Milkymist One
A video synthesizer at the forefront of open source hardware

S. Bourdeauducq

Milkymist project

August 2011
What is open source hardware?
Openly licensed development kits from chip makers?
Beagleboard (TI)
mbed (ARM)
In fact nothing really new...

- They sell (proprietary) chips.
- They always have been happy to see you copy their “reference designs”.
Community-developed devices for hobbyists?
Arduino
RepRap
Grassroots movement
More potential for open innovation

But...
Niche markets
Small volumes
Hackers are cheap
Low price implies low technology
Open source practice for developing “industry-grade” hardware?
Openmoko
Milkymist One
A video synthesizer for...

- **VJs** who want low latency, cutting edge interactive effects.
- **Clubs and concert venues** who can use it like a nice “lighting effect”.
- **Small music bands** who want plug and play VJing.
- (Hackers who want to play with advanced digital technology.)
Those are niche markets and “small” volumes too, but...

- Milkymist One is better and cheaper than existing devices.
  - which sold in the order of 40000 units
  - publicity is of utmost importance (We sold 40 so far)

- Competitive even in small volumes.

- We can afford cutting edge technology.
  - Milkymist One is one of the first commercially available embedded devices with open source CPU
Milkymist One in more detail

- **Milkymist One** is a product containing a FPGA as central component.
- The FPGA implements the **Milkymist System-on-Chip (SoC)** comprised of a CPU, I/O interfaces and accelerators.
- The **Flickernoise** video synthesis software runs on the SoC.
Milkymist One: everything on board for interactive installations

- Video input
- VGA output (1280x1024)
- DMX512
- MIDI
- Line audio and built-in microphone
- Ethernet (OSC, Twitter wall, updates, TODO: streaming...)
- Infrared
- USB
Milkymist SoC: an advanced open source system-on-chip

- LatticeMico32 32-bit RISC CPU core
- High performance DDR SDRAM controller
- All I/O peripherals for Milkymist One
- 2D texturing acceleration
- Floating point vertex shader
- Wishbone, FML and CSR interconnect
- Extensive use of on-chip DMA
Flickernoise VJ software: flexibility and ease of use

- Effects (called “patches”) are fully programmable
- Using the Flickernoise Patching (FNP) language
- Graphical user interface on Milkymist One
- Automatically share patches between Milkymist devices online
Select images to flash.
If your board does not restart after flashing, don’t panic!
Hold pushbutton #1 during power-up to enable rescue mode.

Bitstream image (.FPC):

BIOS image (.BIN):

Application Image (.FBI):

Program flash

Ready.

Flash upgrade

Milkymist One

August 2011 22 / 1

S. Bourdeauducq (Milkymist project)
3.6 Interacting: per-frame equations

To ensure compatibility with MilkDrop, Flickermosio also accepts per_frame.xx where xx can be any number.

then the zoom amount would oscillate between 0.9 and 1.1 over time. The equation says: "take the static value of zoom, then replace it with that value, plus some variation". This particular equation would oscillate (cycle) every 6.28 seconds, since the sin function's period is approximately 6.28 (≈ 2π) seconds.

The "time" parameter is a read-only variable that retrieves the amount of time, in seconds, since Flickermosio started generating the video effects.

If you wanted it to make the zoom cycle every 2 seconds, you could use:

```
per_frame=zoom = zoom + 0.1*sin(time*π)
```

Now, let's say you wanted to make the color of the waveform (sound wave) that gets plotted on the screen vary through time. The color is defined by three values, one for each of the main color components (red, green, and blue), each in the range 0 to 1 (0 is black, 1 is full intensity). You could use something like this:

```
per_frame=wave_r = 0.5 + 0.5*sin(time*π)
per_frame=wave_g = 0.5 + 0.5*sin(time*π)
per_frame=wave_b = 0.5 + 0.5*sin(time*π)
```

Recall from your geometry classes that sin) returns a value between -1 and 1.
Why open source hardware?
JTAG/Serial debugger for Milkymist One (Yanjun Luo, Michael Walle)
Linux port to the Milkymist platform (Takeshi Matsuya, Lars-Peter Clausen, Michael Walle)
OpenWrt port (Lars-Peter Clausen)
LLVM back-end (JP Bonn)
Parts of the RTEMS port to the Milkymist platform (Yann Sionneau)

- RTEMS is a light POSIX real-time operating system
- Ported during Google Summer of Code 2010
- Now flashed on production devices
On-chip GDB server (Michael Walle)

- Debug embedded software on Milkymist One with GDB
- As if it were running on your local PC
  - stack trace
  - breakpoints
  - single stepping
  - evaluations
  - etc.
QEMU emulation support for Milkymist hardware (Michael Walle)
Distribution maintainers, please make our life easier:

PACKAGE QEMU-SYSTEM-LM32!

It’s part of the recently released QEMU 0.15. If you are very nice, also package qemu-user-lm32.

THANKS!
Collaborative hardware debugging (Michael Walle, Werner Almesberger)
Case design (Joachim Steiger / Raumfahrtagentur)
Manufacturing (Sharism at Work Ltd: Wolfgang Spraul, Xiangfu Liu, Adam Wang, Yi Zhang)
Re-uses of Milkymist technology for scientific research

- NASA CoNNeCT experiment
  - Software defined radio
  - Milkymist SoC’s SDRAM controller
  - Scheduled for launch in January 2012
  - Will be put on board the ISS

- GSI/CERN particle accelerator control systems
  - On-chip GDB server
  - In development
  - www.ohwr.org
[your awesome hack here]

- IRC: #milkymist on Freenode
- Mailing list: devel@lists.milkymist.org
- Web: http://www.milkymist.org
- Twitter: @milkymistvj – Github: milkymist