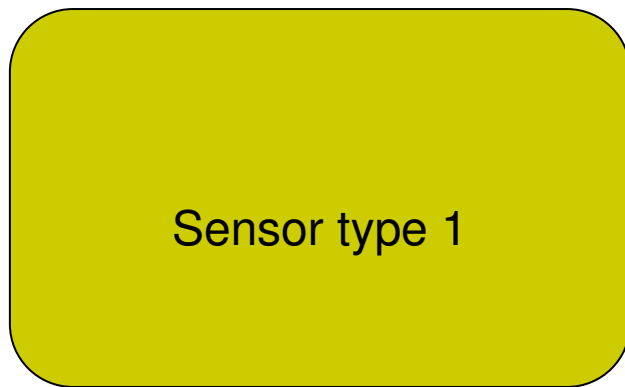
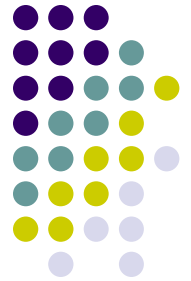
"/transpose 100 midicent"



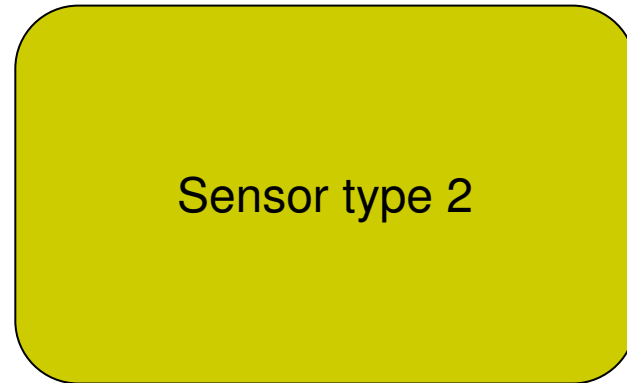
Also understands  
"/windowsize 1 ms"

# IntegraOSC concept example

- Implementation independence



"/distance 3.2 m"

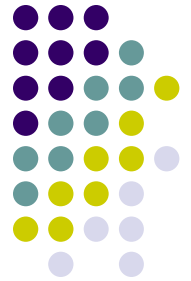


"/distance 3.2 m"

**Output OSC message**

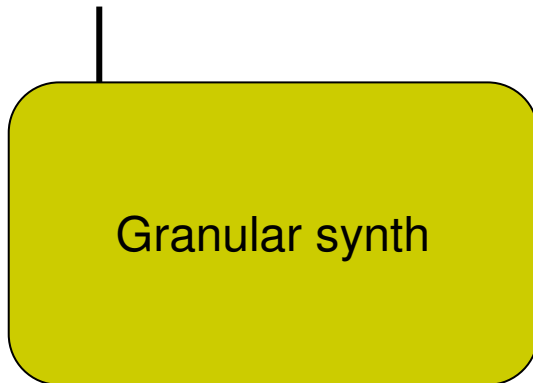
# IntegraOSC concept example

- Cross-tool compability



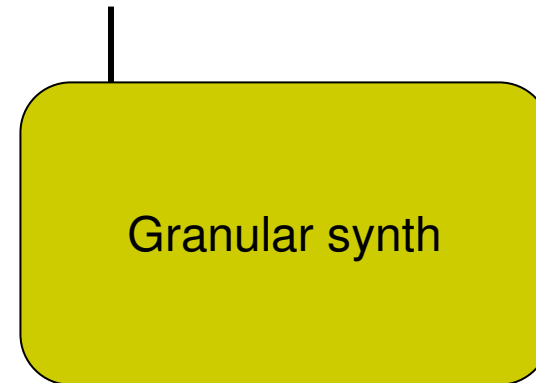
Input OSC message:

"/grainsize 10 ms"



**REAKTOR**

"/grainsize 10 ms"

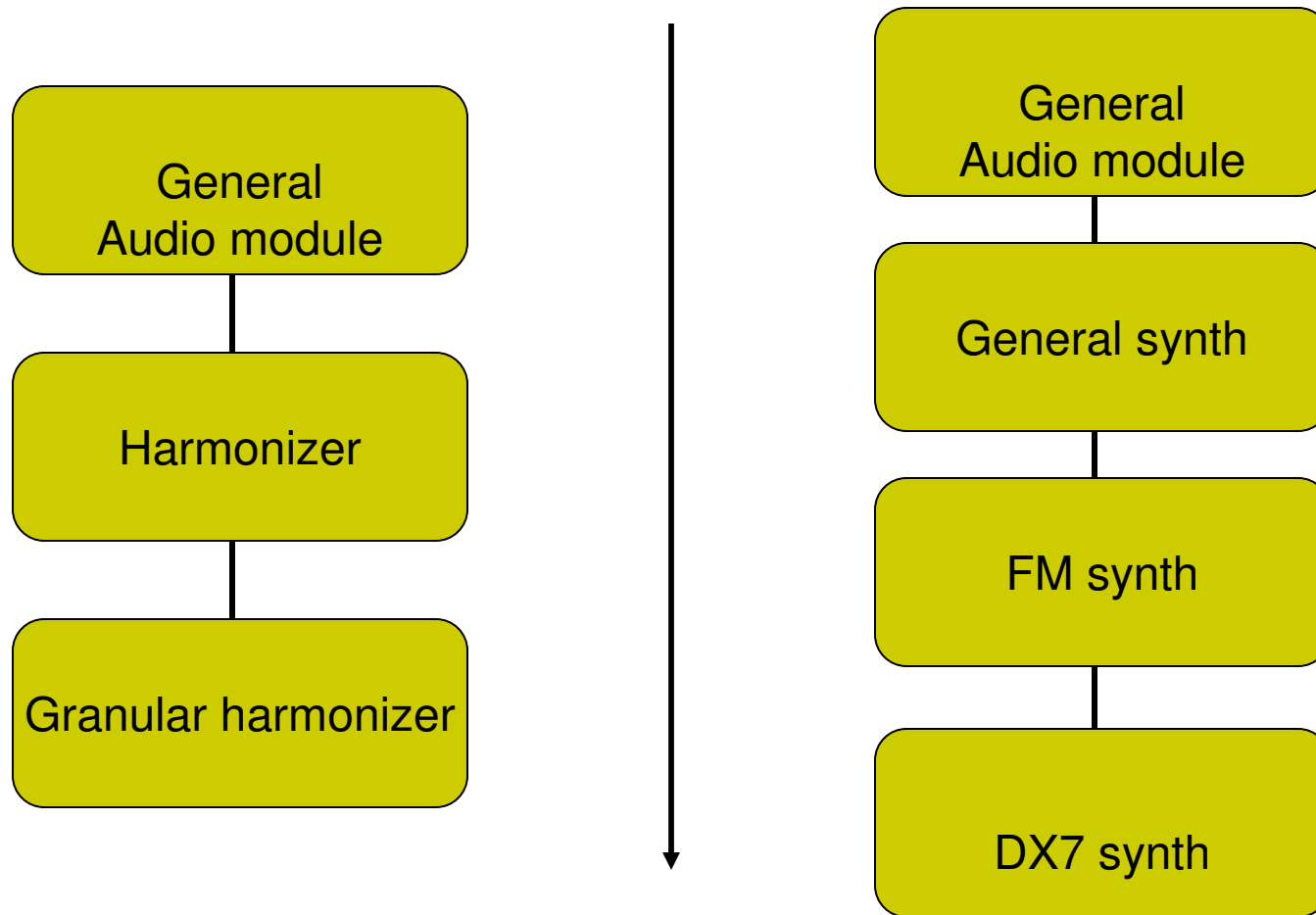


**MAX**

# IntegraOSC class hierarchy

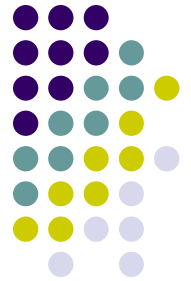


COMMON FUNCTIONS



SPECIALIZED FUNCTIONS

# IntegraOSC module definition



Operational description (DSP formulas or description, mathematical functions, audible characteristics ...)

Namespace, arguments and default values.

# Module documentation example



Browser window showing the documentation for the `/integra/function` module. The page title is `/integra/function` and the URL is `file:///Volumes/LYDDESIGN%202/INTEGRA/PRESENTASJON/example%20html-doc/integra/...`. The page content includes:

## Integra `/integra/function` - EXAMPLE DEFINITION -

one-dimensional linear break point transfer function

[Table of Contents](#) | [Index of Interfaces](#) | [Integralive.org](#)

### Description

Formal description of module...

### Configuration

Interface Type: `control`  
Interface Address: `/integra/function`

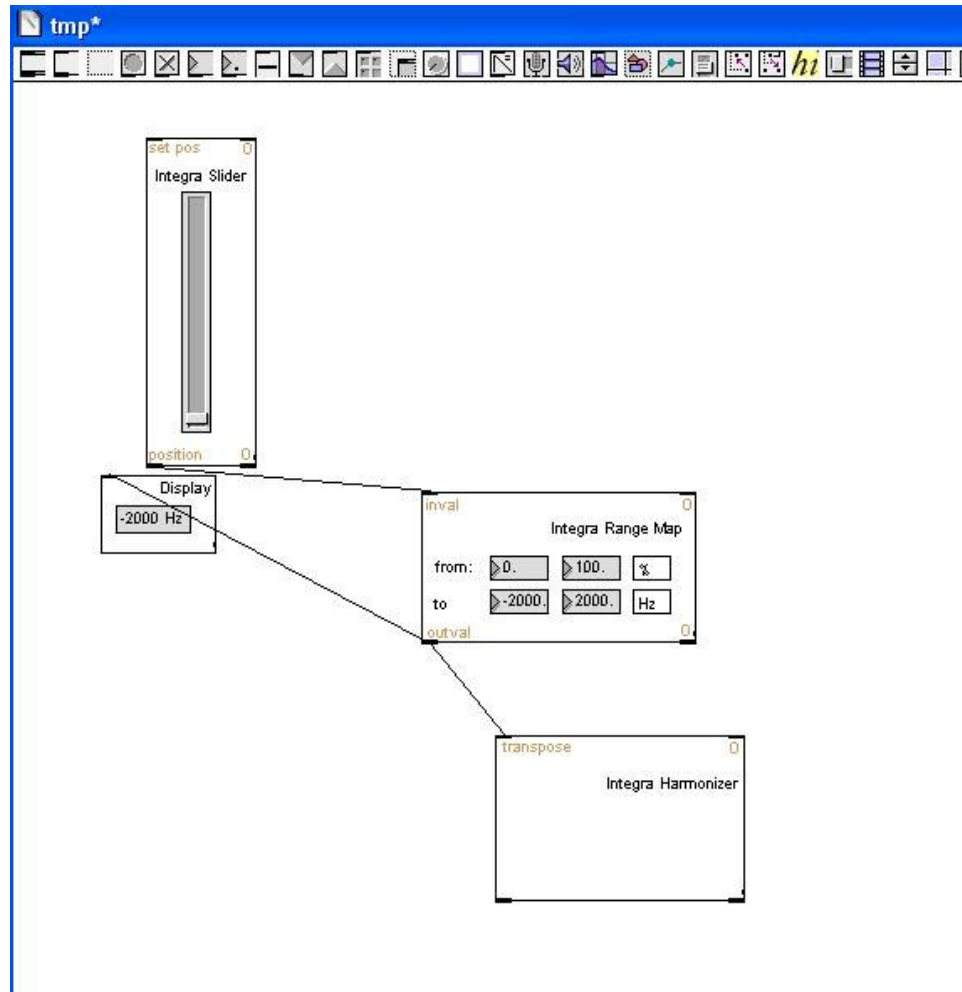
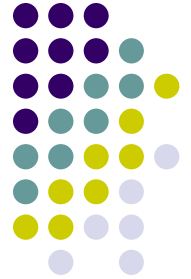
### Module Input Addresses

Name	Default value	Units	Range	Example	Description
<code>/integra/function/x_range &lt;range&gt;</code>	0 1	any	any	0 100 %	Sets the lower and upper limits for the x-axis.
<code>/integra/function/y_range &lt;range&gt;</code>	0 1	any	any	0 2000 Hz	Sets the lower and upper limits for the y-axis.
<code>/integra/function/add_point &lt;point&gt;</code>	-				Adds a break point within the ranges of the x- and y-axes.
<code>/integra/function/clear</code>	-	-	-	-	Removes all break points from the function.
<code>/integra/function/x &lt;value&gt;</code>	0	any	any	10 %	Outputs the corresponding y-value for the given x-value.

### Module Output Addresses

Name	Default value	Units	Range	Example	Description
<code>/integra/function/out/y &lt;value&gt;</code>	0	any	any	200 Hz	Outputs the corresponding y-value for the given x-value.

# IntegraOSC implementation sketch (in Max)



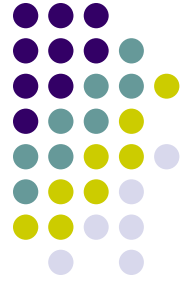
# Tryout on Physis



3 step plan:

- Transport Physis to Integra OSC in Max
- Documentation of the piece
- Reimplement piece in Supercollider using purely knowledge from documentation

# Content outside Integra OSC



- Modules could follow Integra OSC protocol, but **contain** data or code outside Integra OSC, e.g. a picture, or a Lisp script.
- Such data should be represented in an open, well-documented format.
- A more general "Integra Protocol" could have guidelines for documentation of parts outside "Integra OSC protocol".

# Integra OSC as documentation



## COMPLETE PATCH DOCUMENTATION

Patch in Integra OSC format  
+  
Data files in Integra OSC, or open data format  
+  
Composer's notes on intention and performance

Integra Protocol documentation  
(common between all works)

# Survival of Integra OSC



- Protocol survival benefits on popularity / widespread use.
- Helps us: There is a need for a unified standard for inter-program communication.
- Hope: Developers will want to use Integra OSC.
- A difficult choice: Complexity vs popularity of the protocol.



# Integra OSC - challenges

- Defining a sensible class tree for today's live electronics.
- Definitions to be useable for future needs .
- Keeping backwards compatibility.
- Guidelines for documentation of different types of content.