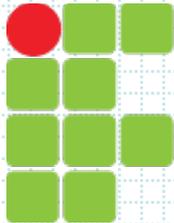


Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina

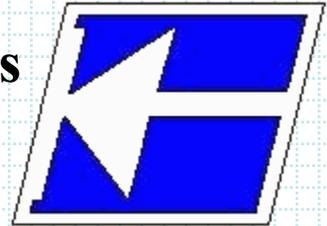
Departamento Acadêmico de Eletrônica

Pós-Graduação em Desen. de Produtos Eletrônicos

Conversores Estáticos e Fontes Chaveadas



INSTITUTO FEDERAL
SANTA CATARINA



Softwares para Eletrônica de Potência

Prof. Clóvis Antônio Petry.

Prof. Joabel Moia.

Florianópolis, fevereiro de 2014.

Bibliografia para esta aula



INSTITUTO FEDERAL
SANTA CATARINA



www.ProfessorPetry.com.br



Disciplina

Plano de Ensino (2009/1)

Base o plano de ensino da disciplina:

Data das avaliações

- Primeira avaliação =
- Segunda avaliação =
- Terceira avaliação =
- Quarta avaliação =
- Seminário =

Notas da disciplina

Divida, entre em contato: petry@ifsc.edu.br

Aulas	Notas de Aula	Apresentações	Complementos	Listas de exercícios
00		Apresentação da disciplina		
01				
02				
03				
04				
05				
06				
07				
08				
09				
10				
11				
12				
13				
14				
15				
16				
17				

Avaliações anteriores

Softwares para eletrônica de potência:

1. Simuladores de circuitos:

- Orcad/pspice;
- Proteus;
- Tina;
- Multisim;
- Psim.

2. Planilhas de cálculo:

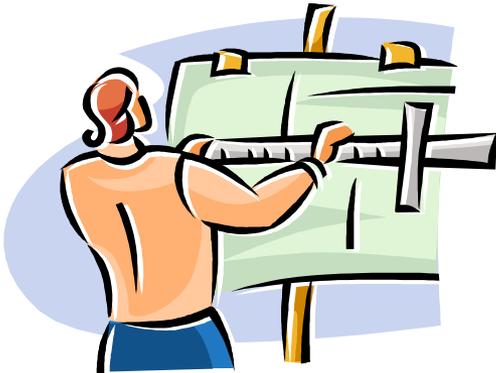
- Matlab;
- Scilab;
- Mathcad;
- Smath;
- Mathematica.

3. Ambientes de desenvolvimento:

- LabView;
- Arduino.

Razões para usar softwares específicos:

- a) Complexidade dos circuitos eletrônicos;
- b) Dificuldade de representar o mundo real;
- c) Possibilidade de uso inúmeras vezes;
- d) Diminuição do custo de projeto;
- e) Aprendizagem via software;
- f) Outras...

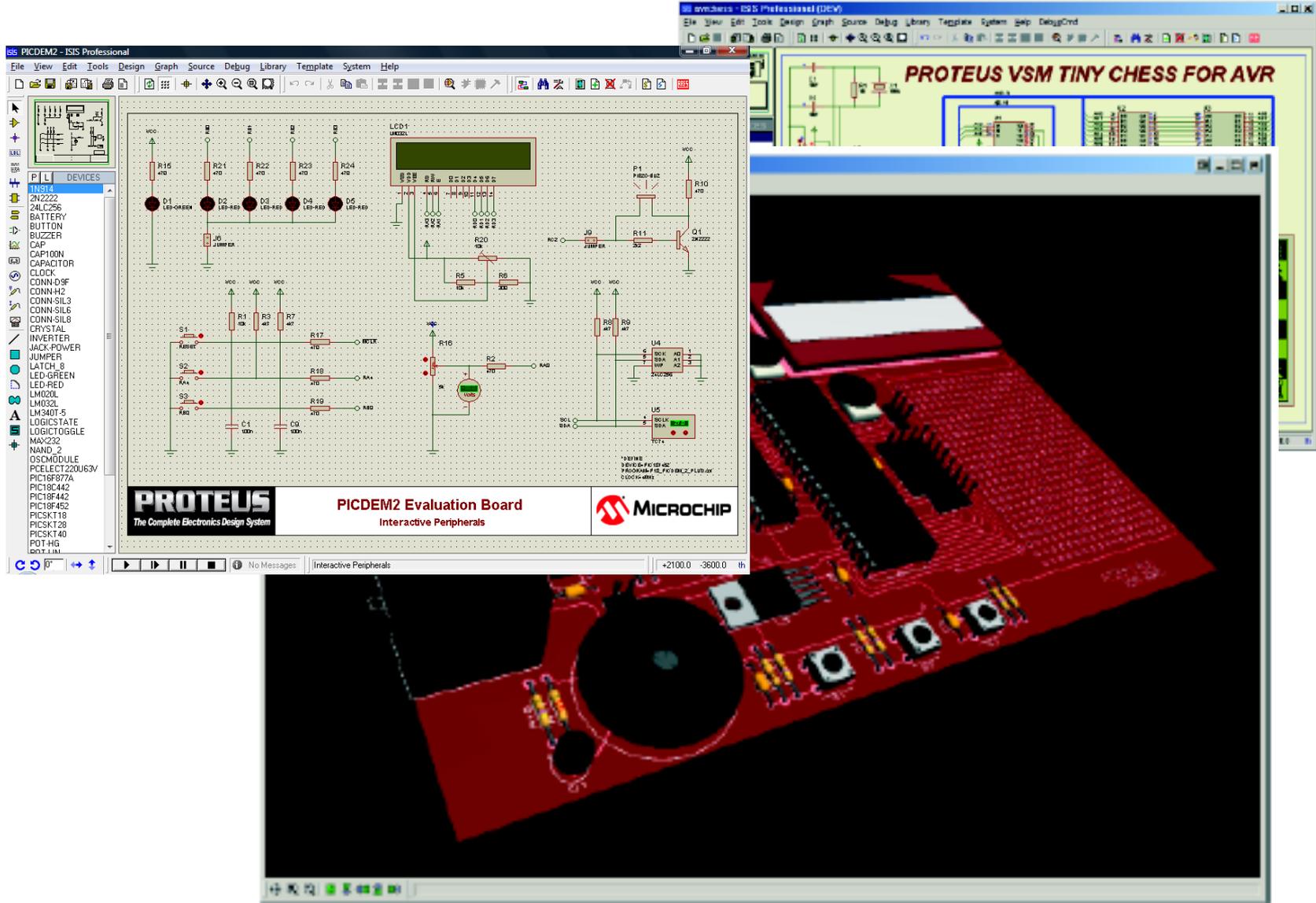


Qual a necessidade de se realizar uma simulação computacional:

- Averiguação do funcionamento do circuito projetado
 - Comprovação com a teoria
 - Evitar gastos desnecessários
 - Tempo
 - Compra de componentes
- Segurança
 - Ex: Aviação, Aeroespacial

Como funcionam os simuladores de circuitos:

- Os componentes elétricos/eletrônicos são modelados matematicamente
 - Ideal
 - Considerando os parâmetros do componente



Invert Gain OPA350 Test Circuit Design - Schematic Editor

File Edit Insert View Analysis Interactive I&M Tools Help

Basic Switches Meters Sources Semiconductors Optoelectronic Spice Macros Gates Flip-flops Logic ICs-MC

Sample Circuit Using the O

Structure

Digital Multimeter

Range: Auto

Function: V, ~V, Freq, I, ~I, Ohm

Input: HI: Out+, LO: Out-

542.82mV

VF1

U1 OPA350

V1 5

Vref 1.5

R4 1k

Invert Gain OPA350 Test Circuit Design

Exit

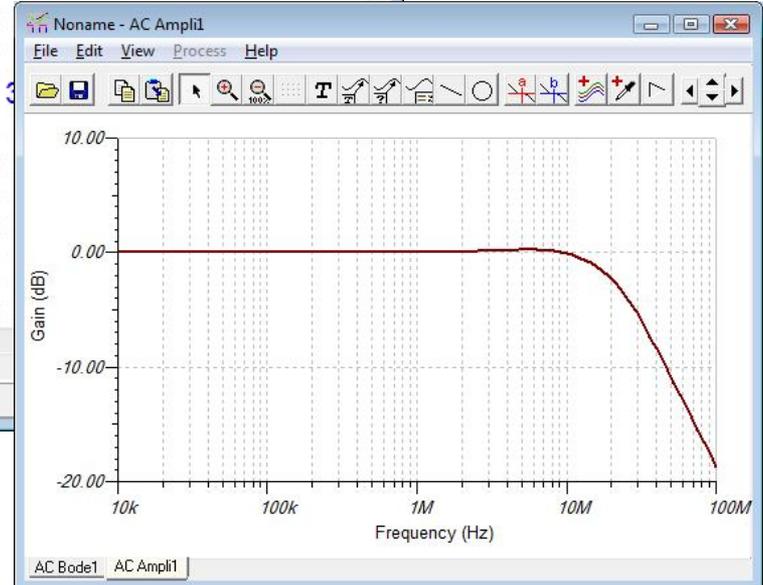
PCB Viewer - EXAMPLESPCBPIC Flasher DIP4SW flex top finished.TPC

File View Options...

Camera position:-0,83,-25,05,19,47 Origo position:-0,83,-5

PCB Viewer - EXAMPLESPCBPICFlasher pro.amp finished.TPC

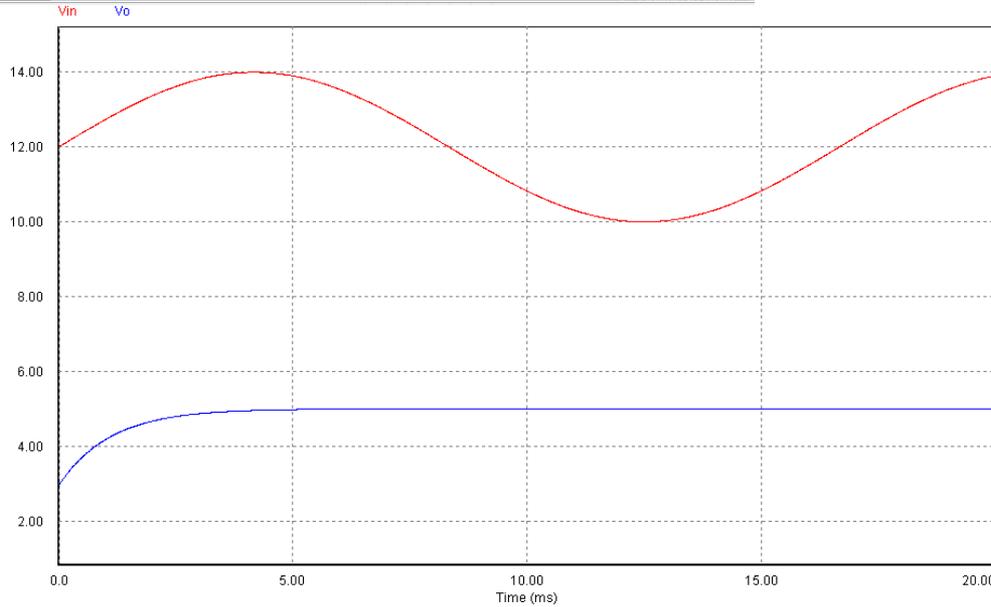
File View Options...

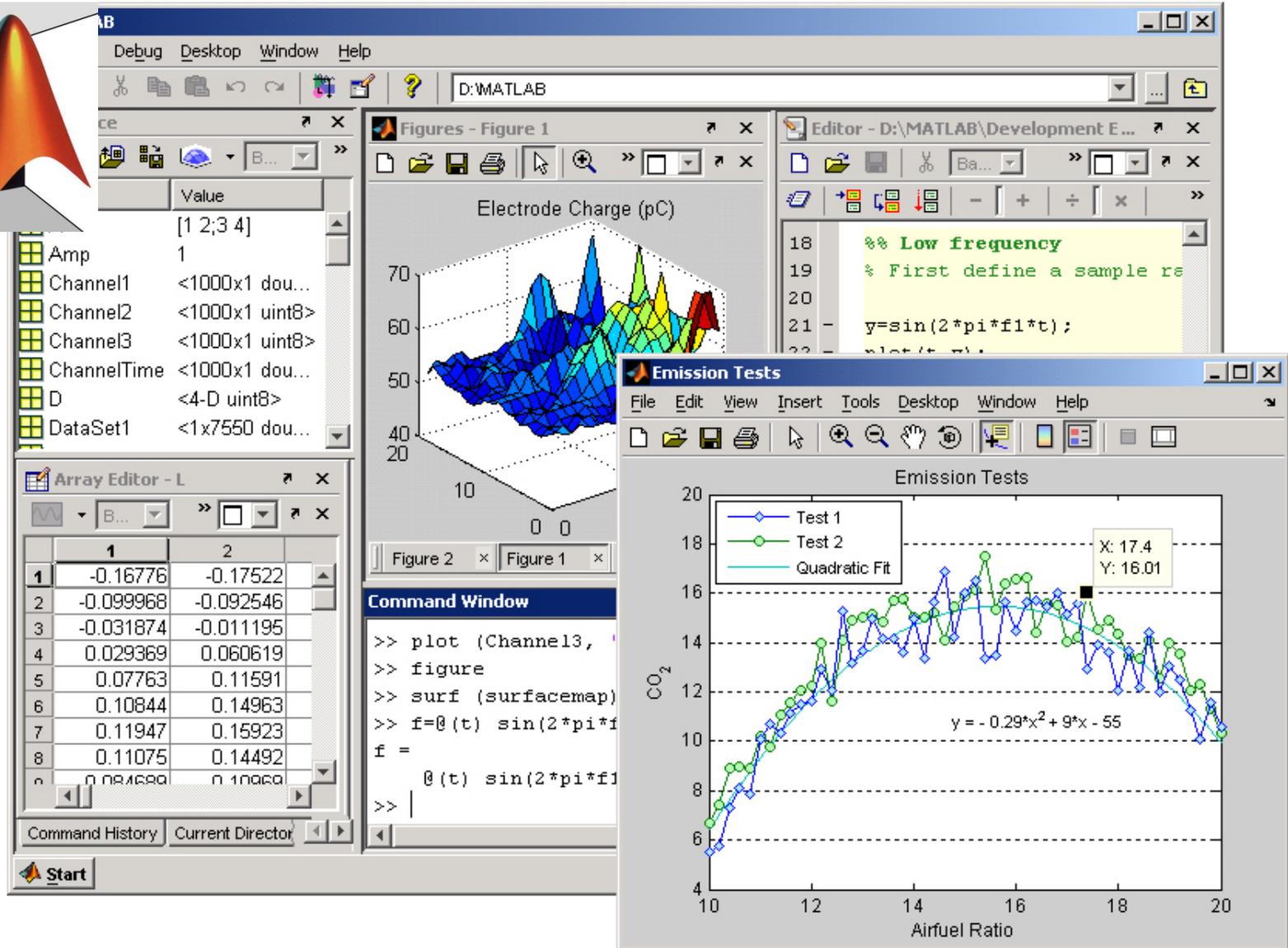
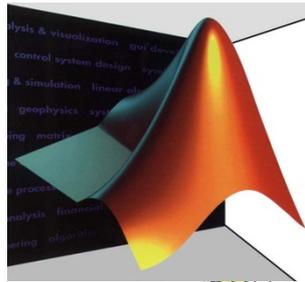


The image displays various components of the Multisim software interface:

- Schematic Editor:** Shows a circuit diagram with components U28A (74HC139D_4V) and U27 (1V). A component selection dialog is open, listing various 74HC logic chips.
- 3D PCB Model:** A 3D rendering of a green printed circuit board with various components mounted on it.
- Logic Analyzer:** A window showing a complex logic diagram with multiple signals and gates.
- Spreadsheet View:** A table listing component details for U28, U27, U26, and U25.
- Virtual Oscilloscope:** A digital oscilloscope window showing two waveforms (CH1 and CH2) on a grid. The settings are CH1: 20mV, CH2: 500mV, M: 500us, and CH2: 1.82V, 999Hz.

RefDes	Sheet	Section	Section Name	Fam...	Value	Footprint	Manufacturer	Description
VDD1	CO...			PO...	5V			
VDD1	CO...			PO...	5V			
U28	CO...	A	A	74H...	74H...	DW016		Number=2;Package=DW016;
U27	CO...	B	B	74H...	74H...	DW016		Number=2;Package=DW016;
U26	CO...			LIN...	MA...	DIP-18	Maxim	Number=4;Package=DIP-18;
U25	CO...			LIN...	DS...	SON8(C...		





The screenshot displays the MATLAB environment with several windows open:

- Editor:** Contains MATLAB code for a sine wave plot:

```
18 %% Low frequency
19 % First define a sample rate
20
21 y=sin(2*pi*f1*t);
22 plot(t, y);
```
- Figures - Figure 1:** A 3D surface plot titled "Electrode Charge (pC)".
- Command Window:** Shows the execution of the following commands:

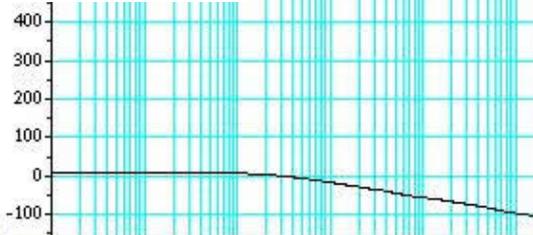
```
>> plot (Channel3,
>> figure
>> surf (surfacemap)
>> f=@(t) sin(2*pi*f
f =
    @(t) sin(2*pi*f1
```
- Array Editor - L:** A table with 2 columns and 8 rows of numerical data:

	1	2
1	-0.16776	-0.17522
2	-0.099968	-0.092546
3	-0.031874	-0.011195
4	0.029369	0.060619
5	0.07763	0.11591
6	0.10844	0.14963
7	0.11947	0.15923
8	0.11075	0.14492
- Emission Tests:** A 2D line plot showing CO₂ concentration versus Airfuel Ratio. It includes two data series (Test 1 and Test 2) and a quadratic fit curve. The fit equation is $y = -0.29x^2 + 9x - 55$. A specific point is highlighted with X: 17.4 and Y: 16.01.



InZoom 3D Rot.

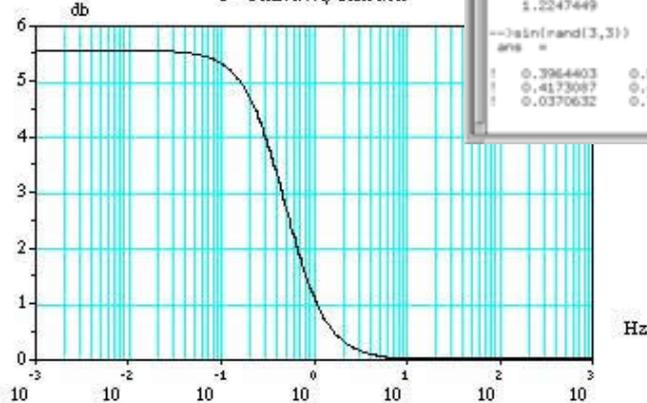
Elementary Sensitivity function



ScilabGraphic1

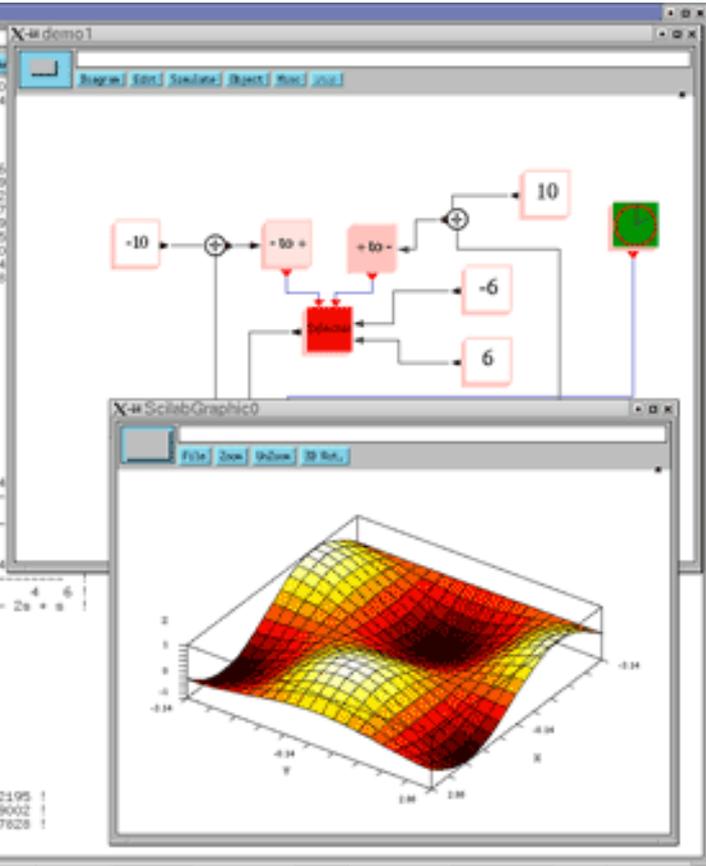
File Zoom UnZoom 3D Rot.

db S = Sensitivity function



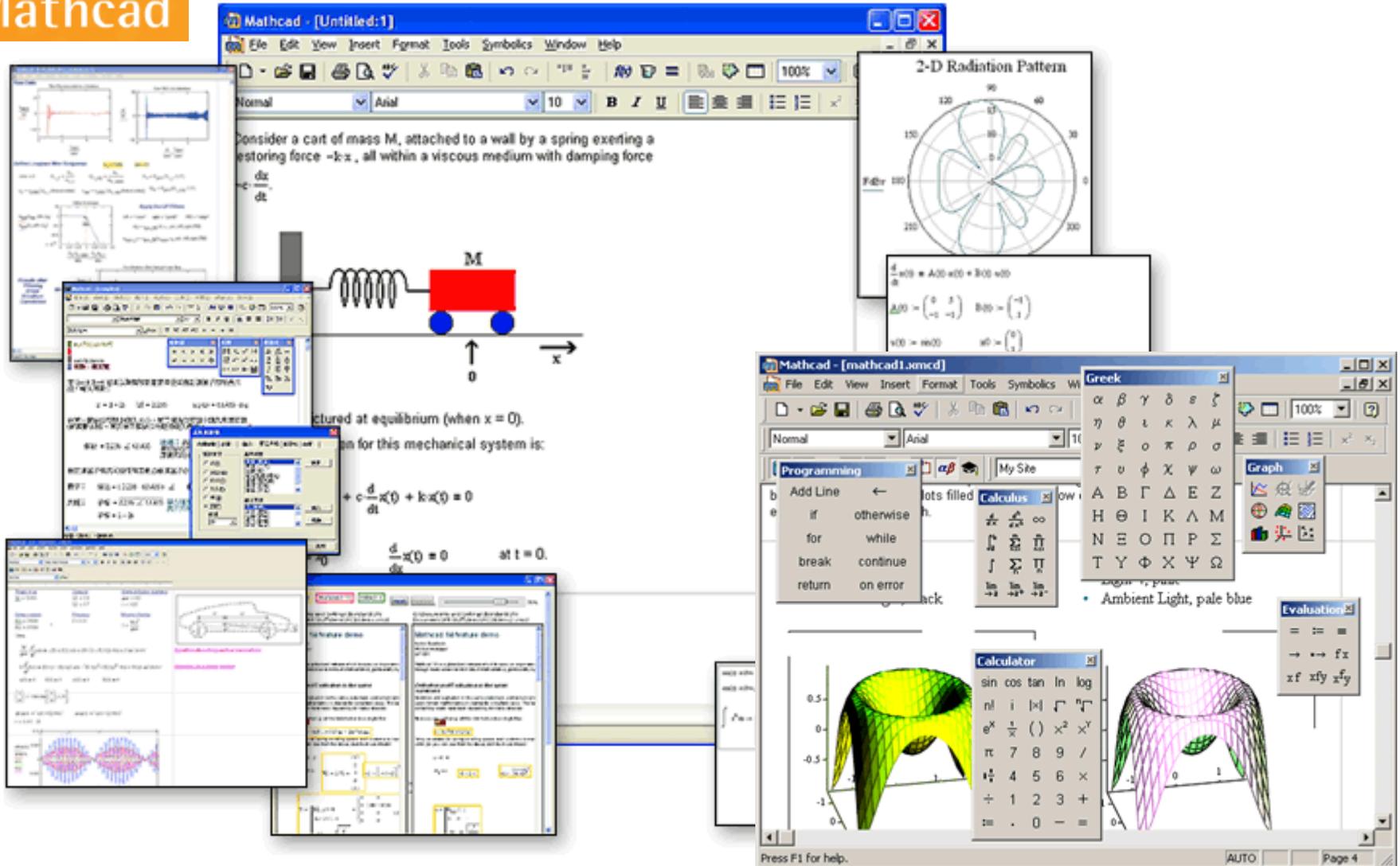
```

scilab-2.7
File Control Devise Graphical Window 2000 Help
1 1.8132668 - 0.5193587 1.1210
2 0.0066408 - 0.4235322 0.5904
-->lin(rand(9,9))
ans =
1 2.7617992 0.8036397 - 3.0656
2 1.3766708 0.7318738 - 2.4869
3 - 3.1347005 - 0.2880930 - 4.3802
4 5.2958496 2.9417706 - 14.667
5 - 2.3095349 - 2.3060676 6.6899
6 - 5.6858408 - 3.0964923 13.495
7 10.600871 3.3874405 - 28.030
8 - 9.6004909 - 3.5149264 27.604
9 0.0411598 0.6686495 - 2.6398
-->sp(1+2*s+3*s^2)/(2-5*s+3*s^3)
p =
1 + 2s - 2
-----
2 - s + 3
-->res(p*p^2;-p*p^2)
n =
1 + 2s - 2      1 + 4s + 2s - 4
-----      -----
2 - s + 3      4 - 4s + s + 4s -
- 1 - 2s + s      1 + 4s + 2s - 4
-----      -----
2 - s + 3      4 - 4s + s + 4s -
-->function r=f(a,b)
-->res(a)+cos(b)
-->endfunction
-->f(pi/12,-pi/12)
ans =
1.2247449
-->sin(rand(3,3))
ans =
1 0.3964403 0.5975281 0.0002195
2 0.4173087 0.4054269 0.4869002
3 0.0370632 0.7444363 0.4577828
    
```



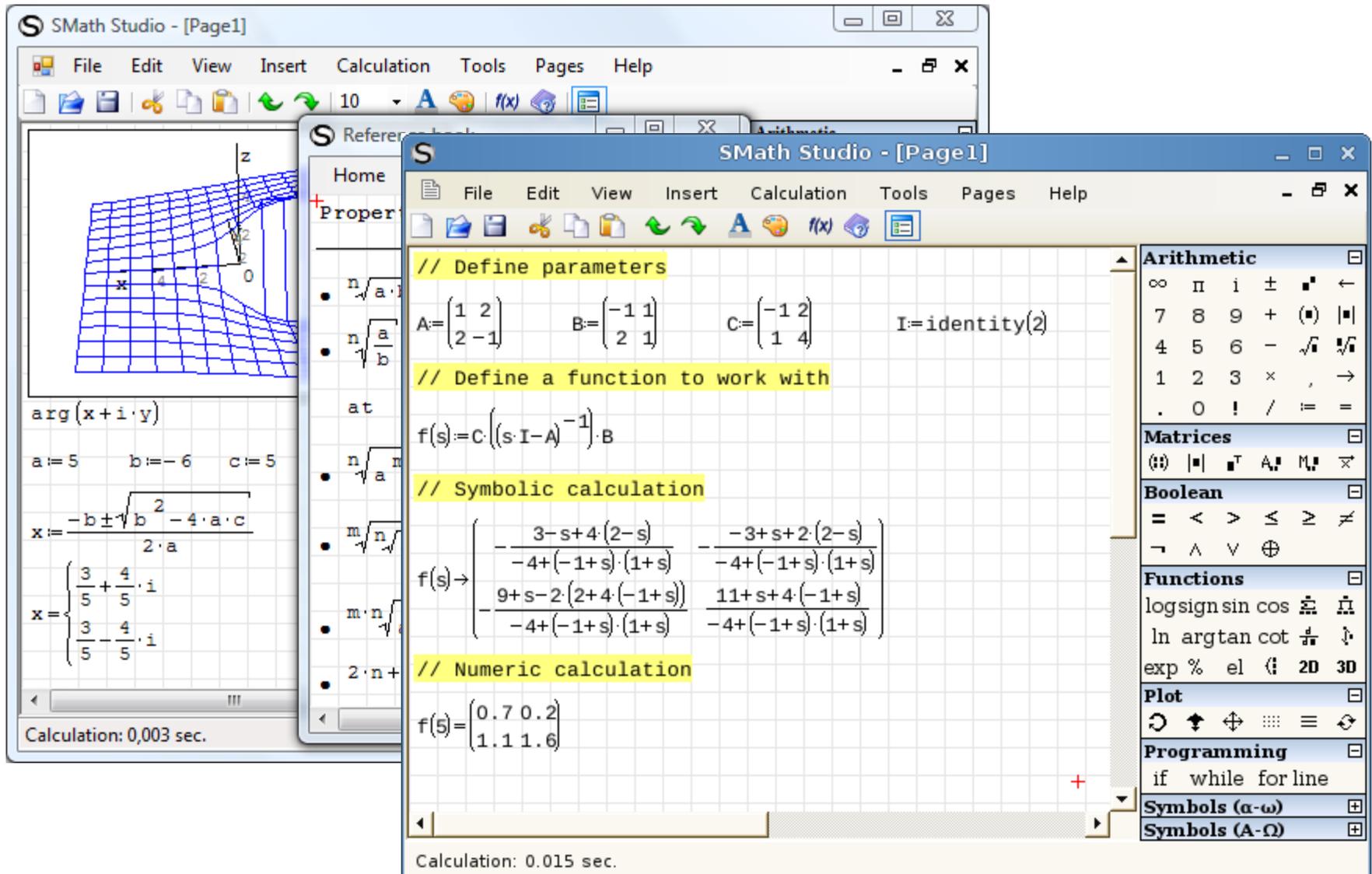
Mathcad

Mathcad



The image displays several overlapping windows from the Mathcad software interface:

- Mathcad - [Untitled:1]**: Shows a text description of a mass-spring-damper system: "Consider a cart of mass M , attached to a wall by a spring exerting a restoring force $-kx$, all within a viscous medium with damping force $-c \frac{dx}{dt}$ ". Below the text is a diagram of a red cart of mass M on wheels, connected to a wall by a spring. The equilibrium position is marked as $x=0$.
- 2-D Radiation Pattern**: A polar plot showing a complex radiation pattern on a circular grid with angles from 0 to 360 degrees.
- Mathcad - [mathcd1.mcd]**: Shows a Greek alphabet selection window with characters like $\alpha, \beta, \gamma, \delta, \epsilon, \zeta, \eta, \theta, \iota, \kappa, \lambda, \mu, \nu, \xi, \omicron, \pi, \rho, \sigma, \tau, \upsilon, \phi, \chi, \psi, \omega, \Lambda, \Gamma, \Delta, E, Z, H, \Theta, I, K, \Lambda, M, N, \Xi, O, \Pi, P, \Sigma, T, Y, \Phi, X, \Psi, \Omega$.
- Programming**: A menu for code blocks including "Add Line", "if", "otherwise", "for", "while", "break", "continue", "return", and "on error".
- Calculus**: A menu for mathematical operations like integration (\int), differentiation ($\frac{d}{dx}$), and limits (\lim).
- Calculator**: A standard scientific calculator window with buttons for trigonometric functions, logarithms, and arithmetic.
- Graph**: A window for plotting data and functions.
- Evaluation**: A window for performing calculations and displaying results.
- 3D Surface Plots**: Two 3D surface plots, one colored green and yellow, and another colored pink and purple, showing complex mathematical surfaces.



The image displays two overlapping windows of the Smath Studio software. The background window shows a 3D plot of a surface in a coordinate system with x, y, and z axes. The foreground window is a code editor titled "Smath Studio - [Page1]" containing the following code and results:

```
// Define parameters
A = [1 2; 2 -1]    B = [-1 1; 2 1]    C = [-1 2; 1 4]    I = identity(2)

// Define a function to work with
f(s) = C * (s * I - A)^-1 * B

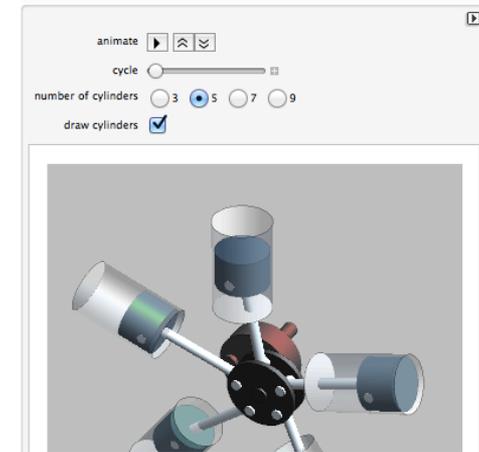
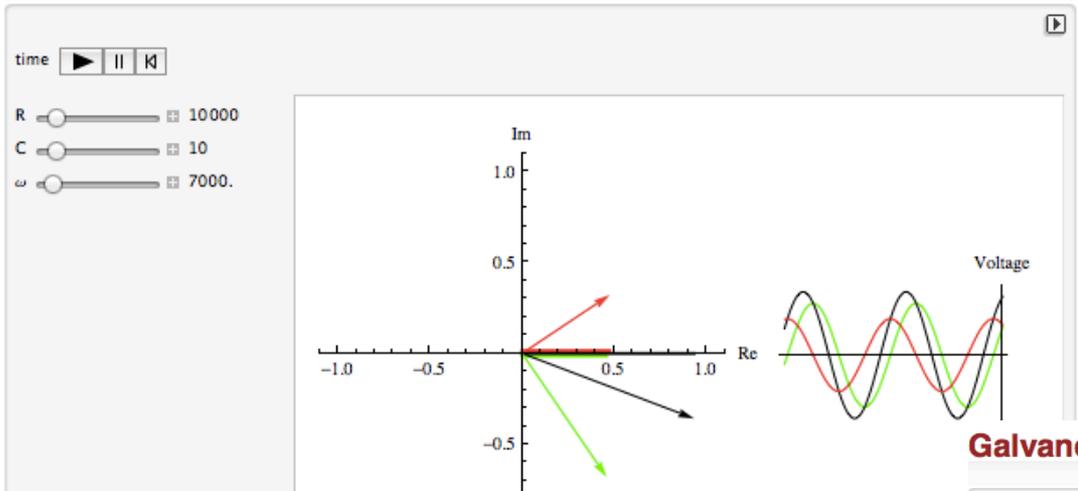
// Symbolic calculation
f(s) -> [ -3-s+4*(2-s) / -4+(-1+s)*(1+s)  -3+s+2*(2-s) / -4+(-1+s)*(1+s) ;
          9+s-2*(2+4*(-1+s)) / -4+(-1+s)*(1+s)  11+s+4*(-1+s) / -4+(-1+s)*(1+s) ]

// Numeric calculation
f(5) = [0.7 0.2; 1.1 1.6]
```

At the bottom of the code editor, it says "Calculation: 0.015 sec." To the right of the code editor is a sidebar with various mathematical symbols and functions categorized into Arithmetic, Matrices, Boolean, Functions, Plot, Programming, Symbols (α-ω), and Symbols (A-Ω).

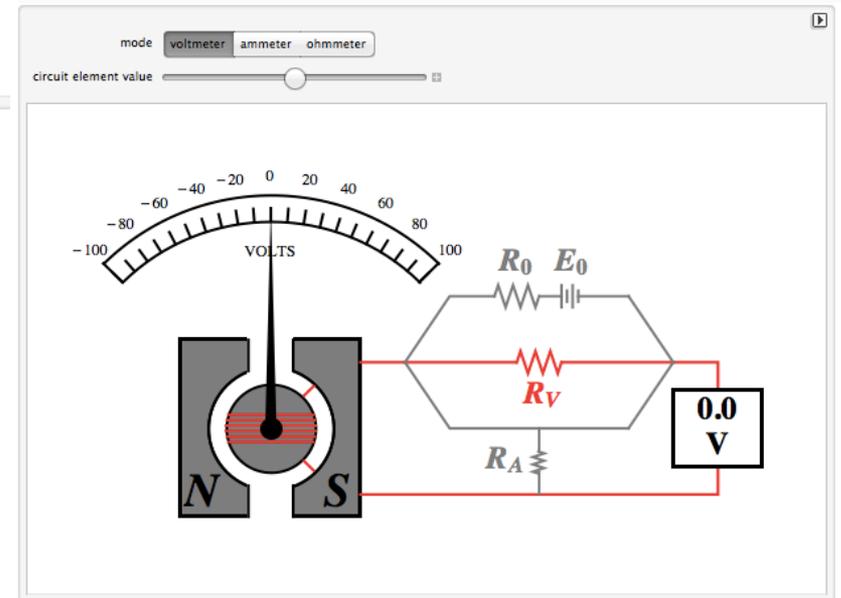
The background window also shows a calculation: $\arg(x+iy)$ with $a=5$, $b=-6$, $c=5$. The result is $x = \frac{-b \pm \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}$, which simplifies to $x = \begin{cases} \frac{3}{5} + \frac{4}{5} \cdot i \\ \frac{3}{5} - \frac{4}{5} \cdot i \end{cases}$. At the bottom, it says "Calculation: 0,003 sec."

Phasor Model for RC Filter Electronic Circuit

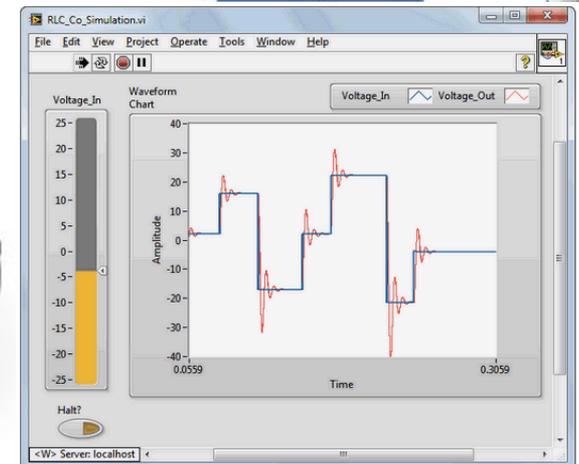
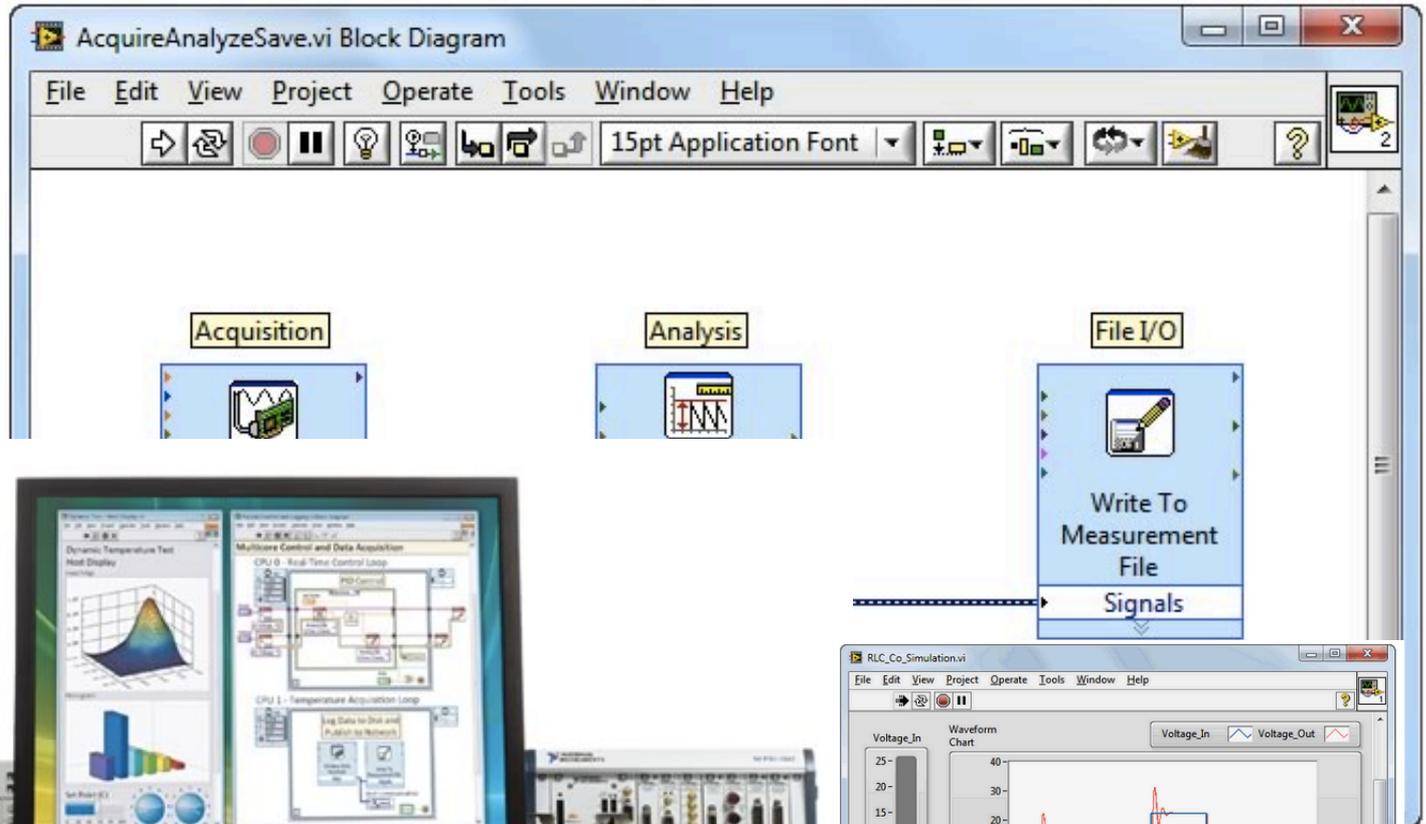
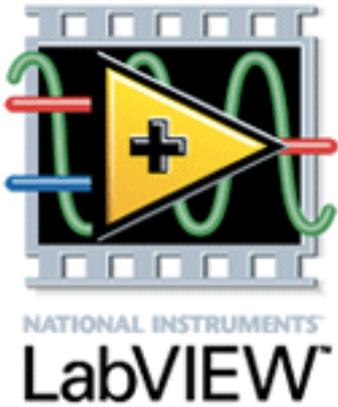


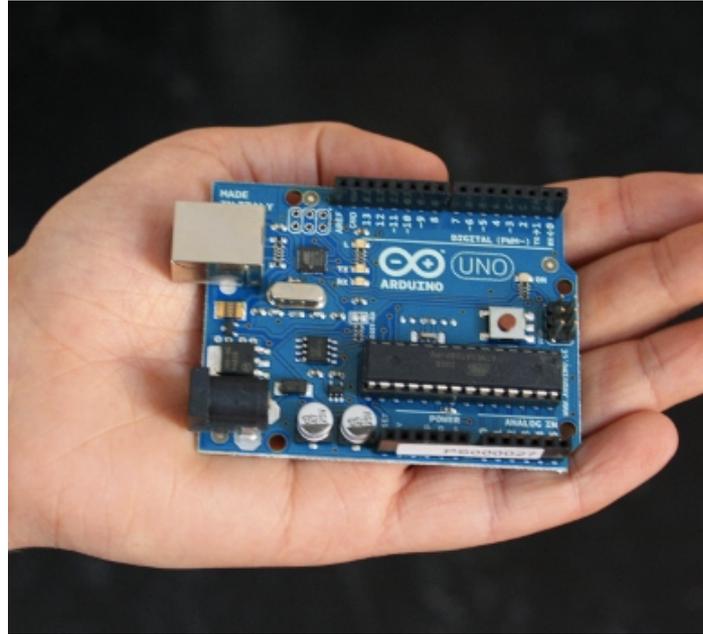
Galvanometer as a DC Multimeter

```
Manipulate[
  Grid[{{
    Show[
      Graphics[
        {
          {Thickness[0.0075], RGBColor[0, 1, 0],
            Line[{{0, -0.015}, {
               $\frac{1}{\sqrt{(\omega R C c \cdot 10^{-9})^2 + 1}}$ 
              Cos[ $\omega t - \frac{\pi}{2}$ ], -0.015}}]}],
          {Thickness[0.0075], RGBColor[1, 0, 0],
            Line[{{0, 0.015}, {
               $\frac{\omega R C c \cdot 10^{-9}}{\sqrt{(\omega R C c \cdot 10^{-9})^2 + 1}}$ 
              Cos[ $\omega t$ ], 0.015}}]}],
          {Thickness[0.0075], Line[{{0, 0}, {
              Cos[ $\omega t - \text{ArcTan}[\frac{1}{\omega R C c \cdot 10^{-9}}]$ ], 0}}]}],
          {RGBColor[1, 0, 0],
            Arrow[{{0, 0}, {
               $\frac{\omega R C c \cdot 10^{-9}}{\sqrt{(\omega R C c \cdot 10^{-9})^2 + 1}}$ 
              Cos[ $\omega t$ ],  $\frac{\omega R C c \cdot 10^{-9}}{\sqrt{(\omega R C c \cdot 10^{-9})^2 + 1}}$ 
              Sin[ $\omega t$ ]}]}]}],
        }
      ]
    }
  ]
]
```



LabVIEW



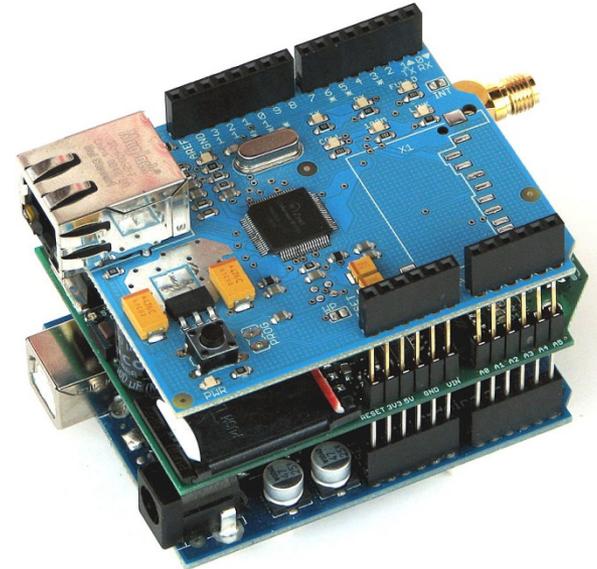
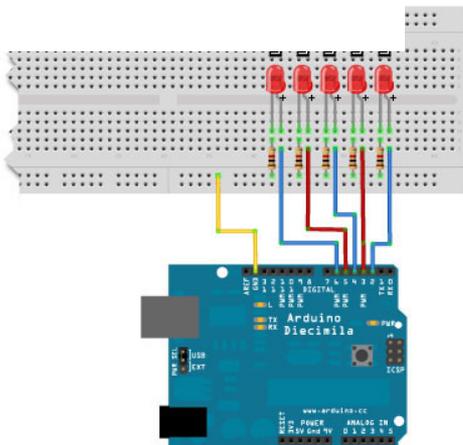


```
Arduino - 0011 Alpha
File Edit Sketch Tools Help
Blink
/*
 * Blink
 *
 * The basic Arduino example. Turns on an LED on for one second,
 * then off for one second, and so on... We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

int ledPin = 13;           // LED connected to digital pin 13

void setup()              // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop()               // run over and over again
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);                // waits for a second
  digitalWrite(ledPin, LOW);  // sets the LED off
  delay(1000);                // waits for a second
}
```



Introdução à Fontes Chaveadas:

1. Revisão de reguladores lineares;
2. Fontes lineares x fontes chaveadas;
3. Fontes chaveadas;
4. Circuitos elétricos de fontes chaveadas;
5. Tecnologias de fontes chaveadas;
6. Projetos de fontes chaveadas;
7. Principais estágios de uma fonte chaveada.

