

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
ESCOLA POLITÉCNICA
DEPARTAMENTO DE ELETRÔNICA E DE COMPUTAÇÃO

PROJETO INTEGRADO

Sistema de Aquisição de Dados Referentes a Testes de Polímeros

Autores: **Fernando Guimarães Ferreira** DRE: 104019002
 Marcelo Caetano Grael Tepedino DRE: 104009968

Avaliador:

Carlos José Ribas D'Avila

Avaliador:

Joarez Bastos Monteiro

DEL
Junho de 2009

Sumário

1	Introdução	2
1.1	Objetivo	2
1.2	Organização do Documento	3
2	Tecnologias Utilizadas	4
2.1	Projeto de Hardware	4
2.1.1	Microcontrolador ATMEL ATMEGA 8	4
2.1.2	Circuito Integrado Temporizador 555	5
2.1.3	Sensor de Temperatura de Precisão Centígrada LM35	7
2.1.4	Conversor de Sinais MAX232	8
2.2	Projeto de Software	9
2.2.1	Módulo PySerial	9
2.2.2	Módulo wxPython	10
2.2.3	SQLite e o Módulo PySQLite	10
2.2.4	Módulo Matplotlib	11
3	Implementação do Projeto	13
3.1	Projeto do circuito	13
3.2	Interface com o Usuário	16
4	Conclusão	22

Capítulo 1

Introdução

1.1 Objetivo

Este projeto tem por objetivo a construção de um sistema para aquisição de dados referentes ao teste de novos sensores de temperatura construídos a partir de polímeros com as seguintes características:

- O sensor de temperatura do circuito deve operar dentro da faixa de 20°C a 100°C , com precisão de $0,1^{\circ}\text{C}$.
- O circuito adquirirá resistências na faixa de $1K\ \Omega$ a $100\ K\Omega$, com precisão de $10\ \Omega$.
- A comunicação com PC é realizada por meio da porta serial padrão RS-232.
- A transferência de dados deve ser feita de modo confiável.
- O circuito deve medir e informar continuamente, em intervalos regulares, a partir do pedido de início da aquisição.
- A interface gráfica deve ser simples, amigável e funcional. Deve servir desde a aquisição e armazenamento dos dados amostrados até a recuperação destes, por meio de gráficos ou planilhas eletrônicas.

Com esse projeto, pretende-se automatizar o processo de testes dos novos sensores poliméricos, atividade atualmente realizada manualmente. Desta maneira, aumenta-se

a confiabilidade e a padronização dos testes realizados. O processo torna-se assim mais rápido e eficaz.

1.2 Organização do Documento

O Capítulo 2 descreve as tecnologias utilizadas para o desenvolvimento do projeto.

O Capítulo 3 mostra a implementação do projeto.

Por fim, o Capítulo 4 apresenta as conclusões.

Capítulo 2

Tecnologias Utilizadas

Este projeto está dividido em um módulo de *hardware* e outro computacional. As seguintes tecnologias foram escolhidas para o desenvolvimento do projeto de acordo com sua disponibilidade, documentação existente, facilidade de utilização, adaptação aos requisitos e por sua aceitação no contexto de projetos de eletrônica.

2.1 Projeto de Hardware

Para o projeto do circuito foram necessários um microcontrolador ATMEL AT-MEGA 8, um temporizador, um sensor de Temperaturas LM35 e um conversor de sinais MAX232 para o protocolo RS-232. Estes encontram-se descritos nas sub-seções que seguem.

2.1.1 Microcontrolador ATMEL ATMEGA 8

Um microcontrolador (microcontroller, também denominado MCU) é um computador em um chip, contendo um processador, memória e funções de entrada/saída. É um microprocessador que enfatiza a alta integração, em contraste com outros microprocessadores de propósito geral (como os utilizados nos PCs). Eles são "embutidos" no interior de algum outro dispositivo (geralmente um produto comercializado) para que possam controlar as funções ou ações do produto. Um outro nome para o microcontrolador, portanto,

é controlador embutido.

Neste projeto foi utilizado o ATMEL ATMEGA 8. Suas especificações podem ser encontradas em seu datasheet [?]. É apresentada a pinagem do CI na Figura 2.1

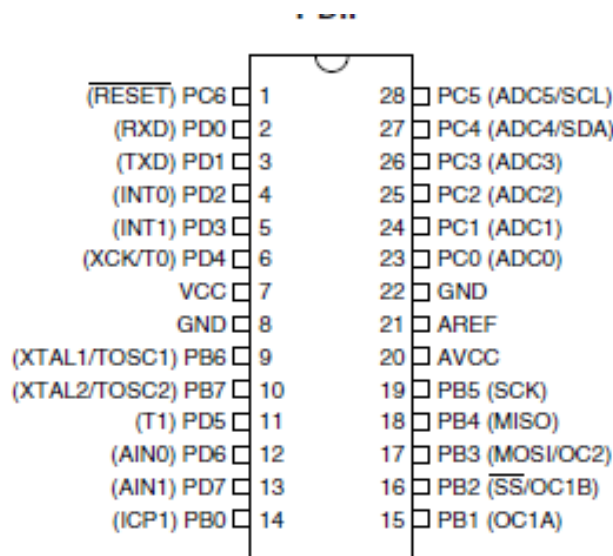


Figura 2.1: Pinagem do Microcontrolador ATMEL ATMEGA8

2.1.2 Circuito Integrado Temporizador 555

O 555 (Figura 2.2(a)) é um circuito integrado utilizado em uma variedade de aplicações de temporização ou como multivibrador. O CI foi desenhado por Hans R. Camenzind, sendo projectado em 1970 e comercializado em 1971 pela Signetics (mais tarde adquirida pela Philips). O temporizador 555 é um dos mais populares e versáteis circuitos integrados jamais produzidos.

O 555 tem três modos de operação:

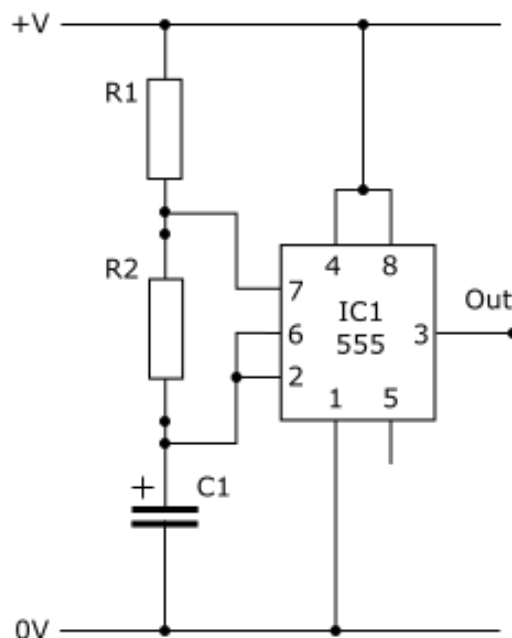
- Modo monoestável: nesta configuração, o CI 555 funciona como um disparador. Suas aplicações incluem temporizadores, detector de pulso, chaves imunes a ruído, interruptores de toque, etc.
- Modo astável: O CI 555 opera como um oscilador. Os usos incluem pisca-pisca de LED, geradores de pulso, relógios, geradores de tom, alarmes de segurança, etc.

- Modo biestável: o CI 555 pode operar como um flip-flop, se o pino DIS não for conectado e se não for utilizado capacitor. As aplicações incluem interruptores imunes a ruído, etc.

Usando apenas um capacitor e um resistor, o intervalo de temporização, ou seja, o tempo durante o qual a saída permanece em nível baixo, pode ser ajustado para de acordo com a necessidade de cada aplicação. Um exemplo de configuração é mostrado na Figura 2.2(b).



(a) Circuito integrado NE555.



(b) Esquemático do 555N

Figura 2.2: Circuito Integrado Temporizador 555

O intervalo de tempo t é dado por:

$$t = RC$$

onde t é o tempo que leva para carregar o capacitor C a 63 % da tensão aplicada. Essa característica permite, para um capacitor fixo, calcular o valor do resistor R em função do tempo, dado em milissegundos.

A Tabela 2.1.2 apresenta a topologia das portas do circuito conforme a numeração apresentada na Figura 2.2(b).

Nr	Nome	Aplicação
1	GND	Terra
2	TR	Gatilho (<i>Trigger</i>) Um valor de tensão baixo ($< 1/3 V_{cc}$) neste pino activa o biestável interno e a saída.
3	Q	Durante um intervalo de tempo, a saída (Q) permanece em $+V_{CC}$.
4	R	Um intervalo de temporização pode ser interrompido pela aplicação de um pulso de Reset.
5	CV	Controle de tensão (diferença de potencial-DDP) permite acesso ao divisor interno de tensão ($2/3 V_{CC}$).
6	THR	O limiar (threshold) Um valor de tensão alto ($> 2/3 V_{cc}$) neste pino desactiva o biestável interno e a saída.
7	DIS	Conectado a um capacitor, cujo tempo de descarga irá influenciar o intervalo de temporização. A sua função é descarregar o condensador.
8	V+, VCC	A tensão (diferença de potencial-DDP) positiva da fonte, que deve estar entre $+5$ e $+15V$.

Tabela 2.1: Portas do circuito integrado 555

Na Tabela 2.1.2, verifica-se as especificações do circuito.

Tensão de alimentação (VCC)	4.5 até 15 V
Corrente de alimentação (VCC = +5 V)	3 até 6 mA
Corrente de alimentação (VCC = +15 V)	10 até 15 mA
Corrente de saída (máxima)	200 mA
Dissipação de potência	600 mW
Temperatura de Operação	0 até 70 °C

Tabela 2.2: Especificações do circuito integrado 555

2.1.3 Sensor de Temperatura de Precisão Centígrada LM35

O modelo do sensor de Temperatura LM35 é bastante simples, necessitando apenas do sensor propriamente dito, de um sistema amplificador de sinal e de uma interface que realize a leitura do sinal amplificador. É um sensor de precisão, que apresenta uma saída de tensão linear relativa à temperatura em que ele se encontrar no momento em que for

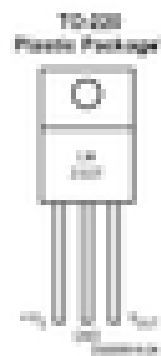
alimentado por uma tensão que pode variar de 4 à 20 V, tendo em sua saída um sinal de 10mV para cada grau Celsius de temperatura.

O LM35 não necessita de qualquer calibração externa ou *trimming* para fornecer, com exatidão, valores dentro da faixa de temperatura de $-55\text{ }^{\circ}\text{C}$ à $150\text{ }^{\circ}\text{C}$. Este sensor tem saída com baixa impedância e tensão linear, fazendo com que o interfaceamento de leitura seja simples.

A Figura 2.3(a) apresenta o circuito integrado e a Figura 2.3(b) mostra sua respectiva pinagem.



(a) Circuito integrado LM35



(b) Esquemático do LM35

Figura 2.3: Sensor de Temperatura de Precisão Centígrada LM35

2.1.4 Conversor de Sinais MAX232

MAX232 é um circuito eletrônico que converte sinais de uma porta serial para sinais adequados para uso em circuitos microprocessados, por exemplo. O MAX232 amplifica/reduz sinais RX, TX, CTX e RTS. A discrepância de voltagem (acima de $\pm 12\text{ V}$ do RS232 para 3.3 V TTL) é gerada por capacitores (normalmente de 10 nF). O MAX232 é um transmissor/receptor duplo que fornece níveis de voltagem TIA/EIA-232-F de uma

única fonte de tensão de 5 V. Cada receptor converte entradas TIA/EIA-232-F para níveis de 5V TTL/CMOS. Estes receptores têm um limiar típico de 1.3 V, uma histerese típica de 0,5 V e podem aceitar ± 30 V de entrada. Cada transmissor converte níveis de entrada TTL/CMOS em níveis TIA/EIA-232-F.

2.2 Projeto de Software

Para o desenvolvimento da interface gráfica do usuário foi utilizado a API wxWidgets, integrado à linguagem computacional Python.

Python é uma linguagem de programação de alto nível, interpretada, imperativa, orientada a objetos, de tipagem dinâmica e forte. A linguagem foi projetada com a filosofia de enfatizar a importância do esforço do programador sobre o esforço computacional. Prioriza a legibilidade do código sobre a velocidade ou expressividade. Combina uma sintaxe concisa e clara com os recursos poderosos de sua biblioteca padrão e por módulos e frameworks desenvolvidos por terceiros.

Neste projeto, além do módulo *WxPython*, foram utilizados os módulos *PySerial*, para poder realizar a interface via porta-serial, e o módulo *Matplotlib*, para a criação de gráficos baseados nos criados com o programa Matlab ©. O armazenamento de dados foi realizado utilizando-se o banco de dados relacional embarcado *SQLite 3* criado a partir do módulo *PySQLite*.

Todos os módulos estão descritos nas sub-seções a seguir.

2.2.1 Módulo PySerial

PySerial é uma biblioteca que suporta a comunicação serial para uma variada gama de dispositivos: porta serial tradicional, tecnologia Bluetooth, portas infra-vermelho, etc. Serve de *back-end* para programas escritos em Python para os sistemas operacionais Windows, Linux, BSD, Jython and IronPython (.NET e Mono).

Para utilizar suas funcionalidades basta usar a diretiva:

```
import serial
```

A documentação pode ser encontrada em [?].

2.2.2 Módulo wxPython

O módulo wxPython é a API WxWidgets C++ embarcada na linguagem Python. Este é um *framework* para a confecção de uma interface gráfica e outras facilidades para mais de uma plataforma. É suportada por todos os sistemas operacionais *MS Windows*, *Unix* with GTK+, e MacOS.

O exemplo “Hello World” pode ser encontrado abaixo.

```
import wx
app = wx.App(redirect=True)
top = wx.Frame(None, title="Hello World", size=(300,200))
top.Show()
app.MainLoop()
```

2.2.3 SQLite e o Módulo PySQLite

SQLite é uma biblioteca C que implementa um banco de dados SQL embutido. Programas que usam a biblioteca SQLite podem ter acesso a banco de dados SQL sem executar um processo RDBMS separado.

SQLite não é uma biblioteca de cliente usada para conectar com um grande servidor de banco de dados. SQLite é o servidor. A biblioteca SQLite lê e escreve diretamente para e do arquivo do banco de dados no disco.

O uso do SQLite é recomendado onde a simplicidade da administração, implementação e manutenção são mais importantes que incontáveis recursos que SGBDs mais voltados para aplicações complexas possivelmente implementam. Entretanto situações onde a simplicidade é a melhor escolha são muito mais freqüentes do que pode-se imaginar.

O módulo *PySQLite* faz a interface entre essa biblioteca e a linguagem Python. Um exemplo pode ser visto abaixo.

```

from pysqlite2 import dbapi2 as sqlite
import datetime

con = sqlite.connect(":memory:", detect_types=sqlite.PARSE_COLNAMES)
cur = con.cursor()
cur.execute('select ? as "x [timestamp]"',
(datetime.datetime.now(),))
dt = cur.fetchone()[0]
print dt, type(dt)

```

2.2.4 Módulo Matplotlib

Matplotlib é uma biblioteca para a criação de gráficos para Python e sua extensão numérica Numpy. É dada uma “pylab” API projetada para funcionar próxima a linguagem MATLAB, tornando sua utilização fácil à usuários do pacote matemático.

A interface “pylab” é procedural e baseada em máquina de estados, como *OpenGL*. Matplotlib pode funcionar embarcada usando pacotes como o wxPython.

O exemplo a seguir produz a saída mostrada na Figura 2.4

```

from pylab import *

t = arange(0.0, 2.0, 0.01)
s = sin(2*pi*t)
plot(t, s, linewidth=1.0)

xlabel('time (s)')
ylabel('voltage (mV)')
title('About as simple as it gets, folks')
grid(True)
show()

```

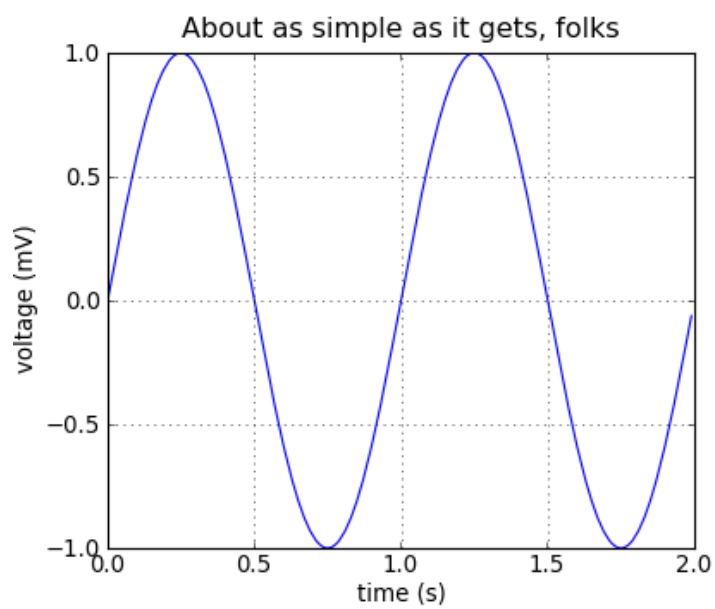


Figura 2.4: Exemplo gerado através do módulo Matplotlib

Capítulo 3

Implementação do Projeto

3.1 Projeto do circuito

O projeto do hardware pode ser dividido em cinco partes que serão explicitadas em detalhe nesta seção. Uma delas responsável pela alimentação de todo o circuito, duas responsáveis pela obtenção dos parâmetros medidos, o microcontrolador e a conexão deste com o PC.

Todo o circuito é alimentado por de uma fonte DC de 12 V, regulada para 5 V pelo circuito mostrado na Figura 3.1.

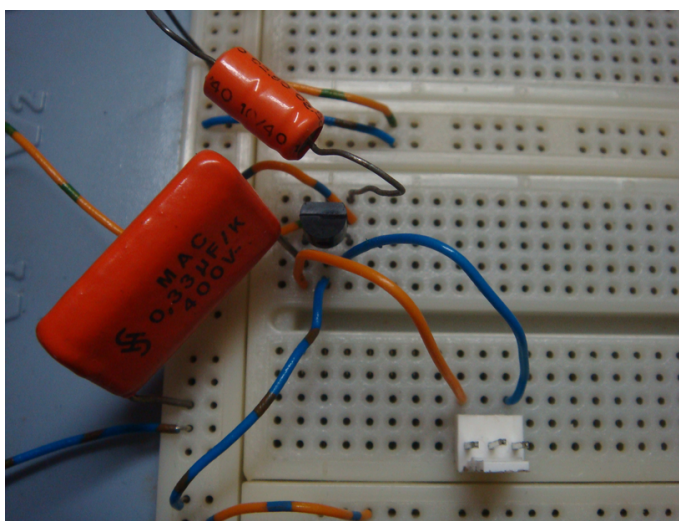


Figura 3.1: Circuito regulador de tensão.

Como os componentes utilizados no projeto requerem uma alimentação por volta deste valor, esta é uma maneira de garantir a estabilidade da tensão de entrada de cada parte do projeto de hardware.

A parte de aquisição da temperatura é realizada por um sensor LM35, com seu sinal de saída amplificado por um amplificador operacional LM358N. É aplicado um fator de 2,56, que faz com que a tensão de entrada no microcontrolador se iguale a tensão de referência interna utilizada no mesmo quando a temperatura medida é de 100°C. Esta amplificação é calculada a partir da fórmula

$$1 + \frac{R_1}{R_2}, \text{ onde } R_1 = 34K3 \, \Omega \text{ e } R_2 = 22 \, K\Omega \quad (3.1)$$

O conversor AD do microcontrolador possui 10 bits, o que permite a leitura de até 1024 níveis diferentes de temperatura, neste caso com variações de 2,56 mV. Esta variação mínima detectada pelo conversor do microcontrolador equivale exatamente à variação de 0,1°C no sensor LM35 amplificada por 2,56. O LM35 e a amplificação de seu sinal de saída são mostrados na Figura 3.2.

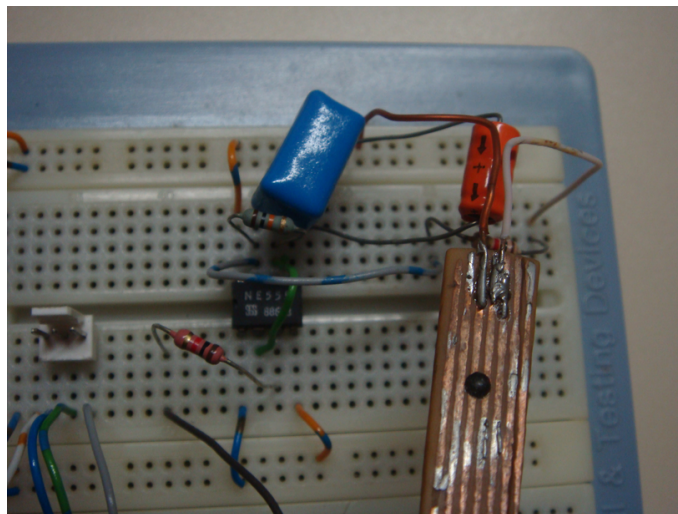


Figura 3.2: Sensor LM35.

O circuito responsável pela medição da resistência do polímero utiliza um oscilador NE555. Esta medição é possível conectando a saída do oscilador a um contador, regulando o tempo de permanência no nível lógico alto do sinal de saída do oscilador. A

fórmula utilizada no cálculo desta resistência é dada por:

$$t_{low} = \ln(2)(R_1 + R_2)C, \text{ onde } R_2 = 10 \text{ K}\Omega \text{ e } C = 1 \text{ }\mu\text{F} \quad (3.2)$$

Este circuito pode ser visto na Figura 3.3.

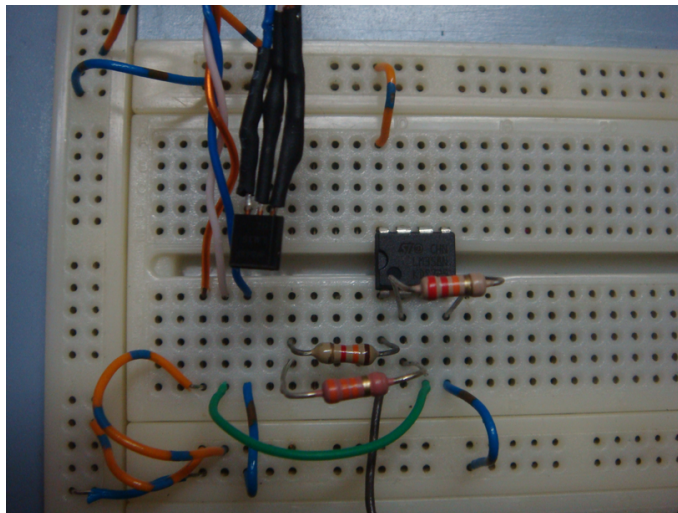


Figura 3.3: Oscilador NE555.

As funções do microcontrolador utilizadas foram as duas descritas anteriormente, bem como um canal de comunicação bilateral com o conversor MAX232 através do pino 2 (de entrada) e 3 (de saída). A programação do ATMEGA foi realizada em BASIC, utilizando-se o programa BASCOM-AVR.

A cada 10 amostragens de temperatura/resistência é extraída a média, que então é enviada ao PC. Desta maneira, procurou-se mitigar a incerteza na aquisição de dados. No cálculo das resistências, foram utilizados os valores reais de resistência e capacitância, obtidos com o auxílio de um multímetro de precisão.

A Figura 3.4 apresenta o setor do circuito onde está localizado o microcontrolador.

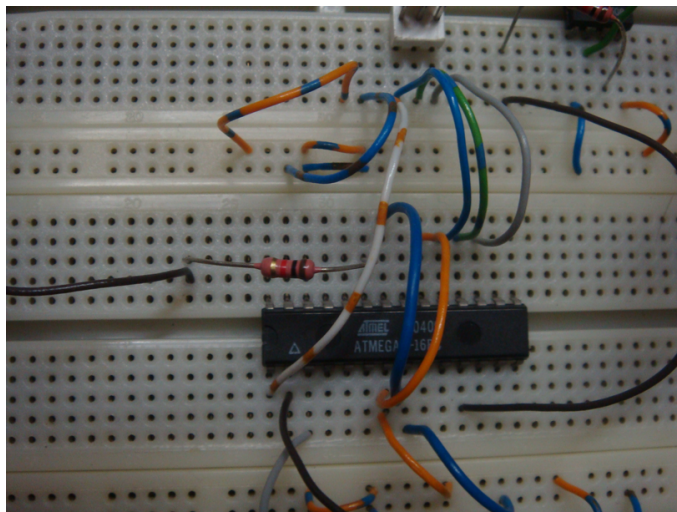


Figura 3.4: Microcontrolador ATMEL ATMEGA8.

A utilização do MAX232 (Figura 3.5) é simples, como fora explicado anteriormente.

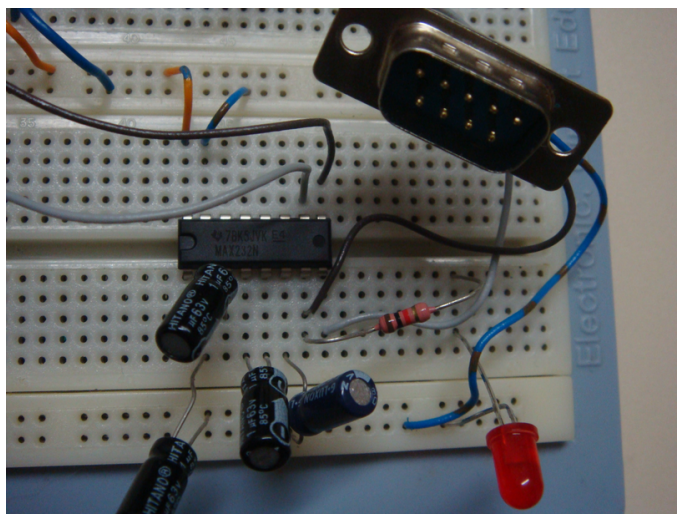


Figura 3.5: Circuito de comunicação MAX232.

3.2 Interface com o Usuário

Como dito, foi realizada uma interface gráfica para a utilização do sistema de aquisição de dados. Nesta seção serão mostradas as principais funcionalidades que vão desde adquirir o dado e armazená-lo, até sua recuperação nos formatos CSV e gráfico. A

esse programa deu-se o nome de *RSampler*.

A Figura 3.6 mostra a tela principal do programa. Nela o usuário pode escolher se deseja adquirir novos dados ou acessar aos que já foram armazenados anteriormente.

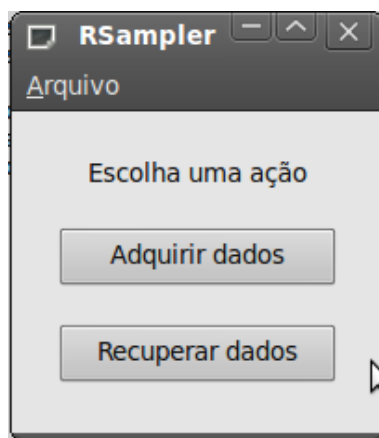


Figura 3.6: Tela principal do sistema RSampler

Tela para Aquisição de Dados

Ao escolher “Adquirir Dados”, através da seleção do botão correspondente na tela principal, o usuário tem acesso à tela apresentada na Figura 3.7.

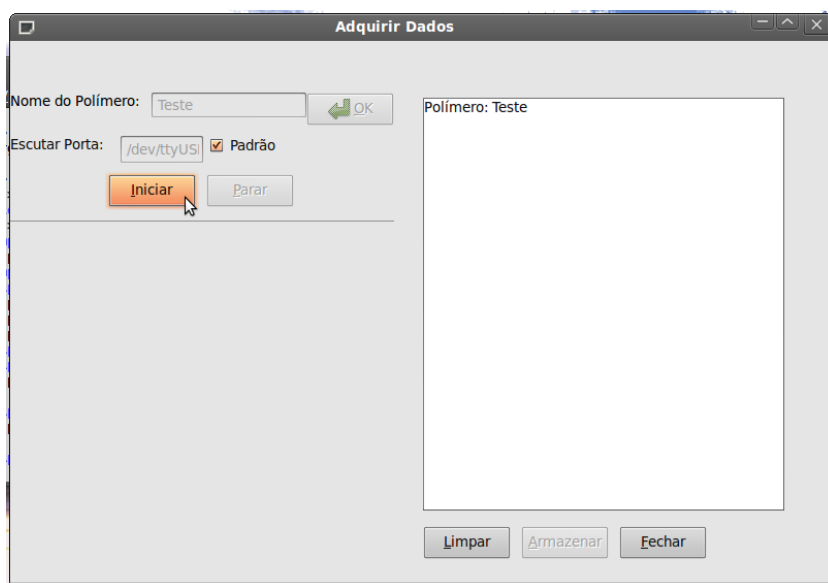


Figura 3.7: Tela para Aquisição de dados.

Primeiramente, o usuário escreve o nome/identificador do polímero correspondente no campo “Nome do polímero”. O usuário ainda pode selecionar a porta na qual a comunicação serial será realizada. Caso essa seleção não seja feita, a porta padrão (“COM1” para Windows, “/dev/ttyS1” para linux e “/dev/ttyUSB1” para o driver “USB-Serial”, como no caso da execução acima) é utilizada. Feito isso, o botão “Iniciar” habilita-se, permitindo que a aquisição de temperaturas e resistências comece. Nesse momento, o programa envia ao microcontrolador do *hardware* a mensagem “START” e espera receber em troca as medições de acordo com o protocolo definido na seção anterior.

O programa adquire essas medição com uma frequência de 1 Hz. Enquanto é realizada a aquisição, são impressas as medidas atuais de temperatura e resistência, assim como o horário de leitura da porta serial. O usuário pode encerrar a aquisição a qualquer momento ao apertar no botão “Parar”. O programa envia, então, a mensagem “STOP” e a comunicação é cessada.

Ao apertar o botão “Armazenar”, os dados impressos na tela são armazenados no banco de dados relacional embarcado *RSampler.db*. Por fim, o usuário pode retornar a tela principal, selecionando o botão “Fechar”.

Tela para Recuperar Dados

Para entrar neste módulo do programa, o usuário necessita selecionar o botão “Recuperar Dados”. A Figura 3.8 apresenta a tela de apresentação desta funcionalidade.

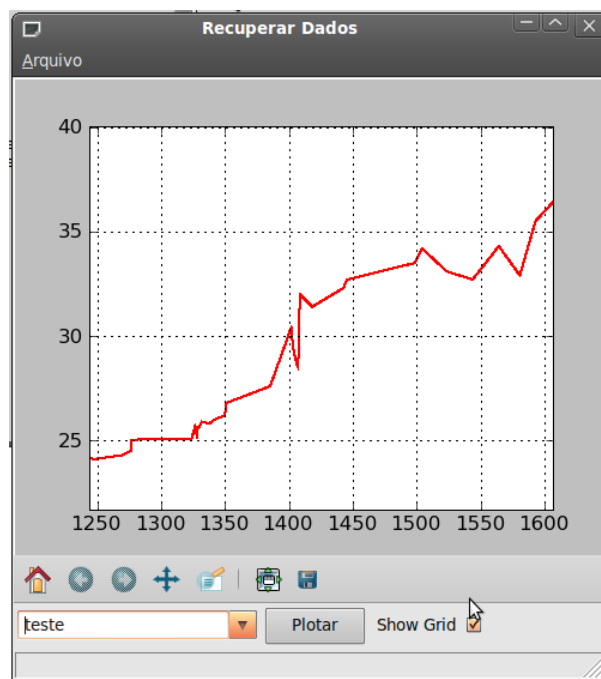


Figura 3.8: Tela para Recuperação de Dados

Essa tela apresenta dois possíveis formatos para a recuperação de dados: CSV e gráficos PNG. Ao iniciar a tela, ela automaticamente carrega todos os polímeros presentes no banco de dados. O usuário pode então selecioná-los através do *combobox* no canto inferior esquerdo da tela. Para carregar os dados, basta pressionar o botão “Plotar”.

Neste momento, um gráfico é gerado e disponibilizado ao usuário. Uma série de funcionalidades são providas para a manipulação deste gráfico.



Reinstaura as configurações iniciais do gráfico.



Botão para armazenar o gráfico em disco, como mostra na Figura 3.9.



Botão que permite ajustar as dimensões do gráfico, como é mostrado na Figura 3.10

O usuário pode ainda utilizar a ferramenta “Zoom” para enquadrar o período de tempo de interesse.

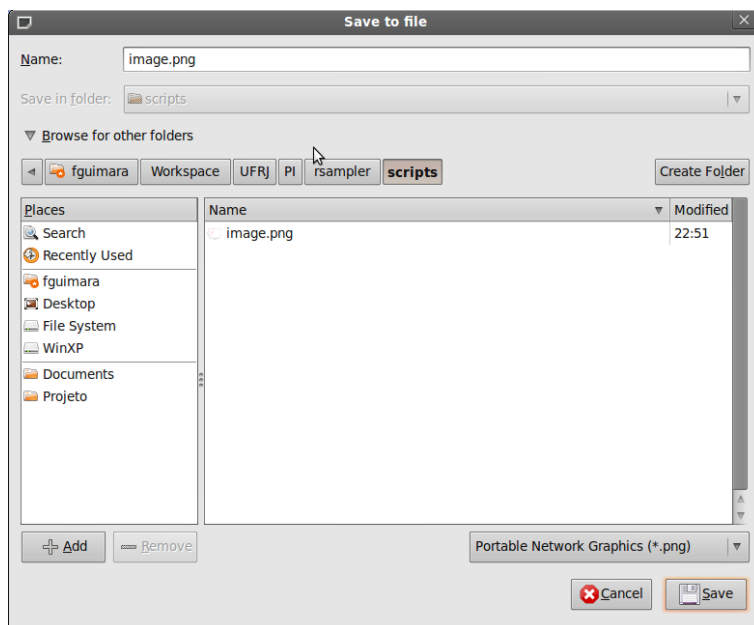


Figura 3.9: Armazenar Gráfico

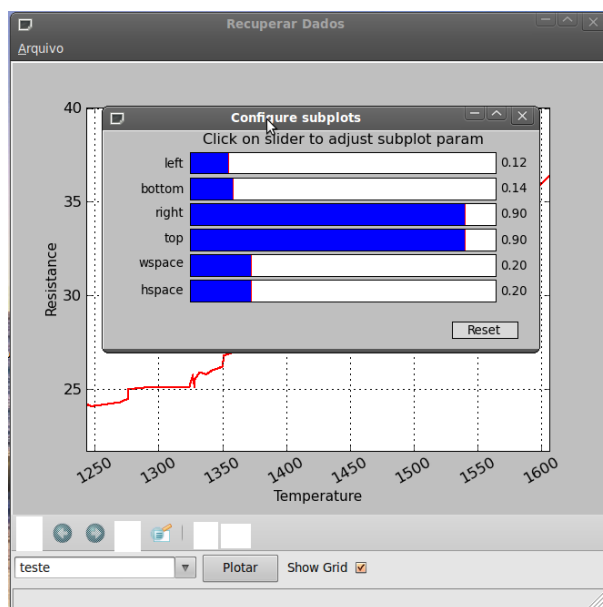


Figura 3.10: Ajustar Dimensões de um Gráfico

O programa ainda salva em formato (CSV) compatível aos principais gerenciadores de planilhas eletrônicas. Basta selecionar no menu principal a opção Arquivo > Salvar em CSV.

Capítulo 4

Conclusão

Durante esse projeto, foi realizado um circuito de aquisição de temperaturas e resistências de polímeros para a criação de novas tecnologias de sensores. Com a utilização do sistema implementado, essa aquisição, antes manual, será automatizada, aumentando a confiabilidade e eficácia dos testes.

O projeto será utilizado no Laboratório de Físico-Química de Polímeros e Instrumentação (Lafiq), localizado no bloco K no Instituto de Macromoléculas (IMA-UFRJ).

Com o circuito projetado, foram alcançados os requisitos especificados que se referem à faixa de operação do sensor testado e à tolerância na medição dos parâmetros observados.

A interface gráfica pode ser executada tanto com o sistema operacional *Windows* quanto com em *Linux*, bastando que esteja instalado o interpretador *Python* e os módulos necessários ao seu funcionamento.

Como próximos passos, sugere-se a gravação da placa de hardware e sua integração ao forno de teste.