AsTeRICS Hands On!
Camera Mouse Solution

FH University of Applied Sciences TECHNIKUM WIEN
AsTeRICS academy
MA23 Wirtschaft, Arbeit Statistik
StaDt+Wien

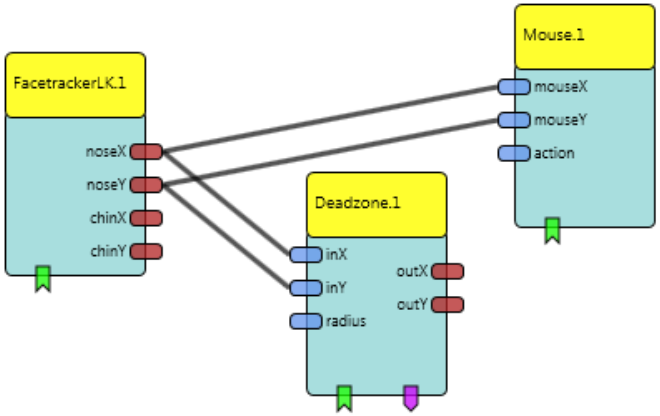
The Basics

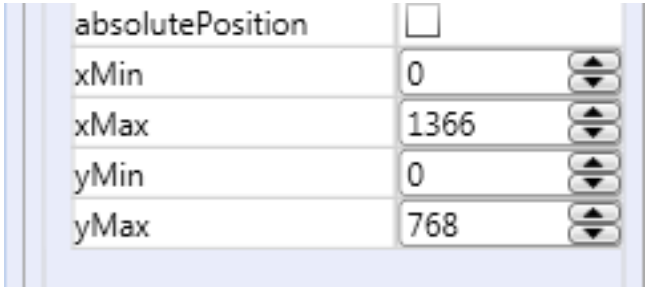
- AsTeRICS can be downloaded from the official site: http://www.asterics.org or https://github.com/asterics/AsTeRICS/releases
- The latest source code is available at GitHub https://github.com/asterics/AsTeRICS
- ACS is the graphical editor, which sends models with connected plugins to the ARE (the runtime environment)

AsTeRICS approach: Graphical Model Design

AsTeRICS Configuration Suite (ACS)

- Connect sensor, processor and actuator modules with signal paths
- Define properties and event conditions





AsTeRICS Runtime Environment (ARE)

- Run model configuration
- Interface sensors and actuators
- Process data

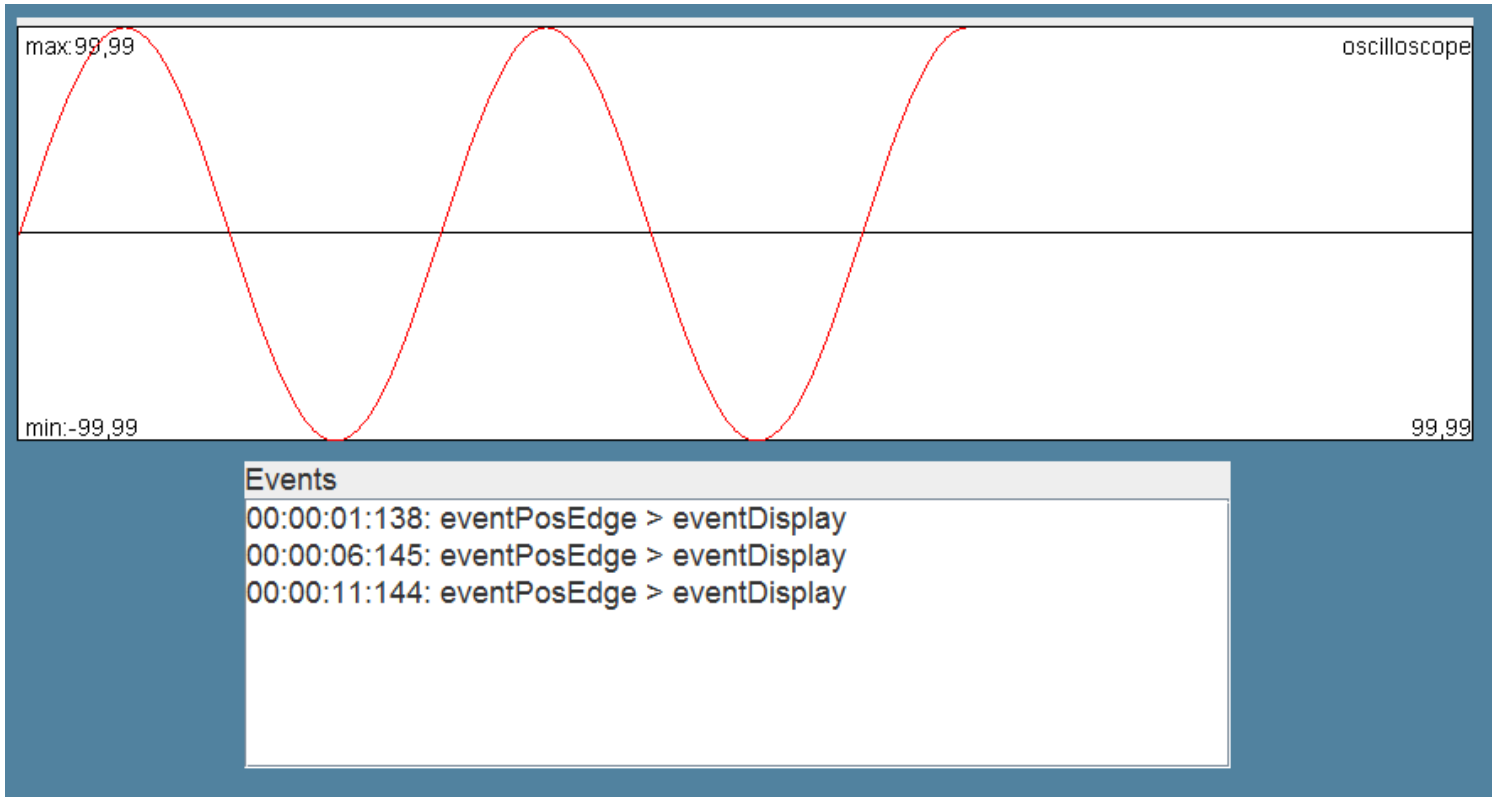
The Basics

- Documentation is avaliable:
  - Use **plugin search bar** in „Components“ tab
  - Read **tooltips** displayed when hovering over graphical elements in ACS (plugin, ports, channels, properties,…)
  - Use **F1** in ACS when plugin is selected to display plugin-help !
  - User Manual with step-by-step model creation guide
  - Developer manual for creating new plugins

Let‘s start ACS and ARE and have a look !

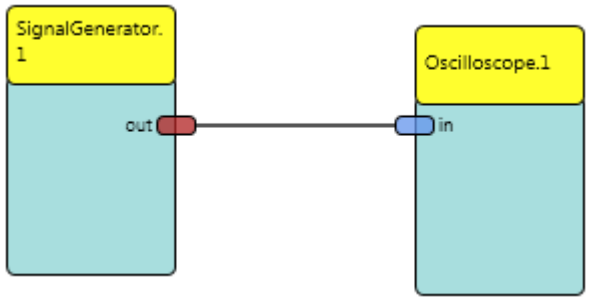
Example 1: SignalGenerator, Oscilloscope

- How to generate signals and display them
- How to trigger event above threshold value



Example 1: SignalGenerator, Oscilloscope

- Create a new model in the ACS (System → **New Model**)
- Insert the „**SignalGenerator**“ sensor component (Components → Sensors → Simulation → SignalGenerator)
- Insert the „**Oscilloscope**“ actuator (Components → Actuators → Graphical User Interface →Oscilloscope)
- Connect port „out“ to „in“



Example 1: SignalGenerator, Oscilloscope

SignalGenerator properties

- Change frequency to **0.2 Hz**
- Also consider sendInterval, (sample rate [ms]), amplitude, phaseShift or offset

Properties	
sendInterval	20
waveForm	sine
frequency	0.2
amplitude	100
phaseShift	0
offset	0

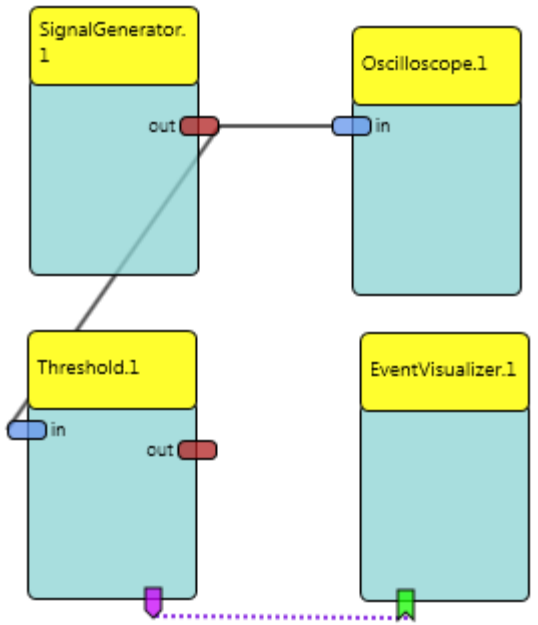
Oscilloscope properties

- Use default properties

Properties	
displayBuffer	3
drawingMode	autoupdate min and max
displayMode	redraw on incoming sample
drawingInterval	100
min	-100
max	100
gridColor	black
channelColor	red
backgroundColor	white
fontSize	14
caption	oscilloscope
displayGUI	☒

Example 1: SignalGenerator, Oscilloscope

- Insert the „**Threshold**“ component
  (Components → Processors→ Basic Math →Threshold)
- Insert the „**EventVisualizer**“ actuator
  (Components → Actuators → Graphical User Interface
  →EventVisualizer)
- Connect port „out“ to „in“ of Threshold



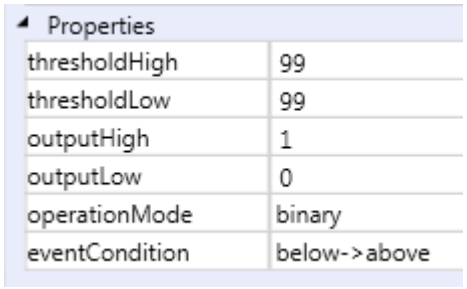
Example 1: SignalGenerator, Oscilloscope

Create Event Condition

- Connect event trigger of Threshold plugin to event listener of EventVisualizer
- Assign “eventPosEdge” to “eventDisplay”

Threshold properties

- Set thresholdHigh and thresholdLow to **99**



Example 1: SignalGenerator, Oscilloscope

- Insert the „**WavefilePlayer**“ component (Components → Actuators → Audio and Voice)
- Connect event trigger of Threshold plugin to event listener of WavefilePlayer
- Assign “eventPosEdge” to “Start”

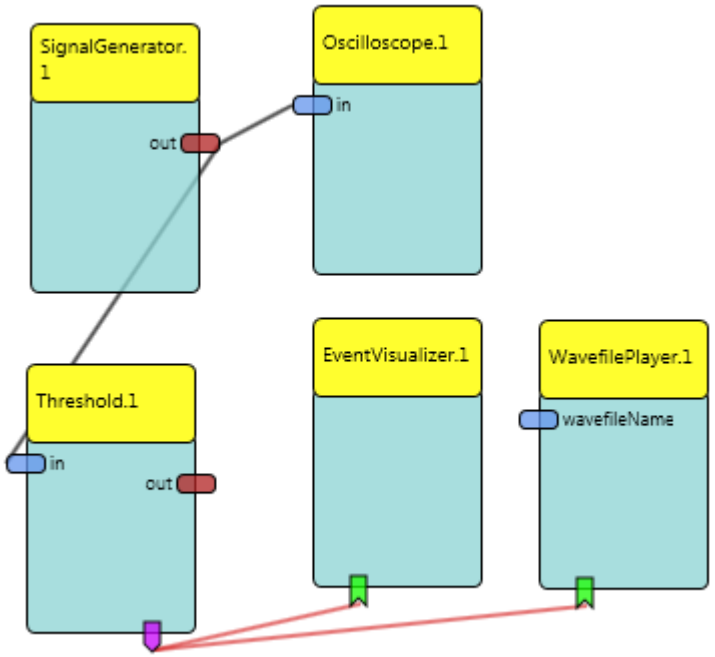
Events (Ctrl-E)

WavefilePlayer.1	Threshold.1
Start	eventPosEdge
Start	---

- Select wave file to play in WavefilePlayer properties

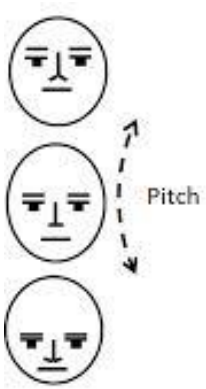
Properties

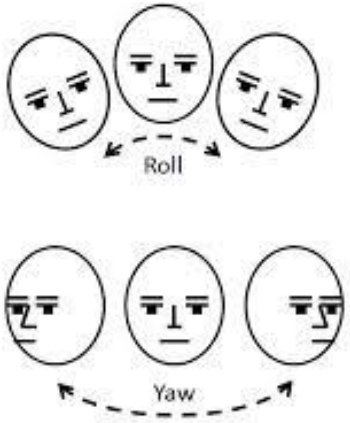
filename	data\sounds\applause.wav

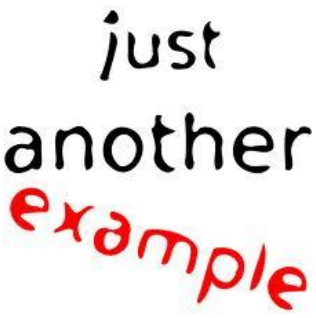


Example 2: Mouse-Control by Head Movements

- In this example we will show how you can control the mouse cursor via head movements
- We use a webcam and the FacetrackerLK sensor plugin (which tracks face movements)
- Mouse clicks are generated via a period of inactivity (no movements)







Note: you find a similar description in the AsteRICS User manual

Example 2: Mouse-Control - Cursor Movement

How to provide headtracking-controlled mouse alternatives for computer input

- The x- and y-position of the local mouse will be controlled by the user‘s head movement

Requirements:

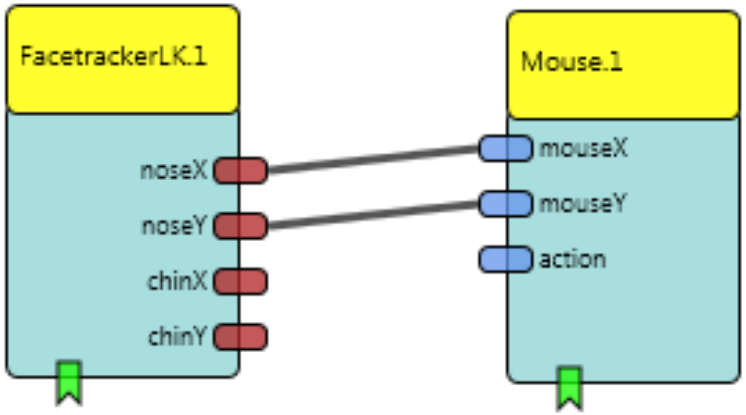
- A webcam

Remarks:

- Camera position directly in front of the user, distance 70-120cm
- No other persons face should be in the field-of-view of the camera

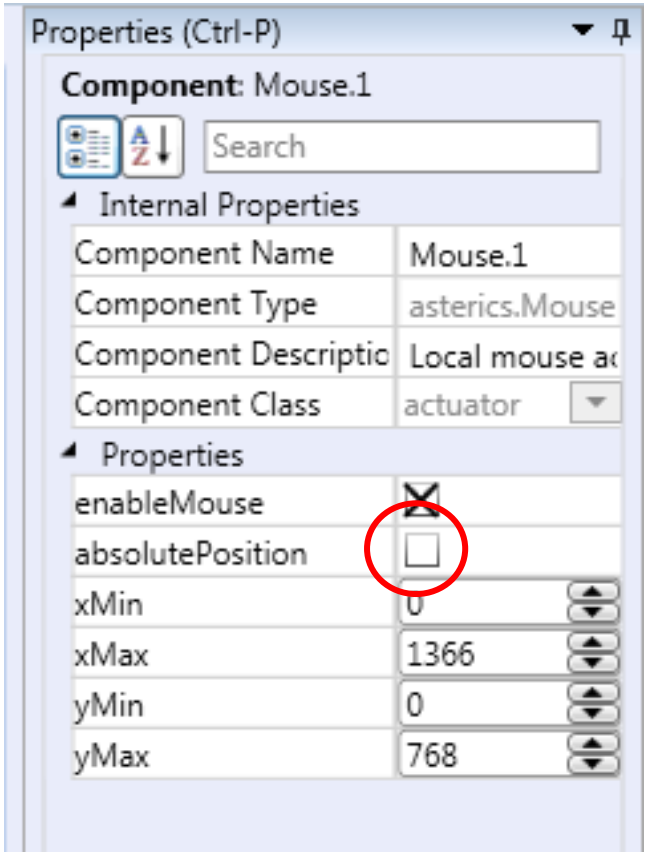
Example 2: Mouse-Control - Cursor Movement

- Create a new model in the ACS (System → **New Model**)
- Insert the „**FacetrackerLK**“ sensor component
  (Components → Sensors → ComputerVision → FacetrackerLK)
- Insert the „**Mouse**“ actuator
  (Components → Actuators → Input Device Emulation → Mouse)
- Connect ports „noseX“ to „mouseX“ and „noseY“ to „mouseY“



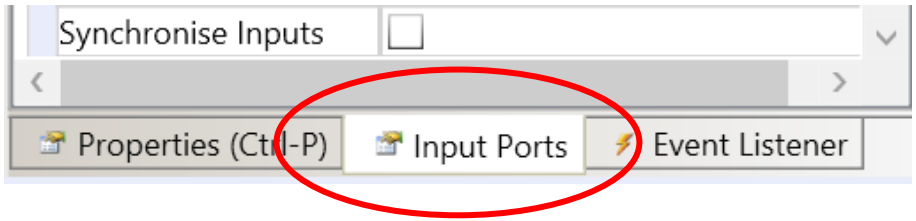
Example 2: Mouse-Control - Cursor Movement

- Mouse plugin properties:
  - adjust **xMax** and **yMax** to the desired values, e.g. screen resolution
  - enter 0 for xMax and yMax to enable automatic detection of screen resolution
  - **deselect** „absolutePostition“ this defines that the X and Y input values are relative changes (The FacetrackerLK plugin only tracks relative movements)



Example 2: Mouse-Control - Cursor Movement

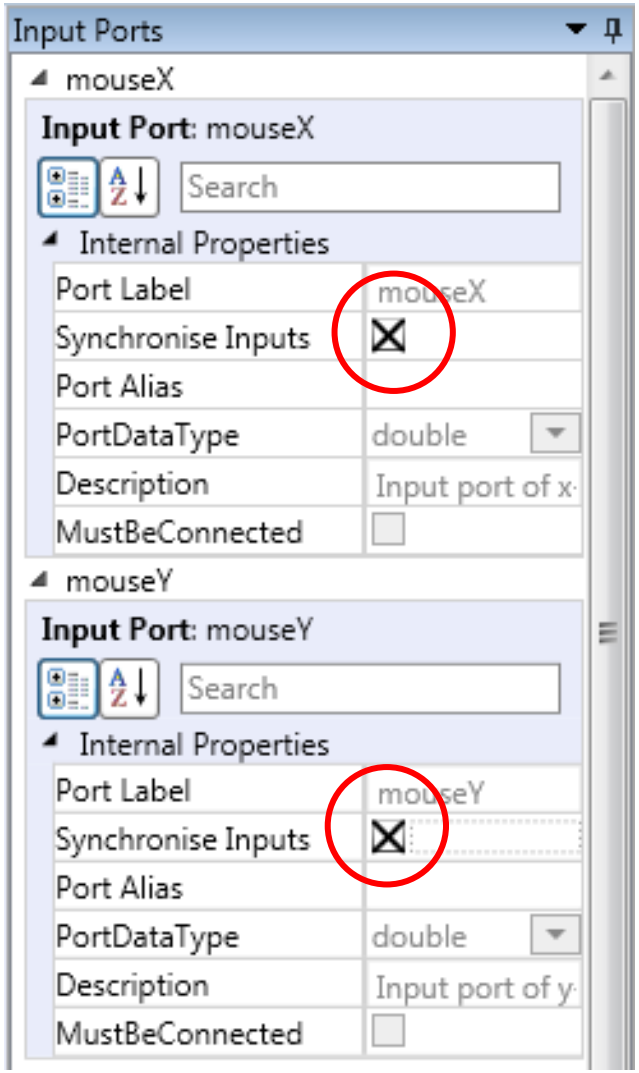
- Mouse plugin, Input Port tab:



  - Select „**Synchronize Inputs**“ option in the Input Port tabs for both inputs (mouseX and mouseY)

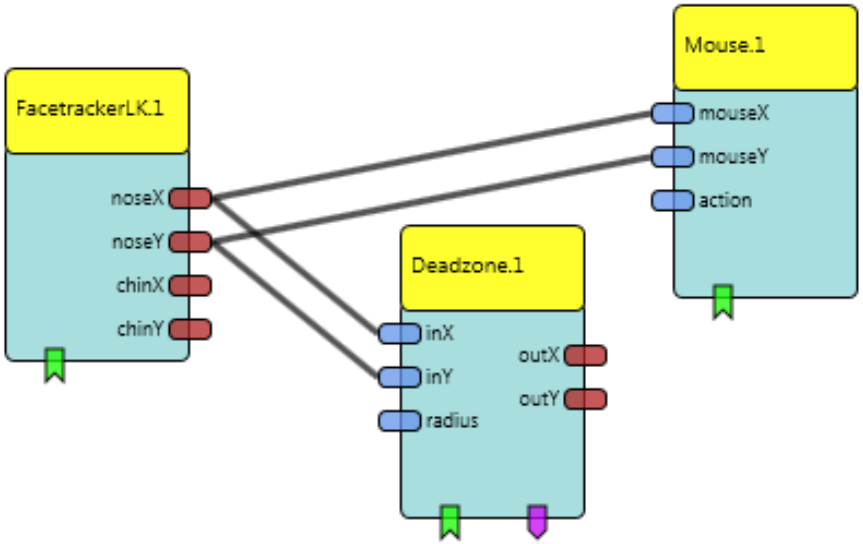
    → this will wait for both input coordinates to arrive before the mouse position is updated

- You can try out your model by uploading it to the ARE and clicking „Start Model“



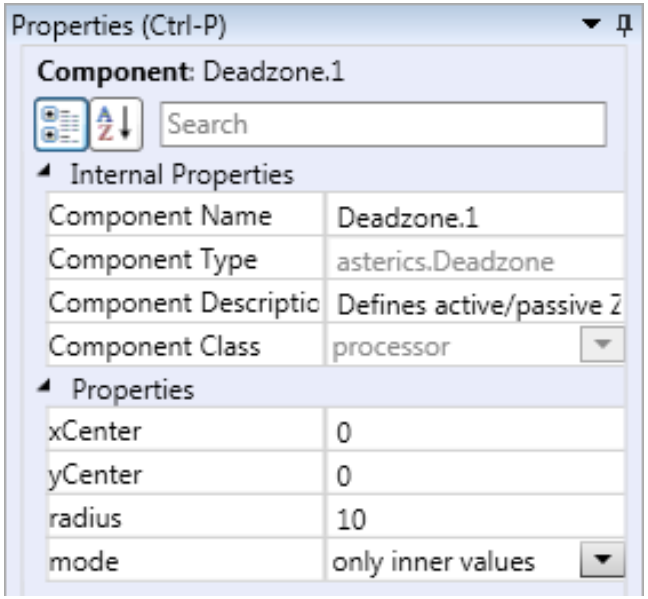
Example 2: Mouse-Control – Add Dwell Click

- No clicking function implemented yet
- → Further development: Add **dwell clicking**
- Adding processing component „**Deadzone**“ (Components – Processors – Signal Shaping – Deadzone)
- Connect noseX/noseY outputs of the FacetrackerLK to the inX/inY inputs of the Deadzone



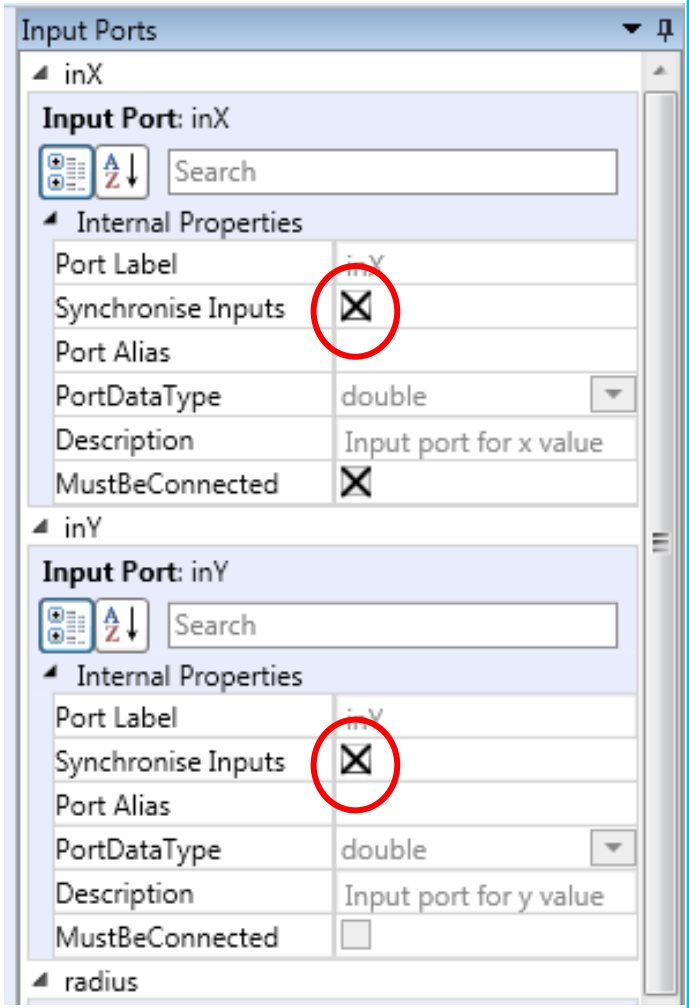
Example 2: Mouse-Control – Add Dwell Click

- Use the **Deadzone** component to define a desired movement level to start or stop the timing for the dwell click
- Deadzone component fade out x/y signal values in an adjustable range and generate event trigger if the x/y values are in- or outside this range
- Parameter „**radius**“ defines this range
  → here it is the amount of
  nose movement
- Leave the radius at the
  default value of 10



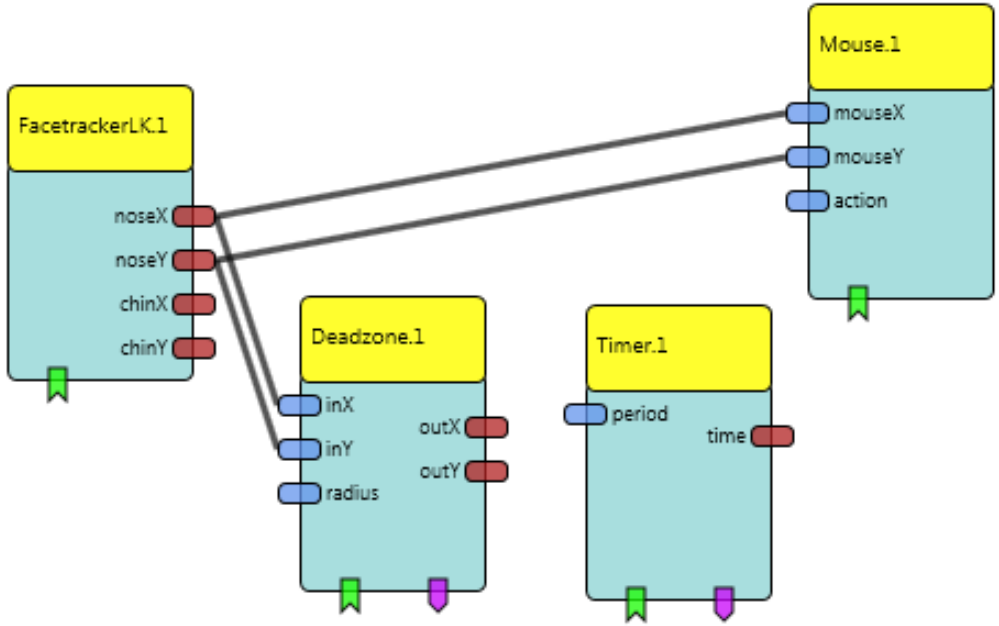
Example 2: Mouse-Control – Add Dwell Click

- Default value 10 – movement range is set to 10 pixels from previous to current position
- Select „**Synchronize Inputs**“ option in the Input Port Riders of the Deadzone plugin for inX and inY



Example 2: Mouse-Control – Add Dwell Click

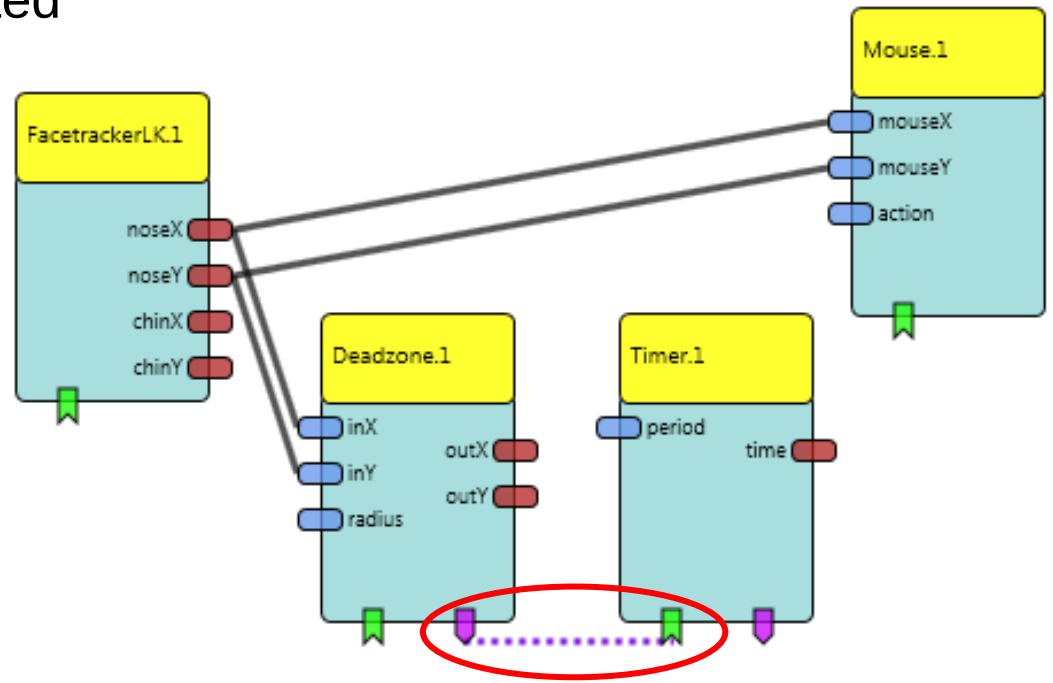
- How can we measure a certain timespan of low movement (to generate a dwell click) ?
- → Insert a **Timer** sensor component (Sensors – Simulation – Timer)
  - this component measures time, generates events if a time period has passed, perfoms timing loops
- Set time-period to 1000 ms in the component's properties



Example 2: Mouse-Control – Add Dwell Click

- Connect event trigger port of the Deadzone component (purple) to the event listener of the Timer component (green)
- Click on the event channel (connection line is now purple dotted)

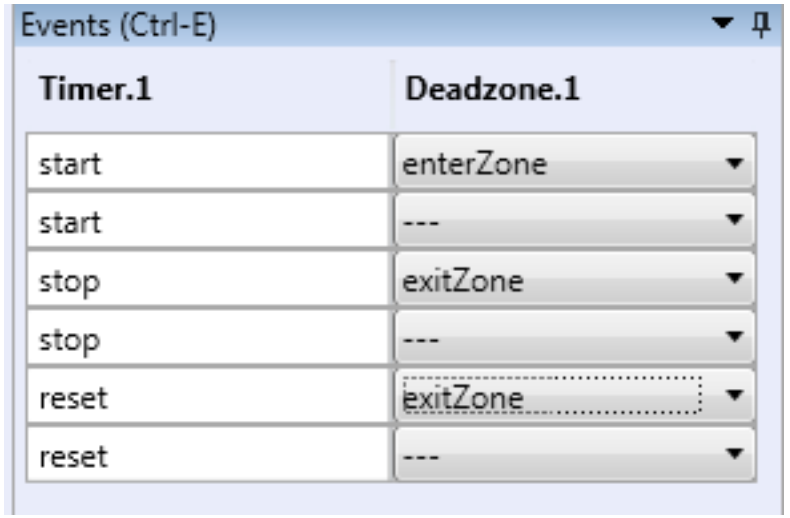
→ Events can be selected from dropdown menu



Example 2: Mouse-Control – Add Dwell Click

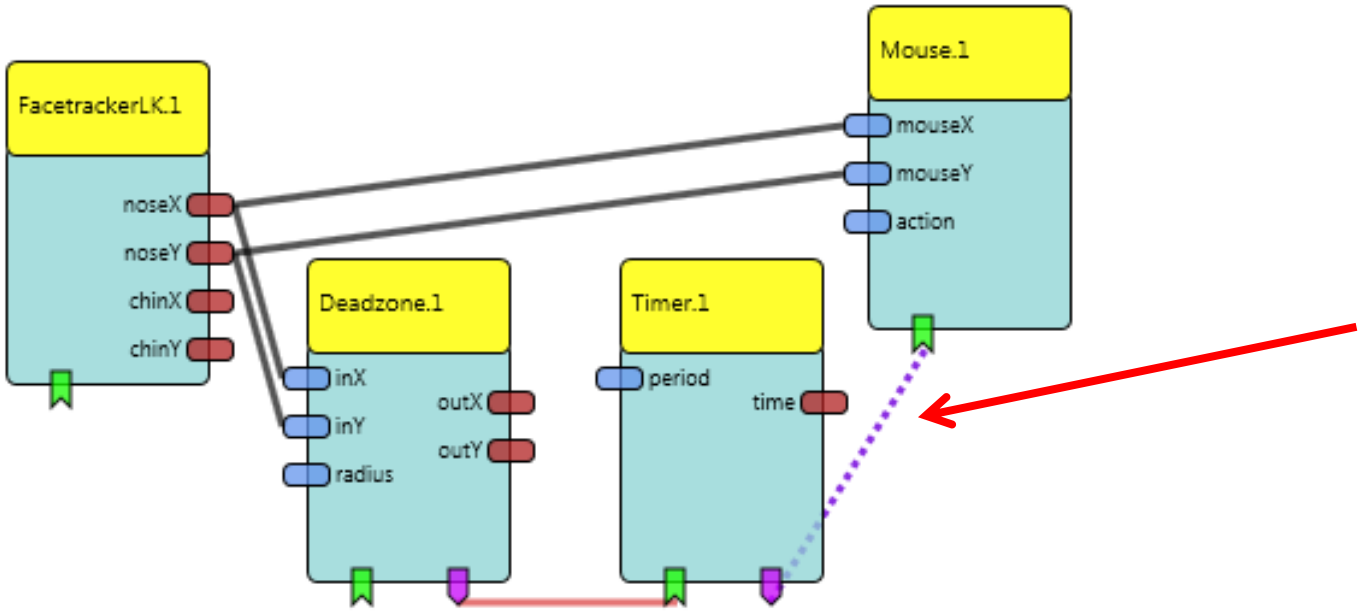
- Select „**enterZone**“ event for the „start“ function
- Select „**exitZone**“ event for the „stop“ and „reset“ function

  - These event connections control the Timer components
  - **If nose movements stay below** selected level of **10 pixels**, the **Timer** is **started**
  - **Else**, the **Timer** is **resetted** to 0 and stopped
  - **If** the **movement stays low** for the full time period, the timer will generate its „**periodFinished**“ **event**.



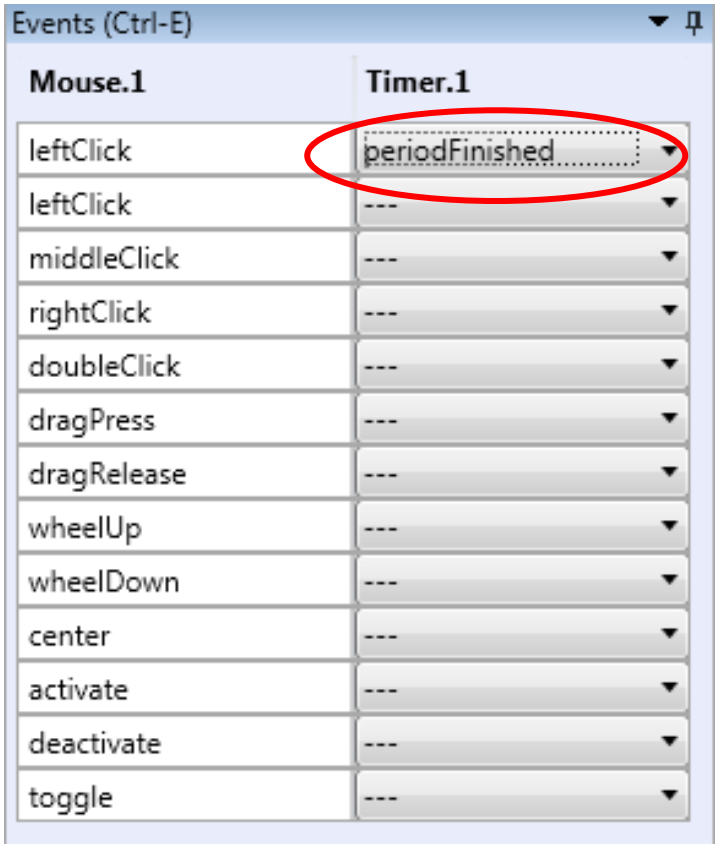
Example 2: Mouse-Control – Add Dwell Click

- Draw a channel from the Timer‘s event trigger port (purple) to the event listener port of the Mouse (green)



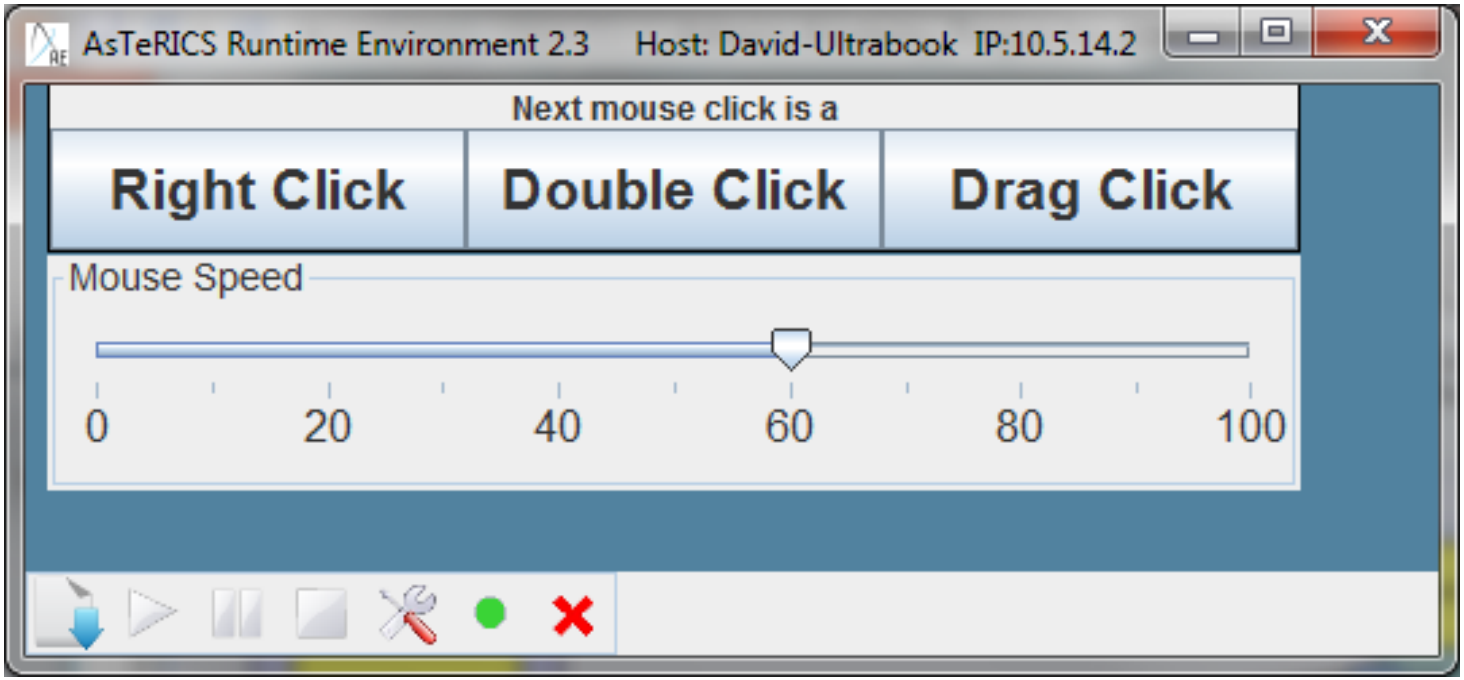
Example 2: Mouse-Control – Add Dwell Click

- By clicking on the new channel (line is now purple dotted), the Events can be adjusted
- Assign the „periodFinished“ event to the „leftClick“ function
- After these settings, the model is usable and provides left-click



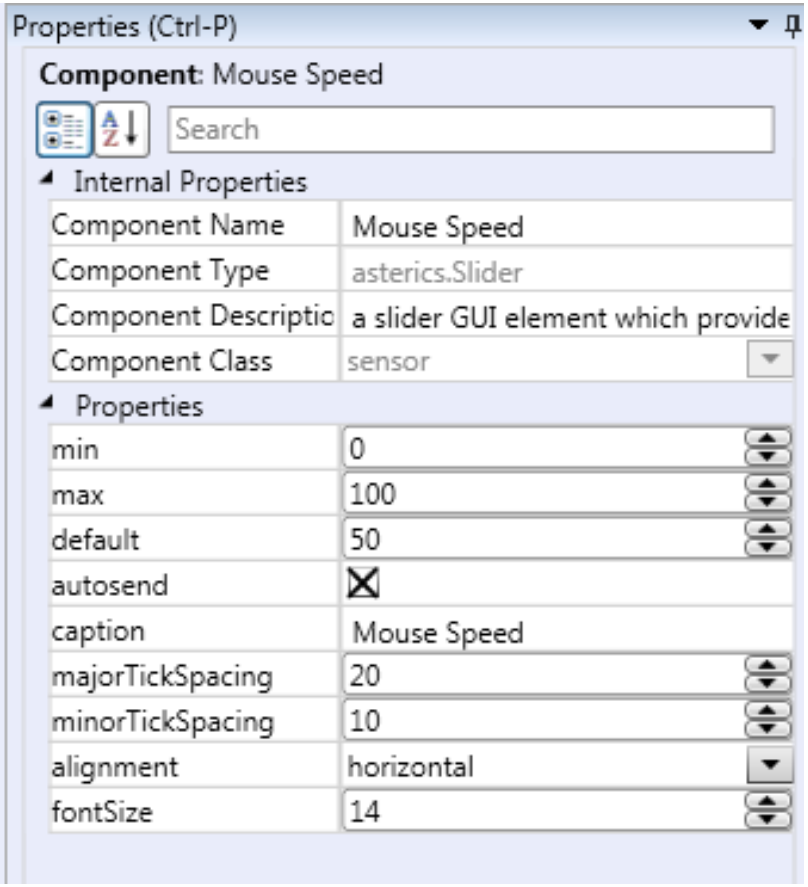
Example 2: Mouse-Control – Add GUI

Include GUI elements for adjustable mouse acceleration and dwell timing as well as different click-actions



Example 2: Mouse-Control – Add GUI

- Add a **Slider** component (Components – Sensors – Graphical User Interface – Slider)
- Slider properties:
  - range of value can be defined (we can leave it at 0-100)
  - set slider's component name to "Mouse Speed"
  - set minorTickSpacing to "10"



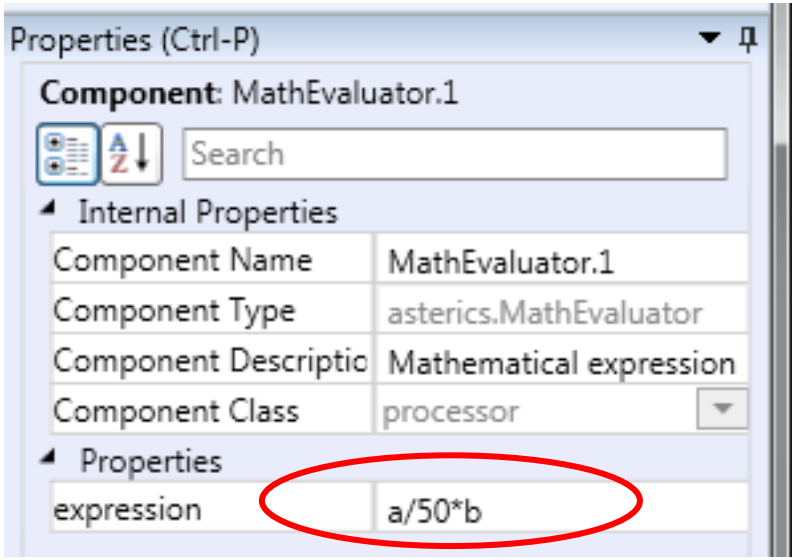
Example 2: Mouse-Control – Add GUI

- To modify the x/y mouse speed with the slider's values, a **MathEvaluator** processing component is needed (Components - Processors – Basic Math – MathEvaluator)
- First, the x-signal is modified:
  - Delete port connection from noseX to mouseX
  - Draw new port connection from "value" (Slider) to "inA" (MathEvaluator)
  - Draw new connection from noseX (Facetracker) to "inB" (MathEvaluator)
  - Draw a new connection from output port (MathEvaluator) to mouseX input port

Example 2: Mouse-Control – Add GUI

- MathEvaluator properties:
  - Adjust "**expression**" property of the MathEvaluator – this defines what will be done with the inputs – in our case we will multiply inA and inB
  - Slider **position < 50** shall **slow down mouse** speed, slider **position > 50** shall **increase mouse** speed

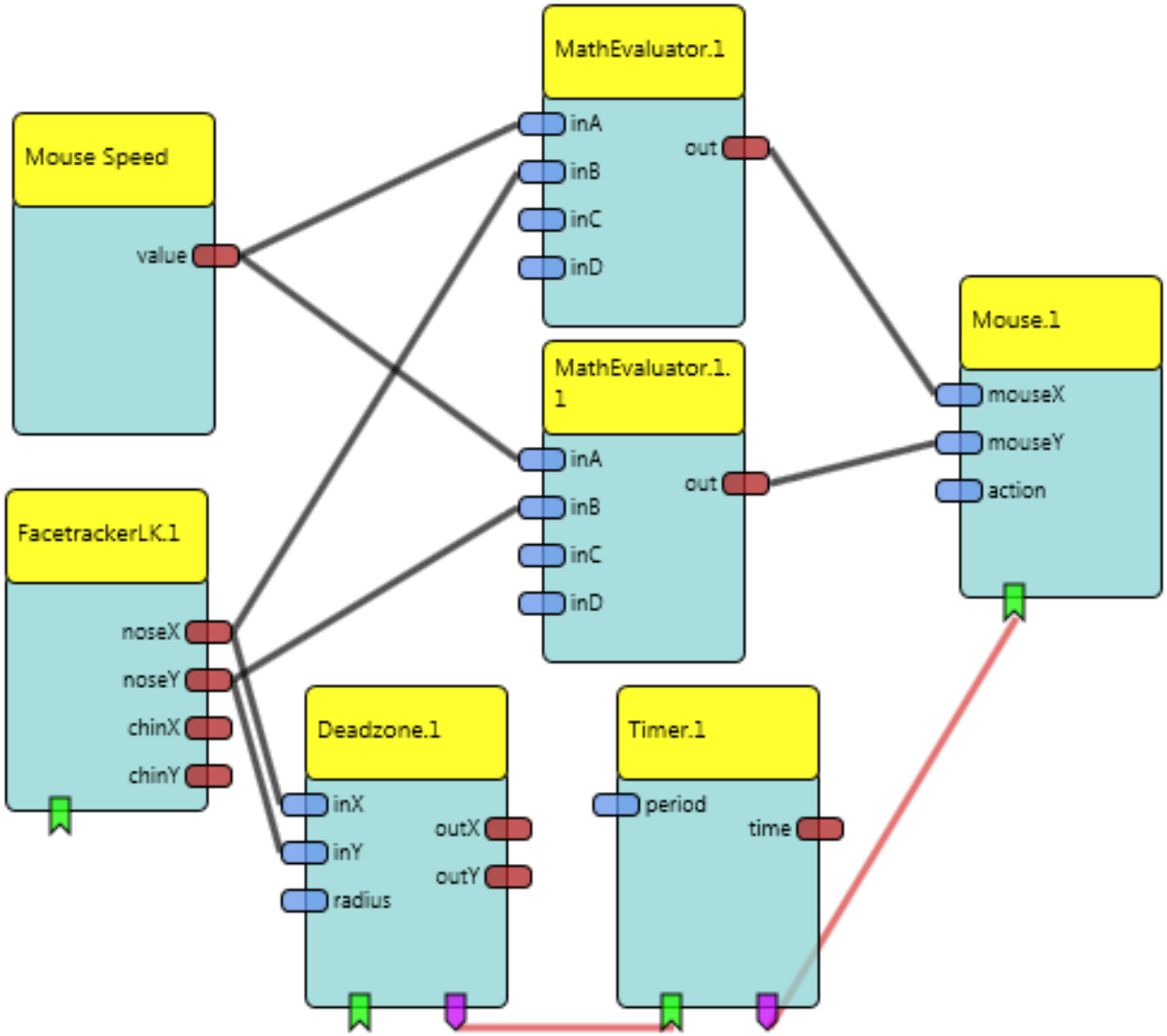
→ a/50*b



Example 2: Mouse-Control – Add GUI

- For modifying the y-signal, copy and paste (Ctrl+C & Ctrl+V) the MathEvaluator
- New connections as for the x-direction:
  - Delete port connection from noseY to mouseY
  - Draw new connection from “value” (Slider) to “inA” of the second MathEvaluator
  - Draw new connection from noseY (Facetracker) to “inB” of the second MathEvaluator
  - Draw a new connection from output port of the second MathEvaluator to mouseY input port

Example 2: Mouse-Control – Add GUI



Example 2: Mouse-Control – Add GUI

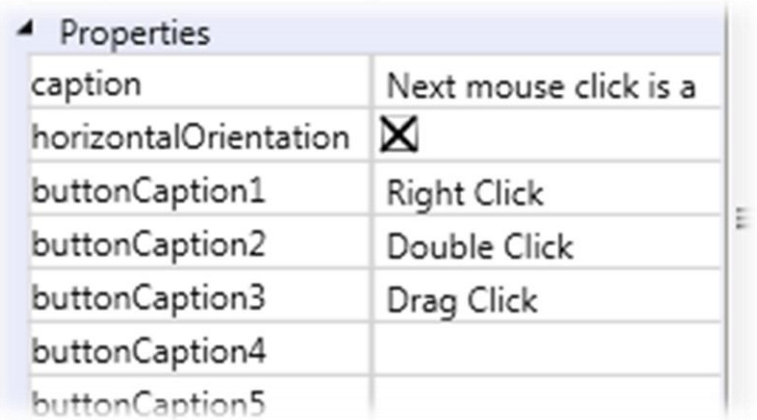
Adding different mouse click activities via GUI by

- adding a **ButtonGrid**
  - to select next click type
- informing mouse element about the next desired mouse click
  - by sending an “action string” to the Mouse element

  Action strings contain commands which are understood by a number of specialized actuator elements. These string contain the addressed component and the desired command
  e.g. “@MOUSE:nextclick,right”

Example 2: Mouse-Control – Add GUI

- Add the **ButtonGrid** component (Component – Sensors – Graphical User Interface – ButtonGrid)
- ButtonGrid properties:
  - Set “buttonCaption” properties of button 1, 2 and 3 to
    - RightClick
    - DoubleClick
    - DragClick
  - Choose “horizontalOrientation” for the ButtonGrid
  - Set a desired caption, e.g. “Next click is a”



Example 2: Mouse-Control – Add GUI

- Add the **StringDispatcher** component (Component – Processors – Event and String Processing – StringDispatcher)
  - Translates incoming events into outgoing strings
  - If buttons are pressed, desired action strings are generated for the Mouse components
- Connect event trigger port of ButtonGrid (purple) to the event listener port of the StringDispatcher (green)
- Click on the event channel and attach
  - button1 to dispatchSlot1
  - button2 to dispatchSlot2
  - button3 to dispatchSlot3

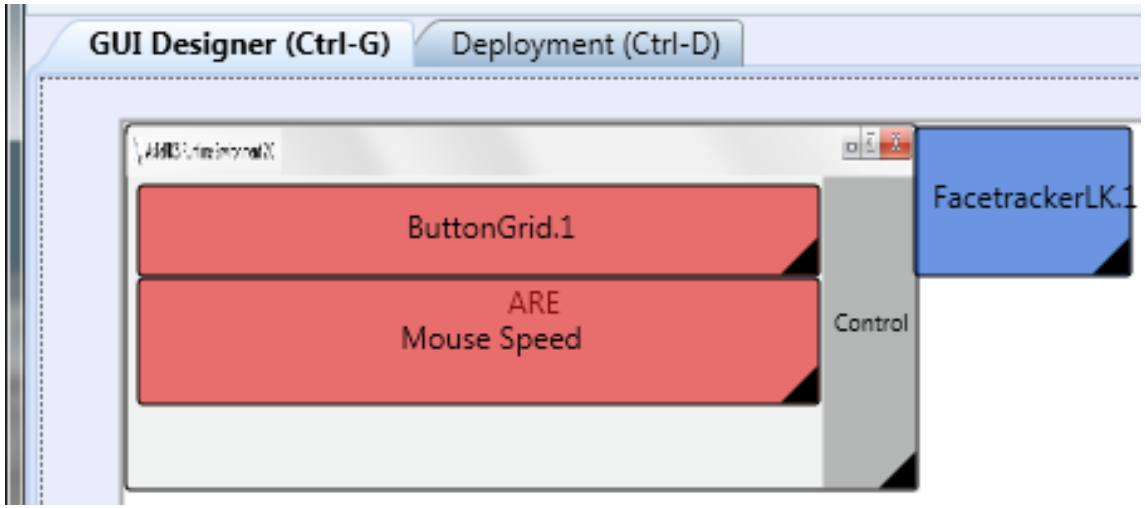
Example 2: Mouse-Control – Add GUI

- Define the strings for slot1-slot3 in the StringDispatcher properties:
  - slot1(button1): “@MOUSE:nextclick,right”
  - slot2(button2): “@MOUSE:nextclick,double”
  - slot3(button3): “@MOUSE:nextclick,drag”

- Connect output port of the StringDispatcher to the “action” input port of the mouse

Example 2: Mouse-Control – Add GUI

- Change to GUI Designer tab to define a desired position for the ButtonGrid
  - Possible positions:

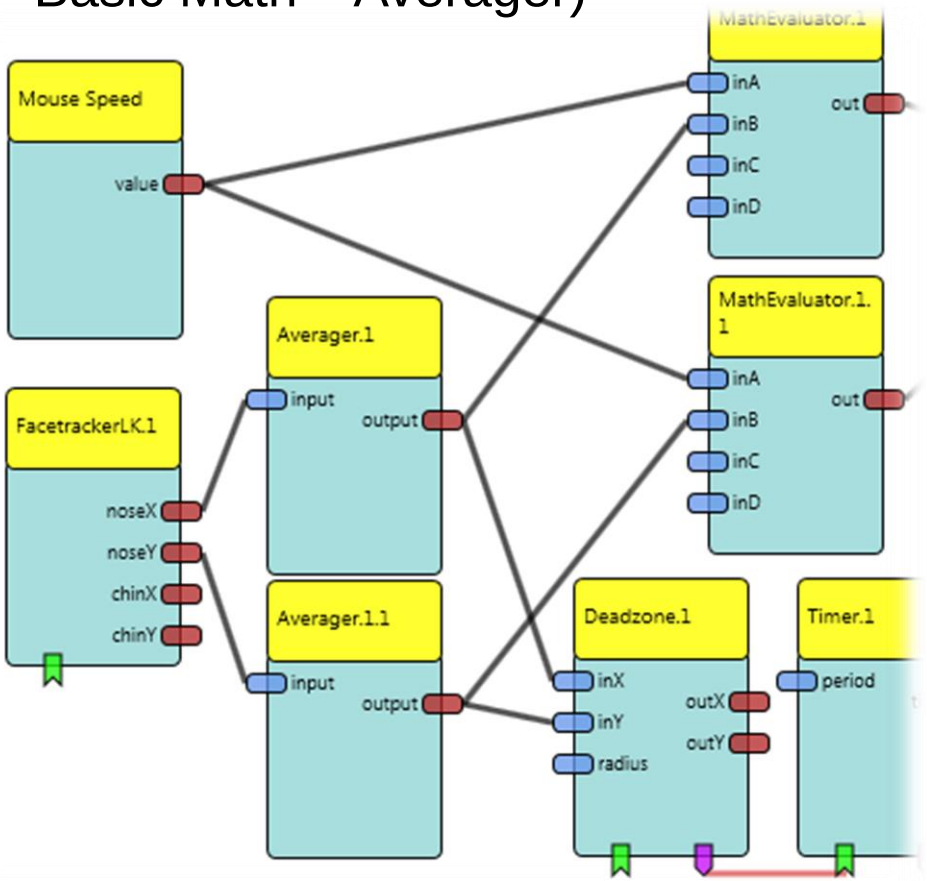


- ”Upload Model” and “Start Model” to try it out!

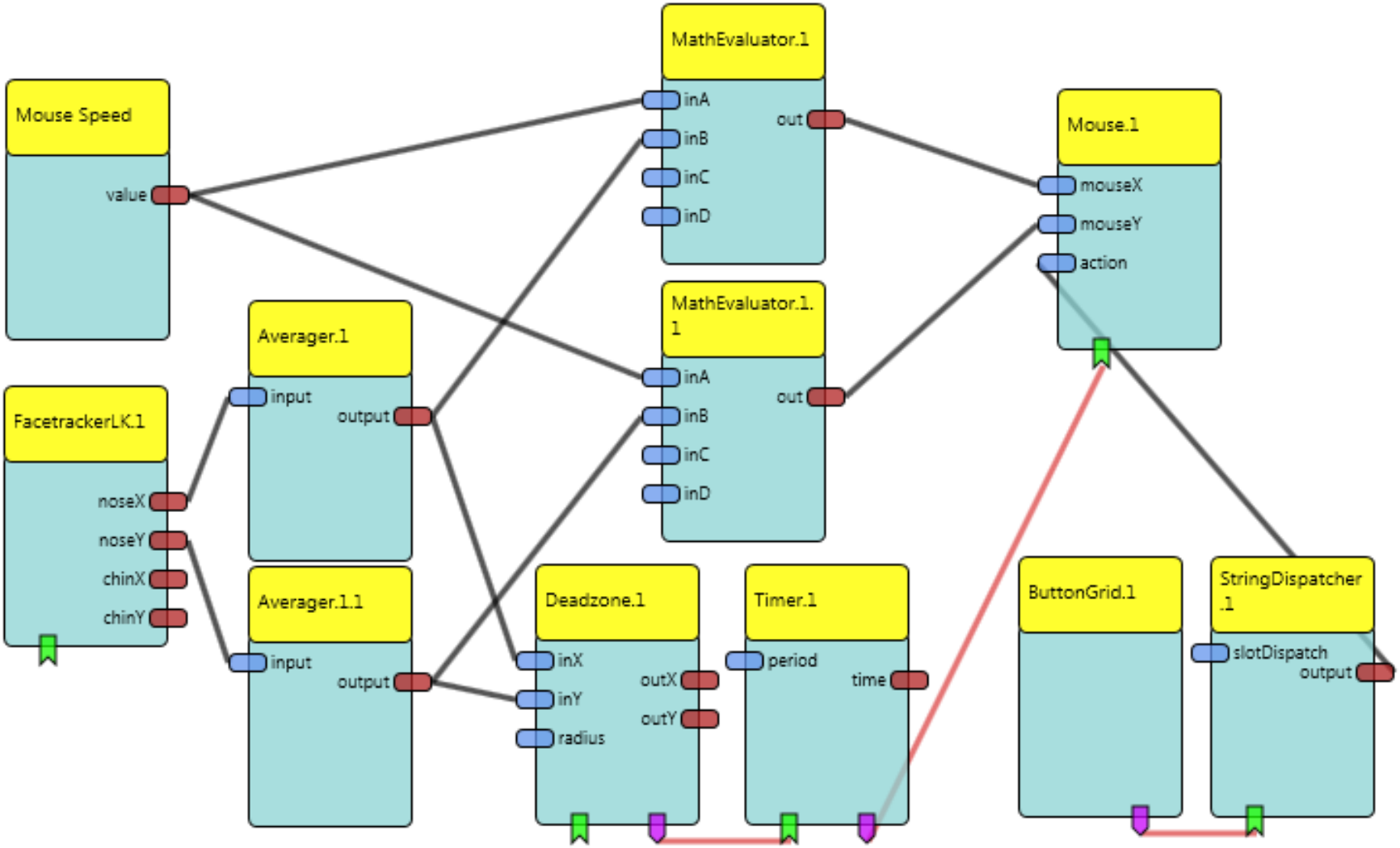
Example 2: Mouse-Control – Improve Jittering

Further improvement tip

- If the mouse cursor is jittering a lot, use an averager (Components – Processors – Basic Math – Averager) with a BufferSize of 5 after Facetracker Outputs



Example 2: Mouse-Control – Improve Jittering



Some (other) useful plugins

- Sensors
  - ComputerVision: (X)FaceTrackerLK, EyeTribe, EyeX, KinectJ4K, TuioReactivision
  - Key/Mouse: KeyCapture, MouseCapture
  - Generic: DigitialIn, AnalogIn, MicGPI
  - GUI: ButtonGrid, Slider, Cellboard
  - Bio-Signal: p2_parser
  - Intertial: Acceleration, WiiMote
  - Simulation: SignalGenerator, Timer
  - Smartphone Sensors (Touch): OscServer
  - Filesystem: LineReader, ReadEDF
- Actuators
  - Audio: MidiPlayer, WavefilePlayer, MediaPlayer
  - TTS:SpeechProcessor/SyntheticVoice
  - Generic: DigitalOut, serialSender
  - GUI: Oscilloscope, BarDisplay, EventVisualizer, TextDisplay
  - HomeControl: FS20Sender, IrTrans
  - InputDeviceEmulation: Mouse, Keyboard
  - Filesystem: ApplicationLauncher, modelSwitcher, LineWriter, WriteEDF
  - Gaming: PongGame
  - PS3 Controller Emulation (HidActuator): RemoteMouse, Remote Keyboard, RemoteJoystick

* Needs dedicated HW or SW / driver

Some (other) useful plugins

- Processors
  - Audio&Voice: SpeechProcessor
  - BasicMath: MathEvaluator, Threshold, Averager, ConstantDispatcher, Comparator, Differentiate / Integrate
  - SignalShaping: Deadzone, SignalTranslation
  - DSP, Filter: IIRFilter
  - Event&StringProcessing: StringDispatcher, EventDispatcher, EventFlipFlop
  - Microcontroller: Arduino
  - OSKA
  - SignalPathways: PathSelector, MultiSource

* Needs dedicated HW or SW / driver