



DISPLAY OLED INOVADOR – UMA PROPOSTA DE VISUALIZAÇÃO QUE TRANSFORMA EXPERIÊNCIAS COM UMA TELA TRANSPARENTE

João Pedro Araujo Porto.

RA: 1040770

Universidade Presbiteriana Mackenzie (UPM).
Rua da Consolação, 930 Consolação, São Paulo – SP, 01302-907 – Brazil.

Abstract: This article discusses the creation of a prototype using Internet of Things (IoT) technology through Arduino, which allows for multiple functionalities via visualization. Everyday items, such as looking at a clock or learning geometry at school, can become unique experiences through innovation.

Resumo: Este trabalho explora o desenvolvimento de um protótipo utilizando a Internet das Coisas (IoT) com base em Arduino, possibilitando uma série de funcionalidades visuais. Através desse projeto, itens comuns do cotidiano, como consultar um relógio ou aprender conceitos de geometria, podem ser transformados em experiências inovadoras por meio da tecnologia.

1. Introdução

A Internet das Coisas (IoT) tem revolucionado a forma como a computação e a automação estão integradas às nossas rotinas diárias (EVANS 2011). Entretanto, muitas pessoas ainda encontram dificuldade em visualizar claramente a aplicação prática da IoT. Observando ambientes cotidianos, identificamos oportunidades simples para demonstrar essa tecnologia. Diante disso, surgiu a ideia de reimaginar a visualização de vários tipos de outros elementos em algo mais interativo e inovador. Utilizando Arduino e um Display OLED, tornou-se possível exibir informações como informações ou formas geométricas diretamente na superfície de uma mesa, transformando a experiência de observação em algo mais atrativo e moderno.

2. Materiais e Métodos

Para a concretização deste projeto, foi realizado um levantamento dos materiais necessários. O protótipo do Display OLED Inteligente foi construído com base em componentes eletrônicos conectados à placa Arduino Nano 3.0.



2.1 Modelo de Montagem do Protótipo

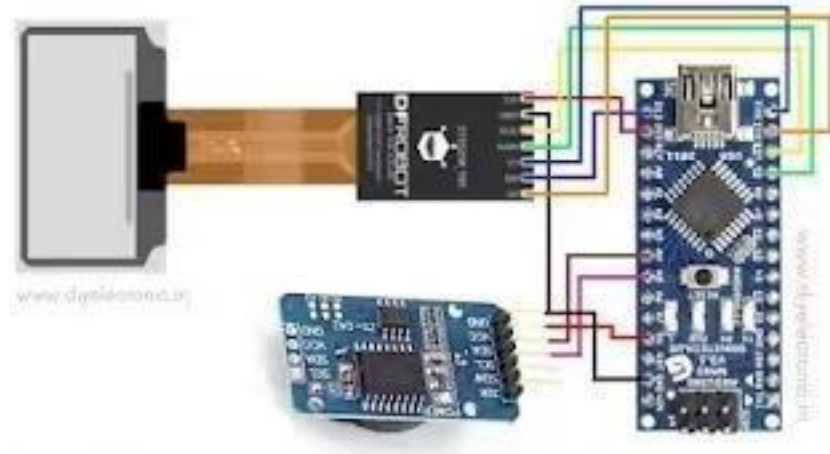


Imagem 1 – Modelo do Protótipo

2.2 Funcionalidade do Arduino Nano 3.0

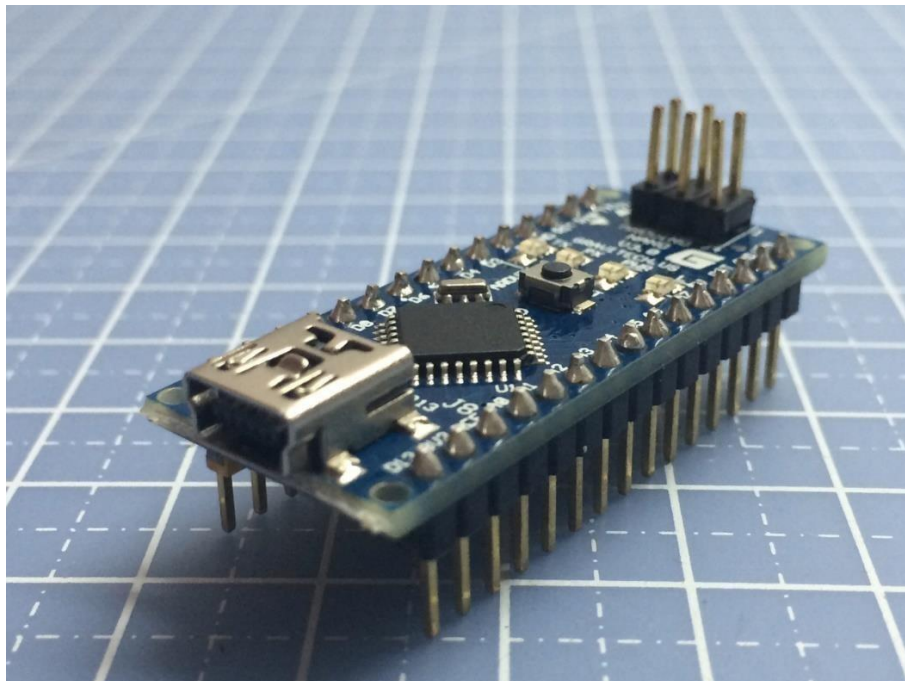


Imagem 2 – Placa Arduino Nano V 3.0 Atmega 328

O Arduino Nano V3.0 é uma placa compacta, com funcionalidades semelhantes ao Arduino Uno R3, mas em um formato reduzido. Uma de suas principais vantagens é a compatibilidade direta com a protoboard, o que facilita muito a montagem dos projetos, eliminando a necessidade de muitos jumpers. As principais características do Arduino Nano incluem:



CPU: Microcontrolador Atmel Atmega 328

Tensão mínima de Alimentação: 3.3V

Tensão Máxima de Alimentação: 20V

Tensão Ideal para Alimentação: 7V – 12V

Pinos Digitais I/O: 14 pinos, no qual 6 destes também servem para saídas PWM

Pinos de Entrada Analógica: 8 entradas com 10 bits de resolução

Corrente DC Por Pino I/O: 40mA

Memória Flash: 16 Kb, sendo que 2 Kb são utilizados pelo bootloader

SRAM: 2 Kb

EEPROM: 1Kb

Velocidade de Clock: 16 MHz

O Arduino Nano possui 14 pinos digitais que podem ser configurados como entradas ou saídas de dados. Esses pinos operam com uma tensão de 5V e são capazes de fornecer ou receber até 40mA de corrente, no máximo, por pino.

2.3 Arduino Nano USB



Imagem 3 – Arduino Nano USB

O cabo Mini USB é amplamente utilizado para comunicação entre o Arduino Nano e o computador, essencial para a gravação do código na placa. Com 20cm de comprimento, ele é ideal para diversas aplicações.



2.4. Módulo de Comunicação

O Arduino UNO é conectado a um computador ou dispositivo intermediário via comunicação serial-USB. Um software ou script rodando no dispositivo intermediário atua como ponte entre o Arduino e o broker MQTT.

Um software roda no dispositivo intermediário para:

Ler os dados provenientes do Arduino pela porta serial.

Publicar esses dados no broker MQTT.

Assinar tópicos no broker MQTT para receber comandos.

Traduzir os comandos recebidos para instruções compreensíveis pelo Arduino via Firmata.

O sensor conectado ao Arduino coleta dados. O Arduino envia os dados via porta serial para o dispositivo intermediário, o script no dispositivo intermediário interpreta esses dados e os publica em um tópico específico no broker MQTT.

Fluxo de Comandos do Broker para o Atuador. Um cliente MQTT publica um comando no tópico, o dispositivo intermediário recebe o comando do broker MQTT. O script traduz o comando para uma instrução Firmata e o envia ao Arduino via serial.

O Arduino executa o comando, acionando ou desativando o atuador.

2.5 Funcionalidade do Display OLED



Imagem 4 – Display OLED

O Display OLED 128x64 de 0.96" é altamente eficiente em termos de energia, pois não necessita de backlight. Sua comunicação via I2C permite fácil integração com placas como Arduino e Raspberry Pi. Especificações:

- Cor do texto: Branco;
- Tipo de comunicação: I2C Serial;
- Tensão de operação: 2,2V a 5,5V;
- Consumo: 0,04W;
- Resolução da imagem: 128 x 64 pixels;
- Pinagem: VCC (2,2V a 5,5V), GND (Ground), SCL (Clock) e SDA (Dados);
- Dimensões: 30 x 28 x 4,5mm.



2.6 Jumper m-m



Imagem 6 – Jumper m-m



O conjunto de 20 jumpers macho-fêmea facilita as montagens de circuitos na protoboard, proporcionando maior agilidade durante o processo de montagem.

- Comprimento do jumper (unidade): 20cm;
- Largura do conector: 2,54mm;
- Secção do fio condutor: 24 AWG;
- Dimensões: 20cm de comprimento (cada jumper).

2.7 Protoboard 400

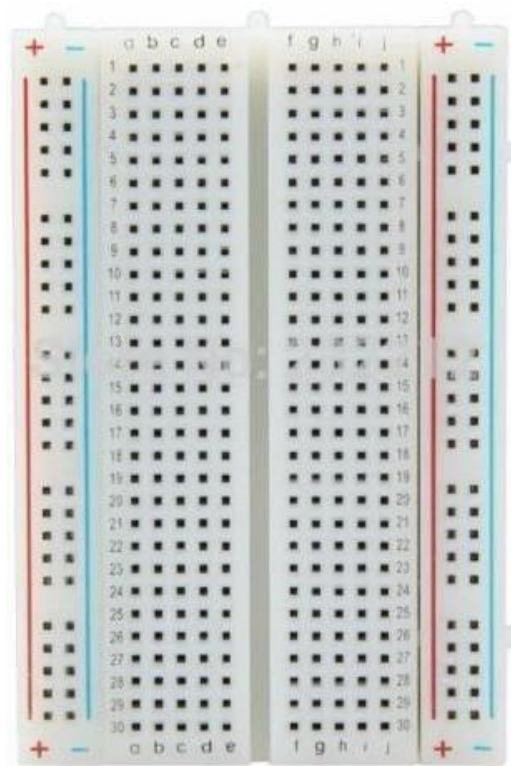


Imagem 7 – Protoboard 400

A protoboard de 400 pontos é fundamental para a montagem de circuitos de teste. Com fita adesiva dupla face, pode ser fixada em diversas superfícies.

- Material: ABS (plástico);
- Encaixável com outras protoboards;
- Faixa de Temperatura: -20 a 80°C;
- Tensão Máxima: 500v AC por minuto;
- Dimensões: 83 mm x 55 mm x 10 mm.



2.8 Arduino IDE 2.2.1

O Arduino IDE é a plataforma de desenvolvimento utilizada para programar o Arduino. Este ambiente de desenvolvimento integrado (IDE) facilita a criação e a gravação de códigos no microcontrolador, sendo essencial para controlar sensores, motores e outros dispositivos conectados ao Arduino.

Metodologia

A primeira etapa da montagem do protótipo envolveu a conexão do Arduino Nano, do módulo RTC DS3231 e do Display OLED na protoboard. Em seguida, esses componentes foram interligados por jumpers, como mostrado na seguinte representação:

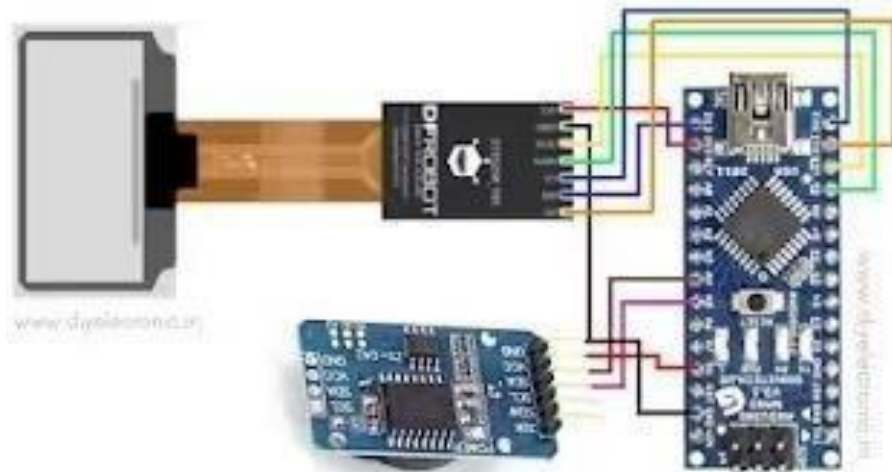


Imagem 8 – Representação da ligação entre o Arduino Nano, o RTC ds3231 e o Display OLED.

Proxima etapa envolveu a transferência do código desenvolvido no Arduino IDE para o Arduino Nano, utilizando a conexão USB. Após carregar e compilar o código no microcontrolador, basta executar o programa para que o Display OLED Inteligente inicie seu funcionamento corretamente.



3. Código

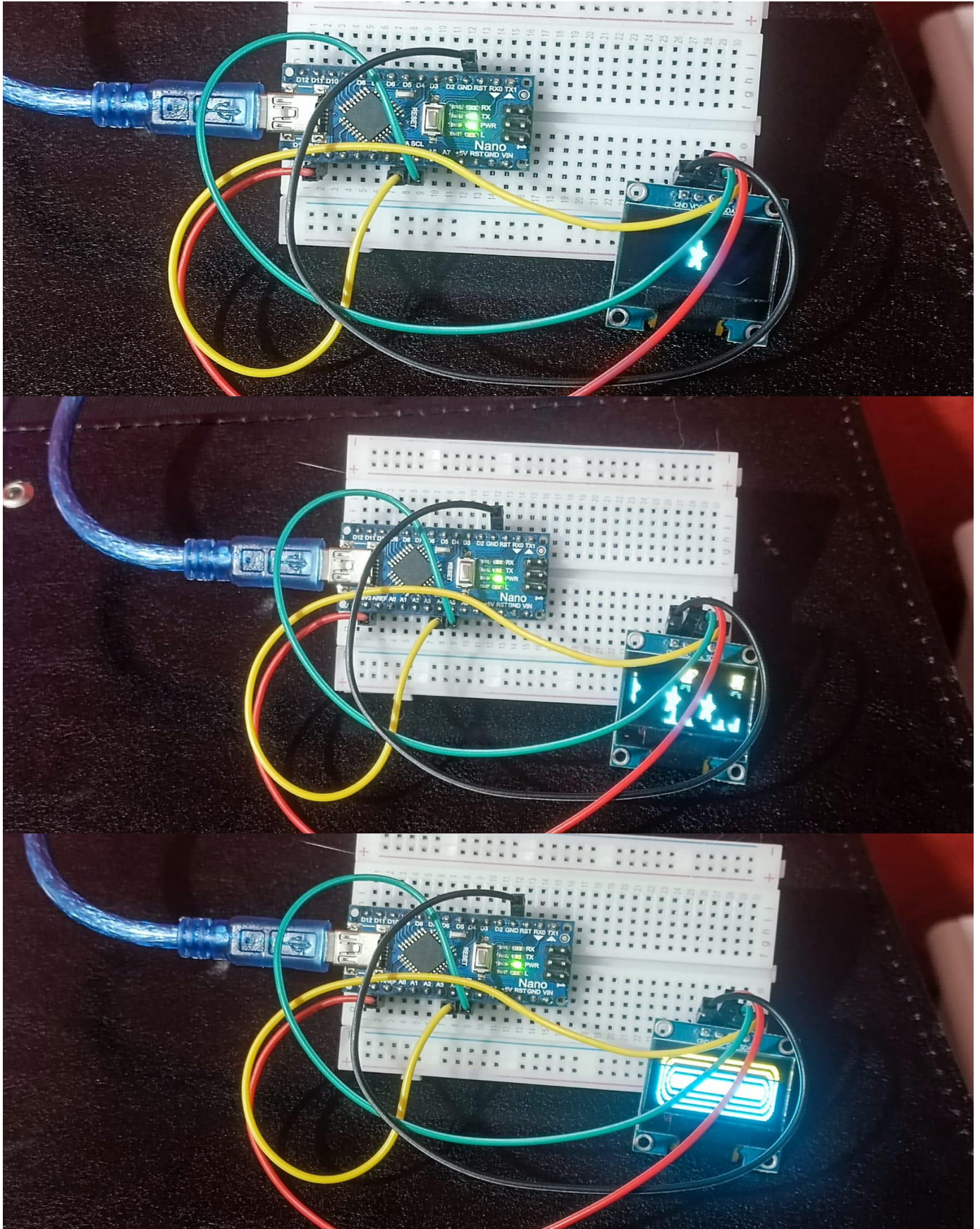
```
1 #include <SPI.h>
2 #include <Wire.h>
3 #include <Adafruit_GFX.h>
4 #include <Adafruit_SSD1306.h>
5
6 #define SCREEN_WIDTH 128 // Largura do Display OLED, em pixels
7 #define SCREEN_HEIGHT 64 // Altura do Display OLED, em pixels
8 #include "Arduino.h"
9 #include "uRTCLib.h"
10
11 // uRTCLib rtc;
12 uRTCLib rtc(0x68);
13
14 char daysOfTheWeek[7][12] = {"Domingo", "Segunda-Feira", "Terça-Feira", "Quarta-Feira", "Quinta-Feira", "Sexta-Feira", "Sábado"};
15
16 // // Declaração para display SSD1306 conectado usando software SPI (caso padrão):
17 #define OLED_MOSI 9
18 #define OLED_CLK 10
19 #define OLED_DC 11
20 #define OLED_CS 12
21 #define OLED_RESET 13
22 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT,
23 OLED_MOSI, OLED_CLK, OLED_DC, OLED_RESET, OLED_CS);
24 void setup() {
25   Serial.begin(9600);
26   rtc.set(0, 0, 3, 3, 11, 10, 22);
27   URTCLIB_WIRE.begin();
28
29   // SSD1306_SWITCHCAPVCC = gera tensão de display de 3,3V internamente
30   if(!display.begin(SSD1306_SWITCHCAPVCC)) {
31     Serial.println(F("Falha na alocação de SSD1306"));
32     for(;;); // Não prossiga, faça um loop para sempre
33   }
34 }
35
36 }
37
38 void loop() {
39
40
41   Serial.print("Configurando data e hora");
42   Serial.print(rtc.year());
43   Serial.print('2023');
44
45   Serial.print(rtc.month());
46   Serial.print('07');
47   Serial.print(rtc.day());
48
49   Serial.print(" (");
50   Serial.print(daysOfTheWeek[rtc.dayOfWeek()-4]);
51   Serial.print(") ");
52
53   Serial.print(rtc.hour());
54   Serial.print('02');
55   Serial.print(rtc.minute());
56   Serial.print('54');
57   Serial.println(rtc.second());
58
59   Serial.print("Temperatura: ");
60   Serial.print(rtc.temp() / 100);
61   Serial.print("\xc2\xb0"); //mostra o caractere de graus
62   Serial.println("C");
63
64   Serial.println();
65   delay(1000);
66   display.clearDisplay();
67
68   display.setTextSize(1);
69   display.setTextColor(WHITE);
70   display.setCursor(0, 0);
71   // Texto estático no Display
72   display.print("    DIY ELECTRONIC    ");
73   display.setTextSize(1);
74   display.setTextColor(WHITE);
75   display.setCursor(55, 50);
76   display.print("TEMP: ");
77   display.print(rtc.temp() / 100);
78   display.println("C");
79
80   display.setTextSize(3);
81   display.setTextColor(WHITE);
82   display.setCursor(0, 20);
83   // Texto estático no Display
84
85   display.print(rtc.hour());
86   display.print(':');
87   display.print(rtc.minute());
88   display.print(':');
89
90   display.println(rtc.second());
91   display.display();
92   rtc.refresh();
93
94 }
```

Imagem 9 – Código Utilizado



4. Resultados

Após todo o processo de montagem e codificação, foi possível visualizar várias formas, imagens, nomes e números. A ligação de todos os componentes foi feita de maneira correta.



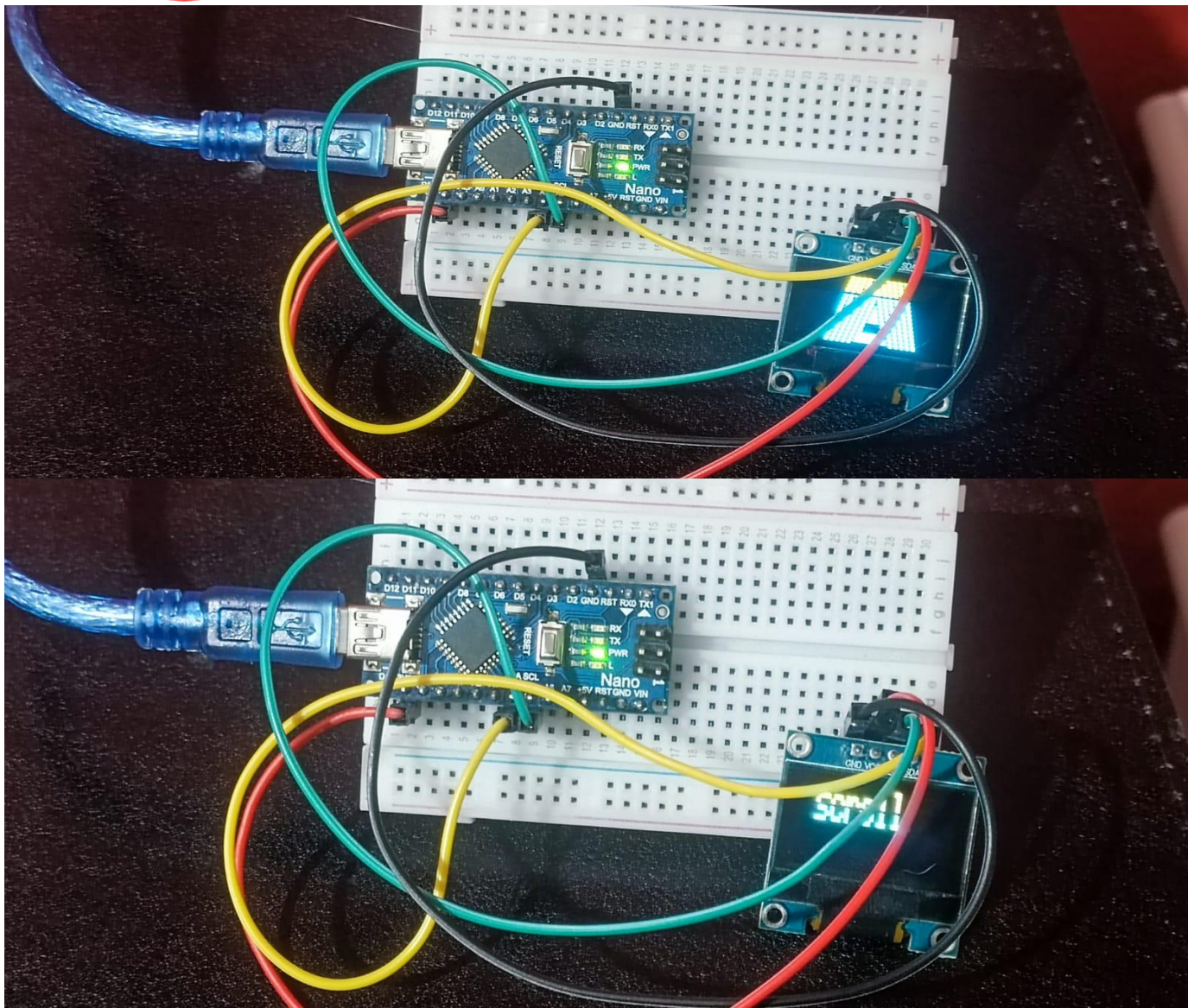


Imagem 10

5. Conclusão

A conclusão deste projeto demonstra que o Display OLED Inteligente consegue ilustrar aspectos da Internet das Coisas por meio de elementos simples do cotidiano. O objetivo principal, foi alcançado, pode-se considerar que a meta foi atingida, pois o display foi capaz de mostrar formas, imagens e nomes, que eram os principais objetivos. Apesar de alguns problemas de codificação e montagem terem impedido a exibição de informações como data e hora, esses obstáculos podem ser corrigidos com o tempo, tornando o projeto melhor. Na verdade, este projeto possui potencial para ser aprimorado. Suas vantagens é o baixo custo, potencial de expansão e facilidade de implementação. As desvantagens são limitações nas demais funções e dependência de componentes externos.

Referências

FAZEDORES. **Arduino Nano 3.0 – Conheça este pequeno e poderoso membro da família Arduino**. Fazedores. 2014. Disponível em: <https://blog.fazedores.com/arduino-nano-3-0/>. Acesso em: 26 set. 2023.

AUTO CORE ROBÓTICA. **Cabo Mini Usb para Arduino Nano**. Auto Core Robótica. Internet. Disponível em: <https://www.autocorerobotica.com.br/cabo-mini-usb-para-arduino->



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática

[nano](#). Acesso em: 26 set. 2023.

SMARTKITS. **Display OLED 128x64 0.96" I2C - Branco**. smartkits. Internet. Disponível em: <https://www.smartkits.com.br/display-oled-128x64-0-96-i2c-branco>. Acesso em: 26 set. 2023.

MAKERHERO. **Real Time Clock RTC DS3231**. MakerHero. Internet. Disponível em: <https://www.makerhero.com/produto/real-time-clock-rtc-ds3231/>. Acesso em: 26 set. 2023.



SMARTKITS. **Protoboard 400 Pontos**. smartkits. Internet. Disponível

em: [https://www.smartkits.com.br/protoboard-400-](https://www.smartkits.com.br/protoboard-400-pontos#:~:text=A%20protoboard%20de%20400%20pontos,para%20fixar%20em%20uma%20superf%C3%ADcie..)

[pontos#:~:text=A%20protoboard%20de%20400%20pontos,para%20fixar%20em%20uma%20superf%C3%ADcie..](https://www.smartkits.com.br/protoboard-400-pontos#:~:text=A%20protoboard%20de%20400%20pontos,para%20fixar%20em%20uma%20superf%C3%ADcie..) Acesso em: 26 set. 2023.

VICTOR VISION HIGH QUALITY DISPLAYS. **Arduino IDE: o que é e como instalar?**. Victor

Vision. Internet, 2023. Disponível em: [https://victorvision.com.br/blog/arduino-](https://victorvision.com.br/blog/arduino-ide/#:~:text=O%20que%20%C3%A9%20Arduino%20IDE,significa%20Ambiente%20de%20Desenvolvimento%20Integrado..)

[ide/#:~:text=O%20que%20%C3%A9%20Arduino%20IDE,significa%20Ambiente%20de%20Desenvolvimento%20Integrado..](https://victorvision.com.br/blog/arduino-ide/#:~:text=O%20que%20%C3%A9%20Arduino%20IDE,significa%20Ambiente%20de%20Desenvolvimento%20Integrado..) Acesso em: 26 set. 2023.