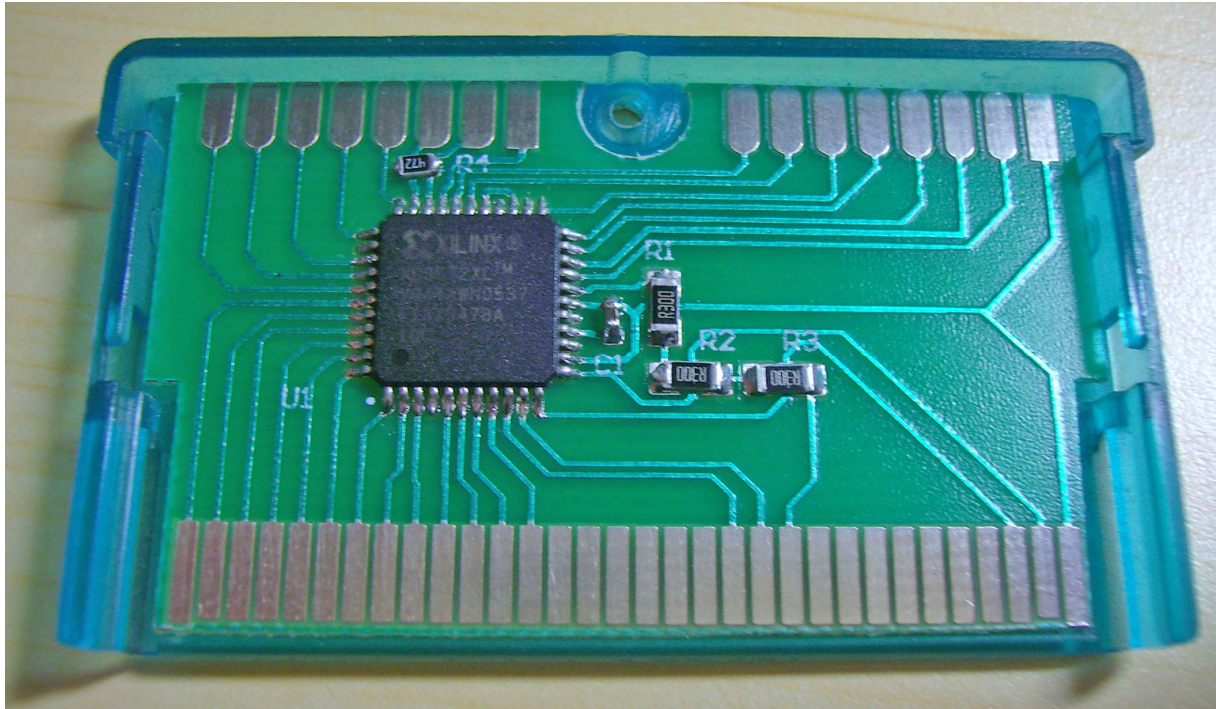


# Xiltendo™ CPLD Starter



## Features

- No programming cable needed – fully in-system reprogrammable
- 72 Macrocell CPLD supported by free development tools (Xilinx WebPack)
- 2 user extension slots with 6+7 User I/Os
- Ready to use hardware configurations
- Many hardware examples with full source code
- Open-source Software for the CPLD configuration

## Hardware Configurations

- UART (IrDA support also possible)
- MIDI interface
- SPI Master
- I<sup>2</sup>C Master
- Onewire™ Master
- Cypress PSoC Programming interface
- MMC/SD/SDIO Interface
- NAND Flash Interface
- JTAG interface
- Timer for performance profiling
- Frequency meter
- Pulse counter
- PWM generator
- R/C Servo Interface
- GPIO Interface
- User defined Interface

## ***Introduction***

Xiltendo CPLD Starter is an entry level CPLD device for the Nintendo Handhelds. It can be used for various applications with supplied hardware configurations, or it can be used as CPLD development board to create new hardware.

### **Console compatibility**

- NDS (Nintendo Dual Screen) <sup>1</sup>
- NDSL (NDS Lite) <sup>1</sup>
- GBA (Gameboy Advance) <sup>2</sup>
- GBA Micro <sup>2</sup>
- GPA SP <sup>2</sup>

Note 1: A Slot-1 homebrew launcher is required.

Note 2: A boot-cable or other option for joybus booting is required.

## **Configuration (CPLD Programming)**

The CPLD configuration interface is connected to the Cart-bus in such way that it is possible to reconfigure (program the CPLD) while the CPLD Starter device is plugged into the cart-slot. The configuration can be done using Xilinx XSVF Player that has been ported to NDS. This player will “read” pseudo JTAG Commands and execute JTAG sequences that will erase and program the CPLD.

### **Standalone XSVF Player (no FAT version)**

This is the simplest version of the Xilinx XSVF Player; it has no user interface and requires no file-system support, the XSVF binary is compiled into the NDS executable file. The XSVF file should be placed into the “data” directory of the project source tree, then the make will build new executable automatically.

### **DSOrganize XSVF Player executable**

This is a simple DSO executable that can be used to reconfigure the CPLD, the XSVF file is embedded so there is no user selection. By executing the file from the browser the reconfiguration is performed automatically.

### **DSOrganize XSVF plugin**

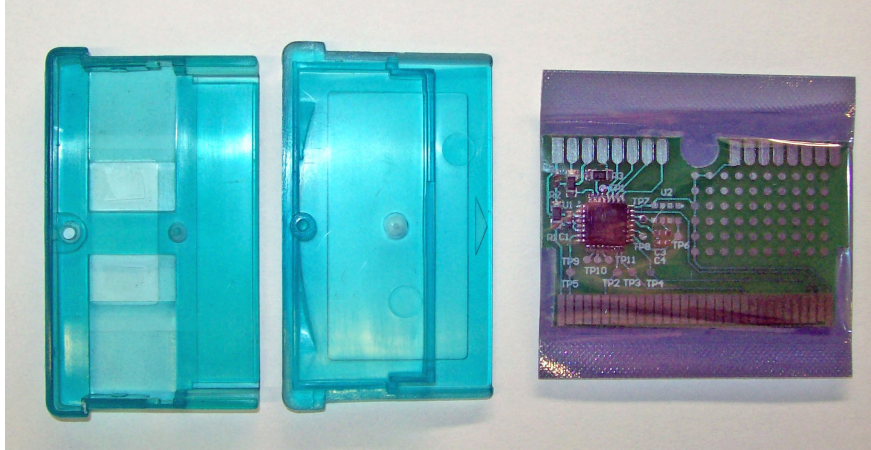
If a DSOrganize plugin configured for the configuration interface is placed into the DSO plugins directory, then selecting a XSVF file from browser allows it to be executed those configuring the CPLD with the new XSVF file. Note that this will disable any XSVF player plugin that is used to configure external devices.

### **Configuration from user application**

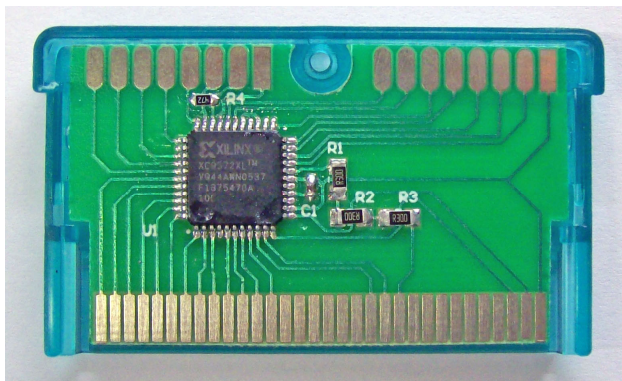
It is very simple to add reconfiguration capability to the NDS applications, a small set of C files should be added to project and one function needs to be called.

## Adding user connectors

The CPLD Starter is usually shipped with no user connectors mounted. It is up to the end user either solder a connector or add wires to the PCB edge pads.



The plastic shell and fully tested and assembled CPLD Starter PCB in ESD bag

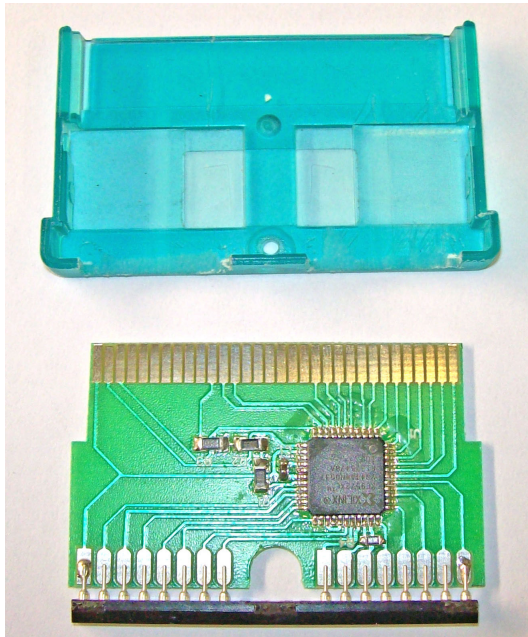


PCB matches the plastic housing, if it does not fit ok the PCB edges can be adjusted a little so that the PCB can be inserted without force.

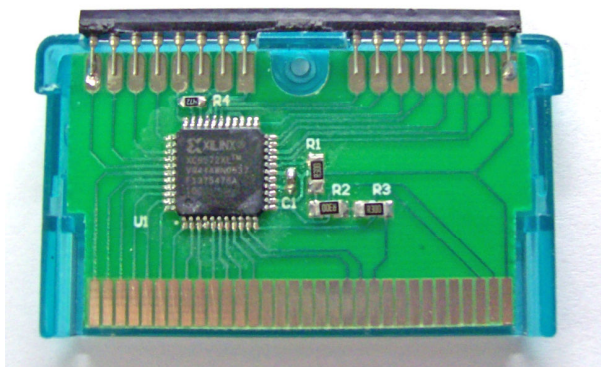


A piece of DIP40 socket cut into half can be used for the user connector.

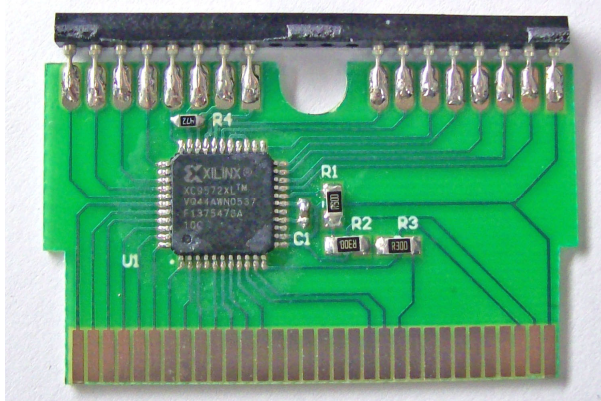




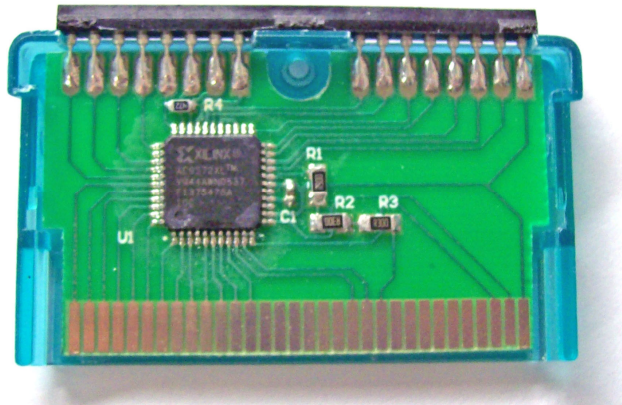
First solder the pin-stripe from the ends only. Now the plastic shell should be cut to have cut-outs for the connector.



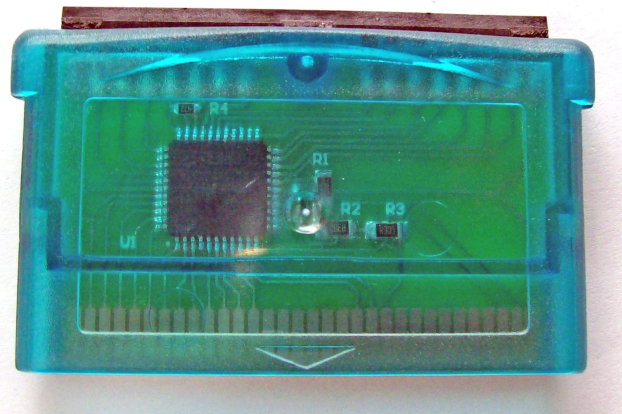
It's a good idea to test the fit before soldering the socket-stripe full.



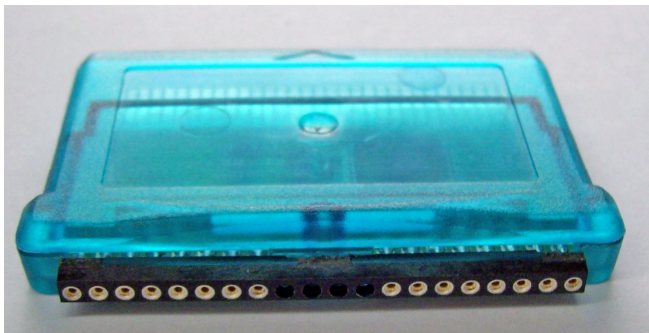
PCB with the fully soldered connector



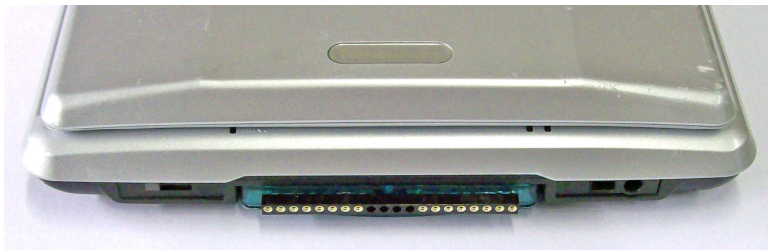
PCB with connector inserted into the shell



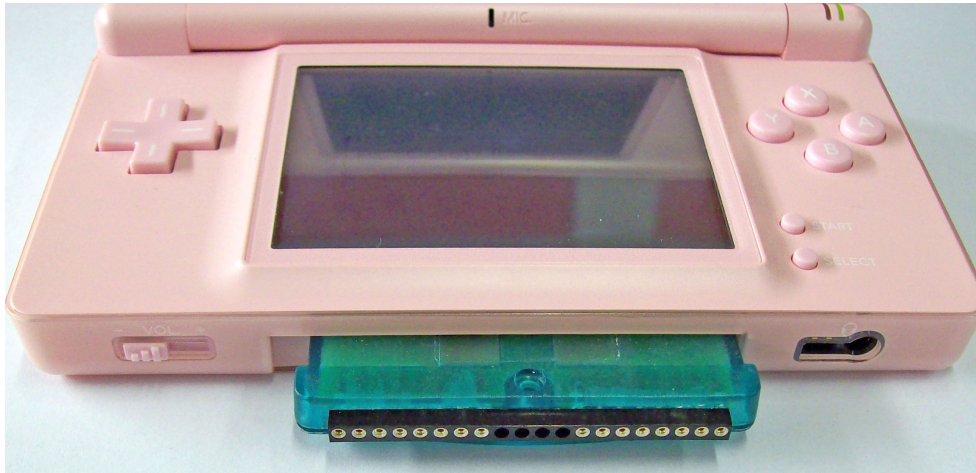
PCB is now inserted into the shell  
f



Here the connector look from the side

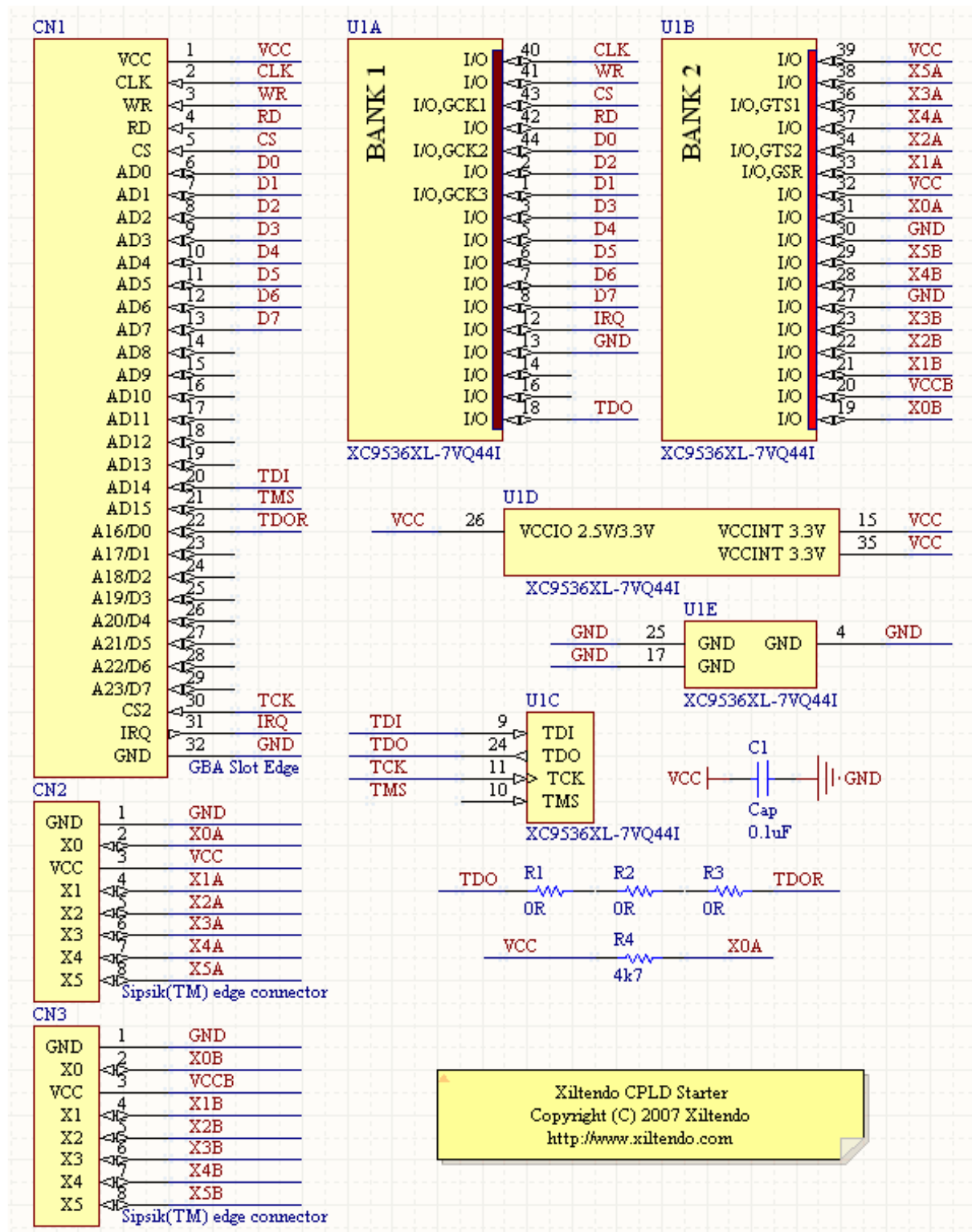


CPLDStarter inserted into NDS console



CPLDStarter with NDSL console

## Schematic





## ***UCF File (pin map constraints)***

```
#
# Xiltendo(tm) CPLD Starter
#
# NDS Slot-2 (GBA Slot)
# Control signals
NET "PHI"          LOC = "P40";
NET "CS"           LOC = "P43";
NET "RD"           LOC = "P42";
NET "WR"           LOC = "P41";
NET "REQ"          LOC = "P12";
# ROM Space Address/Data
NET "AD<0>"         LOC = "P44";
NET "AD<1>"         LOC = "P1";
NET "AD<2>"         LOC = "P2";
NET "AD<3>"         LOC = "P3";
NET "AD<4>"         LOC = "P5";
NET "AD<5>"         LOC = "P6";
NET "AD<6>"         LOC = "P7";
NET "AD<7>"         LOC = "P8";
# User extension Slot A
NET "XA<0>"         LOC = "P31";
NET "XA<1>"         LOC = "P33";
NET "XA<2>"         LOC = "P34";
NET "XA<3>"         LOC = "P36";
NET "XA<4>"         LOC = "P37";
NET "XA<5>"         LOC = "P38";
# User extension Slot B
NET "XB<0>"         LOC = "P19";
NET "XB<1>"         LOC = "P21";
NET "XB<2>"         LOC = "P22";
NET "XB<3>"         LOC = "P23";
NET "XB<4>"         LOC = "P28";
NET "XB<5>"         LOC = "P29";
NET "VCCB"         LOC = "P20";
# Externally connected VCC
CONFIG PROHIBIT = "P39";
```

## ***References***

Xilinx <http://www.xilinx.com>

DevkitPro <http://www.devkitpro.org>

DSOrganize