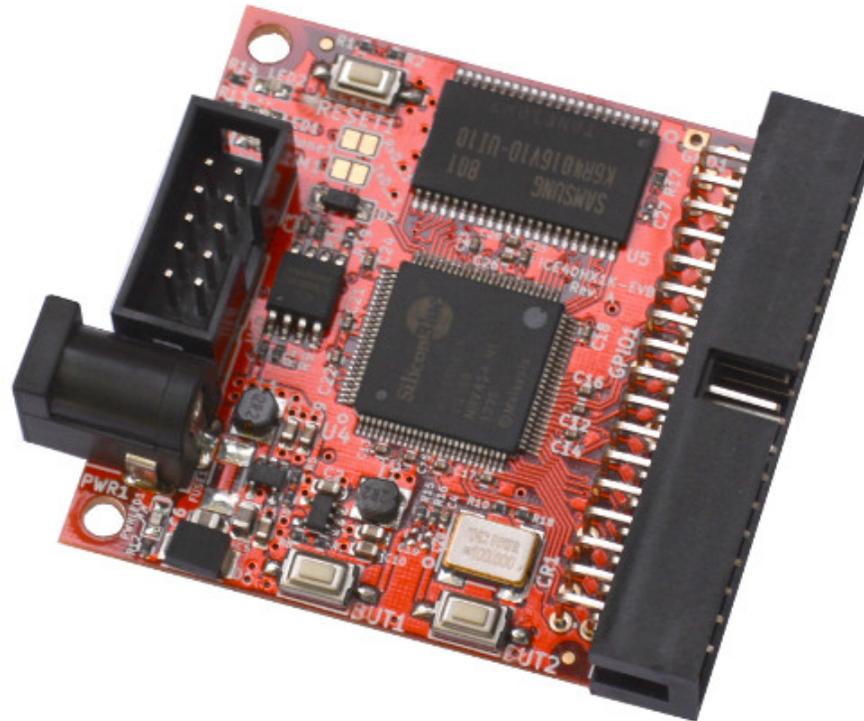


## ICE40HX1K-EVB



### [Contents](#)

[hide]

- [1 Documents](#)
- [2 Hardware](#)
- [3 Software](#)
  - [3.1 Get started under Linux](#)
    - [3.1.1 Installing Icestorm FOSS tools for iCE40](#)
    - [3.1.2 Preparing OLIMEXINO-32U4 as programmer](#)
    - [3.1.3 Hardware connection between OLIMEXINO-32U4 and iCE40HX1K-EVB](#)
    - [3.1.4 Add iceprogduino to IceStorm](#)
    - [3.1.5 Make LED blink with IceStorm](#)
    - [3.1.6 Iceprog with Raspberry PI](#)
  - [3.2 Get started under Windows](#)
- [4 Projects](#)
- [5 FAQ](#)

## Documents

[Product Selection Guide](#)

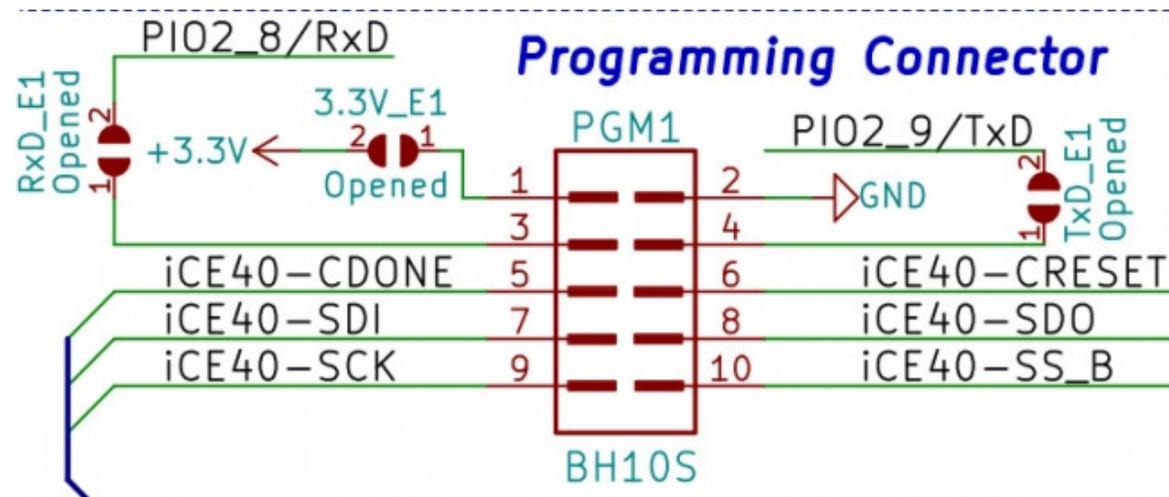
[iCE40 Family Handbook](#)

[iCE40 datasheet](#)

## Hardware

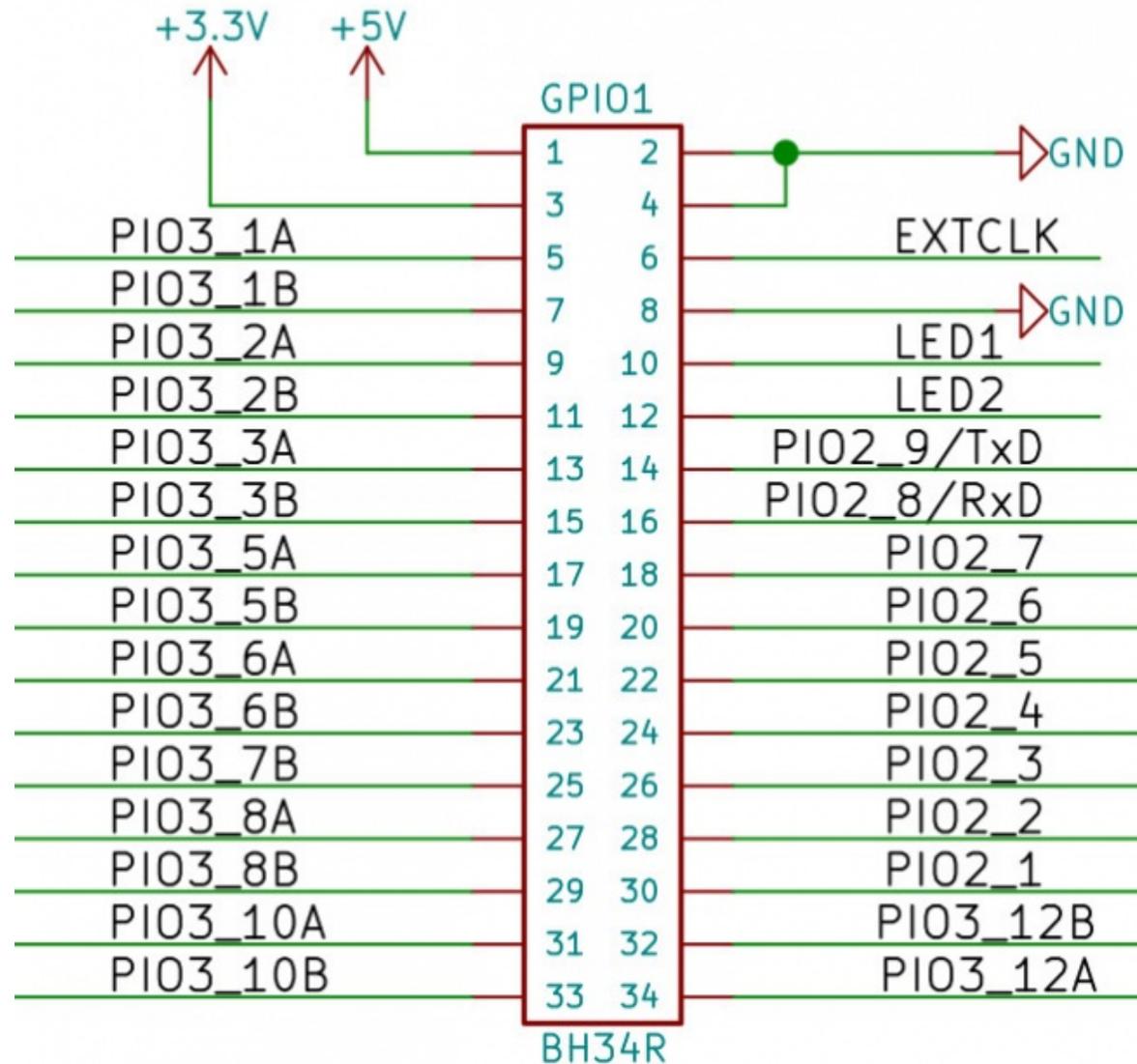
[GitHub repository with the hardware files](#)

iCE40HX1K-EVB programming connector



iCE40HX1K-EVB 34-pin bus connector

# Extension



## Software

Get started under Linux

The hardware requirements to develop for the platform are:

- iCE40HX1K-EVB or iCE40HX8K-EVB
- 3.3V programmer board - OLIMEXINO-32U4 (customers have reported success using other Ard
- IDC10-15cm cable to connect iCE40HX1K-EVB with OLIMEXINO-32U4
- SY0605E 5V power supply adapter
- USB-MINI cable
- optional but recommended USB-ISO

### Installing Icestorm FOSS tools for iCE40

First we need to install the tools necessary to build Icestorm:

```
sudo apt-get install build-essential clang bison flex libreadline-dev \  
gawk tcl-dev libffi-dev git mercurial graphviz \  
xdot pkg-config python python3 libftdi-dev
```

Installing the IceStorm Tools (icepack, icebox, iceprog, icetime, chip databases):

```
git clone https://github.com/cliffordwolf/icestorm.git icestorm  
cd icestorm  
make -j$(nproc)  
sudo make install
```

Installing Arachne-PNR (the place&route tool):

```
git clone https://github.com/cseed/arachne-pnr.git arachne-pnr  
cd arachne-pnr  
make -j$(nproc)  
sudo make install
```

Installing Yosys (Verilog synthesis):

```
git clone https://github.com/cliffordwolf/yosys.git yosys  
cd yosys  
make -j$(nproc)  
sudo make install
```

The Arachne-PNR build converts the IceStorm text chip databases into the arachne-pnr binary chip databases. Always rebuild Arachne-PNR after updating your IceStorm installation.

Notes for Archlinux: just install icestorm-git, arachne-pnr-git and yosys-git from the Arch User Repository (no need to follow the install instructions above).

You might want to try a new alternative tool instead of arachne-pnr: nextpnr is new place and route tool from the IceStorm-project. It has improved capabilities like GUI to show the connections between cells of FPGA and support for TRISTATE buffer. You might want to check it here:

<https://github.com/YosysHQ/nextpnr>

### Preparing OLIMEXINO-32U4 as programmer

Download latest [Arduino IDE](#) we recommend you to use the one from Arduino.cc current revision is 1.6.9.

Copy iceprog.ino sketch from GitHub/iCE40HX1K-EVB/programmer/olimexino-32u4 firmware to examples folder.

Plug the USB cable to OLIMEXINO-32U4 and start the Arduino IDE

```
cd arduino-1.6.9
sudo ./arduino
```

From Tools select 'Arduino Leonardo' as board.

From same menu select the PORT where the board is attached it will be something like: '/dev/ttyACM0 (Arduino Leonardo)' or '/dev/ttyUSB0 (Arduino Leonardo)'

Also install the SPIMemory library version 2.2.0 (formerly known as SPIFlash). Older versions can't be installed via the Arduino Library Manager so you need to do it manually, download the ZIP archive from here:

<https://github.com/Marzogh/SPIMemory/releases/tag/v2.2.0>

Load the library in Arduino from Sketch -> Include Library -> Add .ZIP library...

Open the iceprog.ino sketch

Compile and Upload the sketch.

If everything completes without errors now you have your OLIMEXINO-32U4 set as iCE40HX1K-EVB programmer!

### Hardware connection between OLIMEXINO-32U4 and iCE40HX1K-EVB

It is a pretty straight-forward - use a 10-pin UEXT compatible cable between the two boards (like CABLE-IDC10-15cm). However, also make sure that the switching button of OLIMEXINO-32U4 is set to position 3.3V!

The signals at the PGM1 connector of iCE40HX1K-EVB are at 3.3V DC! If you use a 5V-only board instead of OLIMEXINO-32U4 you need to drive all SPI signals to 3.3V DC externally. Else there might be a short-circuit or the connection would be unsuccessful.

### **Add iceprogduino to IceStorm**

Copy iceprogduino folder from GitHub [programmer](#) to icestorm and build it

```
cd icestorm/iceprogduino
make clean
make
make install
```

### **Make LED blink with IceStorm**

Copy from GitHub [ice40hx1k-evb](#) to icestorm/examples then

```
cd icestorm/examples/ice40hx1k-evb
make
make prog
```

LED1 and LED2 should start blink which means the code is successfully loaded

### **Iceprog with Raspberry PI**

This section is contributed by Andreas Seltenreich



	17	=3v3=		1	=3v3=		
	18	=gpio24=		6	=creset=		
	19	=mosi=		8	=sdo=		
	20	=gnd=		2	=gnd=		
	21	=miso=		7	=sdi=		
	22	=gpio25=		5	=cdone=	optional	
	23	=clk=		9	=sck=		
	24	=ce0=		10	=#cd = ss_b=		
	25	=gnd=					
	26	=ce1=					

enable SPI device on the Pi (needs reboot)

```
echo dtparam=spi=on >> /boot/config.txt
```

build and install flashrom

```
git clone https://www.flashrom.org/git/flashrom.git
cd flashrom
make CONFIG_ENABLE_LIBPCI_PROGRAMMERS=no CONFIG_ENABLE_LIBUSB0_PROGRAMMERS=no
CONFIG_ENABLE_LIBUSB1_PROGRAMMERS=no install
```

Once you have enabled the SPI interface and once you have installed flashrom you you can use manual commands or check the scripts that easen it up below.

#### **Manual operation:**

claim GPIO24 for sysfs-control

```
echo 24 > /sys/class/gpio/export
```

Pull GPIO24 low to put the ice40 into reset. The cdone-LED on the board should turn off.

```
echo out > /sys/class/gpio/gpio24/direction
```

Read the flash chip at 20MHz (for short cabling)

```
flashrom -p linux_spi:dev=/dev/spidev0.0,spispeed=20000 -r dump
```

Simply swap -r for -w to write the dump back

```
flashrom -p linux_spi:dev=/dev/spidev0.0,spispeed=20000 -w dump
```

As generated bitstreams are smaller than size of the flash chip, you need to add padding for flashrom to accept them as image.

I used the following commands to do that:

```
tr '\0' '\377' < /dev/zero | dd bs=2M count=1 of=image  
dd if=my_bitstream conv=notrunc of=image
```

Deassert reset to let the ice40 read the configuration from the bus:

```
echo in > /sys/class/gpio/gpio24/direction
```

**Automated scripts:**

Check Sagar Acharya's repo, he did some testing with RPi3 and RPi3B+ and made scripts that automate the process described above:

[https://notabug.org/sagaracharya/swarajya/src/master/hdl\\_to\\_hx8k](https://notabug.org/sagaracharya/swarajya/src/master/hdl_to_hx8k)

---

---

---

---

---

