

Documentation RotateScreen.mplugin and hardware

The goal of this project was to expand the automatically, mechanically turning of the monitor in my Konami Quadro Games cab in the way that the monitor turns automatically to the right direction in MaLa and MAME when a game is selected.

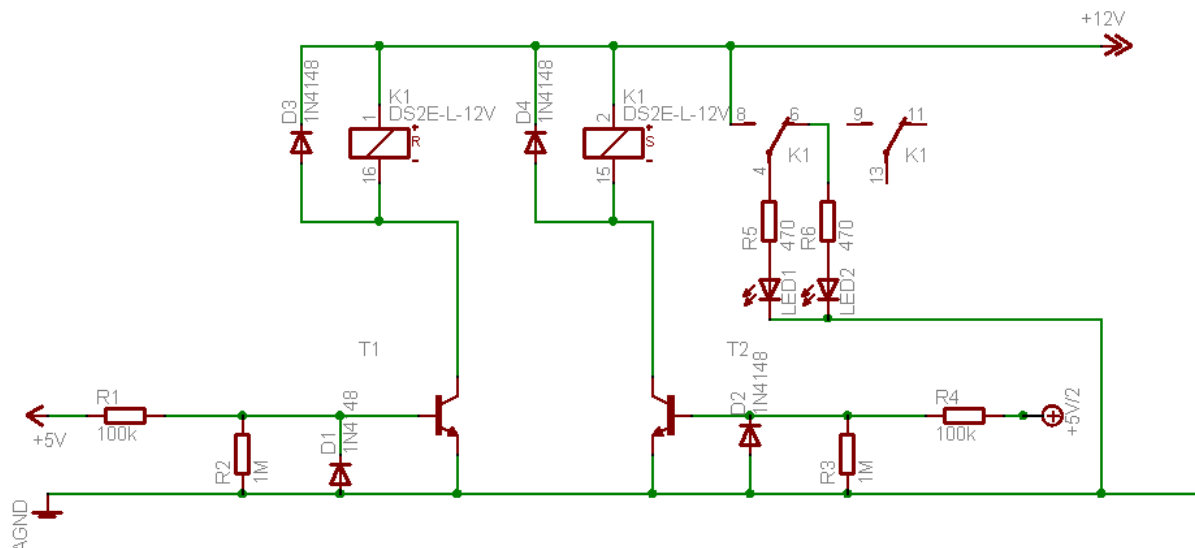
Thereto the mechanical switch on the board for the accordant JAMMA-adaptor witch sets the orientation for the monitor has to be replaced by a electronically controllable switch. Therefore I choose a bistable relay. Bistable means that the after switching of the power the last switching status of the relay persists as like as the mechanical switch does. The relay shall be operate by the serial port. The states of the serial port shall be operated by a DLL as plugin for MaLa (RotateScreen.mplugin) written in Delphi5. This plugin responds to four MaLa-events:

- MaLaStart
- MaLaGameSelect
- MaLaGameStart
- MaLaQuit

I'll describe the single functions beneath.

The hardware

Constructively to a simple relay card I found on the internet (<http://www.franksteinberg.de/erel.htm>), I extended this one in the way that a two coil latching relay can be switch with two data lines. Therefore the circuit was added with a additional transistor and related devices and changed in the way that the desired function was achieved.



The LEDs and series resistors are not essential for the main function of the circuit, so if you want to reduce the cost feel free to leave them out.

Circuit description

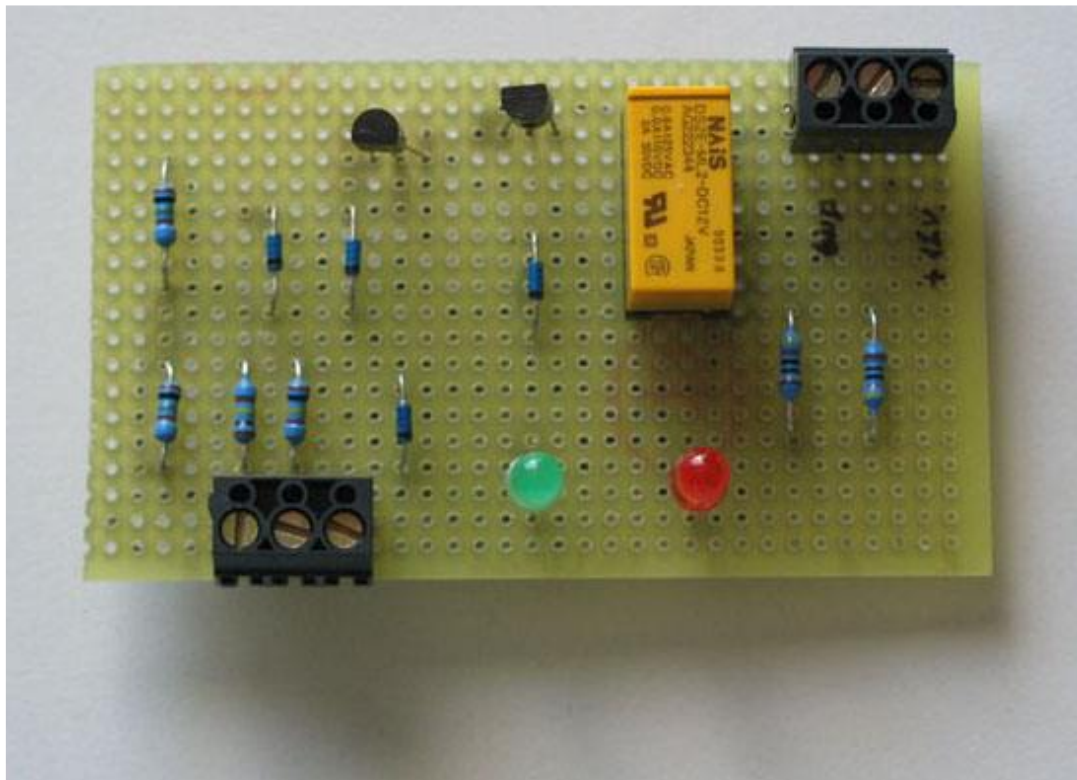
The control voltages from the serial port are routed over the 100 kOhm resistors (R1 and R2) to the bases of the two darlington transistors T1 and T2 (BC 517). If one of the two data lines are high the accordant transistor switches thru and the accordant relay coil is provided with power: the relay is switched to set or reset.

The switching status is indicated by one of the two LEDs with the according series resistors (R5 and R6 each with 470 Ohm).

The induction voltage from the relay coils are drained of via the two protection diodes D3 and D4.

D1 and D2 take care that negative voltage from the serial port is drained of via R2 or R3 to ground.

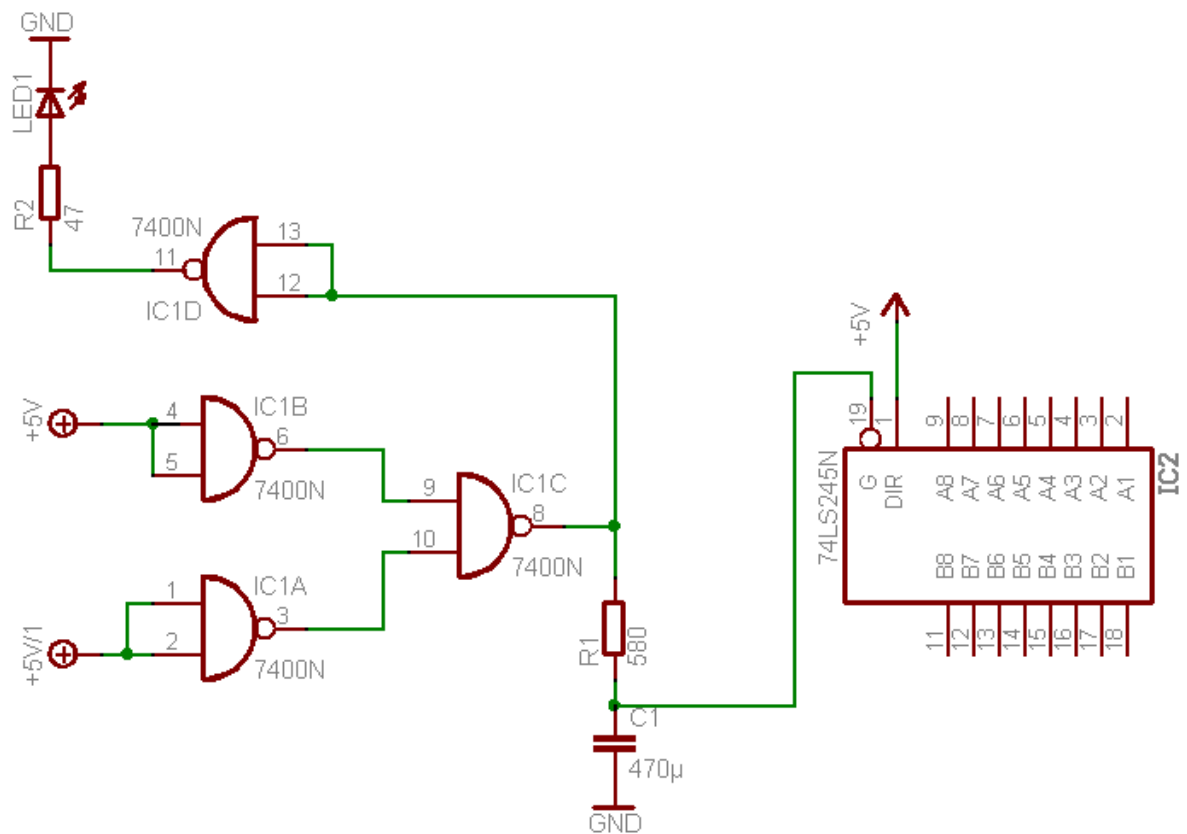
My circuit was made with a hole matrix board. The conductors on the board layout are realised with wire bridges. Pros may optimize this layout and can make a etched PCB.



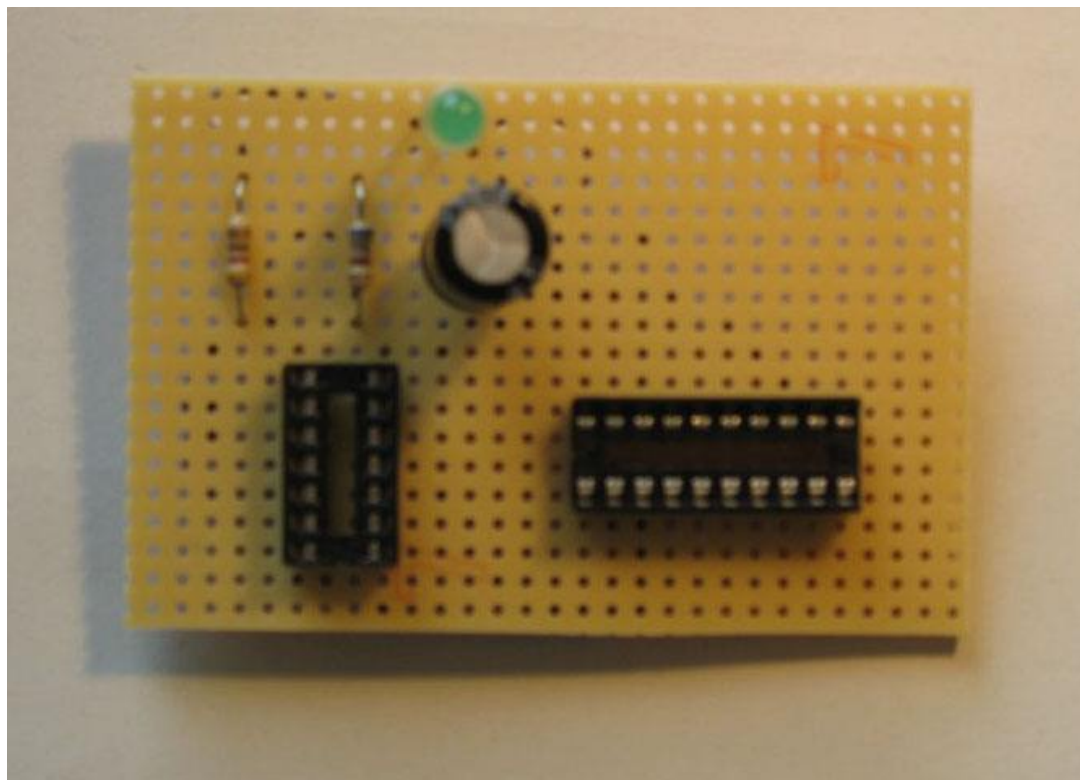
One problem to run the relay card has still to be solved: When the PC boots the serial port is accessed "wild" what may cause the damage of the relay. Therefore we have to take care that no signals are routed to the relay card during the boot and signals are routed not till establishing a predefined state.

For this I found a solution in the internet too:

(<http://www.franksteinberg.de/ZIPS/RoBue-Interface.ZIP>)



I made this circuit with a hole matrix board too.



LPT- Interfacecard circuit description

The next is a abridgement for the LPT-Interfacecard made by R. Buehler (RoBue_Relais.rtf):

Like mentioned above the LPT port and the relay card should be disconnected from each other till predefined signals are send to the relay card. The RESET respectively INIT- line is important for this but its not sufficient cause when the PC has booted the line goes to high (5V) and the relay card would react to the signals at the port.

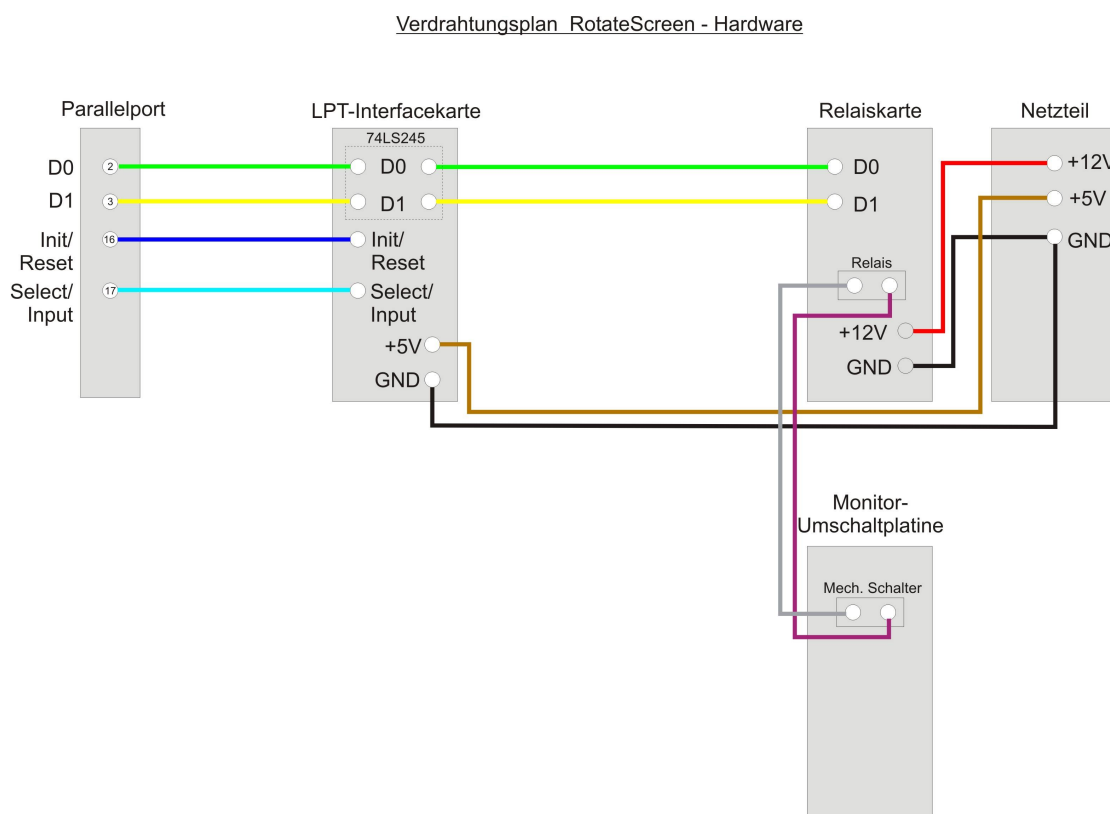
Therefore we need a additional line witch has to be linked logically to the RESET-/INIT- line so that no signals can reach the relay card without engagement of the user. Therefore I choose the SELECTINPUT-line. This line is set to high during boot and changes to low after boot contrariwise to the RESET-/INIT-line. The whole bus is conducted by a 74LS245 octal bus transceiver. It will be opened only if the RESTE-/INIT-line and the SELECTINPUT-line are low (value at the parallel port control register dec. 200 / bin 11001000).

Without external engage this value will never be reached and this is proper for controlling of the interface. But sadly there is as little moment when both lines are changing there states which causes a little impulse (LED is shortly flashing). This in fact is to short to set the coils of the relay but its not acceptable. Therefore I added a little hardware delay witch set the 74LS245 not until 1s. Therewith it doesn't react to the short changes. Now everything seams to be ok ...

This means concretely:

Before you can access a relay you have to activate the port by setting dec. 200 to the parallel port control register (for example hex. 378+2) and the wait for 1 second. Then you can switch the relay via the parallel port data register (hex 378). A high level (5V, binary 1) at one line sets the relay accordant coil.

Next is the wire plan for the complete hardware:



MaLa-plugin "RotateScreen.mplugin" description

As already mentioned the plugin was written in Delphi5. A plugin in for MaLa is nothing else but a DLL-file with the extension mplugin witch reacts on events send by

MaLa. A description for the MaLa-events can be found in the MaLa-plugin-SDK on the official MaLa homepage.

The RotateScreen-plugin reacts on four MaLa-events:

- MaLaStart
- MaLaGameSelect
- MaLaGameStart
- MaLaQuit

At the event "MaLaStart" first the circuit for the gate of the signals to the relay cards was spoken to. By setting dec. 200 to the parallel port control register the RESET-/INIT line was set to low and the 74LS245 will let through the signals to the relay card. Afterwards the defined idle state for the data lines will be adjusted by setting 0 to the parallel port data register. The orientation of the monitor does not change yet.

At the event "MaLaGameSelect" an internal variable of the RotateScreen.dll will be set with a value by means of the event value of the "g.VideoOrientation". The value of this variable represents the value of the parallel port data register and causes witch pin will be high and witch will be low and will be used in the MaLa-event "MaLaGameStart" to set the value of this data register.

At the event "MaLaGameStart" the in "MalaGameSelect" set internal variable will be used to set the data register from 0 to the value of the variable for 250ms. This causes to set the pin D0 or D1 to high for 250 ms and the accordant relay coil is powered. Depending on the position the relay will be switched or keep the position. For to set the data register temporary I have used the component "TTimer". Sadly I wasn't able to compute a created instance of TTimer in the way I want therefore I added a "help" form to the DLL and placed the visual component there. This help form will be created on the program start and stays invisible.

At the event "MaLaQuit" the RESET-/INIT line will be set to high by setting the control register to dec. 204 so that the signals will be blocked at the 74LS245. Afterwards the help form will be removed from the memory.

Hard and software functions at work

When the PC is switched on the relay card is blocked by the LPT interface (green LED is off) and nothing happens. The current position of the relay is indicated by one of the two LEDs on the relay card. On MaLa start the event MaLaStart is triggered and the installed plugin "RotateScreen.mplugin" reacts to this event by setting the LPT interface card to "pass" (dec. 200 to control register, green LED glows) and set both data lines D0 and D1 to low (dec. 0 at data register). Right now obviously nothing happens except the green LED glows cause the relay remembered the last position.

Every time a game is chosen in MaLa the MaLa event "MaLaGameSelect" is send and the plugin everytime sets the value of the internal variable to control the data register. This is not visible by the user. Not until a chosen game is started and thereby MaLa sends the event "MaLaGameStart" the plugin reacts to this event and sets the according data line D0 or D1 to high for 250 ms via the internal variable. Now either the actual monitor orientation fits the orientation of the chosen game and the according coil is powered for 250 ms but the relay doesn't switch or the actual monitor orientation doesn't fit the orientation of the chosen game then other coil is powered for 250 ms and the relay switches over. In the last case the monitor will be

turned automatically via the electronic of the Konami Quadro cap to the new position and another LED will glow (the prior glowing LED will be extinguish). After quitting the game MaLa turns the display caused by the change of the orientation and another game can be chosen in the actual gamelist. If MaLa will be closed it sends the event "MaLaQuit" and the plugin reacts to this event and blocks the LPT interface by setting dec. 204 to the control register. At last the help form will be deleted from the memory.