

**UNIVERSIDADE FEDERAL DE SANTA MARIA  
CENTRO DE TECNOLOGIA  
CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO**

## **RELATÓRIO FINAL**

**RELATÓRIO DA DISCIPLINA DE INSTRUMENTAÇÃO  
ELETRÔNICA**

**Prof. Fernanda de Moraes Carnielutti**

**Andrei Matiazi Schopf  
Davi da Silva Calil  
Guilherme Pando Wollmeister  
Pedro Medeiros de Assis Brasil  
William Rocha**

**Santa Maria, RS, Brasil  
2019**

## SUMÁRIO

<b>INTRODUÇÃO.....</b>	<b>2</b>
<b>CAPÍTULO 1 OBJETIVOS.....</b>	<b>3</b>
1.1 Motivação .....	3
<b>CAPÍTULO 2 DESENVOLVIMENTO.....</b>	<b>4</b>
2.1 Especificações do projeto .....	4
2.1.1 Lista de componentes .....	4
2.1.2 Especificações do sensor LM35.....	4
2.2 Condicionamento do sinal de entrada.....	8
<b>CAPÍTULO 3 VISUALIZAÇÃO DOS RESULTADOS .....</b>	<b>10</b>
3.1 Resultados de simulação.....	10
3.2 Leitura e visualização dos dados.....	12
<b>CAPÍTULO 4 CONCLUSÃO .....</b>	<b>15</b>
<b>CAPÍTULO 5 REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>16</b>
<b>ANEXO A .....</b>	<b>17</b>

## INTRODUÇÃO

Para que seja possível sabermos o valor de alguma grandeza física do mundo real, precisamos utilizar sensores que convertam essas grandezas de interesse em grandezas mais facilmente manipuláveis, como, por exemplo, tensão. Estes dispositivos que transformam um tipo de energia em outro também são definidos como transdutores.

Normalmente, desejamos trabalhar digitalmente com esta variável, pois assim um conversor A/D pode interpretar e transformar a entrada na grandeza correspondente em que se deseja visualizar. Para isso, precisamos condicionar o sinal de entrada na faixa de alimentação específica do conversor A/D utilizado, evitando assim que o sinal ultrapasse o valor de tensão máxima suportado pelo conversor.

Para condicionar o sinal de entrada é necessário realizar amplificações ou atenuações no mesmo, isto geralmente é feito através de amplificadores operacionais com diversas configurações, dependendo do caso.

Além disso, também é feita uma filtragem dos ruídos indesejados e dos efeitos do *aliasing*. O *aliasing* é um efeito onde as réplicas do espectro digital do sinal se sobrepõem, impossibilitando a recriação do mesmo. Usualmente também são utilizados circuitos de proteção para garantir que picos de tensão não causem danos para o conversor A/D.

## CAPÍTULO 1 OBJETIVOS

O objetivo geral deste trabalho é projetar um sistema composto por circuitos de instrumentação, utilizando amplificadores operacionais e um microcontrolador, que realize a emulação de sistemas lineares.

Objetivos específicos:

- Projetar os circuitos de ganho/atenuação, *offset*, proteção e filtro para conversores A/D e D/A;
- Implementar a dinâmica do sistema projetado fisicamente, em um microcontrolador, a partir do sinal de entrada lido no conversor A/D.

### 1.1 Motivação

A motivação deste trabalho foi desenvolver um sistema de medição de temperatura para que seja possível o controle de uma estufa a fim de analisar a influência da temperatura no carregamento/descarregamento de baterias. Para este propósito, o presente relatório tem por finalidade desenvolver um sensor cujos valores devem ser adquiridos por um microcontrolador para, em outro projeto, realizar o controle em malha fechada da estufa. Portanto, este relatório apresenta todo o condicionamento do sinal que será fornecido ao microcontrolador.

## CAPÍTULO 2    DESENVOLVIMENTO

### 2.1    Especificações do projeto

O presente projeto tem por objetivo realizar a leitura de temperatura do ambiente. Para que isto fosse realizado, foi utilizado um sensor LM35 e um microcontrolador ATmega328 presente em uma placa de prototipagem Arduino Mega. Um circuito de instrumentação condiciona o sinal de entrada a fim de adequá-lo aos níveis de tensão do conversor A/D do microcontrolador, de 0 V a 5 V.

#### 2.1.1    Lista de componentes

- 1 sensor de temperatura LM35
- 3 amplificadores operacionais LM741
- 2 diodos
- 1 capacitor de 33nF
- 1 resistor de 61.6 k $\Omega$  (Utilizado um 1 resistor de 56 k $\Omega$  e 1 resistor de 5.6 k $\Omega$  em série)
- 6 resistores de 10 k $\Omega$
- 1 resistor de 1,41 k $\Omega$  (Utilizado 1 resistor de 1.5 k $\Omega$  e 1 resistor de 1 resistor de 33 k $\Omega$  em paralelo)
- 1 microcontrolador ATmega328

#### 2.1.2    Especificações do sensor LM35

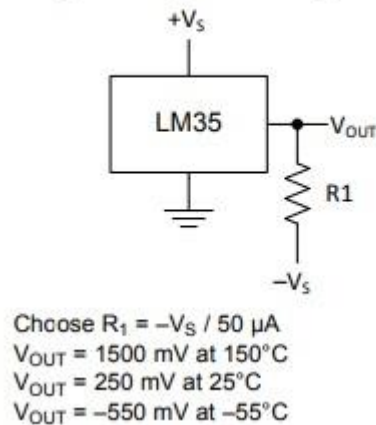
A Tabela 1 mostra a especificação do sensor de temperatura LM35.

**Tabela 1 - Especificações do LM35**  
**Especificações do sensor LM35**

Alcance de medição	-55 a 150 °C
Fator de escala linear	10 mV/°C
Precisão	0.5 °C (a 25°C)
Faixa de operação	4 a 30 V

O sensor LM35 trabalha em uma faixa básica de temperatura que vai de 2° a 150° C, ou em sua faixa completa que vai de -55 a 150 °C. Neste trabalho optou-se por utilizar a configuração de sua faixa completa, conforme a Figura 2-1.

**Full-Range Centigrade Temperature Sensor**

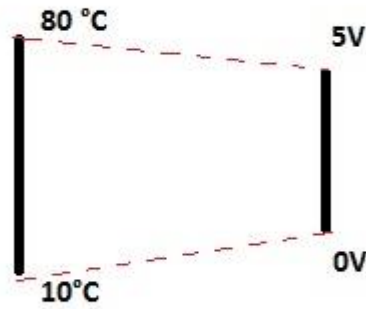


**Figura 2-1 – Configuração de faixa completa do LM35 (retirado do *datasheet*)**

Neste caso, foi escolhido  $V_S$  como 5 V de forma a ser alimentado pelo próprio microcontrolador. Deste modo,  $R_1$  foi projetado conforme Eq. 1.

$$R_1 = -\frac{V_S}{50\mu} = -\frac{5}{50\mu} = 100k\Omega$$

Além disso, optou-se por alterar a faixa de medição de temperatura apenas para valores positivos que estarão entre 10°C e 80°C, representando a faixa de operação escolhida para a estufa. Como o Arduino não pode receber valores de tensão negativa, esta faixa de operação acaba por proteger suas entradas, assim como a parte do controle que deverá evitar que a estufa atinja valores menores que 10°C. A Figura 2-2 mostra a relação entre a faixa de temperatura do sensor e o nível de tensão que entrará no microcontrolador.

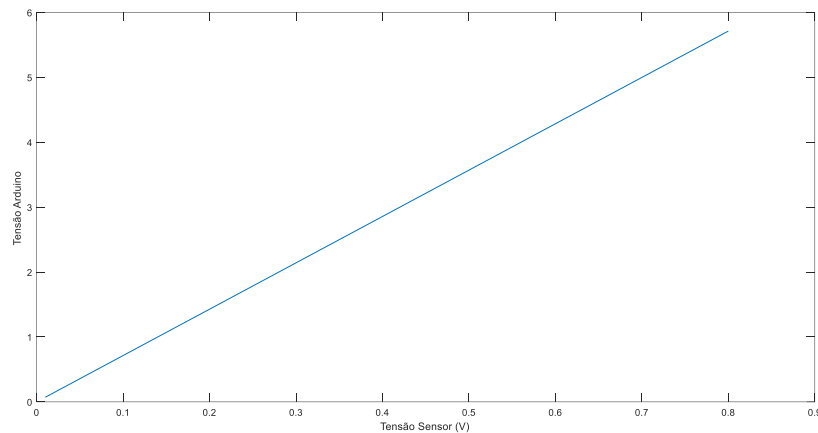


**Figura 2-2 – Relação entre temperatura e tensão do microcontrolador**

Definida a faixa de operação do projeto, é possível calcular os valores de ganho e *offset* do circuito analógico de instrumentação. De posse do fator de escala definida na Tabela anterior, onde a mínima variação da temperatura corresponde a um valor de tensão de 10 mV calculamos o ganho que deve ser dado no circuito conforme Eq. 2 a seguir.

$$a = \frac{V_{ard_{máx}} - V_{ard_{min}}}{\left(80^{\circ}C * \frac{10mV}{^{\circ}C}\right) - \left(10^{\circ}C * \frac{10mV}{^{\circ}C}\right)} = \frac{5 - 0}{800m - 100m} = 7,1429$$

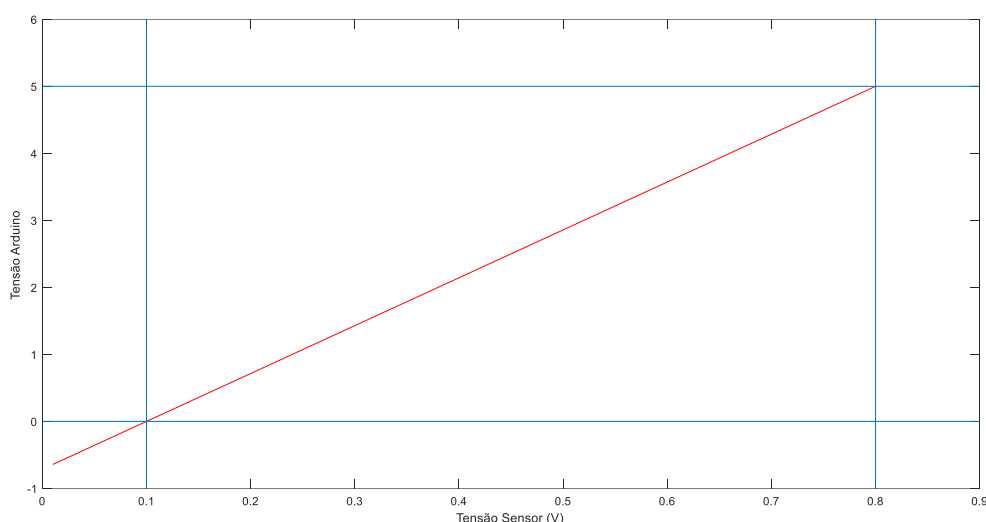
Onde Vard\_max corresponde a tensão máxima que o Arduino suporta, Vard\_min a tensão mínima e ‘a’ o valor do ganho. A Figura 2-3 representa a curva tensão do sensor pela tensão do Arduino, com o respectivo ganho.



**Figura 2-3 – Curva tensão sensor x tensão Arduino sem *offset***

Já o *offset* de tensão deve ser dado conforme a Eq. 3. Sendo necessário devido ao fato que o sensor inicia a leitura de temperatura disponibilizando em sua saída um valor negativo, que não é suportado pelo conversor A/D. Com o *offset* desejamos que a tensão de 0V corresponda a 10°C, e temos que para cada grau de variação a saída varia 10mV. A Figura 2-4 representa o valor do sinal com o ganho e o offset dado no sinal.

$$b = -a * \left( 10^{\circ}\text{C} * \frac{10\text{mV}}{^{\circ}\text{C}} \right) = -0,71429$$



**Figura 2-4 – Curva tensão sensor x tensão Arduino com *offset***

O ganho encontrado para o a será referente a primeiro estágio de instrumentação, no qual foi utilizado um circuito para ganho não-inversor. Neste definimos o Resistor R2 com um valor de 10kΩ e então calculamos o valor de R1 a partir do ganho a, de R2 e da equação referente ao não-inversor. Obtendo um valor de 61,42 kΩ, para o R1, conforme mostrado na Figura 2-5.

O sinal de saída do circuito não-inversor irá ser somado ao *offset*, que foi gerado utilizando um inversor com ganho de 0,7142 V, neste foi definido um valor de 10 kΩ para o resistor R1 e posteriormente aplicado na equação de ganho do circuito e encontrado um valor de 1,42 kΩ, esse circuito é apresentado na Figura 2-6. A partir do *offset* e ganho iremos somar os dois, como demonstrado na Figura 2.7. Após a soma com o *offset* o sinal já está pronto para ser convertido para dados pelo A/D, no entanto é importante realizar a proteção do circuito e filtragem de ruídos.



A proteção foi realizada na simulação com um *rail-to-rail*, porém na prática foram implementados dois diodos em anti-série. A filtragem é realizada por um filtro do tipo RC passa-baixa, com frequência de corte em 4.8kHz, que também é apresentado na Figura 2.7.

## 2.2 Condicionamento do sinal de entrada

O circuito de entrada do microcontrolador pode ser visto nas Figuras 2-5, 2-6 e 2-7. O circuito é composto pelos seguintes estágios: ganho, *offset*, proteção e filtro *anti-aliasing*. No primeiro estágio do circuito é dado um ganho de 7 vezes no sinal de saída do LM35. Após este ganho, é realizado um *offset* de 0.7 V a fim de trabalhar apenas com valores de tensão dentro da faixa de 0 a 5 V, suportado pela entrada A/D do microcontrolador. O estágio seguinte é composto por dois diodos de 5 V em anti-série para garantir que o sinal não ultrapasse o valor máximo de tensão suportado pela porta A/D do microcontrolador. Por último, é implementado um filtro *anti-aliasing* com a frequência de corte na metade da frequência de amostragem.

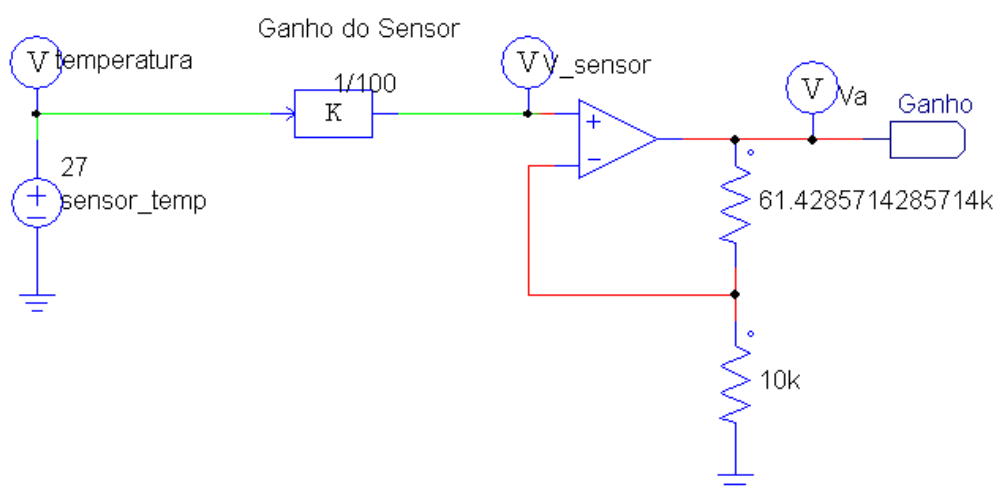


Figura 2-5 – Ganho no sinal de saída do sensor LM35

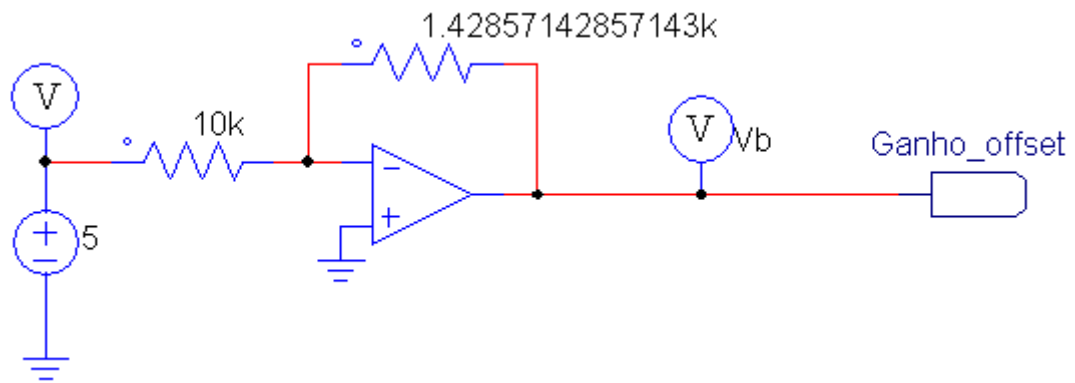


Figura 2-6 – Offset

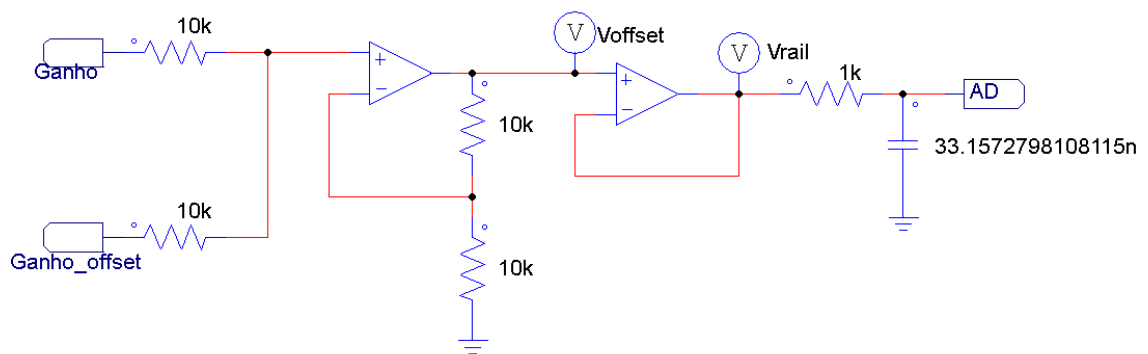


Figura 2-7 – Somador, proteção com amp. op. rail-to-rail e filtro anti-aliasing

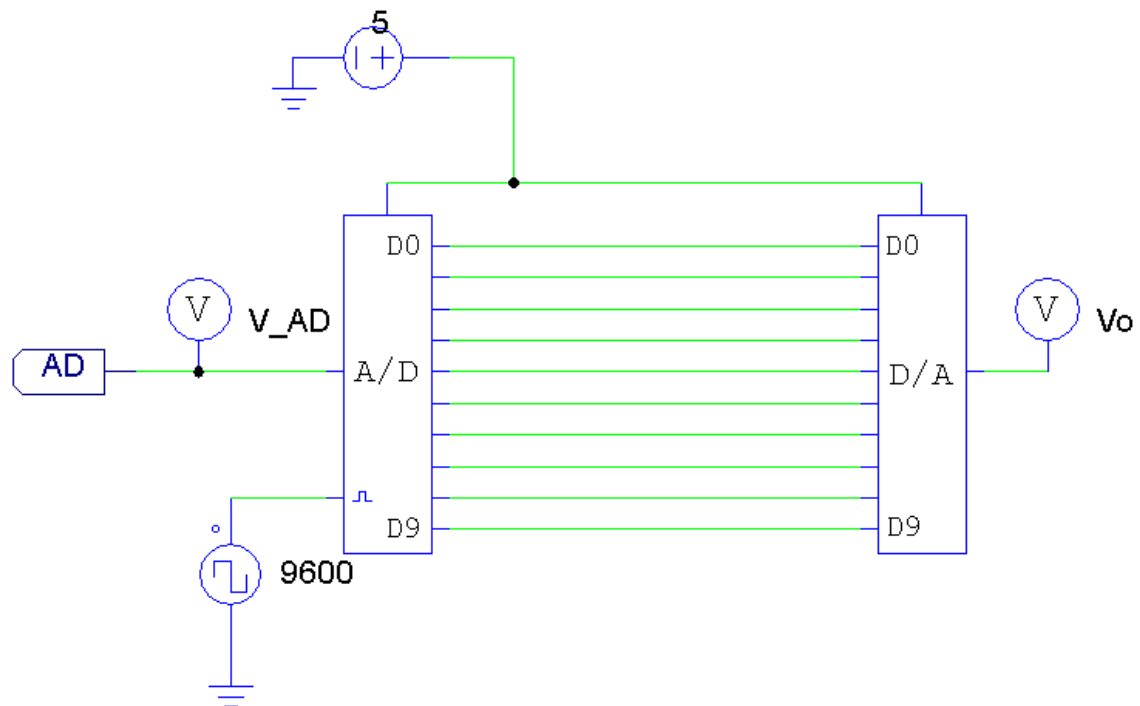


Figura 2-8 – Simulação do A/D

## CAPÍTULO 3 VISUALIZAÇÃO DOS RESULTADOS

### 3.1 Resultados de simulação

Neste capítulo serão apresentados os resultados obtidos a partir de simulação no *software* PSIM.

Na Figura 3-1 a seguir, é mostrado o valor de tensão na entrada do conversor A/D do microcontrolador para a temperatura ambiente de 10 °C.

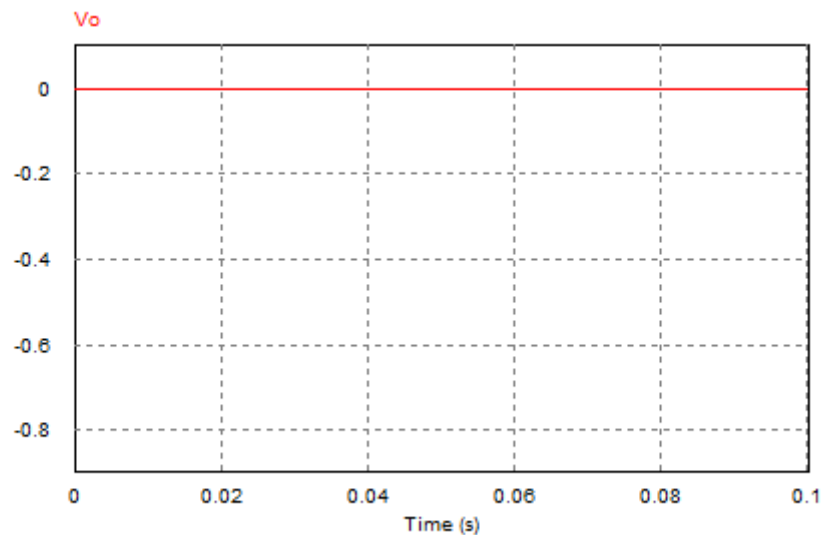
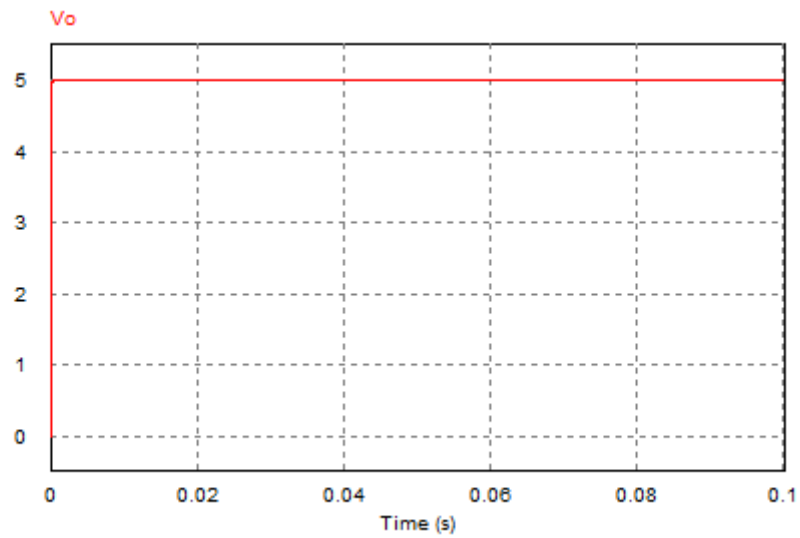


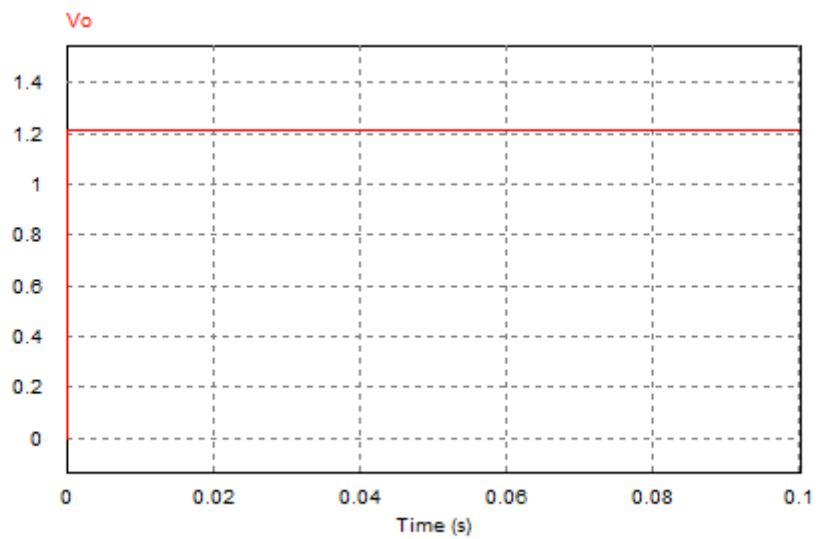
Figura 3-1 – Valor de tensão de entrada do conversor A/D para temperatura de 10°C

Na Figura 3-2 a seguir, é mostrado o valor de tensão na entrada do conversor A/D do microcontrolador para a temperatura ambiente de 80 °C.



**Figura 3-2 – Valor de tensão de entrada do conversor A/D para temperatura de 80 °C**

Na Figura 3-3 a seguir, é mostrado o valor de tensão na entrada do conversor A/D do microcontrolador para a temperatura ambiente de 27 °C.



**Figura 3-3 – Valor de tensão de entrada do conversor A/D para temperatura de 27 °C**

### 3.2 Leitura e visualização dos dados

Após o condicionamento do sinal emitido pelo LM35 faz-se necessária a visualização dos dados, de forma que torne possível comparar com os valores reais. Para isso ele foi lido pelo conversor A/D de um Arduino Mega, também é necessário a escolha de uma das entradas analógicas do microcontrolador, neste projeto escolhemos a entrada analógica 15.

O Conversor A/D possui 10 bits de resolução, tendo assim 1024 diferentes resultados de temperatura possíveis, sendo estes lidos através da função `analogRead()` do dispositivo. Posteriormente os ganhos inseridos pelo circuito de instrumentação são removidos possibilitando a reconstrução do sinal emitido pelo sensor de temperatura. Esse processo é realizado pela função definida na Figura 3-4 abaixo, nela podemos ver que a resolução e a tensão de referência do Arduino também são consideradas, visto que as mesmas são introduzidas devido ao ganho que o conversor A/D insere no sistema. Os demais ganhos são provenientes do circuito de instrumentação, sendo o valor de 0.714285 referente ao *offset* necessário para condicionar a leitura do valor do sensor pelo Arduino a partir de 0° C. Temos também o valor de 7.142857 correspondente a parametrização da reta para que a temperatura inicial fosse 0° C em 0 V. Posteriormente a tensão de saída do sensor deve ser multiplicada por 100 para corresponder a temperatura indicada.

```

vtemp = analogRead(A15);
vtemp = ((vtemp * 5/1023)+0.7142857142857143)*1/7.142857142857142;
temp = vtemp*100;

```

**Figura 3-4 – Conversão do sinal lido para temperatura**

Outra questão que foi realizada foi a leitura dos dados do sensor através de um tempo definido, para realizar essa leitura em tempos iguais realizamos uma interrupção no código a cada determinado tempo. Para este trabalho utilizou-se da biblioteca *TimerOne*, que possibilita a utilização de interrupções no Arduino, na nossa aplicação utilizamos ela conforme está definida na Figura 3-5, nela temos a definição da frequência de amostragem através da definição do *initialize*, também definimos a função que será chamada quando o tempo da interrupção chegar.

```

Timer1.initialize(104);
Timer1.attachInterrupt(timer_isr, 104);

```

**Figura 3-5 – Definição do comportamento da interrupção**

Após a conversão do sinal para um valor de temperatura ele deve ser mostrado, para que seja possível avaliar sua veracidade. A visualização foi realizada através do monitor serial do Arduino, os valores podem ser bem visualizados no vídeo feito com os valores sendo alterados, bem como um comparativo com a temperatura externa. E na figura 3-6 abaixo, encontra-se o código completo implementado no *Arduino*.

---

```

float vo, vtemp, temp;

void timer_isr()
{
    vtemp = analogRead(A15);
    vtemp = ((vtemp * 5/1023)+0.7142857142857143)*1/7.142857142857142;
    temp = vtemp*100;

    Serial.print("Temperature: ");
    Serial.print(temp);
    Serial.print(" - Sensor Voltage: ");
    Serial.println(vtemp);
}

void setup() {
    Serial.begin(38400);
    Timer1.initialize(104); // Inicializa o Timer da interrupção (microsegundos)
    Timer1.attachInterrupt(timer_isr, 104); // Atribui a função teste ao Interrupt
    pinMode(A15, INPUT);
}

void loop() {
}

```

**Figura 3-6 – Código completo**

### 3.3 Montagem do circuito de condicionamento na protoboard

Na figura 3-7 podemos verificar o circuito de condicionamento do sinal do sensor LM35 para o conversor A/D do microcontrolador montado em uma protoboard. Todos os estágios deste projeto para o condicionamento estão destacados na imagem.

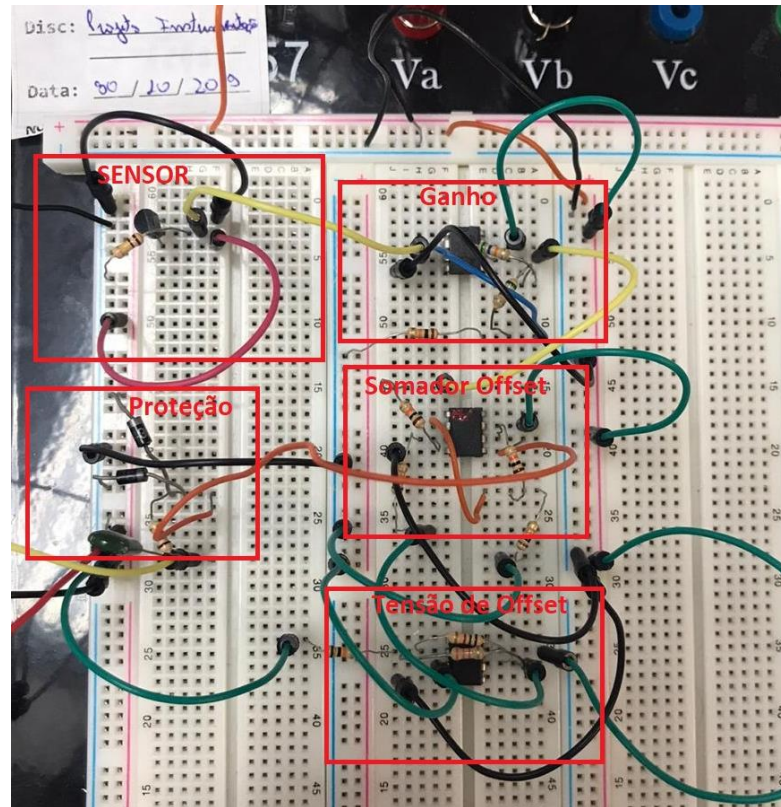


Figura 3-7 - Circuito de condicionamento de sinal montado na prática.

## **CAPÍTULO 4      CONCLUSÃO**

Este relatório descreveu a análise, projeto e implementação do trabalho final da disciplina de instrumentação eletrônica, que consistiu em desenvolver um sistema analógico de condicionamento de sinal de um sensor de temperatura para um microcontrolador Arduino. Deste modo, foram realizados conceitos de ganho/atenuação, offset, proteção e filtragem do sinal dentro dos critérios de amostragem, de forma a inseri-lo no microcontrolador com segurança e sem perdas de precisão do sinal. Feito isto, foi necessário reconstruir o sinal original do sensor dentro do microcontrolador a fim de mostrar qual valor de temperatura estava sendo lida no presente momento.

O projeto atendeu todos os critérios propostos, e abordou diversos conteúdos estudados durante a disciplina permitindo sua realização na prática, o qual possibilitou a verificação e contorno de problemas referentes aos componentes não ideais empregados.



## **CAPÍTULO 5    REFERÊNCIAS BIBLIOGRÁFICAS**

<https://www.alldatasheet.com/datasheet-pdf/pdf/517588/TI1/LM35.html>

## **ANEXO A**

Este anexo apresenta um link do youtube com o projeto em funcionamento.

<https://www.youtube.com/watch?v=-c5AQC8Dizc>