

Solução para captura de análise de sinais analógicos

Mateus Ribeiro	Vitor Vieira
11/0017480	11/0067151
mateusrba@gmail.com	vitor.ecomp@gmail.com

Tiago Lenza	Matheus Pedreira
11/0020987	11/0017749
tiago.sk8@gmail.com	matrpedreira@gmail.com

Daniel Sandoval
09/0109899
daniel@loopec.com.br

28 de fevereiro de 2013

Sumário

1	Introdução	2
1.1	Objetivos	2
1.2	Sinais Analógicos	2
1.2.1	Transmissão de Dados	3
2	Projeto de Hardware	4
2.1	Plataforma Arduino	4
2.2	Metodologia e Desenvolvimento	5
3	Projeto de Software	7
3.1	Plataforma MATLAB	7
3.2	Metodologia e Desenvolvimento	7
3.2.1	Transmissão do Sinal Adquirido	7
3.2.2	Análise do Sinal Adquirido	8
3.2.3	Integração	8
4	Conclusão	11
4.0.4	Resultados Obtidos	11
4.0.5	Considerações Finais	11

Capítulo 1

Introdução

Sinais analógicos são extremamente importantes de serem analisados pois estes são a base para todos os sistemas de telecomunicações, além de nos fornecerem diversas informações sobre fenômenos naturais, tendo em vista que o sinal é uma onda variando no tempo. Um bom exemplo é o som.

Conseguir captar esses sinais analógicos nos possibilita analisá-los de diversas formas dependendo da aplicação. Um exemplo é a análise do som, que é uma vibração no ar, onde podemos separar diferentes sons pelas suas respectivas frequências.

1.1 Objetivos

Nosso objetivo principal ao procurar uma solução para captura de análise de sinais analógicos, era desenvolver uma solução móvel, de baixo custo, que pudesse ser utilizada sem grandes dificuldades, com uma precisão aceitável, e que tivesse uma aplicação real.

Também tínhamos como objetivo que a nossa solução fosse capaz de analisar a frequência do nosso sinal recebido para...

Como um sinal analógico pode representar diversas informações diferentes, procuramos desenvolver cálculos de integral por barra e por trapézio que pudessem ser carregados na solução móvel para facilitar o cálculo da integral do sinal quando esse o fosse necessário.

1.2 Sinais Analógicos

Um sinal analógico é uma onda variável que representa uma medida variando em função do tempo, estes sinais normalmente são utilizados em contexto elétrico, no entanto, também podem estar em um contexto mecânico,

pneumático hidráulico e em muitos outros pois qualquer informação pode ser convertida em um sinal analógico.

Sinais analógicos podem ser usados para medir mudanças em fenômenos físicos como o som, a luz, a temperatura, a posição ou a pressão através de um transdutor de sinal, este têm basicamente a função de converter energia de uma forma para outra.

A principal vantagem em se utilizar um sinal analógico é a boa definição deste sinal pois ele possui uma quantidade infinita de resoluções, comparando com sinal digital percebemos que este possui uma maior densidade e robustez.

A principal desvantagem na utilização de sinais analógicos é a quantidade de ruído presente.

1.2.1 Transmissão de Dados

Para realização da transmissão de dados, é muito mais interessante transmitir um sinal analógico do que um digital. Visto que o dado analógico é mais robusto, ele está menos suscetível à interferências. Uma outra vantagem na utilização de sinais analógicos para transmissão é a capacidade de fácil processamento, o que barateia o custo de transmissão. Para que um sinal digital seja enviado na forma analógica, este deve ser modulado.

Capítulo 2

Projeto de Hardware

Nosso objetivo em hardware é captar um sinal elétrico analógico para análise. Para atingir o objetivo estabelecido, escolhemos a plataforma Arduino, baseada no microcontrolador ATmega328, para captar em sua entrada analógica o sinal e transmiti-lo via interface serial a um computador pessoal.

2.1 Plataforma Arduino

A plataforma Arduino de desenvolvimento de hardware consiste em uma placa de circuito impresso, ilustrada na Figura 2.1, e uma IDE - *Integrated Development Environment* - que nos permite escrever código e programar o microcontrolador.

No modelo utilizado da plataforma, o UNO, a placa de circuito impresso possui um microcontrolador ATmega382, que possui 2KB SRAM, 1KB EEPROM e opera a uma frequência máxima de 16MHz. Além do microcontrolador, estão presentes também alguns LEDs indicativos de estado, 13 pinos de comunicação digital e 6 entradas analógicas.

Utilizando a IDE fornecida, podemos codificar programas que interajam com os pinos digitais (que podem ser configurados como entrada ou saída), fazer leituras das entradas analógicas, e realizar computações sobre esses dados.

As bibliotecas fornecidas com o Arduino tornam a plataforma extremamente poderosa, capaz de cálculos matemáticos, operações com *strings*, comunicação serial, entre outros.

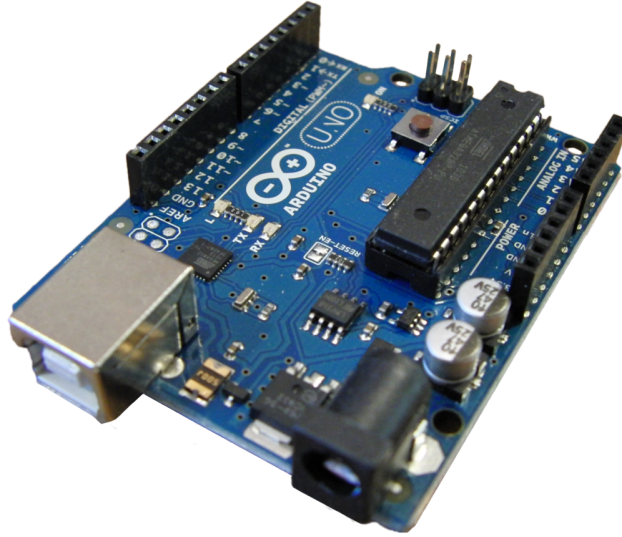


Figura 2.1: Arduino Uno

2.2 Metodologia e Desenvolvimento

Como é possível visualizar no Exemplo 2.1, as bibliotecas disponíveis tornam a solução extremamente simples e elegante. Primeiramente, amostramos o sinal através da porta A0, armazenando o resultado no vetor de inteiros *buffer*. A fim de se obter a máxima taxa de amostragem possível, são realizadas todas as leituras e armazenadas em memória, para que depois possam ser transferidas via interface serial.

A aquisição do sinal foi realizada detectando variações entre 0 e 5V representada em 1024 níveis (0-1023). Devido às limitações de memória da plataforma escolhida, nós podemos obter até 800 leituras a uma taxa $470Hz$.

Segundo Teorema de Nyquist:

“Para um sinal arbitrário de frequência BHz , o sinal da filtragem poderá ser completamente reconstruído pelo receptor caso a frequência de amostragem seja de no mínimo $(2B)Hz$.”

Portanto, temos que nossa solução de hardware é capaz de amostrar sinais com uma frequência de no máximo $235Hz$.

Nosso projeto é capaz de identificar um sinal analógico elétrico a partir das entradas do Arduino, no entanto muitos dos sinais analógicos que são

interessantes de serem analisados não estão em formato elétrico então a primeira coisa que deve ser feita antes de captar esses sinais é convertê-los com o uso de transdutor, um exemplo dessa conversão é a leitura do som de um receptor como um microfone, após o som ser captado por este ele é convertido para o sinal analógico que será usado na nossa solução.

```
1  #define READS 800
2  #define WAIT 2
3
4  void setup() {
5      Serial.begin(9600);
6      pinMode(13, OUTPUT);
7      digitalWrite(13, LOW);
8  }
9
10 int buffer[READS];
11 int i = 0;
12 int j = 0;
13 void loop() {
14     if(i < READS) {
15         int sensorValue = analogRead(A0);
16         buffer[i] = sensorValue;
17         delay(WAIT);
18         i++;
19     }
20     if(i == READS) {
21         if(Serial.available() > 0) {
22             char s = Serial.read();
23             Serial.println(buffer[j], DEC);
24             j++;
25         }
26     }
27 }
```

Exemplo 2.1: Código do Arduino

Capítulo 3

Projeto de Software

Nosso objetivo em software é analisar o sinal captado pela solução de hardware de maneira a identificar frequências presentes no sinal e suas respectivas magnitudes.

3.1 Plataforma MATLAB

A plataforma MATLAB (*MATrix LABoratory*) é um software interativo de alta performance voltado para o cálculo numérico. O MATLAB integra análise numérica, cálculo com matrizes, processamento de sinais e construção de gráficos em ambiente fácil de usar onde problemas e soluções são expressos somente como eles são escritos matematicamente, ao contrário da programação tradicional.

Nossa escolha pela Plataforma MATLAB foi devido à agilidade no desenvolvimento, trazida por suas funções disponíveis e interfaces implementadas.

3.2 Metodologia e Desenvolvimento

Para realizar a análise do nosso sinal pela nossa solução de software é necessário primeiramente receber o dados que estão que foram adquiridos pelo Arduino. Esses dados são 800 números de 0 a 1023 e estão armazenados no vetor *buffer* do microcontrolador.

3.2.1 Transmissão do Sinal Adquirido

Para transmissão dos dados adquiridos, utilizamos a interface serial, emulada de forma transparente sobre a interface USB. A transmissão ocorre da seguinte maneira, via interface serial, entre o Arduino e o computador:

1. Após a realização das 800 leituras, o Arduino fica em estado de espera;
2. O computador envia um caracter ao Arduino, solicitando uma leitura;
3. O Arduino envia ao computador a próxima leitura;
4. Os itens 2 e 3 se repetem até que sejam transmitidas as 800 leituras.

3.2.2 Análise do Sinal Adquirido

A partir dos valores recebidos é possível realizar uma visualização gráfica do dados que nos ajuda a identificar o tipo do sinal, a forma da onda gerada por este sinal, o que, por sua vez, nos possibilita formular estratégias específicas para o tratamento do sinal que está sendo analisado.

Nossa solução computacional nos permite também analisar o sinal no domínio da frequência através de uma Transformada Rápida de Fourier, conhecida como FFT, que, por sua vez, é baseada na Série de Fourier.

A Série de Fourier, representada na Equação 3.1, nos possibilita representar qualquer sinal através de um somatório de senos e cossenos.

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos(\frac{n\pi t}{L}) + b_n \sin(\frac{n\pi t}{L})] \quad (3.1)$$

Uma vez obtida a representação em Série de Fourier do sinal original, os coeficientes a_n e b_n representam a magnitude de cada sinal periódico que o compõe.

A solução de software implementada aplica a Transformada Rápida de Fourier, que está disponível em biblioteca padrão do MATLAB, sobre o sinal original, obtendo os coeficientes a_n e b_n e provendo visualização gráfica da magnitude do sinal sobre o espectro da frequência.

Aplicações de analisar a frequência de um sinal:

- Com o sinal no domínio da frequência é possível remover frequências indesejadas que podem ser geradas tanto por ruídos quanto por interferências, isso se aplica bem no caso de transmissões em redes de telecomunicação onde é possível saber como o sinal era para ter chegado para corrigi-lo caso necessário.

3.2.3 Integração

Integral Computacional

Método de integração computacional por somatório consiste em calcular de forma fixa uma parte da área do gráfico abaixo da função dada, e, após o

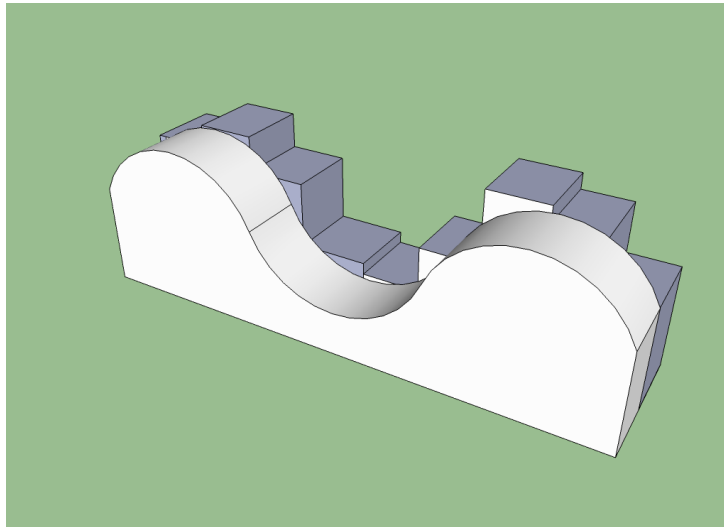


Figura 3.1: Integração computacional

cálculo, faz-se o somatório de todas as áreas. Quanto maior for a amostragem mais áreas serão calculadas fazendo com que o resultado da integral seja mais preciso, porém, com o aumento da amostragem, maior será o custo computacional. O cálculo da área abaixo da função pode ser feita de diversas formas, cada uma tendo suas vantagens em cada tipo de função, os métodos de cálculo de área implementados são:

Trapézio

É traçada uma reta entre dois pontos da amostra e através destes pode-se deduzir um trapézio, e através deste calcular a área entre os dois pontos do sinal. Esse tipo de somatório tem maior eficiência e precisão em sinais cuja a variação do sinal da derivada segunda, da função que representa o sinal, seja pequeno, ou seja funções apenas crescentes ou decrescentes.

```

double IntegralTrapezio::integrate(double a, double b)
2 {
    double x = a+intervalo;
    Ponto *A = new Ponto(funcao(a), a);
    Ponto *B = new Ponto(funcao(x), x);
    double integral = 0;
    Trapesio *vetor = new Trapesio(A, B);
    8
    for(x += intervalo; x < b; x += intervalo)
    10 {
        integral += vetor->sizeOfTrapezio();
        A = new Ponto(funcao(x), x);
    12

```

```

14         vetor->nextTrapesio(A);
15     }
16     delete A;
17     delete B;
18     delete vetor;
19     return integral;
20 }

```

Exemplo 3.1: Integral por trapézio

Por Barra

É definido um ponto do sinal dentro do intervalo da amostragem, que é usado como o valor do sinal até o próximo ponto amostrado, fazendo com que cálculo da área seja maior ou menor do que o valor correto, porém em sinais cuja a variação do sinal da derivada segunda da função, que representa o sinal, seja alta e em funções periódicas mostra-se que a quantidade de área ganha acaba anulando a quantidade perdida, fazendo com que essa integral se saia melhor que a integral por trapézio neste tipo de função.

```

1 double IntegralBarra::integrate(double a, double b)
2 {
3     double x , y;
4     double integral = 0;
5     Barra barra;
6     for(x = a, y = a + intervalo/2; x < b - intervalo; x +=
7         intervalo, y += intervalo)
8     {
9         barra.setValues(x, x + intervalo, funcao(y));
10        integral += barra.areaBarra();
11    }
12    return integral;
13 }

```

Exemplo 3.2: Integral por barras

Vantagens da integral por Somatório

As grandes vantagens, do método por somatório, são a facilidade de implementação e a grande capacidade de paralelização deste processo em vários núcleos de processamento. Outra vantagem desse tipo de integração é a habilidade de não depender de uma função para o cálculo da integral sendo possível assim o cálculo direto de um sinal amostrado.

Capítulo 4

Conclusão

4.0.4 Resultados Obtidos

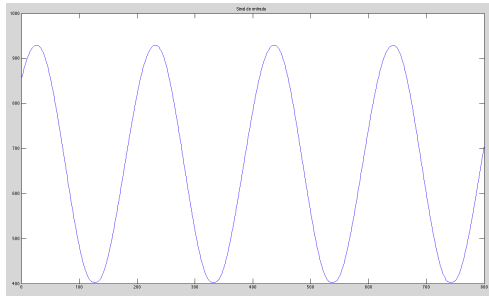
Os resultados obtidos foram de acordo com o esperado. Foi construída solução de hardware e software capaz de amostrar sinais elétricos a uma taxa de $470Hz$ e analisá-los no espectro da frequência. Conforme é possível constatar na Figura 4.1, a visualização gráfica dos sinais amostrados é extremamente satisfatória, bem como a visualização gráfica da distribuição de magnitude no espectro da frequência.

O software implementado para cálculo de integral funcionou corretamente e representa a possibilidade de melhora da solução, uma vez que foi implementado no formato de uma biblioteca de Arduino.

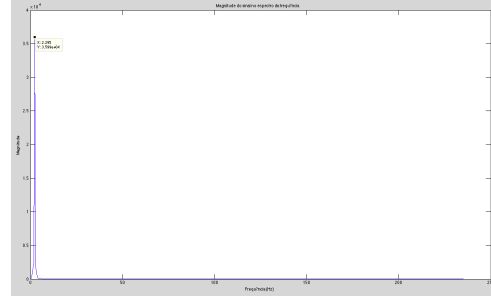
4.0.5 Considerações Finais

A solução desenvolvida em hardware e software representa ferramenta importante de análise de sinais analógicos. A possibilidade de integração em microcontroladores introduzida abre um leque imenso de aplicações sobre sinais analógicos em sistemas embarcados.

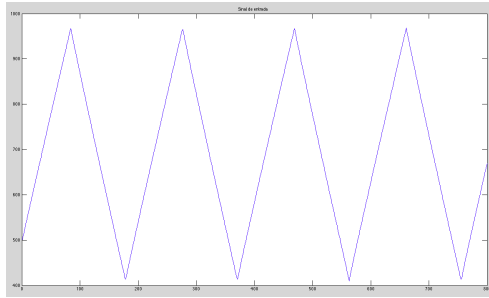
Através do desenvolvimento do projeto, verificou-se inclusive a importância do estudo de sinais analógicos, principalmente no âmbito do eletromagnetismo, importantíssimo para evolução de diversas tecnologias como, por exemplo, as de telecomunicações.



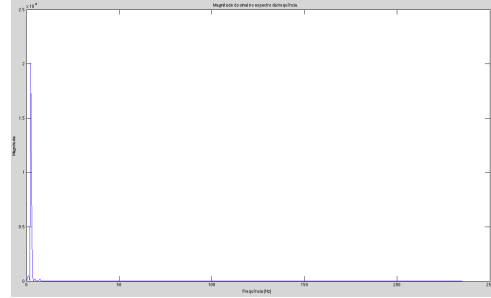
(a)



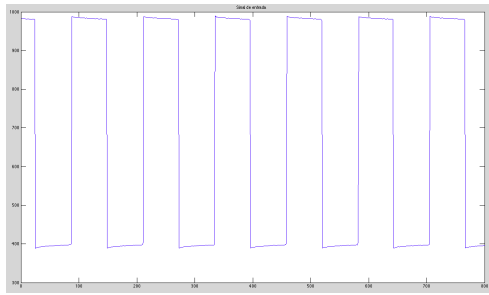
(b)



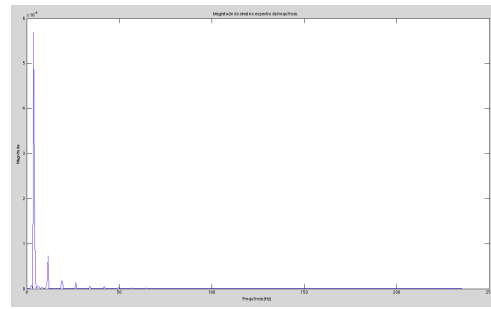
(c)



(d)



(e)



(f)

Figura 4.1: Sinais adquiridos através da solução apresentada: (a) onda senoidal; (c) onda triangular; (e) onda quadrada; e Análise no espectro da frequência dos sinais obtidos: (b) análise do sinal senoidal, (d) análise do sinal triangular; (f) análise do sinal quadrado.

Referências Bibliográficas

- [1] Boesch, F., *Integration by Example - Euler vs Verlet vs Runge-Kutta*. <http://codeflow.org/entries/2010/aug/28/integration-by-example-euler-vs-verlet-vs-runge-kutta/>
- [2] Leithold, L.; , *The Calculus with Analytic Geometry*, 6a Ed., HarperCollins Publishers, 1990.
- [3] ATmega328 <http://www.atmel.com/devices/atmega328.aspx>
- [4] Analog and Digital <http://www.stanford.edu/class/cs101/analog-digital.html>
- [5] What are Transducers? <http://www.wisegEEK.org/what-are-transducers.htm>
- [6] *Processamento Digital de Sinais - Vantagens e Aplicações* <http://pt.scribd.com/doc/49815347/Processamento-Digital-de-Sinais-Vantagens-e-Aplicacoes>