



UNIVERSIDADE DO ESTADO DO AMAZONAS – UEA
ESCOLA SUPERIOR DE TECNOLOGIA – EST

AMANDA DA SILVA BARBOSA
BENILTON SEIXAS ANDRADE
EMERSON DA SILVA MUNIZ
INGRID MAYUMI FONSECA ONO

SENSORES E INSTRUMENTAÇÃO ELETRÔNICA

MANAUS – AM

2021

AMANDA DA SILVA BARBOSA
BENILTON SEIXAS ANDRADE
EMERSON DA SILVA MUNIZ
INGRID MAYUMI FONSECA ONO

TERMÔMETRO DIGITAL

Relatório técnico-científico
apresentado à Universidade do
Estado do Amazonas, ao curso de
Engenharia Eletrônica para obtenção
parcial da disciplina de sensores e
instrumentação eletrônica.

Professor: Weverson dos Santos
Cirino

MANAUS – AM

2021

SUMÁRIO

1. INTRODUÇÃO	4
2. OBJETIVO	5
3. DESENVOLVIMENTO DO PROJETO	6
4. ESPECIFICAÇÕES INICIAIS	7
4.1. Arduino UNO	7
4.2. Sensor de umidade e temperatura DHT11	8
4.3. Display LCD	9
4.4. Protoboard	10
4.5. Buzzer	10
5. ESPECIFICAÇÕES FINAIS	11
5.1. Suporte termômetro	11
5.2. Bateria Alcalina 9V	13
5.3. Transistor BC337	13
6. FUNCIONAMENTO DO PROTÓTIPO	13
6.1. Fator de correção	19
6.2. Visualização final	19
7. CONCLUSÕES	21
8. CRONOGRAMA (5W2H)	22
9. REFERÊNCIAS BIBLIOGRÁFICAS	23

1. INTRODUÇÃO

O projeto do termômetro digital é de extrema importância para a aplicação dos conhecimentos adquiridos na primeira parte do Curso de Sensores e Instrumentação Eletrônica. Os dados e referências servem de ideia para outros projetos a serem estudados. O trabalho trata de um termômetro digital utilizando uma placa Arduino Uno, sensor DTH-11, display LCD I2C para indicar a temperatura do ambiente, ou seja a temperatura que estará sendo medida em tempo real, um *Buzzer* para informar quando o termômetro atingir temperaturas fora do controle, cuja temperatura mínimo é 25,2°C e a temperatura máximo é 32,6°C.

2. OBJETIVO

O objetivo principal é implementar um protótipo de um termômetro digital. Sendo assim, colocar em prática o que foi estudado ao longo do curso ministrado, aplicando o uso do sensor DHT11, controlado pelo Arduino UNO, para monitorar a temperatura ambiente. No projeto, o valor da temperatura será exibido em um display LCD I2C, junto com os valores máximo e mínimo registrados desde que o microcontrolador foi ligado ou reiniciado.

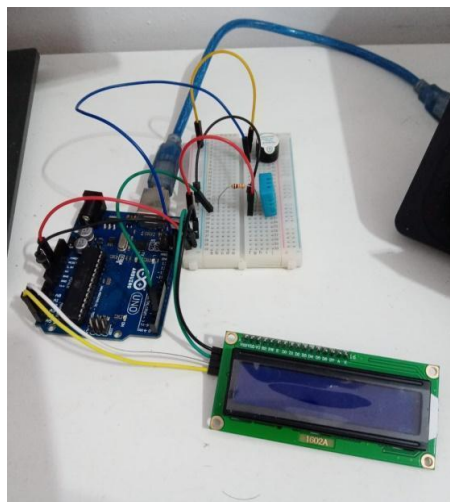
3. DESENVOLVIMENTO DO PROJETO

Sensores de temperatura são dispositivos que identificam a temperatura de um determinado equipamento, processo ou do próprio ambiente, podendo ser utilizados no monitoramento e controle para que se mantenham as condições adequadas de funcionamento ou conforto térmico.

Para o circuito do projeto fez-se uso do módulo DHT11. A ligação do mesmo utiliza apenas um pino para ligação à placa Arduino UNO, uma característica do sensor DHT11 é que ele não fornece informações “quebradas” de temperatura. Isso significa que o sensor vai mostrar as informações de, por exemplo, 18, 20, 25 graus, mas não as casas decimais de 18.2 ou 25.6 graus.

Em primeiro instante será feita a programação no Arduino com a ideia de representar os sinais de saída do sensor de temperatura no display. Assim, o sensor de temperatura DHT11 transforma a temperatura indicada em sinais elétricos que será exibido no Display LCD.

Figura 1. Montagem Inicial do circuito



Fonte: Elaborada pelo Autor

Materiais utilizados:

- Arduino UNO;
- Sensor de temperatura DTH11;
- Display LCD (LiquidCrystalI2C);
- Protoboard;
- Buzzer;
- Resistor;
- Transistor;

4. ESPECIFICAÇÕES INICIAIS

4.1. Arduino UNO

O Arduino Uno é uma placa baseada no microcontrolador Atmega328 e possui estrutura de entradas e saídas, sendo possível a conexão com o sensor. O Arduino possui prototipagem eletrônica open-source que se baseia em hardware e software flexíveis e simples de usar. O microcontrolador na placa é programado com a linguagem de programação Arduino, com base na linguagem Wiring, que possui código aberto e é uma abstração do C/C++ para se programar em microcontroladores. Já o ambiente de desenvolvimento Arduino, baseado no ambiente Processing. A placa pode ser alimentada pela conexão USB ou por uma fonte de alimentação externa. [1]

O software utilizado para executar a programação é o IDE Arduino, que é editor de código, compilador e depurador. Os softwares para Arduino são chamados de Sketches e possuem as características de cortar/colar e buscar/substituir o texto. [1]

Para programar tem-se os componentes de estruturas, variáveis, operadores booleanos, de comparação e aritméticos, estrutura de controle e funções digitais e analógicas.

Figura 2. Placa Arduino UNO



Fonte: <https://produto.mercadolivre.com.br/>

Tabela 1. Especificação Placa Arduino UNO

ESPECIFICAÇÃO	
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 Flash	32 KB (ATmega328) dos quais 0.5 KB são usados pelo bootloader
Memória SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Velocidade do Clock	16 MHz
Dimensões	68,58mm x 53,34mm
Peso	150g

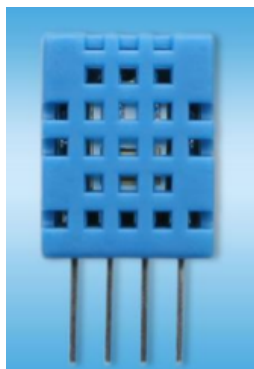
Fonte: <https://www.baudaeletronica.com.br/>

4.2. Sensor de umidade e temperatura DHT11

O Sensor de Umidade e Temperatura DHT11 é um sensor de temperatura e umidade que permite fazer leituras de temperaturas entre 0 a 50 Celsius e umidade entre 20 a 90%, muito usado para projetos com Arduino. [2]

O elemento sensor de temperatura é um termistor do tipo NTC e o sensor de umidade é do tipo HR202, o circuito interno faz a leitura dos sensores e se comunica a um microcontrolador através de um sinal serial de uma via. Confira a pinagem nas imagens. [2]

Figura 3. Sensor de temperatura e umidade DHT11



Fonte: Datasheet DHT11

Tabela 2. Especificação sensor DHT11

ESPECIFICAÇÃO	
Dimensões	23mm x 12mm x 5mm (incluindo terminais)
Alimentação	3,0 a 5,0 VDC (5,5 Vdc máximo)
Corrente	200uA a 500mA, em stand by de 100uA a 150 Ua
Faixa de medição de umidade	20 a 90% UR
Faixa de medição de temperatura	0° a 50°C
Precisão de umidade de medição	± 5,0% UR
Precisão de medição de temperatura	± 2,0 °C
Tempo de resposta	< 5s

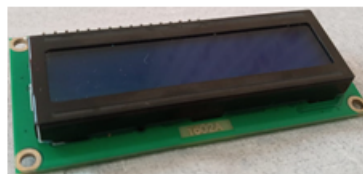
Fonte: vidadesilicio.com.br/

4.3. Display LCD

O Módulo Adaptador I2C para Display LCD (16X2 / 20X4) foi desenvolvido com a finalidade de simplificar a conexão de display LCD (16X2 / 20X4) ao microcontrolador. Para uma conexão de 4 bits entre o display LCD (16X2 / 20X4) e o microcontrolador é necessário ao menos 6 cabos, logo, se o microcontrolador tiver poucas portas digitais isso poderá ser um problema. Com o Módulo Adaptador I2C para Display LCD (16X2 / 20X4) é necessário apenas 2 cabos de comunicação entre o display LCD (16X2 / 20X4) e o microcontrolador. [3]

Em um projeto mais extenso e que é necessário à utilização de muitas portas digitais por parte de outros dispositivos, o Módulo Adaptador I2C para Display LCD (16X2 / 20X4) pode ser a solução simples e prática para que você economize algumas portas digitais na ligação do seu display LCD (16X2 / 20X4). [3]

Figura 4. Módulo adaptador I2C para display LCD



Fonte:



Próprio autor

Tabela 3. Especificação Display LCD com módulo I2C.

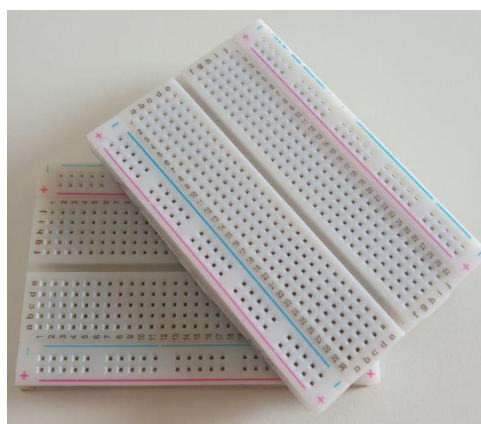
ESPECIFICAÇÃO	
Controlador	PCF8574T
Tensão de operação	5VDC
Interface	I2C
Compatibilidade	display LCD 16×2 e 20×4
Trimpot para ajustar o contraste do display LCD	
Pinos para ligar / desligar o backlight do display LCD	

Fonte: <https://blogmasterwalkershop.com.br/>

4.4. Protoboard

É a ferramenta eletrônica que vai ser utilizada na ligação do sensor com o Arduino. É um excelente componente para a montagem de circuitos eletrônicos, por ser de fácil utilização, rápida e prática para a montagem do projeto. Depois que passamos a entender como é o percurso da corrente elétrica fica muito fácil usar esse componente. Essa ferramenta foi usada somente para montagem inicial do protótipo.

Figura 5. Protoboard



Fonte: Próprio autor

4.5. Buzzer

O Buzzer Ativo 5V é um componente indicado para efeitos sonoros em projetos eletrônicos como alarmes, sistemas de sinalização, jogos, brinquedos, etc. [4]

O buzzer do tipo ativo contém um circuito oscilador embutido, assim basta ser energizado para que o mesmo comece a emitir um beep contínuo. Ao conectar o buzzer ativo no circuito, verificar a polaridade (pinos + e -). [4]

Figura 6. Buzzer



Fonte: <https://www.baudaeletronica.com.br/>

Tabela 4. Especificação Buzzer

ESPECIFICAÇÃO	
Tensão de trabalho	4 a 8V
Tensão recomendada	5V
Corrente máxima	40mA
Diâmetro	12mm
Altura	10mm
Peso	1,6g
Peso com embalagem	1,8g

Fonte: <https://www.usinainfo.com.br/>

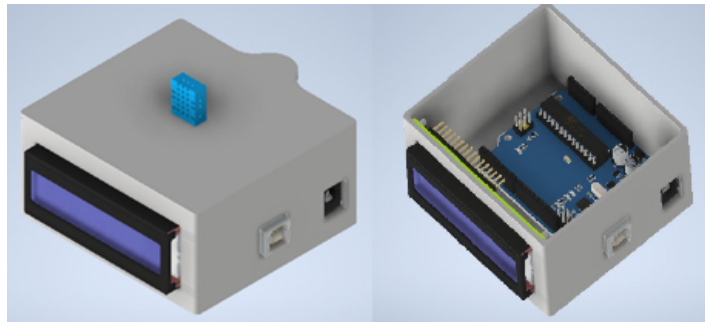
5. ESPECIFICAÇÕES FINAIS

Para melhor compreensão do projeto o protótipo inicial foi montado no protoboard, com todas as conexões necessárias, logo após a compilação e carregamento do código na placa fez uso dos componentes representados abaixo.

5.1. Suporte termômetro

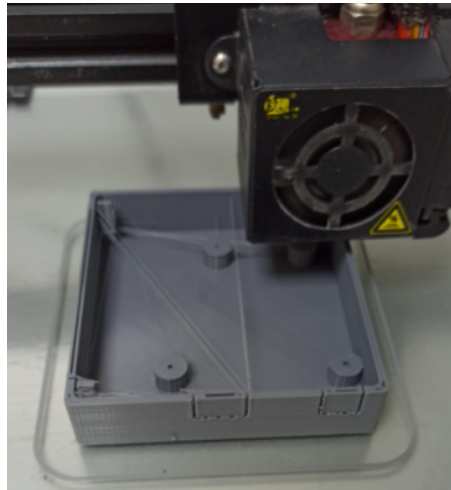
Para suporte do termômetro foi confeccionado em impressora 3D a estrutura representada na Figura 7, o design foi feito no software *Inventor da Autodesk*.

Figura 7. Suporte para o termômetro digital



Fonte: Print Software Inventor Autodesk

Figura 8. Impressão do suporte em Impressora 3D



Fonte: Próprio Autor

Após a confecção da base para suporte do protótipo como um todo, obteve-se como resultado a estrutura apresentada logo abaixo:

Figura 9. Representação final



Fonte: Próprio Autor

5.2. Bateria Alcalina 9V

Para alimentação da placa Arduino UNO fez uso de uma bateria de 9V conforme a Figura 10.

Figura 10. Bateria



Fonte: <https://produto.mercadolivre.com.br/>

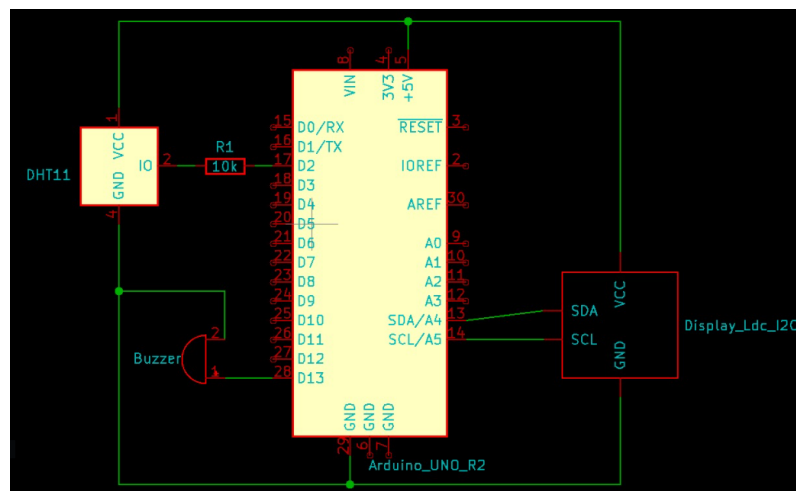
5.3. Transistor BC337

Para controle de acionamento desnecessário do buzzer foi utilizado um transistor NPN BC337.

6. FUNCIONAMENTO DO PROTÓTIPO

O esquema para auxiliar na montagem no protótipo foi feito no *Software Kicad* e está representado na Figura 11 .

Figura 11. Esquemático do Protótipo



Fonte: Próprio autor

Devido à utilização display com módulo I2C, a montagem tornou-se algo mais simples considerando-se que esse protocolo de comunicação serial é um dos mais práticos, simples e de baixo custo disponíveis no mercado, para projetos

microcontrolados. O protocolo I2C se torna uma mão na roda, visto que possibilita uma modularidade de dispositivos.

O versionamento dos arquivos do protótipo foi feito no GitHub devido a facilidade do controle de versões dos arquivos e compartilhamento entre os participantes da equipe e se encontra no seguinte endereço: https://github.com/benilton02/termometro_digital.

Fez-se uso da biblioteca *ArduinoThreads* no código do termômetro digital. Esta biblioteca implementa e ajuda a:

- Agendar, gerenciar e simplificar tarefas paralelas e periódicas, definir tempo fixo ou variável entre as corridas;
- Organizar o código em qualquer tipo de projeto, coloque todas as leituras do sensor em uma discussão, mantenha o circuito principal limpo;
- Esconder a complexidade do gerenciamento de threads;
- Executar tarefas de "pseudo-background" usando interrupções de temporizador;

Piscar um *LED* geralmente é a primeira coisa que um usuário do Arduino aprende. E isso demonstra que realizar periodicamente uma única tarefa, como alternar o estado de um LED, é realmente fácil. No entanto, pode-se descobrir rapidamente que gerenciar várias tarefas periódicas não é tão simples se as tarefas têm horários diferentes. [5]

Deve-se notar que estes não são “threads” no significado real do termo em ciência da computação: as tarefas são implementadas como funções que são executadas periodicamente. Por um lado, isso significa que a única maneira de uma tarefa render a CPU é retornado ao chamador e, portanto, não é aconselhável *delay* fazer longas esperas dentro de qualquer tarefa. Por outro lado, isso torna o *ArduinoThreads* amigável com a memória, já que nenhuma pilha precisa ser alocada por tarefa. [5]

No programa em questão foram incluídas as bibliotecas necessárias para execução do projeto: *LiquidCrystal_I2C*, *ThreadController*, *Thread* e *dth11*. O usuário define um objeto Thread para cada uma dessas tarefas e, em seguida, permite que a biblioteca gerencie sua execução programada. Logo após foram definidas constantes para o código e a especificação do devido pino a ser utilizado, conforme o especificado no código o sensor de temperatura DTH11 lerá a cada 700ms.

Para a programação deste protótipo, fez-se uso do sketch abaixo:

```
#include <LiquidCrystal_I2C.h>
#include <ThreadController.h>
#include <Thread.h>
#include <dht.h>

#define threadSleep 700
#define DHT11_PIN 2
#define address 0x27
#define row 2
#define column 16
#define buzzer_pin 13

LiquidCrystal_I2C lcd(address, column, row);
ThreadController threadController = ThreadController();
Thread showLcdThread = Thread();
Thread buzzerThread = Thread();
dht DHT;

void createThread();
void startBuzzer();
void setInitialValues();
void setLcd();
void showCelsius();
float minCelsius();
float maxCelsius();

float minValueCelsius;
float maxValueCelsius;
float celsius;

void setup(){
    Serial.begin(9600);

    pinMode(buzzer_pin, OUTPUT);
    digitalWrite(buzzer_pin, LOW);
    setInitialValues();
    setLcd();
    createThread();
}
```

```

void loop() {
    threadController.run();
    Serial.println(celsius, 2);
}

void createThread(){
    showLcdThread.setInterval(threadSleep);
    buzzerThread.setInterval(threadSleep);
    showLcdThread.onRun(showCelsius);
    buzzerThread.onRun(startBuzzer);

    threadController.add(&showLcdThread);
    threadController.add(&buzzerThread);
}

void startBuzzer(){
    if (celsius < 25.2 || celsius > 32.6){
        digitalWrite(buzzer_pin, HIGH);
        delay(400);
        digitalWrite(buzzer_pin, LOW);
        delay(200);
        Serial.print(celsius, 2); // debug
        Serial.println(" °C"); // debug
    }
}

void setInitialValues(){
    DHT.read11(DHT11_PIN);
    minValueCelsius = DHT.temperature;
    maxValueCelsius = DHT.temperature;
}

void setLcd(){
    lcd.init();
    lcd.backlight();
    lcd.clear();
}

```



```

void showCelsius(){
    DHT.read11(DHT11_PIN);
    celsius = DHT.temperature;
    lcd.setCursor(0, 0);
    lcd.print("Temp: ");
    lcd.print(celsius);
    lcd.write(223);
    lcd.setCursor(0, 0);
    lcd.print("");

    lcd.setCursor(8, 1);
    lcd.print("");
    lcd.print(maxCelsius());
    lcd.write(223);

    lcd.setCursor(0, 1);
    lcd.print("");
    lcd.print(minCelsius());
    lcd.write(223);
}

float minCelsius(){
    if (celsius < minValueCelsius){
        minValueCelsius = celsius;

        Serial.print(minValueCelsius, 2); // debug
        Serial.println(" °C minValueCelsius"); // debug
    }
    return minValueCelsius;
}

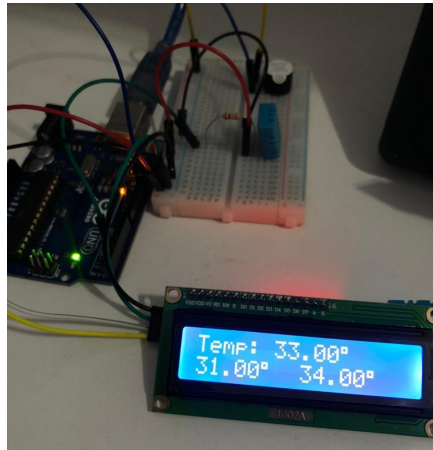
float maxCelsius(){
    if (celsius > maxValueCelsius){
        maxValueCelsius = celsius;

        Serial.print(maxValueCelsius, 2); // debug
        Serial.println(" °C maxValueCelsius"); // debug
    }
    return maxValueCelsius;
}

```

As informações adquiridas pelo sensor DHT11 são enviadas para o Arduino e exibidas no display LCD I2C na unidade Celsius (°C) conforme a Figura 13. Uma montagem inicial foi feita para aplicação do código acima citado e apresentação da temperatura no display.

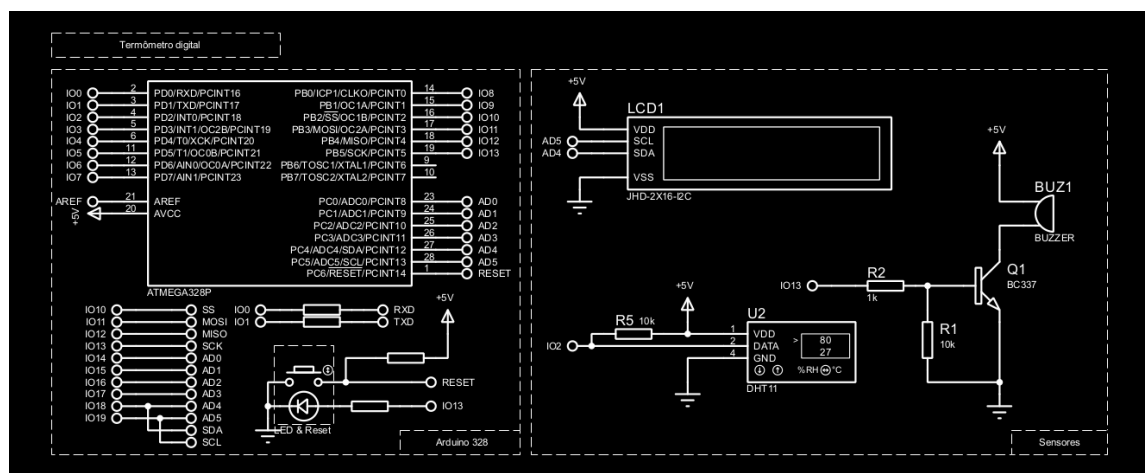
Figura 12. Circuito montado no protoboard



Fonte: Próprio autor

Após alguns testes foi possível perceber alguns acionamentos desnecessários do buzzer, e para evitar possíveis problemas na aquisição dos dados advindos do sensor e acionamentos indesejáveis foi acoplado ao circuito um transistor para que o dispositivo não acione com qualquer interferência. Logo o circuito inicial foi alterado, e uma nova versão foi elaborada no *Software Proteus*, conforme representado na Figura 13.

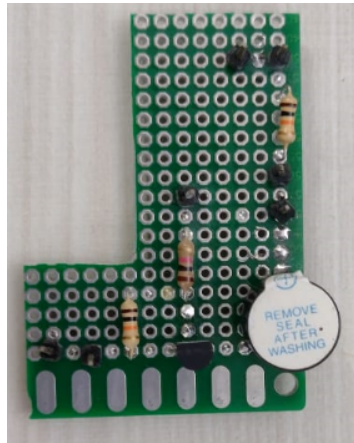
Figura 13. Esquemático



Fonte: Print Software Proteus

Por questão de estética do circuito, a montagem foi feita em uma placa universal conforme representado na Figura 14.

Figura 14. Circuito montado na placa universal



Fonte: Próprio autor

6.1. Fator de correção

O fator de correção foi determinado fazendo-se aquisição da temperatura medida por um sensor comercial e subtraindo-se da temperatura medida pelo sensor utilizado no projeto DHT11, obtivemos como resultado o valor 0,2, podemos ver na figura 15.

Figura 15. Circuito montado na placa universal



Fonte: Próprio autor

6.2. Visualização final

Após todas as atualizações e modificações necessárias, para a comprovação e efetivação do funcionamento do protótipo do termômetro digital, foram feitos testes e comparações com um termômetro comercial, e podemos efetivar o real funcionamento do protótipo, na Figuras 16 a demonstração.

Figura 16. Circuito montado na placa universal



Fonte: Próprio autor

Para Visualização do vídeo de explicação do projeto basta acessar o link:
https://drive.google.com/drive/folders/1vqllG1QsWd9qqYDe4cfl24rv3g0pfQa4?usp=s_haring

7. CONCLUSÕES

Colocar em prática o que foi visto ao longo da unidade apresentada foi primordial para aplicação de todo o conhecimento adquirido nas aulas. Com a obtenção positiva dos resultados, o processo do aprendizado chegou ao nível esperado, ocorrendo uma expansão dos conhecimentos, conciliando cada área da tecnologia, despertando também a curiosidade em fazer outros projetos em nossas residências envolvendo outras ideias e experimentos.

8. CRONOGRAMA (5W2H)

Link para o cronograma:

<https://docs.google.com/spreadsheets/d/1wbIWk6-iwrk5nlDe8a9g54l2NCYMPKfOJ0tCxBJIQ/edit?usp=sharing>

O que?	Porque?	Onde?	Quem?	Quando?	Como?	Quanto?
Pesquisa de trabalhos similares	Para tomar como ponto de partida	Internet	Amanda	9/março	Foram feitas pesquisas no datashreets e em sites diversos sobre projetos anteriores que utilizaram o sensor de temperatura e umidade DHT11 e dispositivos similares para compreensão inicial do projeto.	Gratuito
Estudo bibliográfico dos componentes	Para aplicação no projeto	Internet	Amanda	10/março	Estudo dos datashreets de cada componente e dispositivo utilizado na elaboração do circuito e consequentemente aplicação dos mesmos na simulação.	Gratuito
Elaboração do esquema elétrico	Construção da estrutura elétrica	Proteus	Ingrid	15/março	Verificando datashreets do arduino Uno, Display LCD, DHT11 e buzzer para elaboração do esquema.	Gratuito
Revisão do Esquema elétrico	Verificar possíveis falhas	Proteus	Emerson	16/março	Verificação dos componentes a serem utilizados e sua posição no esquemático	Gratuito
Pesquisar componentes do projeto	Calcular custos, verificar a adequação ao projeto	Internet	Emerson	11/março	Houve uma reunião com a equipe para dimensionarmos os componentes a serem usados	Gratuito
Aquisição dos componentes	Construção e montagem do projeto	Mercado Livre	Emerson	12/março a 16/março	Após realizada a pesquisa, houve a compra no site	R\$ 73,90
Montagem do circuito auxiliar	Construção e montagem do projeto	LSE	Ingrid	17/março	Montando componentes na placa universal, criando trilhas com solda.	Gratuito
Revisão da montagem do circuito auxiliar	Montagem conforme o esquema	LSE	Emerson	18/março	Olhando o esquemático e testando conexões	Gratuito
Elaboração do projeto do case	Case para proteção das placas e fios de conexões	Inventor	Ingrid	19/março	Projeto feito em CAD com os dimensionamentos de cada placa e display a serem agrupados.	Gratuito
Confeção do case	Case para proteção das placas e fios de conexões	Impressora 3D	Ingrid	19 a 20/março	Feito o fatiamento do case no programa Cura, foi colocado para imprimir no material em PLA cinza.	Gratuito
Montagem das placas no case	Case para proteção das placas e fios de conexões	LSE	Ingrid	20/março	Parafusagem e encaixe das placas montadas no case	Gratuito
Versionamento de arquivos do projeto	Preservar histórico de alterações no código e arquivos de modelagem do projeto	https://github.com/benilton02/termomometro_digital.git	Benilton	11/março até 19/março	Git e Github	Gratuito
Embarcar código	Gravar código no Arduino	Arduino	Benilton	16/março	Arduino IDE	Gratuito
Testes de funcionamento	Garantir funcionamento do projeto de acordo com os requisitos	Medir a temperatura em diferentes ambientes domésticos: garagem, quarto, banheiro etc.	Benilton	17/março	Definir diferentes cenários de testes para verificar o funcionamento do protótipo real	Gratuito
Relatório Técnico	Para exposição de todas as fases efetuadas no projeto.	Word	Amanda	16/março	Foram documentadas todas as fases do projeto, desde a pesquisa bibliográfica inicial até o teste de funcionamento final.	Gratuito
Apresentação	Apresentação dos processador seguidos até a finalização do projeto.	PowerPoint	Amanda	25/março	Foram documentadas todas as fases do projeto, desde a pesquisa bibliográfica inicial até o teste de funcionamento final, para apresentação aos colegas e professores.	Gratuito

9. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Souza, Fábio. **Introdução ao Arduino - Primeiros Passos na plataforma**. Embarcados, 2013. Disponível: <<https://www.embarcados.com.br/arduino-primeiros-passos/>>. Acesso em 09 de mai. de 2021
- [2] **Sensor de Umidade e Temperatura DHT11**. FILIPEFLOP. Disponível em: <<https://www.filipeflop.com/produto/sensor-de-umidade-e-temperatura-dht11/>>. Acesso em 09 de mai. de 2021
- [3] Oliveira, Euler. **Como usar com Arduino - Módulo Adaptador I2C para Display LCD (16x2 / 20x4)**. Master Walker. Disponível em: <<https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-modulo-adaptador-i2c-para-display-lcd-16x2-20x4>>. Acesso em 09 de mai. de 2021
- [4] **Buzzer Ativo 5V**. FILIPEFLOP. Disponível em: <<https://www.filipeflop.com/produto/buzzer-ativo-5v/>>. Acesso em 09 de mai. de 2021
- [5] Seidel, Ivan. **ArduinoThreads**. Disponível em: <<https://github.com/ivanseidel/ArduinoThread>>. Acesso em 13 de mai. de 2021
- [6] **All Data Sheet**. DHT11. Disponível em: <<https://www.alldatasheet.com/view.jsp?Searchword=Dht11%20datasheet>>. Acesso em 13 de mai. de 2021
- [7] **All Data Sheet**. BC377. Disponível em: <https://www.alldatasheet.com/view.jsp?Searchword=Bc337%20datasheet&gclid=Cj0KCQjw78yFBhCZARIsAOxgSx2qwFVrXxBsz7A0k7R0-ygYig9fylBLDEJNPzrqL6geDiES7Eh6PVEaAvrYEALw_wcB>. Acesso em 13 de mai. de 2021