

ROBÓTICA



Miniestação meteorológica - II



GOVERNADOR DO ESTADO DO PARANÁ

Carlos Massa Ratinho Júnior

SECRETÁRIO DE ESTADO DA EDUCAÇÃO

Roni Miranda Vieira

DIRETOR DE TECNOLOGIA E INOVAÇÃO

Claudio Aparecido de Oliveira

COORDENADOR DE TECNOLOGIAS EDUCACIONAIS

Marcelo Gasparin

Produção de Conteúdo

Adilson Carlos Batista

Andrea da Silva Castagini Padilha

Viviane Dziubate Pittner

Validação de Conteúdo

Cleiton Rosa

Darice Alessandra Deckmann Zanardini

Revisão Textual

Kellen Pricila dos Santos Cochinski

Projeto Gráfico e Diagramação

Edna do Rocio Becker

Apoio Técnico

Equipe UFMS

2024

SUMÁRIO

Introdução	2
Objetivos desta aula	2
Roteiro da aula	3
1. Contextualização	3
2. Conteúdo	4
3. Montagem e programação	6
4. Feedback e finalização	19
Referências bibliográficas	19

Introdução

Você compreendeu na aula anterior que estações meteorológicas são ferramentas essenciais para entender o clima e fazer previsões do tempo para contribuir com diversas áreas da sociedade. Teve também a oportunidade de construir sua própria miniestação meteorológica. Nesta aula, pretendemos ampliar a estação incluindo novos componentes como tela, display OLED e buzzer para mostrar os resultados das leituras do sensor de chuva e sensor de temperatura e umidade.

Objetivos desta aula

- Desenvolver o protótipo iniciado na última aula de uma miniestação meteorológica;
- Compreender os conceitos fundamentais da meteorologia, como temperatura, umidade, pressão atmosférica e velocidade do vento;
- Aprender a construir um circuito eletrônico utilizando componentes como sensores, resistores e atuadores;
- Programar os módulos sensor de chuva e sensor de umidade e temperatura para exibir, via display OLED, os valores lidos e implementar o buzzer para mostrar intensidade de chuva.



Lista de materiais

- Arduino;
- Protoboard;
- 01 LED LGB;
- 2 Jumpers fêmea-fêmea;
- 11 Jumpers macho-macho;
- Módulo sensor DHT11;
- Sensor de chuva YL-83;
- Display OLED;
- Buzzer.

Roteiro da aula

1. Contextualização

A crescente preocupação com a poluição, como as queimadas, que interferem diretamente na qualidade do ar e aquecimento global, faz com que estações meteorológicas equipadas com componentes robóticos se tornem ferramentas indispensáveis para o monitoramento ambiental. A possibilidade de coletar de forma automatizada dados de umidade e temperatura, em tempo real e em diversos pontos estratégicos, permite identificar áreas mais vulneráveis com menor umidade relativa do ar, propensão a incêndios florestais e do acompanhamento da evolução da poluição atmosférica. Ao integrar esses dados, é possível gerar alertas precoces e auxiliar na tomada de decisões para combater as queimadas e melhorar a qualidade de vida da população, contribuindo para um planejamento urbano mais sustentável.

Uma estação meteorológica é um conjunto de instrumentos que mede, registra e coleta dados sobre as condições atmosféricas, como temperatura, umidade, pressão atmosférica, radiação solar, precipitação e velocidade do vento. Como visto na aula anterior, esses dados são cruciais para a meteorologia, a ciência que estuda a atmosfera, permitindo a compreensão e a previsão dos fenômenos climáticos.

O tempo se refere ao estado momentâneo da atmosfera em um determinado local, enquanto o clima representa o padrão médio das condições atmosféricas ao longo de um período mais extenso, como um ano ou décadas. Ao coletar dados em tempo real, as estações meteorológicas contribuem para a análise do tempo presente e para o estudo do clima passado, auxiliando na elaboração de previsões mais precisas e na compreensão das mudanças climáticas.

Figura 1 - Estação Meteorológica Embrapa



Fonte: <https://www.embrapa.br/uva-e-vinho/dados-meteorologicos>.

Em resumo, as estações meteorológicas são ferramentas essenciais para monitorar o tempo e o clima, fornecendo informações valiosas para diversas áreas, como agricultura, aviação, energia e planejamento urbano.

2. Conteúdo

OLED

OLED, sigla para *Organic Light-Emitting Diode* ou Diodo Orgânico Emissor de Luz, é uma tecnologia de display que utiliza materiais orgânicos para emitir luz. Diferentemente dos LCDs tradicionais que necessitam de uma fonte de luz externa, cada pixel em um display OLED é capaz de gerar sua própria luz, resultando em cores mais vibrantes, negros mais profundos e ângulos de visão mais amplos.

Como funciona?

Quando uma corrente elétrica passa através dos materiais orgânicos da tela, eles emitem luz. Isso permite que cada pixel seja controlado individualmente, resultando em:

- **Preto puro:** pixels desligados resultam em um preto profundo e intenso, proporcionando um contraste excepcional.
- **Cores vibrantes:** as cores são mais ricas e saturadas, com um amplo espectro de cores.
- **Ângulos de visão amplos:** a qualidade da imagem é praticamente a mesma em qualquer ângulo de visão.
- **Respostas rápidas:** os pixels respondem rapidamente às mudanças, oferecendo imagens nítidas e sem borrões, especialmente em conteúdos em movimento.

Especificações técnicas

Tensão: 3 a 5 V

Controlador: SSD1306

Cor: Branco

Nível Lógico: Compatível 3,3 V ou 5 V

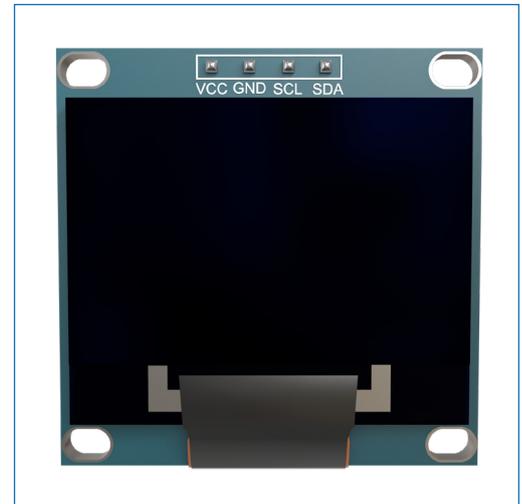
Interface: I2C

Resolução: 64 X 128 pixels

Dimensão do módulo: 27 mm X 26 mm

Dimensão da tela: 25 mm X 14 mm

Figura 2 – Display OLED



Fonte: Roberto Carlos Rodrigues, 2024.

BUZZER

Buzzer é um dispositivo eletromecânico ou eletrônico que produz um som, geralmente um zumbido ou bip, quando uma corrente elétrica passa por ele. Ele é amplamente utilizado em diversos dispositivos eletrônicos para indicar um estado, como um alarme, uma notificação ou um erro.

Tipos de buzzers:

- **Eletromecânicos:** utilizam uma pequena membrana vibratória acionada por um eletroímã. São mais baratos, mas podem ser mais ruidosos e ter uma vida útil limitada.
- **Eletrônicos:** utilizam um circuito eletrônico para gerar o som, geralmente por um pequeno alto-falante. São mais silenciosos, mais confiáveis e podem produzir uma variedade de tons.

Figura 3 – Buzzer



Fonte: Roberto Carlos Rodrigues, 2024.

Especificações técnicas:

As especificações técnicas de um buzzer podem variar dependendo do tipo e da aplicação, mas algumas das mais comuns incluem:

Tensão de alimentação: a tensão necessária para operar o buzzer. Pode variar de alguns volts a dezenas de volts. O buzzer presente no kit é de 5V.

Corrente: a quantidade de corrente que o buzzer consome quando está em funcionamento.

Frequência: a frequência do som produzido, medida em Hertz (Hz).

Nível de ruído: a intensidade do som produzido, medida em decibéis (dB).

Tipo de som: o tipo de som produzido, como contínuo, intermitente ou tonal.

3. Montagem e programação

Com os componentes em mãos, iniciamos a montagem. Primeiro, encaixamos o LED RGB na protoboard observando alguns cuidados.

O LED RGB é um componente com quatro terminais. Um comum (que pode ser ânodo ou cátodo) e os outros três que correspondem a cada uma das cores que apresenta (R-Red/vermelho, G-green/verde e B-blue/azul).

Na sequência, inserimos os resistores de 220Ω (ohms) visto que o LED é componente sensível a altas correntes e precisamos protegê-lo. A conexão direta de um LED a uma fonte de alimentação pode causar danos por excesso de corrente. Conforme figura, deixe somente o terminal comum do LED RGB sem resistor. Se tiver dúvidas, retorne à aula 34 para ver a definição do tipo de LED RGB que você tem no kit.

Depois dos componentes inseridos na protoboard, fazemos as demais conexões do RGB. Com 3 jumpers verdes macho-macho, conecte os pinos dos resistores às portas 9, 10 e 11 do Arduino. Com um jumper azul macho-macho, conecte a perna maior ao negativo da protoboard – se tiver um modelo LED RGB ânodo. Caso necessário, retome a aula 34 para ver as indicações de montagem de cada tipo de LED RGB.

Usando dois jumpers fêmea-fêmea, conecte a sonda ao módulo do sensor de chuva. Insira o módulo de sensor de chuva YL-83 na protoboard, observando sempre os pinos VCC: Alimentação do módulo (5V); GND: Terra; DO: Saída digital. Indica se está chovendo ou não (nível alto ou baixo); AO: Saída analógica. Proporciona um valor analógico que varia de acordo com a intensidade da chuva. Conforme figura, conecte com um jumper vermelho macho-macho o pino VCC do módulo do sensor ao positivo da protoboard, depois com um jumper azul macho-macho o pino GND ao negativo da protoboard e, por último, com um jumper amarelo macho-macho, conecte a saída digital (DO) à porta 3 do Arduino.

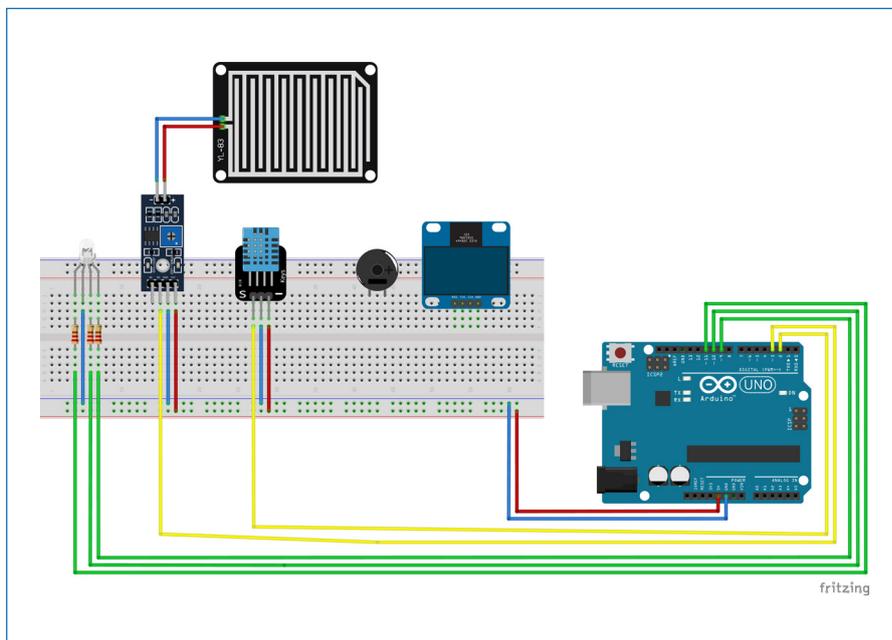
O sensor de umidade e temperatura DTH11, deve ser conectado na protoboard da seguinte maneira. O terminal sinal do DTH11 deve ser ligado com um jumper macho-macho (amarelo na Figura 6) e conectado ao pino à porta 2 do Arduino. Conecte com um jumper macho-macho o pino VCC do sensor ao positivo da protoboard e o pino GND do sensor ao negativo da protoboard.

Com um jumper macho-macho vermelho, conecte o 5V do Arduino ao positivo da protoboard e a porta GND com um jumper macho-macho azul do Arduino ao negativo na protoboard.

Finalizaremos fazendo as conexões do display OLED na protoboard, observando bem as conexões indicadas no display para confirmar a ordem dos pinos. No pino GND coloque um jumper macho-macho e ligue essa coluna até a lateral negativa da protoboard. O pino VCC deve ter um jumper ligando-o até a lateral positiva da protoboard. O pino SCL deve ser ligado com a porta A5 do Arduino e o pino SDA ligado à porta A4. Por último, as conexões do buzzer na protoboard. Conecte um jumper macho-macho do terminal positivo à lateral positiva e outro jumper do terminal negativo do buzzer à porta 8 do Arduino.

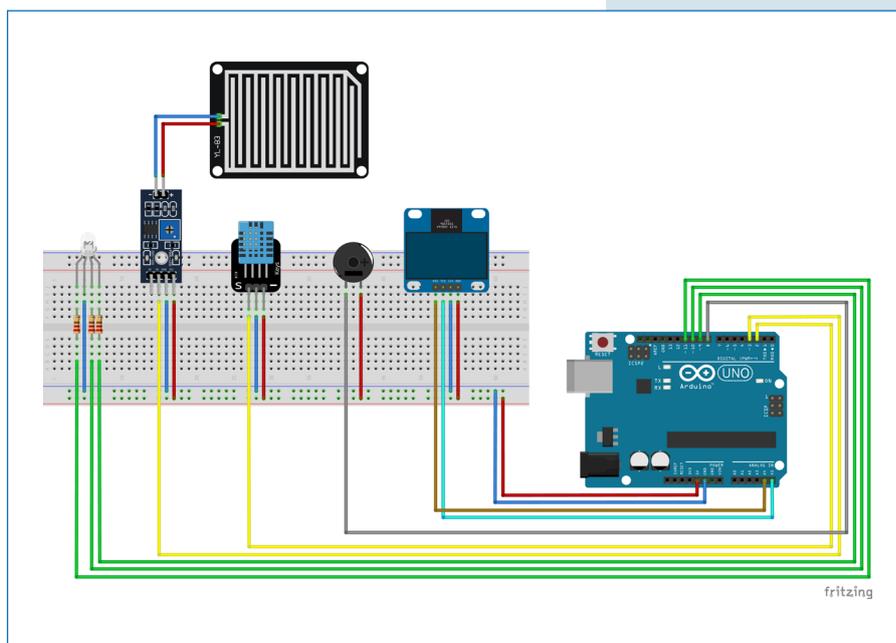
Com essas conexões, sua miniestação meteorológica (iniciada na aula anterior) estará pronta, assim, vamos para a programação.

Figura 4 – Montagem componentes na protoboard



Fonte: Fritzing, 2024.

Figura 5 – Conectando os componentes com jumpers



Fonte: Fritzing, 2024.

Programação

Iniciamos a programação sempre com o bloco **<Quando o Arduino Iniciar>** e, na sequência, definir a porta que está sendo utilizada para conectar o sensor DHT11, assim, entre em extensões, digite na busca a palavra “Cleiton” para buscar a extensão criada por esse professor. Em RP – DHT11, clique em adicionar, com isso, os blocos relacionados a esse sensor estarão disponíveis para uso no mBlock. Entre na aba “DHT11” e arraste o bloco **<Iniciar o sensor DHT na porta_____>**. Nesse parâmetro vazio, você deverá indicar qual pino deverá ser feita a leitura, em “Portas”, indicar a porta **2**.

Na sequência, indique quais as portas serão utilizadas para o LED RGB, mais uma vez deve-se baixar as extensões para o LED, entre em “Extensão”, digite “Cleiton” e em RP – LEDS, clique em adicionar.

Na aba “LEDS”, arraste o bloco **<Iniciar o LED de Cátodo comum nas portas PWM – Vermelho: 11 | Verde: 10 | Azul: 9>** indicando as portas 11 vermelho, 10 verde e 9 azul.

Figura 7 – Indicar as portas do LED RGB



Fonte: mBlock, 2024.

Da mesma forma, para controlar o display OLED, será necessário instalar a extensão correspondente. Entre em “Extensão”, digite “Cleiton” e em RP – Display OLED 128 x 64 I2C, clique em adicionar. Na aba Display OLED 128 x 64 I2C, localize e arraste os blocos **<Iniciar o display OLED 128x64 I2C com endereço 0X3C>**, inclua o bloco **<Define a cor da fonte Claro>** e em seguida arraste o bloco que vai definir o tamanho da fonte **<Tamanho da fonte 2>**. Em LEDs selecione **<Ligar LED RGB na cor R:0 | G:0 | B:0>** e em seguida, na aba Display OLED, coloque o bloco **<Limpa o display>**.

Figura 6 – Bloco início e porta do sensor



Fonte: mBlock, 2024.

Figura 8 – Configuração do display OLED

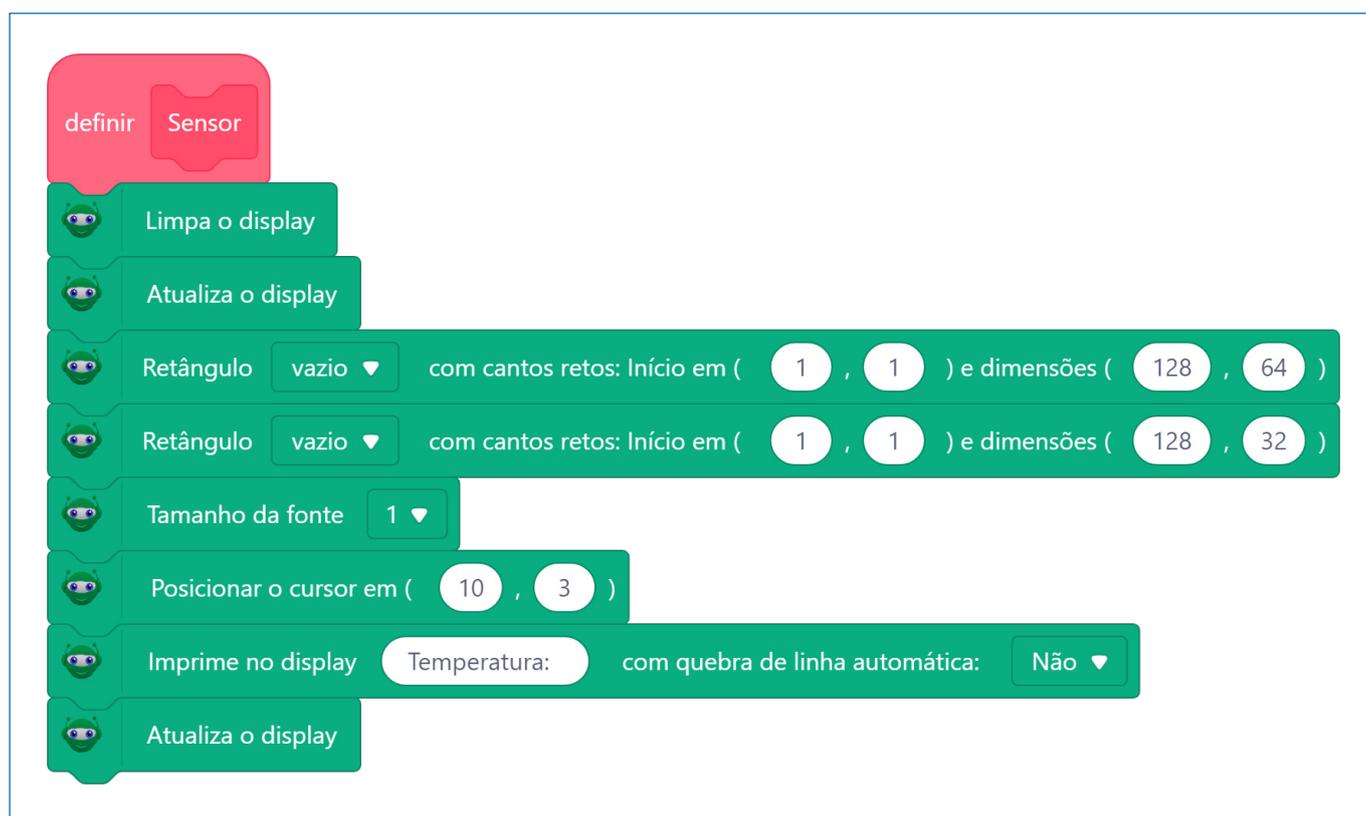


Fonte: mBlock, 2024.

O funcionamento do display OLED será dinâmico, ele mostrará as alterações captadas pelos sensores de temperatura e umidade, e para tanto, será criado um bloco personalizado, que vai indicar formatos de caixa retangular e dentro as informações coletadas em tempo real. Para fazer isso, clique na aba “Os Meus Blocos” e crie um bloco (você já fez essa ação na aula 23 Teatro de Sombras - parte 2). Nomeie o bloco como **<Sensor>**. Aparecerá no palco um bloco **<definir Sensor>**. Esse bloco vai dizer o que será feito quando o bloco **<Sensor>** for inserido na programação principal. Abaixo do bloco **<definir Sensor>**, conecte o bloco **<Limpa o display>** que está na aba Display OLED 128x64, ele prepara a tela para exibir as informações. Depois, o bloco **<Atualiza o display>**. Isto é necessário para limpar tudo o que estiver armazenado no display. Agora você irá desenhar dois retângulos para aparecer na tela. Coloque o bloco **<Retângulo [vazio] com cantos retos: Início em (1, 1) e dimensões (128, 64)>** no código e em seguida outro bloco como esse **<Retângulo [vazio] com cantos retos: Início em (1, 1) e dimensões (128, 32)>**. Atenção para a mudança no último número desse bloco (antes era 34 e agora é 32). Agora, estabeleça o tamanho da fonte, colocando o bloco **<Tamanho da fonte 1>**. Indique onde o cursor vai escrever a mensagem, para isso insira o bloco **<Posicionar o cursor em (10, 3)>**. Esse bloco determina a posição exata na tela onde o próximo texto será exibido. As coordenadas dentro dos parênteses indicam as linhas e colunas onde o texto começará. Para colocar

a mensagem "Temperatura", insira o bloco **<Imprime no Display Temperatura: com quebra de linha automática [Não]>**. Veja como fica a programação do bloco sensor até o momento. Agora, o bloco **<Atualiza o display>**.

Figura 9 - Programação parcial do bloco sensor

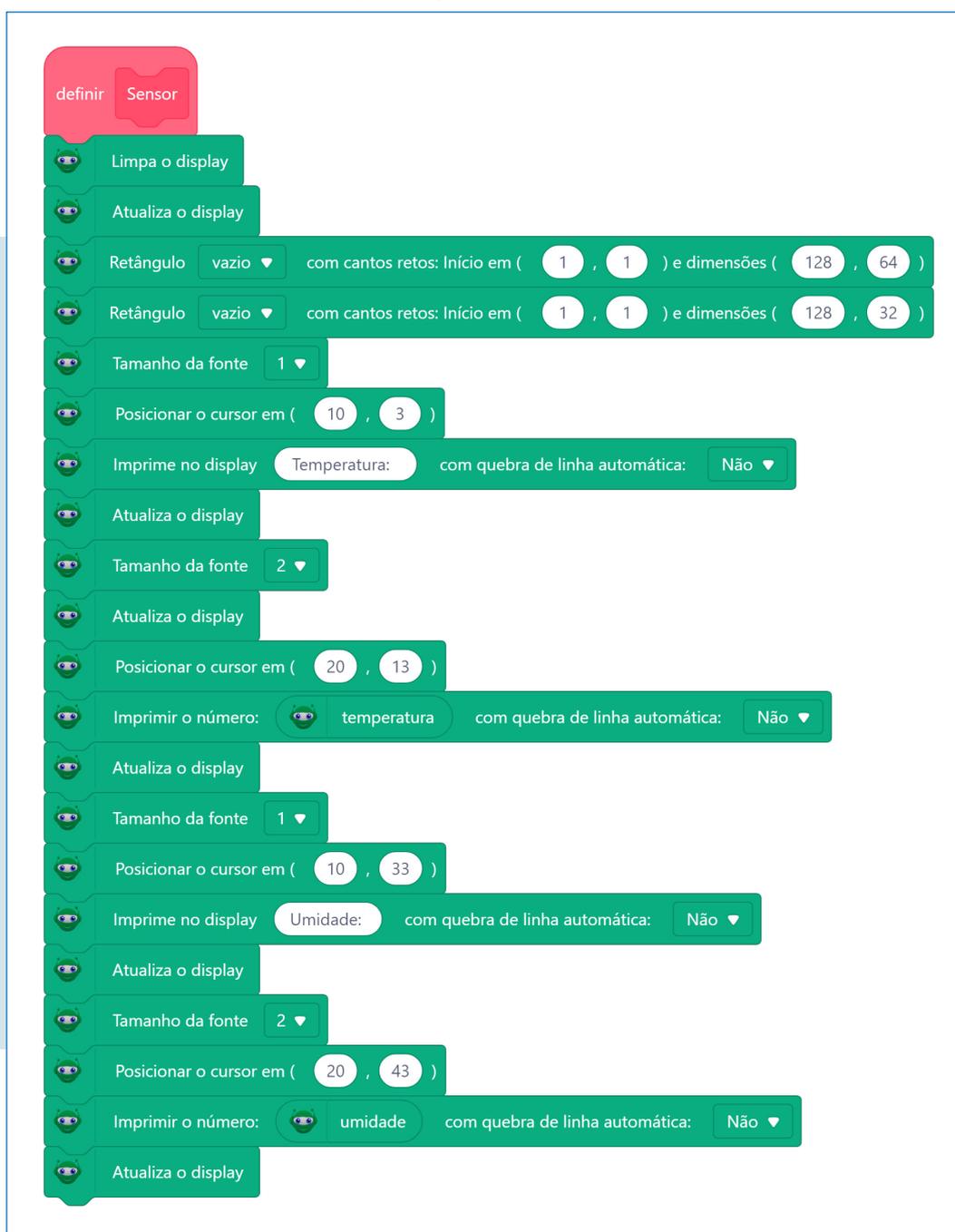


Fonte: mBlock, 2024.

Continue a programação desse bloco personalizado. Queremos que apareça abaixo de temperatura, a informação que será captada pelo sensor de temperatura. Então, abaixo do último bloco da Figura 11, insira os blocos **<Tamanho da fonte 2>**, **<Atualiza o display>** e **<Posicionar o cursor em (20, 13)>**. Agora, insira o bloco **<Imprima o número ... com quebra de linha automática [Não]>**. No lugar do número deste bloco, insira a variável **<Temperatura>**, encontrada na aba DHT11. Na sequência, o bloco **<Atualiza o display>** e **<Tamanho da fonte 1>**. O mesmo procedimento será feito para aparecer a palavra umidade e seu valor correspondente. Para isso, conecte os seguintes blocos: **<Posicionar o cursor em (10, 33)>**, **<Imprime no Display Umidade: com quebra de linha automática [Não]>**; **<Atualiza o display>** e **<Tamanho da fonte 2>**. Para aparecer no display OLED o valor da umidade, insira os

blocos a seguir: **<Posicionar o cursor em (20, 43)>**, **<Imprime o número ... com quebra de linha automática [Não]>**. No lugar do número, insira o bloco **<Umidade>** presente na aba DHT11. Finalmente, insira o bloco **<Atualiza o display>**.

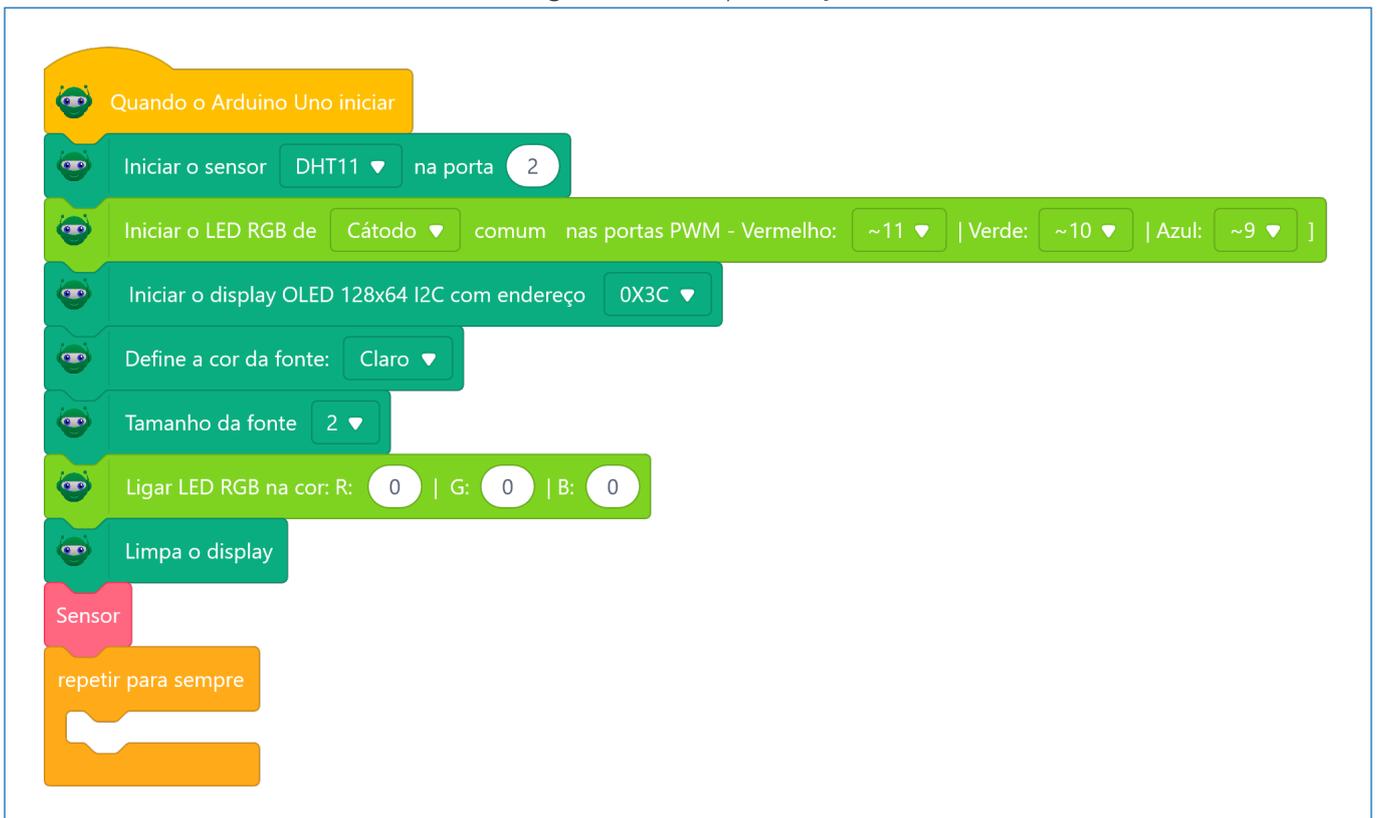
Figura 10 - Programação completa do bloco sensor



Fonte: mBlock, 2024.

Após programar o sensor, é a hora de inseri-lo na programação principal. Para isso, na aba "Os Meus Blocos" conecte o bloco **<Sensor>** abaixo do último bloco (**<Limpa o display>**). Agora, conecte o bloco de loop **<repetir para sempre>**.

Figura 11 – Loop ou laço



Fonte: mBlock, 2024.

Para definir a frequência da amostragem para o funcionamento do senso DHT11, você deve criar as variáveis, **<Contador>** e **<Estado da Chuva>**. Clique na aba Variáveis e depois em Criar uma Variável (situação já realizada em diversas aulas anteriores). Criadas as variáveis, é necessário programar como elas serão acionadas.

Para isso, selecione o bloco **<se __ então>**. No espaço do parâmetro, coloque o bloco **<__=__>**. Agora sobre esse bloco de igualdade, no primeiro parâmetro insira a variável **<Contador>**. No espaço do segundo parâmetro, o numeral 50000. Esse conjunto de blocos indica o valor que o Arduino contará para acionar o sensor DHT11, que refletirá em um intervalo temporal de segundos. Esse valor deve ser ajustado conforme o necessário. Então, se a condição do contador for atendida, deve ocorrer algo.

Queremos que o sensor DHT11 seja acionado. Para isso, vá na aba Os Meus Blocos e conecte o bloco **<Sensor>**. Abaixo dele, insira o bloco **<definir Contador para ____>**. Nesse último parâmetro, insira o numeral 0.

O Contador, quando acionado, deve ser atualizado. Para isso, conecte o bloco **<definir Contador para ____>**. Nesse último parâmetro, insira o bloco de operação de soma **<__+__>**. No primeiro parâmetro, insira a variável **<Contador>** e na segunda o numeral 1. Em seguida, o bloco **<definir Estado de chuva para ____>**. Nesse último parâmetro, indique o pino, assim, em "portas", arraste o bloco **<ler pino digital__>**, preencha o parâmetro com o número 3. Esse pino realiza a leitura da porta do sensor um valor booleano (0 ou 1), zero vai indicar a presença de chuva e 1 indica a ausência.

Na sequência, estabeleça o que a mi-

niestação meteorológica fará quando o Estado de Chuva for zero (quando começar a chover). Selecione o bloco **<se ____ então>** e no parâmetro use o operador **<__=__>**. No primeiro parâmetro, insira a variável **<Estado da Chuva>** e no segundo o numeral zero. Ou seja, se o estado de chuva for igual a 0, então realize as ações a seguir. O display OLED deve ser limpo e atualizado e deverá aparecer a palavra "Chuva" e ser atualizado. Para que isso aconteça, os seguintes blocos serão requeridos no código, na sequência apresentada: **<Limpa o display>**; **<Atualiza o display>**; **<Tamanho da fonte 4>**; **<Posicionar o cursor em (1, 20)>**; **<Imprime no Display Chuva: com quebra de linha automática [Não]>**; e finalmente **<Atualiza o display>**. Ao fazer isso, o sensor deve esperar um pequeno tempo para ligar o LED RGB e tocar o Buzzer avisando a condição detectada.

Figura 12 - Programação da condicional contador



Fonte: mBlock, 2024.

Figura 13 - Definição do estado de chuva



Fonte: mBlock, 2024.

Conecte o bloco **<esperar__segundos>**, preenchendo os parâmetros com 0,5 segundos. A seguir, arraste o bloco **<Ligar LED RGB na cor: R: 0 | G: 255 | B:255>**, novamente **<esperar__segundos>**, com o parâmetro de 1 segundo. Agora, arraste o bloco **<tocar o buzzer da porta 6 na frequência de 262 Hertz>** e repita o bloco **<esperar__segundos>**, com o parâmetro de 1 segundo. Altere o funcionamento do LED RGB, colocando o bloco **<Ligar LED RGB na cor: R: 0 | G: 0 | B:0>**, em seguida, desligue o buzzer colocando o bloco **<desligar o buzzer da porta 6>** e esperar um segundo, com o bloco **<esperar__segundos>**, com o parâmetro de 1 segundo. Por fim, o display OLED deve ser atualizado e limpo. Para tanto, inclua na sequência os blocos, **<Atualiza o display>**; **<Limpa o display>**.

Figura 14 – Estado da chuva pinos do LED RGB



Fonte: mBlock, 2024.

Para o sensor de temperatura e umidade DHT11, trabalharemos com a condicional se...então. Na aba "Controlo", arraste o bloco **<se _____ então>** e preencha os parâmetros com as seguintes informações - entre em "Operadores", arraste o bloco **<__>__>** e **<__>__>**, no primeiro parâmetro entre em DHT11 e arraste o bloco **<Temperatura>**, no segundo parâmetro, complete com o valor 30 e nos outros dois

parâmetros, para o primeiro arraste o bloco **<umidade>** e no segundo complete com 30, ou seja, se a temperatura do sensor for maior que 30 e umidade maior que 30 liga o LED RGB na cor vermelho. Insira o bloco **<Sensor>** que acionará os sensores da miniestação meteorológica. Entre na aba "LEDS" e arraste o bloco **<Ligar LED RGB na cor: R: 255 | G: 0 | B:0>** lembrando que o zero indica que a cor estará completamente apagada e 255 indica sua máxima intensidade. Em seguida, arraste o bloco **<tocar o buzzer da porta 6 na frequência de 100 Hertz>** e repetir o bloco **<esperar__segundos>**, com o parâmetro de 0,5 segundo. Altere o funcionamento do LED RGB, colocando o bloco **<Ligar LED RGB na cor: R: 0 | G: 0 | B:0>** e, em seguida, desligue o buzzer, insira os blocos **<desligar o buzzer da porta 6>** e **<esperar__segundos>**, com o parâmetro de 1 segundo.

Figura 15 - Programação condicional de temperatura e umidade



Fonte: mBlock, 2024.

Repita a sequência do bloco **< se _____ então >**, com os demais blocos dentro dele, alterando detalhes como os operadores e numerais e da cor que o LED emitirá, para as outras unidades de medidas de temperatura e umidade. Tenha atenção para os operadores que serão inseridos nessas novas sequências. Programar uma situação

em que a temperatura não excede 20 e a umidade for maior que 50, requer a inserção dos operadores `<__<__>` e `<__>__>`. O primeiro parâmetro do primeiro bloco operador menor deve ser completado com a variável `<temperatura>` e o segundo parâmetro desse mesmo bloco com o número 20. No segundo operador, o primeiro parâmetro terá a variável `<umidade>` e o segundo o numeral 50. Mantenha a sequência de blocos como da Figura 17, alterando a definição de cor do bloco `<Ligar LED RGB na cor: R: 0 | G: 0 | B: 255 >` e o valor da frequência do buzzer para 100 Hertz. Esse bloco ficará com a conformação da figura a seguir.

Figura 16 - Programação de temperatura até 20°C e alta umidade



Fonte: mBlock, 2024.

A programação completa pode ser conferida na imagem a seguir.

Figura 17 – Programação completa

```

Quando o Arduino Uno iniciar
  Iniciar o sensor DHT11 na porta 2
  Iniciar o LED RGB de Cátodo comum nas portas PWM - Vermelho: ~11 | Verde: ~10 | Azul: ~9
  Iniciar o display OLED 128x64 I2C com endereço 0X3C
  Define a cor da fonte: Claro
  Tamanho da fonte: 2
  Ligar LED RGB na cor: R: 0 | G: 0 | B: 0
  Limpa o display

repetir para sempre
  se Contador = 50000 então
    Sensor
    define Contador para 0
    definir Contador para Contador + 1
    definir Estado da Chuva para ler pino digital 3
    se Estado da Chuva = 0 então
      Limpa o display
      Atualiza o display
      Tamanho da fonte: 4
      Posicionar o cursor em: ( 1 | 20 )
      Imprime no display Chuva com quebra de linha automática: Não
      Atualiza o display
      esperar 0.5 segundo(s)
      Ligar LED RGB na cor: R: 0 | G: 255 | B: 255
      esperar 1 segundo(s)
      tocar o buzzer da porta 6 na frequência de 262 Hertz
      esperar 1 segundo(s)
      Ligar LED RGB na cor: R: 0 | G: 0 | B: 0
      desligar o buzzer da porta 6
      esperar 1 segundo(s)
      Atualiza o display
      Limpa o display
    se temperatura > 30 e umidade > 30 então
      Sensor
      Ligar LED RGB na cor: R: 255 | G: 0 | B: 0
      tocar o buzzer da porta 6 na frequência de 100 Hertz
      esperar 0.5 segundo(s)
      Ligar LED RGB na cor: R: 0 | G: 0 | B: 0
      desligar o buzzer da porta 6
      esperar 0.5 segundo(s)
    se temperatura < 20 e umidade > 50 então
      Sensor
      Ligar LED RGB na cor: R: 0 | G: 0 | B: 255
      tocar o buzzer da porta 6 na frequência de 100 Hertz
      esperar 0.5 segundo(s)
      Ligar LED RGB na cor: R: 0 | G: 0 | B: 0
      desligar o buzzer da porta 6
      esperar 0.5 segundo(s)
  
```

Fonte: mBlock, 2024.

Com isso, finalizamos nossa estação meteorológica, agora você pode averiguar o tempo da sua região.



Desafios:

Que tal...

Modificar os parâmetros das cores do LED RGB e trocar as cores programadas?

Realize o acompanhamento das variações climáticas por um período (por exemplo, um mês) e compare os dados obtidos do seu protótipo com as informações divulgadas nos meios de comunicação e sites de climatologia especializados.

Trabalhar com um módulo de comunicação sem fio como ESP01 ou Módulo Rádio nRF24L01 para enviar dados coletados para um servidor ou dispositivo?

E se...

O projeto não funcionar?

- Verifique primeiramente se os jumpers estão conectados corretamente às portas e pinos.
- Uma dica de ouro! Realize a montagem e programação simples de cada componente (por exemplo, fazer funcionar o LED RGB, depois fazer o display OLED mostrar alguma mensagem, etc.). Assim você garante que os componentes estão prontos para receberem a programação indicada.
- Revise a aula 34 para conferir se o LED inserido é cátodo ou ânodo comum, caso for a segunda opção, terá que reconfigurar a montagem e programação.



4. Feedback e finalização

- a. Confira, compartilhando seu projeto de miniestação meteorológica com os demais colegas, se o objetivo foi alcançado.
- b. Analise seu projeto desenvolvido, de modo a atender aos requisitos de fazer as leituras da temperatura e umidade, com a inserção dos componentes adicionais (LED RGB e buzzer).
- c. Reflita se as seguintes situações ocorreram:
 - i. Colaboração e cooperação: você e os membros de sua equipe interagiram entre si, compartilhando ideias que promoveram a aprendizagem e o desenvolvimento deste projeto?
 - ii. Pensamento crítico e resolução de problemas: você conseguiu identificar os problemas, analisar informações e tomar decisões de modo a contribuir para o projeto desenvolvido?
- d. Reúna todos os componentes utilizados nesta aula e os organize novamente, junto aos demais, no kit de Robótica.



REFERÊNCIAS

ARDUINO. **Documentação de Referência da Linguagem Arduino**. Disponível em: <https://www.arduino.cc/reference/pt/>. Acesso em: 27 mai. 2024.

METEORÓPOLE. **Qual a Estação Meteorológica mais antiga do mundo (ainda em operação)**. Disponível em: <https://meteoropole.com.br/2015/03/qual-a-estacao-meteorologica-mais-antiga-do-mundo/>. Acesso em 06 ago. 2024.

WIKIPEDIA. **Estação meteorológica**. Disponível em: [Estação meteorológica – Wikipédia, a enciclopédia livre \(wikipedia.org\)](https://pt.wikipedia.org/wiki/Estação_meteorológica). Acesso em 29 ago. 2024.

UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL (UFMS)
FACULDADE DE COMPUTAÇÃO (FACOM)

PROFESSORES

- Amaury Antônio de Castro Junior
- Anderson Corrêa de Lima
- Glauder Guimarães Ghinozzi
- Graziela Santos de Araújo
- Said Sadique Adi

ESTUDANTES (elaboração prévia)

- Filipe de Andrade Machado - Ciência da Computação
- Gabriel Alves Massuda Duarte - Engenharia de Computação
- José Augusto Lajo Vieira Vital - Ciência da Computação
- Lorena Valente Cavalheiro - Engenharia de Computação
- Matheus Kazumi Silva Miyashiro - Engenharia de Computação
- Nathalia dos Santos Melo - Engenharia de Software
- Yan Arruda Cunha - Engenharia de Computação
- Thiago Ferronato - Ciência da Computação
- Vitor Hugo dos Santos Duarte - Engenharia de Computação
- Wilker Sebastian Afonso Pereira - Ciência da Computação

ESTUDANTES (revisão)

- Arthur Henrique Andrade Farias - Ciência da Computação
- Bruno Pereira Wesner da Silva - Engenharia de Computação
- Fernanda das Neves Merqueades Santos - Ciência da Computação
- Gabriel Pereira Falcão - Ciência da Computação
- Jenniffer Oliveira Checchia - Ciência da Computação
- Leonardo Vargas de Paula - Sistemas de Informação
- Marcos Gabriel da Silva Rocha - Engenharia de Computação
- Maria Paula do Nascimento Santos - Engenharia de Computação
- Nathanael Martins Wink - Ciência da Computação
- Victor Luiz Marques Saldanha Rodrigues - Ciência da Computação

DIRETORIA DE TECNOLOGIAS E INOVAÇÃO (DTI)
COORDENAÇÃO DE TECNOLOGIAS EDUCACIONAIS (CTE)

EQUIPE ROBÓTICA PARANÁ

- Adilson Carlos Batista
- Ailton Lopes
- Andrea da Silva Castagini Padilha
- Cleiton Rosa
- Darice Alessandra Deckmann Zanardini
- Edna do Rocio Becker
- Kellen Pricila dos Santos Cochinski
- Marcelo Gasparin
- Michele Serpe Fernandes
- Michelle dos Santos
- Roberto Carlos Rodrigues
- Sandra Aguera Alcova Silva
- Viviane Dziubate Pittner

Os materiais, aulas e projetos da “Robótica Paraná”, foram produzidos pela Coordenação de Tecnologias Educacionais (CTE), da Diretoria de Tecnologia e Inovação (DTI), da Secretaria de Estado da Educação do Paraná (SEED), com o objetivo de subsidiar as práticas docentes com os estudantes por meio da Robótica. Este material foi produzido para uso didático-pedagógico exclusivo em sala de aula.



Este trabalho está licenciado com uma Licença
Creative Commons – CC BY-NC-SA
[Atribuição - NãoComercial - Compartilha Igual 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

