

Building Sophisticated Robots with ViewPort and the Parallax Propeller

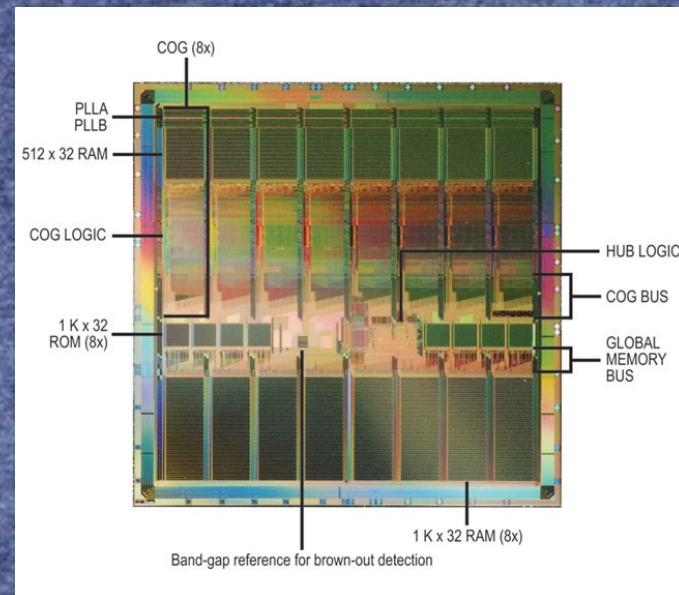
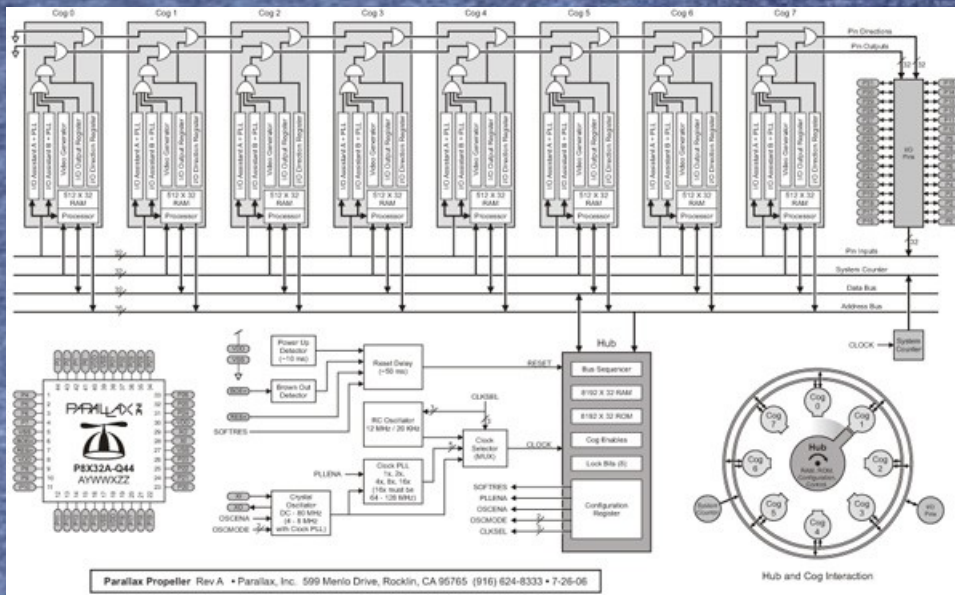
Hanno Sander
Silicon Valley HomeBrew Robotics Club
April 29th, 2009
<http://mydancebot.com>

Demos:

- DanceBot balancing and guided by vision
- Tune DanceBot with ViewPort
 - measure tilt, position, IO, NTSC
 - Tune fuzzy logic
- Debug Propeller Spin Code
- Find Objects with OpenCV
 - face, color, circle
- PropScope
- Simulated DanceBot

Parallax Propeller

- Eight 32 bit processors @80MHz for \$8
- Divide sophisticated projects into 1 cog/task.
- Spin high level object oriented language
- Asm fast enough to input/output video signals.



Demo #1: DanceBot



Sensors

Camera

Quadrature Encoder

Gyroscope
Accelerometer

Pre-Process

Location of Person

Position, Velocity

Tilt, Rate of Turn

Logic

Find person and
set target position

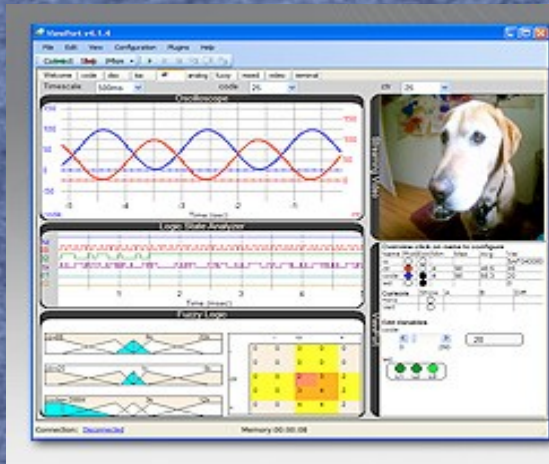
Fuzzy PID Loop to
balance and
achieve target
position

Actuators

Drive 2 Motors

Dancing with 8 Cores

ViewPort Debugger



Monitor/Change Variables
 Analyze Data
 OpenCV Vision Engine
 PhysX World Simulation
 Integrated Debugger

Propeller w 8 Cores DanceBot

Software Frame Grabber NTSC->ADC->4 Pins Memory
Image Processing Frame->Variables
Position Encode pulse length/phase
Tilt SPI Accelerometer Hobby Gyro
Kalman 50Hz update to eliminate drift
Fuzzy Control Balance and Keep position
PWM Proportional fwd/rev
Conduit Share Data

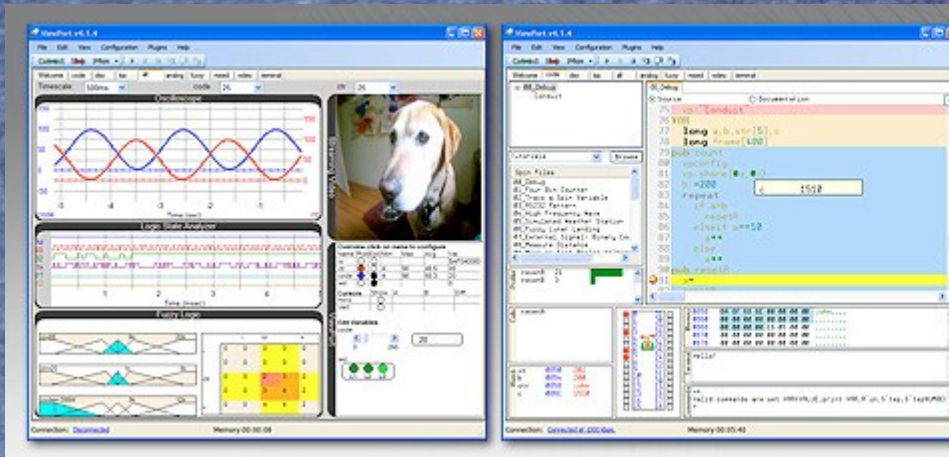


Gyro
 Accelerometer
 Encoder
 Camera

 Motors

Demo #2: ViewPort

- Debugging Tool
- Monitor/Change variables in live program
- Simulated Instruments: DSO, LSA, Spectrum...
- Parallax Product: Magazine Ad, Press Release



ViewPort Software

The premier debugging environment for the Propeller. The tool combines an integrated debugger with powerful graphics that show you what's going on within the chip!



ViewPort

Propeller
Shared Memory
and Input/Output

Your Program
on 7 Cogs

ViewPort shares
data using 1 Cog

Stream Data over USB at 2Mb/S

Change Data

The screenshot shows the ViewPort software interface. At the top, there are buttons for 'Connect', 'Stop', and 'USB0 - 2Mbps'. Below this is a menu bar with 'Welcome', 'analog', 'digi', 'isa', and 'mixed'. The main area is an 'Oscilloscope' window displaying three waveforms: a blue one at the top, a green one in the middle, and a red one at the bottom. The x-axis is labeled 'Time (msec)' and ranges from -20 to 20. The y-axis is labeled 'signal (v)' and ranges from -4 to 6.0. A cursor is positioned at approximately 5ms on the x-axis, with a tooltip showing '[P1 Time signal=0.24v delay=0.24v]'. To the right of the oscilloscope are several control panels. The top one is a large knob for 'Vertical' scale, with values from 10uV to 100mV. Below it are two smaller knobs for 'Horizontal' scale, with values from 20ns to 500ns. Further down are two more knobs for 'DC' and 'AC' coupling. At the bottom right is an 'Overview' table with columns for Name, Post, Min, Max, Avg, and Vol. The table contains data for 'signal', 'delsig', 'mode', 'fsc', and 'hsc'. Below the table are 'Cursors' and 'Measure' sections. At the bottom of the interface are several control buttons: 'Settings', 'Trigger', 'Cursor', and 'Measure'. There are also buttons for 'Horizontal', 'Vertical', and 'Float' with green indicator lights. A 'Zoom' slider is also present.

Custom Views

Intuitive Controls

View Data

Demo #3: Spin Code Debugger

- Spin is C++ like language: **repeat, if, outa, x:=2**
- Integrated ViewPort Debugger includes:
 - Breakpoint
 - Pause
 - 3 types of Step
 - Profiler
 - Interpreter
 - Call stack
 - View Memory
 - View IO States

The screenshot shows the ViewPort 4.1F BETA debugger interface. The main window displays the source code for a Spin program named '16_Text'. The code includes a 'repeat' loop and a 'strAdd' function. The debugger is currently paused at line 34, which is highlighted in yellow. A red line indicates the current execution point at line 36. The interface also shows a 'Watch' window with variables 'str', 'a', 'b', 'v1', 'v3', 'v4', 'v5', and 'c'. The 'Memory' window shows a memory dump starting at address 0AF0. The 'IO States' window shows the output of the program, including the prompt 'Type your name' and the user input 'Hanno'.

```

27 vpconfig
28 vp.share(@OutStr,@b)
29 a~
30 vp.txt(@OutStr,string("Type your name"))
31 repeat
32   b++
33   repeat while InStr[0]==0
34     a++
35     strAdd(@OutStr,string("Hello "),@InStr)
36     vp.txt(@OutStr,0)
37     InStr[0]:=0
38 pub strAdd(out,in1,in2)|len1,len2
39   len1:=strsize(in1)
40   bytemove(out,in1,len1)
41   len2:=strsize(in2)
42   bytemove(out+len1,in2,len2+1)
  
```

Memory dump:

0AF0	48 65 6C 6C 6F 20 48 61	Hello.Ha
0AF8	6E 6E 6F 00 60 66 00 00	nno.mf..
0B00	01 00 00 00 00 00 00 00
0B08	6F 00 00 00 00 00 00 00	o.....
0B10	00 00 00 00 00 00 00 00
0B18	74 20 09 00 02 00 00 00	t-.....
0B20	00 00 00 00 00 00 00 00
0B28	00 00 00 00 00 00 00 00
0B30	00 00 00 00 00 00 00 00

IO States:

```

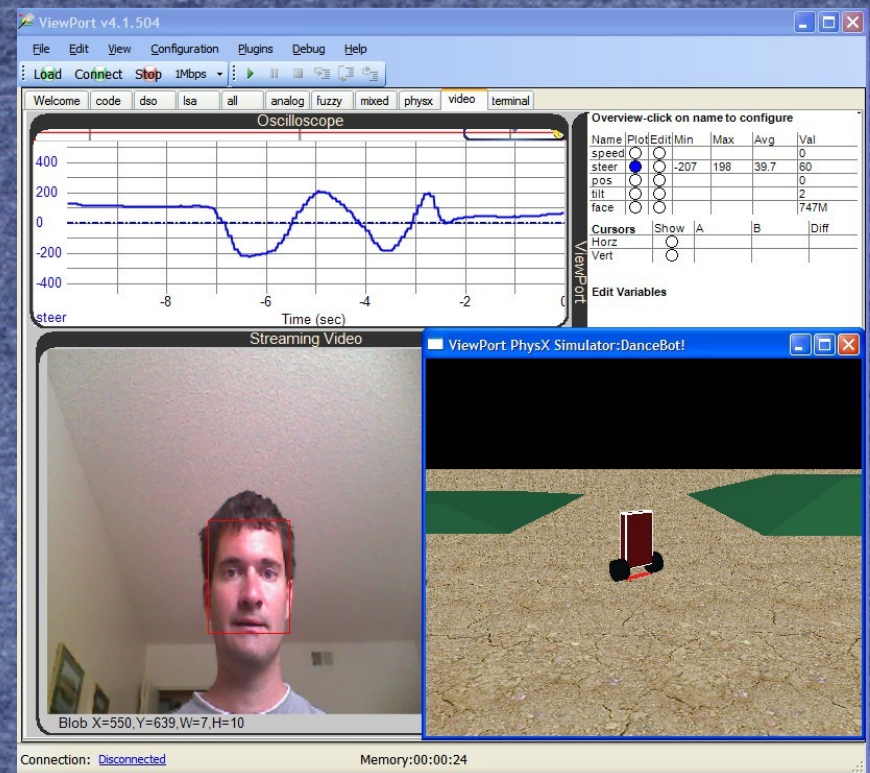
Type your name
>Hanno
Hello Hanno
  
```

Demo #4: OpenCV

- OpenCV is computer vision library maintained by Willow Garage (funded by Google)
- Find objects: face, color, circle
- Process object location with Spin Code on Propeller

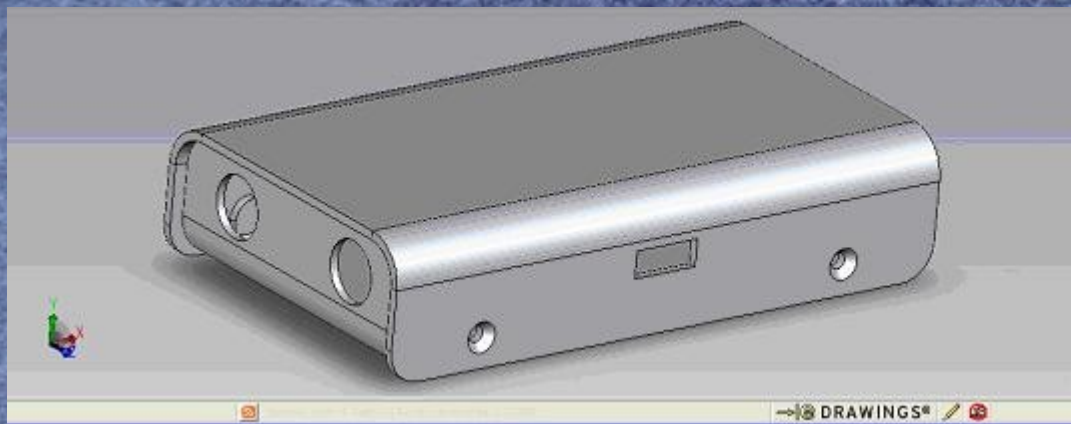
Demo #5: PhysX

- Nvidia PhysX: GPU-accelerated physics simulator: Friction, gravity, joints...
- Simulated DanceBot, controlled by code on Propeller
- Steered by Face location as found by OpenCV
- Inspired by recent IEEE talk



Demo #6: PropScope

- Multi-function USB Oscilloscope/Function Generator/Logic Analyzer for \$199
- Retail Product by Parallax
- Powered by ViewPort library



New Zealand

- Discovered on honeymoon
- Beautiful country
- Wonderful people
- Tech Culture
- Moved from Silicon Valley
3½ years ago:
 - +quality of life,
 - +raise kids
- Big focus on High Tech:
“Silicon Plains”



“From Sheep to Chips”

