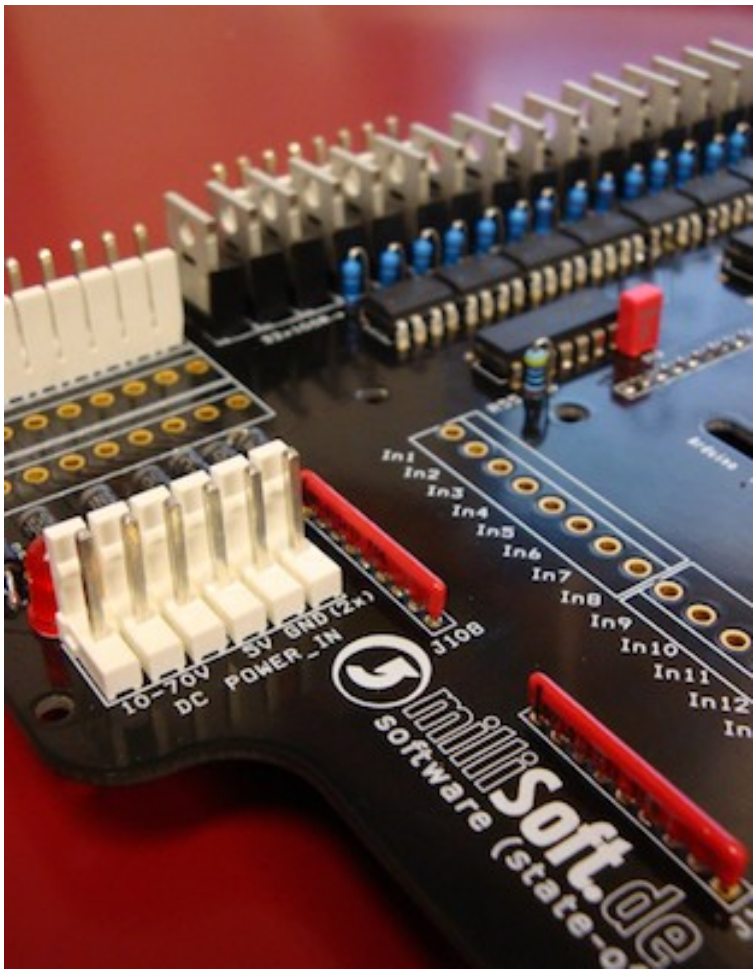
PINterface

-USB pinball interface –

Manual

version 25.09.11

milliSoft Softwareentwicklung Saar GbR Hewener, Speicher
info@milliSoft.de
www.milliSoft.de



Content

PINterface – general information

This guide includes information and tips about hardware, software and mechanics for the construction of a custom pinball mashine when using the PINterface system. With this manual it should be possible for anyone to build an individual Pinball and thus to obtain a globally unique pinball mashine.
Below you will find a general description and more detailed information about the structure and use of PINterface system.

The "milliSoft PINterface" is a system for controlling and programming a custom pinball mashine. The set offers hardware boards for a power supply, to control the solenoids and detect input switches, a customizable software and this guide with tips for the custom pinball construction.

The best times for pinball mashines was in the 80's and 90's and faded after the withdrawal of the largest manufacturers at the end of the 90's. Today only the company "Stern Pinball" produces a few different models per year that are often being used "home use only".
For private collectors the model variety is reducing more and the desire to build an own unique pinball mashine from spare parts is rising. For most hobby engineers, this wish fails at hardly reprogrammable electronics based on existing commercial systems; especially game play (-rules), the graphic and acoustic output cannot be changed.
For these enthusiasts this pinball construction kit was developed and provides boards for electronic control and power supply, and parameterizable software to integrate own sound, graphics and videos for your individual mashines.
Our development includes therefore a complete concept, hardware and software: the concept provides a detailed guide and components proposed by us, and tips to building the base system. The boards developed for the control of flippers-like solenoids, game field solenoids and flashers use a microcontroller board, which connects the interface to a PC. On this the game control is based on the included software PINterface operation system, outputs MP3 stereo sounds and music or presents graphics and video animations (DivX, XVid,...) on a LCD screen.
This solution allows free configuration of game control, sound and graphics output based on a XML configuration file. No programming skills are necessary to implement the custom pinball game. You only need a good idea to create something unique and some design ideas and motivation to implement your idea.
This hardware relies on standard PC hardware for game control and graphical/sound output so the electronics is affordable and can be used by hobbyists and collectors of pinball mashines.

Warning

You may only work on the system, if you read the manual and understood the system operation!
Working at pinball machines and with these electronics can cause fatal electrical strokes or mechanical hazards if not used properly at some circumstances. Only use this system, if your education and skills in dealing with highly voltage electronics are sufficient, and you have sufficient experience in maintaining and repairing pinball mashines.

Errors and omissions

If you notice any omissions or errors in this development or documentation, or have any suggestions for improving this documentation, please contact milliSoft with specific information about the improvement and the number of pages of the manual. We correct all reported errors or omissions.

Safety information

Introduction

The following precautions must be fulfilled during all phases of operations, service and repair of the system. Failure to comply with this security or certain warnings elsewhere in this guide is not given the correct and safe use of the system.

Human safety

The work with the PINterface system may only be done by persons with an education or similar qualifications for working with high-voltage systems. These people must be aware of the dangers of such.

"Danger":
In this system, hazardous voltages can occur which can lead to the death of the user. Special care is needed when connecting, maintaining, testing and operating the system.

Conventions used in this manual

The symbols, labels and conventions are described in this chapter.

Safety information can be classified into different levels of security measure. The different levels are characterized by a pictogram that precedes the message. Known or potential hazards are identified in one of three types:

"Danger" shows the existence of a risk and serious injury or death are possible if the statement is ignored.

"Warning" is used to indicate a hazard that can cause serious injury and damage to property, if the statement is ignored.

"Caution" is used to indicate the existence of a threat and to signal that slight bodily injury and property damage can occur, if the statement is ignored.

Mechanical safety

"Warning":
Always switch the device off before you work on the wiring or electrical components.

"Warning":
Ensure sufficient ventilation of the system, to ensure the cooling of all components. Block no ventilation areas! When the system fails and overheats, it can lead to fire.

Electrical safety

Follow these guidelines to minimize shock hazards, if you are working with the system:

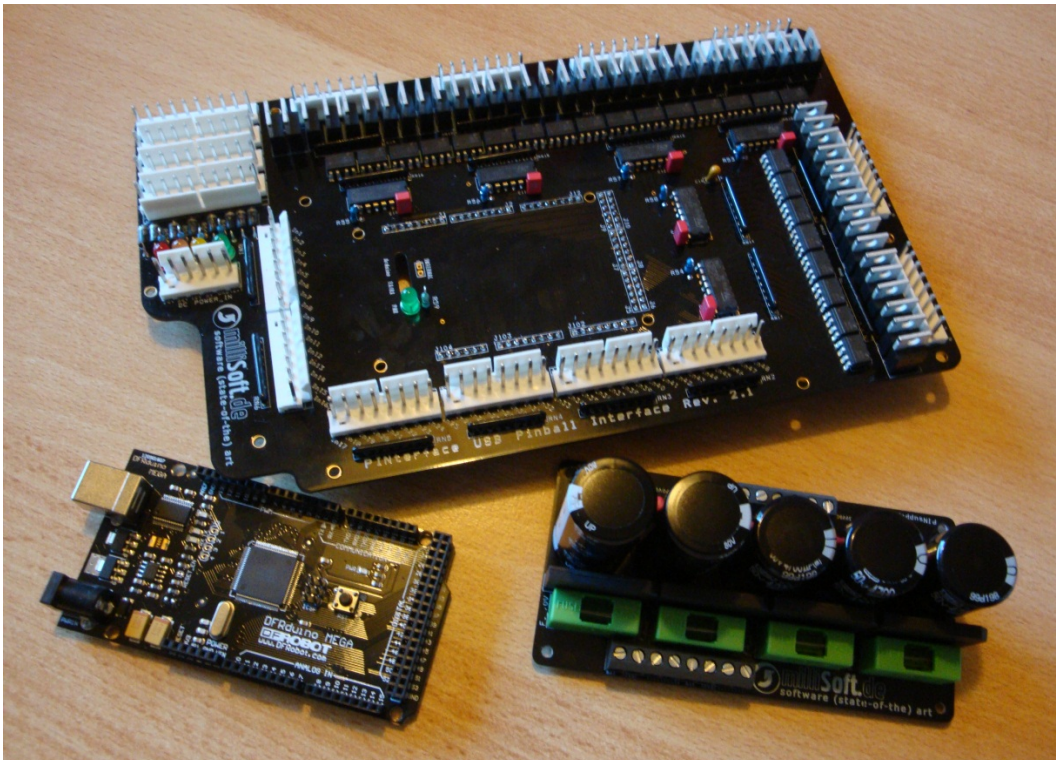
"Danger":
This equipment contains dangerous voltages which can lead to serious injury or death. If shortcomings are observed or malfunctions occur, stop your work and check with a qualified professional. Information can be provided directly by milliSoft.

"Danger":
The electrical circuits can save streams in the capacitors after switching off the instrument for a long time, so there is still a risk for the period after the separation of the mains voltage!

To avoid injury:

- Ensure an adequate grounding and connect the system to a reliable grounding.
- Place no liquids on or through the system. Spilled liquid can reach electrical items and increase the risk of electric shock.

Contents of the PINterface package



Your PINterface package contains

- A PINterface board „IO board" for connecting the components (switches, solenoids,...)
- A PINterface interface board for the rectification of the transformer voltage (Rectifier-Board)
- License to use the software „PINterface OS" for the regular pinball operation interpreting its own rules of the game
- License to use the software „RuleSetConfigurator" for the graphical configuration of custom game rules
- License to use the software „COMunicator" to test the PINterface hardware
- A printed version of this manual
- A CD with the above software, a digital version of this manual and the tutorial videos
- Free updates of the PINterface software as a Web download from our homepage

First steps

This chapter recommends first steps in the custom pinball mashine construction with the PINterface. If you follow these steps, you learn best about the operation of the system.

PCB assembly

You must assemble the large input and output board and the rectifier power supply before using the PINterface system. As you do so you will find detailed instructions in the chapter "PINterface hardware, Assembly of the Board". The micro controller for USB connection is fully equipped and ready to use with the PINterface. It must only be connected to the input/output Board with its headers.

Building test cables

To understand the function of the board and perform first tests (use of PINterface COMunicator) we recommend to create test cables that can be used to test the inputs and outputs. This way the assembly of the board including its correct soldering can be tested. You can also test custom rules later on without jeopardizing coils and the transistor switches on the Board for rule errors.

Cableing

The power of the board must be connected to check the operation with the test cables. On the one hand, a 5V-DC power source (e.g. using a PC power supply) can be connected directly to the current inputs of the PINterface of input / output Board, or a pinball transformer is connected to the rectifier/power supply board of the PINterface system. For details please refer to the chapter later in this manual or see our tutorial videos.

First tests

Before the first test, it may be necessary that the firmware of the microcontroller must be updated. There may also be a hint to do so in the change list that appears at the top of the software installation when updating to newer software versions. You can see the version of your microcontroller in the PINterface Updater software, which also can update the installed microcontroller version with the latest version just with a mouse click.

For the first tests with the test cables you can use the „COMunicator" software (linked on your desktop):
If the test cables are connected and voltage is apllied, the inputs and outputs can be tested if they function correctly. Closed switches connect the voltage of 5V to the input pins, the COMunicator software displays the switch as enabled. Outputs that are connected to a supply voltage can be tested by the software while being connected to Ground so that a output is activated and the solenoid is fired or a lamp/LED lights up.

There is a also a demonstration video on your PINterface CD showing the COMunicator test software.

Testing demo game rules

The PINterface software comes with predefined demo rules. These include a rule file and a set of graphics, videos and sounds. With these you can see basic modes of operation of a PINterface game. You can also use these rules as a basis for your own rule set after you tested the basic operations (switch input numbers, output activation). Run the PINterface OS and play a bit around with the demo game. Even with missing or incorrect communication settings (COM-port), you can trigger buttons in test mode and understand the gameplay using a computer mouse.

A video on the CD shows the usage of the PINterface OS.

Changing the demo rules

If the underlying inputs and outputs of the PINterface board are tested and you have understood how this is donw you can customize the rules to create your own game. The rules can be edited with the graphical editor „RuleSetConfigurator" or a text editor based on the "rules.xml" file.

There are demonstration videos on the CD to show the use of the rule set editor to modify the demo rules for your own game.

The PINterface hardware

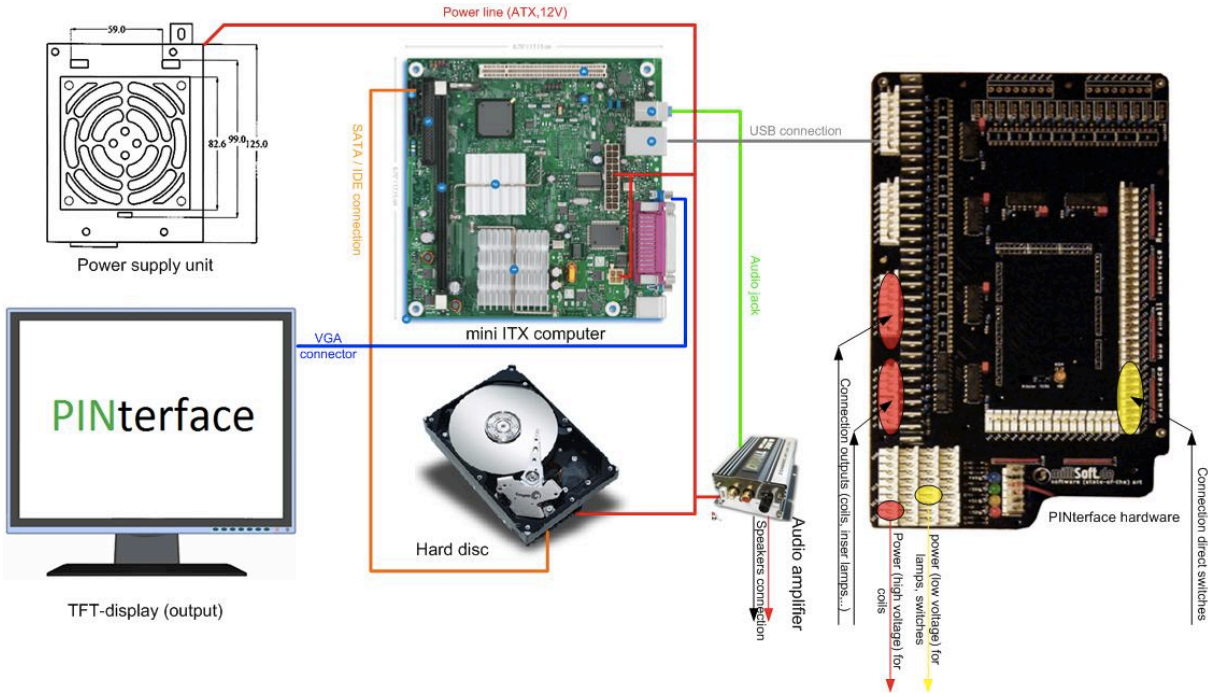
In this chapter you will find details about the PINterface interface hardware setup. The documentation is divided in:

- Working principle of the overall system
- Overview of the neccessary hardware components
- Working principle of the PCBs
- Principles of operation and connectivity of the PINterface interface board

Working principle of the overall system

The PINterface system consists of a PC component that controls and reads data from the PINterface I/O board which is connected to the PC over a USB connection.
The software located on the Windows PC is able to interpret a custom game rule set file to react to inputs of switches connected to the I/O board and timers.
The PC unit uses standard hardware components and can be realized by using an old PC or choosing a new lowcost system that is chosen specifically to be used in the pinball mashine (fanless,...).

The following shematic shows the electronics of the custom pinball that is needed beside the playfield components.
The rectifier board included in the PINterface package is not shown here and has to be included in the power supply path.

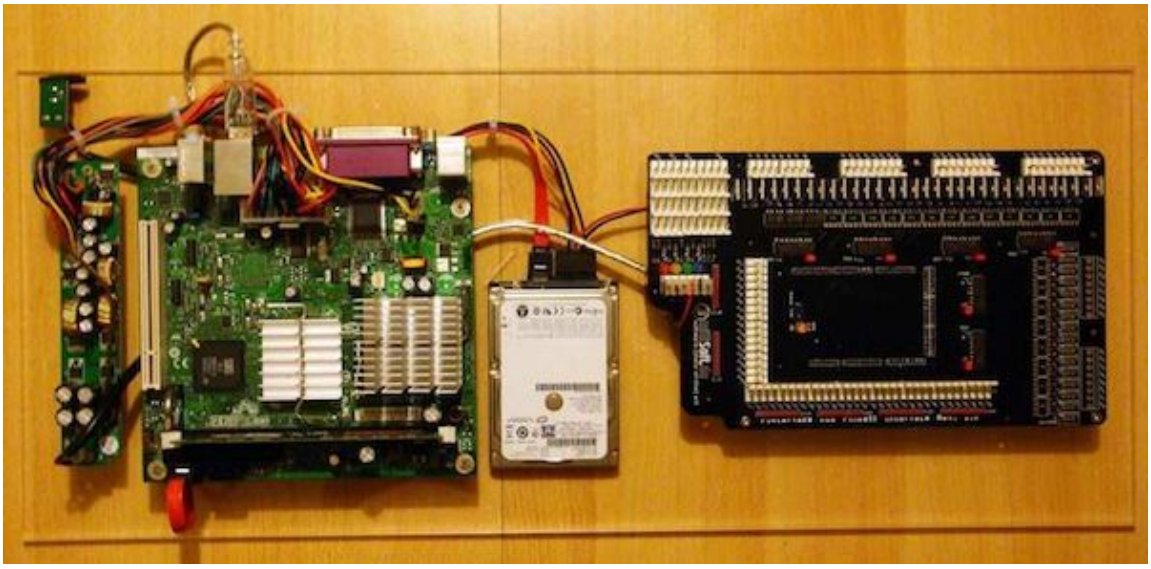


All playfield components are connected to the connectors of the I/O board without needing additional existing pinball electronics except the playfield parts. The Pinball

operating system on the PC takes control over the playfield components (coils, lamps and graphical output on the screen) based on the input events that are detected by the I/O board.

Overview over neccessary components (PC, boards,...)

In this chapter you will find an overview and a description of all the components needed for the operation of your custom pinball mashine. It describes the components that you received with the PINterface kit as well as other necessary components of an old pinball mashine from the electronic side.



The PC-component

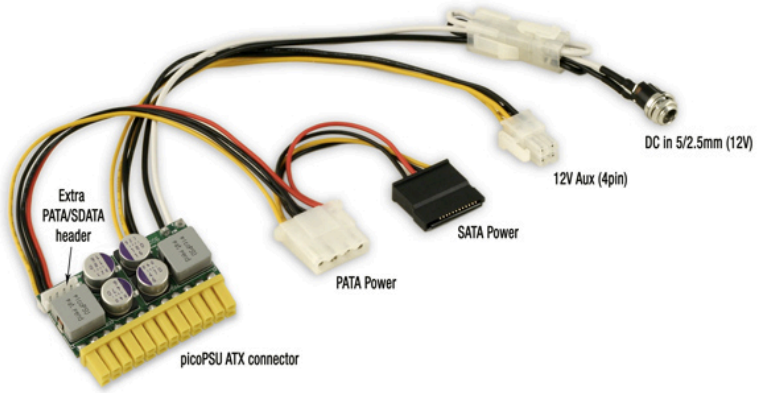
The main processing unit that handles all game transitions and multimedia output is a standard PC (or better an embedded system in miniITX format) powered by an standard processor. Current lowcost and low performance systems like Intel Atom CPU or AMD E350 CPU are working perfectly. You do not need a high tech system for controlling the game. For best results (fast media playback,...) a dual core system with passive cooling is recommended. Such specs are fulfilled for example by the Intel Desktop Board D510MO for a low cost budget of approx. 65€ (http://www.intel.com/content/www/us/en/motherboards/desktop-motherboards/desktop-board-di510mo.html).

passive cooled integrated mainboard with CPU soldered on it for just 65€

Just add RAM and a hard disc to this solution and you have a completely new PC system for under 100 €. A solid state disc would be perfect for a PC system to be included inside a pinball mashine. This way the PC components could be completely without moving parts (fans, mechanical moving hard disc parts) if a fanless power supply is chosen (see below). This makes the PC electronics very shock reistant and dust and dirt cannot block any fans. But the price for a small solid state disc is slightly higher than the price for a hard disc.

The power supply to connect to the PC system can also power 5V to the PINterface IO board to use with the switches. Here it is also recommended to use a power supply that is fanless. A platform like ebay provides a lot of power supplies that come equipped without a fan. Some have the formfactor 1U...

.... and some can be plugged in the ATX connector directly:



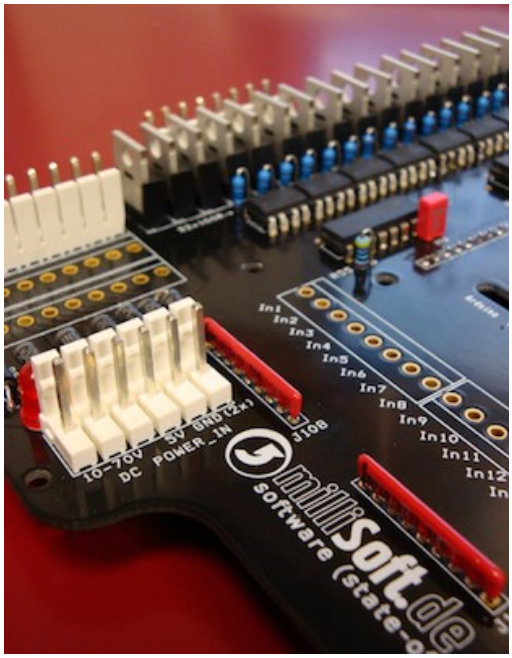
Both are available for approx 30 €.

The PINterface boards

The PINterface interface board allows the connection of up to 48 direct inputs (switches such as targets, rollover) and up to 48 direct outputs (solenoids with high voltage or switchable insert lamps with low voltage).
The communication is done using a microcontroller via USB connection with the PC. This controller is included ready-to-be-used and has to be installed under the PINterface I/O board.
The microcontroller is specially designed with the "milliSoft PINOS" - microcontroller operating system and can only be exchanged by milliSoft in case of a malfunction.

PCB assembly

In the PINterface kit you receive the naked PCBs of the boards with all components needed so soldering is still required to build the system. We used no SMD or small components, so the component placement can be done easier. This way you can also build the boards if you are not a professional in soldering. Alternatively, you can contact us for a preassembled version of the PINterfcace. We can assemble the boards by our partners if you do not trust your own soldering work.



Assembly of the PCBs

Introduction

This guide includes a general section dealing with the soldering work in General, as well as two more parts that respond to specific work on PINterface interface and PINsupply.

General soldering tips

You should be experienced in soldering of PCBs to assemble the PINterface kit. If

you are not experienced please learn soldering by reading some online soldering guides and test your skills by assembling simple PCBs or integrated circuits. (german examples: http://www.Elektronik-kompendium.de/sites/GRD/0705261.htm And http://www.stayathome.ch/Loeten.htm or http://www.qsl.net/dk6rx/dice/6.htm)

Since the board has very thick copper surfaces on the top and bottom, it is recommended to start soldering all connection that are not connected to the copper surfaces (ground and 5V). If all the components are soldered, you can heat the underside of the PCB with a commercial hair dryer to make soldering of these pads easier.

Soldering the PINterface

You should solder groups of components at once, starting with the capacitors and resistors to the IC. The resistors have to be bended, so that it can be installed vertically. The installation position of the ICs is marked in the placement plan and on the boards by a small rectangle on one of the component pages and on the IC by a remote point.

Next you can solder the resistor arrays, because it is harder to reach them later on. The mounting direction is marked by the ground pad on the board (rectangular) and a printed dot on one of the component pages.

Now the series resistors used for the LEDs should be soldered. Bend one end so that the resistor can be installed vertically. The included resistors are designed in a way that they work correctly with the printed voltage values on the circuit board.

- 70V 34K Ohm
- 30V 14,3K Ohm
- 10V 4,22K Ohm
- 5V 1,74K Ohm
- 3,3V 909 Ohm (Microcontroller PWR)

If different voltages are applied, the resistors are to be amended accordingly.

When installing the diodes please check the correct polarity. This is marked by a white or silver ring on the diode and a crossline on the PCB print.

The post connectors for connecting the microcontroller have to be inserted from the bottom of the board and soldered on the top.

After this the diodes have to be set regarding the polarity by a long (anode) and a short (cathode) end. The anode is placed on the board in the rectangular pad.

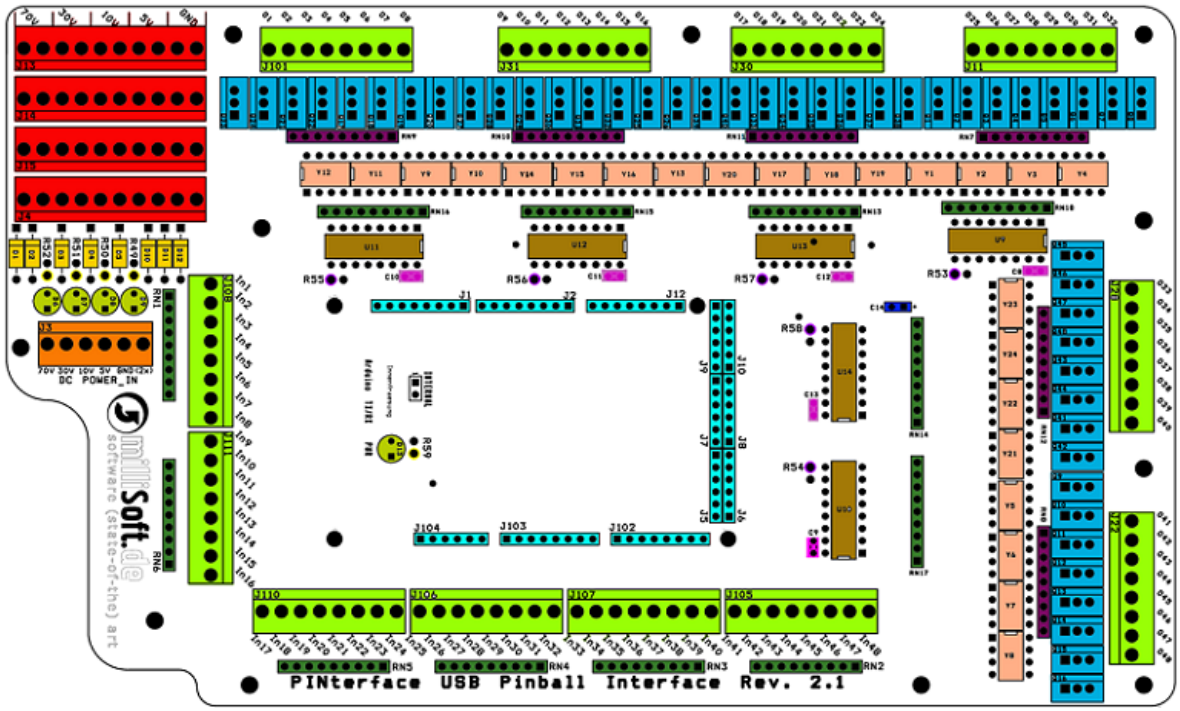
When installing the tantalum capacitors be sure to check the correct polarity. This is shown at the silkscreen on the board and by a small „+"-mark next to the pressure

component and a black dot or line on the side of the component above the respective legs.

Now, the transistors are inserted. You have to check the orientation according tot he marking print on the PCB. This must match the back of the transistor (metallic side) alignment.

Finally, the plugs for input and output of voltages have to be inserted and soldered - Done!

PINterface			
Bauteilbeschriftung	Bauteil	Stückzahl	Farbe im Bestückungsplan
C8-C13	MKS-02 Kondensator 100n	6	
R53- R58	Widerstände ShiftReg 470 Ohm	6	
U9- U13	ShiftReg 74HC595	6	
Y1- Y24	ILD74 Optokoppler	24	
RN1- RN6, RN13- RN18,	Widerstandsarray 8+1 470R	12	
RN7- RN12	Widerstandsarray 8+1 100R	6	
R49- R52, R59	Vorwiderstände LEDs (siehe Beschreibung)	5	
D1- D5; D10- D12	1N5060 Diode 2A	8	
D6- D9; D13	LED low current, div. Farben	5	
J1, J2, J5- J10, J102- J104	Pfostenstecker 1-reihig 40-Fach(Arduino)	3	
C14	Tantal Elko 1uF Tropfen	1	
Q1- Q48	IRF540N MOSFET	48	
J11, J22, J28, J30, J31, J101, J105- J107, J110	MOLEX 8-Fach 26-64-4080	16	
J4, J13- J15	MOLEX 10-Fach 26-64-4100	4	
J3	MOLEX 6-Fach 26-64-4060	1	

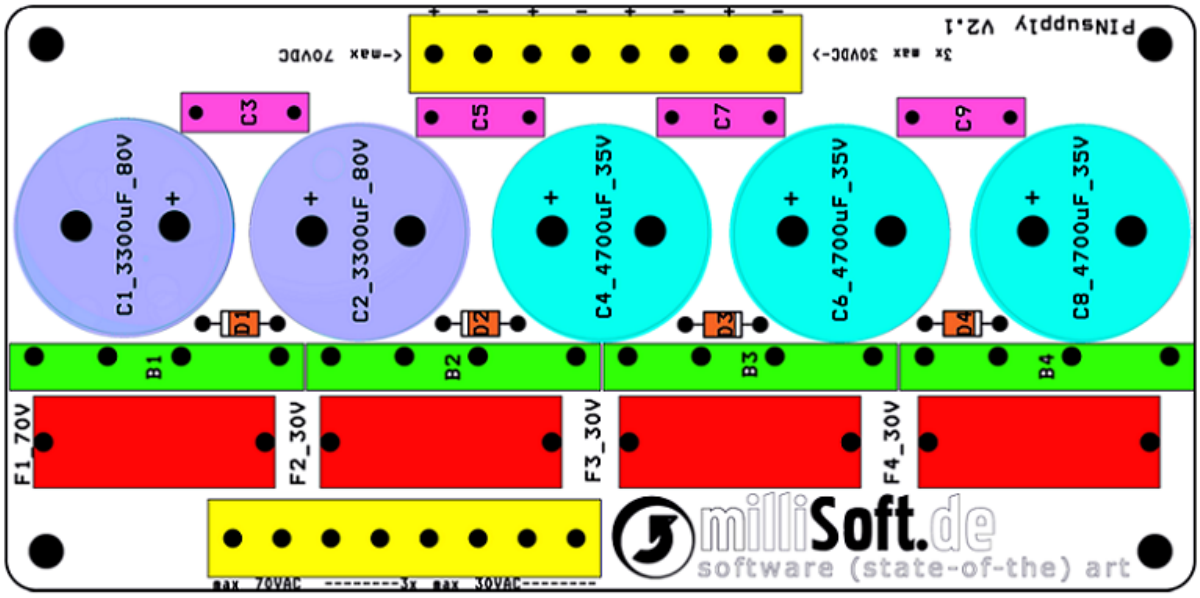


Soldering of PINsupply rectifier

It is recommended to start with the smallest components (diodes + MKS capacitors). The diodes are connected in the reverse direction, the polarity is shown on the silkscreen (see below).

Then the plastic clamps have to be inserted, followed by the fuse holders and the rectifiers. Finally the electrolytic capacitors are inserted. **The polarity has to be checked**, as these electrolytic capacitors are destroyed directly with reversed polarity and this can cause injuries! The positive pad is marked on PCB and silkscreen by a „+" sign. On the electrolytic capacitors the negative side itself is marked.

PINsupply			
Bauteilbeschriftung	**Bauteil**	**Stückzahl**	**Farbe im Bestückungsplan**
J1, J2	Platinenschraubklemmen 8Fach	2	
F1- F4	Sicherungshalter	4	
B1- B4	Brückengleichrichter GBI25D	4	
D1- D4	Dioden 1N4002	4	
C1, C2	Elkos 80V 3300uF	2	
C4, C6, C8	Elkos 35V 4700uF	3	
C3, C5, C7, C9	MKS Kondensator 100n RM10	4	



Principle of operation and connections of the PINterface board

This chapter explains the PINterface board in detail and describes the functionality and connectivity.

Connection of solenoids

All solenoids on the playfield can be connected to the high voltage (i.e. 30V or 70V(max)) with the cathode. The anode is connected to one of the outputs of the I/O board to a transistors. Using the opto couplers, the transistor has a high impedance at rest and no electrical flows exist so that the coil is not activated. The transistor can be switched and the solenoid is connected to ground. Doing so a high current is applied to the solenoid, which causes a strong magnetic field in the coil with its inner metal core that is mechanically moved at high speed and intensity as a result. The solenoid activates. During shut off, the magnetic field breaks down again and the metal core is led back to its starting position by a spring mechanism on the mechanics of the solenoid.

We tested Williams 11630 coils for the flipper finger units. They contain a stroke winding with thick wire and a hold winding with thin wire. Both coils are raised when actuation of the flipper finger key. You need 2 outputs on the I/O board for this type of coil. The stroke (high intensity activation for short time) part (70V) was created at the first exit with a duration of 100 ms, then the output is reset. The part that keeps the flippers up (also 70V) at the second exit is operated permanently from the activation of the switch until the flipper finger switch is released.

For playfield solenoids mainly Williams AF 23-800 coils were used for various other components. These have only a winding and need only an output on the PINterface I/O board.
More details about the duration of activation and switching delay of coils are covered in the chapter on the rules of the game.

Lamps are operated simply with a lower voltage (5-12V) compared to the solenoids but the principle is the same. However, several lamps that should be activated or deactivated together can be combined to one output.

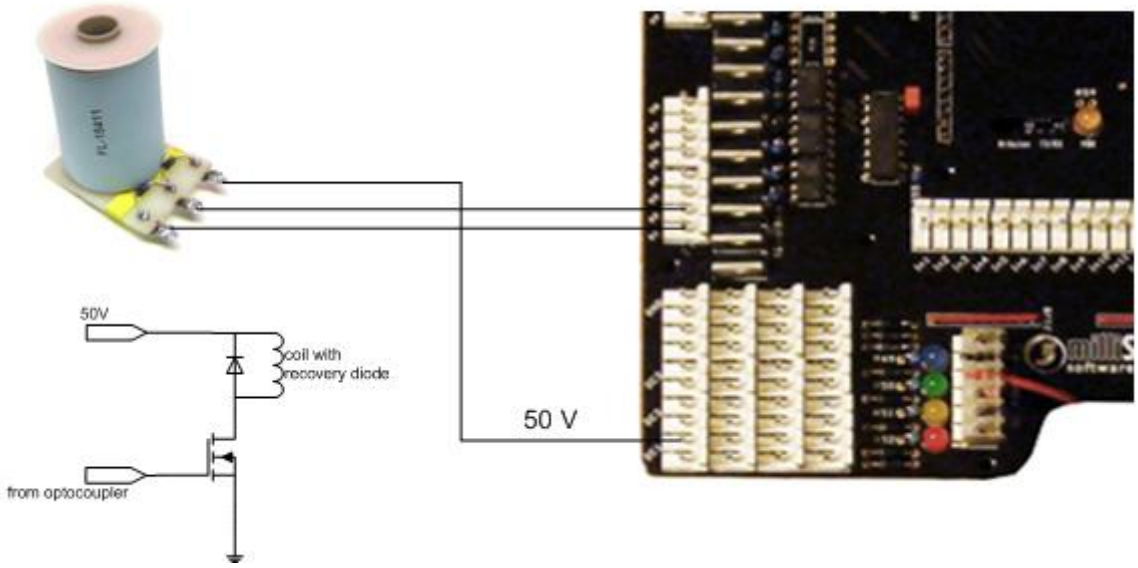
Connection of power supply to the PINterface I/O board

The different voltages for the PINterface board can be connected to J3 jumper pins 1-4 with different voltages and pins 5 and 6 are ground connections.
The voltages are spread over the diode D1-D11 the connectors J4 J13, J14, J15.
Pin 1 of J3 distributed on pin 1 and 2 of J4, J13, J14 and J15 pin 2 of J3 distributed on pin 3 and 4 of J4, J13, J14 and J15 pin 3 of J3 distributed on pin 5 and 6 of J4, J13, J14 and J15 pin 4 of J3 distributed on pin 7 and 8 of J4, J13, J14 and J15 pin 5 and 6 of J3 distributed over pin 9 and 10 of the J4 and J13, J14, J15

Wiring of playfield components

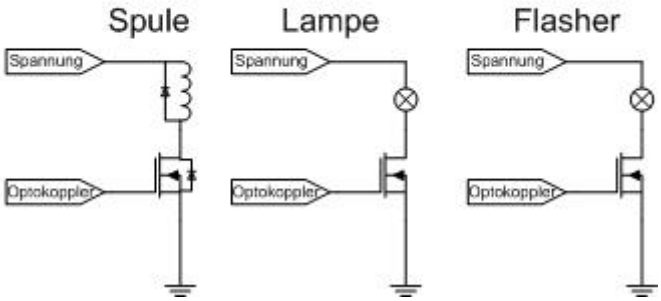
Solenoids and flipper finger can be connected to the PINterface interface board in

the following way:



Schematic of wiring of the solenoids together with the electrical schematic

The connection can be done in the same way but using only one output of the I/O board for other outputs (normal solenoids, lamps, flasher):



Connecting switches to the input area of the PINterface I/O board can be done in the following way:

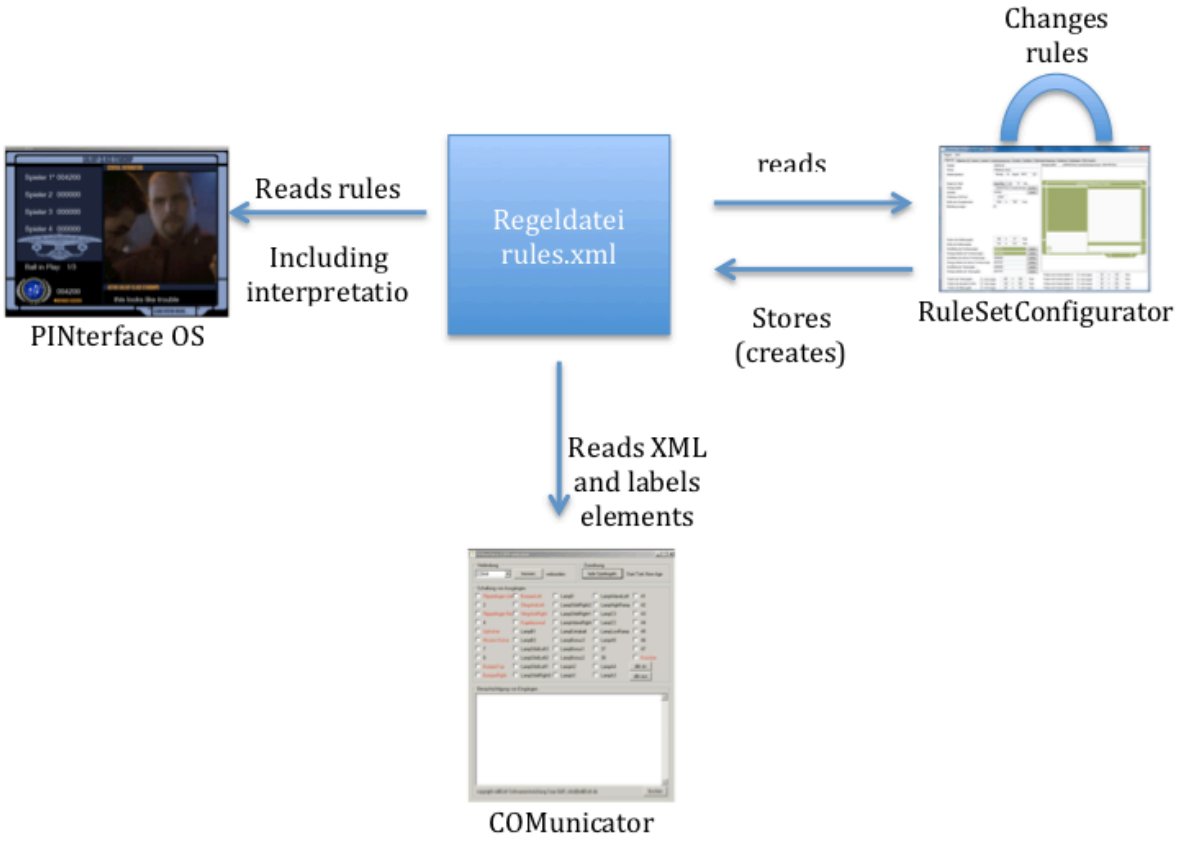
Ensure that you do not apply more than 5V to the input pins because damage can be done to the microcontroler otherwise!

Software

Software overview

The system includes various programs to test electronics, create the game rules and to interprete these rules during the playing process itself.

This software can be used on a PC that has installed the Microsoft.NET framework in version 2.0 or better. If you haven't installed it yet you can download it for free as a Windows system update the Microsoft home page (http://www.Microsoft.com/downloads/details.aspx?displaylang=en & familyid = 0856eacb-4362-4b0d-8edd-aab15c5e04f5).



The PINterface OS software runs as pinball operating system on the PC and interprets the rules created during the operation (playing) of the pinball mashine.

For the definition of the game rules the program RuleSetConfigurator is included in the software package, which provides a graphical rule set editor. With this software, you can customize the included demo game and create your own game for your device.

With the PINterface COMunicator, you have the possibility to activate individual outputs for tests and read input states. This way you can test your cabling and solenoids without having to create extra rules for these tests. This tool supports you

during the development of your custom pinball and can be used best for troubleshooting of wiring problems.

The software PINterfaceUpdater is part of every new software releases and updates the firmware of the microcontroller in the PINterface interface input / output Board if necessary.

The following chapters describe the individual software programs in more detail.

Pinball operating system (PINterface OS)

The Pinball operating system provides the actual game playing operations and interprets the XML rule set file that has been created manually or using the graphical editor „RuleSetConfigurator".

Game output

The sound output (effects, background music) are done using the default sound output of the PC and video on the TFT screen. The software communicates with the PINterface interface hardware (board with the microcontroller) using the USB port of the PC.

The program works even without a connected hardware, so the actual game with sound and video output and transitions of the game rules can be tested without the actual hardware.

Example of an output of a custom game

The program processes switch actuations and executes rules of the game. Furthermore, it controls the use of a credit (coin) based game (if enabled) and shows only the output window (in full-screen mode) to a pinball player during normal operations.

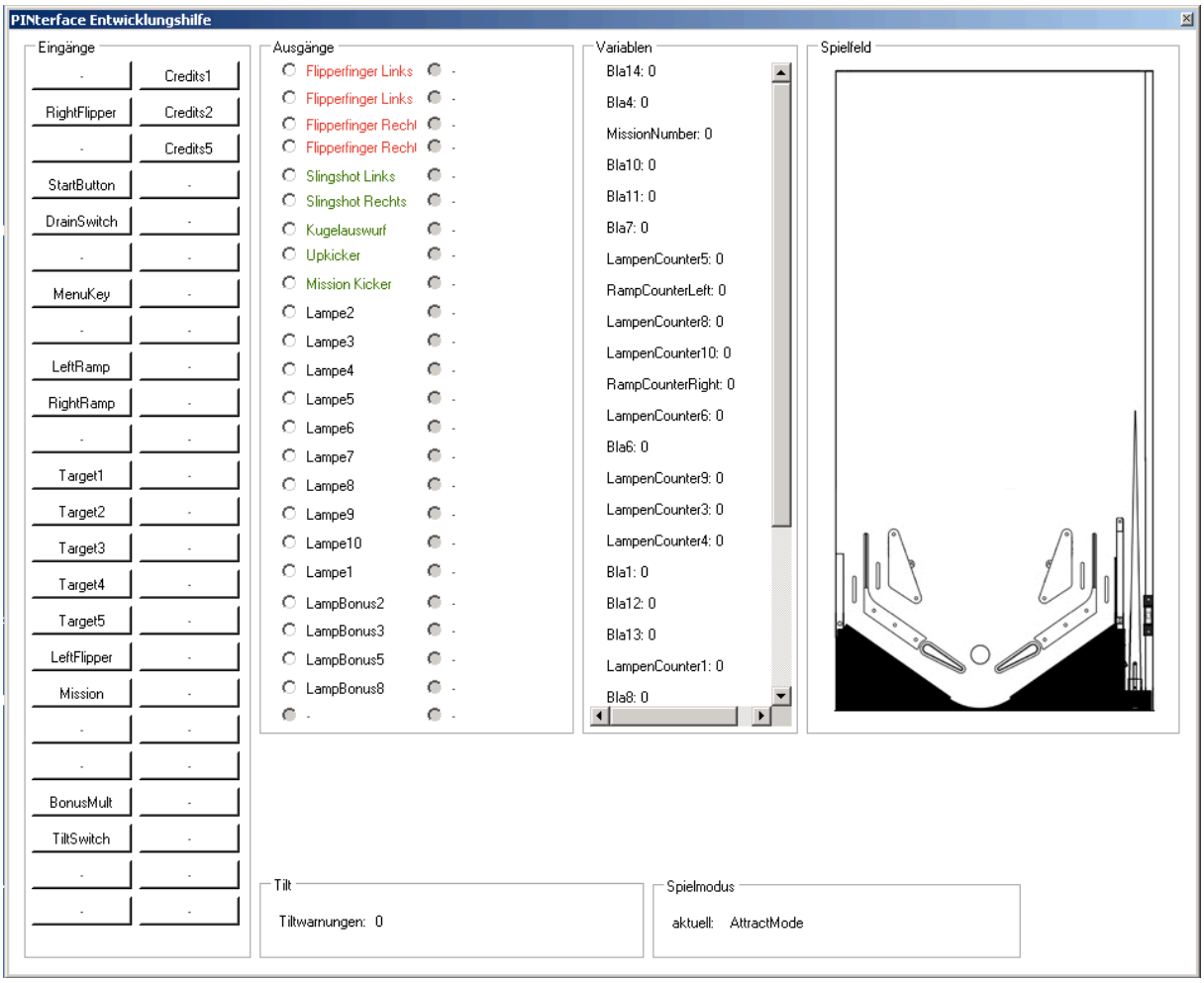
The graphic output on the display is completely parameterised and allows the output of points for up to 4 players, the output of the points of the current player, the image or video output and the output of additional text for single events. The window size and position of elements (text and video) can be customized. Items can be hidden completely if their output is not desired.

You can design the graphical output completely your way and you can even include high definition videos. The window size can be chosen freely, and thus it is possible to use small TFT screens (resolution 800 x 600-1024 x 768) or large wide screen displays (e.g. 1920 x 1080 pixel resolution). The output power is limited only by the speed of the installed computer and graphics adapter.

Development helper window

You can enable a development window in the rules of the game (using the game rule editor „RuleSetConfigurator") and this window appears then next to the standard game output. This development guide gives you the possibility to trigger the defined switches with the mouse when your hardware layout of the game is not yet finished, but the rules can be tested this way. The defined outputs are also displayed next to the buttons, variable names and their values are shown and the current game mode will be displayed below (see screenshot). For timed game modes, the remaining time is also displayed if a timed game mode is active.

The buttons on the left show the name of the switch according to the rules of the game and if you mouse over the elements for switches, outputs, variables and game mode the comment of the corresponding item is shown as a "ToolTip" (small text bubble) if a comment has been defined for the item in the rules.



Screenshot of the development helper window

The outputs are highlighted (as seen already in the „COMunicator"), you can distinguish if they are a coil (red) or a protected coil (green). For more details about this coloring and protection please refer to the coil definition section in the „RuleSetConfigurator" software.

Furthermore the tilt counter and the time until it is reduced is shown at the bottom to indicate the Tilt state.

In-Game-Menu

The diagnostic menu is intended for quick tests during the normal operation of the pinball mashine. You can use this menu during playing if a playfield element failure occurs. This menu is similar to the in game diagnostics of common pinball mashines where you invoke this tests using the buttons in the coin door and navigate the menus with the flipper buttons and the start button. The keys to open and operate the diagnostic menu are defined in the XML rule set file.

The in-game-menu after invoking the menu button (main selection)

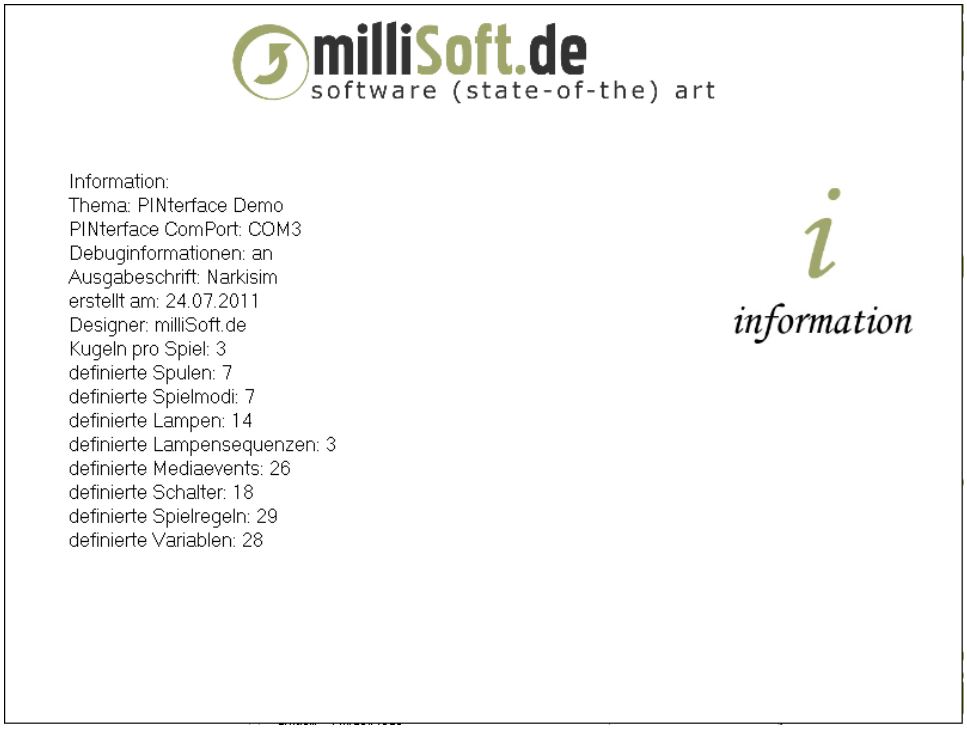
The menu is opened by the menu-enter button and displays the main screen with the choice of the submenus:

- General information
- Solenoid test menu
- Switch Diagnostics dialog
- Lamp test menu

The keys for Menukey left and right (can be defined in the XML rule set file) are used to select the individual menu items. The selection is made by using the switch that has been assigned menu key „select". It is recommended the assign the Menu switch to one of the switches that only is available after opening the cooin door and using the flipper buttons on the side of the mashine for the functions "Left" and "Right" to move through the menus. For the "select" function the Start button on the front of the Pinball is a good choise.

General informationen dialog

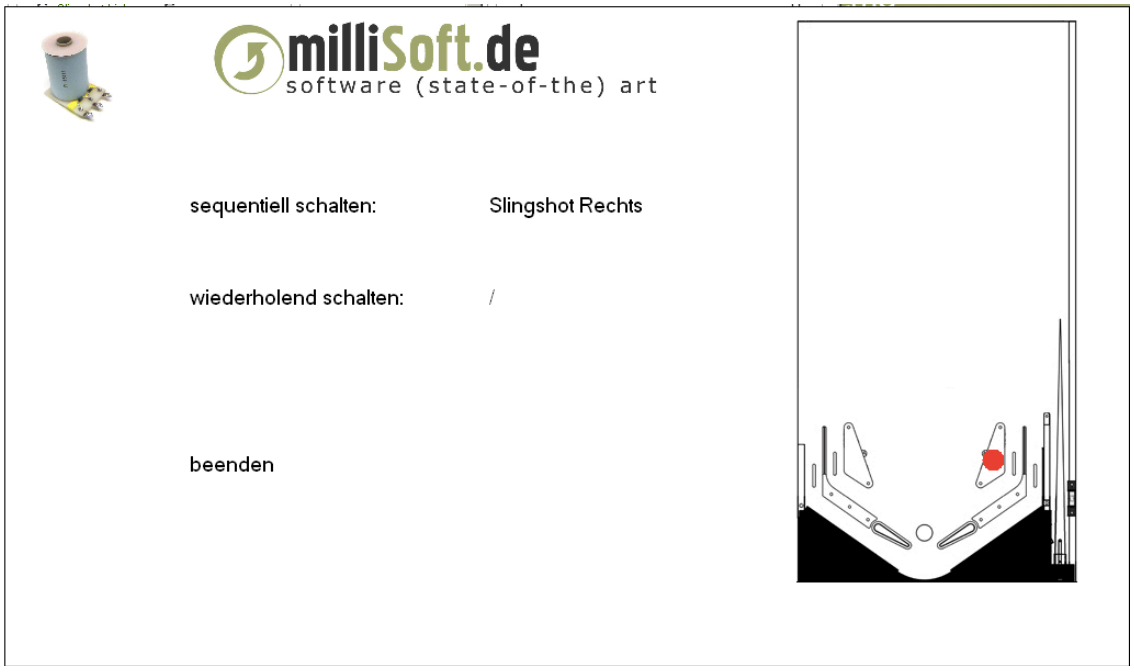
This submenu item displays general information about the custom pinball mashine that have been defined in the game rule set definitions in the XML file.



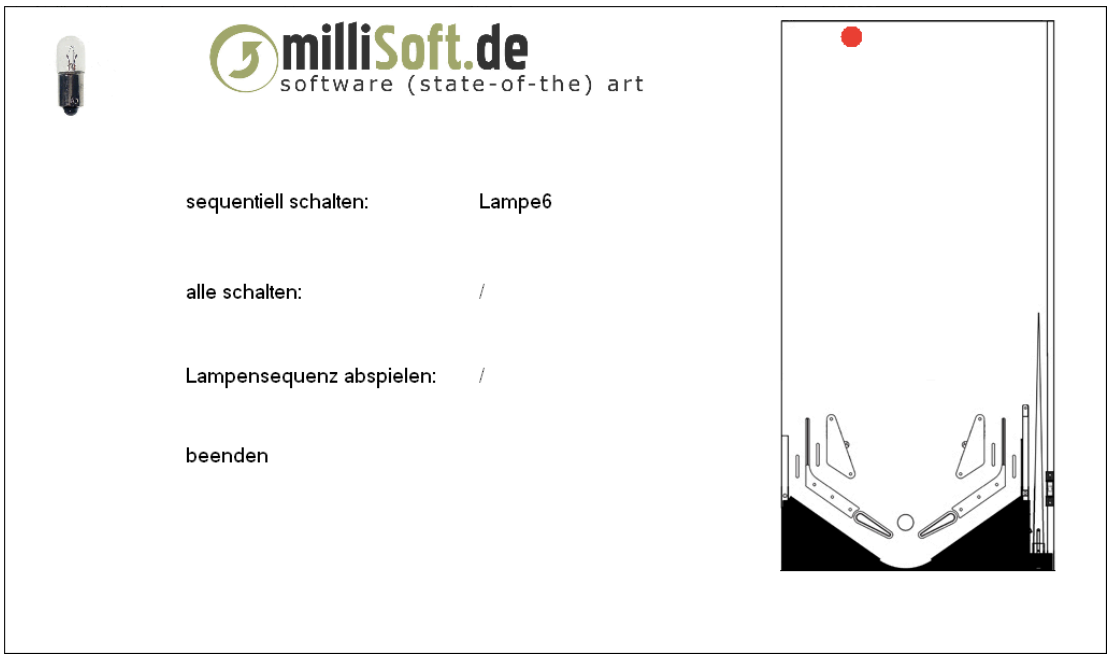
The in-game-menu showing information of this pinball mashine

Solenoid test menu

In this menu, all defined solenoids can be checked sequentially or in the endurance test. During the endurance test each coil is fired for a few milliseconds repeating this procedure every second.



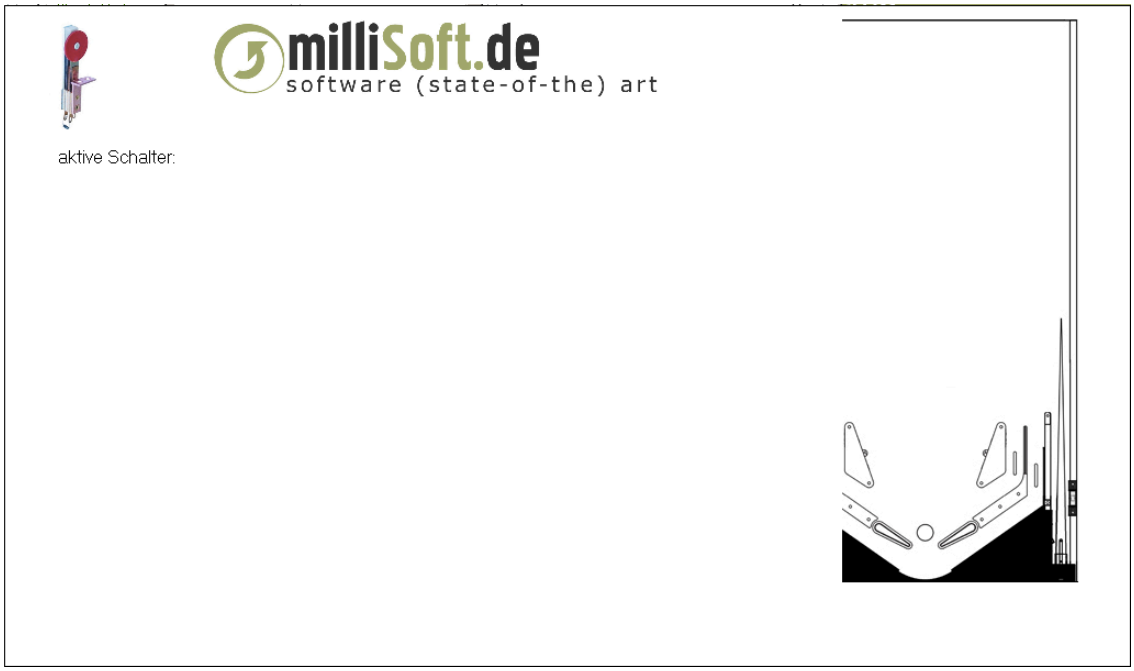
Lamp test menu



The lamp test menu

In this menu, all defined lamps can be checked individually sequentially or you can choose to toggle all lamps at once. Furthermore it is possible to play one of the lamp sequences that are defined in the rule set file for this pinball mashine.

Switch diagnostics menu



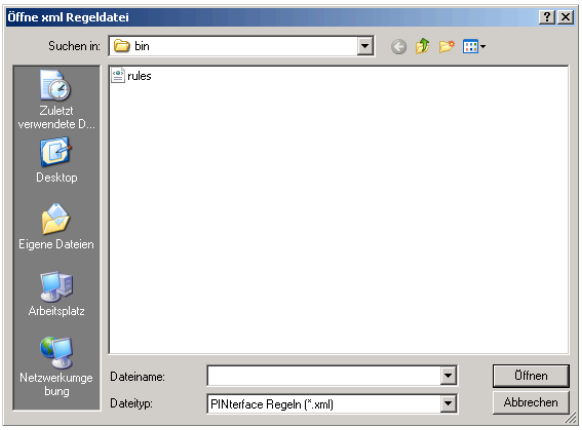
The lamp test menu

In this menu, the switches that are activated are shown based on their name, input number on the controller board and position on the playfield. This way the system can visualize the closed switches while working on the playfield hardware.

Game Rule Editor „RuleSetConfigurator“

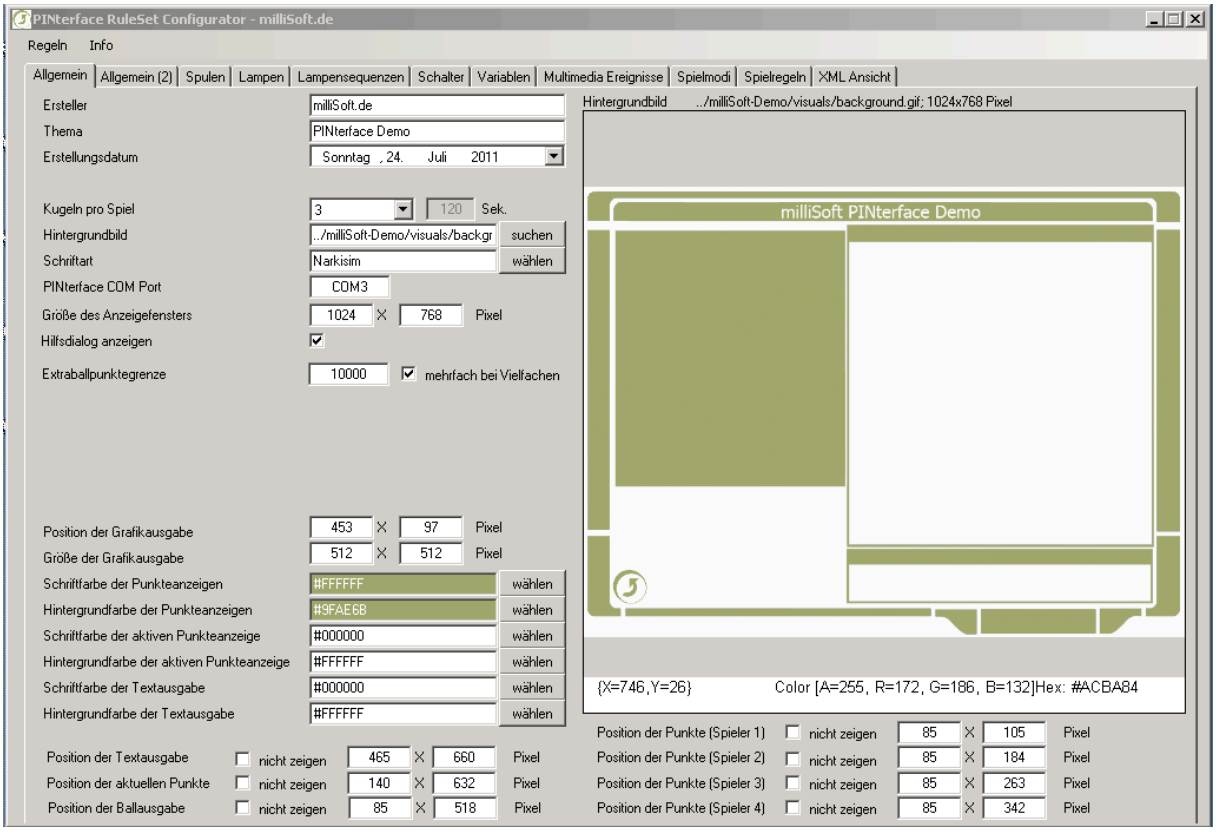
The graphical editor allows the creation and modification of rules graphically without working with an XML file in an text editor. The program loads the rules from a XML file, allows the modification of individual settings and game rules and stores the result as XML file fulfilling the PINterface syntax.

At the start of the program you have to point the software to a XML rule set file to edit:



In the following dialog there are different tabs to change the general settings, solenoid definitions, lamp definitions, light sequences, switch definitions, game modes, game rules, multi media events and variable definitions. In addition, the XML file can be shown here before saving the text file.
After doing changes in this program you can save once again the XML rules file under the name "rules.xml" in the program folder of PINterface to use the changed rules in the game itself.

General Settings (Part 1)



In the first tab, you can change the settings for connection to the PINterface interface controller and some graphical output parameters. Furthermore you can also define details about the pinball manufacturer (subject, name of the Builder, completion date). For the graphical elements, you can configure the following parameters:

„Creator" defines the name of the Builder: Here you can enter your name and it will be displayed in the menus of pinball

The „theme" defines the name of your pinball theme. This appears as well as the name of the Builder in the menus.

The date of your configuration can be specified to distinguish different versions during the development. This date will appear in menus as well.

Typically, the number of balls per game are "3" or "5", however, you can also enable a timed game which requires a fixed duration per player. If the player drains a ball, it will be put back int play directly until the time is up. The timed game can be specified by inserting the duration in seconds. All values above "12" are not interpreted as a number of balls, but as playing time in seconds.

The output font can be configured with size and location of the various items (if no output is wanted individual items can be hidden completely).
The font can be selected from the fonts installed in the system.

The name of the font is used for the font view and the font size can be changed. The font must be installed in the Windows folder and can be used if it also can be used in a program like Microsoft Word. TrueType fonts are supported.

The COM port that the PINterface interface is connected to can be selected. The microcontroller is detected by the Windows PC as a virtual COM port. This can be seen in the Device Manager, or tested using the „COMunicator".

The background image of the output is displayed on the right side and can be used to position the other output items.
To get positions and colors of the background image, you can drag the mouse over the background image, read pixel positions at the cursor position and underlying color values at these coordinates. These values can then be entered in the position properties of each single configuration object.
The output colors are to be configured as hexadecimal information, where the first 2 digits represent the red component, followed by the green component in the middle two digits and the last two digits specify the blue component.
To disable an individual items, you can enable the checkbox next to the item name. This item does not appear then during the game.

"The score display font color" is the color of the text of the score display of all 4 players for the overview.
"The score display background color" defines the background color of the score display of all 4 players.
"The active score display font color" is the color of the text for the score display of the current (active) player (including bonus points).
"Background color of the active score display" is the background color of the score display of the current (active) player (including bonus points).
"Font color of text" is the color of the text output of multimedia events.
"Background color of text" is the background color of the text output of multimedia events.

"Position of the current score" defines the output location of the current score of the current player.

"Position of the points (player one)" defines permanent position of the output of the first player's score.
"Position of the points (player two)" defines permanent position of the output of the second player's score.
"Position of the points (players three)" defines permanent position of the output of third player's score.
"Position of the points (players four)" defines permanent position of the output of the fourth player's score.

"Location of text" defines the print position of text output (of the media event).
"Position of the Ball output" defines the position of the text output of the remaining balls or playing time.

The option "don't show" next to each item stores the position "-1;-1" so that the element in the PINterfaceOS is not shown.

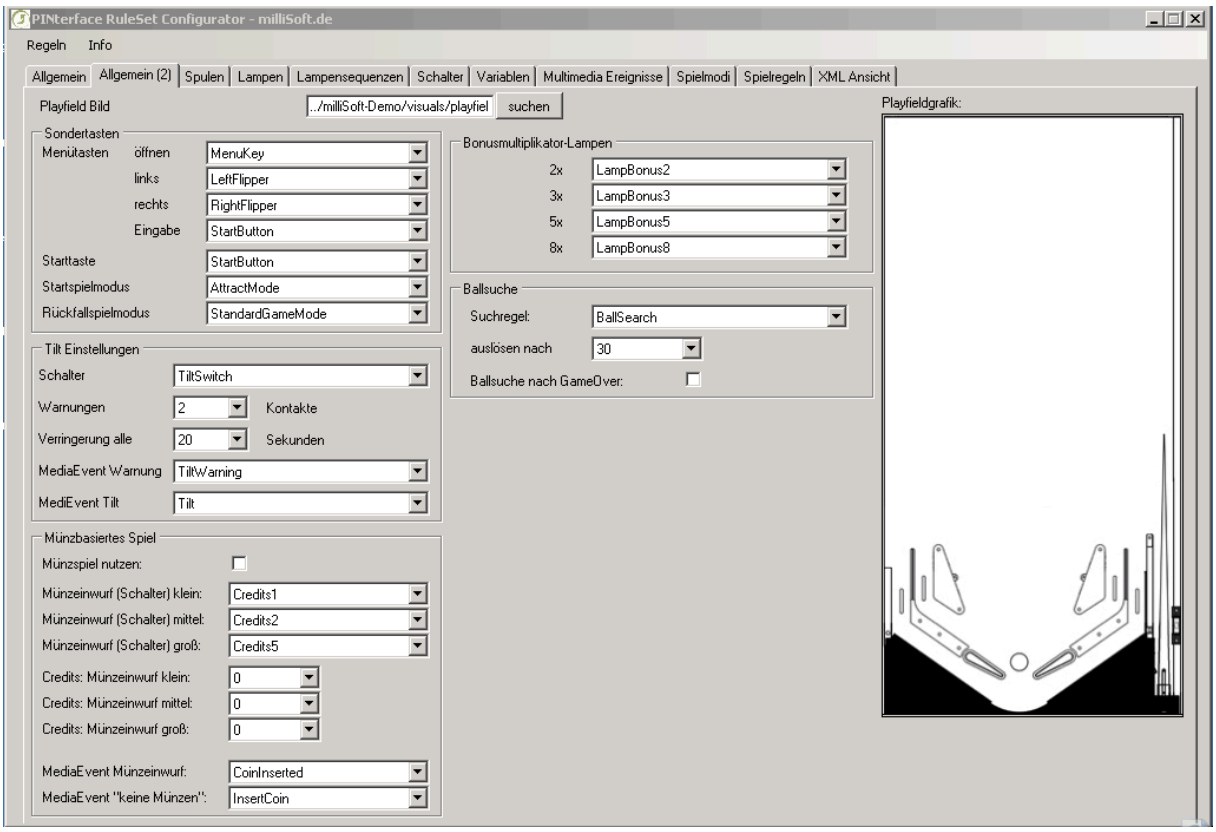
Background image and the size of the output window
The background image must be a graphics file with the size of the output window used on the TFT screen.
The size of the output window is specified in pixels, and should also match the screen to the size of the output of the TFT.

"Position of graphics output" defines the position and size of the visual output (graphics and videos). This size should match the size of the videos and images.

"Size of graphics output" defines the output size of the total PINterface OS window. Configuring the screen resolution to the TFT resolution and deactivating the debugging output (see below) will show the PINterface OS in full-screen mode, as it should be used for regular playing.

"Display help dialog" parameterizes the output of additional Information on the developer helder window, which allows the activation of individual switches with the mouse. The additional test window shows the status of the individual outputs (coils, lamps) also the current values of variables and the current game mode.

General Settings (Part 2)



The second part of the General settings includes the definition of the playfiled graphics, which can be used for documentation of single playfield part locations and helps in the game to identify problems using the in game menu.

The right pane of the tab is used to issue of this playfield image, which can be specified in the text box and the search button next to it. This graphic should be 256 x 512 pixels (width x height) in size. This graphic is used in the other tabs to highlight position of the game elements.

Keys to activate and control in the "in the game menu":
During the regular game you can open the in game menu by activating the switches defined here.

The switch to open the menu can be selected in the "menu button - open" element. This is the name of one of the defined switch from the "switches"-section. Another activation of this switch closes the in game menu again and returns to the normal game, when you press it in the main menu. If you are in a sub menu you will close this submenu and move on level up in the menu structure.

The switch to navigate left and right in the menus can be selected in the "menu button – left" and "menu button - right" elements. This is the name of one of the defined switch from the "switches"-section. Hereby, the selection can be changed in the menu.

The switch to select/highlight a menu item of is used in the game menus can be selected in the "menu button - input" element. This is the name of one of the defined switch from the "switches"-section.

For more information, see the chapter in-game menu.

The switch that can be used to add a new/another player, can be selected at the "Start button" element. This is the name of one of the defined switch from the "switches"-section. When this switch is activated and no game is active the first player will be added and activated. A new ball will be put onto the playfield by using the "startball" game rule. If one or more players are already enabled, pressing this button adds another (up to 4) player to the game. This only works as long as the 2nd ball has not yet begun.

"Extra ball point limit" is the score value that must be exceeded to gain an extra ball after the current one. The extension "multiple" specifies, whether there is only an extra ball at the specified point value or whether any multiple will provide more extra balls. If, for example, the point threshold to 1000000 is with "multiple times in multiple" enabled, so the player in addition to the extra ball when gets even more extra balls when 2000000, 3000000, 4000000 of 1000000 points and so on.

Button and behavior for the tilt mechanism:
To describe the behavior of the system for vibration protection (tilt), you can first define the switch that is connected to the Tilt switch at a PINterface interface input.

Dodging the pinball can be punished by reading this Tilt switch. The behavior of the PINterface of OS (operating system) is parameterized with the number of warnings and the Decreasetime. If the contact is closed more times than the warning count specifies it then terminates the current ball, no bonus is granted and the game continues with the next player or ball. Still, the Tilt warnings media event and the tilt GameOver media event can be displayed graphically referencing "warning" and "gameover".

The number of Tilt warnings to be issued before draning the current ball can be parameterized. To reduce the Tilt warnings the counter is decreased by 1 after the time that the parameter"reducing every X seconds" has passed.

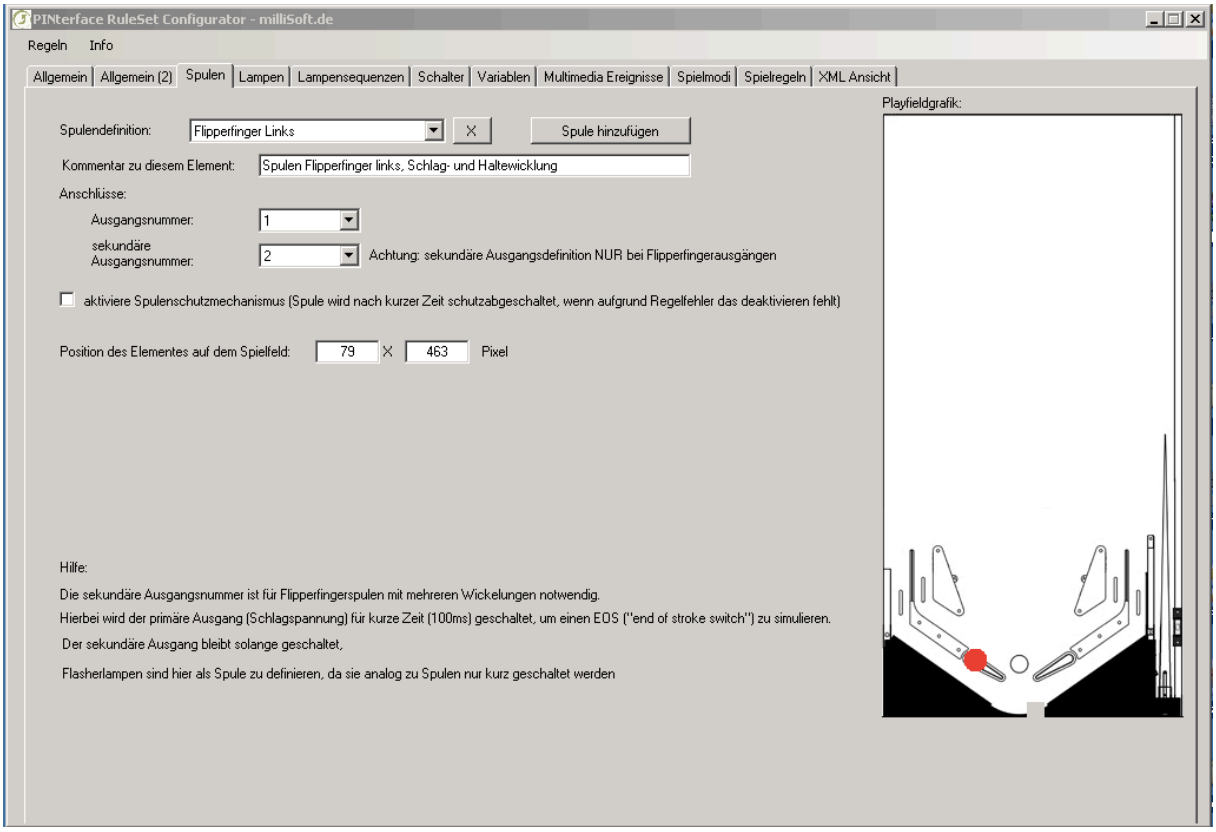
The „standard game mode" defines which game mode is selected initial after starting the software. Typically, this is the AttractMode, which allows no activation of the flipper fingers and shows nice lamp sequence effects to attract new players.

The "Fallback mode" will be activated after timed game modes (e.g. missions) are over. Typically, this is a game mode that represents a normal game. This is used, for example, when the game from an official bonus mode returns, in which all the points were counted several times.

The lamps which display a reached bonus multiplier are also defined here. There are configurations for the bonus multiplier from 2 x, 3 x, 5 x to 8 x.

A ball search rule can be defined to activate selected solenoids to search for dead balls on the playfield to reduce the number of mission balls after some time without playfield action or after a ball drained and some balls are still missing.

Solenoid definition



The outputs of the PINterface interface board can be used for the solenoids, flasher lamps and conventional lamps on the playfield (inserts).

For the definition of the solenoids, the third tab defines the coils with a name and specifying the output port ("O"). The name of the coil must be unique among all defined elements. Existing coil definitions can be removed with the "X" button next to the combobox. To add a new solenoid you can use the "add new solenoid" button.

For each solenoid an output must be defined, to which the element is connected on the I/O board. These are marked throughout the system with "O", so that the output number 17 is identified as "O17".

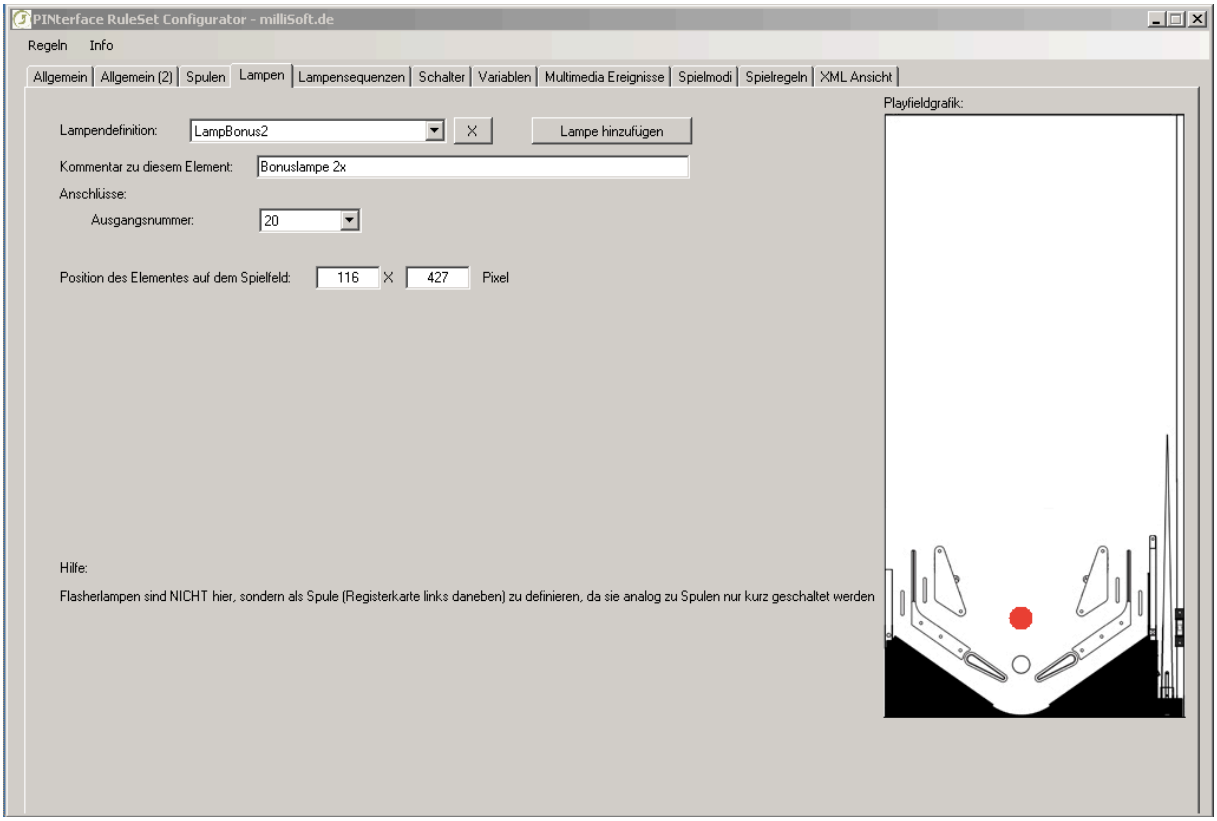
In addition to the standard output number, special solenoids like the flipper fingers can have secondary outputs. These solenoids have a primary output for the high intensity stroke of the flipper and a secondary output is connected to another part of the solenoid that is responsible for holding the flippers up (lower intensity). Later in the game play the primary output is enabled only a few milliseconds and will switch off automatically, while the secondary output remain enabled until the triggering button is released again.

The combobox to select an output port for a element shows all of the outputs with a note visualizing if the respective outputs are still available or already allocated to other solenoids or lamps.

For each output element a shut-off protection can be activated if the element uses only one output port (only primary connection and no secondary output port). The system then automatically disables an output on the microcontroller after half a second to protect the solenoid on the playing field and the output transistor on the I/O board. Otherwise a stuck output (caused by a software crash after rule errors) results in a dead transistor or solenoid because the output does not hold off the enormous current of solenoid circuits.
In the XML rules file the outputs are identified as "protected" and shown in green font in the „COMunicator" and „PINterfaceOS" development helper dialog while unprotected coils are presented in red.

On the right side of the tab the previously specified playfield image is shown and it is possible to click in the image with the mouse to define the position of the element on the playfield. This information is then used in the in game menu to highlight this element if activated.

Lamp definitions



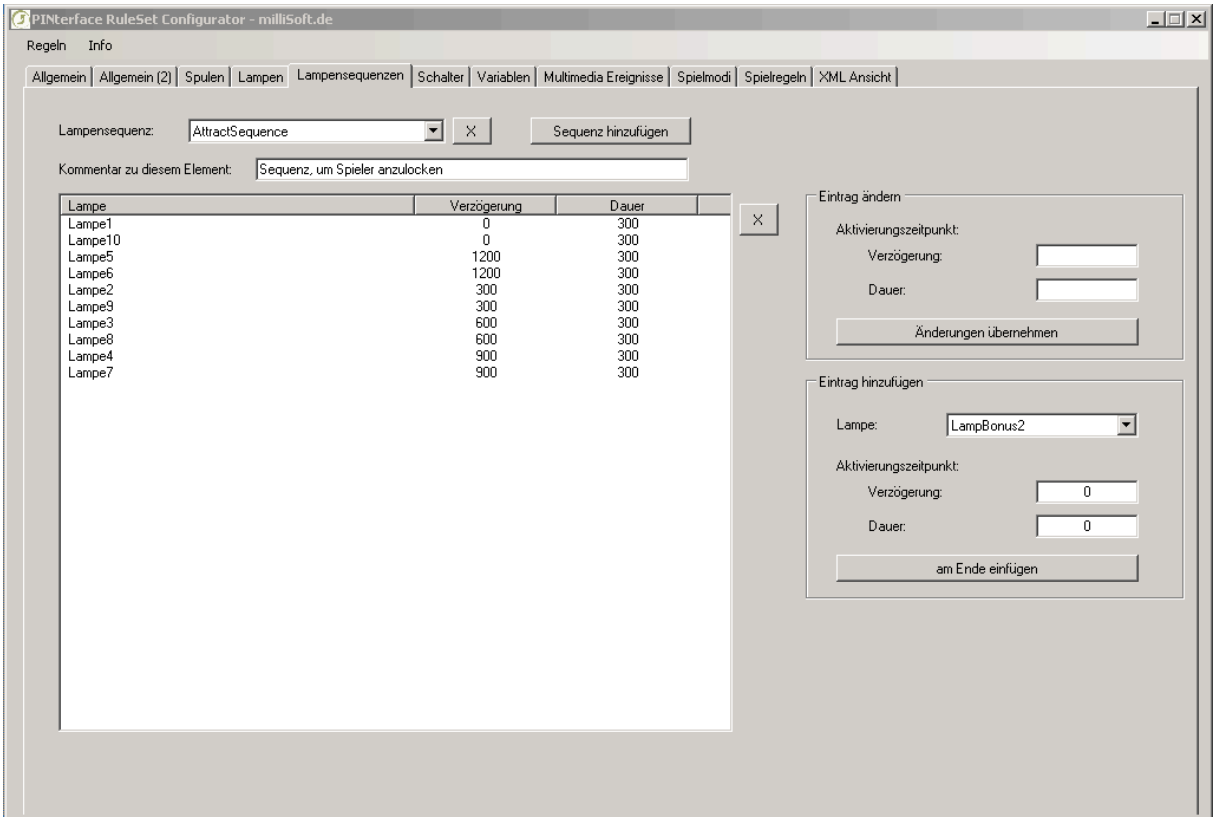
The lamp definitions can be done the same way solenoids are defined by specifying the number of I/O board output.
The page "Lamps" configures the "lamps" part of the XML game rule file.

On the right side of the tab the previously specified playfield image is shown and it is possible to click in the image with the mouse to define the position of the element on the playfield. This information is then used in the in game menu to highlight this

element if activated.

Also please note that the names of the lamp definitions must be unique!

Definition of lamp sequences



The light sequences can be used to create visual animations on the playfield. A lamp sequence can be defined as a background to a game mode (i.e. special animations in "attract" mode or emphasis of ramps during a mission by flashing the lights at the ramp entrance).

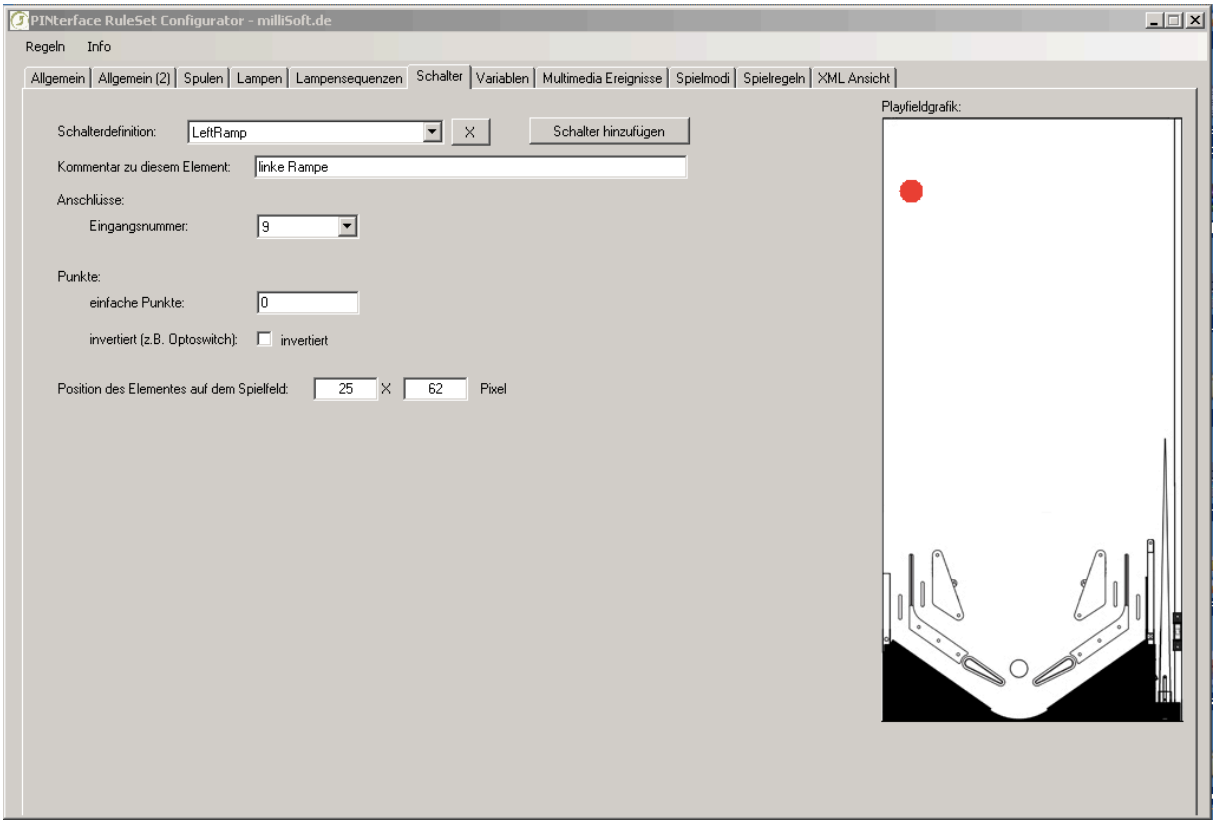
The definition of light sequences is based on single sequence elements containing information about each lamp.

Light sequences are composed of a list of names of lamp with delays (switch on delay) and illumination time (how long it will be switched on), which are executed sequentially. The times are given in milliseconds (1 second = 1000 milliseconds). For each entry in a lamp sequence definition, the lamp ("duration") must be specified the name of the lamp ("name"), the delay relative to the start of the lamp sequence ("delay") and the duration of the connection.

After you select a light sequence, the list view shows each item of the sequence by name of the lamp, switch-on time and duration below. A selected entry in the

sequence can be removed with the "X" button next to the list or changed in its details. Single lamps can be added with additional on-time and duration defined. The start time is always relative to the periodic start of the light sequence.
The "Light sequences" page configures the "Lampsequences" part of the rules of the XML game rule file.

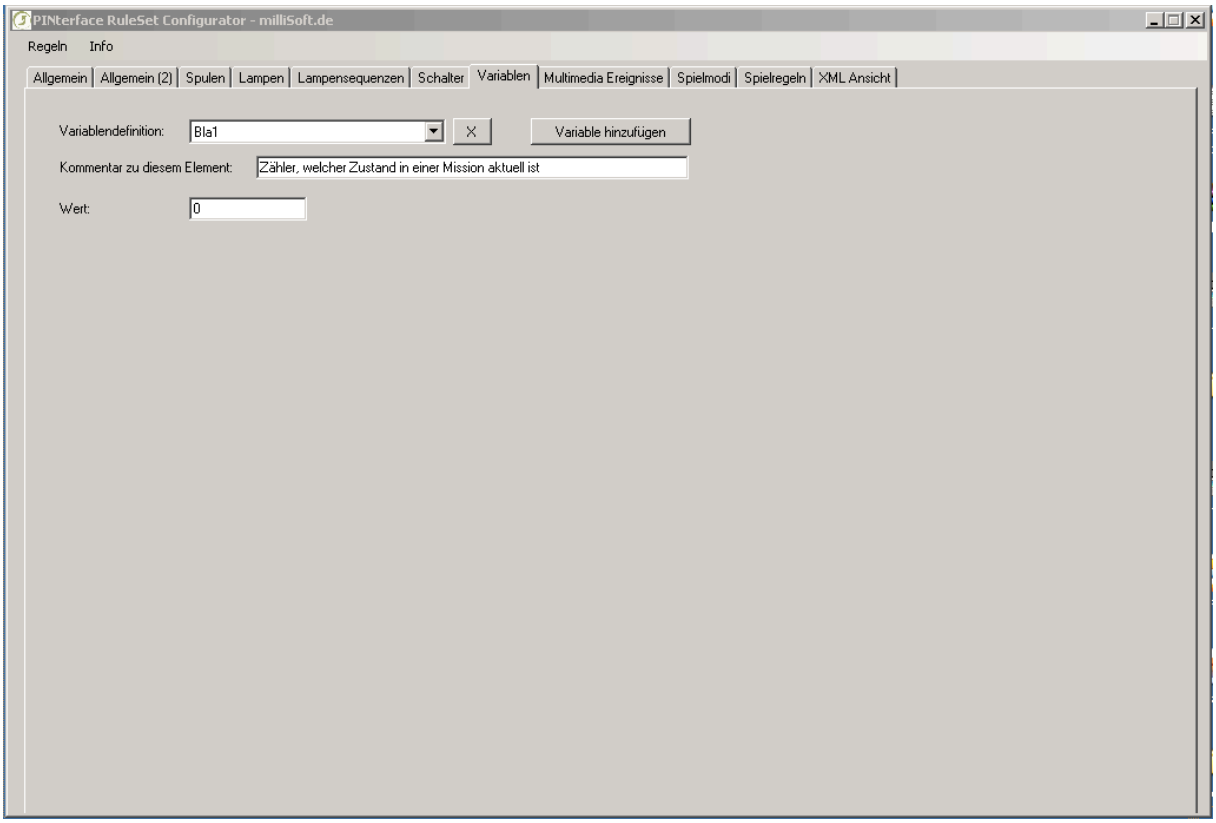
Switch definition



The switch definition contains the score that the player is granted when activating this switch (e.g. hitting the targets) next to the name and the input number.
Switches with optical elements (e.g. OPTOS) instead of mechanical contacts must be defined "inverted" using the checkbox because these switches electronically open when logically activated instead of being closed.
The page "Switches" configures the "HardwareSwitch" part of the game rule XML file.

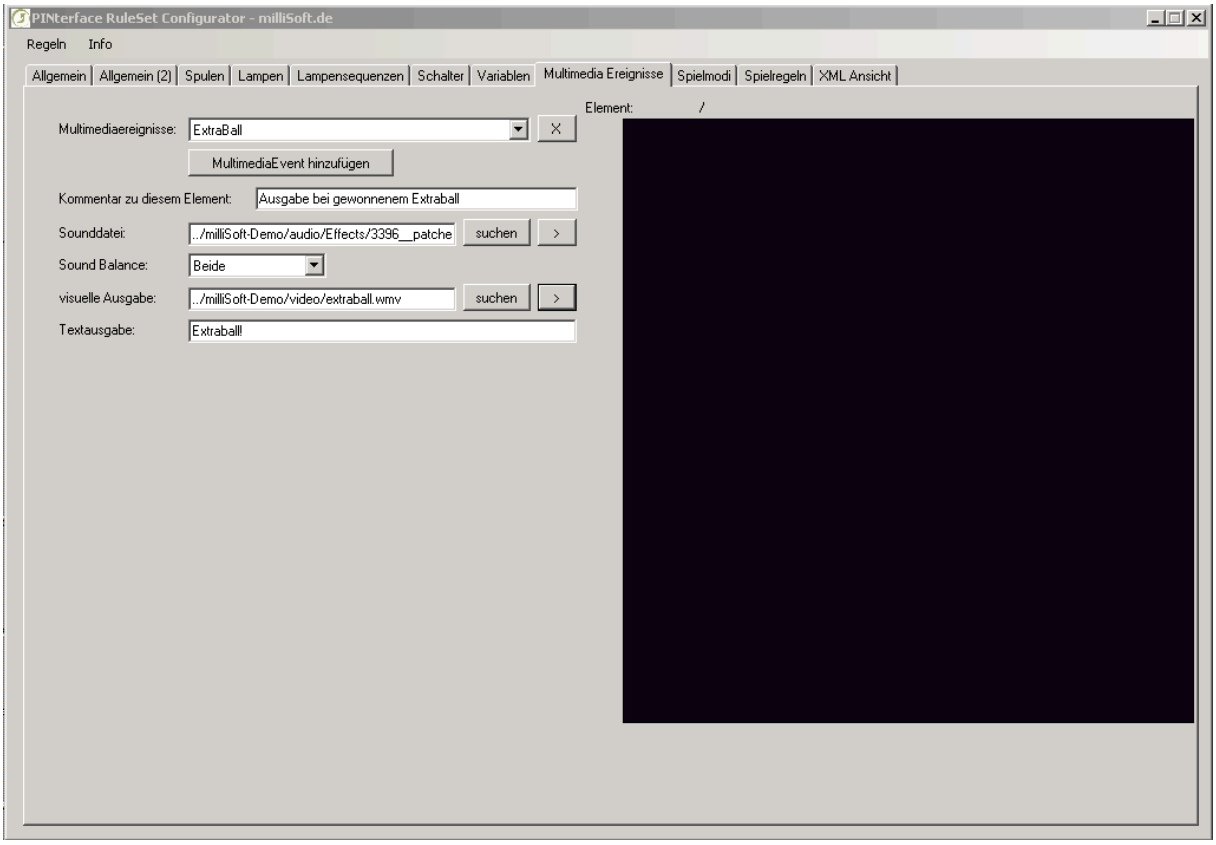
On the right side of the tab the previously specified playfield image is shown and it is possible to click in the image with the mouse to define the position of the element on the playfield. This information is then used in the in game menu to highlight this element if activated.

Variables definition



Variables can be used to implement counters (i.e. how many times a target or ramp is hit) and game storage containers. Each Variable has a name, comment and initial value and can be tested and changed in the game rules.

Multimedia output of sounds, graphics and videos



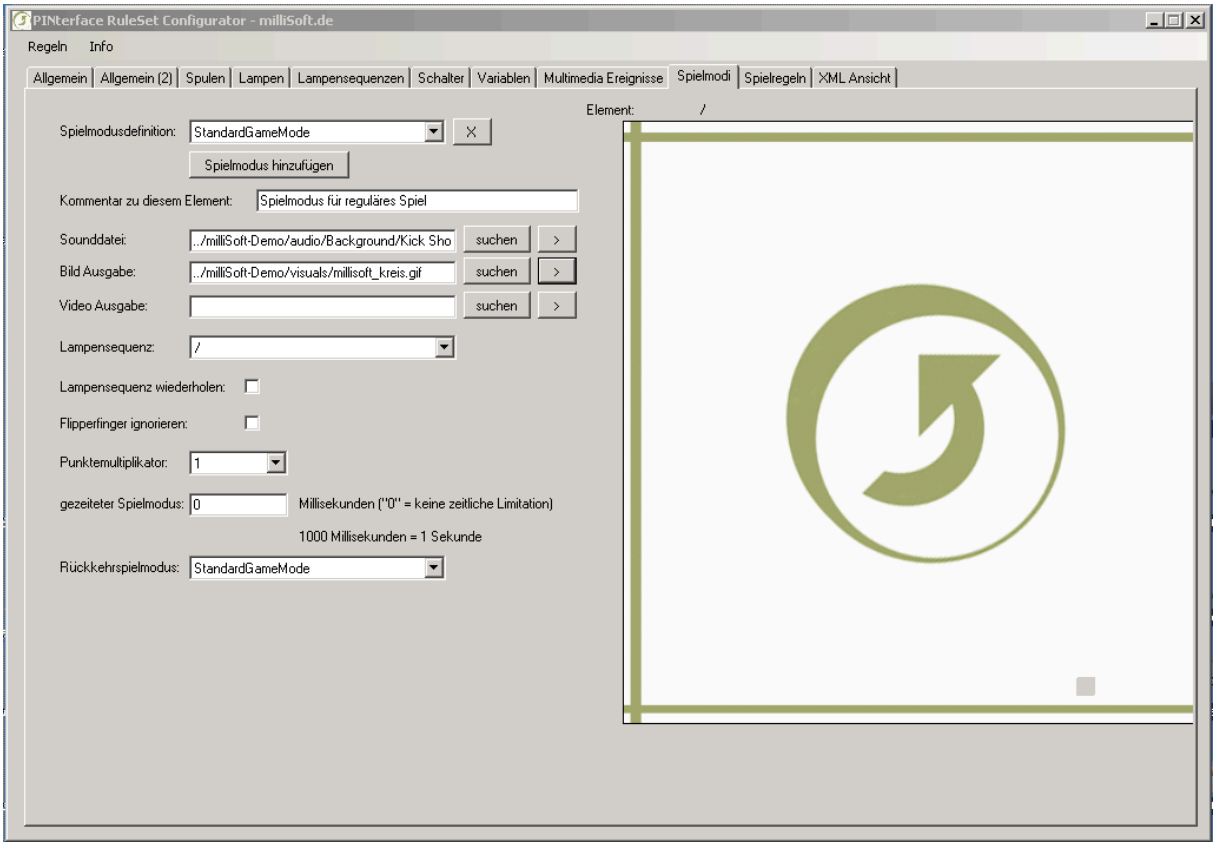
The multimedia events define sounds, images, videos and text output that can by played or shown in the PINterface OS. The sound files can be played in stereo on both speakers (left speaker, right speaker, both speakers). The Visual output specifies a picture or video that will be played by the system depending on the type of file.
The "Media events" page configures the "MediaEvents" section of the rules of the XML game rule file.

The output of music or sound effects can be done using files in Wave or MP3 format. Depending on the theme of your pinball you may benefit from sound files, that were used in movies, audio CDs, or computer games. Topics which a rich merchandising are usually good templates to create a pinball mashines.

The playback of video sequences can be done using "media events". Game rules can output such events according to game rules i.e. after hitting a target or reaching a target score. The video sequences can be played using files in formats the PC can playback.
The optimal video resolution for playback in the PINterface program is defined by the graphical output size configured earlier in the rules file (see above).
Instead outputing videos you can also display simple graphics in JPG, BMP or GIF format; these appear for a short time on the TFT after the media event is shown. This is a simpler graphics output which can be used if videos creation is not yet

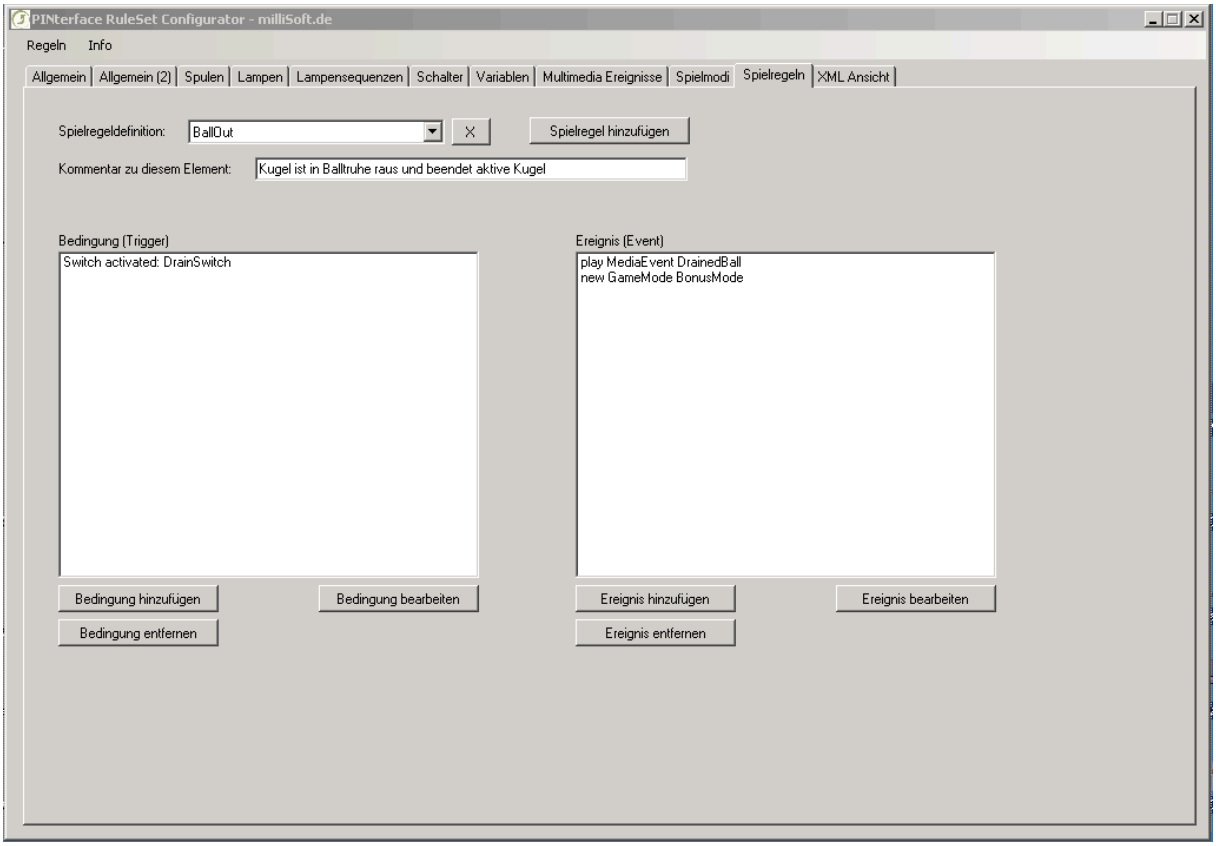
finished, or not possible or wished for your pinball creation.

Gamemodes of custom pinball



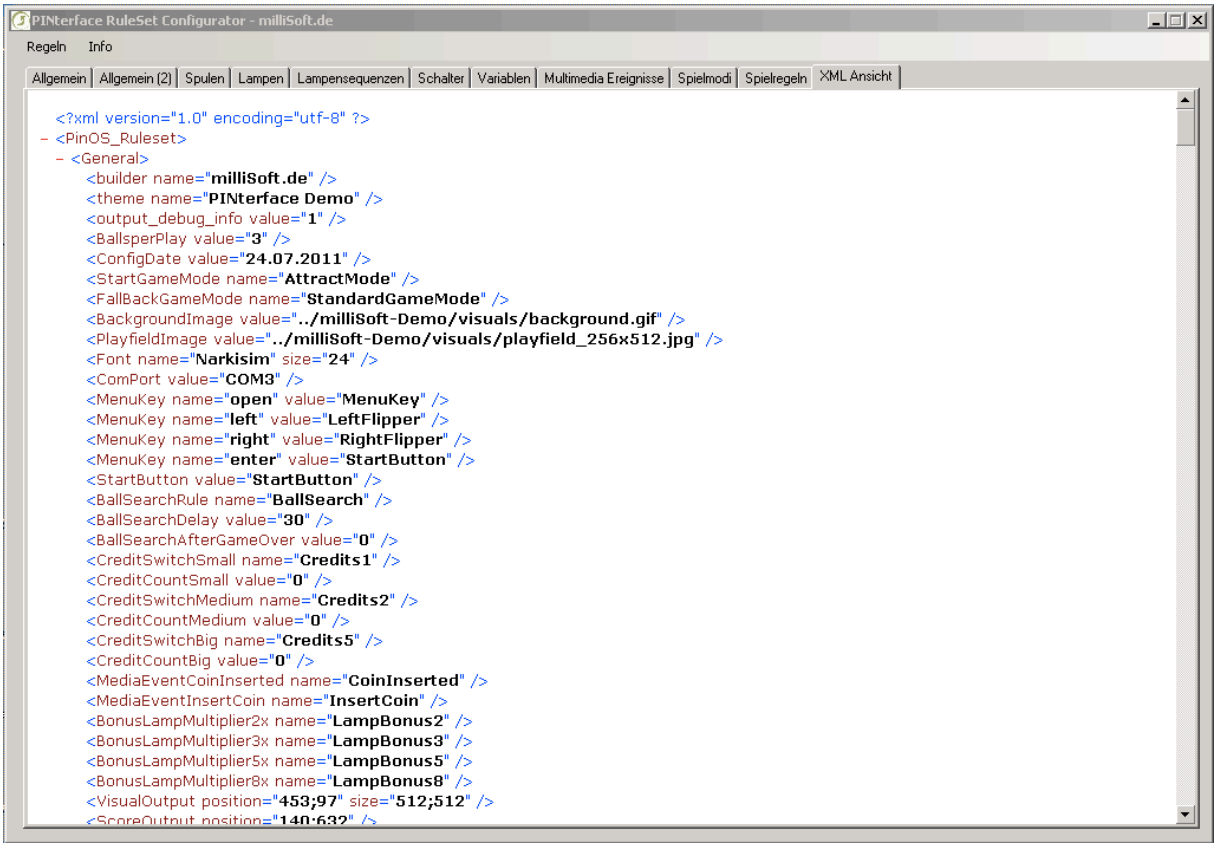
The tab "Game modes" configures the "game modes" part of the rules of the game. These gamemodes are the basis for different states of the pinball mashine like a standard game mode or a mission mode (i.e. granting more score for all targets). Each game mode can define a background music, image output (background image and video sequence that is played back after changing to this game mode), a score multiplier that is applied to all score added to the player score, a lamp sequence for playfield lamp action and a time if the game mode should only be used for a certain amount of time. After this time the game will change to the gamemode defined as fallback gamemode.

Gamerules for transitions from one state to another



The "Rules" tab configures the "rules of the game" part of the rules of the game. These game rules are the heart of every pinball definition because these define the transitions from one game mode to another or tell the system when to output which media event (graphical and sound output). The game rules can trigger hardware switches from the playfield, reached pont scores or fulfilled variable conditions. Each game rule defines triggers and events. Every time the system detect an action it tests if all triggers of a game rule are fulfilled. If all triggers are met then the events are applied. This way variable values can be changed, lamps or solenoids can be activated or media event can be output.

Output of game rules as XML text file for inspection



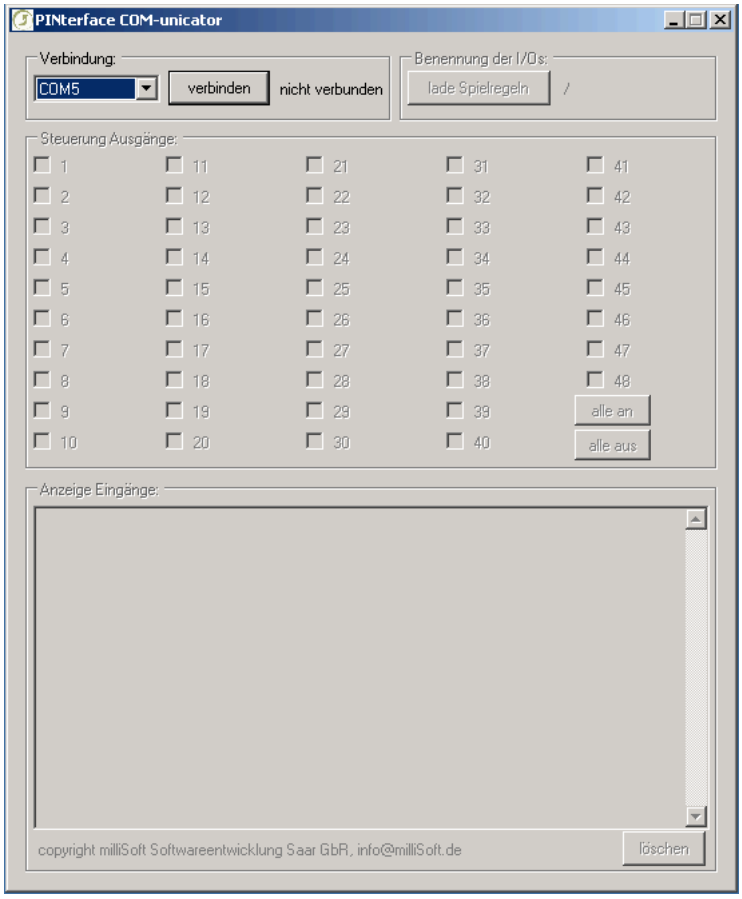
The output of the entire game rules as XML is a preview of the rule file that will be stored for the PINterfaceOS. Here you can verifiy once again how individual elements have been implemented in the XML text file. The XML display allows grouping entire segments (e.g. "General") to speed up navigating in the file.

Test software COMunicator

The „COMunicators" can test the inputs and outputs of the I/O board without having to apply a set of rules. This way defects in input and output wiring can be detected or single elements can be checked directly during the development of the system.

The PINterface software installation creates a shortcut to this program on the desktop.

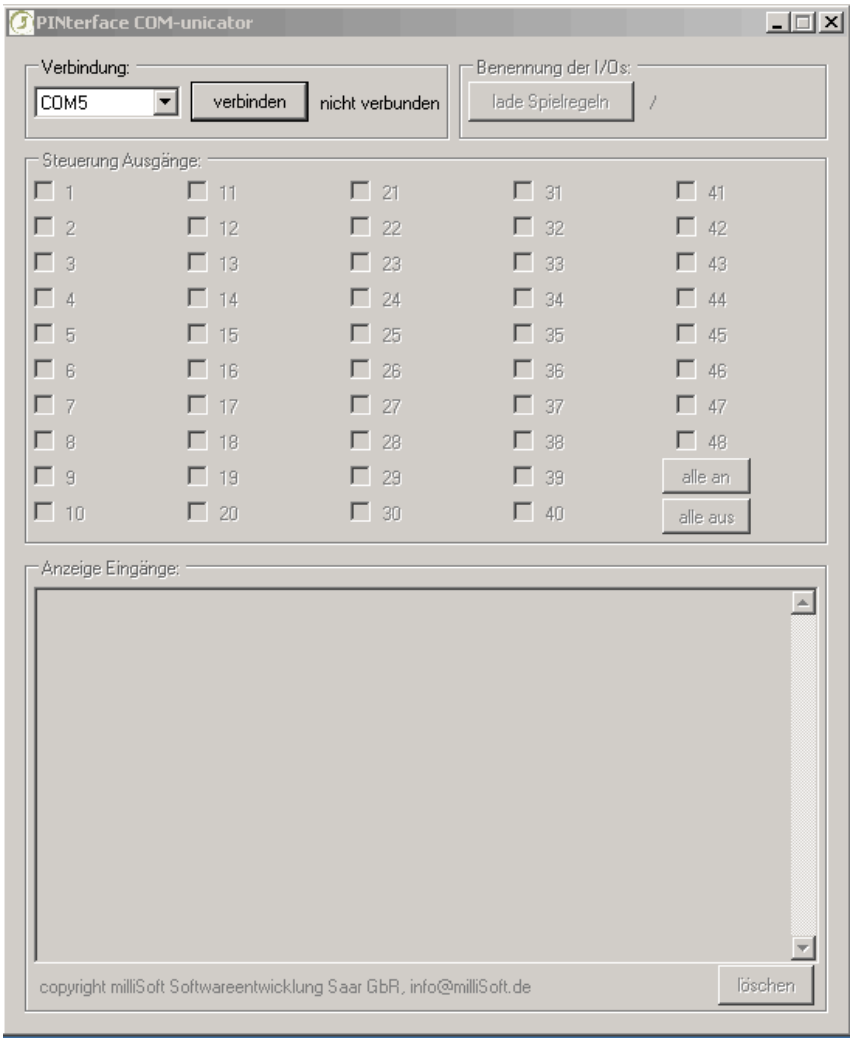
The program itself lists found COM ports on which a PINterface interface module can be connected to communicate at the top.



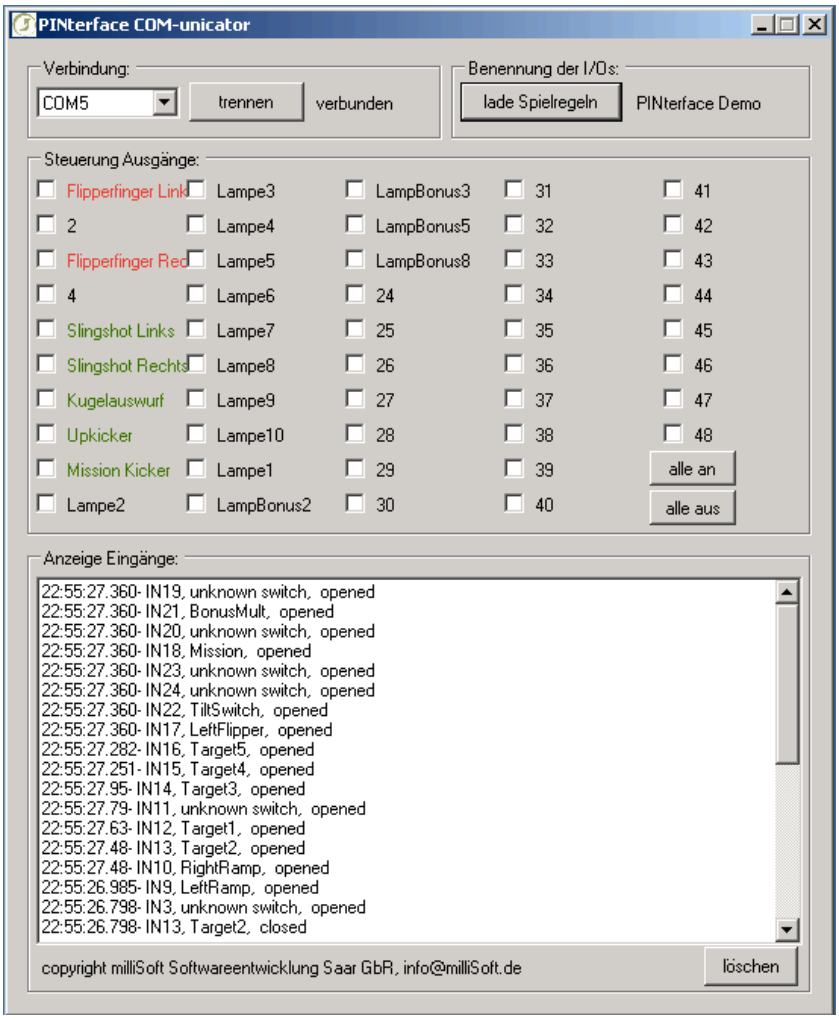
After selecting the correct com port a module can be connected by using the button "connect". Please ensure that no other program already has established a connection to the PINterface, or the connection will fail.
After successful connection, you can enable or disable the individual outputs by activating or deactivating the checkbox next to the output number (or description). Please note that the current with high-voltage may cause permanently damage to the solenoid as well as for the switching transistor if the outputs are activated for a second or more. Please be sure to disable the outputs again soon after enabling!

Activating inputs (like switches) will output the switch information in the lower text area output. The closing and opening of a switch will be output with the time (up to the millisecond) and name of the switch.

If you already created a custom rule file for your game, you can load it into „COMunicator" to rename the outputs to the correct names even for the inputs:



COMunicator without named outputs and inputs before connection



COMunicator after connecting to the I/O board and loading the XML game rule file

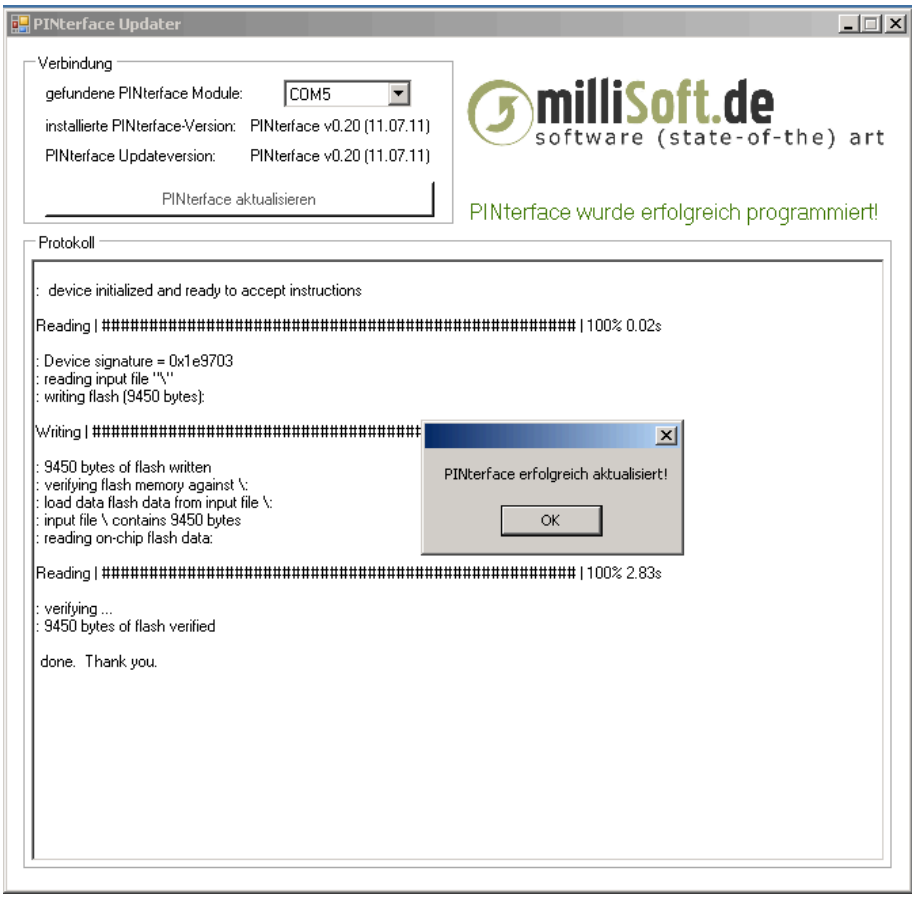
The solenoid/lamp outputs are colored red if protected and green if unprotected so that you get a visual representation of the shut-off protection to prevent any damage to the solenoid and transistor.

Firmware Updater

The software „PINterfaceUpdater" is part of every new software releases and can update the microcontroller in the PINterface interface input / output board with the latest firmware. This update is listed in the release notes of a new software if it is necessary to reprogram the microcontroller.
If you are not sure whether your microcontrollers is up to date, you can start the program and see the currently installed firmware version including the firmware version of the updater at the top.

The update program is easy to use and can upgrade with a single click automatically:

the updater after staring the program displaying current and update firmware versions



the updater installed the new firmware successfully

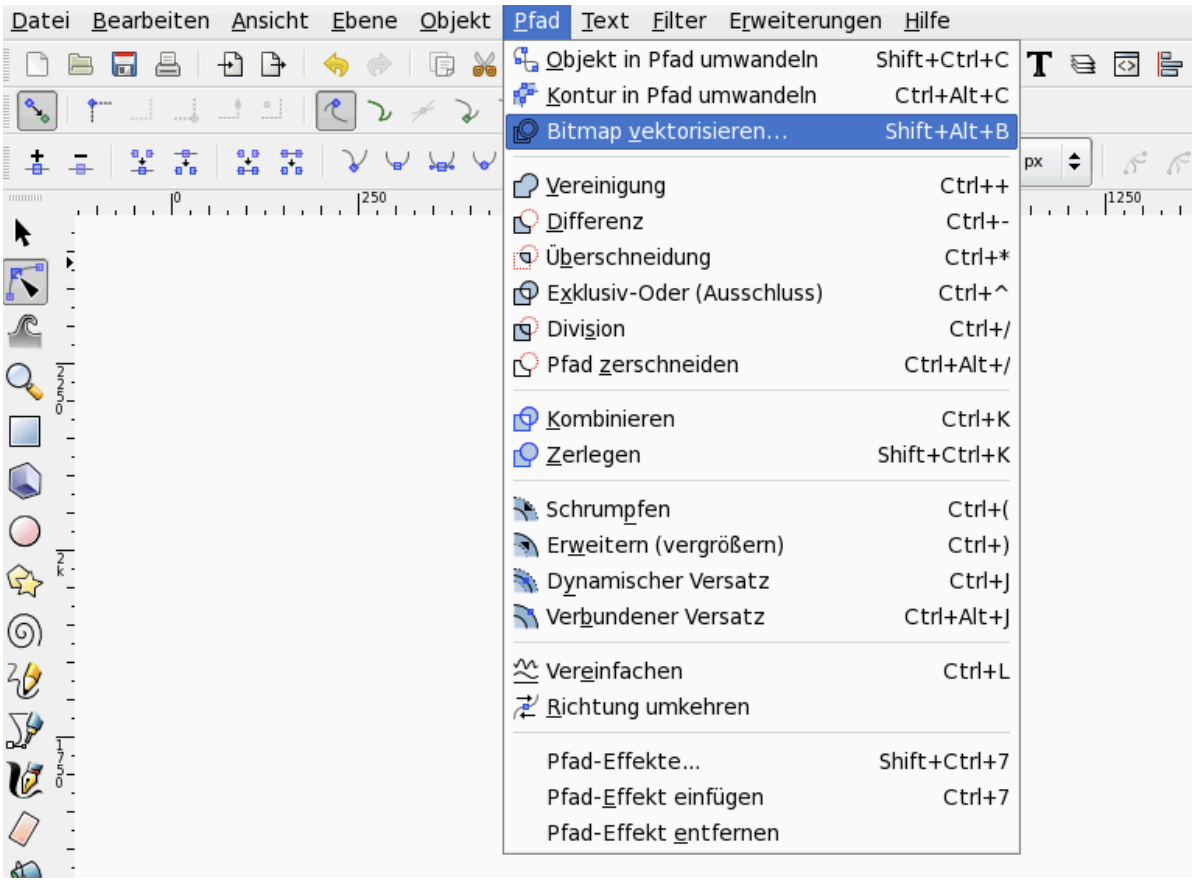
Mechanical assembly

Creating vector images from pixel images

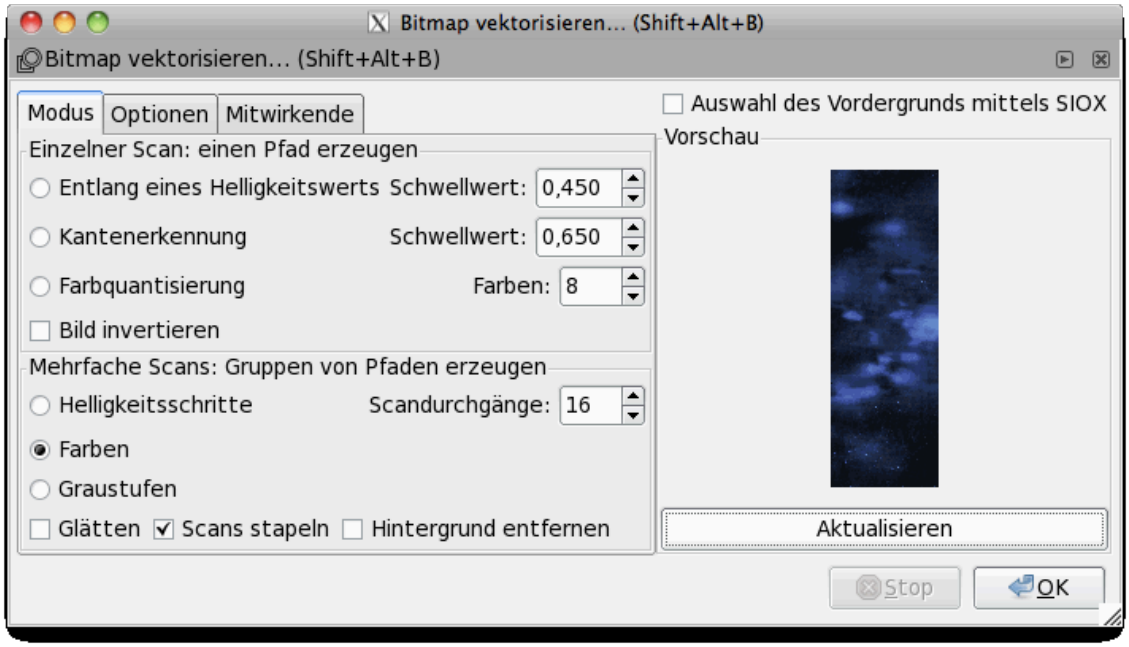
If you have small images with low detail information (like a logo or comic graphics) a technique called „vectorization" can be easily used to enlarge these graphics in order to use them as large area printed graphics for example at the cabinet side of your custom pinball mashine.

There are both free and commercial programs for the vectorization of pixel graphics. This is a graphic representation based on lines and curves instead of the points-based (pixel) representation. These vector graphics can be printed then without introducing artifacts caused by pixel enlargement in any size. A small, poorly-resolution graphics can be used thus after vectorization on a cabinet side.

A free tool for vectorizing pixel images is integrated into the free program „Inkscape":

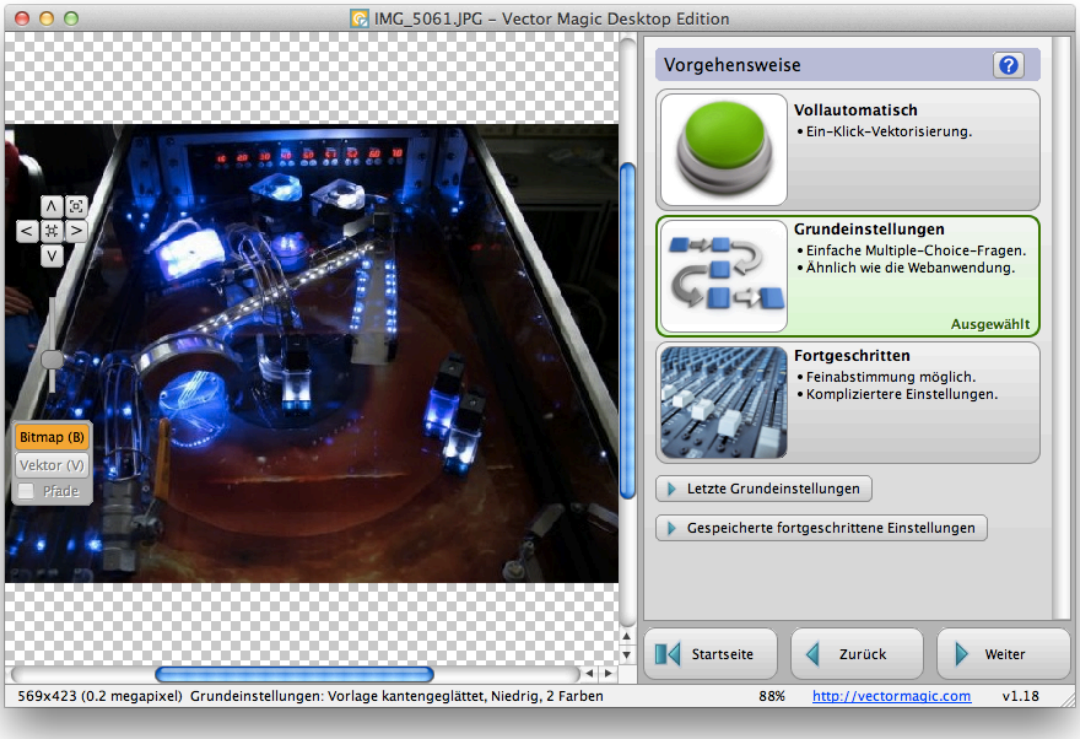


Inkspace vectorization command



Inkspace vectorization options

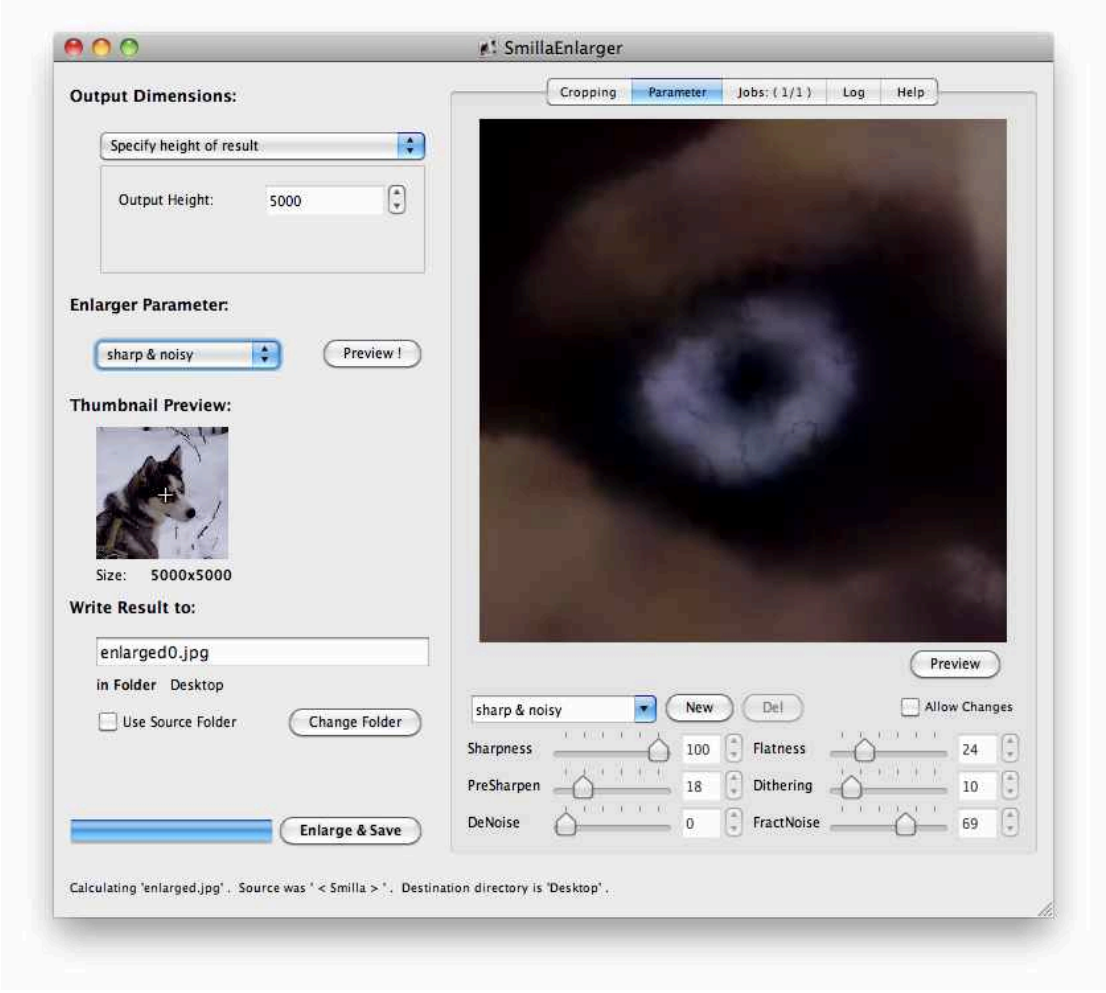
There are alternatives that are not free to use but these are professional tools such as, for example, Vector Magic:



Vector Magic

Increasing the image resolution of pixel images

If a vector conversion (see above) is not possible due to too many image details or desired photo-realism, a very high resolution version of an image can be created with special programs and algorithms:
SmillaEnlarger(http://sourceforge.net/projects/imageenlarger) is a free way to increase the image resolution for large-area printing. If you want a tool more comfortable there is also a professional and commercial (and thus not truly cheap) tool such as Benvista photo zoom Pro / Classic.



SmillaEnlarger Screenshot

After creating a basis for graphics layout and stitching everything together using a vector layout tool you have to print these large area images using an online printing service. The cost for printing approx. 5 square meter of adhesive fpol costs about 100€ when using an online printing service.
These print can then by applied to the clean cabinet (see below) using a bit of water and some time.

Wood work

A custom Pinball mashine needs also a specially adapted housing.
An old and empty Cabinet/Backbox can be a basis for the new custom pinball mashine but then it has to be reworked a lot before applying new graphics to it.
Grinding down of old stickers or prints is the first step:

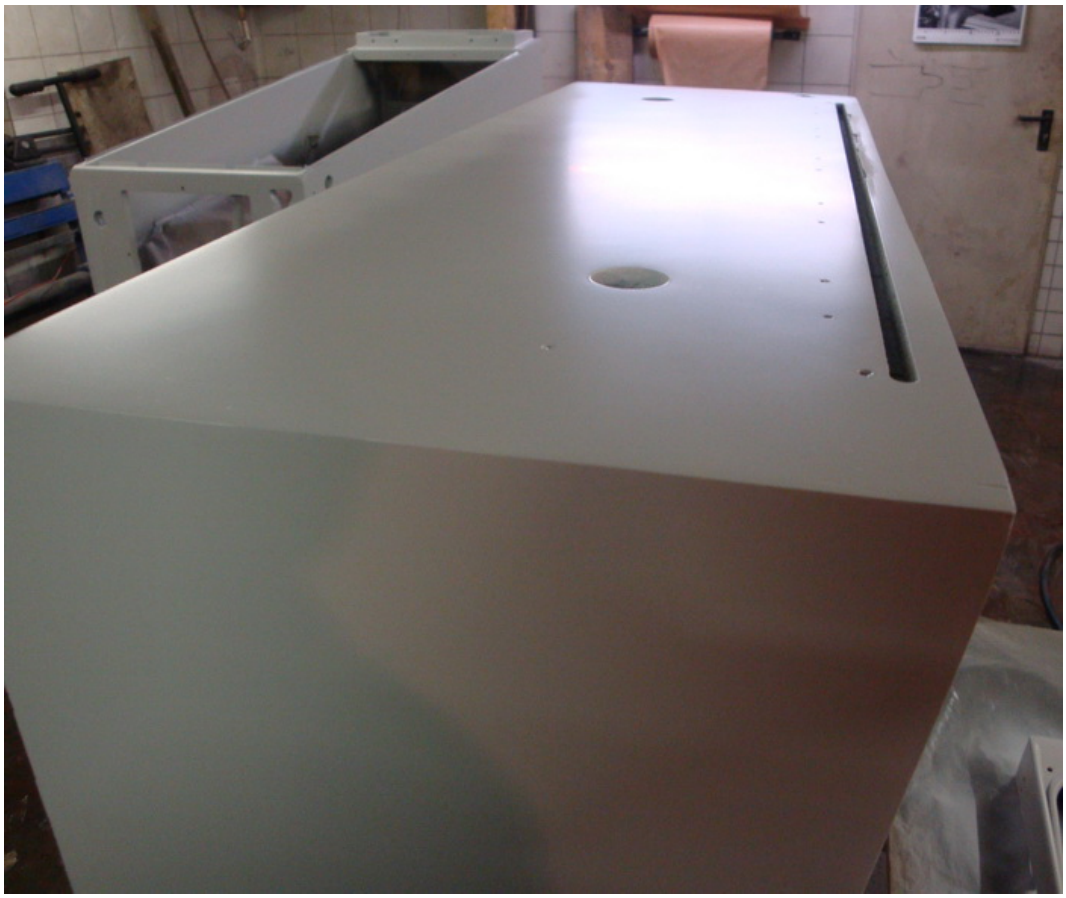
An old cabinet (theme: secret service) as the basis for a custom pinball. Ideally, a transformer is still included.

The old cabinet was polished.

The repair of scratches and aging effects can be patched by applying filler to the case. In this way, a very smooth surface is guaranteed for the painting later on.

Injection plaster was applied to the casing

Spray filler grind of the housing



Spray filler finish polished by the housing

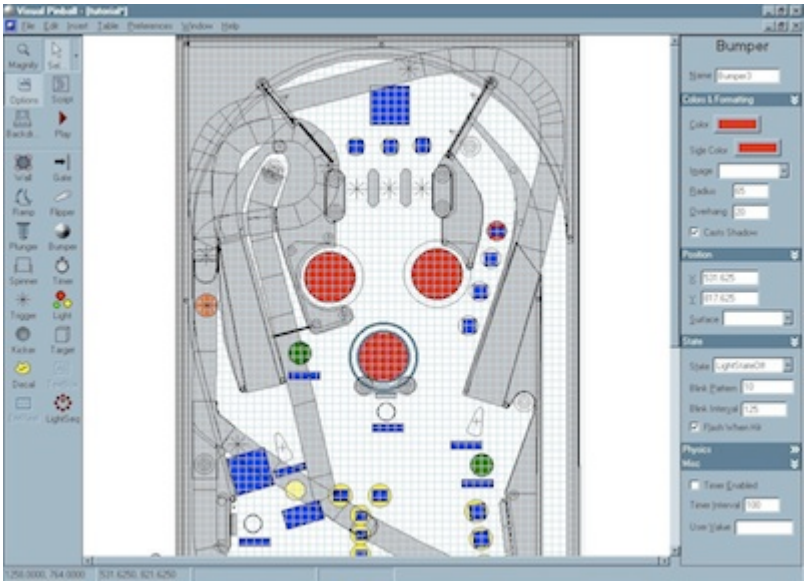
Now the next steps may be different depending on the type of "processing": the use of stickers require an ideally smooth surface, prepainted in a color (the best: black) to identify any differences in colour.

black lacquered Cabinet ready for sticking

The housing should be a colored varnish, airbrush or similar is getting necessary grinding and fillers as preprocessing probably not to this extent. For this, you ask but preferably the craftsmen of your confidence.

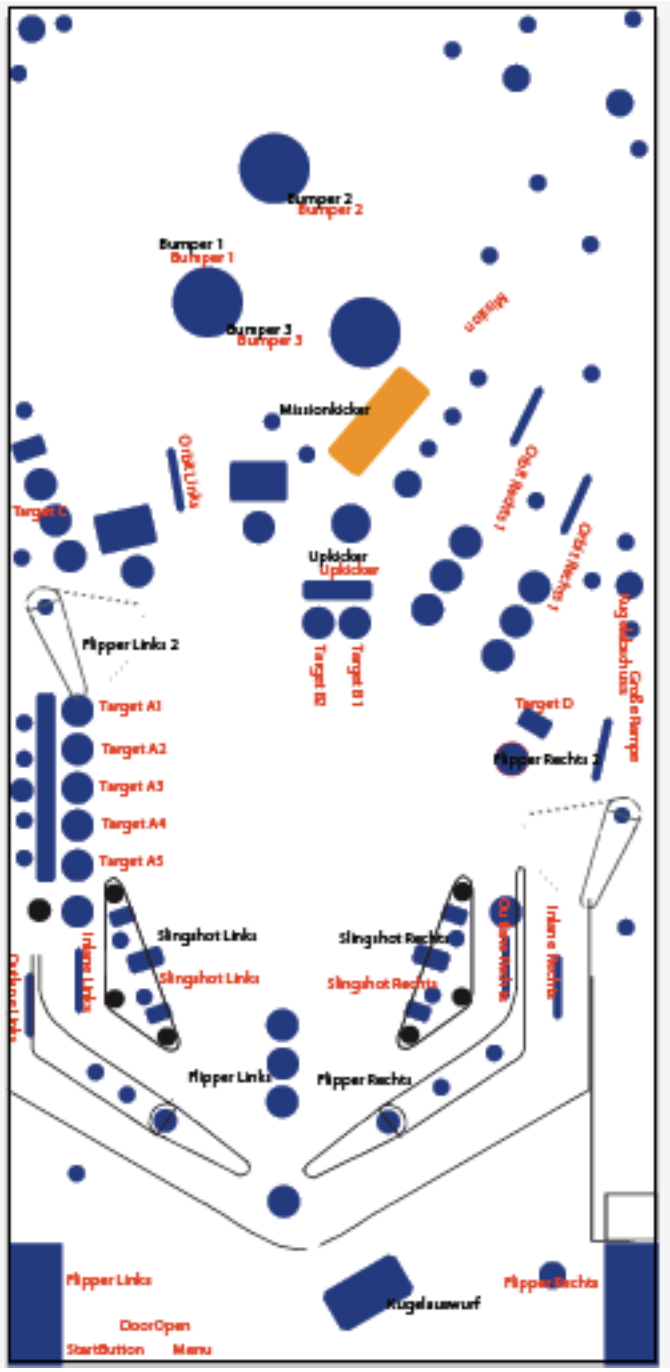
Playfield work

The layout of playfield components can be made by planning it on the PC. For this purpose programs like Visual Pinball or Future Pinball or simply a vector graphics program can be used.
The advantage of Visual Pinball is the possibility of testing the layout by playing it on the PC. Even if the physics simulation of such tools is not perfect it is possible to avoid flaws in the first design phase.



Visual Pinball Screenshot

The advantage of using a vector graphics program comes with the direct processing possibilities of the layout and graphics design. This way the a playfield can be developed quickly, processed digitally (Internet printing,...) and be applied directly on the playfield wood. Also a clipping mask can be created quickly by the same service that did the printing to be used as a mask for drilling of holes and inserts (see next image).

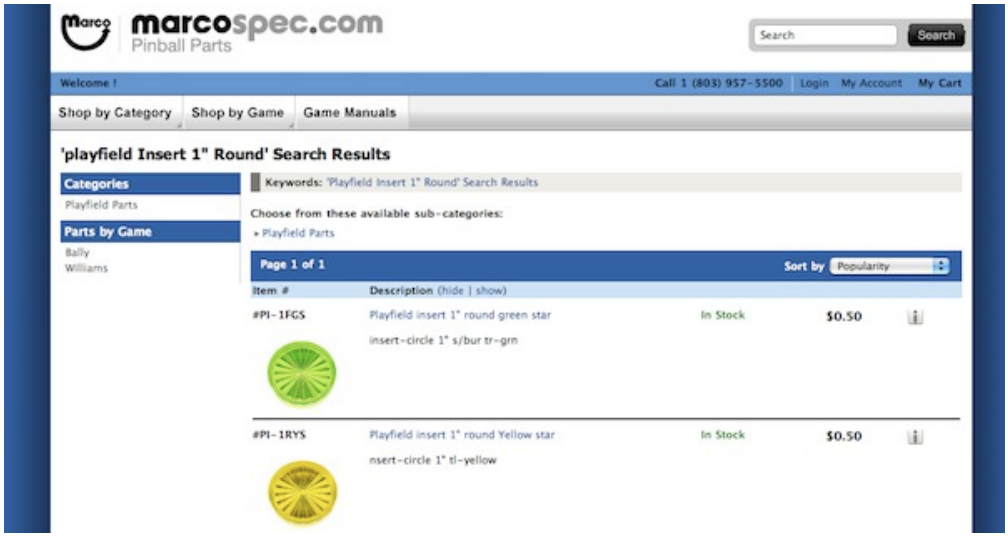


clipping mask that also was printed to localize positions of drilling

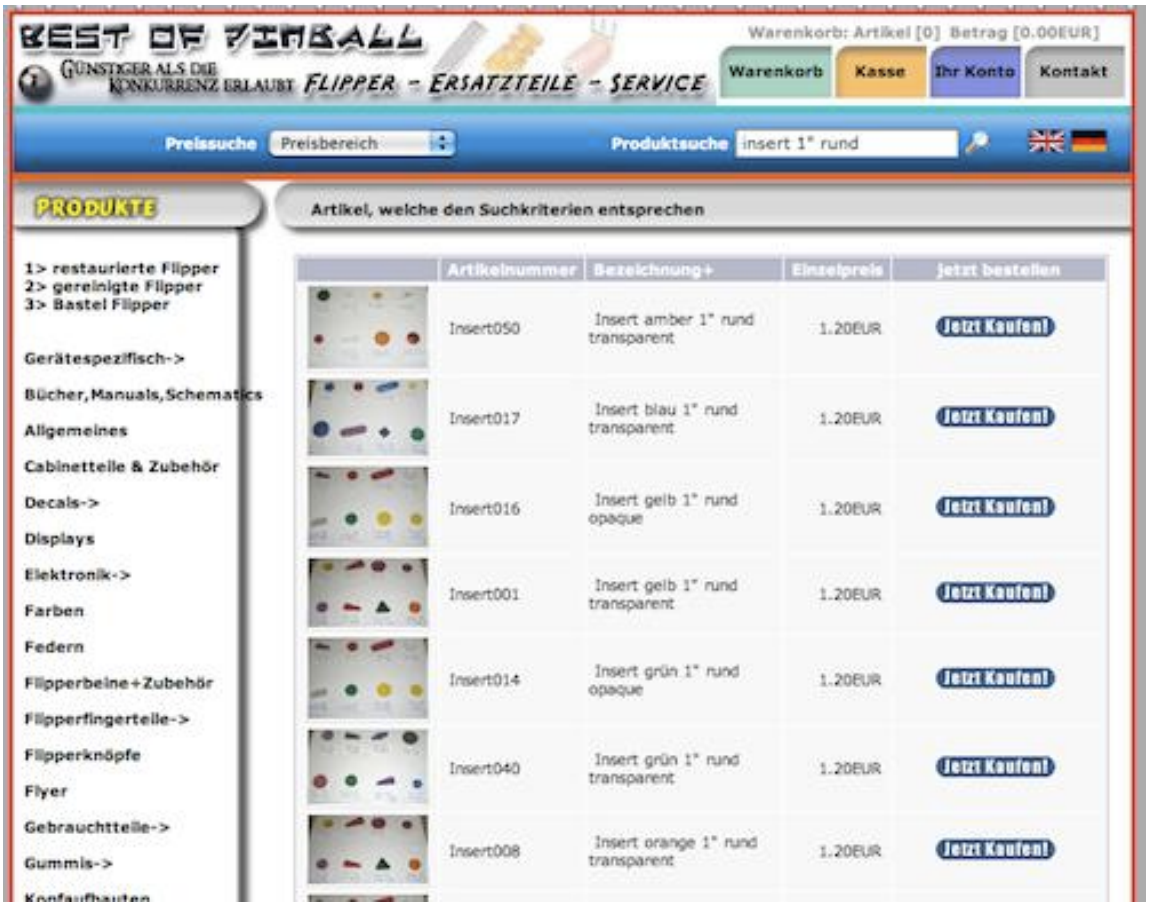
For the playfield wood a screen printing plate has proven itself as good material. This is a solid multiplex plywood, which is usually so smooth on the top that a printed label can be applied directly to it.

screen printing plate as playfield wood

For the construction of inserts we recommended the size of 1 inch (25, 4 mm) that can be created with a 25 mm Forster drill.
Matching 1 inch of inserts with star structure are available as new elements at the shop Marcospecialities in the United States.



Alternatively, there is the „best of pinball" shop in Germany that offers clear inserts with size 1".



best of pinball online shop

Annex

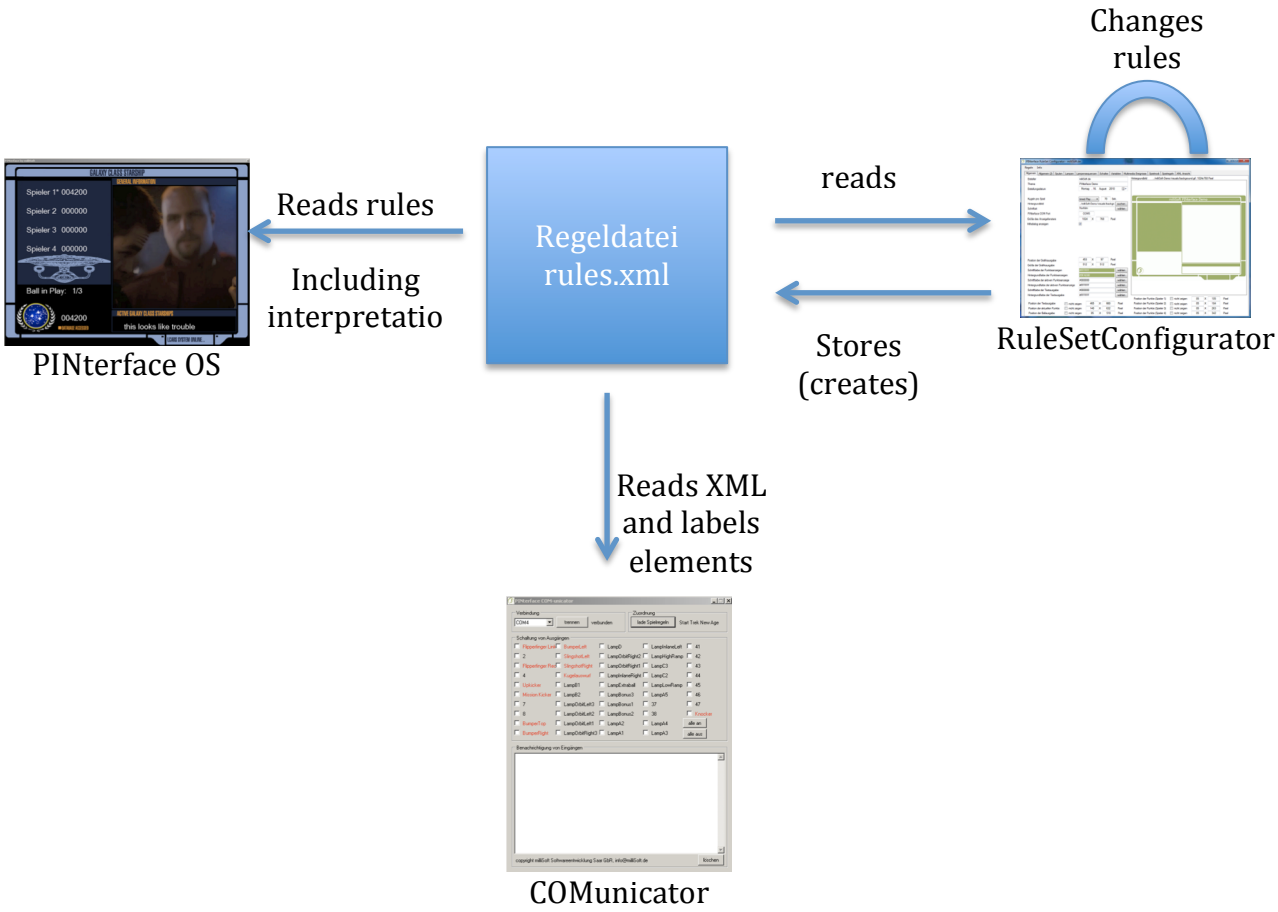
Game rule file

The rules for the operation of the pinball includes all coils, lamps, switches, multi media events and their interaction. Basis for the rules of the game is a text file in XML format that can be adjusted by hand with a text editor, or ore comfortable using the graphical configuration tool „RulesetConfigurator" included in the PINterface package.

You create a game by defining

- General settings
- Coils
- Lamps
- Light sequences
- Switch
- Game modes
- Multimedia events (music, video and sound effects)
- Game mode transitions (rules)
- Variables

The following image shows how the game rule file is used by the different programs:



An example of a rules file in the XML format is the rules.xml that is included in the demo game after installation of the software. You can use this as a basis for your modifications.

There is also a file „rules.xsd" in addition to the rules XML file but do not change this one because this file describes only the schema which elements can be used in the rules.xml. If you change the XSD file content the software is very likely to crash due to unforseen circumstances.

The next chapters give an insight into the individual elements of the rules file. You must configure these individual elements for your custom game but you can do this very comfortably using our „RuleSetConfigurator" described above.

General settings

The General information is stored in this section of the rules file. This includes the definition of standard keys for the diagnostic menu in the game and the standard game modes, where the game starts and returns to official game modes in addition to the details of the Builder and the theme.

```
<General>
      <!-- allgemeine Einstellungen-->
      <builder name="milliSoft.de"/>
      <!-- Bauer des Flippers-->
      <theme name="PINterface Demo"/>
      <!-- Thema des Flippers-->
      <BallsperPlay value="80"/>
      <!-- Anzahl der Kugeln pro Spiel-->
      <ConfigDate value="16.08.2010"/>
      <!-- Zeitpunkt der Konzeption-->
      <StartGameMode name="AttractMode"/>
      <!-- Name des Spielmodus, in dem der Flipper startet -->
      <FallBackGameMode name="StandardGameMode"/>
      <!-- Name des Spielmodus zum Wechsel nach Zeitspielmodus -->
      <BackgroundImage value="../milliSoft-Demo/visuals/background.gif"/>
      <!-- Bilddatei als Hintergrund für die Anzeige in 1024x768 Pixeln -->
      <Font name="Narkisim" size ="24"/>
      <!-- Schriftart zur Anzeige der Schriften (Punkte, "Spieler 1",. -->
      <ComPort value="COM5"/>
      <!-- Communication Port für die Platine -->

      <MenuKey name="open" value="MenuKey"/>
      <MenuKey name="left" value="LeftFlipper"/>
      <MenuKey name="right" value="RightFlipper"/>
      <MenuKey name="enter" value="StartButton"/>
      <MenuKey name="inverse" value="MenuButton"/>

      <StartButton value="StartButton"/>
      <Tilt switchname = "TiltSwitch" warnings = "2" decreasetime = "20"
warning = "TiltWarning" gameover = "Tilt"/>

      <Color name="PlayerScoresBack" value="#9fae6b" />
      <Color name="PlayerScoreActiveBack" value="#FFFFFF" />
      <Color name="TextoutBack" value="#FFFFFF" />
      <Color name="PlayerScores" value="#FFFFFF" />
      <Color name="PlayerScoreActive" value="#000000" />
      <Color name="Textout" value="#000000" />
```

```
      <ScoreOutput position="140; 632" />
      <Score1Output position="85; 105" />
      <Score2Output position="85; 184" />
      <Score3Output position="85; 263" />
      <Score4Output position="85; 342" />
      <TextOutput position="465; 660" />
      <BallOutput position ="85; 518" />

      <DialogOutputSize size ="1024;768" />
      <output_debug_info value="1"/>
      <!-- Ausgabe von zusaetzl. Infos auf dem Display-->

      <VisualOutput position="453;97" size ="512;512" />
      <ExtraballScore value="10000" multiple ="1"/>
  </General>
```

Solenoids

```
<!-- ____________________ -->
<!-- Spulendefinitionen -->
<!-- ____________________ -->
<Coils>
      <Coil id="1" addid="2" name="Flipperfinger Links"/>
      <Coil id="3" addid="4" name="Flipperfinger Rechts"/>
      <Coil id="5" name="Slingshot Links" isProtected= "1" />
      <Coil id="6" name="Slingshot Rechts" isProtected= "1" />
      <Coil id="7" name="Kugelauswurf" isProtected= "1" />
      <Coil id="8" name="Upkicker" isProtected= "1" />
      <Coil id="9" name="Mission Kicker" isProtected ="1" />
</Coils>
```

Lamps

The definition of lamps is analogous to the definition of coils of specifying the name of the lamp and the port "id", to which it is connected to the I/O Board:

```
<!-- ____________________ -->
<!-- Lampendefinitionen -->
<!-- ____________________ -->
<Lamps>
      <Lamp id="9"  name="Lampe1"/>
      <Lamp id="10" name="Lampe2"/>
      <Lamp id="11" name="Lampe3"/>
      <Lamp id="12" name="Lampe4"/>
      <Lamp id="13" name="Lampe5"/>
      <Lamp id="14" name="Lampe6"/>
      <Lamp id="15" name="Lampe7"/>
      <Lamp id="16" name="Lampe8"/>
      <Lamp id="17" name="Lampe9"/>
      <Lamp id="18" name="Lampe10"/>
</Lamps>
```

Also please note that the names of the lamp definitions must be unique!

Lamp sequences

An example of two light sequences are:

```
<!-- _____________ -->
<!-- LampSequences -->

  <LampSequence name="CircleSequence" duration="5220">
    <!-- Blinken der Lampen von unten nach oben im 200ms Intervall -->
    <Lamp name="LampExtraball" delay ="0" duration ="200"/>
    <Lamp name="LampBonus1" delay ="200" duration ="200"/>
    <Lamp name="LampBonus2" delay ="400" duration ="200"/>
    <Lamp name="LampBonus3" delay ="600" duration ="200"/>
    <Lamp name="LampInlaneLeft" delay ="800" duration ="200"/>
    <Lamp name="LampA1" delay ="1000" duration ="200"/>
    <Lamp name="LampA2" delay ="1200" duration ="200"/>
    <Lamp name="LampA3" delay ="1400" duration ="200"/>
    <Lamp name="LampA4" delay ="1600" duration ="200"/>
    <Lamp name="LampA5" delay ="1800" duration ="200"/>
    <Lamp name="LampC3" delay ="2000" duration ="200"/>
    <Lamp name="LampC2" delay ="2200" duration ="200"/>
    <Lamp name="LampLowRamp" delay ="2400" duration ="200"/>
    <Lamp name="LampHighRamp" delay ="2600" duration ="200"/>
    <Lamp name="LampB1" delay ="2800" duration ="200"/>
    <Lamp name="LampB2" delay ="3000" duration ="200"/>
    <Lamp name="LampOrbitLeft1" delay ="3200" duration ="200"/>
    <Lamp name="LampOrbitLeft2" delay ="3400" duration ="200"/>
    <Lamp name="LampOrbitLeft3" delay ="3600" duration ="200"/>
    <Lamp name="LampOrbitRight3" delay ="3800" duration ="200"/>
    <Lamp name="LampOrbitRight2" delay ="4000" duration ="200"/>
    <Lamp name="LampOrbitRight1" delay ="4200" duration ="200"/>
    <Lamp name="LampD" delay ="4400" duration ="200"/>
    <Lamp name="LampInlaneRight" delay ="4600" duration ="200"/>
    <Lamp name="LampBonus3" delay ="4800" duration ="200"/>
    <Lamp name="LampBonus2" delay ="5000" duration ="200"/>
    <Lamp name="LampBonus1" delay ="5200" duration ="200"/>
  </LampSequence>

  <LampSequence name="MissionRampsBlink" duration="610">
    <!-- Blinken der Lampen vor den Rampen im 300ms Intervall -->
    <Lamp name="LampLowRamp" delay ="0" duration ="300"/>
    <Lamp name="LampHighRamp" delay ="0" duration ="300"/>
  </LampSequence>

</LampSequences>
```

The first sequence can, for example, as a background animation for an attract mode or standard mode used (see game mode definitions). The second sequence can highlight by flashing at the ramp entrance ramps because here every 600 ms of the lamps for 300ms are switched on. This results in a blink of the lamps.

Switches

An example of the definition of the switch is:

```
<!-- ________ -->
<!-- Switches -->
```

```
<!-- _________ -->
<Switches>
      <Switch name="LeftFlipper" input="1"/>
      <Switch name="RightFlipper" input="2" />
      <Switch name="StartButton" input="4" />
      <Switch name="DrainSwitch" input="5" />
      <Switch name="MenuKey" input="7" />
      <Switch name="LeftRamp" input="9" />
      <Switch name="RightRamp" input="10" />
      <Switch name="Target1" input="12" Score="100" />
      <Switch name="Target2" input="13" Score="100" />
      <Switch name="Target3" input="14" Score="1000"/>
      <Switch name="Target4" input="15" Score="100" />
      <Switch name="Target5" input="16" Score="100" Inverted="1" />
      <Switch name="TiltSwitch" input="18" />
      <Switch name="Mission" input="20" />
</Switches>
```

Game modes

An example of the definition of game modes is as follows:

```
<!-- __________ -->
<!-- GameModes -->
<!-- __________ -->
<GameModes>
      <Mode name="AttractMode">
            <!-- Definition des Namen und ID pro Mode-->
            <ScoreMultiplier value="0"/>
            <!-- Multiplikator fuer beispielsweise hoeheren Punkten bei
Multiball-->
            <IgnoreFlippers value="1"/>
            <!-- Deaktivieren der Flipperfinger im Modus-->
            <BackgroundMusic value="../milliSoft-
Demo/audio/Background/BlockMan.mp3"/>
            <BackgroundVideo value="../milliSoft-
Demo/video/millisoftIntro.wmv" replay="1"/>
            <BackgroundPicture value="../milliSoft-
Demo/visuals/millisoft_kreis.gif"/>
            <LampSequence name="CircleSequence" replay="1"/>
      </Mode>
      <Mode name="StandardGameMode">
            <!-- Definition des Namen und ID pro Mode-->
            <ScoreMultiplier value="1"/>
            <!-- Multiplikator fuer beispielsweise hoeheren Punkten bei
Multiball-->
            <IgnoreFlippers value="0"/>
            <!-- Deaktivieren der Flipperfinger im Modus-->
            <BackgroundMusic value="../milliSoft-Demo/audio/Background/Kick
Shock.mp3"/>
            <BackgroundPicture value="../milliSoft-
Demo/visuals/millisoft_kreis.gif"/>
            <!-- <LampSequence name="BlinkSequence" replay="1"/> -->
      </Mode>
      <Mode name="BonusMode">
            <!-- Definition des Namen und ID pro Mode-->
            <ScoreMultiplier value="0"/>
            <!-- Multiplikator fuer beispielsweise hoeheren Punkten bei
Multiball-->
            <IgnoreFlippers value="1"/>
            <!-- Deaktivieren der Flipperfinger im Modus-->
```

```
            <BackgroundVideo value="../milliSoft-
Demo/video/drainball_de.wmv" replay="0"/>
      </Mode>

      <Mode id="10" name="Mission1">
      <!-- Mission 1-->
            <ScoreMultiplier value="2"/>
            <IgnoreFlippers value="0"/>
            <ModeTime value="120"/>
            <BackgroundMusic value="../milliSoft-Demo/audio/Background/Home
Base Groove.mp3"/>
            <BackgroundPicture value="../milliSoft-
Demo/visuals/millisoft_kreis_misison1.gif"/>
            <LampSequence name="MissionRampsBlink" replay="1"/>
      </Mode>

      <Mode id="11" name ="Mission1End">
      <!-- Mission 1 zu Ende-->
            <ScoreMultiplier value="0"/>
            <IgnoreFlippers value="1"/>
            <ModeTime value="2"/>
            <BackgroundPicture value="../milliSoft-
Demo/visuals/millisoft_kreis_misison1.gif"/>
      </Mode>

</GameModes>
```

Multimedia events

The following is an example of the definition of media events:

```
<!-- __________ -->
<!-- MediaFiles -->
<!-- __________ -->
<MediaFiles>
      <!-- Multimedia-daten-->
      <MediaEvent name="TiltWarning">
            <filename_video value="../milliSoft-
Demo/video/tiltwarning.wmv"/>
            <filename_audio value="../milliSoft-
Demo/audio/Effects/3398__patchen__Rhino.wav"/>
            <!-- Dateiname des abzuspielenden Soundfiles-->
            <alternate_text value="Viel Spass!"/>
            <balance value="b"/>
            <!-- Balance des Abspielens (Stereo-Effekt) l(inks), r(echts),
b(eide)-->
      </MediaEvent>

      <MediaEvent name="Tilt">
            <filename_video value="../milliSoft-Demo/video/tilt.wmv"/>
            <filename_audio value="../milliSoft-
Demo/audio/Effects/3398__patchen__Rhino.wav"/>
            <!-- Dateiname des abzuspielenden Soundfiles-->
            <alternate_text value="Viel Spass!"/>
            <balance value="b"/>
      </MediaEvent>

      <MediaEvent name="StartGame">
            <filename_video value="../milliSoft-Demo/video/start_de.wmv"/>
            <filename_audio value="../milliSoft-
Demo/audio/Effects/millisoftpinterface.wav"/>
```

```
            <!-- Dateiname des abzuspielenden Soundfiles-->
            <alternate_text value="Viel Spass!"/>
            <balance value="r"/>
      </MediaEvent>

      <MediaEvent name="HitTarget1">
            <filename_video value="../milliSoft-
Demo/video/target1_de.wmv"/>
            <filename_audio value="../milliSoft-
Demo/audio/Effects/One.wav"/>
            <alternate_text value="Target 1 getroffen!"/>
            <balance value="l"/>
      </MediaEvent>
</MediaFiles>
```

Game rules

Game rules are defined like the following examples:

```
<!-- __________ -->
<!-- GameRules -->
<!-- __________ -->
<GameRules>
      <GameRule name="BeginBall">
            <Trigger>
                  <BeginBall />
                  <!-- Diese Regel wird genutzt wenn das System eine Kugel
auswerfen muss (Begin einer Kugel) -->
            </Trigger>
            <Event>
                  <MediaEvent name="StartGame"/>
                  <!-- Löse Ton- und Videoausgabe für einen neuen Spieler
aus -->
                  <GameMode name="StandardGameMode"/>
                  <!-- ...und wechsel in den MusikauswahlModus, um danach
das Spiel zu starten -->
                  <FireCoil value="Kugelauswurf" delay="300"
duration="500"/>
            </Event>
      </GameRule>

      <GameRule name="Treffer1">
            <Trigger>
                  <Switch name="Target1"/>
            </Trigger>
            <Event>
                  <MediaEvent name="HitTarget1"/>
                  <Variable name ="LampenCounter1" action="increase" />
            </Event>
      </GameRule>

      <GameRule name="LampOnTarget1">
            <Trigger>
                  <Variable name ="LampenCounter1" state="Equal" value
="1"/>
            </Trigger>
            <Event>
                  <Variable name ="LampenCounter1" action="increase" />
                  <Lamp name ="Lampe1" state="1" />
            </Event>
      </GameRule>
```

```
      <GameRule name="TrefferLeftRamp1">
            <Trigger>
                  <Switch name="LeftRamp"/>
                  <Variable name ="RampCounterLeft" state="Less" value
="3"/>
            </Trigger>
            <Event>
                  <MediaEvent name="LeftRamp1"/>
                  <Variable name ="RampCounterLeft" action="increase" />
            </Event>
      </GameRule>

      <GameRule name="BallOut">
            <Trigger>
                  <Switch name="DrainSwitch"/>
            </Trigger>
            <Event>
                  <MediaEvent name="DrainedBall"/>
                  <GameMode name="BonusMode"/>
            </Event>
      </GameRule>

      <GameRule name="ExtraBall">
            <Trigger>
                  <PointsReached value="10000"/>
            </Trigger>
            <Event>
                  <MediaEvent name="ExtraBall"/>
                  <AddBonusMultiplier/>
                  <ExtraBall />
            </Event>
      </GameRule>

      <GameRule name="LeftFlipper">
            <Trigger>
                  <Switch name="LeftFlipper"/>
            </Trigger>
            <Event>
                  <FireCoil value="Flipperfinger Links" delay="0"
duration="0"/>
            </Event>
      </GameRule>

      <GameRule name="RightFlipper">
            <Trigger>
                  <Switch name="RightFlipper"/>
            </Trigger>
            <Event>
                  <FireCoil value="Flipperfinger Rechts" delay="0"
duration="0"/>
            </Event>
      </GameRule>


      <GameRule name="Mission1Start">
            <Trigger>
                  <Switch name="Mission"/>
                  <GameMode name="StandardGameMode"/>
                  <Variable name ="MissionNumber" state="Equal" value
="0"/>
            </Trigger>
            <Event>
```

```
                <GameMode name="Mission1"/>
                <FireCoil value="Mission Kicker" delay="12000"
duration="300"/>
                <AddBonus value="1000"/>
                <!-- <MediaEvent name="Mission1Stage1"/> -->
            </Event>
      </GameRule>
</GameRules>
```

Variables

The definition of variables looks for example like this:

```
<!-- _________________________ -->
<!-- Variablendefinitionen -->
<!-- _________________________ -->
<Variables>
      <Variable value="0" name="LampenCounter1"/>
      <Variable value="0" name="LampenCounter2"/>
      <Variable value="0" name="LampenCounter3"/>
      <Variable value="0" name="LampenCounter4"/>
      <Variable value="0" name="LampenCounter5"/>
      <Variable value="0" name="LampenCounter6"/>
      <Variable value="0" name="LampenCounter7"/>
      <Variable value="0" name="LampenCounter8"/>
      <Variable value="0" name="LampenCounter9"/>
      <Variable value="0" name="LampenCounter10"/>
      <Variable value="0" name="RampCounterLeft"/>
      <Variable value="0" name="RampCounterRight"/>
      <Variable value="0" name="VariableMission"/>
      <Variable value="0" name="MissionNumber"/>
</Variables>
```

Imprint

Contact:
If you have questions or comments please contact us:

Address:
milliSoft Softwareentwicklung Saar GbR
Holger Hewener und Daniel Speicher
Gewerbe- und Technologiezentrum (GTZ)
Rathausstraße 75-79
beim Weltkulturerbe Völklinger Hütte
66333 Völklingen

Telephone (mobil): +49 (0)176 666 09 023
Fax: +49 (0) 321 210 92 538

eMail: info@milliSoft.de
Internet: http://www.milliSoft.de