

# AsTeRICS Deliverable D4.6b

## Final Prototype of Signal Processing Modules - Plugins

FHTW

## Document Information

Issue Date	30 June 2012
Deliverable Number	D4.6
WP Number	WP4 Software
Status	Final
Dissemination Level	CO PU Public PP Restricted to other programme participants (including the Commission Services) RE Restricted to a group specified by the consortium (including the Commission Services) CO Confidential, only for members of the consortium (including the Commission Services)

### **AsTeRICS – Assistive Technology Rapid Integration & Construction Set**

Grant Agreement No.247730

ICT-2009.7.2 - Accessible and Assistive ICT

Small or medium-scale focused research project

## Disclaimer

The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

The document reflects only the author's views and the Community is not liable for any use that may be made of the information contained therein.

## Version History

<b>Version</b>	<b>Date</b>	<b>Changed</b>	<b>Author(s)</b>
0.1	05/06/2012	First draft	Chris Veigl (FHTW)
0.2	13/06/2012	Structure with new / updated plugins provided	Chris Veigl (FHTW)
0.3	18/06/2012	Added KI-I's plugin contributions and updates	David Thaller (KI-I)
0.4	18/06/2012	Added Model descriptions and differences to the D4.3 plugin set	Chris Veigl (FHTW)
0.5	18/06/2012	Added Starlab's plugin descriptions and updates	Aureli Soria Frisch (Starlab)
0.6	18/06/2012	Added Harpo's plugin descriptions and updates	Karol Pecyna (HARPO)
0.7	18/06/2012	Integrated all contributions, reworked some chapters, some minor corrections	Chris Veigl (FHTW)
0.8	25/06/2012	Worked in peer review comments	Chris Veigl (FHTW)

## Table of Content

Document Information .....	2
Version History .....	3
Table of Content.....	4
Glossary .....	7
1 Introduction .....	8
2 ARE Plugins .....	9
2.1 Plugin Summary.....	10
2.1.1 AsTeRICS Sensor Plugins .....	10
2.1.2 AsTeRICS Processor Plugins.....	12
2.1.3 AsTeRICS Actuator Plugins .....	13
2.2 Sensor Plugin Updates.....	15
2.2.1 AutostartEvent.....	15
2.2.2 AnalogIn.....	15
2.2.3 ButtonGrid.....	16
2.2.4 CellBoard .....	16
2.2.5 DigitalIn.....	19
2.2.6 Enobio.....	19
2.2.7 Eyetracker.....	20
2.2.7.1 Eyetracker Modes .....	21
2.2.7.2 The calibration procedure.....	22
2.2.7.3 Eyetracker plugin requirements, ports and properties.....	22
2.2.8 FacetrackerLK.....	24
2.2.9 FacetackerCLM.....	25
2.2.9.1 Opened/Closed eyes training tools.....	27
2.2.10 FS20 Receiver .....	29
2.2.11 Kinect.....	31
2.2.12 OpenVibe .....	32
2.2.13 PlatformDigitalIn.....	33
2.2.14 RFID Reader .....	34
2.2.15 SignalShaper .....	35
2.2.16 WiiMote.....	35
2.3 Processor Plugin Updates.....	38

2.3.1	Arduino .....	38
2.3.2	BasicTRAlgorithms.....	39
2.3.3	Blink Detection .....	40
2.3.4	BlinkDetectionTraining .....	41
2.3.5	DoubleToString .....	42
2.3.6	EventBlock .....	42
2.3.7	EventDelay.....	42
2.3.8	Filter.....	43
2.3.9	FS20 Command Interpreter.....	44
2.3.10	IntToString .....	45
2.3.11	NeuralNetworkLoader .....	45
2.3.12	OneEventManyActions.....	45
2.3.13	OskalInternalScanning .....	46
2.3.14	OskaExternalScanning1D .....	47
2.3.15	OskaExternalScanning2D .....	49
2.3.16	MultiSource .....	50
2.3.17	MultiSourceString.....	51
2.3.18	Quantizer .....	51
2.3.19	RegularExpression.....	51
2.3.20	RelativeMoveSampler .....	52
2.3.21	Sampler.....	52
2.3.22	StringPathMultiplexer .....	53
2.3.23	StringPathSelector .....	53
2.3.24	SSVEP .....	54
2.3.25	SpeechProcessor.....	54
2.3.26	StringToDouble .....	56
2.3.27	StringToInt .....	56
2.4	Actuator Plugin Updates.....	57
2.4.1	AndroidPhoneControl.....	57
2.4.2	ApplicationLauncher.....	58
2.4.3	DigitalOut .....	59
2.4.4	FS20 Sender.....	59
2.4.5	GSMModem.....	61
2.4.6	ImageBox.....	62
2.4.7	MidiPlayer .....	62
2.4.8	ModelSwitcher.....	63
2.4.9	PlatformDigitalOut .....	63
2.4.10	PlatformLCD .....	64
2.4.11	RemoteTablet .....	64
2.4.12	SyntheticVoice .....	66

- 2.5 Other Changes in the Plugin Implementations.....67
- 3 Implemented AsTeRICS Configuration Suite (ACS) Examples.....68
  - 3.1 A Camera-controlled Mouse with GUI .....68
  - 3.2 A Sip/Puff-controlled Mouse with Clicking Functions .....70
  - 3.3 Mouse Control by One Switch .....71
  - 3.4 Send and Receive SMS using a GSM modem .....72
  - 3.5 Accessible Gaming Configurations .....73
    - 3.5.1 Accessible “Marvel” Pinball .....73
    - 3.5.2 An Accessible Racing Game: “ModNation Racer” .....74
  - 3.6 Additional Demo Scenarios and Use Cases .....75
- References.....76

## Glossary

ACS	AsTeRICS Configuration Suite
ADC	Analogue to Digital Converter
ARE	AsTeRICS Runtime Environment
ASM	Active Shape Model
CIM	Communication Interface Module
CSP	Common Spatial Patterns
DAC	Digital to Analogue
EEG	Electroencephalogram
EMG	Electromyogram
EOG	Electrooculogram
FPR	False Positive Rate
GPIO	General Purpose Input and Output
HID	Human Interface Device
IMU	Inertial Measurement Unit
KNN	K-nearest neighbour
KNX	Konnex Home Automation Standard
LDA	Linear discriminant analysis
OSKA	On Screen Keyboard Application
OVR	One versus the rest
PR	Precision-Recall
ROC	Receiver Operating Curve
SVM	Support vector machine
TPR	True Positive Rate

# 1 Introduction

Deliverable 4.6 describes accomplished works within AsTeRICS project's WP4, which includes software implementation activities. Particularly, the deliverable describes the development of the final AsTeRICS system prototype (Prototype-2) within the following tasks according to the Description of Work (see [1], page 58):

- T4.8 - Signal Processing Framework development and refinement for final Prototype
- T4.9 - Video Signal Processing Components development and refinement for final Prototype
- T4.10 - Development and refinement of Software toolkit for BCI based on machine learning for final Prototype

Due to the strongly heterogeneous contents of D4.6 and the recommendation of the reviewers of deliverable D4.3 (signal processing modules for Prototype-1), D4.6 is split into two parts, which considerably improves the consistency and readability of the deliverable:

- D4.6a – *Algorithms*, which focusses on the scientific work in algorithms for BNCI and Computer Vision (SVM) and covers the tasks T4.9 and T4.10
- D4.6b – *Plugins*, (this document) which describes the software engineering work for ARE-plugins and AsTeRICS models and covers task T4.8

The activity T4.8 includes the adaptation of existing plugins and signal processing functionalities for the AsTeRICS Runtime Environment (ARE), and the development of new plugins according to the results of the user evaluation of Prototype-1. Furthermore, several new plugins had to be developed to provide the intended features of the final system prototype. As in Prototype-1, the basic purpose of the developed plugins is to be included in models for the realization of more complex tasks, which can be configured via the AsTeRICS Configuration Suite (ACS). Section 3 of this deliverable shows several examples of AsTeRICS models using some of the previously described plugins..

Within task T4.9, the Computer Vision algorithms and strategies for the Smart Vision Module (SVM) were refined and evolved. These efforts include the improvement of the webcamera-based ("remote") head tracking algorithms and the development of a new head-mounted camera system which is capable of head-tracking, iris-tracking and gaze estimation. A dedicated hardware for the head-mounted system has been developed which consists of a microcontroller-board, inertial measurement unit (IMU) and 3d-printed mounting parts. The available SVM-features for the final system include head pose estimation, recognition of facial gestures, eye blink detection and integration of inertial sensing data.

Activities in task T4.10 cover the works related to Brain-Neural-Computer Interfaces, respectively the finalisation of the BNCI toolkit and its application. These works have been focused in the BCI paradigms of motor imagery and p300, and SSVEP where special hardware for visual stimulation has been developed. Additionally to integration with BCI2000 [4] and BioSig [3], a complete OpenVibe [5] support for the Enobio system and a bridge from OpenVibe to AsTeRICS has been developed, and interoperability of BNCI functionalities with the TOBI [2] project have been evaluated.



## 2 ARE Plugins

The *plugins* represent the functional blocks of the Assistive Technology Construction Set. They can be combined to tailored Assistive Technology setups via the graphical AsTeRICS Configuration Suite software application (ACS), where plugins can be selected, connected via channels, and parameterized. Thus, the ACS can be considered as a user friendly environment to arrange the plugins. The result of this configuration process is an .xml file containing the deployment model for the AsTeRICS Runtime Environment (ARE).

There are 3 types of plugins: sensor modules, processing modules and actuator modules.

- Sensors include all software modules which make physical or simulated data available to other AsTeRICS plugins. Examples include a digital switch interface, an analogue sip/puff sensor, a webcam or a signal generator.
- Processors include mathematical transformations, feature detectors, threshold level monitors etc. which are necessary to scale or combine signals or detect interesting events in the data streams acquired from the sensors.
- Actuators enable assistive functionalities like mouse- or keyboard replacement, visual or acoustic feedback or direct interaction with the environment, e.g. infrared remote control, home automation or physical manipulation.

New plugins can easily be added by AsTeRICS developers, by following a documented software development approach, utilizing free IDEs (e.g. Eclipse) and the OSGi bundle management framework.

In the following sections, the plugins for the AsTeRICS framework will be presented. A majority of these plugins already existed in the first system prototype and have been described in D4.3b. Several of these plugins have been updated with new functions, and a lot of plugins have been newly developed in the development phase of Prototype-2.

To give a good overview and avoid redundant information, section 2.1 shows all available plugins in a table, indicating updates, additions or removals with respect to D4.3b [8], and adds a subcategory and a short description of every plugin. Subsequently, plugin changes and newly added plugins are described in detail.

Currently, 106 different plugins are available for the final AsTeRICS prototype, offering various sensing, processing and control capabilities. The framework continues to grow due to contributions by AsTeRICS partners and by people from the open source community.

## 2.1 Plugin Summary

The following section presents a list of Sensor-, Processor- and Actuator plugins for the AsTeRICS Runtime Environment – available by delivery date of the final prototype to the European Commission (June 2012).

In total, 106 plugins have been created to provide different assistive features to system developers and end users. About 70 plugins did already exist in the first system prototype and have been described in D4.3. However, many improvements have been implemented in these existing plugins.

To identify the work which has been accomplished in the second software development phase of the AsTeRICS project, the tables of subsections 2.1.1, 2.1.2 and 2.1.3 indicate new plugins (**bold** plugin name) and updated plugins (*Italic* plugin name). In the subsequent sections, only new or considerably updated plugins will be described in detail.

### 2.1.1 AsTeRICS Sensor Plugins

Category	Name	Description	Hardware / Driver Requirements
Bioelectric Measurement	Enobio	Provides 4-channels of bioelectric signals (EEG, EOG, ECG, of EMG) measured with the Enobio wireless biosignal acquisition unit by Starlab. <a href="http://starlab.es/products/enobio">http://starlab.es/products/enobio</a>	Enobio device, drivers installed
	<b>OpenVibe</b>	Delivers data from the OpenVibe BCI framework over a UDP/OSC connection (up to 16 signals and 61 stimulation-codes can be transmitted)	Any OpenVibe-compatible Biosignal acquisition device
Computer Vision	<b>EyeTracker</b>	Provides different eye- and gaze-tracking algorithms suitable for the head-mounted SVM solution (head-gear with camera to transmit live images of one eye, IMU-sensorboard)	Head-mounted SVM including head-gear, camera and IMU sensorboard
	<i>FaceTrackerCLM</i>	Active-Shape model based head tracking algorithm which delivers face landmark positions which are extracted from live images of a webcam.	High quality webcam, drivers installed
	<i>FaceTrackerLK</i>	Very fast head tracking computer vision algorithm, outputs nose and chin coordinates. Drift of the tracking points may occur.	High quality webcam, drivers installed
	<b>Kinect</b>	Provides skeleton joint angle data from a connected Microsoft Kinect sensor	Microsoft Kinect
Generic Control Input	<i>AnalogIn</i>	Delivers 2 channels of voltage- or resistance values from devices connected to the Analog-to-Digital converter module (AnalogIn CIM)	AnalogIn CIM by AsTeRICS partner IMA
	<i>DigitalIn</i>	Delivers 5 channels of digital values of switches connected via standard jack-plugs to the DigitalIn CIM	DigitalIn CIM by AsTeRICS partner IMA
Graphical User Interface	<i>ButtonGrid</i>	A selectable number of buttons for user interaction	None
	<b>CellBoard</b>	Displays a board with the cells which can be selected by mouse action or events. Each cell can contain text and picture. The CellBoard plugin provides basic scanning options	None

Graphical User Interface	EditBox	Edit Box for text input, could be used to e.g. generate speech	None
	Slider	A Slider with adjustable range, to set parameters of a model	None
	TextFieldReader	Edit Box with recognition of up to 7 command strings. A possible application is an interface with Voice Recognition Software	None
Inertial Measurement	Acceleration	3-axis acceleration sensor with wireless ZigBee interface; Delivers x/y/z axis acceleration data.	Wireless Acceleration Sensor by IMA;
	RazorIMU	9 Degree-of-Freedom inertial measurement unit; Delivers pitch, yaw and roll orientation data. <a href="http://www.sparkfun.com/products/9623">http://www.sparkfun.com/products/9623</a>	9 DOF Razor IMU (Sparkfun)
Personal Platform	<i>PlatformDigitalIn</i>	Internal digital input of switches with standard jack-plug connectors which are connected to the AsTeRICS Personal Platform.	AsTeRICS Personal Platform by IMA
Simulation	EventGenerator	Generates simulated events with a selectable period	None
	SignalGenerator	Generates simulated signals (sine, rectangle, sawtooth and random signal) of adjustable amplitude.	None
	SignalShaper	Can be used to compose a simulated signal from basic linear signals	None
Standard Input Devices	JoystickCapture	Captures Joystick movement and button activities of a local gamepad.	Standard (HID) PC-Joystick or Gamepad
	KeyboardCapture	Captures keyboard input and provides key codes to other plugins.	Standard PC-Keyboard
	MouseCapture	Provides movement and button data of the local mouse cursor.	Standard PC-Mouse
	SpaceNavigator 3DMouse	Provides data of the 6-Degree-of-Freedom SpaceNavigator Mouse. <a href="http://www.3dconnexion.com">http://www.3dconnexion.com</a>	3dConnexion Spacemavigator mouse
	WiiMote	Provides movement-, IR-camera- and button- data of the Nintendo WiiMote.	Nintendo WiiMote + Bluetooth connection
Others	AutostartEvent	Sends an event when the model is started after a specified delay	None
	FS20Receiver	Receives commands of the home automation system FS20 for ELV Electronics. Depending on the received commands, events will be fired <a href="http://www.elv.de/FS20-PC-Empfauml;nger-FS20-PCE/">http://www.elv.de/FS20-PC-Empfauml;nger-FS20-PCE/</a>	FS20 Receiver by ELV
	RFIDReader	Can read RFID tags and recognize defined Tag-IDs. <a href="http://www.sparkfun.com/products/8628">http://www.sparkfun.com/products/8628</a>	ID-Innovation RFID-Reader module
	Timer	Measures time and creates periodic events.	None

Table 1: Summary of Sensor Plugins

## 2.1.2 AsTeRICS Processor Plugins

Category	Name	Description
Basic Math	Amplifier	Amplify or attenuate the incoming signal (Gain)
	Averager	Calculate the average (arithmetic mean) of a selectable number of samples
	<i>Comparator</i>	Compare two signals or one signal with a constant, trigger event on changes
	ConstantDispatcher	Delivers one of several constant values to other plugins, values can be selected via events
	Decimation	Reduces the sampling rate of the incoming signal
	Derivative	Calculates the derivative of the incoming signal
	Differentiate	Subtracts the last from the current value of the signal
	Integrate	Calculates a running sum of incoming signals, upper and lower limits can be given
	<i>MathEvaluator</i>	Calculates a given expression containing values of up to 4 signals and standard mathematical functions
	MinMax	Remembers Minimum and Maximum of the connected signal and puts out updates to these
	<b>Quantizer</b>	Performs the quantization of the input signal. The value of the output signal is the input value rounded to the nearest multiple of the <code>quantizationStep</code>
	<b>RelativeMoveSampler</b>	Sums incoming relative positions for three coordinates. The sums are sent to the output ports periodically, according to sampling rate
	SampleAndHold	Remembers the values of up to 4 connected input signals and provides them at the outputs
	<b>Sampler</b>	Provides a constant data rate for the input port's signal
Threshold	Compares the input signal with lower and upper threshold values, creates events when the threshold is passed	
<b>Data Converters</b>	<b>DoubleToString</b>	Converts a double precision floating point input into a string (alphanumeric) output
	<b>IntToString</b>	Converts an integer input into a string output
	<b>StringToDouble</b>	Converts a string input into a double output
	<b>StringToInt</b>	Converts a string input into an integer output
DSP and Feature Extraction	<i>BlinkDetection</i>	Detects Eye-Blinks in Enobio channels of electrophysiological data
	<b>BlinkDetectionTraining</b>	Implements a training protocol to obtain the optimum values for a specific user to be used in the <code>BlinkDetection</code> plugin.
	Dissimilarity	Compares two signals and outputs their dissimilarity coefficient
	<b>Filter</b>	Filters the incoming signal according to the programmed band. It outputs the signal power in the selected band pass
	<b>SSVEP</b>	This plugin implements the SSVEP BNCI paradigm to detect whether a user is looking at a flickering panel
Event and String Processing	<b>Eventblock</b>	Pass or suppress events depending on the plugin state
	EventCascade	Creates a series of events with selectable delays
	EventCounter	Counts incoming events and puts out the current number (increase / decrease is possible)
	<b>EventDelay</b>	Can delay incoming events for a selected period of time before passing them to the output port
	EventDispatcher	Translates incoming strings (text commands) into events
	EventFlipFlop	Creates two alternating outgoing events from one incoming event
	<b>OneEventManyActions</b>	Allows the user to control up to 10 event triggers using one or two input events
	<b>RegularExpression</b>	Processes strings with regular expressions.
	StringDispatcher	Provides twenty slots for text strings. Depending on the incoming event, a string is sent through the output port
	TextSender	Sends string data through the output port when it receives a trigger event
<b>Neural Network</b>	<b>NeuralNetworkLoader</b>	Loads a neural network configuration stored in the Encog EG file. The neural network is calculated for the input data

OSKA	<i>OSKAEventScanning</i>	Interfaces to OSKA Keyboard; key selection in OSKA can be done by incoming events
	<i>OSKAExternalScanning1D</i>	Interfaces to OSKA Keyboard; key selection in OSKA can be done by one input port (position values in two steps)
	<i>OSKAExternalScanning2D</i>	Interfaces to OSKA Keyboard; key selection in OSKA can be done by input ports (position values)
	<i>OSKAInternalScanning</i>	Interfaces to OSKA Keyboard; automatic scanning in OSKA is activated
Signal Pathways	<b>MultiSource</b>	Passes signals from four inputs to one output port
	<b>MultiSourceString</b>	Passes strings from four inputs to one output port
	PathMultiplexer	Routes one of up to 4 incoming signals to one output port, selected by events
	PathSelector	Routes the incoming signal to one of up to 4 output ports, selected by events
	<b>StringPathMultiplexer</b>	Forwards strings from the selected input port to the output port
	<b>StringPathSelector</b>	Allows routing of incoming strings to up to 4 output ports
Signal Shaping	Deadzone	Cuts off or attenuates a 1 or 2-dimensional input around a given centre
	SignalTranslation	Maps the range of an incoming signal to a selectable output range
Others	<b>Arduino</b>	Interfaces to the Arduino UNO microcontroller and provides digital input/output and analogue input functions (Arduino hardware with firmware required)
	<b>BasicTRAlgorithms</b>	Three algorithms for user hand tremor reduction
	Benchmark	Counts incoming samples or events in a given time period
	<b>FS0Command Interpreter</b>	Receives strings containing commands of the home automation system FS20 and generates events upon recognition
	<b>SpeechProcessor</b>	Starts a speech recognition engine to detect several configurable commands in a selectable installed language

Table 2: Summary of Processor Plugins

### 2.1.3 AsTeRICS Actuator Plugins

Subcategory	Name	Description	Hardware / Driver Requirements
Audio and Voice	<b>MidiPlayer</b>	Generates MIDI tones, MIDI instrument and tone scale can be selected via a GUI.	None
	<i>Synthetic Voice</i>	Generates speech output of incoming text (string).	Microsoft SAPI
	WavfilePlayer	Plays a selectable wav file on incoming event.	None
File Output	FileWriter	Writes incoming signal values to a text file.	None
	<i>Digitalout</i>	Interfaces to the DigitalOut CIM which provides open collector and relais outputs to connect external devices.	DigitalOut CIM by AsTeRICS partner IMA
Graphical User Interface	BarDisplay	Displays a bar-graph of the incoming signal in the ARE GUI.	None
	EnobioDisplay	Displays up to 4 biosignal channels of the Enobio device. <a href="http://starlab.es/products/enobio">http://starlab.es/products/enobio</a>	Enobio device, drivers installed
	EventVisualizer	Displays the connected incoming events in a text box in the ARE GUI.	None
	<b>ImageBox</b>	A GUI component, which displays image files	None
	Oscilloscope	Displays the connected incoming signal in an oscilloscope view in the ARE GUI.	None
	TextDisplay	Displays an incoming string in a text box in the ARE GUI.	None

Home Control	<b>FS20Sender</b>	Interfaces to a FS20 network for home automation.	FS20 devices
	IrTrans	Interfaces to the IrTrans Universal Infrared remote control unit, which can emit learned IR codes of generic devices like TV, music player, etc. <a href="http://www.irtrans.de/en/">http://www.irtrans.de/en/</a>	IrTrans Device (USB, WiFi or Ethernet version)
	Konnex	Interfaces to a Konnex network for home automation.	KNX actuators/network with KNX/IP router
Input Device Emulation	Keyboard	Generates simulated keyboard input (key codes) on the local computer.	None
	Mouse	Generates simulated mouse input (x/y position, clicking)- on the local computer.	None
	RemoteJoystick	Generates joystick activity on a remote computer via the HID Actuator dongle. Can also be used to emulate the 6 axis controller of the PS3 game console.	AsTeRICS HID Actuator USB dongle by IMA;
	RemoteKeyboard	Generates keyboard activity on a remote computer via the HID Actuator dongle.	AsTeRICS HID Actuator USB dongle by IMA;
	RemoteMouse	Generates relative mouse activity on a remote computer via the HID Actuator dongle.	AsTeRICS HID Actuator USB dongle by IMA;
	<b>RemoteTablet</b>	Generates absolute mouse position activity on a remote computer via the HID Actuator dongle.	AsTeRICS HID Actuator USB dongle by IMA;
Personal Platform	<u><i>PlatformDigitalOut</i></u>	Controls the open collector output of the Personal Platform to connect external devices.	AsTeRICS Personal Platform by IMA;
	<u><i>PlatformLCD</i></u>	Allows to display messages and to change selections on the Personal Platform LCD with touchscreen.	AsTeRICS Personal Platform by IMA;
Others	<b>AndoidPhoneControl</b>	Control SMS and Voice messages of an Android phone (running the AndroidServer.apk app)	Mobile Phone with Android 2.3 or higher
	<b>ApplicationLauncher</b>	Can start defined software applications on incoming events (the path to the executable file is given in the plugin properties).	None
	<b>GSModem</b>	Enables send and receive SMS actions through the GSM modem	A GSM modem with SMS option connected to the platform
	<b>ModelSwitcher</b>	Can change to a different AsTeRICS model on incoming event.	None
	PhoneControl	Interfaces to a mobile phone to send/receive SMS messages and calls.	Mobile Phone with Windows Mobile 6.5 and installed Phone Server Application
	<b>RFID Reader</b>	Interfaces the ID-Innovations RFID reader device	ID-Innovations RFID reader startup kit

Table 3: Summary of Actuator Plugins

## 2.2 Sensor Plugin Updates

In the following subsections, major updates or newly added sensor plugins will be described. Sensors are defined as components which generate data to be processed within the processing chain set up in the AsTeRICS Runtime Environment. Sensors can either generate data in the component itself (e.g. a signal generator) or they can read data from an external device. Sensors provide output ports in their bundle descriptors to make data available to other plugins.

### 2.2.1 AutostartEvent

The AutostartEvent component sends an event after model start, with a delay defined by the Delay property.

- Requirements: No special hardware or software required
- Port Description: No input or output ports available
- Event Trigger Description:  
output: The event which is sent after model start with delay defined by the Delay property.
- Properties:  
Delay [integer]: Defines the event delay [ms]

### 2.2.2 AnalogIn

The final prototype of the AsTeRICS personal platform required a re-design of the ADC CIM providing only two inputs that can measure either voltage or a resistor value. The ADC input component had to be updated to work with this new version of the ADC CIM. The component provides two output ports which correlate to the inputs IN1 and IN2 of the CIM. The component can be set up to sample the inputs periodically or upon an incoming event.

- Requirements:  
This software component requires an ADC CIM (CIM ID: 0x0402) connected to the platform (see CIM protocol specification [6]).
- Output Port Description:  
in1 to in2 [double]:  
Each output corresponds to one input of the ADC CIM. Output values are forwarded directly from the CIM thus corresponding to the description given in the CIM specification.
- Event Listener Description:  
adcSampleTrigger:  
An incoming event on this port will cause the ADC CIM to sample data on its inputs. This event will only take effect if the periodicUpdate property is set to 0.

- Properties:  
periodicUpdate [integer]:  
This property holds the time between data sampling requests to the ADC CIM. The software component will send requests to the CIM in the given intervals without waiting for the answer to the request. Thus the interval has to be set according to the sampling reaction time of the CIM  
  
activateInput1 to activateInput2 [Boolean]:  
These properties correspond the CIM's ADC inputs IN1 to IN2, if the property is true, the corresponding output port of the software component will send the sampled data..

### 2.2.3 ButtonGrid

The Button Grid component is a simple GUI on-screen keyboard. It sends events after buttons have been pressed.

- Requirements:  
No special hardware or software required
- Port Description:  
No input or output ports available
- Event Trigger Description:  
button1 to button20: these events are fired as the corresponding buttons are pressed

Properties:  
caption [string]: the text shows as component caption

horizontalOrientation [boolean]:  
If selected, the keyboard will be placed horizontally, otherwise the keyboard will be placed vertically.

buttonCaption1 to buttonCaption20 [string]:  
The text shown on the buttons 1 to 20. If the text is empty, the button will not be displayed at all.

### 2.2.4 CellBoard

The CellBoard plugin is a GUI plugin which can display a board with cells. Each cell can contain text and a picture. The CellBoard plugin provides basic scanning options and can be used as a simple on-screen keyboard or on-screen selection aid.





Figure 1: GUI of the CellBoard plugin

- **Input Port Description:**
  - row [integer]: This port can be used in directed scanning mode to set the row of the highlighted cell.
  - column [integer]: This port can be used in directed scanning mode to set the column of the highlighted cell.
  - cellNumber [integer]: This port can be used in directed scanning mode to set the number of the highlighted cell.
- **Output Port Description:**
  - selectedCell [integer]: Sends the number of the selected cell.
  - selectedCellText [string]: Sends the action text of the selected cell.
- **Event Listener Description:**
  - scanMove: Moves the scanning highlight frame in the row-column and column-row scanning modes.
  - scanSelect: Selects the highlighted cell in the row-column, column-row and directed scanning modes.
  - moveUp: Moves the scanning highlight up in the directed scanning mode.
  - moveRight: Moves the scanning highlight to the right in the directed scanning mode.
  - moveLeft: Moves the scanning highlight to the left in the directed scanning mode.
  - moveDown: Moves the scanning highlight down in the directed scanning mode.

- Event Trigger Description:
  - cellClicked: This event is triggered when one of the cells is clicked
  
  - cell1 to cell36: This event is triggered when a cell is selected.
- Properties:
  - caption [string]: The component caption.
  
  - rows [integer] (combobox selection): The number of the cell rows (1-6).
  
  - columns [integer] (combobox selection): The number of the cell columns (1-6).
  
  - textColor [integer] (combobox selection): The color of the cell text.
  
  - backgroundColor [integer] (combobox selection): The color of the cell background.
  
  - scanColor [integer] (combobox selection): The color of the scanning highlighting.
  
  - scanMode [integer] (combobox selection): This property defines the scanning mode:
    - "none": The board is not scanned. Users can select the cell by clicking on it.
    - "row-column": In this mode, rows of the cells are scanned. When the user selects the row, cells in the row are scanned. The scanning frame can be moved using the scanMove event port. The row and the cell itself can be selected using the scanSelect event port.
    - "column-row": In this mode, columns of the cells are scanned. When the user selects the column, cells in the column are scanned. The scanning frame can be moved using the scanMove event port. The column and the cell itself can be selected using the scanSelect event port.
    - "directed": In this mode only one cell is highlighted. The user can move the scan highlighting via the moveUp, moveRight, moveLeft or moveDown event ports or the row, cellNumber and column input ports. The cell can be selected using the scanSelect event port.
    - "hover selection": In this mode the user can select the cell by hovering the mouse pointer above the cell.
  
  - hoverTime [integer]: This parameter is used in the hover selection mode. It defines the time of hovering needed to select the cell (in milliseconds).
  
  - cellText1 to cellText36 [string]: The text displayed on the cell.
  
  - cellImage1 to cellImage36 [string]: The path of the image displayed on the cell.
  
  - actionText1 to actionText36 [string]: The text sent through the selectedCellText output port, when the cell is selected

## 2.2.5 DigitalIn

The final prototype of the AsTeRICS personal platform required a re-design of the GPIO CIM reducing the amount of inputs and necessitating changes in the software component. The GPIO input component provides an interface to read the inputs of the new version of the GPIO CIM and to provide events to the ARE on signal state changes.

- **Requirements:**  
This component requires the GPIO CIM (CIM Id: 0x0202) to operate (see CIM protocol specification [1]).
- **Port Description:**  
The component does not have any input or output ports.
- **Event Trigger Description:**  
**in1High to in6High:**  
Each of these event ports is linked to one physical input port, if the device connection to this input port delivers a signal which changes to high level, an event will be raised.  
  
**in1Low to in6Low:**  
Each of these event ports is linked to one input port, if the device connection to this input port delivers a signal which changes to low level, an event will be raised on the corresponding port.
- **Properties:**  
**activateEventIn1 to activateEventIn6 [Boolean]:**  
These properties declare for each port whether or not a signal transition on the actual input port should result in an event being triggered in the ARE. If a property is set true for one input, it will raise events on signal transitions, if it is set to false it will not.  
  
**periodicUpdate:**  
Period in milliseconds for update messages about state of device inputs. If the property is set to zero, the plugin will receive messages from the CIM on signal transitions, if the property is non-zero, the CIM will send status messages in the defined intervals without extra event messages on signal changes. Both modes will have the same effect on the software plugin, it will raise events on its trigger ports.

## 2.2.6 Enobio

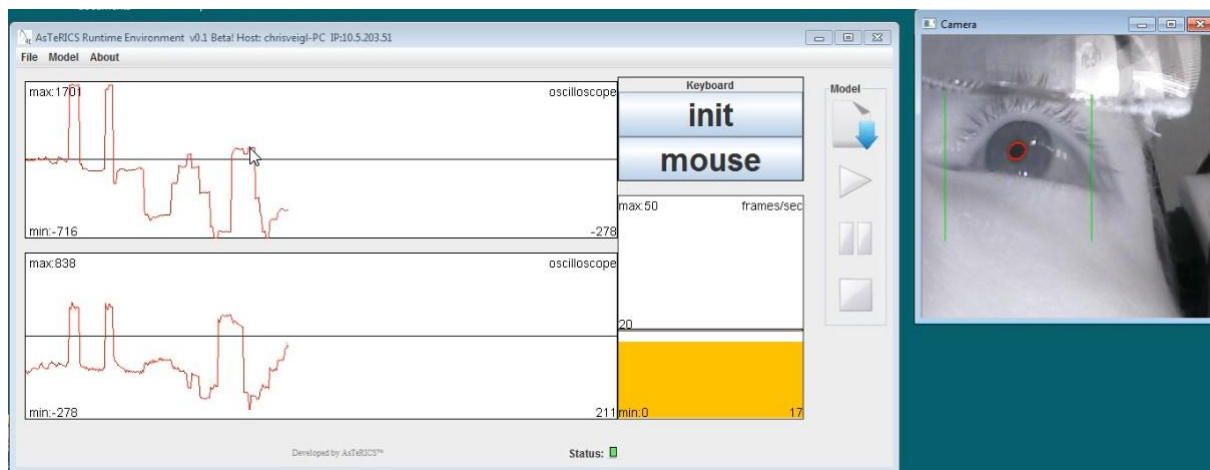
A new version of the Enobio sensor has been distributed among the AsTeRICS partners. This new version performs a better calibration of the signal apart from accepting an external trigger. The Enobio plugin has been accordingly updated so it implements the new protocol and has a new event trigger port that informs about the external trigger signal transitions. The port names as well as the property names have been changed so they are more self-explanatory.

- **Requirements:**  
This software component requires an Enobio receiver connected to the platform, the Enobio device switched on and the electrodes correctly placed on the user.

- **Output Port Description:**  
Channel1 to Channel4 [integer]: Each output corresponds to the sampled data from its corresponding Enobio channel. The integer represents the millivolts of the electro-physiological signal read by Enobio. The data might be pre-processed according to the value of the properties of the component.  
  
Status [integer]: This port provides information regarding both the calibration status of the four channels and the status of the wireless link. For every integer value that is available in the data output ports, another integer value is available in this port with the corresponding status information. The information is proprietary codified within a 16-bit integer. This includes information of calibration status of each channel and the status of the wireless link. This information would be kept away for the moment from the ARE programmers and provided upon request if necessary.
- **Event Trigger Description**  
externalSignalPosEdgeEvent: This event is fired when the external trigger signal level changes from logic '0' to logic '1'.  
  
externalSignalNegEdgeEvent: This event is fired when the external trigger signal level changes from logic '1' to logic '0'.
- **Properties:**  
IsChannel1Activated to IsChannel4Activated [Boolean]:  
If this property is set to true, the corresponding channel is calibrated, thus the raw data from this channel will be meaningful.  
  
HighPassFilterInChannel1 to HighPassFilterInChannel4 [Boolean]:  
If this property is set to true, a high pass filter is applied to the data from the corresponding channel.  
  
LineNoiseFilter [Boolean]:  
If this property is set to true, a 50 Hz band pass filter is applied to the data before it is passed to the output port. This filter is useful when the environmental electrical noise is present in the signal

## 2.2.7 Eyetracker

This component provides different computer vision tracking algorithms which can be selected via the "trackingMode" property. The available modes include "blob-tracking", "calibrated eye-tracking" and "calibrated eye-tracking with head pose estimation". The two eye-tracking modes deliver estimations of the x/y-positions where the user is looking on the computer screen which can be used for cursor control (gaze estimation).



**Figure 2: Screenshot of the Eyetracker Plugin in “calibrated eyetracking” mode**

### 2.2.7.1 Eyetracker Modes

- “Only blob tracking”:  
In this mode, the plugin just outputs the x/y coordinates of a round surface detected in the live camera images. No calibration procedure is needed. The plugin starts immediately with the coordinate output at the ports "x" and "y". A possible application for this mode is tracking of a round marker which could be placed anywhere on the body.
- “Calibrated eye tracking”:  
In this mode, the plugin expects close-up images of an eye, recorded by a head-mounted setup. The suggested hardware configuration is the AsTeRICS headmounted SVM system (see image below). The eye-pupil location is determined and mapped to an estimated position at the computer screen where the person is looking at. For this mode, the plugin needs to be calibrated as outlined in section 2.2.7.2.
- “Calibrated eye tracking with head pose estimation”  
This mode is an extension of the "calibrated eye tracking" mode. The calibration has to be performed as outlined in section 2.2.7.2. Additionally to the head-mounted camera, an external LED-frame mounted on the computer monitor is used to minimize the negative effects of head movements to the tracking accuracy. Please note that this mode needs the head-mounted SVM device plus the external Led-Frame.



**Figure 3: Virtual 3d-model of the AsTeRICS head-mounted SVM support gear**

### 2.2.7.2 The calibration procedure

- First, a rectangular region of interest (ROI) has to be selected by clicking into the live camera image while holding the CTRL/STRG key. The ROI should cover the area where the eyeball is moving when the user is looking in different directions. Then, the calibration sequence can be started by sending an event to the "calibrate" event listener port. During calibration, the cursor is moved to several locations on the screen and the user is supposed to look at these positions without moving the head. The changing of the cursor position is indicated with an acoustic signal.
- Calibration succeeds if all calibration locations can be recorded without excessively distorted values. If the calibration cannot be accomplished successfully, an acoustic signal is emitted and the Eyetracker plugin does not start to put out x/y data. In this case the calibration procedure has to be repeated.
- The coordinate-output of the eye tracking only works correctly if no head movements occur. Any head movement will compromise the correctness of the x/y estimation for the cursor position.

### 2.2.7.3 Eyetracker plugin requirements, ports and properties

- Requirements:  
A camera has to be available in the operating system, to detect round blobs or the eye-pupil position. Depending on the mode (if head pose estimation is selected), a head mounted camera which films a close-up of the eye and an external LED-mount are needed.
- Output Ports Description:  
X [integer]: The x-coordinate delivered by the tracking algorithm. The effect of this value depends on the selected mode.

Y [integer]: The y-coordinate delivered by the tracking algorithm. The effect of this value depends on the selected mode.

- **Event Listener Description:**  
calibrate: If this event is triggered, calibration procedure for the Eyetracker is started (this is not relevant for the mode "only blob tracking").

showCameraSettings:

An incoming event displays the settings window for the camera device, where parameters like image brightness or contrast can be adjusted.

- **Properties:**  
cameraSelection [string] (combobox selection):  
Using this property, the utilized camera can be chosen. Possible values range from "first camera" to "fifth camera". If only one camera is available in the system, "first camera" shall be chosen.

cameraResolution [string] (combobox selection):

This selection box provides several standard camera resolutions. Changing the resolution affects accuracy and performance (CPU load of the runtime system). Provided selections include "160x120", "320x240", "640x480", "800x600", "1024x768" and "1600x1200". If the selected resolution cannot be delivered by the image acquisition device, the next matching resolution is chosen by the plugin.

cameraDisplayUpdate [integer]:

This property allows to select the update rate for the camera display in milliseconds. If "0" milliseconds is chosen, no window for the live-video will be displayed. If "100" is chosen, the live image window will be updated 10 times a second. Please note that this property does not influence the frame rate of the camera nor the processing interval for new camera frames, only the display in the GUI is adjusted.

tracking mode [string] (combobox selection):

The selection of the plugin's mode of operation ("only blob tracking", "calibrated eye tracking", or "calibrated eye tracking with head pose estimation")

xMin [integer]: the minimum value for the x-coordinate output

xMax [integer]: the maximum value for the x-coordinate output. If "0" is selected, the plugin auto-detects the screen resolution and uses the X-Size of the computer screen.

yMin [integer]: the minimum value for the y-coordinate output

yMax [integer]: the maximum value for the y-coordinate output. If "0" is selected, the plugin auto-detects the screen resolution and uses the Y-Size of the computer screen.

calibrationStepsX [integer]:

the number of rows for generating calibration positions

calibrationStepsY [integer]:

the number of columns for generating calibration positions. For example: if 4 x-steps and 3 y-steps are chosen, the user has to look at 12 cursor positions during the calibration phase. More positions increase the gaze-tracking accuracy but result in a longer calibration phase.

averaging [integer]: the length of the averaging window for smoothing the output values.

## 2.2.8 FacetrackerLK

This component provides a face tracking computer vision algorithm which puts out estimated movement of a user's nose and chin in x and y coordinates. The underlying mechanism builds upon the OpenCV library (in particular a trained cascade of haar-like features and an optical flow algorithm). The x- and y- coordinates can be used in camera-mouse configurations or to enable selection or control tasks. The values are only provided if a face can be tracked by the algorithm and are updated with the achievable frame rate. Note that the x- and y- values represent relative movement in pixels and have to be accumulated (e.g. via the integration component) to generate e.g. absolute mouse positions.

This plugin has been considerably updated in the PT-2 works, including a complete rework of the underlying image acquisition algorithms to improve stability and compatibility, and a rework of the properties to harmonize the features with the other computer vision plugins. This cured some problems which were reported in the user evaluation (switching of camera focus to another face, problems with some web cameras).

- Requirements:  
A camera has to be available (this can be any webcam or a camera which is available as image acquisition device via the operating system).
- Output Port Description:  
noseX [integer]:  
This value specifies the relative change in the x coordinate of the user's nose with respect to the previous image frame.  
  
noseY [integer]:  
This value specifies the relative change in the y coordinate of the user's nose with respect to the previous image frame.  
  
chinX [integer]:  
This value specifies the relative change in the x coordinate of the user's chin with respect to the previous image frame.  
  
chinY [integer]:  
This value specifies the relative change in the y coordinate of the user's chin with respect to the previous image frame.



- Event Listener Description:  
init:  
An incoming event starts a re-calibration of the face. This is useful if the exact nose/chin coordinates get lost.  
  
showCameraSettings:  
An incoming event displays the settings window for the camera device, where parameters like image brightness or contrast can be adjusted
- Properties:  
cameraSelection [string] (combobox selection):  
Using this property, the utilized camera can be chosen. Possible values range from “first camera” to “fifth camera”. If only one camera is available in the system, “first camera” shall be chosen.  
  
cameraResolution [string] (combobox selection):  
This selection box provides several standard camera resolutions. Changing the resolution affects accuracy and performance (CPU load of the runtime system). Provided selections include “160x120”, “320x240”, “640x480”, “800x600”, “1024x768” and “1600x1200”. If the selected resolution cannot be delivered by the image acquisition device, the next matching resolution is chosen by the plugin.  
  
cameraDisplayUpdate [integer]:  
This property allows to select the update rate for the camera display in milliseconds. If “0” milliseconds is chosen, no window for the live-video will be displayed. If “100” is chosen, the live image window will be updated 10 times a second. Please note that this property does not influence the frame rate of the camera nor the processing interval for new camera frames, only the display in the GUI is adjusted.

### 2.2.9 FacetrackerCLM

This component has been profoundly updated in order to make full use of the fine facial landmark detection delivered by the Active Shape Models. The first novelty concerns the core approach to detect and track the face. The constrained local model facetracker (CLM) has been adopted thanks to the permission to use the library implementation by its author J. Saragih<sup>1</sup>. The CLM facetracking library has a much better performance and it's specifically engineered to closely track the face once it has been initialised. The available information from the tracker can be used to associate head movements and face gestures to mouse movements and interaction events. As a recap, the facetracker CLM software returns the best fit of a deformable face model over the current image. A deformable model in case of faces is a sparse collection of distinguishable facial landmarks. The face is a *non-rigid* structure thus the fitting of a generic model to the current instance requires determining both global deformation parameters (rotation, scale and translation) and the set of local deformations that minimize the mismatch with the current facial pose and expression.

---

<sup>1</sup> <http://web.mac.com/jsaragih/FaceTracker/FaceTracker.html>

Let's look in particular at the values exported by the plugin. Outputs can be put in two categories, those related to the head pose and those related to facial gestures.

- **Output Port Description:**
  - Roll, Pitch and Yaw [double]: Angles of the current head pose.
  - PosX, PosY [double]: Image coordinates (2D) of the head centre.
  - Scale [double]: The scale of the face.
  - EyeLeft, EyeRight [integer]: The eye state (Opened=1, Closed=0).
- **Event Listener Description:**
  - init: At an incoming event, the face tracker is re-calibrated (a new face recognition is performed to re-align the model). This is useful if the correct face alignment gets lost during operation of the tracking.
  - showCameraSettings: An incoming event displays the settings window for the camera device, where parameters like image brightness or contrast can be adjusted
  - setReferencePose: An incoming event sets the reference pose of the CLM algorithm.
- **Event Trigger Description**
  - EyebrowsRaised: This event is sent out as the eye brows are raised.
- **Properties:**
  - cameraSelection [string] (combobox selection): Using this property, the utilized camera can be chosen. Possible values range from "first camera" to "fifth camera". If only one camera is available in the system, "first camera" shall be chosen.
  - cameraResolution [string] (combobox selection): This selection box provides several standard camera resolutions. Changing the resolution affects accuracy and performance (CPU load of the runtime system). Provided selections include "160x120", "320x240", "640x480", "800x600", "1024x768" and "1600x1200". If the selected resolution cannot be delivered by the image acquisition device, the next matching resolution is chosen by the plugin.
  - cameraDisplayUpdate [integer]: This property allows to select the update rate for the camera display in milliseconds. If "0" milliseconds is chosen, no window for the live-video will be displayed. If "100" is chosen, the live image window will be updated 10 times a second. Please note that this property does not influence the frame rate of the camera nor the processing interval for new camera frames, only the display in the GUI is adjusted.
  - Camera Index [integer]: choose which camera connected to the system should be used.

modelFileName [string]:

file name of a classification data file for the blink detector. The blink detector needs to load a trained classifier (model) for recognising the state of the eyes. The model is generated by a set of routines that are introduced in the following section. Thus it is possible to learn different models in different lighting conditions and use the one that better performs in a given circumstance. A preloaded, generic model file will be used if no model filename is given – this generic model is present in the *data/facetrackerCLM* directory in the *bin* folder of the ARE runtime.

### 2.2.9.1 Opened/Closed eyes training tools.

The blink detector is included in the facetrackerCLM plugin but requires a model description file as input. This file is produced by a set of routines that takes as input a collection of samples for both classes to detect: opened and closed eyes. There are two tools accompanying the detector integrated in the facetrackerCLM plugin: the samples recorder and the training routine.

#### Recording Samples

The samples recorder is called *EyeStateRecord* and manages the training dataset. The implemented operations allow creating, updating and modifying the training set that consists of a set of labelled samples  $\{E_i, L_i\}$  where  $E_i$  is the  $i$ -th cropped eye sample and  $L_i$  is the associated label: **1** if  $E_i$  is an instance of the 'Closed' class, **0** otherwise (catches all other appearances). The output dataset is a YAML file (an XML-like markup format) that is the official technology used by openCV to serialize/deserialize matrices and data structures.

#### Command line parameters

At the moment, the executable needs a number of command line parameters, here some examples of invocations:

```
EyesStateRecord --create subject1.yml --camera 0 --width 25
```

```
EyesStateRecord --append subject1.yml --camera 0 --width 25 --resample 250
```

- **create/append** followed by the name of the file with 'yml' as extension.
  - create will create a new log.
  - append will add new samples to some existing repertoire of samples.
- **camera idx**: where  $idx$  is the integer index of the camera we'd like to use
- **width w**: it's the width in pixel of the normalised patch of each eye (stay around 25/30 pixels).
- **resample num**: when using several times *append* the dataset file will grow quite a lot. If we want to acquire new samples but keep only a subset of them, then it's possible to specify how many eye-pairs we want to save. Samples are randomly sampled at the end of the session.



**Figure 4** — In the upper right corner the current detected eyes, normalised in scale and rotation.

#### Runtime Usage:

The program needs some interaction by a secondary user to correctly associate labels to samples according to the current state of both eyes. Normally the software is in an *idle* state; this means that it doesn't save any information but just shows the current image and the cropped images of both eyes normalised and corrected in scale and rotation.

The interaction takes place through the keyboard:

- Press 'o' (the letter) to acquire 'OPENED eyes' samples. Press again 'o' or SPACEBAR to stop.
- Press 'c' to acquire 'CLOSED eyes' samples. Press 'c' again or SPACEBAR to stop acquisition.
- Press 'r' if the facetracking status is not satisfying (the visible mesh is not correctly aligned to the face).
- Finally press 'ESC' to save and quit the recording.

Figure 5 shows two examples of the acquisition window during both phases. As a suggestion consider acquiring multiple short sequences (few seconds) for both states alternating opened and closed eyes states in order to avoid the unavoidable discomfort in holding the eyes opened or closed for longer times.

Although there are no specific recommendations about the light settings of the environment, it is advisable to avoid high contrast situations in which the face receives lights mainly from one side. In this case, the facetracker is likely to be fooled very often thus affecting the quality and reliability of the cropped eyes samples. The ideal situation could be represented by a uniformly distributed illumination.

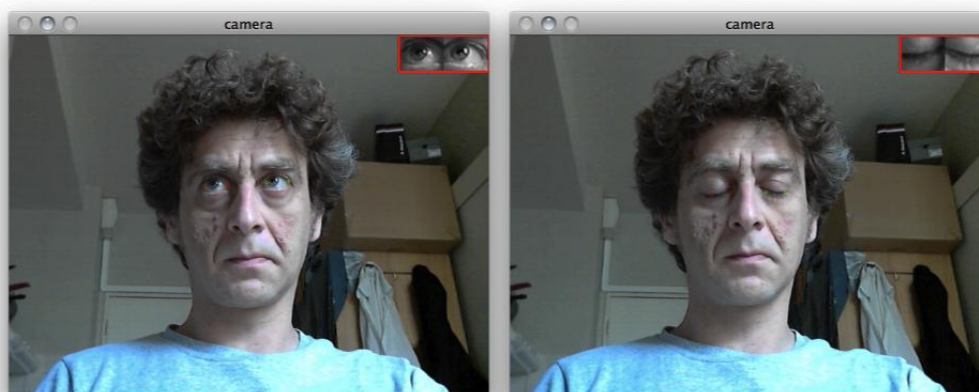


Figure 5 - During the acquisition of the training set. Open (left) and closed (right) eyes.

### Learning the model

This is a crucial routine since it computes the parameters for the PCA projection and the SVM classifier (described in D4.6a). The generic call format is the following:

```
EyesStateTrain --input subject1.yml --eigenratio 0.8 --samples 250 --model subject1_model.yml
```

- **input** [string]: the name (with extension) of the file containing the dataset we intend to use.
- **eigenratio** [double] : this is the parameter that influences the number of eigenvectors of the reduced dimensionality subspace (values between 0.8/0.9 will work fine).
- **samples** [integer]: how many pairs for each category to use in learning the SVM classifier (so, in this case there will be 4\*N total samples). When absent all available samples will be used to compute the PCA projections and train the SVM classifier. The number of samples will be in any case checked against the real number of available samples.
- **model** [string]: the name of the model (will be the input to the detector).

### 2.2.10 FS20 Receiver

The FS20Receiver receives commands from the home automation system FS20 by ELV Electronics. Depending on the received commands, events will be fired. The FS20 Receiver is a singleton, only one receiver can be used in a model. If several senders (FS20 switches, remote controls, etc.) are used, the FS20 command interpreter plugin must be used, translating the FS20 command string to events.

- Requirements:  
This component requires the FS20 PCE (see [http://www.elv.de/FS20-PC-Empfauml;nger-FS20-PCE/x.aspx/cid\\_74/detail\\_10/detail2\\_31219](http://www.elv.de/FS20-PC-Empfauml;nger-FS20-PCE/x.aspx/cid_74/detail_10/detail2_31219)) connected to a USB port.

- Output Port Description:  
fs20command [string]: The received FS20 command will be sent out by the plugin. The data has the following format: housecode\_sendaddress\_command, e.g. 11111111\_3343\_17
- Event Trigger Description  
Each received command triggers an event, being mapped to this command. The following table describes these events:

Event	Command
Off	0
Level1	1
Level2	2
Level3	3
Level4	4
Level5	5
Level6	6
Level7	7
Level8	8
Level9	9
Level10	10
Level11	11
Level12	12
Level13	13
Level14	14
Level15	15
Level16	16
OnOldLevel	17
Toggle	18
Dim Up	19
Dim Down	20
Dim Up and Down	21
Program internal timer	22
Off for timer then old brightness level	24
On for timer then off	25
On old brightness level for timer then off	26
On for timer then old brightness level	30
On for old level then previous state	31

**Table 4: List of FS20 Commands**

- Properties
  - housecode [integer]  
The housecode, the system should react on. The housecode has 8 digits, each from 1 to 4. This property influences the event triggers, but not the fs20command output port.
  - sendaddress [integer]  
The sendaddress, the system should react on. The sendaddress has 4 digits, each from 1 to 4. This property influences the event triggers, but not the fs20command output port.

### 2.2.11 Kinect

The Kinect plugin creates a connection to the Microsoft Kinect camera over the OpenNI library. It provides the skeleton data of one person in front of the Kinect camera and can be used to create user interfaces based on body gestures of the user, e.g. hand movements to control the mouse.



**Figure 6: Microsoft Kinect Sensor**

- **Requirements:**  
A Microsoft Kinect must be connected to the system. The installed drivers must be compatible to the framework e.g. the SensorKinect drivers. The OpenNI Framework including the NITE Middleware must be installed
- **Output Port Description:**  
HeadX [double]: The X position of the Head  
HeadY [double]: The Y position of the Head  
HeadZ [double]: The Z position of the Head  
leftHandX [double]: The X position of the left Hand  
leftHandY [double]: The Y position of the left Hand  
leftHandZ [double]: The Z position of the left Hand  
rightHandX [double]: The X position of the right Hand  
rightHandY [double]: The Y position of the right Hand  
rightHandZ [double]: The Z position of the right Hand  
rightFootX [double]: The X position of the right Foot  
rightFootY [double]: The Y position of the right Foot  
rightFootZ [double]: The Z position of the right Foot  
leftFootX [double]: The X position of the left Foot  
leftFootY [double]: The Y position of the left Foot  
leftFootZ [double]: The Z position of the left Foot
- **Properties:**  
visualize [boolean]:  
When set to true a window is displayed which shows the tracked skeleton. While the window is grey, no skeleton has been found by the tracker yet.

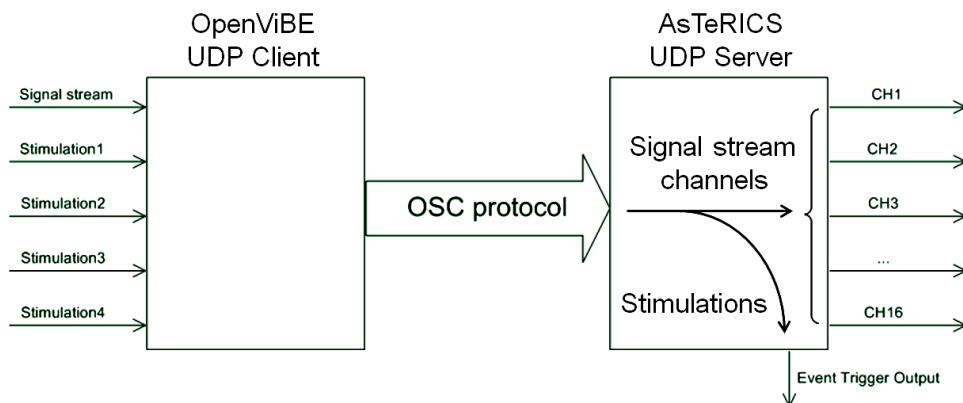
centerZeroPoint [boolean]:

Specifies whether the Point (0,0) is at the top left corner (centerZeroPoint=false) or in the middle of the frame (centerZeroPoint=true). The range of the coordinates is 0 px – 640 px for the x coordinates and 0 px – 480 px for the y coordinates if centerZeroPoint is set false and -320 px - 320 px for the x coordinate and -240 px – 240 px for the y coordinate when set to true

### 2.2.12 OpenVibe

The OpenVibe plugin allows obtaining data from a connection to the OpenVibe BCI software [5] which is currently one of the most prominent BCI frameworks. OpenVibe supports a wide range of biosignal and EEG acquisition devices and provides sophisticated signal processor training and classification algorithms. The graphical design concept is similar to the ACS, and plugins can be committed by the open source community.

To send data from OpenVibe to the ARE plugin, a dedicated OpenVibe Plugin called “AsTeRICS connection” has been created. This OpenVibe plugin sends up to 16 channels of signal data and up to 61 different stimulation events to the ARE plugin via a UDP connection, by using the Open Sound Control (OSC) protocol. The following figure shows this communication flow. For more information please refer to deliverable D4.6a (Algorithms).



**Figure 7: Data flow from OpenVibe to AsTeRICS**  
(left: AsTeRICS Connection plugin in OpenVibe, right: OpenVibe plugin for the ARE)

- Requirements
 

The OpenVibe framework has to be installed and the AsTeRICS connection plugin must be available (this means that OpenVibe has to be built from sources and the AsTeRICS connection plugin is included in the build process). Furthermore, the correct UDP port must be set in the properties of both plugins, and the ARE plugin has to be active when OpenVibe is started (because the ARE plugin acts as server and opens a listening port).



- Output Port Description

CH1 – CH16:

These are the output ports for the OpenVibe signals. Up to 16 signals can be sent.

- Event Trigger Description

OVRTK\_StimulationId\_Label\_00 – 0C:

OVRTK\_StimulationId\_Letter\_0 – Z:

OVRTK\_StimulationId\_Label\_Target:

OVRTK\_StimulationId\_Label\_NonTarget:

These are the event triggers which can be linked to the corresponding OpenVibe stimulation IDs. In total, 56 different stimulations can be processed.

- Properties:

Port: the UPD port which is opened by the AsTeRICS plugin to wait for the OpenVibe connection.

### 2.2.13 PlatformDigitalIn

Due to the fact that the final prototype is based on a different platform than the first one, the software components handling the on-board inputs and outputs obviously needed rework. The PlatformDigitalIn component provides an interface to read the three digital inputs of the AsTeRICS Personal Platform. On state changes of the connected signals (transitions to high or low level), the component generates corresponding trigger events.

- Requirements:

This component requires the Core Expansion CIM (CIM Id: 0x0602) to operate (see CIM protocol specification [1]).

- Port Description:

The component does not have any input or output ports.

- Event Trigger Description:

in1High to in3High:

Each of these event ports is linked to one input port, if the device connection to this input port delivers a signal which changes to high level, an event will be raised on the corresponding port.

in1Low to in3Low:

Each of these event ports is linked to one physical input port, if the device connection to this input port delivers a signal which changes to low level, an event will be raised on the corresponding port.

- Properties:

activateEventIn1 to activateEventIn3 [Boolean]:

These properties declare for each port whether or not a signal transition on the actual input port should result in an event being triggered in the ARE. If a property is set true for one input, it will raise events on signal transitions, if it is set to false it will not

### 2.2.14 RFID Reader

The RFIDReader component provides an interface to the ID-Innovations RFID reader modules. These modules are available for example from Sparkfun electronics. The recognized ID-Tags are transferred from the module to a COM port, where the RFIDReader component reads the ID and puts it to the output port as an ASCII string. The RFIDReader can be useful to switch to a different a model (e.g. for another user or use case) or to change parameters of a running model.



**Figure 8: ID-Innovations integrated RFID reader component (here: ID-20)**

- **Requirements**  
This software component requires an ID-Innovations RFID reader to be connected to a COM port via the Sparkfun USB breakout board or a UART/USB bridge or converter cable. Dedicated drivers have to be installed (e.g. the FTDI VCP drivers for the Sparkfun breakout board). All needed components are contained in the Sparkfun RFID Starter Kit
- **Output Port Description**  
tagID [string]: A recognized TagID is put out on this port as a sequence of 12 hexadecimal numbers in an ASCII string
- **Properties:**  
comPort [string]: The COM port where the RFID reader module is connected to (e.g. "COM5")  
  
baudRate [integer]: The baudrate for communication with the RFID reader module, should be 9600 for the ID Innovations modules

### 2.2.15 SignalShaper

The SignalShaper component can be used to generate a signal composed from basic linear functions.

- Requirements: No special hardware or software required
- Output Port Description:  
output [double]: The output port for the signal.
- Event Listener Description:  
start: Start the signal generation. If this event is received during signal generation, the signal generation is restarted.
- Properties:  
interval [integer]: The sampling rate in milliseconds.

numberOfLines [integer] (combobox selection):

The number of linear signals used to shape the output signal: 1- 5.

behaviourAfterFinish [integer] (combobox selection):

Defines the component's behavior after all signal lines have been sent:

- do nothing: stop any action.
- repeat: Send the signal lines from beginning.
- send the last value: Continue to send the last value of the signal.

beginValue1 to beginValue5 [double]: The begin value of the linear signals.

endValue1 to endValue5 [double]: The end value of the linear signals.

time1 to time5 [integer]: The duration of the linear signals in milliseconds

### 2.2.16 WiiMote

The WiiMote component interfaces to the Nintendo WiiMote controller via Bluetooth and the WiiYourself! library. It provides various sensor values including the Wiimote buttons and the tracked IR-points of the Wiimote IR front camera and the Joystick position of the Nunchuk extension.

- Requirements  
A Wiimote controller has to be available and paired with the Bluetooth radio module of the system
- Output Port Description  
pitch [integer]: The calculated pitch orientation of the Wiimote controller (in degrees)  
roll [integer]: The calculated roll orientation of the Wiimote controller (in degrees)  
point1X [integer]: The X position of the first visible IR point (0-1023)

point1Y [integer]: The Y position of the first visible IR point (0-768)  
point2X [integer]: The X position of the second visible IR point (0-1023)  
point2Y [integer]: The Y position of the second visible IR point (0-768)  
nunX [integer]: The X-position of the analog joystick on the Nunchuk extension (if connected)  
nunY [integer]: The Y-position of the analog joystick on the Nunchuk extension (if connected)  
battery [integer]: The battery level (0-100%)



**Figure 9: Nintendo WiiMote**

- Event Trigger Description
  - pressedUp: Up direction button pressed
  - releasedUp: Up direction button released
  - pressedDown: Down direction button pressed
  - releasedDown: Down direction button released
  - pressedLeft: Left direction button pressed
  - releasedLeft: Left direction button released
  - pressedRight: Right direction button pressed
  - releasedRight: Right direction button released
  - pressedA: A direction button pressed
  - releasedA: A direction button released
  - pressedB: B direction button pressed
  - releasedB: B direction button released

pressed1: 1 direction button pressed  
released1: 1 direction button released  
pressed2: 2 direction button pressed  
released2: 2 direction button released  
pressedPlus: Plus direction button pressed  
releasedPlus: Plus direction button released  
pressedMinus: Minus direction button pressed  
releasedMinus: Minus direction button released  
pressedHome: Home direction button pressed  
releasedHome: Home direction button released  
pressedNunchuckC: NunchuckC direction button pressed  
releasedNunchuckC: NunchuckC direction button released  
pressedNunchuckZ: NunchuckZ direction button pressed  
releasedNunchuckZ: NunchuckZ direction button released

- Properties  
updatePeriod [integer]: This property defines how often the WiiMote is queried for new data (in milliseconds)

## 2.3 Processor Plugin Updates

In the following subsections, major updated or newly added processor plugins will be described. Processor plugins are components which modify data provided by other plugins. Thus, processors provide input ports as well as output ports and are placed in the middle of the ARE processing chain. Types of processors are either data modifiers (such as amplify- or average- plugins) or flow control elements (such as conditional path selectors).

### 2.3.1 Arduino

The Arduino component provides an interface to the Arduino Uno microcontroller and makes available analog inputs and digital inputs and outputs. This allows connection of non-standard sensors and actuators to the AsTeRICS platform and opens endless possibilities for DIY-enthusiasts to extend the system.

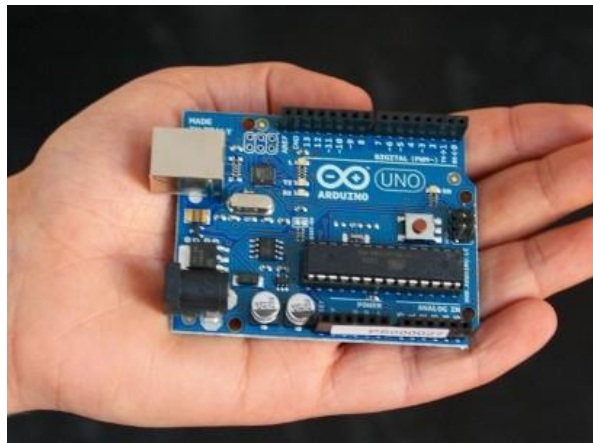


Figure 10: Arduino Uno microcontroller

- Requirements  
An Arduino UNO microcontroller board has to be connected to a USB port. The Arduino CIM firmware must have been downloaded into the Arduino to communicate via the CIM protocol with the Arduino component
- Input Port Description  
pwm3 [integer]: currently not supported (will provide a PWM output channel on the Arduino)  
  
pwm5 [integer]: currently not supported (will provide a PWM output channel on the Arduino)  
  
pwm6 [integer]: currently not supported (will provide a PWM output channel on the Arduino)

- Output Port Description  
A0 - A5 [integer]: The readings of the 6 analogue channels of the Arduino (0-1023)
- Event Listener Description  
setPin2 - setPin13: An incoming event will set the corresponding digital output pin on the Arduino to high level (5V)  
  
clearPin2 - clearPin13: An incoming event will set the corresponding digital output pin on the Arduino to low level (0V)
- Event Trigger Description  
pin2ChangedToLow - pin13ChangedToLow: This event is triggered if the corresponding input pin on the Arduino switches from high to low level (connected to 0V)  
  
pin2ChangedToHigh - pin13ChangedToHigh: This event is triggered if the corresponding input pin on the Arduino switches from low to high level (connected to 5V)
- Properties  
periodicADCUpdate [integer]: This property defines how often the ADC values are measured and put out on the ports A0-A5 (0=disable ADC)  
  
pin2Mode - pin13Mode [integer] (combobox selection):  
This property defines the mode of the Arduino Pins. The value can be selected via a ComboBox. Possible selections are: "not used", "Input without pullup resistor", "Input with pullup resistor", "output default low", "output default high"

### 2.3.2 BasicTRAlgorithms

This plugin contains three algorithms for user hand tremor reduction: Arithmetic Mean, Outlier Reduction and Exponential Smoothing. The Arithmetic Mean algorithm calculates the cursor position as an average of the past  $n$  cursor positions.  $N$  is defined by the `bufferSize` parameter. The Outlier Reduction algorithm keeps the maximum distance between two followed cursor positions. The maximum distance is defined by the `maxDistance` parameter. If the distance between two cursor positions is greater than `maxDistance`, it is reduced to the value of `maxDistance`. The Exponential Smoothing algorithm implements the Exponential Smoothing technique [11]. The `factor` parameter defines the Exponential Smoothing and the `degree` parameters define the degree of the equation.

- Input Port Description:  
inputX [integer]: Input mouse X position.  
  
inputY [integer]: Input mouse Y position.  
  
bufferSize [integer]: The new buffer size value for the Arithmetic Mean algorithm.  
  
maxDistance [double]: The new maximum distance value for the Outlier Reduction

algorithm.

factor [double]: The new factor value for Exponential Smoothing algorithm.

- Output Port Description:  
outputX [integer]: Output mouse X position.

outputY [integer]: Output mouse Y position

- Properties:

algorithm [integer] (combobox selection):  
Defines the algorithm used for the tremor reduction.

eventsType [integer] (combobox selection):  
Defines if the mouse coordinates are absolute or relative values.

bufferSize [integer]: The buffer size value for the Arithmetic Mean algorithm.

maxDistance [double]: The maximum distance value for the Outlier Reduction algorithm.

factor [double]: The factor value for Exponential Smoothing algorithm.

degree [integer] (combobox selection):  
The degree of the equation for the Exponential Smoothing algorithm (1-4).

### 2.3.3 Blink Detection

The user evaluations showed that the performance of the blink detection plugin changed from user to user. The plugin developed for the prototype 1 did not accept any parameter so it used standard values for all the users. In order to better adapt the plugin to different users the plugin has been updated to accept some parameters that characterize the user's blink so a better detection can be performed. The optimum values for those parameters can be obtained through the Blink Detection Training plugin that is described below.

- Requirements:  
The input signal shall correspond to an electro-oculogram signal, i.e., an output port of the Enobio component when the corresponding electrode is placed on the user's forehead.
- Input Port Description:  
Input [double]: Input values that correspond to a sampled electro-oculogram signal.
- Event Trigger Description:  
BlinkDetected: This event port fires an event if a blink is detected in the input sequence of values.



- DoubleblinkDetected: This event port fires an event if a double blink is detected in the input sequence of values.
- Properties:
  - SampleRate [integer]: The input signal sample rate.

Threshold [integer]: The value used to tune the threshold that determines whether the signal is behaving as an eye blink.

BlinkLength [integer]: Maximum length of the blink in milliseconds.

DoubleBlinkSeparation [integer]: Maximum separation between blinks to be considered as double blink

### 2.3.4 BlinkDetectionTraining

As it has been described above, the BlinkDetection plugin now accepts some parameters to better characterize the blinks of different users. The BlinkDetectionTraining plugin can be used to obtain the optimum values for a specific user. In order to obtain those values the plugin performs a training protocol where the user is asked to blink or double blink their eyes at some concrete times. The plugin analyses the incoming signal to characterize the user's blink and provides the values that might be used in the Blink Detection plugin.

- Requirements:
  - The input signal shall correspond to an electro-oculogram signal, i.e., an output port of the Enobio component when the corresponding electrode is placed on the user's forehead. In addition the user shall follow the instructions that the plugin provides like blink or double blink after it is requested.
- Input port Description:
  - Input [double]:  
Input values that correspond to a sampled electro-oculogram signal.
- Output Port Description:
  - Output [string]:  
This port outputs the messages that shall be displayed to the user in order to let him follow the training protocol. At the end of the protocol the plugin shows through that output port the optimum values to be used with the Blink Detector plugin for the current user.
- Event Trigger Description
  - There are no events associated with this plugin.
- Properties:
  - This component does not have any property values.

### 2.3.5 DoubleToString

This component converts the double values at the input port to string values at the output.

- Input Port Description:  
input [double]: Input port for the double values to be converted.
- Output Port Description:  
output [string]: Output port for the converted string values.
- Properties:  
This component does not have property values.

### 2.3.6 EventBlock

This plugin, depending on its state, can pass or block events from the input port.

- Event Listener Description:  
input: Input port for the events.  
  
pass: Pass the events.  
  
block: Block the events.  
  
change: Change the state of component: pass/block to the opposite.
- Event Trigger Description:  
output: Output port for events.
- Properties:  
block [boolean]: If is set to true, the component will block the events after start.  
  
blockAfterEvent [boolean]:  
If is set to true, the component will block the events after passing one event.

### 2.3.7 EventDelay

This plugin passes received events after a defined delay.

- Event Listener Description:  
input: Input port for the events.
- Event Trigger Description:  
output: Output port for events.
- Properties:  
delay [integer]: Delay between input and output event (in milliseconds).

### 2.3.8 Filter

This plugin implements a generic FIR filter which order can be programmed through its properties. The filter plugin can work either as a low pass filter, a band pass filter or a high pass filter. The -3dB cut frequencies for the three types of filters are provided via properties. Apart from the output port with the filtered signal, the plugin provides an output with the computed signal power in the pass band of the filter. This output can be configured to provide an output at a different rate than the original signal. The size of the buffer for computing the signal power can be configured too. This functionality overlaps with the one provided by the ComputeBandPower plugin developed for the prototype 1.

- Requirements  
The values in the input port shall correspond to a time series.
- Input Port Description  
Input [double]:  
Input port for the values of the time series that is filtered by the plugin.
- Output Port Description:  
Output [double]: The filtered signal

SignalPower [double]:

The power of the filtered signal taking into account the last N output samples, where N is defined by the property SignalPowerBufferSize. A one-Ohm termination resistor is considered for the computation of the signal power. This port can deliver the output value at a lower rate than the Output port according to the SignalPowerUpdateRate property.

- Event Trigger Description  
There are no events associated with the filter plugin.
- Properties:  
Order [integer]: The filter order.

CutoffFrequency1 [double]:

-3dB cutoff frequency for low and high pass filter types. In case of band pass filters it is the low cutoff frequency of the band.

CutoffFrequency2 [double]:

In case of band pass filters it is the high cutoff frequency of the band.

SamplingRate [integer]:

The sampling rate of the incoming signal in samples per second.

Type [integer]: The filter type: "Low pass", "Band pass" or "High pass".

SignalPowerUpdateRate [integer]:

Update ratio for the SignalPower output port. The SignalPower output port provides a new value every time the Output port has provided N values, where N is the value of this property.

### 2.3.9 FS20 Command Interpreter

The FS20CommandInterpreter receives strings containing commands of the home automation system FS20 from ELV Electronics. Depending on the received commands, events will be fired.

- **Input Port Description**  
command [string]: The command string containing the FS20 Command. The data must have the following format: housecode\_sendaddress\_command, e.g. 11111111\_3343\_17
- **Event Trigger Description**  
Each received command triggers an event, being mapped to this command. The following table describes these events:

Event	Command
Off	0
Level1	1
Level2	2
Level3	3
Level4	4
Level5	5
Level6	6
Level7	7
Level8	8
Level9	9
Level10	10
Level11	11
Level12	12
Level13	13
Level14	14
Level15	15
Level16	16
OnOldLevel	17
Toggle	18
Dim Up	19
Dim Down	20
Dim Up and Down	21
Program internal timer	22
Off for timer then old brightness level	24
On for timer then off	25
On old brightness level for timer then off	26
On for timer then old brightness level	30
On for old level then previous state	31

**Table 5: List of Commands for the FS20CommandInterpreter**

- **Properties**  
housecode [integer]  
The housecode, the plugin should react on. (8 digits, each from 1 to 4).  
  
sendaddress [integer]  
The sendaddress, the plugin should react on. (4 digits, each from 1 to 4).

### 2.3.10 IntToString

This component converts the incoming integer values to the string values at the output port.

- Input Port Description:  
input [integer]: Input port for the integer values to be converted.  
  
Output Port Description:  
output [string]: Output port for the converted string values.
- Properties:  
hexadecimalOutput [boolean]:  
If this property is set, the integer is converted into a hexadecimal string.

### 2.3.11 NeuralNetworkLoader

This plugin uses the Encog framework (version: 3.0.1) [12]. The plugin can load a neural network configuration stored in the Encog EG file. The neural network's output is calculated for the input data.

- Input Port Description:  
input1 to input32 [double]: The neural network inputs.
- Output Port Description:  
output1 to output32 [double]: The Neural network outputs.
- Properties:  
filePath [string]: The EG file path.

### 2.3.12 OneEventManyActions

This component allows the user to control up to 10 actions using just one or two input events. In the first step, the event actions are scanned so that the user can choose the action, in the next step the selected action event is triggered. The detailed functionality depends upon the selected mode.

- Port Description:  
No input or output ports available.
- Event Listener Description:  
input: The Control event.  
  
inputOff: Control event used in mode 2 for selecting the action by scanning.
- Event Trigger Description:  
action1 to action10: The event triggers for the actions selected by user.

actionShown1 to actionShown10:

These events are used in the scanning process to highlight the currently selected action via a GUI element (e.g. the CellBoard).

- Properties:

actionsNumber [integer] (combobox selection): Number of used actions.

mode [integer] (combobox selection): The component's working mode:

- mode 1: The input event starts the scanning, the inputOff event selects the action.
- mode 2: The first input event starts the scanning, the second input event selects the action.
- mode 3: The first input event starts the scanning and highlights the first action, the next input events highlight the next actions. If there is no input event for the selected delay time, the currently highlighted action is selected.

delay [integer]:

Delay used for the scanning process (in milliseconds).

### 2.3.13 OskalInternalScanning

The OSKA plugins are used to interface the ARE to the On Screen Keyboard Application (OSKA). The outputs of OSKA (respectively the contents of the selected cells) are forwarded to the output ports of the OSKA plugins. In order to make the OSKA component of the first AsTeRICS prototype more comprehensible for users, the component was split into three components with only the necessary inputs and outputs for the desired mode of operation.

The OskalInternalScanning component starts OSKA in automatic (internal) scanning mode, which can be operated via one single switch or two switches as desired. The component exposes the two-button-input that OSKA can work with.

- Requirements: this component requires Sensory Software's OSKA keyboard.

- Output Port Description:

action [string]:

This output sends the action string which is attached to a specific key on the keyboard to connected components.

keycodes [string]:

This output sends the keycode string of a specific key on the keyboard to connected components.

- Event Listener Description:

increaseScanSpeed:

Incoming events will increase the internal scanning speed of OSKA by one.

decreaseScanSpeed:

Incoming events will decrease the internal scanning speed of OSKA by one.

pressSwitch1:

Incoming events start the automatic scanning or switch to the next selection (to speed up the scanning)

pressSwitch2:

Incoming events switch from column- to row scanning (or in the next step select the cell). If the scanning is stopped, it will be started.

- Properties:

port [integer]:

This property defines the TCP port that the component listens on for connections of the OSKA.

title [string]:

This property defines the caption to be displayed in OSKA's title bar.

oskaPath [string]:

The absolute path to the OSKA player as well as the program name is needed in order for the component to start OSKA by itself.

keyboardPath [string]:

If this property does not hold an empty string, the component will ask OSKA to load the keyboard referred to by this property.

scanSpeed [integer]:

This property relates to the internal row column scanning method of OSKA and sets the speed of scanning, the value range is between 1 and 10.

highlightStyle [integer] (combobox selection):

This property selects the style of highlighting used in OSKA via a combobox. Possible selection are: "KeyBorders", "InvertKeys" and "BlockBehindKeys".

settingsFile [string]:

if this property is not empty OSKA will be started with this settings file as a command line parameter.

windowDecorated [boolean]: if true OSKA will display a decorated window frame, otherwise only a lightweight frame.

### 2.3.14 OskaExternalScanning1D

This component communicates with the On Screen Keyboard Application (OSKA) and forwards its inputs to the AsTeRICS system. Scanning is done in a two-step fashion and selection of columns and cells is performed upon incoming events and input values. The event input ports allow switching from row- to column scanning, moving the cell highlighter and selecting a cell.

- Requirements: this component requires Sensory Software's OSKA keyboard.
- Output Port Description:
  - action [string]:  
This output sends the action string which is attached to a specific key on the keyboard to connected components.
  - keycodes [string]:  
This output sends the key codes which are attached to a key via the @KDB command.
- Event Listener Description:
  - switch: Incoming events either switch from column- to row-scanning or (in the second step) select the currently highlighted cell
  - highlightNext: Incoming events highlight the next column (or cell in a row).
  - highlightPrev: Incoming events highlight the previous column (or cell in a row).
- Properties:
  - port [integer]:  
This property defines the TCP port that the component listens on for connections of the OSKA.
  - title [string]: This property defines the caption to be displayed in OSKA's title bar.
  - oskaPath [string]: The absolute path to the OSKA player as well as the program name is needed in order for the component to start OSKA by itself.
  - keyboardPath [string]: If this property does not hold an empty string, the component will ask OSKA to load the keyboard referred to by this property.
  - highlightStyle [integer] (combobox selection):  
This property selects the style of highlighting used in OSKA via a combobox. Possible selection are: "KeyBorders", "InvertKeys" and "BlockBehindKeys".
  - settingsFile [string]: if this property is not empty OSKA will be started with this settings file as a command line parameter.
  - windowDecorated [boolean]: if true OSKA will display a decorated window frame, otherwise only a lightweight frame.
  - eventScanningEnabled [boolean]: if true, incoming events can move the highlighter.
  - valueScanningEnabled [boolean]: if true, incoming values can move the highlighter.
  - inputRange [boolean] (combobox): the range of the incoming input port values: either float values between 0 and 1, or integer values or integer values from 0 to the maximum element (number of cells in a respective row or column of the grid).



### 2.3.15 OskaExternalScanning2D

This component communicates with the On Screen Keyboard Application (OSKA) and forwards its inputs to the AsTeRICS system. Scanning is operated using two float inputs which take inputs in a range defined by the according property (between 0.0 and 1.0. or integer values). The two inputs control the position of the highlighted cell in a grid with values being mapped to a corresponding position in the grid. An event input allows pressing selected cells in the grid.

- Requirements: this component requires Sensory Software's OSKA keyboard.
- Input Port Description:
  - posColumn [double]:  
This port takes values between 0.0 and 1.0 while clipping other values to 0.0 or 1.0. The input moves the highlighted key to the corresponding column. The step size between each column is dependent on the number of columns.
  - posRow [double]:  
This port takes values between 0.0 and 1.0 while clipping other values to 0.0 or 1.0. The input moves the highlighted key to the corresponding row. The step size between each row is dependent on the number of rows.
- Output Port Description:
  - action [string]:  
This output sends the action string which is attached to a specific key on the keyboard to connected components.
  - keycodes [string]:  
This output sends the key codes which are attached to a key via the @KDB command.
- Event Listener Description:
  - press: Incoming events will trigger a press action on the selected key in OSKA.
  - highlightNextX: Incoming events will highlight next cell in the selected row.
  - highlightPrevX: Incoming events will highlight the previous cell in the selected row.
  - highlightNextY: Incoming events will highlight the next cell in the selected column.
  - highlightPrevY: Incoming events will highlight the previous cell in the selected column.
- Properties:
  - port [integer]:  
This property defines the TCP port that the component listens on for connections of the OSKA.

title [string]:

This property defines the caption to be displayed in OSKA's title bar.

oskaPath [string]:

The absolute path to the OSKA player as well as the program name is needed in order for the component to start OSKA by itself.

keyboardPath [string]:

If this property does not hold an empty string, the component will ask OSKA to load the keyboard referred to by this property.

highlightStyle [integer]:

This property selects the style of highlighting used in OSKA via a combobox. Possible selection are: "KeyBorders", "InvertKeys" and "BlockBehindKeys".

settingsFile [string]:

if this property is not empty OSKA will be started with this settings file as a command line parameter.

windowDecorated [boolean]: if true OSKA will display a decorated window frame, otherwise only a lightweight frame.

eventScanningEnabled [boolean]: if true, incoming events can move the highlighter.

valueScanningEnabled [boolean]: if true, incoming values can move the highlighter.

inputRange [boolean] (combobox): the range of the incoming input port values: either float values between 0 and 1, or integer values or integer values from 0 to the maximum element (number of cells in a respective row or column of the grid).

### 2.3.16 MultiSource

This plugin passes signals from up to four input ports to one output port.

- Input Port Description:  
input1 to input4 [double]: Input ports 1 to 4
- Output Port Description:  
output [double]: The output port where all input signals are routed.
- Properties:  
This component does not have property values.

### 2.3.17 MultiSourceString

This plugin passes string inputs from up to four input ports to one output port.

- Input Port Description:  
input1 to input4 [string]: Input port 1 to 4
- Output Port Description:  
output [string]: Output port.
- Properties:  
This component does not have property values.

### 2.3.18 Quantizer

This plugin performs a quantization of the input signal. The value of the output signal is the input value rounded to the nearest multiple of the quantizationStep property value.

- Input Port Description:  
input [double]: Input port for the values to be quantized.
- Output Port Description:  
output [double]: Output port for the quantized values.
- Properties:  
quantizationStep [double]: The quantization step.

### 2.3.19 RegularExpression

This component processes strings with regular expressions. It can work in two modes: match strings with the pattern or replace string parts which match the pattern with another string.

- Input Port Description:  
input [string]: Input port for strings.
- Output Port Description:  
output [string]: The port for strings which match the pattern or which were changed.
- Event Trigger Description:  
match: This event is sent if the string matches the pattern.  
  
notMatch: This event is sent if the string does not match the pattern.  
  
replace: This event is sent if parts of the string were replaced by the replaceString.  
  
notReplace: This event is sent if no replacement occurred.

- Properties:

pattern [string]: Regular expression pattern.

replace [boolean]:

If the property is set to true, the component will search parts of the string which match the pattern and replace these parts with the replaceString, otherwise the component will match the whole string with the pattern.

replaceString [string]: The string which replaces expressions matching the pattern.

### 2.3.20 RelativeMoveSampler

This plugin was developed for models where a constant rate of value updates is needed. The plugin sums incoming relative position packages for three coordinates. The sums are sent to the output ports according to sampling rate. When the sums are sent, the accumulation values are reset to zero.

- Input Port Description:  
inputX [integer]: Input X position.  
  
inputY [integer]: Input Y position.  
  
inputZ [integer]: Input Z position.
- Output Port Description:  
outputX [integer]: Output X position.  
  
outputY [integer]: Output Y position.  
  
outputZ [integer]: Output Z position.
- Properties:  
samplingRate [integer]: Defines the rate of the sampling (in Hz).

### 2.3.21 Sampler

This plugin generates a constant rate of sampling for the input port signal.

- Input Port Description:  
input [double]: Input port for the signal.
- Output Port Description:  
output [double]: Output port for signal with constant rate of the sampling.

- Properties:
  - samplingRate [double]:  
The rate of sampling, which defines the data generation frequency at the output port
  - responseTime [integer]:  
Response time in milliseconds: If the time since the arrival of the last input data exceeds the response time, the plugin stops sending the output data. If the responseTime is set to 0, it is not used.
  - sendNullSamples [boolean]:  
If this property is set to true and there is no input data or the response time is exceeded, the plugin sends samples equal to zero.

### 2.3.22 StringPathMultiplexer

The String Path Multiplexer component forwards strings from the selected input port to the output port.

- Input Port Description:  
input1 to input4 [string]: The input ports for strings to be multiplexed.
- Output Port Description:  
output [string]: The string output, which sends the data of the selected input port.
- Event Listener Description:  
passPort1 to passPort4:  
Incoming events select one of the four input ports to be forwarded to the output port.  
  
passNextPort:  
Selects the next input port. If the currently used port is the maximum port (defined by the number property), input port 1 will be selected.  
  
passPreviousPort:  
Selects the previous input port. If the currently used port is port 1, the maximum port (defined by the number property) will be selected.
- Properties:  
number [integer] (combobox selection): The maximum port number (can be 1 to 4).

### 2.3.23 StringPathSelector

The StringPathSelector component allows routing of incoming strings between up to 4 output ports. The desired output port can be directly selected by a dedicated event listener port, or the strings can be switched to the next or previous output port.

- Input Port Description:  
in [string]: The incoming string port to be routed.
- Output Port Description:  
out1 to out4 [string]: Four output ports where the incoming strings can be routed to.

- Event Listener Description:

select1 to select4:

An incoming event at these ports directly activates the associated output path (e.g. as an event comes in at select 3, the input string port will be routed to out3).

selectNext:

The next output port is selected for string output. The maximum number of active ports is set via the activePorts property. If the current number is already the maximum, the selectNext event will wrap around the active port number and port 1 will be selected.

selectPrevious:

The previous output port is selected for string output. If the current port is out1, the maximum port number (given by the activePorts property) will be selected.

- Properties:  
activePorts [integer] (combobox selection):  
The maximum port number in use (can be 1 to 4).

### 2.3.24 SSVEP

This plugin will implement the Steady State Visually Evoked Potential paradigm so it will decide whether the incoming signal contains the evoked potentials provoked when the user looks at a flickering panel. The plugin shall provide which flickering panel the user is looking at for further actions in the model where this plugin is used. In order to work this plugin needs some parameters that are obtained through a training protocol. For that purpose there will be an auxiliary plugin that will implement that protocol and will output the values that should be configured in the SSVEP plugin. Neither the SSVEP plugin nor the training one has been developed yet at this stage of the project. The full description of those plugins with their corresponding inputs and output ports as well as their properties will be provided later in WP6.

### 2.3.25 SpeechProcessor

The SpeechProcessor component provides methods to use a speech recognition engine and a speech synthesizer, provided via the Microsoft Speech Platform Server 10.2 (see <http://www.microsoft.com/en-us/download/details.aspx?id=10208>). The Microsoft Speech Platform SDK provides a comprehensive set of development tools for managing voice-enabled applications and enabling recognition of spoken words and for generation of

synthesized speech (text-to-speech, TTS). The Engines and language packs can be downloaded for free (see above link).

The component works with an external program implementing the Speech Platform interface and communicating with the plugin via TCP, using the remote connection management service of the AsTeRICS Runtime Environment. This program is located in the ARE subfolder “./tools/SpeechProcessor.exe” and configures the speech engines according to the plugin properties. Recognized words in a selectable language (culture) are then transferred back to the ARE plugin and can trigger events there. Moreover, the component can receive strings which are spoken via the selected speech synthesizer.

- **Requirements:**  
This component requires Microsoft’s Speech Platform Server and the desired language packs to be installed on the platform running the plugin.
- **Input Port Description:**  
**speak [string]:**  
This port receives strings containing sentences or words that should be spoken via a speech synthesizer which fits the selected language (text-to-speech).
- **Output Port Description:**  
**command [string]:**  
This output sends strings which have been recognized by the speech recognition engine (matching one of the commands defined by property values).
- **Event Triggerer Description:**  
**activated:** triggered when the recognition is activated via the defined activation command  
  
**deActivated:** triggered when the recognition is deactivated via the defined command  
  
**recognizedCommand1 – recognizedCommand10:**  
If an incoming word matches a command defined in the plugin properties, the corresponding event will fire.
- **Properties:**  
**language [integer] (combobox selection):**  
A combobox which defines which speech pack is to be used (currently: English, German, Spanish, Polish) – these language packs need to be installed !  
  
**recognitionConfidence [double]:**  
This value defines the needed confidence for a valid recognition. The value can range from 0 to 1. A higher value results in a more precise pronunciation needed to detect a command, a lower level could lead to more false-positive recognition results.  
  
**mode [integer] (combobox selection):**  
This property selects one out of three operating modes for the speech recognition:  
  
“always active”: the command recognition is always running

“voice-triggered activation and deactivation”: the command recognition is started by the recognition of the activation command and stopped by the deactivation command (these commands are defined in the according properties)

“automatic deactivation after command recognition”: after a recognized command, the speech recognition will be bypassed until another activation command has been recognized

activationCommand [string]: a speech command to start the recognition of the other commands

deactivationCommand [string]: a speech command to stop the recognition of the other commands

command1 to command10:

The command strings which are checked by the speech recognition engines (these build the recognition grammar)

### 2.3.26 StringToDouble

This component converts the incoming string values into double values at the output port.

- Input Port Description:  
input [string]: Input port for the string values to be converted.
- Output Port Description:  
output [double]: Output port for the converted double values.
- Properties:  
This component does not have property values.

### 2.3.27 StringToInt

This component converts the incoming string values into integer values at the output port.

- Input Port Description:  
input [string]: Input port for the string values to be converted.
- Output Port Description:  
output [integer]: Output port for the converted integer values.
- Properties:  
hexadecimalInput [boolean]:  
If this property is set, the input string data are regarded as a hexadecimal data string.



## 2.4 Actuator Plugin Updates

In the following subsections, major updated or newly added actuator plugins will be described. Actuators are components which take input data to control external processes (like hardware peripherals) or make data visible (in the Graphical User Interface). Actuators only have input ports since they are data sinks in the ARE.

### 2.4.1 AndroidPhoneControl

This component controls a mobile phone with Android operating system through a TCP/IP connection. Currently this component is able to perform such action as: call a remote phone, drop a phone call, send and receive SMS message.

- Requirements:  
Android phone running the AsTeRICSPhoneServer application.
- Input Port Description:  
phoneID [string]: The phone number used for outgoing SMS and make phone calls.  
  
SMSContent [string]: The SMS content which will be used for the send SMS action.  
  
command [string]:  
String command that can be sent to this component from other plugins to trigger phone actions. Currently supported commands are:
  - @PHONE: SMS:Phone\_ID, "Message\_content"
  - @PHONE: CALL: Phone\_ID
  - @PHONE: ACCEPT
  - @PHONE: DROP
- Output Port Description:  
remotePhoneID [string]: This is the phone number of the caller or SMS sender.  
  
receivedSMS [string]: This is the content of the incoming SMS.  
  
errorNumber [integer]: The number of the error.
- Event Listener Description:  
  
sendSMS: Sends an SMS message.  
  
makePhoneCall: Makes a phone call.  
  
acceptPhoneCall: Accepts the incoming phone call.  
  
dropPhoneCall: Drops the phone call.

- Event Trigger Description:

idleState: Phone is in the idle state.

ringState: Phone is in the ring state.

connectedState: Phone is connected with the remote phone.

newSMS: There is a new SMS.

error: An error occurred.

- Properties:

connectionType [integer] (combobox selection):

Defines connection type for the plugin: client or server.

IP [string]: IP of the remote server used in the client mode.

port [integer]: TCP/IP port of the service.

defaultPhoneID [string]: Default phone number for outgoing SMS and phone calls.

## 2.4.2 ApplicationLauncher

The ApplicationLauncher component can be used to run an external executable application. The application name is given to the plugin via an input port. A default application can be started via an incoming event. Together with the Keyboard- or RemoteKeyboard components, the ApplicationLauncher plugin can perform complex automation tasks, for example open Skype, choose a contact and make a call.

- Input port Description:

filename [string]: The filename of the application to be started (including path).

- Event Listener Description:

launchNow: An incoming event on this port will start the application.

closeNow: An incoming event on this port will close the current application.

- Properties:

defaultApplication [string]: Full path and filename of the default application

autoLaunch [boolean]: Defines if the application automatically launched at startup

autoClose [boolean]: Defines if the current application is closed when the model is stopped

onlyByEvent [boolean]: If this property is set to true, incoming application files names will not be started immediately (only the launchNow event will start the application)



- Requirements: The PCS sender module must be attached to the system
- Input Port Description:  
houseCode [int]: The houseCode to which the command should be sent. Overrides the houseCode set in the properties

address [int]: The address of the target device. Overrides the houseCode set in the properties.

action [string]: Action input to send commands from other components which output a string, for example OSKA. The string format is as follows:

houseCode\_address\_command; e.g. 11111111\_1234\_18 to send the toggle command to the device with housecode 11111111 and address 1234. For the indices of the commands see the table below.

- Event Listener Description:  
See table below for a list of all commands that can be triggered:

Event	Command
Off	0
Level1	1
Level2	2
Level3	3
Level4	4
Level5	5
Level6	6
Level7	7
Level8	8
Level9	9
Level10	10
Level11	11
Level12	12
Level13	13
Level14	14
Level15	15
Level16	16
OnOldLevel	17
Toggle	18
Dim Up	19
Dim Down	20
Dim Up and Down	21
Program internal timer	22
Off for timer then old brightness level	24
On for timer then off	25
On old brightness level for timer then off	26
On for timer then old brightness level	30
On for old level then previous state	31

**Table 6: List of Commands for the FS20Sender**

- Properties:  
houseCode [integer]:  
The default housecode for the component, can be overridden by the input port  
  
address [integer]:  
The default address for the component, can be overridden by the input port

## 2.4.5 GSMModem

This component can perform send and receive SMS action through a GSM modem.

- Requirements:  
A GSM modem with SMS option connected to the platform. The modem has to enable its port in the system.
- Input Port Description:  
phoneID [string]: Phone number which will be used for the send SMS action.  
  
SMSContent [string]: SMS content which will be used for the send SMS action.
- Output Port Description:  
remotePhoneID [string]: This is the phone number of the SMS sender.  
  
receivedSMS [string]: This is the content of the incoming SMS.  
  
errorNumber [integer]: The number of the error.
- Event Listener Description:  
sendSMS: Sends the SMS message.
- Event Trigger Description:  
newSMS: There is a new message.  
  
error: An error occurred.
- Properties:  
serialPort [string]:  
The modem COM port. If this parameter is empty, the component uses the port of the first modem found.  
  
pin [string]:  
The PIN code for the SIM card. If the PIN is not needed this property should be empty.  
  
smsCenterID [string]:  
SMS Center ID. If the Center ID is not needed this property should be empty.  
  
defaultPhoneID [string]:  
This is the default phone number, which will be used for the send SMS actions.

## 2.4.6 ImageBox

The Image Box is a GUI component which displays images loaded from image files.

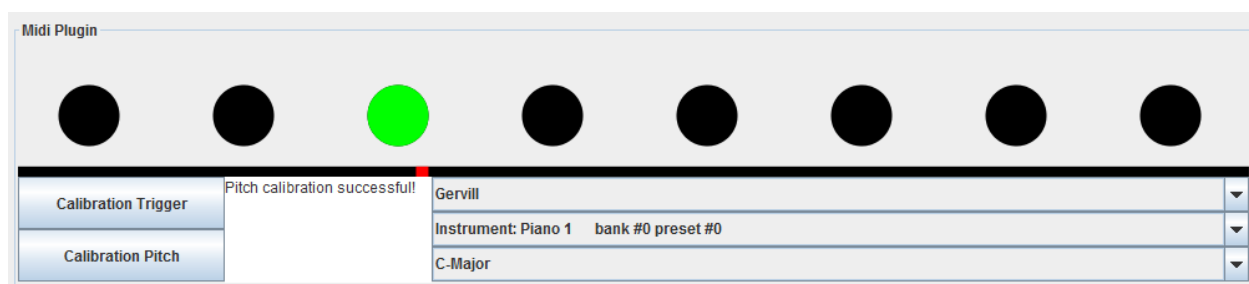
- Input Port Description:  
input [string]: The path of the image file, which will be displayed.
- Event Listener Description:  
clear: Removes the image from the component.
- Event Trigger Description:  
clicked: The event is triggered when the user clicks on the component.
- Properties:  
caption [string]: Caption of the component.

default [string]: The path of the image file, which is displayed after start.

backgroundColor [integer] (combobox selection): Defines background color.

## 2.4.7 MidiPlayer

The MidiPlayer component can generate MIDI tone output on the default MIDI device of the system. This can be used for audio feedback (e.g. for a selection or click) or for creation of musical instruments. The MidiPlayer Plugin features an optional GUI where additional tone parameters (instrument, tonescale) can be selected.



**Figure 12: GUI of the MidiPlayer plugin, displaying MIDI device information and triggered tone**

- Input port Description:  
trigger [integer]: The input strength (tone trigger).  
  
pitch [integer]: The tone height input.
- Properties:  
triggerThreshold [integer]: The value of the tone trigger threshold. Tones will only be generated if the trigger input is bigger than this value.

triggerMax [integer]:

The maximum value of the trigger input. This value influences the volume of the tone. (Tones will get louder as the trigger input value approaches the triggerMax value)

pitchMin [integer]: The minimum value for pitch input. Defines the pitch input value for the lowest tone output

pitchMax [integer]: The maximum value for pitch input Defines the pitch input value for the highest tone output.

toneScale [string] (combobox selection): One of several tone scales (currently, C-Major, D-Major and all 127 tones are provided).

displayGUI [boolean]:

If this property value is set to true, the GUI window for the MidiPlugin will be displayed. The GUI window allows to set advanced properties like instrument name, tonescale, trigger calibration procedure and shows visual feedback for incoming values and tones.

## 2.4.8 ModelSwitcher

The ModelSwitcher component allows switching from the running model to another model which will be deployed and started. This makes it possible to build menus for different use-cases or switch from one use-case to another.

- Input port Description:  
modelName [string]: The name of the model (including extension, for example "CameraMouse\_sensitive.acs". The switch is performed as soon as the model name is received. The model must exist in the ARE/models folder of the runtime environment.
- Event Listener Description:  
switchModel: an incoming event on this ports will switch to the model which is specified under the property "model".
- Properties:  
model [string]:  
A fixed model name can be given here. This model must exist in the ARE/models folder of the runtime environment. The model switch is performed when the switchModel event is received.

## 2.4.9 PlatformDigitalOut

The PlatformDigitalOut plugin operates the output ports of the final prototype of the AsTeRICS Personal Platform. The output ports 1-2 are open-collector outputs with a deactivated pull-up resistor.

- Requirements:  
This component requires the Core Expansion CIM (CIM Id: 0x0602) to operate (see CIM protocol specification [1]).
- Input Port Description:  
command [string]:  
The plugin reacts to incoming action strings starting with "@DIGITALOUT:" and a command. Valid commands are "set", "clear", "toggle" and "press". The command has to be followed by a comma and the port number, for example: "@DIGITALOUT:set,1" or "@DIGITALOUT:toggle,2".

The following examples illustrate the available action strings:

"@DIGITALOUT:set,1":Output port 1 of the Personal Platform will be set (5V)

"@DIGITALOUT:clear,2":Output port 2 of the Personal Platform will be cleared (0V)

"@DIGITALOUT:toggle,1":Output port 1 of the Personal Platform will be changed

- Event Listener Description:  
setOutput1 to setOutput2:  
An incoming event on these ports will cause the corresponding output port to go to the high level.  
  
clearOutput1 to clearOutput2:  
An incoming event on these ports will cause the corresponding output port to go to the low level.

#### 2.4.10 PlatformLCD

The update to the new platform of the final prototype also required changes in the PlatformLCD component. The PlatformLCD component handles interaction with the display and its touchscreen and provides methods to implement a menu on it. The component allows other components to send messages to the display.

- Requirements:  
This component requires the Core CIM (CIM Id: 0x0602) of the AsTeRICS Personal Platform.
- Input Port Description:  
input [string]: The Input of the port will be displayed on the Core CIM's display

#### 2.4.11 RemoteTablet

The RemoteTablet component interfaces the AsTeRICS Personal Platform to a second computer via the HID actuator CIM (USB dongle, plugged into the target computer).

This plugin acts similar to the RemoteMouse plugin, with the difference that the coordinates are set absolutely via the table device, not relatively like a mouse device. The HID actuator emulates a standard USB Tablet on the target computer (no special driver software is needed).



The Tablet x-position, y-position, press/release actions of Tablet buttons and wheel movements can be controlled via desired input values and event triggers. Note that multiple instances of the Remote components (RemoteJoystick, RemoteKeyboard, RemoteMouse and RemoteTablet) can be used concurrently with one HID actuator USB dongle, e.g. to provide different key actions for up to four different input devices on the target computer.

- Requirements:  
The HID Actuator CIM (CIM ID 0x0101) has to be plugged into a free USB port of the target computer and the cable has to be connected to the AsTeRICS platform (see CIM protocol specification [1]).

- Input Port Description:  
mouseX [integer]: The desired X-Position of the mouse

mouseY [integer]: The desired Y-Position of the mouse

action [string]:

Input port for a command string. This command string allows to modify the action of the next left mouse click – it can be set to trigger other types of mouse clicks.

Following command strings are accepted:

“@MOUSE: nextclick, right”: next left click event will create a right mouse button click

“@MOUSE: nextclick, double”: next left click event will create a double click

“@MOUSE: nextclick, middle”: next left click event will create a middle button click

“@MOUSE: nextclick, drag”: next left click event will hold the left mouse button

“@MOUSE: nextclick, release”: next left click event will release the left mouse button

“@MOUSE: action, enable”: enables all mouse actions

“@MOUSE: action, disable”: disables all mouse actions

“@MOUSE: action, toggle”: enables / disables all mouse actions

- Event Listener Description:  
leftClick:  
An incoming event at this port creates a mouse button click. A left click will be generated, unless a valid “nextclick ...” command has been received at the cmd input which changed the type of the click (see above)  
  
middleClick: A click with the middle mouse button is generated.  
  
rightClick: A click with the right mouse button is generated.  
  
doubleClick: A double click with the left mouse button is generated.  
  
dragPress: The left mouse button is pressed (but not released again).  
  
dragRelease: The left mouse button is released.  
  
wheelUp: The mouse wheel is turned one position from the user.  
  
wheelDown: The mouse wheel is turned one position towards the user.

- Properties:  
absolutePosition [boolean]:  
Currently not supported. All mouse position input values are interpreted as relative changes of the position (X- and Y-movement).

### 2.4.12 SyntheticVoice

The Synthetic Voice component uses the SAPI 5 technology to generate synthetic voice.

Requirements: The appropriate voice should be installed on the platform

- Input Port Description:  
input [string]: The text sentence, which will be converted into speech.
- Properties:  
volume [integer]: Defines the volume of the voice. The volume property values should be between 0 and 100.  
  
speed [integer]:  
Defines the speed of the voice. The speed property values should be in range between -10 and 10.  
  
voice [string]: Specifies the voice used for the speech synthesis.  
  
xmlTags [boolean]: Defines if the XML tags in the input text will be supported.

## 2.5 Other Changes in the Plugin Implementations

Additionally to the plugin updates and improvements which were described in the previous sections, some minor changes have been applied to a number of plugins since the delivery of D4.3. These changes were mostly related to a better usability of the respective plugins, for example renaming of the plugin functions and properties to make their use more intuitive, or addition of certain features which proved reasonable during the system tests and user evaluations. The following table gives a summary of the changes:

Plugin Type	Plugin Name (Version D4.3)	Changes in PT-2
Sensor	Adcinput	Renamed to AnalogIn Adaptions for the new AnalogIN CIM hardware
	EventKeyboard	Renamed to ButtonGrid
	Generator	Renamed to SignalGenerator Added new waveform type (random signal)
	IMU	Renamed to RazorIMU
	Keyboard Hook	Renamed to KeyboardCapture
	Joystick Hook	Renamed to JoystickCapture
	Mouse Hook	Renamed to MouseCapture
	Webcamera	Renamed to FaceTracker LK New features: camera settings window display, selection of the image capture resolution, positioning of the live-image window, increased compatibility with different camera products
	ASM FaceDetector	Renamed to FaceTrackerCLM New features: camera settings window display, selection of the image capture resolution, positioning of the live-image window, increased compatibility with different camera products, new gesture detection features can be used (eg. eyeblinkg, eyebrow raised)
Processor	CenterZone	Renamed to DeadZone Replaced the “waitForBoth” property with input port synchronization properties
	CrossCorrelation	Removed due to unsupported vector format in other plugins
	Comparator	Replaced the “waitForBoth” property with input port synchronization properties
	DoubleDispatcher	Renamed to ConstantDispatcher New feature: autosend value at model startup
	Epoch Averaging	Removed due to unsupported vector format in other plugins
	Epoch Cutting	Removed due to unsupported vector format in other plugins
	FFT	Removed due to unsupported vector format in other plugins
	FIR1 – FIR 8-500	Will be replaced by one configurable filter element with magnitude output
	MatrixProduct	Removed due to unsupported vector format in other plugins
	Mathevaluator	Replaced the “waitFor...” properties with input port synchronization properties
TremorReduction	Replaced by BasicTRAlgorithms (featuring 3 different TR approaches)	
Actuator	DAC Output	Removed because not needed in PT-2 due to deprecated analog out CIM
	GPIO Output	Split in General Purpose Input CIM (GPI) and General Purpose Output CIM (GPO), to address the hardware changes in course of the PT-2 developments
	HID Joystick	Renamed to RemoteJoystick New features: additional axis and buttons available, compatibility to the Playstation3 Sixaxis controller to enable accessible gaming configurations
	HID Mouse	Renamed to RemoteMouse Replaced the “waitForBoth” property with input port synchronization properties
	HID Keyboard	Renamed to RemoteKeyboard

**Table 7: List of changes in the plugin implementations from D4.3 to D4.6b**

## 3 Implemented AsTeRICS Configuration Suite (ACS) Examples

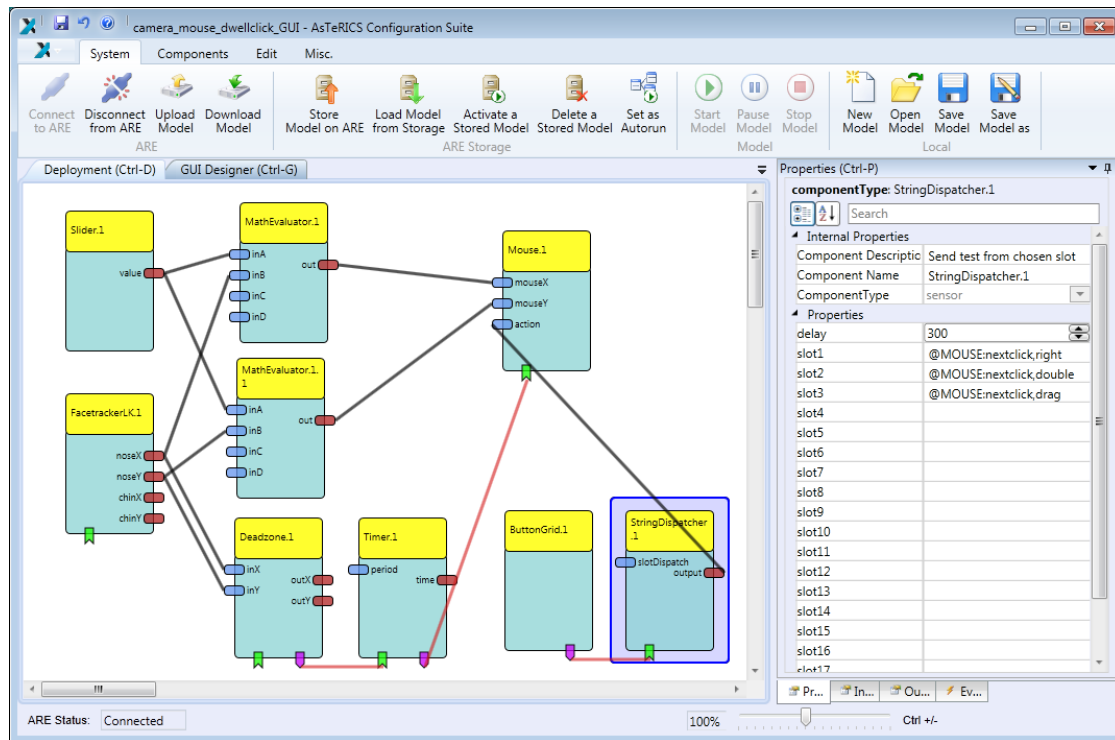
In the following sections, working examples of AsTeRICS models for the runtime environment will be described. These models utilise some of the plugins which have been newly developed or considerably updated in the PT-2 development phase. Nevertheless, all use case demonstrations which have been described in Section 3 of D4.3 are still valid and can be implemented with the final AsTeRICS system, too.

All of these models have been designed using the final prototype of the AsTeRICS Configuration Suite and combine sensor-, processor- and actuator plugins described in section 2, to form functional Assistive Technology scenarios.

### 3.1 A Camera-controlled Mouse with GUI

This AsTeRICS model provides a functional head tracking based mouse-emulation. Click selection and acceleration control are possible via a dedicated Graphical User Interface, built with the GUI-plugins. The creation of this model is described in detail in the AsTeRICS User Manual. The main functional components of the model are:

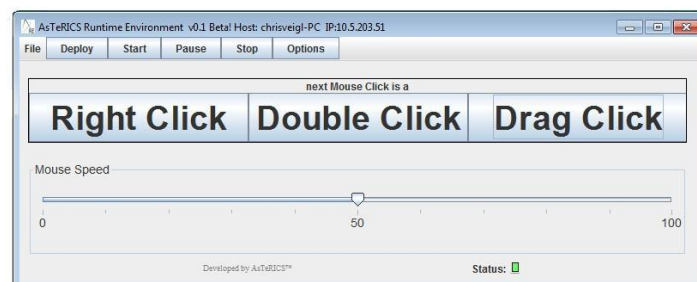
- The FacetrackerLK sensor plugin, which delivers head movement information from a web camera
- The Mouse actuator plugin, which allows controlling the local computer mouse cursor and clicking behaviour
- The GUI plugins Slider and ButtonGrid to control the next mouse click-type and the acceleration
- A DeadZone and a Timer plugin to created mouse clicks when the movement drops under a certain level for a specified timespan (dwelling)



**Figure 13: ACS model of the head-tracking controlled mouse emulation with clicking support**

The above screenshot shows the model in the AsTeRICS configuration Suite. The data flow is initiated in the FacetrackerLK plugin, which delivers x/y coordinates of the head movement (respectively the nose position). These coordinates are scaled to a desired acceleration using the Mathevaluator plugin and the Slider GUI element. Then, the x/y values are delivered to the Mouse actuator plugin. Please note that the RemoteMouse plugin could be used in the same way to control a mouse function of a different computer, where the AsTeRICS HID actuator is attached.

The mouse click functionality is established via the Deadzone plugin, which detects the movement activities in the x/y coordinates and starts the Timer plugin if the movement is below an adjustable threshold. The timer is stopped as soon as the movement exceeds the threshold again. If the timer stays active for an adjustable timespan (e.g 1,5 seconds), it fires a click event to the mouse plugin. The type of the next click can be selected via a ButtonGrid element, which sends corresponding action strings via the StringDispatcher plugin to the Mouse plugin.



**Figure 14: Runtime GUI for click selection and acceleration control of the head tracking controller mouse emulation**

### 3.2 A Sip/Puff-controlled Mouse with Clicking Functions

This AsTeRICS model uses the Arduino – an affordable and easy-to-use physical computing platform – to digitize analogue values of a pressure sensor, thereby creating a DIY-sip/puff sensor. Please note that such a solution is not medically certified and that special care has to be taken to keep the plastic tubing and used mouthpieces clean. An appropriate membrane to avoid spoiling the sensor with bacteria has been developed by AsTeRICS partner KI-I and is described in the WP3-deliverables.

The sip/puff values represent a 1-dimensional, analogue input to the system which is usually mapped to one degree of freedom of the controlled application (for example the x-coordinate of the mouse). The challenge was to enable full mouse control (x/y and clicking) just by this one-dimensional input.

The AsTeRICS model is composed of the following functional plugins:

- The Arduino plugin delivers the digitized values of the pressure sensor
- A Mathevaluator plugin scales the values according to an adjustable acceleration factor
- A Deadzone plugin is used together with Timer plugins and a PathSelector plugin to create mouse activities. Here, the first timer triggers the Pathselector to switch between X- and Y- control: a short inactivity in the sip/puff changes from X- to Y- axis and back, which is signalled via a sound (Waveplayer plugin).
- Mouse clicks are performed by a longer time of inactivity via the second timer.

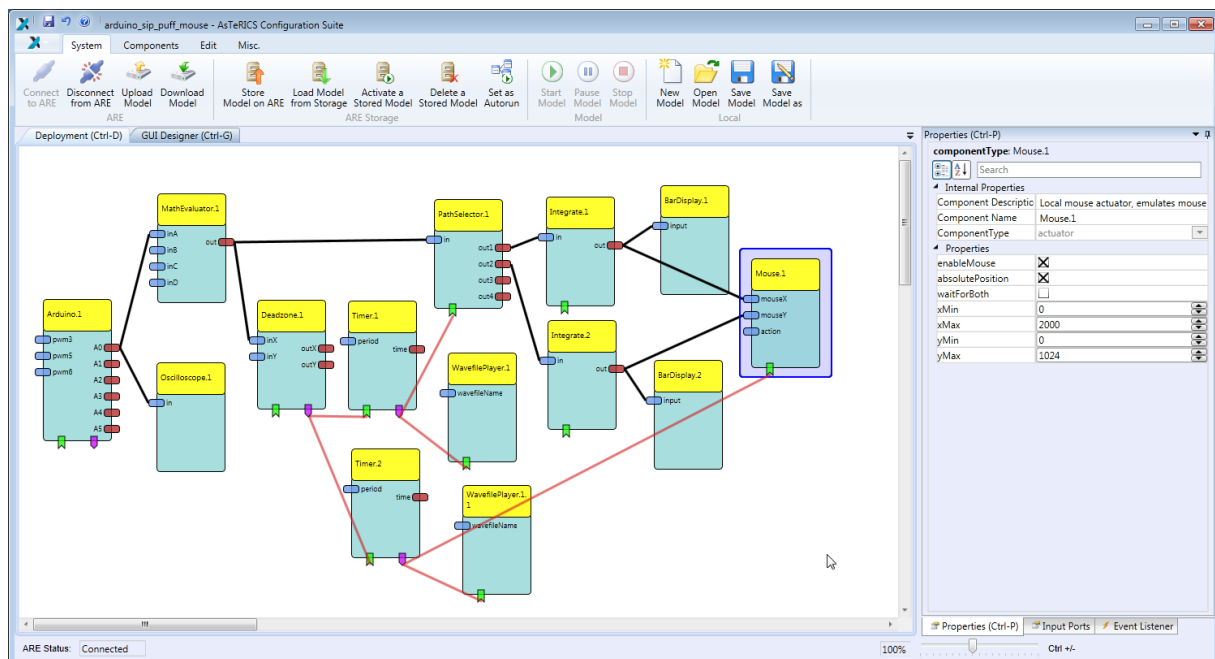


Figure 15: ACS model of the sip/puff controlled mouse emulation

### 3.3 Mouse Control by One Switch

This model provides computer mouse control by one single switch input. When the user presses the switch, mouse actions are scanned in the following sequence: left, right, up, down, left click, right click, double click. The highlighted action is displayed in the TextDisplay component. When the action chosen by user is highlighted, the user has to release the switch to run it.

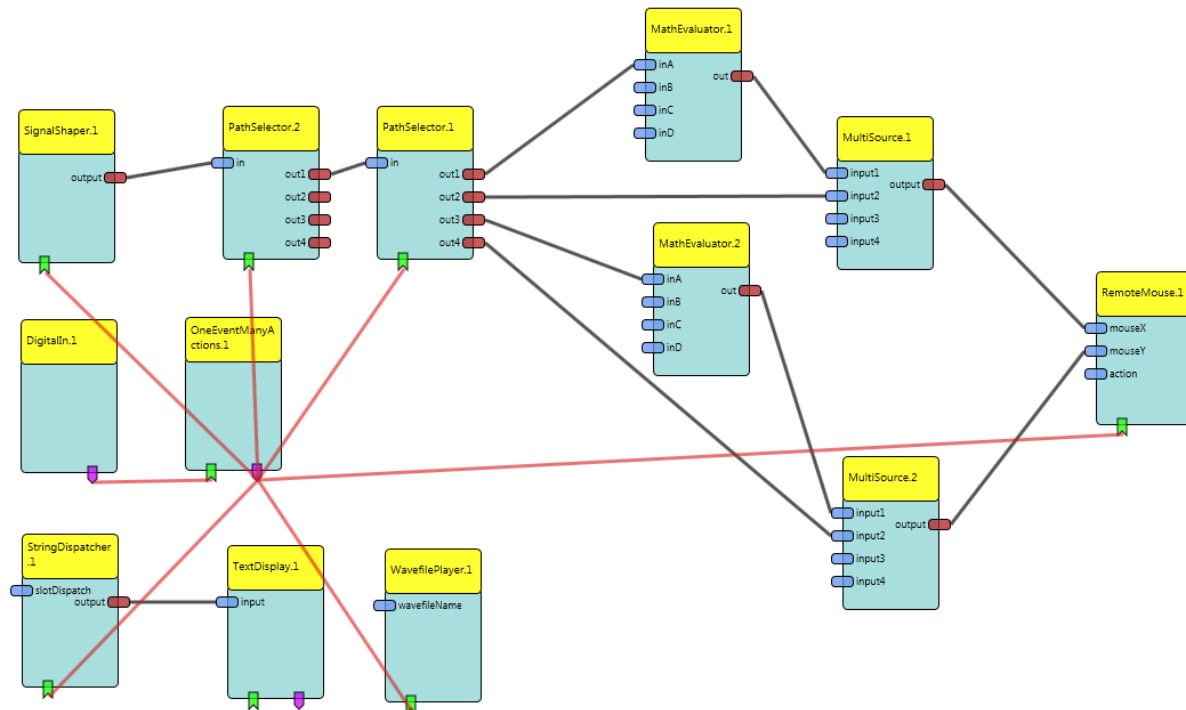


Figure 16: ACS model for a 1-switch controlled mouse

The AsTeRICS model is composed of the following functional plugins:

- The OneEventManyActions component works in mode 1. It provides the scanning operation and controls other components.
- The SignalShaper component models the mouse speed. At the beginning, the mouse moves slowly to allow the user a precise placement of the mouse pointer. Then, the speed of the mouse pointer increases to allow the user efficient mouse pointer navigation over long distances in a short time.
- TextDisplay and WavefilePlayer components provide visual and auditory feedback
- DigitalIn reads the state of the switch.
- RemoteMouse controls the mouse on a remote computer via the wireless HID actuator.

### 3.4 Send and Receive SMS using a GSM modem

This model controls a GSM Modem and enables the user to send predefined SMS messages. To send a message, the user has to press the switch to activate the scanning. When the desired message is displayed, the user presses the switch again to send this message. Received messages are displayed and automatically read by the speech synthesizer.

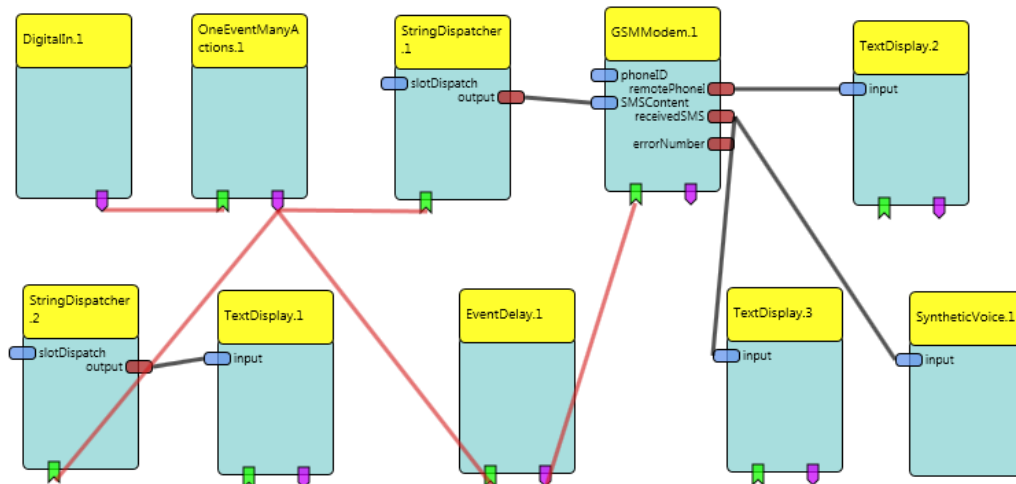


Figure 17: GSM modem model

The AsTeRICS model is composed of the following functional plugins:

- The OneEventManyActions component works in the mode 1. It provides the scanning operation and controls other components.
- DigitalIn reads the state of the switch
- GSMModem controls the GSM modem connected to the platform
- SyntheticVoice speaks the incoming messages via the voice synthesizer.
- Various TextDisplay components display: the message to send, an incoming message and the sender phone number.



Figure 18: The used USB GSM modem



### 3.5 Accessible Gaming Configurations

The following two examples will show how to control games on a Sony PlayStation 3 (PS3) with AsTeRICS. The main component, controlling the PS3 is the HID Actuator hardware module represented in the ACS by the plugin *RemoteJoystick*. Both models generate the control signals and send them to the Joystick.

#### 3.5.1 Accessible “Marvel” Pinball

The first chosen game was Marvel Pinball<sup>2</sup>, where only three buttons are needed to control the game. The player needs to trigger the left and right paddle and push one button to insert the ball. These functions were implemented using three push buttons, connected to a General Purpose Input/Output module (*DigitalIn.1*), which can be positioned according to the requirements of the gamer. In the ACS model (see Figure 19) these push buttons were mapped to the PS3 buttons (*RemoteJoystick.1*), needed to control the game.

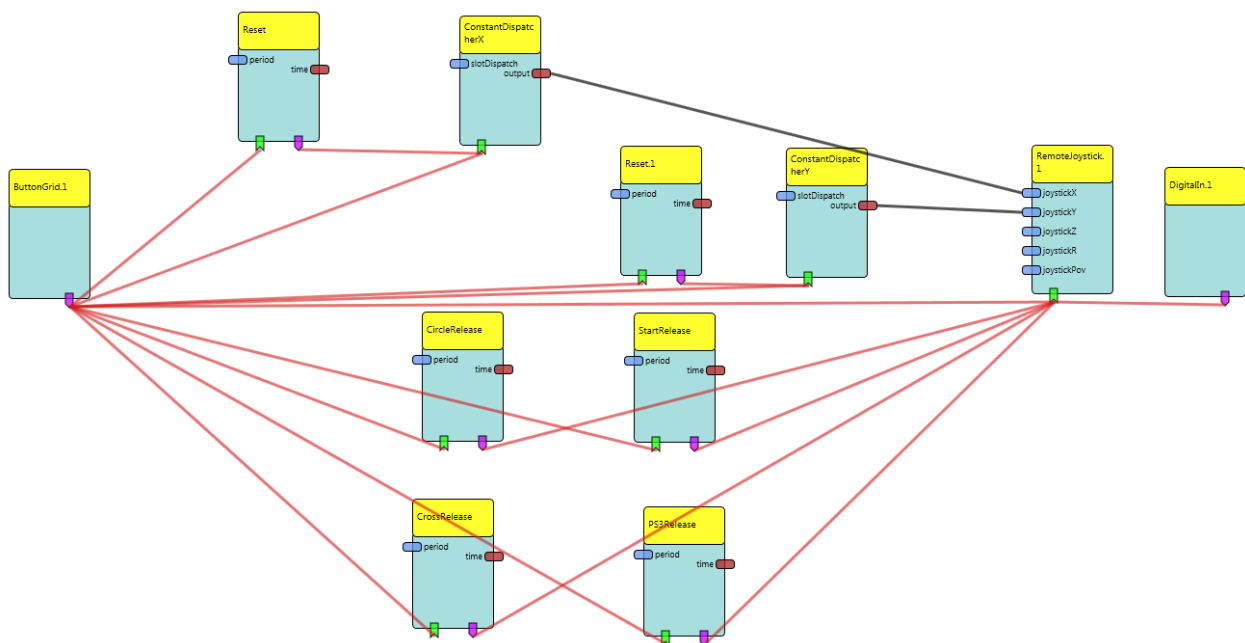


Figure 19: The Marvel Pinball Model

Additionally, the gamer must be able to browse menus, to actually start the game. Therefore he/she needs to move up, down, left, right, select menu items and jump to the previous menu. In the ACS model, *ButtonGrid.1* is used to display the buttons, needed to navigate the menus. Instead of the *ButtonGrid*, OSKA (see 2.3.13) could also be used. The *ButtonGrid* (as shown in Figure 20) is on the one hand directly connected to the *RemoteJoystick* to trigger the button-pressed events, on the other hand connected to timers, which trigger a button-released event after 300ms. This behavior is needed, because the PS3 filters out very short button events.

<sup>2</sup> <http://marvelpinball.com/>

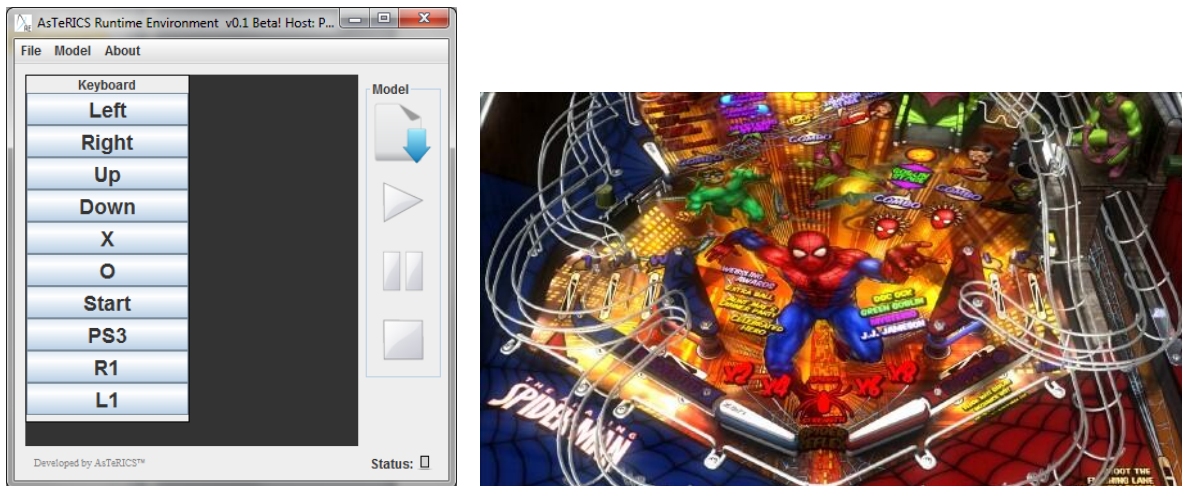


Figure 20: The ButtonGrid to control the PS3, application in Marvell Pinball

### 3.5.2 An Accessible Racing Game: “ModNation Racer”

This AsTeRICS model uses the Kinect plugin and the RemoteJoystick plugin to enable people who cannot use the original gamepad of the PS3 but can still move their hands to play a racing game – in this example the game *ModNation Racer*<sup>3</sup>. As input the model uses the Kinect platform to get the hand and head position of the user. The basic usage of the model is as follows: The user sits in front of the Kinect. To steer to the left, the player’s left hand must be lower than the right hand. To steer right, the right hand must be lower than the player’s left hand. To brake, the right hand must be near the body, to throttle, the hand must be away from the body (more than 20 cm).

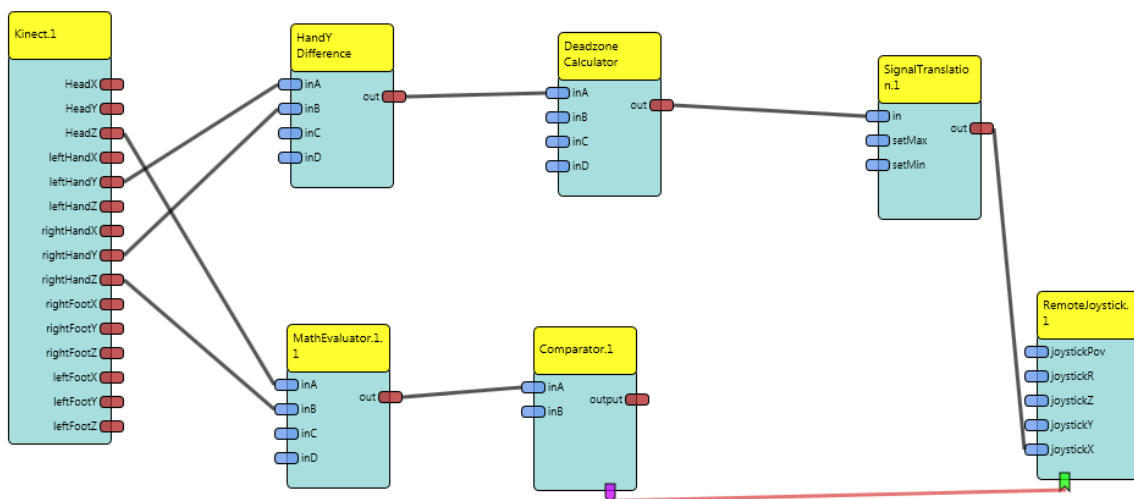


Figure 21: ACS model of the ModNation Racer control model

The steering is modelled with 2 Math Evaluator plugins and the SignalTranslation plugin. The first MathEvaluator, labelled “HandY Difference” in Figure 21, calculates the difference of the Y coordinate of the left and right hand. The second MathEvaluator uses this value and calculates a deadzone around the 0 value with a radius of 30 pixels and divides the

<sup>3</sup> [http://www.modnation.com/en\\_gb/index/index.htm](http://www.modnation.com/en_gb/index/index.htm)

difference by 1.5 to get a lower sensitivity for the steering of the car. Finally, the SignalTranslation plugin is used to limit the values to the range between 0 and 255, which are the bounds for the RemoteJoystick plugin's x values.



**Figure 22: ModNation Racer, played via the AsTeRICS Kinect interface**

The throttle control uses one MathEvaluator and one Comparator to calculate the distance between the right hands Z coordinate and the users head Z coordinate and checks if the value is over 200. If that is the case the car accelerates, otherwise it slows down and after some time it goes backwards

### 3.6 Additional Demo Scenarios and Use Cases

The examples in section 3 only focus on some aspects of what is possible with the final prototype of the AsTeRICS system. It is simply not possible to write a comprehensive description of all system configurations and applications in this deliverable.

However, a User Manual has been created which describes the creation process of AsTeRICS / ACS models in detail and gives further examples. The User Manual can be downloaded from the AsTeRICS homepage (<http://www.asterics.eu>), together with the runtime- and source code versions of the AsTeRICS software and the Developer Manual.

In addition to the plugins and examples described in this deliverable and in the User Manual, the AsTeRICS consortium members are still working on the implementation and test of plugins and models in two areas requested by the project reviews in terms of the first full review report: Improved Speech Recognition Integration and Gesture Recognition Integration. Both areas are covered within the WP6-efforts and will be reported in the respective WP6 deliverables.

## References

- 1 AsTeRICS Description of Work, technical Annex 1 or the AsTeRICS Grant Agreement
- 2 TOBI: Tools for Brain Computer Interaction, European Integrated Project;  
<http://www.tobi-project.org/welcome-tobi>
- 3 Biosig: An open source software library for biomedical signal processing, featuring for example the analysis of biosignals; <http://biosig.sourceforge.net/>
- 4 BCI2000: A general-purpose system for brain-computer interface (BCI) research;  
<http://www.bci2000.org/BCI2000/Home.html>
- 5 OpenVibe: Software for Brain Computer Interfaces and Real Time Neurosciences;  
<http://openvibe.inria.fr/>
- 6 AsTeRICS Deliverable D2.3 “Report on API specification for sensors to be integrated to the personal platform”
- 7 AsTeRICS Deliverable D4.3a “Prototype 1 of Signal Processing Modules - Algorithms”
- 8 AsTeRICS Deliverable D4.3b “Prototype 1 of Signal Processing Modules - Plugins”
- 9 AsTeRICS Deliverable D4.4 “Final Prototype of the AsTeRICS Configuration Suite”
- 10 AsTeRICS Deliverable D4.5 “Final Prototype of the AsTeRICS Runtime Environment”
- 11 The Exponential Smoothing Algorithm: [http://en.wikipedia.org/wiki/Exponential\\_smoothing](http://en.wikipedia.org/wiki/Exponential_smoothing)
- 12 The ENCOG machine learning framework : <http://www.heatonresearch.com/encog>