

Pereira, T.S.S. <sup>1</sup>, Araújo, J. W. B. <sup>2</sup>, Ferrando, D. F. C. <sup>2</sup>, Kakuno, E. M. <sup>2</sup>

<sup>1</sup> Instituto Federal Sul-Rio-Grandense (IFSUL) – Bagé – RS – Brasil

<sup>2</sup> Universidade Federal do Pampa (UNIPAMPA) – Bagé – RS – Brasil

## Introdução

O uso de microcontroladores como o Arduino vem se popularizando cada vez mais, e com isso práticas de medir grandezas físicas através de transdutores (sensores) comerciais também, porém pouco se discute as características dos sensores utilizados, como por exemplo, sua precisão, acurácia e resolução. Medidas realizadas de forma ingênua acabam por inviabilizar a reprodução dos experimentos, ou no pior dos casos o mesmo é reproduzido de forma “grosseira”, o que pode levar ao experimentador tirar conclusões equivocadas. Neste contexto, propomos uma análise de abordagens de mediadas de temperatura com o sensor LM35D e umidade com o sensor HT11, são de baixo custo e frequentemente usados com o Arduino. Verificamos a sua confiabilidade e precisão. Por confiabilidade entendemos o quanto o dispositivo se aproxima das especificações do fabricante, e por precisão o grau de coerência entre os valores medidos, durante medições repetidas, no mesmo objeto ou em objetos similares, em mesmas condições. Também sugerimos alguns procedimentos que podem melhorar a qualidade dos valores obtidos a partir das leituras, através da utilização de filtros por hardware e por software.

## Metodologia, Resultados e Discussões

Foram selecionados seis sensores de temperatura LM35D e cinco sensores de temperatura e umidade DHT11. Segundo o fabricante, o DHT11 consegue medir a umidade com uma acurácia de  $\pm 5\%$  a uma temperatura de  $25^\circ\text{C}$  e para a temperatura o fabricante informa uma acurácia de  $\pm 2^\circ\text{C}$  a  $25^\circ\text{C}$ . O DHT11 possui um conversor análogo digital interno de 16 bits, tornando possível fazer a leitura de sua saída a partir de entradas digitais do Arduino, para isso deve utilizar o padrão de comunicação indicado pelo fabricante, porém já existem bibliotecas do DHT11 para o Arduino, a utilizada neste trabalho foi disponibilizada por Adi Dax em 2015 (<https://github.com/adidax/dht11>).

O LM35 é produzido por diversos fabricantes, o utilizado nas medidas foi o modelo D produzido pela Analog Devices Incorporation - ADI, segundo o fabricante em seu Data Sheet, o sensor possui uma sensibilidade de 10 mV por  $1,0^\circ\text{C}$  e uma linearidade de  $\pm 0,25^\circ\text{C}$ .

As medidas com o DHT11 resultaram em divergências de mais de 50% nas leituras de umidades, comparando-se as leituras dos cinco sensores entre si, em um mesmo ambiente e registrados no mesmo instante. O menor valor registrado foi de 37% e o maior de 57%. Estes resultados indicam que este sensor não deve ser utilizado sem que haja uma prévia aferição do mesmo, e desta forma implementar possíveis ajustes de leituras por meio do software.

O LM35 gera um sinal analógico, que deve ser convertido pelo ADC do Arduino, de modo a aproveitar melhor a faixa dinâmica do ADC, utilizamos a tensão de referência interna do Arduino, o que corresponde a 1,1 V, o qual ainda tem a vantagem de ser mais estável que a fonte de alimentação de 5 V. Os seis sensores foram lidos quase que simultaneamente com intervalo entre cada conjunto de medida de 3,5 ms.

O LM35 mostrou-se mais fiel as especificações do fabricante do que DHT11, tendo uma variação nas medidas menores do que  $\pm 0,2^\circ\text{C}$ , dessa forma optamos por focar a análise nos do LM35.

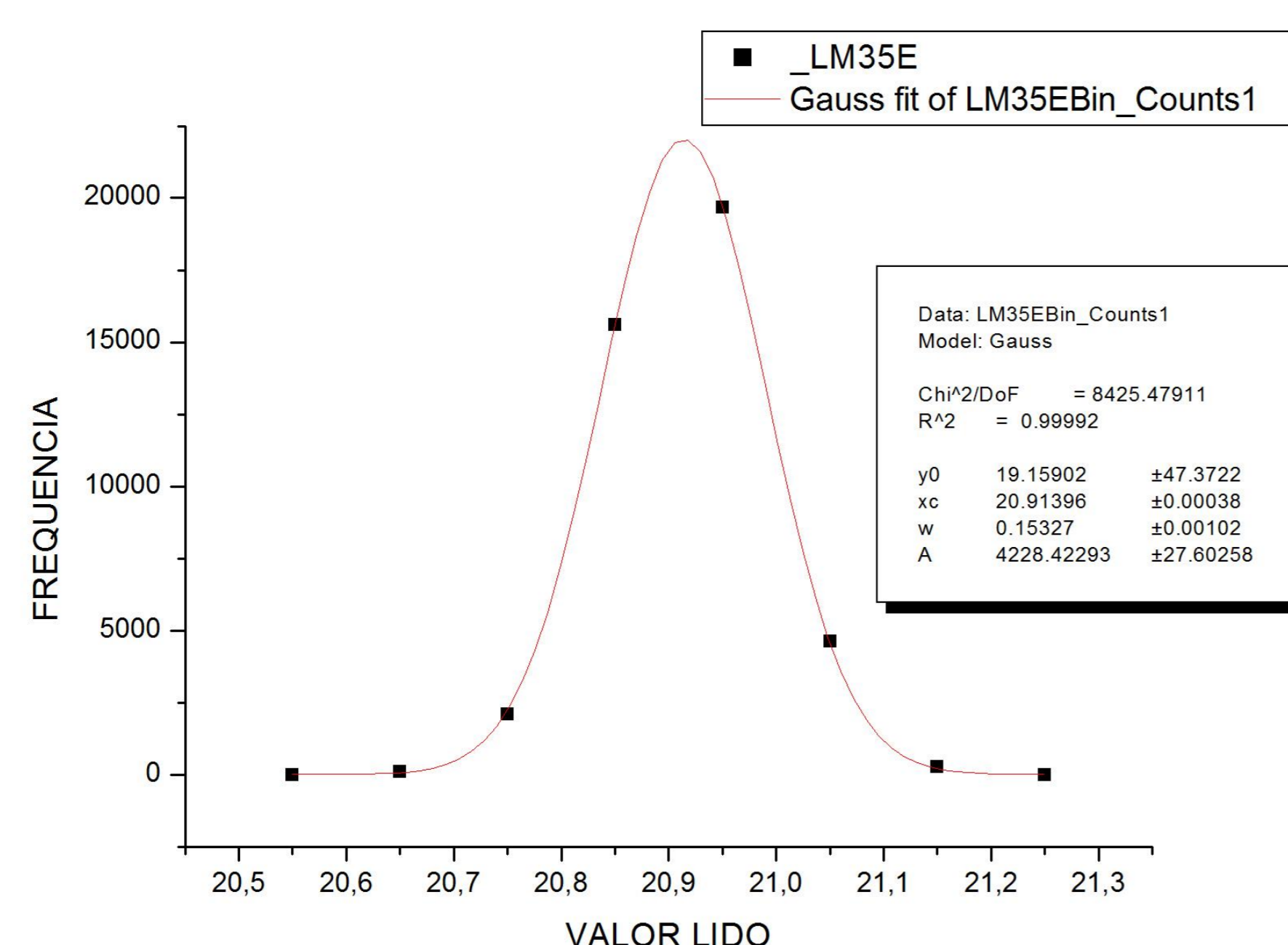


Figura 1: Ajuste gaussiano na frequência de medida.

A Figura 1, mostra um ajuste gaussiano sobre um histograma de medidas feitas sobre um dos seis sensores escolhido de forma aleatória, o ajuste indica que uma baixa dispersão das medidas, tendo uma largura média de  $0,153^\circ\text{C}$ , demonstrando que o sensor possui uma boa precisão.

As conexões entre o Arduino e os sensores eram da ordem de 5 cm, e ainda sim observamos ruído de caráter aleatório, de forma a explorar esse aspecto das medidas, resolvemos prolongar o tamanho das conexões, realizando medidas (a) utilizando um cabo de rede fios rígido e pares trançado de 2m e (b) 6 m de comprimento (c) cabo blindado de 10 m de comprimento. Para o caso (a) Foram realizadas 101900 medidas que resultou em um sensor aleatório uma largura de  $1,82^\circ\text{C}$  a uma temperatura de  $23,52^\circ\text{C}$ , para o caso (b) foram realizadas 65600 medidas com uma dispersão entre  $0^\circ\text{C}$  a  $80^\circ\text{C}$ , não sendo possível realizar um ajuste, no caso (c) não foi possível realizar medidas, pois a saída do LM35D apresentava forte oscilação, conforme Figura 2, que de acordo com o fabricante isso ocorre devido a um excesso de carga capacitiva na saída do sensor. Na Figura 3 temos o histograma das medidas realizadas no caso (c).

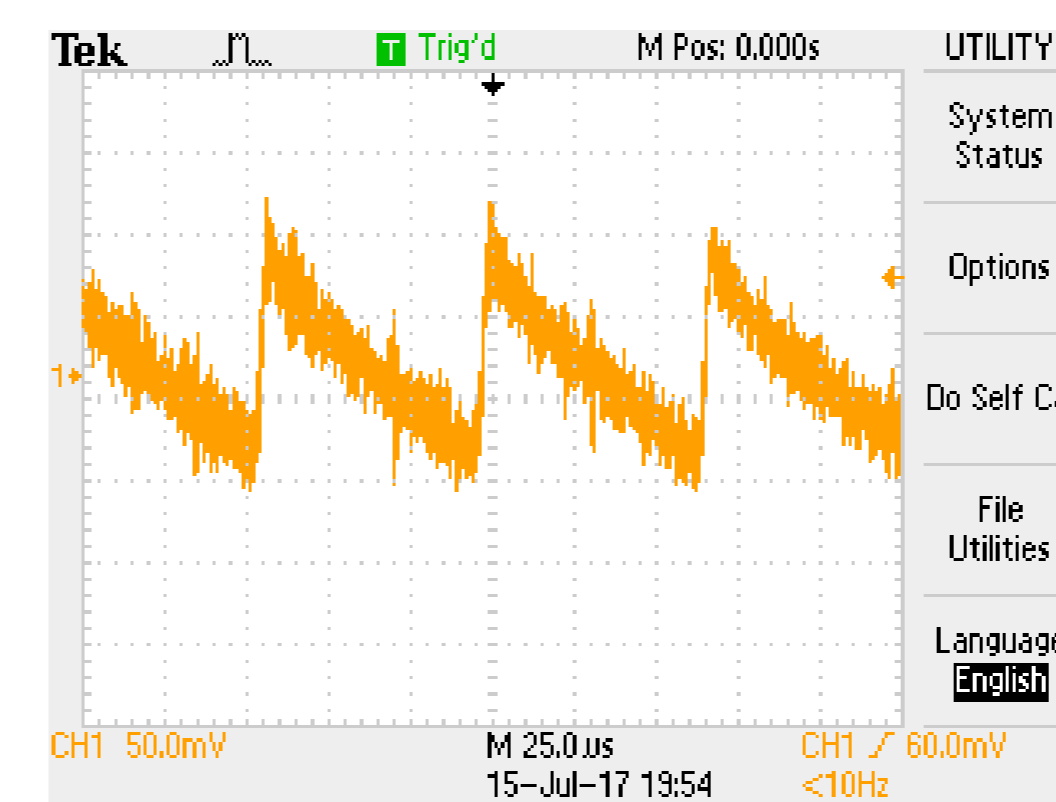


Figura 2: Sinal na saída do LM35 com cabo blindado de 10 m.

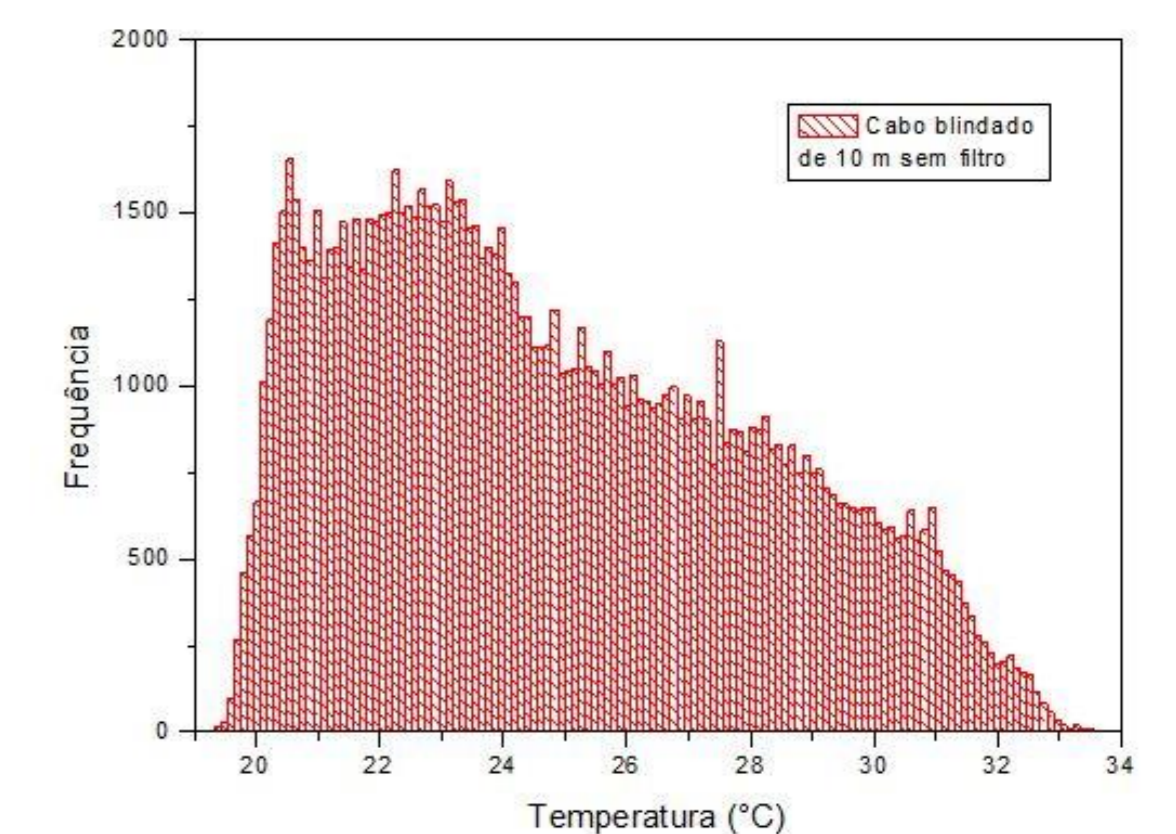


Figura 3: Histograma do sinal da Figura 2.

Ainda foram realizadas medidas com o cabo blindado de 10 m, com (d) filtro resistivo, R e (e) filtro resistivo-capacitivo, RC, ambos recomendado pelo fabricante no *data sheet*. Para o caso (e) foram realizadas 88800 medidas e resultou em uma largura de  $0,32^\circ\text{C}$  a uma temperatura de  $23,09^\circ\text{C}$ . Com o filtro resistivo, caso (d) a performance é um pouco menor que o caso utilizando o filtro RC, contudo bem melhor do que sem filtro nenhum caso (c).

Este trabalho também sugere um filtro de software que seleciona as leituras analógicas dentro de um intervalo escolhido pelo usuário, por exemplo  $\pm 20\%$  do valor médio, considerando leituras fora deste intervalo como ruído eletrônico e não variações do sensor, melhorando significativamente o resultado das leituras, porém isso acarreta um atraso nas medidas.

## Conclusão

Este trabalho chama a atenção para a seleção dos sensores quanto a sua confiabilidade, sendo bastante recomendado uma aferição prévia antes de sua utilização e quanto a utilização das entradas analógicas do Arduino quanto a seleção da referência interna e dos ruídos captados pelo ADC do Arduino. Resultados mais detalhados podem ser obtidos no endereço <https://github.com/KakiArduino/Arduino>, acessado em 29/07/2017.

## Agradecimentos

Os autores reconhecem o apoio material e financeiro do Programa Institucional de Bolsa de Iniciação à Docência – PIBID/UNIPAMPA, através do Edital CAPES no. 061/2013, e pela Portaria CAPES nº 096/2013 e CNPQ processo 405472 / 2015-3.