

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA DE CIÊNCIAS EXATAS E DA COMPUTAÇÃO
GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO



CONTROLE DE TEMPERATURA E UMIDADE INFORMATIZADO

GABRIEL ANGELO GUILIANE MENDES DE ALMEIDA

GOIÂNIA
2020

GABRIEL ANGELO GUILIANE MENDES DE ALMEIDA

CONTROLE DE TEMPERATURA E UMIDADE INFORMATIZADO

Trabalho de Conclusão de Curso apresentado à Escola de Ciências Exatas e da Computação, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador(a):

Prof. Me. Gustavo Siqueira Vinhal

Banca examinadora:

Prof. Me. Fernando Gonçalves Abadia

Prof. Me. Rafael Leal Martins

GABRIEL ANGELO GUILIANE MENDES DE ALMEIDA

CONTROLE DE TEMPERATURA E UMIDADE INFORMATIZADO

Trabalho de Conclusão de Curso aprovado em sua forma final pela Escola de Ciências Exatas e da Computação, da Pontifícia Universidade Católica de Goiás, para obtenção do título de Bacharel em Engenharia de Computação, em ____/____/_____.

Orientador(a): Gustavo Siqueira Vinhal

Prof. Me. Ludmilla Reis Pinheiro dos Santos
Coordenador(a) de Trabalho de Conclusão de Curso

GOIÂNIA
2020

A Deus pela minha vida e oportunidades de crescimento.

Aos meus familiares

Aos meus pais e companheira, pela paciência, orientações e apoio.

AGRADECIMENTOS

Agradeço a Deus e aos Orixás por sempre me mostrar o caminho certo.

Sou grato aos meus pais Abraão de Almeida e Andreia Guiliane pelo incentivo aos estudos e pelo apoio incondicional.

À minha querida namorada Hazel Mae pelo seu amor incondicional e me ensinar o verdadeiro significado de dedicação.

A todos os mestres que contribuíram com a minha formação acadêmica e profissional durante a minha vida.

Grato pela confiança depositado pelo meu orientador Gustavo Vinhal que dedicou inúmeras horas para sanar as minhas questões e me colocar na direção correta.

À Pontifícia Universidade Católica De Goiás – PUC GOIÁS e todos os seus professores que sempre proporcionaram um ensino de alta qualidade.

“A tecnologia é só uma ferramenta. No que se refere a motivar as crianças e conseguir que trabalhem juntas, um professor é o recurso mais importante.”

Bill Gates

RESUMO

Dispositivos integrados com microcontroladores podem resolver problemas através das informações disponibilizadas. Tornando processos industriais e residenciais mais eficientes. Dentro do setor produtivo se encontram as empresas do ramo da agricultura. Este segmento contribui de forma significativa com o avanço socioeconômico do Brasil, mas ele enfrenta sérios problemas. Entre eles, está a ausência do monitoramento da temperatura na maioria dos silos e armazéns das microempresas deste setor, devido a questões financeiras e falta de mão de obra especializada. Este trabalho de pesquisa tem como objetivo disponibilizar um sistema microcontrolado inteligente de baixo custo capaz de indicar as atitudes necessárias para conduzir de forma eficiente à produção e armazenamento dos diversos produtos armazenados. Para isso, o sistema traça um perfil da temperatura no interior do local desejado durante alguns processos de armazenamento e movimentação através da visualização e registro dos valores da grandeza. É apresentado o projeto de medição de temperatura já funcionando com leituras a curta e longa distância, variando conforme a aplicação e local desejado.

***Palavras-Chave:** Arduino, Sinterização, Monitoramento, Sensores, Agricultura.*

ABSTRACT

Devices integrated with microcontrollers can solve problems through the information provided. Making industrial and residential processes more efficient. Within the productive sector are companies in the agricultural sector. This segment contributes significantly to Brazil's socioeconomic advance, but it faces serious problems. Among them, there is the absence of temperature monitoring in most of the micro-company warehouses and warehouses in this sector, due to financial issues and a lack of specialized labor. This research work aims to provide a low cost intelligent microcontrolled system capable of indicating the necessary attitudes to efficiently conduct the production and storage of the various stored products. For this, the system traces a temperature profile inside the desired location during some storage and movement processes through the visualization and recording of the magnitude values. The temperature measurement project is already working, with readings at short and long distance, varying according to the application and desired location.

Keywords: *Arduino, Sintering, Monitoring, Sensors, Agriculture.*

LISTA DE ILUSTRAÇÕES

Figura 1 - Modelos de Arduino.	19
Figura 2 - Estrutura Arduino Uno ATmega323.	23
Figura 3 - Shield Relé 5V	24
Figura 4 - Pinagem do módulo relé.	25
Figura 5 - Nanoshield Termopar.	26
Figura 6 - Fluxo de funcionamento.	27
Figura 7 - Conexão cabos Termopar.	28
Figura 8 - Termopar e Relé detalhamento.	29
Figura 9 - Interface IDE	30
Figura 10 - Distribuição dos pontos de leitura.	31
Figura 11 - Sistema com todos módulos conectados.	31
Figura 12 – Resultado da medição de temperatura e acionamento relé.	32

LISTA DE TABELAS

Tabela 1 - Características do Arduino	23
Tabela 2 - Características do Shield Relé.	25

LISTA DE SIGLAS

A	O ampere é uma unidade Ampère.
AREF	é o conversor analógico-digital
AVR	Microcontrolador de chip único com uma arquitetura Harvard
C	Linguagem de programação de alto nível para computadores
CA	Corrente alternada
CC	Corrente continua
EEPROM	<i>Electrically Erasable Programmable Read-Only Memory</i>
GND	<i>Graduated Neutral Density Filter</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
ICSP	<i>In Circuit Serial Programming</i>
IDE	<i>Integrated Development Environment</i>
KB	Quilobyte (Símbolo: kB) é um múltiplo de uma unidade byte
K Ω	Ohm - unidades usadas para medir a resistência.
LCD	<i>Liquid Crystal Display</i>
LED	<i>Light Emitting Diode</i>
MA	Símbolo de miliampere
MHZ	É uma unidade de frequência
MISO	<i>Master In Slave Out</i>
MM	Símbolo de milímetro
MOSI	<i>Master Out Slave In</i>
NA	Normalmente Aberto
NF	Normalmente Fechado
PWM	<i>Pulse-Width Modulation</i>
RX	Pino de recepção do Arduino
SCK	<i>Serial Clock</i>
SCL	Clock de sincronização
SDI	<i>Slave Data IN</i> - Pino de dados de entrada;
SPI	<i>Serial Peripheral Interface</i>
SRAM	<i>Static Random Access Memory</i>
SS	<i>Slave Select.</i>
TTL	<i>Transistor–transistor logic</i>

TWI	I2C, também conhecido como TWI (<i>Inter-Integrated Circuit</i>)
TX	Pino de transmissão do Arduino
USB	<i>Universal Serial Bus</i>
V	O volt (símbolo: V) é a unidade de tensão elétrica
VCC	Alimentação positivo de voltagem
VIN	Local onde é feita a alimentação de entrada

SUMÁRIO

1	INTRODUÇÃO	14
1.1.	Motivação	14
1.2.	Objetivo Geral	15
1.3.	Objetivos Específicos	15
1.4.	Estrutura do Trabalho	15
2	FUNDAMENTAÇÃO TEORICA	16
2.1.	Contexto histórico	16
2.2.	Armazenamento de grãos	16
2.3.	Tecnologia integrada a necessidade	17
3	MATERIAIS UTILIZADOS	18
3.1.	Plataforma Arduino	18
3.2.	Shield Arduino Relé 5V – 2 Canais	24
3.3.	Nanoshield Termopar	25
4	DESENVOLVIMENTO	27
4.1.	Fluxo de Funcionamento	27
4.2.	<i>Software</i> IDE Arduino	29
4.3.	Testes com o Protótipo	30
5	CONSIDERAÇÕES FINAIS	33
6	REFERÊNCIAS BIBLIOGRÁFICAS	34

1 INTRODUÇÃO

Um sistema de informação tem a capacidade de obter informações no momento e local desejado. Tais sistemas são projetados para tratar informações relevantes, como dados e informações. O Brasil é um dos grandes exportadores de grãos a nível mundial. São cultivados na agricultura brasileira milho, soja, feijão, entre outros.

Aliado ao acréscimo na produção, existem também a demanda por locais adequados para esses produtos por determinados períodos de tempo, com a necessidade de conservar as propriedades dos grãos colhidos. Uma vez que passaram pelo processo do plantio e colheita, é preciso armazenar, a fim de preservar a qualidade e as características que possa garantir o menor risco.

Apesar dessa produção estar crescendo, sua capacidade de armazenagem não teve um desenvolvimento no mesmo ritmo. Isso faz com que os produtores não possam armazenar seus grãos para a venda na entressafra, onde o preço tem seu aumento proporcional, fazendo assim, com que o produtor sem o devido local e o devido monitoramento obtenha um lucro menor, tendo que vender o grão durante a safra.

Porém, um dos grandes desafios se concentra no armazenamento destes grãos. É importante que os locais tenham uma temperatura específica para evitar perdas e, conseqüentemente, prejuízo.

O monitoramento de temperatura, atualmente, é feito de forma manual na grande maioria dos locais. Isso é prejudicial, pois a interferência humana está sujeita a erros. A demora da aferição pode levar a tomada de decisão tardia e, por conseguinte, perdas.

Aferir a temperatura de forma automatizada pode trazer benefícios ao produtor rural, uma vez que erros de leitura serão minimizados e o tempo de resposta maximizado. Portanto, este projeto visa sanar problemas na aferição de temperatura em locais de armazenamento de grãos, podendo ser analisados e tratados ainda a tempo. Assim, evitando proporcionalmente a perda por excesso ou falta de temperatura.

1.1. Motivação

A criação de um sistema de monitoramento automatizado de baixo custo não visa somente à agricultura, como também os hospitais que necessitam de um sistema de

monitoramento. O desenvolvimento de um sistema de medição de temperatura utilizando a plataforma Arduino é de grande utilidade pois, além da facilidade da utilização e acesso a essa tecnologia, possibilita também o desenvolvimento de um sistema preciso e com um custo muito baixo, promovendo a oportunidade de todos usufruírem os benefícios do sistema.

Os produtores com menor poder aquisitivo que sofrem com a venda na safra, não na entressafra terem a oportunidade de crescer proporcionalmente fazendo o uso do monitoramento constante de seus grãos já colhidos, se destacando como os concorrente de forma natural.

1.2. Objetivo Geral

O objetivo deste trabalho é disponibilizar um sistema microcontrolado inteligente de baixo custo capaz de indicar as atitudes necessárias para conduzir de forma eficiente à produção e armazenamento dos diversos produtos armazenados.

1.3. Objetivos Específicos

- Efetuar aferições de temperatura em vários locais.
- Permitir acesso a produtores de menor poder aquisitivo.
- Possibilitar a inclusão de mais placas, possibilitando o acesso remoto via internet.
- Integrar várias áreas de negócios como frigoríficos.

1.4. Estrutura do Trabalho

Este trabalho está organizado como segue. No Capítulo 2 será apresentado a fundamentação teórica para o trabalho, conceituando a sua aplicação no ramo da agricultura, referente ao monitoramento dos grãos armazenados em silos e armazéns graneleiros. No Capítulo 3 os materiais e métodos utilizados. No Capítulo 4 apresenta o protótipo final produzido. Por fim no Capítulo 5, as considerações finais do trabalho.

2 FUNDAMENTAÇÃO TEORICA

2.1. Contexto histórico

O agronegócio é uma atividade rural de desenvolvimento agropecuário e é uma rede econômica entre o comércio e a indústria. Com o decorrer do tempo a agricultura Brasileira enfrentou diversas etapas, como por exemplo, época colonial onde a economia se resumia na produção de cana-de-açúcar. Após um breve período se expandiu para o café e soja. Até os tempos atuais o agronegócio e a agricultura sofrem mudanças e muitas melhorias, como a implantação e uso de tecnologias a área do agronegócio e agricultura.

Com a modernização surgiram as máquinas que auxiliaram na mão de obra humana, fazendo com que o processo de produção rendesse e aumentasse proporcionalmente aos longos dos anos. Essas transformações, colaboraram com a economia do país, tornando o Brasil tão conhecido por sua exportação para os demais países.

Junior,(2007), afirma que “o conceito de agricultura vem sendo modificado ao longo dos anos”, onde até o início dos anos 60 a agricultura era vista como um processo de arar, plantar e colher, além do cuidado com animais, porém atualmente houve uma mudança radical nesse conceito com a influência da industrialização, havendo a modernização nos processos de plantar, colher e vender, com isso a agricultura passou a ser um grande negócio financeiro, antes se o homem plantava, cultivava e colhia hoje as máquinas fazem isso e produzem muito mais, trazendo um maior retorno financeiro.

O Brasil é o maior exportador de soja do mundo atualmente e esta produção depende de condições climáticas desde o plantio até o armazenamento após a colheita.

2.2. Armazenamento de grãos

Após o plantio e colheita do grão, é necessário realizar a armazenagem, o qual se baseia em guardar um volume de grãos por um determinado período. A armazenagem é feita de duas maneiras: a granel onde os grãos estarão todos soltos em silos (metal, concreto entre outros) e em volumes, o produto é embalado em sacos e empilhado, normalmente em galpões.

Todos esses produtos estão sujeitos a condições climáticas e aos predadores naturais, necessitando assim, que após a safra o produto seja monitorado tanto da temperatura em que se encontra para evitar a proliferação de fungos e bactérias, quanto das condições do ambiente em que foram alocados os grãos.

2.3. Tecnologia integrada a necessidade

A tecnologia e relacionada a integração das diversas áreas industriais, vem sendo amplamente utilizada no setor comercial, forçando cada vez mais a integração tecnológicas. Pode ser definida como "O conjunto de serviços proporcionados por sistemas tecnológicos integrados como o melhor meio de satisfazer as necessidades básicas de segurança, comunicação, gestão energética e conforto de uma habitação" (OLIVEIRA, 2005).

Segundo Azevedo et al. (2008, p. 2) “[...] o Brasil é um dos países que mais se destaca no cenário mundial da agricultura, devido à sua crescente expansão na produção de grãos”. Este fato, trata com grande evidência a importância do país para a produção de alimentos, logo a sua distribuição para os demais países.

Para que esses produtos sejam utilizados posteriormente é necessário realizar o processo de armazenagem, o qual se baseia em reunir e guardar um volume de grãos por um determinado período, a fim de que se possa preservar as características normais dos mesmos evitando fungos, pragas, umidade excessiva e entre outras (SILVA et al., 2012).

Após feita a armazenagem, os grãos precisam ser monitorados para que o produto não sofra ações diretas do clima e de predadores naturais. O monitoramento constante da temperatura, provocará naturalmente o aumento da durabilidade do produto armazenado evitando a perda em grande escala devido ao descontrole na temperatura ambiente.

Nos últimos anos, a preocupação com a capacidade estática de armazenamento vem aumentando devido a investimentos nessa área. Porém, a capacidade de armazenagem não consegue acompanhar a evolução da agricultura, que é favorecida por muitos fatores, como o avanço da tecnologia e a expansão das áreas plantadas (LEITE, 2013). Segundo Maia et al. (2013) “[...] os estados do Sul e do Sudeste são, aparentemente, os menos afetados por problemas relacionados à armazenagem”

Portanto, a utilização de tecnologias como auxílio para monitoramento dos grãos, traz muito benefícios ao ramo agroindustrial do país. O monitoramento da temperatura se faz necessária para manter esses grãos saudáveis para consumo.

Uma destas tecnologias de fácil acesso é de baixo custo são os sensores termopares, capazes de efetuar leituras e suportar grandes cargas de calor. Sendo possíveis a integração destas tecnologias através do uso de placas de livre programação, como são os Arduino.

3 MATERIAIS UTILIZADOS

3.1. Plataforma Arduino

A plataforma Arduino foi criada em meados de 2000 por um grupo de 5 pesquisadores: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis. O primeiro modelo do Arduino foi apresentado em 2005, com o intuito de auxiliar estudantes e outras pessoas com pouco ou nenhum conhecimento em programação e microcontroladores a aprenderem sobre eles. Ele adota o conceito de *hardware* livre, tornando-o acessível a todos. O Arduino possui vários modelos com construção e especificações diferentes que podem ser usados de acordo com a necessidade do projeto. A programação do Arduino é realizada através de uma IDE (*Integrated Development Environment*, ou Ambiente de Desenvolvimento Integrado) utilizando uma linguagem baseada em C/C++ (FONSECA; LA VEGA, 2011).

Os Arduinos possuem uma gama de modelos de placas, cada uma com diferenças de *hardware*. Alguns exemplos dessas placas são o Mega, Uno, Nano, Leonardo entre outros. A diferença entre as versões se concentra na quantidade de recursos que cada uma possui. A estrutura da placa funciona de forma simples e prática, com suas entradas e saídas sendo definidas no programa, assim como toda a comunicação e controle, além de não haver a necessidade de componentes externos pois a placa já vem pronta para o uso. O Arduino pode vir com diferentes microcontroladores dependendo do modelo.

O conceito do Arduino surgiu na Itália, em 2005, com o objetivo de criar um dispositivo para controlar projetos/protótipos construídos de uma forma menos onerosa do que outros sistemas disponíveis no mercado (FONSECA; LA VEGA, 2011).

Segundo Souza et al. (2011), o Arduino é ideal para o desenvolvimento de dispositivos capazes de interagir com o ambiente, através da utilização de sensores e atuadores. Os sensores podem medir valores de grandezas como: temperatura, luminosidade, chuva, distância, pressão, umidade etc. Já os atuadores podem agir sobre o meio no qual está inserido e modificá-lo. São

exemplos de atuadores: LED (*Light Emitting Diode*, ou Diodo Emissor de Luz), LCD (*Liquid Crystal Display*, ou Tela de Cristal Líquido), motores, servo motores e outros.

O Arduino é disponibilizado na forma de placas para o desenvolvimento de dispositivos diversos. É constituído por uma plataforma de computação física, onde sistemas digitais, ligados a sensores e atuadores, por meio de trilhas de cobre, formam uma placa de circuito impresso, podendo medir grandezas do meio físico, realizar cálculos numéricos e tomar decisões lógicas no ambiente computacional.

A Figura 1 apresenta alguns dos modelos de Arduino disponíveis que variam de acordo com a pinagem, condições de uso, custo e possíveis aplicações.

Figura 1 - Modelos de Arduino.



Fonte: <https://deinfo.uepg.br/~alunoso/2019/SO/ARDUINO/ARQUITETURA/index.html>

O Arduino Uno R3 é uma placa baseada no microcontrolador ATmega328. Ele tem 14 pinos de entrada/saída digital (dos quais 6 podem ser usados como saídas PWM (*Pulse Width Modulation*, ou Modulação de Largura de Pulso)), 6 entradas analógicas, um cristal oscilador de 16MHz (Unidade de frequência, equivalente a um milhão de Hertz ou mil *kilohertz*), uma

conexão USB (*Universal Serial Bus*), uma entrada de alimentação, uma conexão ICSP (*In Circuit Serial Programming* ou Programação Serial do Circuito) e um botão de *reset*.

O Arduino Uno R3 difere de todas as placas antecessoras no sentido de não utilizar o chip FTDI (*Future Technology Devices International*) para conversão do sinal serial. Utiliza no seu lugar um Atmega8U2 programado como conversor de USB para serial.

Existem dois outros pinos adicionados:

- IOREF, que permite aos *shields* (placas de propósito específico) se adaptarem à voltagem fornecida pela placa. No futuro os *shields* serão compatíveis tanto com as placas que utilizam o AVR (Microcontrolador de chip único com uma arquitetura Harvard) e operam a 5V, como com o ARduino Due que operará a 3,3V.
- O segundo pino não está conectado e é reservado para propósitos futuros.

O Arduino Uno pode ser alimentado pela conexão USB ou com uma fonte de alimentação externa. A alimentação é selecionada automaticamente. Alimentação externa (não USB) pode ser tanto de um adaptador CA (Corrente Alternada) para CC (Corrente Contínua) ou bateria. Há um conector para alimentação de 2,1mm (Unidade de medida, milímetro) com o positivo no centro. Cabos vindos de uma bateria podem ser inseridos diretamente nos pinos GND (*Graduated Neutral Density Filter*, ou Filtro Graduado De Densidade Neutra) e VIN (Local onde é feita a alimentação de entrada) do conector de alimentação.

Esta placa pode funcionar com uma fonte de alimentação externa de 6 a 20 volts. No entanto se a alimentação for inferior a 7V, o pino 5V pode fornecer menos de cinco volts e a placa pode se mostrar instável. E se a alimentação for maior do que 12V o regulador de voltagem pode superaquecer e danificar a placa. A faixa recomendada é de 7 a 12 volts.

Os pinos de alimentação são os seguintes:

- VIN: A entrada de alimentação para a placa Arduino quando se está utilizando uma fonte de alimentação externa (em oposição à conexão USB ou outra fonte de alimentação regulada). Você pode fornecer alimentação através deste pino, ou se estiver utilizando o conector de alimentação acessar esta voltagem aqui.
- 5V: A fonte de alimentação regulada usada para o microcontrolador e para outros componentes na placa. Pode vir tanto do VIN através do regulador embarcado ou da conexão USB ou outra fonte regulada em 5V.

- 3V3: Uma fonte de 3,3V gerada pelo regulador embarcado. A corrente máxima suportada é de 50mA (símbolo de miliampere).
- GND: Pinos terra.

Cada um dos 14 pinos digitais do Uno R3 podem ser utilizados como uma entrada ou uma saída utilizando-se as funções `pinMode()`, `digitalWrite()`, e `digitalRead()`. Eles operam a 5V. Cada pino pode fornecer ou receber um máximo de 40mA e tem um resistor *pull-up* interno (desconectado por padrão) de 20-50K Ω (Ohm e valores derivados dele, como kilohms (k Ω) e megahms (M Ω) são as unidades usadas para medir a resistência). Além disso alguns pinos têm funções especializadas:

- Serial: 0 (RX (Pino de recepção do Arduino)) e 1 (TX). Usados para receber (RX) e transmitir (TX (Pino de transmissão do Arduino)) dados seriais TTL (*Transistor-Transistor Logic*, Lógica transistor-transistor). Estes pinos são conectados aos pinos correspondentes do chip serial USB-para-TL ATmega8U2.
- Interruptores Externos: 2 e 3. Estes pinos podem ser configurados para disparar uma interrupção de acordo com alguma variação sensível pelo circuito.
- SPI (*Serial Peripheral Interface*, registro de deslocamento que serão ligados entre si por duas linhas de dados): 10 (SS (*Slave Select* - Seleciona qual *Slave* receberá os dados)), 11 (MOSI (*Master Out Slave In*, para os dados do programador)), 12 (MISO (*Master In Slave Out*, o que indica o pino usado para envio de dados)), 13 (SCK (*Serial Clock*, *Clock* de sincronização para transmissão de dados)). Estes pinos dão suporte à comunicação SPI utilizando a biblioteca SPI.
- LED: 13. Há um LED integrado ao pino digital 13. Quando este pino está no valor HIGH este LED está aceso, quando o pino está em LOW o LED está apagado.
- I2C: 4 (SDA) e 5 (SCL (*Clock* de sincronização)). Fornecem suporte a comunicação I2C (TWI (I2C, também conhecido como TWI, realiza uma comunicação *half duplex*, é possível transmitir e receber informações, mas não ao mesmo tempo)) utilizando a biblioteca Wire.
- AREF: (conversor analógico-digital). Voltagem de referência para as entradas analógicas. Utilizado com a função `analogReference()`.
- Reset: Envio o valor LOW para esta linha para resetar o microcontrolador. Tipicamente usado para adicionar um botão de reset para Shields montados sobre a placa original.

- O Arduino Uno tem 6 entradas analógicas, etiquetadas de A0 a A5, cada uma tem 10 bits de resolução (i.e. 1024 valores diferentes). Por padrão elas medem de 0 a 5V, embora seja possível alterar o limite superior utilizando o pino AREF e a função `analogReference()`.

O ATmega328 tem 32KB (Quilobyte (Símbolo: kB) é um múltiplo de uma unidade byte) (dos quais 0,5 são utilizados pelo *bootloader*). Também tem 2KB de SRAM (*Static Random Access Memory*, armazenamento temporário para as informações utilizadas pelo programa) e 1KB de EEPROM (*Electrically Erasable Programmable Read-Only Memory*, ou Memória Somente de Leitura Programável Apagável) (que pode ser lido ou gravado com a biblioteca EEPROM).

O Arduino Uno possui uma série de facilidades para se comunicar com um computador, outro Arduino ou outros microcontroladores. O ATmega328 fornece comunicação serial UART TTL (5V) que está disponível nos pinos digitais 0 (RX) e 1 (TX). Um ATmega8U2 na placa canaliza esta comunicação para a USB e aparece como uma porta virtual para o *software* no computador. O *firmware* utiliza os drivers padrão USB e nenhum driver externo é necessário. O *software* do Arduino inclui um monitor serial que permite dados textuais ser enviados e recebidos da placa. LEDs conectados ao RX e TX piscam enquanto dados estiverem sendo transmitidos pelo *chip* USB-para-serial e pela conexão USB (mas não para comunicação serial nos pinos 0 e 1).

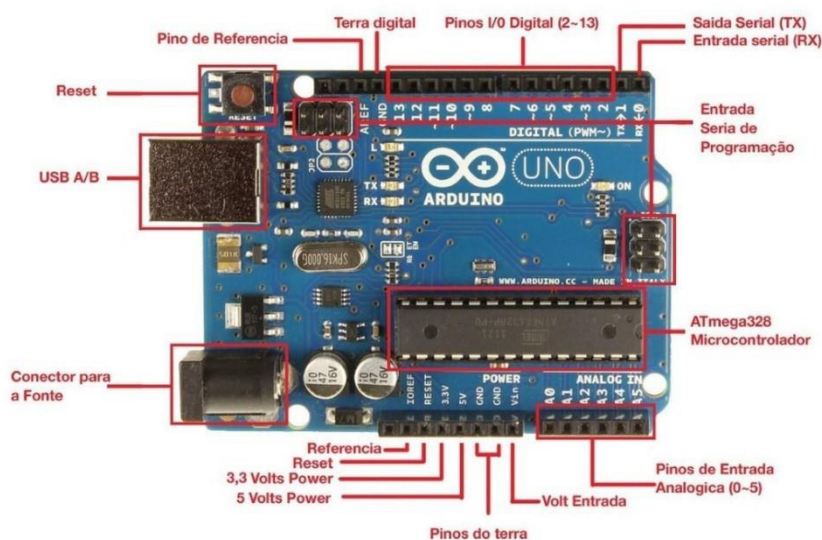
O Arduino Uno pode ser programado com o *software* Arduino. Simplesmente selecione "Arduino Uno" no menu Tools > Board. O ATmega328 no Arduino Uno vem pré-gravado com um *bootloader* que permite a você enviar código novo para ele sem a utilização de um programador de *hardware* externo.

O Arduino Uno possui um *polyfuse* resetável que protege a porta USB do seu computador contra sobre-corrente e curtos circuitos. Embora muitos computadores tenham sua própria proteção interna, o fusão fornece uma camada a mais de proteção. Se mais de 500mA forem aplicados a porta USB ele automaticamente irá interromper a conexão até que o curto ou a sobrecarga seja removido.

A largura e o comprimento máximos da placa do Uno são 68,58 e 53,34mm respectivamente (2,7" x 2,1"), com os conectores USB e de alimentação estendendo-se além destas dimensões. Quatro orifícios para parafusos permitem que a placa seja fixada a uma superfície ou encapsulamento.

A Figura 2 apresenta o Arduino utilizado para a realização deste projeto.

Figura 2 - Estrutura Arduino Uno ATmega323.



Fonte: <https://techsuleletronicos.com.br/product/arduino-uno-com-atmega328-r3/>

A Tabela 1 apresenta, de forma sumarizada, as características do Arduino UNO ATmega 323, utilizado neste trabalho.

Tabela 1 - Características do Arduino

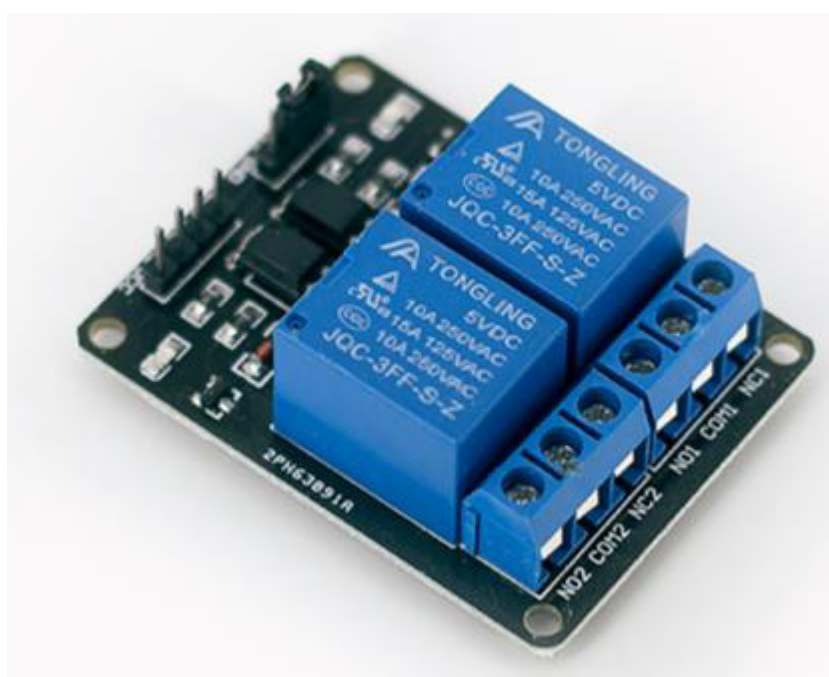
Microcontrolador	ATmega328
Tensão de operação	5V
Tensão de alimentação (recomendada)	7-12V
Tensão de alimentação (limite)	6-20V
Entradas e saídas digitais	14 das quais 6 podem ser PWM
Entradas analógicas	6
Corrente contínua por pino de I/O	40 mA
Corrente contínua para o pino 3.3V	50 mA
Memória <i>Flash</i>	32 KB (ATmega328) dos quais 0.5 KB são usados pelo <i>bootloader</i>
Memória SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Velocidade do <i>Clock</i>	16 MHz
Dimensões	68,58mm x 53,34mm
Peso	150g

Fonte: Elaborada pelo autor.

3.2. Shield Arduino Relé 5V – 2 Canais

O relé 5V pode ser usado junto ao Arduino Uno ATmega328 – R3. Usando dois pinos do módulo relé Arduino, é possível controlar cargas como lâmpadas, motores e sinais de leitura, desde que a corrente de operação não ultrapasse 10A (ampères). A Figura 3 apresenta um exemplo de Shild Relé de 5V.

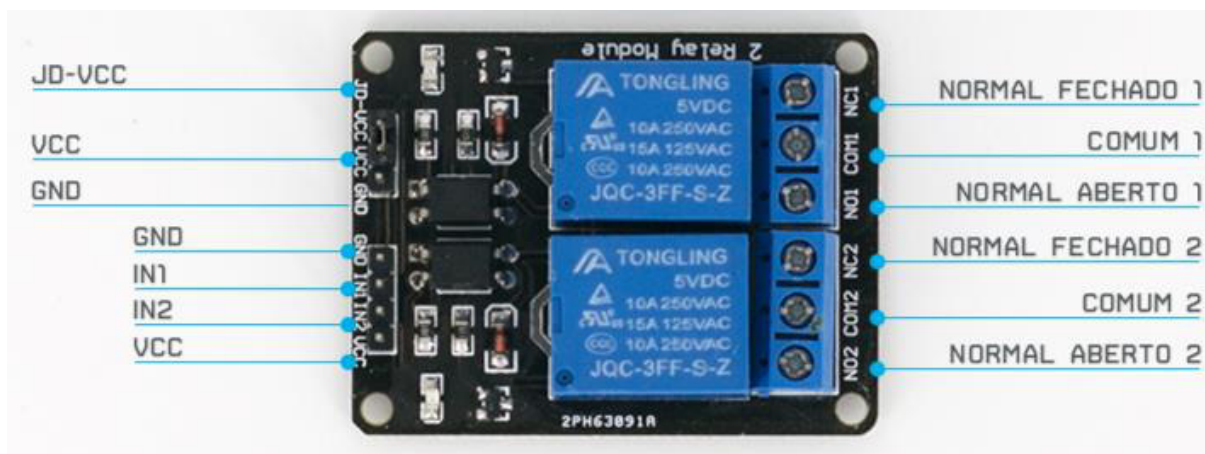
Figura 3 - Shield Relé 5V



Fonte: <https://www.filipeflop.com/blog/controle-modulo-rele-arduino/>

O Shield funciona como uma chave (interruptor). No borne cinza há 3 conexões: NA (Normalmente Aberto), C (Comum) e NF (Normalmente Fechado). Ou seja, quando o Módulo Relé estiver "desligado", C estará conectada à NF. Quando estiver ligado, C estará conectada à NA. Na Figura 4 é possível observar a pinagem do módulo relé Arduino. No lado esquerdo superior os pinos JD-Vcc (Alimentação positivo de voltagem), Vcc e GND, que permitem que seja conectada uma fonte externa de 5V. Abaixo, os pinos GND, IN1 (aciona o relé 1), IN2 (aciona o relé 2), e o Vcc. Ao lado dos relés, os contatos NC (Normalmente Fechado), C (Comum), e NA (Normalmente Aberto).

Figura 4 - Pinagem do módulo relé.



Fonte: <https://www.filipeflop.com/blog/control-modulo-rele-arduino/>

A Tabela 2 apresenta, de forma sumarizada, as características do Shield Relé utilizado neste trabalho.

Tabela 2 - Características do Shield Relé.

Modelo Relé	SRD-05VDC-SL-C
Tensão de operação	5V
Permite controlar cargas de	220V AC
Nível de sinal dos pinos IN1 e IN2	5 VDC
Corrente de operação	15 ~ 20 mA
Tempo de resposta	5 ~ 10 ms
Dimensões reduzidas	51 x 38 x 20 mm

Fonte: Elaborada pelo Autor.

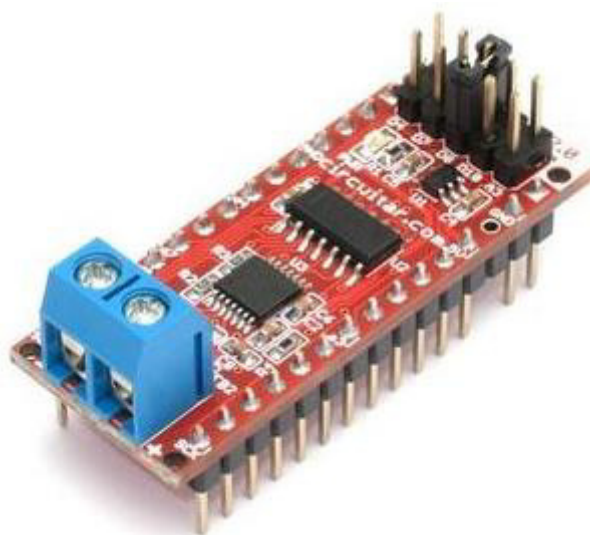
Além das placas, existe uma grande variedade de módulos e Shields para o Arduino assim como acessórios que aumentam ainda mais a variedade de aplicações do dispositivo como conexão com a internet, sensor termopar e módulo de relé.

3.3. Nanoshield Termopar

Com o Nanoshield Termopar é possível medir temperaturas por meio de um sensor de termopar, de forma prática e simples. Ele é implementado com o CI MAX31856, fabricado pela Maxim Semiconductor, que contém todo o circuito necessário para realizar esse tipo de medição: sensor interno de temperatura (junção fria), amplificador e conversor analógico-

digital. Dessa forma, a leitura já é dada diretamente em graus Celsius e é lida pelo microcontrolador através de uma interface SPI. A Figura 5 apresenta um exemplo do nanoshield Termopar utilizado neste trabalho.

Figura 5 - Nanoshield Termopar.



Fonte: <https://www.circuitar.com.br/nanoshields/modulos/termopar/index.html>

O Nanoshield Termopar possui um borne de parafuso onde devem ser fixadas as pontas do sensor de termopar. Termopares são sensores utilizados principalmente quando se necessita de uma ampla faixa de medidas e em temperaturas (ou ambientes) não suportados por outros tipos de sensores, como termistores ou sensores semicondutores. Também é possível obter respostas bastantes rápidas, dependendo do tipo e da geometria do sensor termopar utilizado. Como aplicações típicas tem-se medição em fornos, turbinas, motores, processos industriais (siderurgia, usinagem, etc.), entre outros.

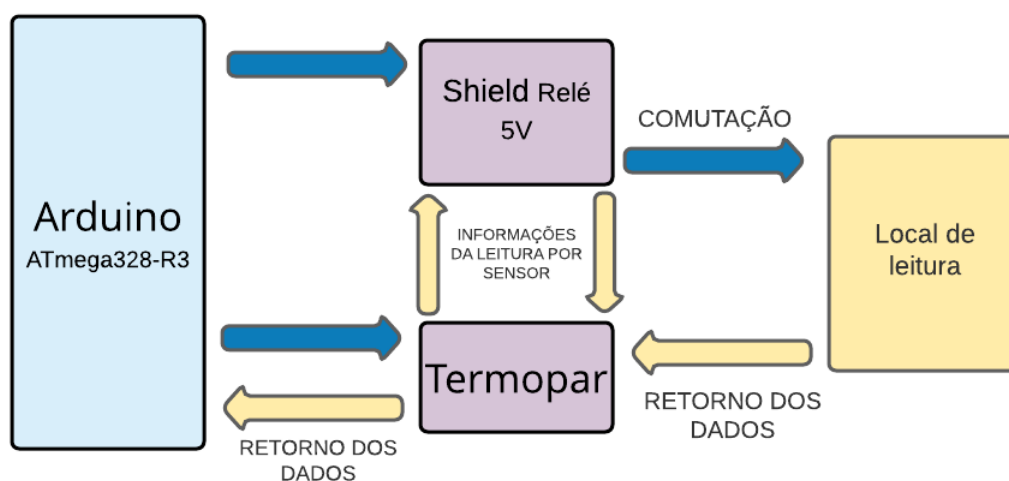
Sensores de termopar podem exibir erros de $\pm 1,5^{\circ}\text{C}$ até $\pm 9^{\circ}\text{C}$, dependendo do tipo e da faixa de medidas. A precisão interna do chip é de $\pm 0.15\%$. Para uma medição mais confiável e segura, deve-se manter o termopar sempre isolado de partes metálicas ou condutivas.

4 DESENVOLVIMENTO

4.1. Fluxo de Funcionamento

A Figura 6 apresenta o fluxo de funcionamento do projeto de leitura automatizada com o Arduino ATmega328-R3 que tem por sua aplicação o uso do microcontrolador ATmega328, sendo simples, de baixo custo e baixa potência.

Figura 6 - Fluxo de funcionamento.

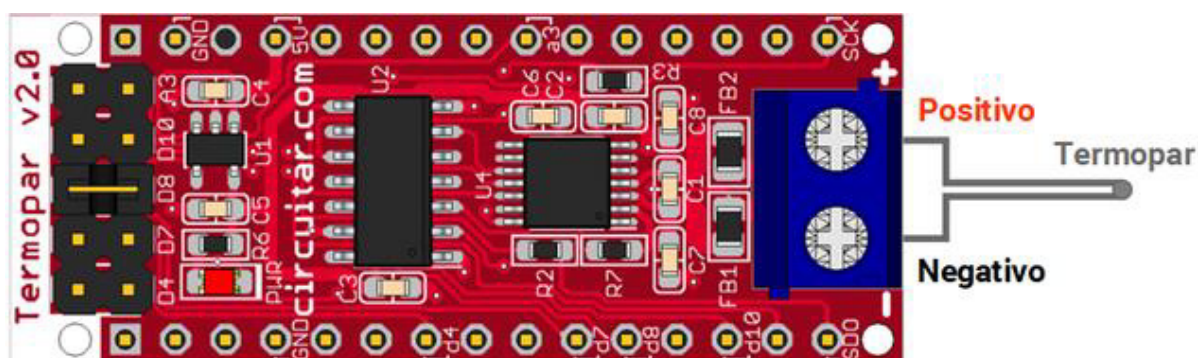


Fonte: Elaborada pelo Autor.

De acordo com a Figura 6, o Arduino tem a função de ligar o módulo Relé de 5V e o módulo Termopar para que ambos efetuem a troca de informações. A leitura do primeiro sensor é identificada como resultado do primeiro relé.

O Nanoshield Termopar possui um borne de parafuso onde foram fixadas as pontas do seu sensor de termopar. Feito isso, foram feitas as medições relacionadas a temperaturas com a ponta de medição. A Figura 7 ilustra como devem ser feitas as conexões.

Figura 7 - Conexão cabos Termopar.



Fonte: <https://www.circuitar.com.br/nanoshields/modulos/termopar/index.html>

O funcionamento para leitura, trabalha com o sinal de ligado para os canais 1 e 2 do módulo de Relé, posteriores ao envio do sinal para o Termopar. Temos um fio em comum com os dois Relés, que só funciona quando o canal 1 ou 2 é ativado.

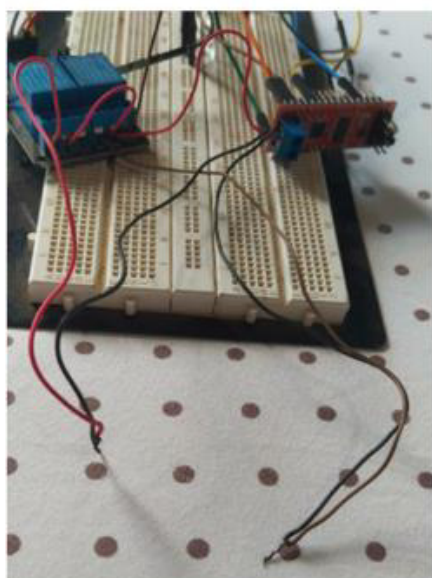
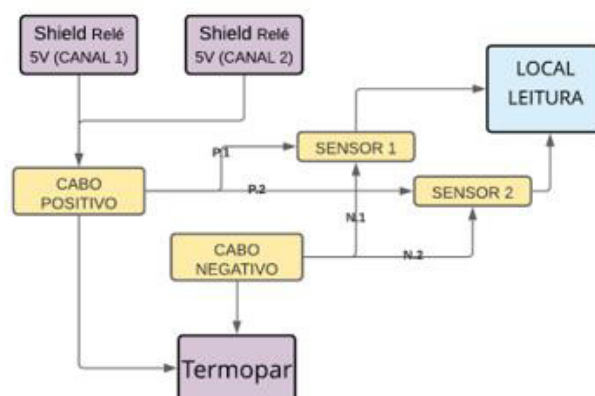
Então, junto ao Sensor Termopar no cabo positivo temos o ligamento no canal 1 e posterior o canal 2 do relé, possibilitando chavear a leitura, sendo assim, podendo ser acrescentado mais canais posteriormente para aumentar locais de leitura.

Na Figura 08 é possível analisar o funcionamento detalhado, relacionando o local onde será aplicada a leitura e os possíveis sensores disponibilizados por cada canal.

Sendo os cabos na cor preta que é comum para os dois canais em relação ao Termopar, tem a função de retornar os valores medidos em cada ponta de leitura. Os cabos na cor vermelha têm a função de comutar qual canal está sendo lido, sendo todos eles ligados ao sensor Termopar.

O Sensor Termopar poderá ser ligado a vários locais de leitura, sendo escolhidos conforme a comutação do canal definido pelo Relé. Neste protótipo foi usado um Módulo com 2 canais para leitura.

Figura 8 - Termopar e Relé detalhamento.




Fonte: Elaborada pelo Autor.

4.2. Software IDE Arduino

Para que o Arduino pudesse ser carregado e programado, foi utilizado o IDE Arduino. Este *software* é um editor de texto que conecta a parte física do Arduino (*hardware*) possibilitando a implementação dos módulos conectados, através da linguagem de programação, verifica se a programação pode ser executada e faz o *upload* para a placa correspondente.

Na Figura 9 é possível observar a interface do *software* acompanhada de uma parte da programação desenvolvida para testes com o projeto.

Figura 9 - Interface IDE



```

SerialThermometer_final | Arduino 1.8.13 (Windows Store 1.8.42.0)
Arquivo Editar Sketch Ferramentas Ajuda

SerialThermometer_final$
#include <Wire.h>
#include <SPI.h>
#include "Nanoshield_Termopar.h"

//Porta ligada ao pino IN1 do modulo
int porta_rele1 = 7;
//Porta ligada ao pino IN2 do modulo
int porta_rele2 = 6;

//Botão
int botao = 1;

// Termopar Nanoshield on CS pin D8, type K thermocouple, no averaging
Nanoshield_Termopar tc(8, TC_TYPE_K, TC_AVG_OFF);

void setup()
{
  //Define pinos para o rele como saida
  pinMode(porta_rele1, OUTPUT);
  pinMode(porta_rele2, OUTPUT);

  Serial.begin(9600);
  Serial.println("-----");
  Serial.println(" Nanoshield Serial Thermometer");
  Serial.println("-----");
  Serial.println();

  tc.begin();
}

void printErrors() {
  if (tc.isOpen()) {
    Serial.print("Open circuit");
  } else if (tc.isOverUnderVoltage()) {
    Serial.print("Overvoltage/Undervoltage");
  } else if (tc.isInternalOutOfRange()) {
    Serial.print("Internal temperature (cold junction) out of range");
  } else if (tc.isExternalOutOfRange()) {
    Serial.print("External temperature (hot junction) out of range");
  }
}

```

Fonte: Elaborado pelo Autor.

4.3. Testes com o Protótipo

Com o auxílio do sensor Termopar e a distribuição dos canais referentes ao sensor Relé de 2 canais se torna possível a distribuição de pontos de leitura distintos. Para realização desse teste foram utilizadas duas fontes de calor distintas: uma vela acesa e um copo d'água gelado. A Figura 10 apresenta os testes realizados.

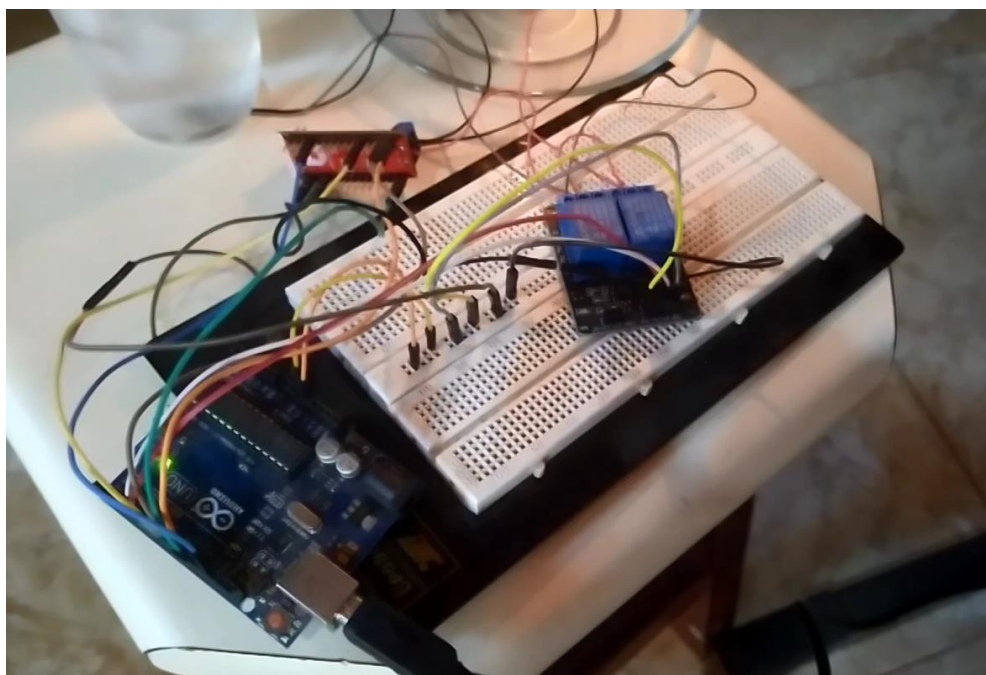
Figura 10 - Distribuição dos pontos de leitura.



Fonte: Elaborado pelo Autor.

Os pontos de leitura distintos, possuem uma diferença no valor medido de temperatura, gerados através do calor e do frio. Em seguida, foi acrescentado ao sistema o shield relé, conforme apresenta a Figura 11.

Figura 11 - Sistema com todos módulos conectados.



Fonte: Elaborado pelo Autor.

Após as ligações, foram realizados novos testes para medição da temperatura já com o shield relé conectado. A Figura 12 apresenta as leituras efetuadas no *software* IDE de nome Arduino descrito neste capítulo 4 com o protótipo já conectado no computador via USB.

Figura 12 – Resultado da medição de temperatura e acionamento relé.

```
-----  
Nanoshield Serial Thermometer  
-----  
Internal 1: 29.78 | External RELE 1: 20.93 | Internal 2: 29.72 | External RELE 2: 29.62  
-----  
Nanoshield Serial Thermometer  
-----  
Internal 1: 29.06 | External RELE 1: 15.56 | Internal 2: 28.98 | External RELE 2: 30.08  
-----  
Nanoshield Serial Thermometer  
-----  
Internal 1: 28.98 | External RELE 1: 15.41 | Internal 2: 28.92 | External RELE 2: 88.52
```

Fonte: Elaborado pelo Autor.

De acordo com a Figura 12 é possível observar que o resultado das leituras efetuadas como teste, a primeira leitura foi efetuada com o Relé de canal 1 colocado sobre um copo com água e gelo. A segunda leitura, temos após um breve momento a alteração no valor medido do sensor referente ao Relé de canal 1 pois o mesmo estava retornando os valores medidos dentro do copo com água. A terceira e última leitura, ambos canais e sensores sofreram alterações o primeiro Relé de canal 1 foi mantido dentro do copo com água e gelo e o Relé de canal 2, foi colocado sobre a chama de uma vela acesa.

Ambos os testes efetuados para validação das aferições de temperatura, tem por finalidade proporcionar leituras constantes em locais diferentes.

5 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo disponibilizar um sistema microcontrolado inteligente de baixo custo capaz de indicar as atitudes necessárias para conduzir de forma eficiente à produção e armazenamento dos diversos produtos armazenados. Para isso, foram apresentados componentes de baixo custo para construção de tal sistema.

A aplicação desse sistema é relevante em função da necessidade de integração entre as áreas relacionadas a tecnologia e possíveis campos de atuação. Além disso, este projeto torna possível o estudo interdisciplinar referente ao desenvolvimento de pesquisas conjuntas nas áreas de computação e afins em suas necessidades. A conclusão deste trabalho é uma contribuição direta para área de tecnologia, mantendo a sua preocupação para os problemas e desafios reais inicialmente de trabalhadores e produtores agrícolas. Tal envolvimento e colaboração se torna possível a partir da demanda do mercado brasileiro.

Como trabalhos futuros, sugere-se colocar em uma placa nova, removendo ela da placa de ensaio, colocando sobre uma placa ou em uma caixa onde ficaria protegida das ações climáticas, inserir uma bateria para permitir o uso fora da conexão USB e possibilitaria a implementação de novos módulos como armazenamento em cartões de memória das leituras já efetuadas e o momento em que foi efetuado quanto o módulo de conexão via internet.

6 REFERÊNCIAS BIBLIOGRÁFICAS

ARQUITETURA ARDUINO. Disponível em: <<https://deinfo.uepg.br/~alunoso/2019/SO/ARDUINO/ARQUITETURA/index.html>>. Acesso em 14 de set. 2020.

AZEVEDO, Loianne Faria et al. **A capacidade estática de armazenamento de grãos no Brasil**, Rio de Janeiro, outubro de 2008. Disponível em: <http://abepro.org.br/biblioteca/enegep2008_TN_STP_069_492_11589.pdf>. Acesso em 29 de nov. 2020.

BARONI, Gabriel Debarna. et al. **Cenários prospectivos da produção e armazenagem de grãos no Brasil**, Rio Grande do Sul, 2017. Disponível em: <<http://periodicos.ifsul.edu.br/index.php/thema/article/download/452/681/3323>> Acesso em 30 de nov. 2020.

CABÓ, Antônio Eudson Costa et al. **Sistema microcontrolado inteligente de baixo custo para o monitoramento de altas temperaturas em fornos da indústria de cerâmica vermelha**, Mossoró, 2015. Disponível em: <<https://ppgcc.ufersa.edu.br/wp-content/uploads/sites/42/2014/09/antonio-eudson-costa-cabo.pdf>> Acesso em 4 de junho de 2020.

CIRCUITAR, 2018. Disponível em: <<https://www.circuitar.com.br/nanoshields/modulos/termopar/index.html>>. Acesso em 18 de set. 2020.

CORDEIRO, Marcos Vinícius de Sousa. et al. **Automação residencial via Web e App utilizando módulos Wi-Fi ESP8266 em conjunto com sensores**, Curitiba, 2019. Disponível em: <[http://www.eletrica.ufpr.br/tcc/2019/2s/Marcos%20Vin%C3%ADcius%20de%20Sousa%20Cordeiro/Trabalho%20de%20conclus%C3%A3o%20de%20curso%20\(TCC%20B\)%20-%20MARCOS%20V.%20DE%20S.%20CORDEIRO.pdf](http://www.eletrica.ufpr.br/tcc/2019/2s/Marcos%20Vin%C3%ADcius%20de%20Sousa%20Cordeiro/Trabalho%20de%20conclus%C3%A3o%20de%20curso%20(TCC%20B)%20-%20MARCOS%20V.%20DE%20S.%20CORDEIRO.pdf)>. Acesso em 30 de nov. 2020.

FILIFELOP. Disponível em: <<https://www.filipeflop.com/blog/controle-modulo-rele-arduino/>>. Acesso em 15 de set. 2020.

FONSECA, E. G. P.; VEJA, A. S. **Tutorial sobre Introdução a Projetos Utilizando o Kit de Desenvolvimento Arduino**. Anais: XXXIX Congresso Brasileiro de Educação em Engenharia, Cobenge. Blumenau: FURB, 2011.7 p.

LEITE, Guilherme Leão. **Capacidade de armazenamento e escoamento de grãos do estado do Mato Grosso**, Brasília, março de 2013. Disponível em:<https://bdm.unb.br/bitstream/10483/4739/1/2013_GuilhermeLe%c3%a3oDiasLeite.pdf>. Acesso em 29 de nov. 2020.

MAIA, Guilherme Baptista da Silv. et al. **Panorama da armazenagem de produtos agrícolas no Brasil**. Revista do BNDES, n. 40, p. 161-194, dezembro de 2013. Disponível em: <<http://www.bndes.gov.br/SiteBNDES/export/sites/default/bndespt/Galerias/Arquivos/conhecimento/revista/rev4005.pdf>>. Acesso em 29 de nov. 2020.

JUNIOR, João Batista Padilha; MENDES, Judas Tadeu Grassi. **Agronegócio: uma abordagem econômica**. 1 ed. São Paulo: Pearson Prentice Hall, 2007.

SILVA, Angélica Leme da. et al. **Um estudo acerca da capacidade de armazenagem de grãos no município de Palmital-SP**, Palmital, 2012. Disponível em: <http://www.etcpalmital.com.br/_biblioteca/_tcc/_logistica/_2012/_arquivos/UMESTUDOACERCADACAPACIDADEDEARMAZENAGEMDEGRAOSNOMUNICIPIODEPALMITAL.pdf>. Acesso em 29 de nov. 2020.



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
PRÓ-REITORIA DE GRADUAÇÃO

Av. Universitária, 1069 • Setor Universitário
Caixa Postal 86 • CEP 74605-010
Goiânia • Goiás • Brasil
Fone: (62) 3946.1021 | Fax: (62) 3946.1397
www.pucgoias.edu.br | prograd@pucgoias.edu.br

Termo de autorização de publicação de produção acadêmica

O estudante **GABRIEL ANGELO GUILIANE MENDES DE ALMEIDA** do Curso de **Engenharia de Computação**, matrícula **2014.1.0033.0204-3**, telefone **(62) 9 8124-1485**, e-mail **almeidaguiliane@gmail.com**, na qualidade de titular dos direitos autorais, em consonância com a Lei nº 9.610/98 (Lei dos Direitos do Autor), autoriza a Pontifícia Universidade Católica de Goiás (PUC Goiás) a disponibilizar o Trabalho de Conclusão de Curso intitulado **Controle de Temperatura e Umidade Informatizado**, gratuitamente, sem ressarcimento dos direitos autorais, por 5 (cinco) anos, conforme permissões do documento, em meio eletrônico, na rede mundial de computadores, no formato especificado (Texto(PDF); Imagem (GIF ou JPEG); Som (WAVE, MPEG, AIFF, SND); Vídeo (MPEG, MWV, AVI, QT); outros), específicos da área para fins de leitura e/ou impressão pela internet, a título de divulgação da produção científica gerada nos cursos de graduação da PUC Goiás.

Goiânia, 03 de dezembro de 2020

Assinatura do autor:

Nome completo do autor:

Gabriel Angelo Guiliane Mendes de Almeida

Assinatura do professor – orientador:

Nome completo do professor – orientador:

Gustavo Siqueira Vinhal