

# Física com Arduino para iniciantes

(Physics with Arduino for beginners)

Marisa Almeida Cavalcante<sup>1</sup>, Cristiane Rodrigues Caetano Tavoraro<sup>1</sup> e Elio Molisani<sup>2</sup>

<sup>1</sup>Grupo de Pesquisa em Ensino de Física, Pontifícia Universidade Católica, São Paulo, SP, Brasil

<sup>2</sup>Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, Brasil

Recebido em 13/5/2011; Aceito em 21/10/2011; Publicado em 5/12/2011

Apresentamos neste trabalho diferentes modos de operar o Arduino para funcionar como uma interface alternativa na aquisição e automação de dados em atividades experimentais de física via porta USB do computador. Selecionamos como exemplo de aplicação o estudo de carga e descarga de um capacitor. As etapas de interação com o Arduino passam pelo processo de construção do circuito, coleta e armazenagem de dados em formato txt e visualização gráfica em tempo real através da linguagem *Processing*. Para cada etapa apresentamos propostas didáticas de utilização, todos os códigos fontes necessários para a interação com o Arduino, além de links para acesso a tutoriais que possibilitam a reprodução deste e outros experimentos.

**Palavras-chave:** Arduino, processing, aquisição de dados.

In this paper we present different ways of operating the Arduino to work as an alternative interface for the automation of data acquisition in physics experiments through the computer's USB port. Selected as an example of applying the study of charging and discharging a capacitor. The steps of interaction with the Arduino go through the process of building the circuit, collecting and storing data in txt format and graphic display in real time over the language *Processing*. For each step we present proposals for didactic use, all source code necessary to interact with the Arduino, and links to access tutorials that provide playback of this and other experiments.

**Keywords:** Arduino, processing, acquisition of data.

## 1. Introdução

É inegável que o computador é uma importante ferramenta cognitiva [1], isto é, permite ao estudante desenvolver habilidades, interiorizar conhecimentos e organizá-los de modo a construir uma interpretação do mundo que o cerca. Podemos exemplificar sua interatividade mediante problemas-jogo, tais como os applets, onde o estudante modifica parâmetros e verifica os resultados obtidos, e dessa forma o computador é uma ferramenta que permite organizar e sistematizar informações [2,3]. Segundo Barroso [3] estes objetos educacionais podem ser utilizados tanto presencialmente quanto à distância.

Outro exemplo é a modelagem computacional. O software Modellus [4,5] permite que o estudante investigue um fenômeno construindo uma animação a partir de objetos abstratos da matemática, como equações, funções, vetores e relações geométricas de um dado modelo de fenômeno físico. Ao construir seu próprio modelo e as formas de representá-lo o estudante torna-se um agente da construção do conhecimento, interpretando e internalizando a linguagem simbólica, tão im-

portante na física.

A utilização do computador como agente transformador no ensino de física o ensino e aprendizagem da física, buscando diminuir esta distância tecnológica entre a escola e o estudante tem sido incansavelmente relatada também como instrumento de laboratório [6-17] em que computadores são conectados a sensores permitindo fazer experimentos que dificilmente seriam realizados com os instrumentos usuais de um laboratório de ensino, podendo apresentar os resultados quase imediatamente.

A introdução da metodologia experimental de aquisição de dados por computador representa a possibilidade real de uso das técnicas de análise estatística de dados experimentais estudadas no curso de física e engenharias do primeiro período da graduação. Além da melhoria da precisão dos resultados, a redução no tempo de coleta de dados e a rápida representação dos mesmos em forma de gráficos, permitem criar no laboratório de física um ambiente de construção do conhecimento físico. O estudante pode observar o fenômeno, prever o resultado, isto é, formular hipóteses, rapidamente comparar os resultados obtidos com os previstos

<sup>1</sup>E-mail: marisac@puccsp.br.

pelo modelo teórico, explicar possíveis diferenças entre o previsto e o observado e ainda, reformular suas hipóteses, fazer ajustes experimentais e testá-las novamente. O dinamismo desse processo provoca a curiosidade e maior interesse dos estudantes já que a aula de laboratório torna-se desafiadora: as antigas justificativas encontradas nas conclusões dos relatórios não são mais aceitas. Justificativas para um resultado ruim tais como: “foram obtidos poucos dados” ou, “o instrumento de medida não estava devidamente calibrado”, sequer podem ser cogitadas.

Contudo, apesar das suas potencialidades, o computador ainda é pouco utilizado em laboratórios de física, quer pela falta de informação dos professores sobre os recursos oferecidos, quer pela dificuldade em adquirir interfaces e programas de aquisição [7], devido ao elevado custo e também por serem sistemas fechados que impossibilitam um estudo mais detalhado de suas características e modificações que permitam adaptações para outros experimentos.

Ao analisar as últimas publicações em revistas de ensino de física nacionais, relacionadas à aquisição de dados via computador, notamos algumas tendências todas elas buscando desenvolver sistemas de aquisição de dados de baixo custo e totalmente livres.

Algumas delas [7-13] utilizam-se fotossensores e microfones conectados diretamente na entrada de microfone do PC, de modo que impulsos provenientes correspondentes à coleta de dados são lidos através de versões shareware ou freeware de programas de análise sonora disponíveis na Internet.

Nesses softwares, os impulsos são traduzidos em intensidade medida em decibéis em função do tempo de modo que os intervalos de tempo de passagem de objetos interrompendo a luz incidente nos fotossensores podem ser medidos diretamente na tela do microcomputador o que permite estudar tanto características cinemáticas, tais como velocidade e aceleração de objetos, quanto obter informações sobre a quantidade de movimento. Por outro lado, informações associadas às intensidades dos impulsos, como por exemplo, medidas de temperatura, pressão, intensidade de luz, etc., não podem ser avaliadas, inviabilizando um grande número de experimentos.

Numa outra tendência [14-17], estes sensores podem ser conectados as portas serial, paralela e USB do microcomputador, dependendo do que se pretende medir e do aparato experimental, se a medida é analógica ou digital. O principal motivo para se usar esta porta para coleta de dados em um laboratório didático é a segurança e facilidade com que podemos conectar sensores, que devem ter como característica a variação da resistência em função da grandeza física que se pretende medir. Os softwares necessários para a coleta e análise de dados, estão disponíveis na Internet (shareware ou freeware), porém, exigem que o usuário domine linguagem de programação. Alguns exemplos são

linguagens de programação como Pascal, VisualBasic, Delphi, C++, LOGO e planilhas eletrônicas. Além das dificuldades com a programação, o usuário precisa construir o circuito de aquisição [17].

Nesta linha, mas minimizando as dificuldades com a construção do circuito e programação, a placa Arduino vem sendo utilizada com muito sucesso [18,19]. A plataforma para o desenvolvimento dos programas de controle está disponível na *Internet* e existem diferentes versões do circuito no mercado nacional por preços acessíveis quando comparados às interfaces disponibilizadas no mercado por empresas como CIDEPE, Pasco Scientific e Phywe [20-22], por exemplo.

Neste trabalho mostramos mais uma aplicação do Arduino num experimento de física: carga e descarga de capacitores. Apresentamos desde o acoplamento do circuito RC à placa, os códigos de programação desenvolvidos para esta aquisição específica até o tratamento de dados além disso mostramos como proceder à visualização dos resultados em tempo real.

### 1.1. A interface Arduino

O Arduino [23] é uma plataforma que foi construída para promover a interação física entre o ambiente e o computador utilizando dispositivos eletrônicos de forma simples e baseada em softwares e hardwares livres.

Resumidamente, a plataforma consiste em uma placa de circuitos com entradas e saídas para um microcontrolador AVR, um ambiente de desenvolvimento e o *bootloader* que já vem gravado no microcontrolador [24].

O microcontrolador é constituído de um microprocessador, memória e periféricos de entrada/saída e pode ser programado para funções específicas, como, por exemplo, o controle de máquinas e diferentes automações.

Existem outras plataformas construídas para microcontroladores, mas o Arduino tem se destacado no cenário mundial pela facilidade de programação, versatilidade e baixo custo [25]. Mesmo para os que desejam interações de alto nível, o Arduino tem atendido às expectativas.

Uma das versões mais difundidas [26] do um Arduino é a Duemilanove (Fig. 1) que apresenta 6 portas analógicas e permitem a medida de tensões externas, sendo possível receber informações de uma série de sensores como medidores de temperatura, pressão, umidade, distância, sensores de gases, fototransistores, etc. Existem, também, 14 portas digitais, onde é possível ler e escrever dois estados, 0/1 ou HIGH/LOW, permitindo, por exemplo, manter um LED ligado ou desligado. Quando necessário algumas dessas portas digitais podem ser configuradas para atuarem como portas de saída analógicas, através de modulação por largura de pulso ou *Pulse-Width Modulation* (PWM).

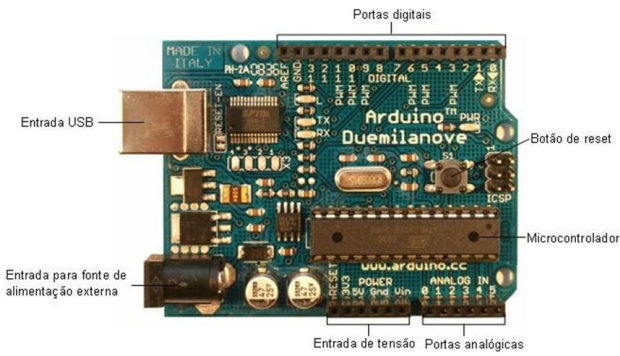


Figura 1 - Placa Arduino com especificações de entradas e saídas.

## 2. Proposta experimental

A proposta experimental consiste em estudar a carga e descarga em capacitores. Para isso se faz necessário uma revisão dos conceitos básicos envolvidos neste estudo.

Um capacitor ou condensador é um dispositivo eletrônico capaz de armazenar energia elétrica. Tal armazenamento se dá em decorrência da existência de um dielétrico que separa as suas placas condutoras. O primeiro capacitor, conhecido como garrafa de Leyden, foi inventado no século XVIII, de modo independente, pelos cientistas; Ewald Georg Van Kleist (alemão) e pelo holandês Pieter Van Musschenbroek (Leyden). A sua invenção foi de fundamental importância no estudo de fenômenos em eletrostática, pois permitia armazenar grandes quantidades de carga em um recipiente de pequenas dimensões.

A Fig. 2 mostra diferentes tipos de capacitores disponíveis

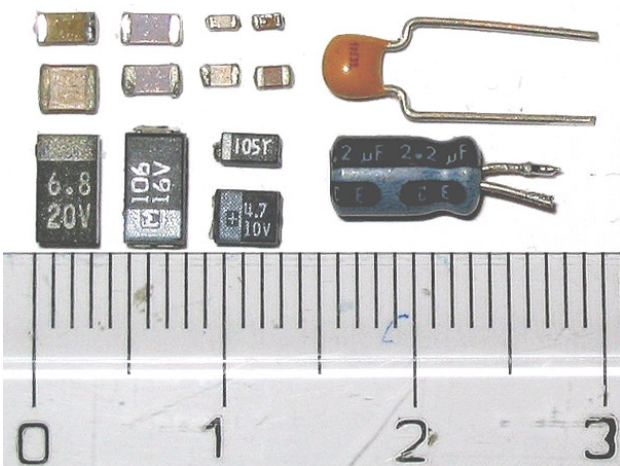


Figura 2 - Diferentes tipos de capacitores com diferentes dielétricos e formas geométricas. Fonte: <http://upload.wikimedia.org/wikipedia/commons/8/86/Photo-SMDCapacitors.jpg>.

Definimos capacitância de um capacitor por uma grandeza que relaciona a quantidade de carga armazenada e a diferença de potencial propiciada nos seu

terminais, ou seja

$$C = \frac{Q}{V}. \quad (1)$$

Assim percebe-se que quanto maior o valor de capacitância maior será a sua capacidade de armazenamento de carga.

A Fig. 3 mostra um esquema para o estudo da carga em um capacitor. Na posição 1 a chave K comutadora possibilita carregar o capacitor através do circuito RC série e quando na posição 2 o capacitor é descarregado instantaneamente.

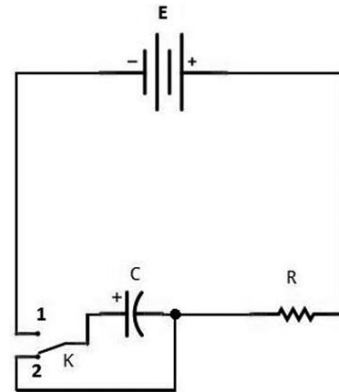


Figura 3 - Circuito RC série. A chave K ligada em 1 possibilita carregar o capacitor. Ao conectarmos K na posição 2 descarregamos o capacitor.

Considere agora o seguinte procedimento; inicialmente conectamos a chave K na posição 2 de modo a descarregar por completo o capacitor C e em seguida conectamos na posição 1. No instante em que a chave comutadora K for ligada em 1, o capacitor começa a ser carregado através da corrente  $i$ , que circula pela resistência R, com a fonte previamente ajustada a um valor de tensão nominal E.

Considerando  $V_C$  a tensão (ou ddp) no capacitor e  $V_R$  a diferença de potencial observada nos terminais do resistor temos pela lei de Kirchoff que

$$V_C + V_R = E. \quad (2)$$

Substituindo-se  $V_C$  de acordo com a Eq. (1) temos que

$$\frac{q}{C} + iR = E. \quad (3)$$

Como  $i = \frac{dq}{dt}$  temos

$$\frac{q}{C} + \frac{dq}{dt}R = E, \quad (4)$$

ou ainda

$$\frac{dq}{dt} = \frac{E}{R} - \frac{q}{RC}. \quad (5)$$

Para resolver a equação diferencial de 1ª ordem, vamos separar as variáveis deixando de um lado da

equação os termos que dependem de carga  $q$  e os termos dependentes do tempo  $t$  do outro lado da equação

$$\frac{dq}{dt} = \frac{EC}{RC} - \frac{q}{RC}, \quad (6)$$

$$\frac{dq}{(EC - q)} = \frac{1}{RC} dt. \quad (7)$$

A resolução desta equação é dada por

$$q(t) = EC(1 - e^{-t/RC}). \quad (8)$$

A grandeza  $EC$  corresponde à carga total ( $Q$ ) que pode ser armazenada no capacitor. Assim teremos

$$q(t) = Q(1 - e^{-t/RC})$$

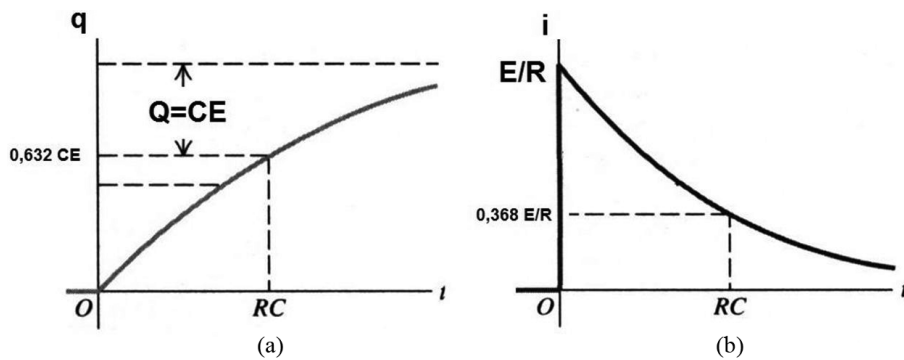


Figura 4 - A figura mostra à esquerda o gráfico da carga em função do tempo em um capacitor para um circuito RC serie e à direita o gráfico de corrente em função do tempo no mesmo circuito.

Utilizando a mesma linha de raciocínio, na descarga do capacitor, chegaremos a uma expressão exponencial para o decréscimo da sua carga com a constante  $RC$ .

Algumas aplicações importantes em circuitos, decorrentes deste processo de carga e descarga de capacitores são os filtros capacitivos conhecidos como passa-alta e passa-baixa.

É fácil perceber que:

- Os capacitores devem ser dispositivos que não permitam a passagem de componentes contínua de corrente elétrica, já que a corrente destes circuitos é utilizada para carregá-los.

- Se submetemos um capacitor a um sinal alternado a resposta de tensão nos seus terminais será tanto maior quanto menor for à frequência do sinal aplicado e para sinais de alta frequência o sinal obtido nos terminais do capacitor será menor.

Na Fig. 5a temos a indicação de um filtro passa-baixa em que o sinal de saída é obtido nos terminais do capacitor. Já para um filtro passa-alta o sinal deve ser obtido nos terminais do resistor como indica a Fig. 5b.

Para encontrar o valor da corrente elétrica, basta diferenciar a expressão acima uma vez que  $i = \frac{dq}{dt}$ . Desta forma teremos

$$i(t) = \frac{E}{R} e^{-t/RC}$$

A constante  $RC$  é definida como constante de tempo do circuito ( $\sigma\tau$ ) e corresponde ao tempo necessário para a corrente no circuito reduzir a  $1/e$  do seu valor inicial ( $E/R$ ).

À medida que a corrente no circuito se reduz a carga no capacitor aumenta e transcorrido o tempo  $RC$  esta carga atinge 63,2% do seu valor total  $EC$ . A Fig. 4 representa a carga no capacitor aumentando com o tempo enquanto a corrente no circuito cai exponencialmente.

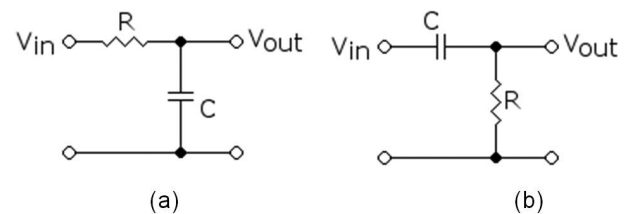


Figura 5 - Em (a) temos um filtro passa-baixa em que o sinal de saída é observado nos terminais do capacitor. Em (b) temos um filtro passa-alta em que o sinal de saída é observado nos terminais do resistor.

## 2.1. Estudo da carga com o Arduino em um circuito convencional

O objetivo deste experimento é mostrar como podemos incorporar o Arduino em montagens tradicionalmente utilizadas nos laboratórios didáticos.

A Fig. 6 mostra um arranjo experimental simples em que um capacitor e uma resistência estão associados em série utilizando como fonte de alimentação a tensão de saída +5 V na placa do Arduino. A tensão obtida nos terminais do capacitor em função do tempo foi analisada através da entrada analógica 0 do Arduino. Neste

primeiro experimento associamos uma chave externa K, conectada aos terminais do capacitor, cuja função é descarregar totalmente o capacitor. Ao desligarmos a chave K o capacitor inicia sua carga e, sua tensão cresce exponencialmente com o tempo. A Fig. 7 mostra um esquema elétrico deste circuito.

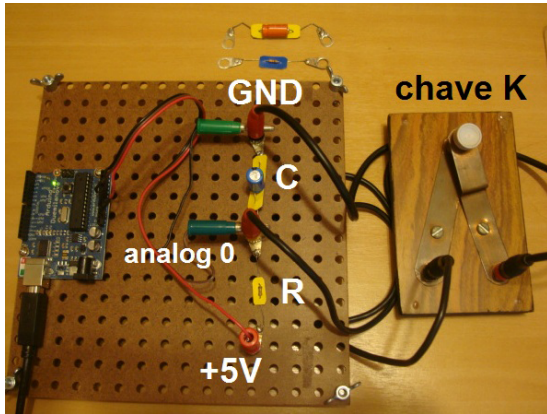


Figura 6 - Montagem para o estudo da carga com chaveamento externo.

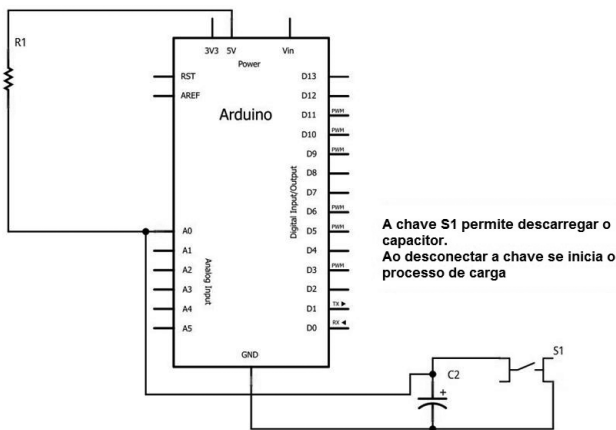


Figura 7 - Diagrama elétrico do circuito utilizado.

O código fonte utilizado para esta aquisição no Apêndice 1.

A Fig. 8 mostra aquisições com diferentes valores de *delay* fixados. Consideramos que a obtenção destes gráficos, pode ser uma dentre outras tarefas propostas em sala de aula visando verificar a influencia de diferentes valores de *delay* nos resultados obtidos.

Para a determinação da constante *RC* é também recomendável estudar como a corrente em um circuito RC evolui com o tempo no processo de carga de um capacitor. Para isso basta um pequeno ajuste na programação introduzindo o calculo desta corrente (valor da tensão obtida no resistor e dividido pela resistência associada ao circuito). O código fonte utilizado para esta aquisição está disponível no Apêndice 2.

A Fig. 9 mostra os resultados obtidos em escala logarítmica para a corrente no circuito com diferentes *delays*.

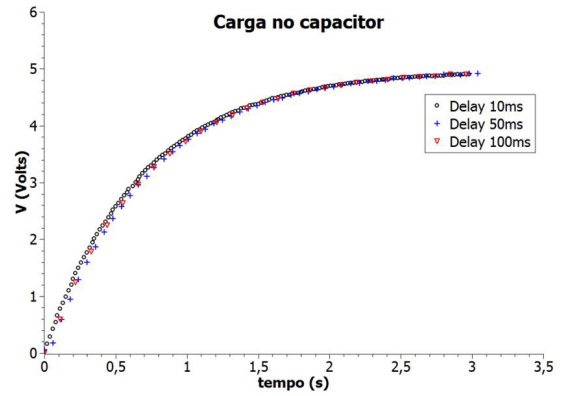


Figura 8 - Gráfico obtido para a carga em um circuito RC série com diferentes valores de *delays* fixados e com  $R = 1,5 \text{ k}\Omega$  e  $C = 470 \mu\text{F}$  (Gráficos construídos na versão freeware do software SciDavis disponível para download em <http://scidavis.sourceforge.net/index.html>).

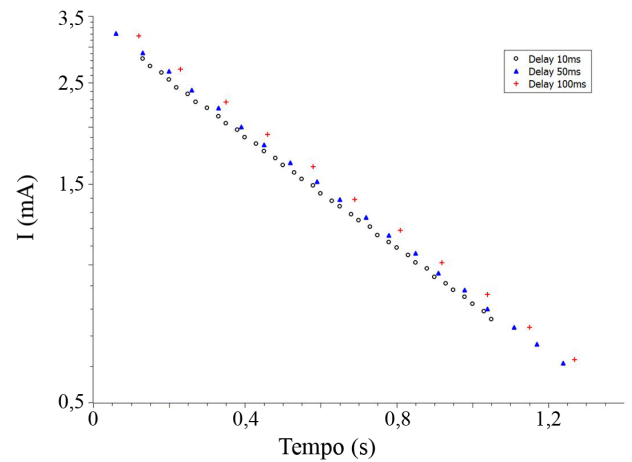


Figura 9 - Gráficos  $\ln(I)$  vs. tempo para o circuito RC série obtido para diferentes valores de *delays*, com  $R = 1,5 \text{ k}\Omega$  e  $C = 470 \mu\text{F}$ .

Os valores para a constante *RC* com respectivas incertezas foram:  $(0,692 \pm 0,021)$ ;  $(0,682 \pm 0,023)$  e  $(0,680 \pm 0,024)$  para os *delays* de 10, 50 e 100 ms respectivamente.

Observa-se que estes valores diferem essencialmente na incerteza associada à inclinação de cada uma destas retas, reduzindo com a diminuição do tempo de *delay*. Este é um estudo bastante interessante que pode ser realizado e debatido em sala de aula, contribuindo no aprendizado em análise estatística e contribuição de erros em medidas físicas.

## 2.2. Automação do experimento de carga e descarga

Para o estudo em um único experimento da carga e descarga em um capacitor devemos utilizar duas chaves externas A Fig. 10 mostra um esquema adotado para esta medida com o Arduino e duas chaves externas.

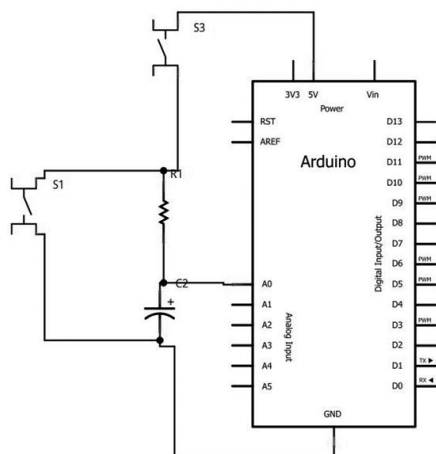


Figura 10 - Carga e descarga em um capacitor com chaves externas. A chave S3 conecta o resistor em +5 V e possibilita a carga do capacitor. Desligando a chave S3 e ligando a chave S1 conectamos o resistor em GND e possibilitamos a descarga do capacitor.

Esta montagem apesar de viável representa um risco considerável para a entrada USB do PC e da placa Arduino, tendo em vista que qualquer descuido do operador pode colocar o circuito em curto (conectando simultaneamente as duas chaves). Para viabilizar este estudo vamos recorrer a outros recursos que o Arduino pode oferecer que corresponde a possibilidade de enviarmos comandos através das portas digitais. Isso significa que estas portas podem se comportar como portas de entrada ou saída de sinais, o que abre um leque inesgotável de possibilidades de automação em experimentos. Sendo assim vamos utilizar uma de suas portas digitais como saída, comandada através de software.

Para a elaboração do programa vamos utilizar a função `digitalWrite()` que escreve uma informação digital, ou seja, 0 (LOW) ou 1 (HIGH). O resistor será conectado em uma porta digital que vai funcionar como uma chave comutadora. Em um dado instante de tempo estará no estado alto (HIGH) promovendo a carga do capacitor e depois de um dado intervalo de tempo irá para o estado baixo (LOW), possibilitando a descarga. Este chaveamento deve ocorrer em um intervalo de tempo controlado pelo operador. A Fig. 11 mostra o esquema do circuito utilizado e o Apêndice 3 fornecemos. O código fonte para este experimento é dado no Apêndice 3

A Fig. 12 mostra diferentes intervalos fixados para a carga e descarga automática.

A Fig. 10 mostra a influencia do tempo de chaveamento nas medidas realizadas. Nota-se, como esperado, que para tempos maiores temos uma melhor visualização do processo de carga e descarga. No nosso entender esta pode ser mais uma atividade experimental proposta em sala, bastante interessante, pois possibilita ao estudante maior domínio da tecnologia e do sistema de medida utilizado.

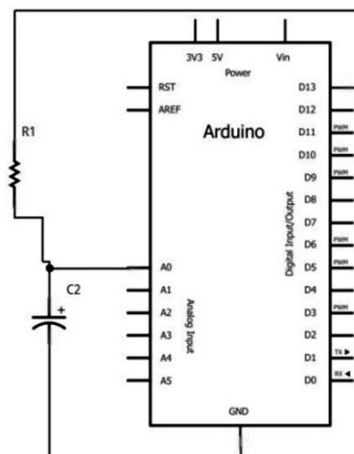


Figura 11 - Resistor conectado a porta digital 13 que deverá alternar o seu estado em intervalos de tempo determinados de modo que em estado alto o capacitor será carregado e em estado baixo descarregado.

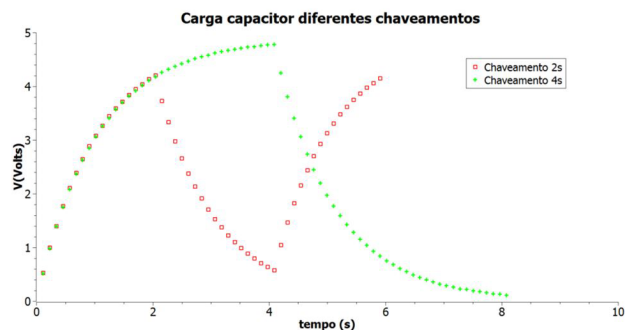


Figura 12 - Fixamos tempos de chaveamento em 2 e 4 s para uma constante  $RC$  igual a 4,7 s.

Outras atividades podem ser propostas aos estudantes tais como;

- Inserir na programação um chaveamento totalmente automatizado, ou seja, realizado a partir da leitura do valor de tensão da porta analógica. Em outras palavras, “se o valor lido de tensão lido no capacitor for maior ou igual a 95% do valor de pico (5,0 V) transite a chave para o estado baixo e neste caso, a descarga se inicia. Este processo permanece até que o valor lido seja menor ou igual a 0,05% do valor total”.

- Transformar o Arduino em um “capacímetro” [27], em que a capacitância pode ser obtida através da medida da constante de tempo e do valor da resistência utilizada no circuito.

### 2.3. Gráficos em tempo real

Muitas vezes é importante visualizar os dados em tempo real em um determinado experimento. Neste item vamos apresentar de que maneira podemos visualizar gráficos representativos da grandeza estudada utilizando *Processing*.

O *Processing* é uma linguagem/ambiente de código aberto que possibilita programar imagens, animações ou interações.

Foi desenvolvido inicialmente para ensinar os fundamentos da programação de computadores dentro de um contexto visual, e hoje milhares de estudantes e profissionais de todo o mundo a utilizam, em prototipagem e produção. Assim como o Arduino, apresenta uma vasta biblioteca que possibilita inúmeras aplicações.

Há uma comunicação muito simples que acopla o Arduino ao Processing, permitindo, por exemplo, a construção de jogos e animações interativas dentro de um contexto físico, ou seja, buscando informações através de sensores.

Para a observação de dados em tempo real, com apenas a instalação de algumas bibliotecas e alguns exemplos podemos observar os sinais de saída das portas digitais e analógicas do Arduino. Caso tenha interesse em saber como conseguir as visualizações gráficas em tempo real, acesse o tutorial que elaboramos disponível na Ref. [28].

A Fig. 13 mostra os gráficos obtidos para a captura simultânea de dois circuitos RC com diferentes constantes de tempo, utilizando o *SimpleSerialArduinScope*. As portas observadas são as analógicas 0 e 1, as demais foram mantidas em GND.

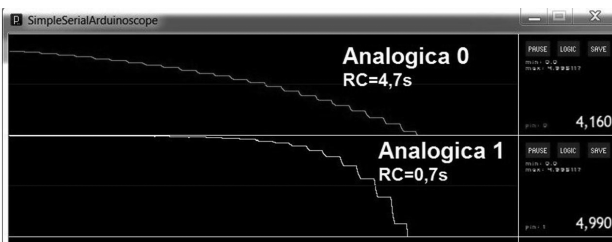


Figura 13 - Gráfico em tempo real utilizando o *ArduinScope* com o *Processing*.

Outra possibilidade é a visualização em uma melhor resolução utilizando o *Osciloscópio Zoom*. Neste caso podemos visualizar uma única saída por vez. As Figs. 14 e 15 mostram os gráficos do valor de corrente elétrica no circuito na carga do capacitor, observados para diferentes constantes  $RC$ .

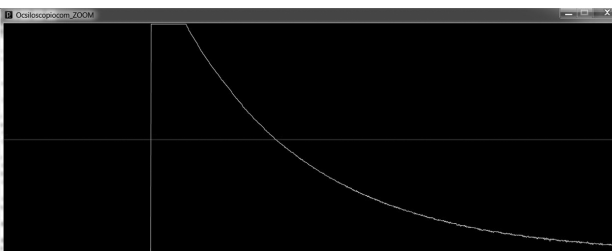


Figura 14 - Corrente elétrica versus tempo na carga do capacitor para uma constante de tempo  $RC = 4,7$  s.

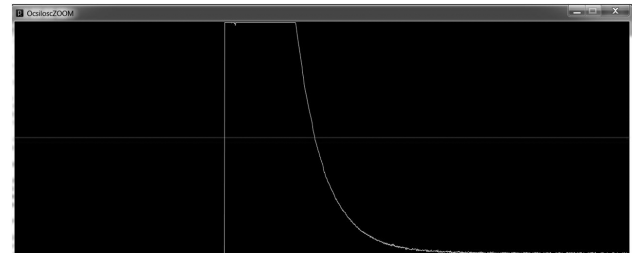


Figura 15 - Corrente elétrica versus tempo na carga do capacitor para uma constante de tempo  $RC = 0,7$  s.

### 3. Considerações finais

Neste trabalho mostramos um exemplo de aquisição e automação de dados utilizando a plataforma Arduino. A carga e descarga de um circuito RC série foi realizada através de dois procedimentos: coleta de dados num circuito convencional, isto é, os circuitos didáticos de carga e descarga de capacitores podem ser acoplados ao Arduino e num circuito automatizado. Os códigos de programação são bastante simples e com pequenas modificações mostramos como podemos transitar de uma aplicação em que o Arduino se comporta apenas como uma interface de aquisição de dados, para uma aplicação em que utilizamos um chaveamento automático, explorando, portanto as potencialidades que um micro-controlador pode oferecer.

Mostramos também que é possível observar os resultados em tempo real através de gráficos utilizando o *Processing*. Percebemos que não há a necessidade de grandes conhecimentos em eletrônica para efetuar as conexões do circuito RC com o Arduino e que a disponibilidade de bibliotecas e códigos de programação, facilita consideravelmente a utilização da plataforma em experimentos didáticos.

A possibilidade de escrever e ler em portas digitais do Arduino abre uma infinidade de aplicações para a experimentação em física e consideramos que sua inserção em sala de aula representa um passo importante para uma maior apropriação da tecnologia na construção do conhecimento.

Toda a plataforma Arduino, tanto em *software*, quanto em *hardware*, é *open-source*, o que facilita muito o seu uso e disseminação. Ou seja, existe um número infundável de bibliotecas e tutoriais disponíveis na web para muitas aplicações.

Esses fatores nos permitem enfatizar a simplicidade do uso da plataforma Arduino como aquisição e automação de dados, aliado ao custo relativamente baixo e bons resultados apresentados, contribuindo significativamente para tornar o laboratório didático de física um ambiente de investigação.

## Apêndice 1

```
int pinocapacitor = 0; //pino que está ligado o capacitor (porta analogica 0)
```

```

int valorLido = 0; //valor lido na entrada analogica
float tensaocapacitor = 0; //tensaocapacitor valor convertido para volts
unsigned long time;
void setup() {
Serial.begin(9600); //Inicializa comunicação Serial numa taxa de 9600 bps
}

void loop() {
time = millis(); // base de tempo para a coleta
valorLido = analogRead(pinocapacitor);
// leitura da porta analógica em que o capacitor está conectado
tensaocapacitor = (valorLido * 0.0048876); // 5V / 1023 = 0.0048876 (precisão do A/D)
Serial.println(tensaocapacitor); //imprime o valor da tensão do capacitor em linha
Serial.print(time); // imprime o tempo em linha
Serial.print("\t"); // imprime o tempo em tabela

delay(100); //espera 100 milisegundos para fazer nova leitura
}

```

## Apêndice 2

```

int pinocapacitor = 0; //pino que está ligado o capacitor (porta analogica 0)
int valorLido = 0; //valor lido na entrada analogica
float tensaocapacitor = 0; //tensaocapacitor valor convertido para volts
float valorI=0; // corrente I valor em mA
#define valorresistor 1.5 ; // valor do resistor em Kohms
unsigned long time;

void setup() {
Serial.begin(9600); //Inicializa comunicação Serial numa taxa de 9600 bps
}

void loop() {
valorLido = analogRead(pinocapacitor);
// leitura da porta analógica em que o capacitor está conectado
tensaocapacitor = (valorLido * 0.0048876); // 5V / 1023 = 0.0048876 (precisão do A/D)
valorI=(5.0-tensaocapacitor)/valorresistor ;
// calcula a corrente elétrica no circuito em mA
time = millis(); // base de tempo para a coleta
Serial.println(time); // imprime o tempo em linha
Serial.print(tensaocapacitor);
Serial.print("\t"); //imprime o valor da tensão do capacitor em tabela
Serial.print(valorI);
Serial.print("\t");

delay(100); //espera 100 milisegundos para fazer nova leitura
}

```

## Apêndice 3

```

/* Carga e descarga em um capacitor usando a função BlinkWithoutDelay */

const int resistorPin = 13; // define o pino em que o resistor será conectado

// Variáveis que serão utilizadas
int estadoresistor = LOW; // estado inicial do pino do resistor
long previousMillis = 0; //armazena o instante em que o estado do pino do resistor mudou
int pinocapacitor = 0; //pino que está ligado o capacitor (porta analogica 0)
int valorLido = 0; //valor lido na entrada analogica
float tensaocapacitor = 0; //tensaocapacitor valor convertido para volts

long interval = 20000; // intervalo de tempo em milisegundos em que o estado do pino do resistor
// irá mudar de estado ou seja, em alto, carrega o capacitor
// e em baixo descarrega.

unsigned long time;

void setup() {

pinMode(resistorPin, OUTPUT); // define o modo de funcionamento da saída digital
// ligada ao resistor, modo saída.
}

```



```

Serial.begin(9600); //Inicializa comunicação Serial numa taxa de 9600 bps
}

void loop()
{
valorLido = analogRead(pinocapacitor); // leitura do pino analogico do capacitor
tensaocapacitor = (valorLido * 0.0048876); // 5V / 1023 = 0.0048876 (precisão do A/D)
time=(millis()); // base de tempo de coleta de dados
Serial.println(time); //imprime o valor de tempo em linha
Serial.print(tensaocapacitor); //imprime o valor da tensao do capacitor
Serial.print("\t"); //imprime o valor da tensao do capacitor em tabela

delay(100); // define o intervalo de tempo para a coleta de dados

if(time - previousMillis > interval) {
// salva o ultimo instante em que o pino do resistor mudou de estado
previousMillis = time;

// se o pino estiver em alto mude para baixo e vice-versa
if (estadoresistor == LOW)
estadoresistor = HIGH;
else
estadoresistor = LOW;

//escreva na porta digital em que o resistor está ligado o estado do resistor
digitalWrite(resistorPin, estadoresistor);
}
}

```

## Referências

- [1] M.M. Magalhães, D. Schiel, I.M. Guerrini e E. Marega Jr., *Revista Brasileira de Ensino de Física* **24**, 97, (2002).
- [2] V. Heckler, M. de F.O. Saraiva e K. de S.F. Oliveira, *Revista Brasileira de Ensino de Física* **29**, 267 (2007).
- [3] M.F. Barroso, G. Felipe e T. Silva, in: *X Encontro de Pesquisa em Ensino de Física*, Londrina (2006).
- [4] D.M. Aliprandini, E. Schuhmacher, e M.C. Santos, in: *Simpósio Nacional de Ensino de Ciência e Tecnologia*, Ponta Grossa (2009).
- [5] P. Dorneles, I. Araujo e E.A. Veit, *Revista Brasileira de Ensino de Física* **28**, 487 (2006).
- [6] C.E. Aguiar, *Computadores no Ensino de Física* (CEDERJ, Rio de Janeiro, 2005), disponível em <http://omnis.if.ufrj.br/~marta/aprendizagememfisica/ceaguiar.pdf>, acessado em 9/5/2011.
- [7] E. Montarroyos e W.C. Magno, *Revista Brasileira de Ensino de Física* **23**, 57 (2001).
- [8] R. Haag, *Revista Brasileira de Ensino de Física* **23**, 176 (2001).
- [9] M.A. Cavalcante, E. Silva, R. Prado e R. Haag, *Revista Brasileira de Ensino de Física* **24**, 150 (2002).
- [10] W.C. Magno e E. Montarroyos, *Revista Brasileira de Ensino de Física* **24**, 497 (2002).
- [11] M.A. Cavalcante, C.R.C. Tavoraro e D. Guimarães, *Física na Escola* **7**(2), 73 (2006).
- [12] M.A. Cavalcante, A.E. Bonizzia e L.C.P. Gomes, *Revista Brasileira de Ensino de Física* **30**, 2501 (2008).
- [13] M.A. Cavalcante, A.E. Bonizzia e L.C.P. Gomes, *Revista Brasileira de Ensino de Física* **31**, 4501 (2009).
- [14] G. Dionísio e W.C. Magno *Revista Brasileira de Ensino de Física* **29**, 287 (2007).
- [15] M.C.R. Pessanha, S.G. Cozendey e M. de O. Souza, *Revista Brasileira de Ensino de Física* **32**, 4503 (2010).
- [16] R.R. Soares e P. de F. Borges, *Revista Brasileira de Ensino de Física* **32**, 2501 (2010).
- [17] A.B. Vilar, A.R. Correia, L.F. Oliveira e A.L. Pereira, in: *XVII Simpósio Nacional de Ensino de Física*, São Luis (2007).
- [18] A.G. Bezerra Jr., L.E. Merkle, E.S. de Souza, L.S. Spolaore, E. Ricetti, G.A. Giménez-Lugo, e N.C. Saavedra Filho, in: *XVIII Simpósio Nacional de Ensino de Física*, Vitória (2009).
- [19] A.R. Souza, A.C. Paixão, D.D. Uzêda, M.A. Dias, S. Duarte, e H.S. Amorim, *Revista Brasileira de Ensino de Física* **33**, 1702 (2011).
- [20] Pasco Scientific, disponível em <http://www.pasco.com>, acesso em 9/5/2011.
- [21] Phywe Wysteme GMBH & CO.KG, disponível em <http://www.phywe.com>, acesso em 9/5/2011.
- [22] Centro Industrial de Equipamentos de Ensino e Pesquisa, disponível em <http://www.cidepe.com.br>, acesso em 9/5/2011.
- [23] <http://arduino.cc>, acessado em 9/5/2011.
- [24] <http://www.sabereletronica.com.br/secoes/leitura/1307>, acessado em 23/9/2011.
- [25] <http://blog.spock.com.br/2009/07/arduino-uma-plataforma-open-source.html>, acessado em 23/9/2011.
- [26] [www.vivaolinux.com.br](http://www.vivaolinux.com.br), acessado em 23/9/2011.
- [27] <http://www.arduino.cc/playground/Portugues/LearningCapacitanceMeter>: proposta de construção de um capacitômetro utilizando o Arduino, acessado em 12/5/2011.
- [28] <http://labduino.blogspot.com/p/tutoriais.html> tutoriais sobre o Arduino e Processing, acessado em 12/5/2011.