

Um exemplo de aplicação da filosofia “Faça Você Mesmo” para suprir a falta de instrumentação na fruticultura de precisão: construção de estação meteorológica de baixo custo

An example of applying the philosophy of “Do It Yourself” to overcome the lack of instrumentation in precision fruit growing: construction of a low-cost meteorological station

Luciano Gebler¹, Mirabor José Leite Júnior², Lucas de Ross Marchioretto³

¹ Pesquisador, Embrapa Uva e Vinho, Vacaria (RS), Brasil, luciano.gebler@embrapa.br

² Engenheiro Agrônomo, Mestre, Universidade Federal de Santa Maria (UFSM), Santa Maria (RS), Brasil, mirabor@outlook.com

³ Doutor em Fitotecnica, Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre (RS), Brasil, lucas.marchioretto@ufrgs.br

RESUMO

A fruticultura vem se destacando como uma das principais atividades agrícolas do país, exercendo grande influência sobre o setor social e econômico pela geração de emprego e renda em pequenas áreas e o alto valor agregado em seu produto. Por se tratar de uma atividade realizada, na maioria das vezes, de forma manual e convencional, ela possui um grande potencial de crescimento tecnológico, principalmente ao adotar os conceitos aplicados pela agricultura de precisão (AP). Por ser uma atividade que depende de fatores climáticos que em geral não podem ser controlados, o monitoramento de variáveis ambientais é uma importante ferramenta para o planejamento de ações no pomar, sejam elas preventivas ou corretivas. Um dos grandes problemas é que a agricultura, por ser exercida em um ambiente aberto sem a capacidade de controle total, quando apresenta variabilidade de resposta, demanda que a instrumentação seja dimensionada sob medida para cada local, pois o ambiente produtivo é alterado rapidamente na paisagem. Isso pode dificultar o uso de equipamentos comerciais, dado o custo elevado que normalmente é observado. Para atender a essa crescente demanda por informações e tecnologias, o presente trabalho tem como objetivo o desenvolvimento de um protótipo de uma microestação meteorológica de baixo custo baseando-se na filosofia “Faça você mesmo”. Foram utilizados sensores de temperatura e umidade do ar, luminosidade, entre outros, que coletam dados ambientais que permitem criar subsídios para tomadas de decisões. Além dos sensores, a estação possui módulo de armazenamento de dados e relógio em tempo real para serem tratados e interpretados pelo usuário final. O dispositivo construído se mostrou eficiente na coleta de dados como as estações meteorológicas comerciais e oficial do INMET, o que foi verificado por meio de testes de validação em laboratório e campo.

Palavras-chave: arduino; agricultura de precisão; monitoramento.

ABSTRACT

Fruit growing has stood out as one of the main agricultural activities in the country, exerting great influence on the social and economic sectors through the generation of employment and income in small areas and the high-added value of its product. As it is an activity carried out, most of the time, manually and conventionally, it has great potential for technological growth, especially when adopting the concepts applied by precision agriculture (PA). As it is an activity that depends on climatic factors that in most cases cannot be controlled, monitoring environmental variables is an important tool for planning actions in the orchard, whether preventive or corrective. One of the biggest problems is that as agriculture is carried out in an open environment without the capacity for total control when it presents response variability, the instrumentation must be tailored to each location since

<https://doi.org/10.4322/978-65-86819-38-0.1000044>

 Este é um capítulo publicado em acesso aberto (Open Access) sob a licença Creative Commons Attribution-NonCommercial-NoDerivatives, que permite uso, distribuição e reprodução em qualquer meio, sem restrições desde que sem fins comerciais, sem alterações e que o trabalho original seja corretamente citado.



the productive environment changes quickly in the landscape. This can make the use of commercial equipment difficult, given the high cost that is normally observed. To meet this growing demand for information and technologies, the present work aims to develop a prototype of a low-cost micrometeorological station based on the “do it yourself” philosophy. We used sensors for air temperature, humidity, and luminosity, among others, which collect environmental data that creates support for decision-making. In addition to the sensors, the station has a data storage module and a real-time clock to be processed and interpreted by the end user. The constructed device proved to be efficient in collecting data like commercial and official INMET meteorological stations, which was verified through validation tests in the laboratory and field.

Keywords: arduino; precision agriculture; monitoring.

1 INTRODUÇÃO

Nos últimos anos, tem havido uma série de tentativas de se aplicar as técnicas de agricultura de precisão (AP) utilizadas em culturas anuais ou bianuais na fruticultura, sem, entretanto, obter a mesma taxa de sucesso. Sabe-se que à medida que a planta perene envelhece, ela acumula reservas e sofre antropização expressiva, afastando-a das condições regidas somente pelo fornecimento da adubação, condição basilar para a AP em plantas de ciclo anual. A partir dessa constatação, torna-se necessária a aplicação de novas técnicas de AP voltadas às culturas perenes, sendo que o controle de dados e geração da informação ao longo do tempo tem aumentado sua importância.

À medida que o tamanho das áreas de produção aumenta, o custo da mão de obra torna-as impeditivas para a coleta, organização e manutenção de dados, inviabilizando a aplicação das técnicas necessárias. Assim, a instrumentação do pomar ou vinhedo passou a ser uma opção viável para o sucesso da fruticultura de precisão, principalmente em grandes áreas de cultivo, onde nem sempre ela está disponível, seja pela inexistência de equipamentos adaptados às condições edáficas brasileiras, ao sistema de condução adotado nos pomares, seja pelo custo do equipamento.

Para suprir esse vácuo, é possível lançar mão de princípios do movimento “Faça você mesmo”, do inglês “Do it yourself” (DIY), criado em 1912 e com grande força atualmente em diversas áreas da economia mundial (Gebler, 1997; Mintel, 2005). Em sua versão atualizada, também conhecida como movimento “maker”, esta filosofia prega que é possível criar ou consertar qualquer objeto a partir de peças simples e conhecimentos compartilhados, plenamente aplicável a qualquer sistema eletrônico, a partir de módulos de diversas plataformas disponíveis no mercado (Arduino, Adafruit, Zetta, dentre outras) ou mesmo construir utilizando uma impressora 3D (Pearce, 2012).

A principal vantagem desse movimento é a disponibilização de soluções sob medida para os problemas a baixo custo, bastando para isso a delimitação

do problema e um algoritmo para sua resolução. Uma das áreas em que esse movimento “maker” teria aplicação imediata e ampla na fruticultura hoje em dia é a coleta de informações meteorológicas pontuais em pomares e vinhedos, respeitando sua variabilidade espacial e a configuração das plantas, gerando dados microclimáticos (Paula et al., 2017).

Um dos relatos mais constantes acerca da dificuldade de sua obtenção e manutenção de uma série temporal confiável é a distância entre as estações meteorológicas disponíveis (oficiais e comerciais) e o ponto de interesse (confiabilidade dos dados).

No Brasil, o Instituto Nacional de Meteorologia (INMET) administra mais de 750 estações, convencionais ou automáticas, espalhadas por todo o território nacional, e que monitoram condições climáticas relevantes não apenas para o agronegócio brasileiro, como também para todos os outros setores da economia. Dessa forma, sua missão é prover informações meteorológicas à sociedade brasileira e influir construtivamente no processo de tomada de decisão, contribuindo para o desenvolvimento sustentável do país (Instituto Nacional de Meteorologia, 2024). Ainda assim, o volume de estações meteorológicas instaladas em nosso extenso território ainda é considerado baixo devido a sua importância, especialmente para a agricultura (Vianna et al., 2021).

Uma alternativa à falta de informações provenientes dessas estações meteorológicas, principalmente em áreas remotas ou de difícil acesso, é a aquisição de unidades comerciais (Monteiro, 2009). Porém, o alto custo de aquisição, instalação e manutenção desses equipamentos, considerado proibitivo pela maioria dos produtores, reforça a necessidade do desenvolvimento de uma solução mais barata, robusta, confiável e com potencialidades de formação de redes, principalmente se forem consideradas a utilização de sensores de baixo custo, aliada à facilidade de integração com sistemas informatizados, levando em conta a relação custo-benefício (Armani; Galvani, 2006; Krishnamurthi et al., 2015).

Sendo assim, utilizando como base a filosofia DIY e atendendo a essa crescente demanda por informações e tecnologias no monitoramento de variáveis ambientais, o presente trabalho apresenta o desenvolvimento de sistemas de sensores para coleta de dados meteorológicos pontuais (micro e mesometeorológicos) de baixo custo, permitindo sua aplicação por qualquer tamanho de produção e propriedade.

2 PRIMEIRO EXEMPLO: CONSTRUINDO UMA MICROESTAÇÃO METEOROLÓGICA

Inicialmente, devido à necessidade de execução de um trabalho envolvendo monitoramento climático em pomares, houve a necessidade de construção de uma estação meteorológica que atendesse à demanda existente dentro do prazo de um ano. Esta construção deu-se a partir da plataforma Arduino, escolhida pelo seu baixo custo, disponibilidade de peças e sensores variados no mercado varejista, além de dispor de autorização Creative Commons, tanto para hardware como software, facilitando sua programação e montagem, sem oposição de obrigações legais e comerciais e seguindo os critérios da filosofia DIY.

Por ser uma plataforma de código aberto, diversos usuários em todo mundo criam suas próprias bibliotecas e programas que podem ser compartilhados sem restrição, o que facilita o desenvolvimento de projetos e aperfeiçoamento de sistemas que utilizem este padrão (Banzi; Shiloh, 2011).

A montagem da estação foi realizada utilizando conhecimentos básicos de eletrônica e microeletrônica e consultas a sites da comunidade DIY, sem a necessidade de contratação de especialista. Inicialmente foi feita a escolha da plataforma da microcontroladora, que é uma unidade de processamento capaz de manipular variáveis do ambiente externo, transformadas em um sinal elétrico correspondente, por meio de sensores ligados aos seus terminais de entrada. Uma vez recebido o dado, ele poderá ser manipulado computacionalmente através de um ambiente de programação com linguagem específica, também conhecida pela sigla IDE (Integrated Development Environment). Posteriormente, foi feita a seleção dos sensores que deveriam fazer parte da estação, que atendiam às necessidades de controle ambiental nas plantas do pomar:

a) Sensor de temperatura do ar e umidade relativa do ar (DHT22): é um componente eletrônico de baixo custo e alta precisão que permite fazer leituras de temperaturas entre -40°C e $+80^{\circ}\text{C}$, com uma acurácia de $0,5^{\circ}\text{C}$ (máximo: $1,0^{\circ}\text{C}$), e umidade na faixa de 0 a 100%, com uma acurácia de 2,0%, sen-

do muito fácil de usar com as plataformas Arduino, Raspberry e outros microcontroladores, pois possui apenas um pino com saída digital.

- b) Sensor de luminosidade (BH1750) integrado ao módulo GY-302: módulo que atende bem a necessidade de monitoramento da luminosidade na forma digital, tendo como base o circuito integrado BH1750, com interface I2C, cuja unidade de medida em lux tem a capacidade de dizer o fluxo luminoso em uma determinada área.
- c) Sensor de molhamento foliar, na forma de placas, responde ao ambiente variando entre o estado de alta energia (high = 1), quando o clima está seco, e o estado de baixa energia (low = 0), quando há o molhamento, por exemplo, por uma gota de chuva, permitindo criar uma variação de estado de molhamento. Tais placas são revestidas em ambos os lados com um tratamento de níquel contra oxidação, melhorando assim a condutividade, desempenho e duração.
- d) Sensor de umidade do solo de fácil manuseio e leitura, mas necessita de calibração para avaliar corretamente a disponibilidade hídrica do sistema. O limite entre seco e úmido pode ser ajustado para uma escala percentual por meio do potenciômetro presente no sensor, com uma regulação do valor analógico que varia de 1 a 1024, nativo do sensor, para um valor entre 0 e 100%.
- e) Módulo Datalogger com SD Card e Relógio de Tempo Real (RTC) (DS1307): permite a coleta obtida pelos sensores e o armazenamento de informações em um cartão de memória SD, coordenados pela microcontroladora e sua programação. Além disso, o RTC tem por função fornecer informações de data e hora, fundamentais para a organização do banco de dados no cartão de memória. Isso permite também que a microcontroladora controle os intervalos de tempo para a coleta dos dados, feitas em intervalos de tempo programados.

Todos esses componentes utilizam como interligação uma matriz de contato (*proto-board*), que é útil na etapa de testes da configuração da arquitetura do dispositivo, sendo utilizada como base para a criação de circuitos eletrônicos. Ela contém furos ligados tanto no eixo vertical como no horizontal, permitindo sua conexão através de placas metálicas para cada eixo, localizada logo abaixo da superfície. Seu uso permite dispensar momentaneamente o uso de solda durante a fase de prototipagem de circuitos, efetuando-se as ligações provisórias por meio de fios conectores, também popularmente chamados de *jumpers*.

Antes de incorporar os componentes na *proto board*, foram realizados testes individuais em cada sensor diretamente no hardware (microcontroladora) para a construção do primeiro protótipo da estação, tornando possível a melhor compreensão da funcionalidade dos componentes para realização da montagem das versões funcionais.

Para o uso projetado dessa primeira estação, foi dispensado um sistema de energização complementar, utilizando-se somente baterias para o período de coleta de dados, através de um *powerbank* de 10.000 mA. Pelo consumo do equipamento montado, esta bateria foi suficiente para funcionamento do sistema por cinco dias, sendo substituído por uma bateria recarregada ao final desse período, quando eram coletados também os dados do cartão de memória.

A necessidade de energização complementar desses sistemas, que mantém a bateria sempre carregada, deve ser considerada ponto a ponto, pois haverá casos em que o aporte poderá utilizar a rede elétrica disponível próxima de pomares, ou quando não há disponibilidade de rede, seja feita através de luz solar, eólica ou mesmo de microgeradores hidrelétricos.

Uma vez solucionada a questão da energização, foi realizada a programação do protótipo da estação através do ambiente de programação do Arduino, baseando-se no apoio fornecido por sites da comunidade DIY e em *scripts* de programação já compilados e disponibilizados para alguns dos sensores (Anexo A).

Alguns dos pacotes de programação necessários ao projeto não foram encontrados prontos e tiveram de ser desenvolvidos, como, por exemplo, os dados de luminosidade, que foram convertidos em radiação solar por meio de um cálculo matemático a partir do valor aferido pela estação construída (em lux), para a unidade de medida aferida na estação meteorológica oficial do INMET (KJ/m^2), seguindo estudo realizado por Michael (2019), que estabeleceu um guia teórico e laboratorial de conversão, cujo fator de conversão é apresentado na equação 1:

$$\text{Rad} = \text{Lum} / 116 \times 3,6 \quad (1)$$

Em que:

Lum é a luminosidade (lux) e Rad é a radiação visível (KJ/m^2)

O equipamento (Figura 1) foi validado no campo em comparação com uma estação oficial do INMET e duas estações comerciais tradicionais, comprovando sua funcionalidade, robustez e confiabilidade para o trabalho (Figura 2). Posteriormente, foi operado



Figura 1. Instalação do equipamento e posicionamento dos sensores de luminosidade¹ (A), temperatura e umidade (B), molhamento foliar (B) e umidade do solo (C).

em duas condições ambientais (com e sem cobertura de tela antigranizo), avaliando seu comportamento em relação às estações comerciais instaladas próximas aos modelos DIY (Figura 3).

Os detalhes construtivos e demais informações sobre o trabalho com a estação podem ser encontrados em Leite Junior (2022).

3 SEGUNDO EXEMPLO: CONSTRUINDO E OPERANDO UM RADIÔMETRO PARA DUAS LEITURAS SIMULTÂNEAS DE LUMINOSIDADE

Em um experimento envolvendo um pomar de macieira com a cobertura de tela antigranizo, foi necessária a determinação das modificações sobre o microclima. Para isso, havia a disponibilidade de duas estações meteorológicas que somente mediam a temperatura e a umidade relativa do ar, sendo uma esta-

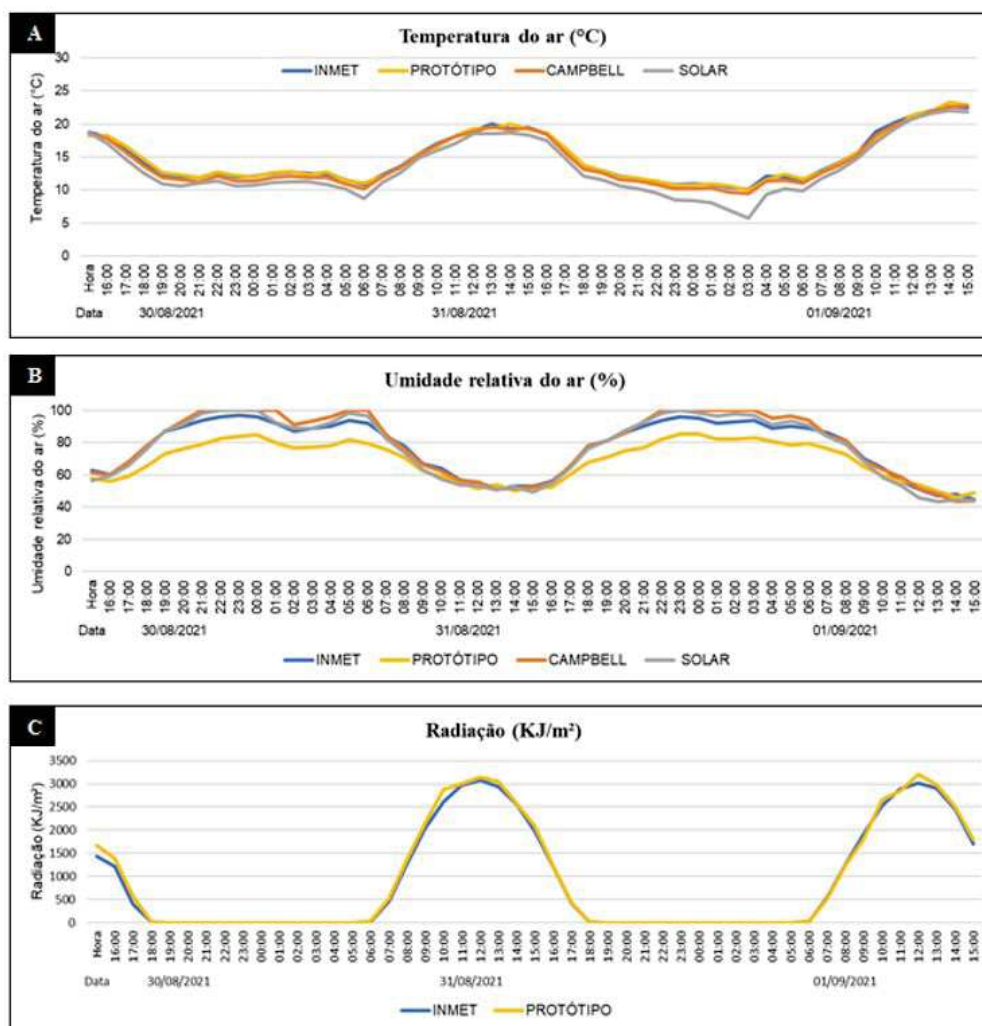


Figura 2. Resultados das comparações entre o protótipo, estações meteorológicas Campbell, Solar e padrão INMET para as variáveis de temperatura do ar (A), umidade relativa do ar (B) e radiação solar (C).

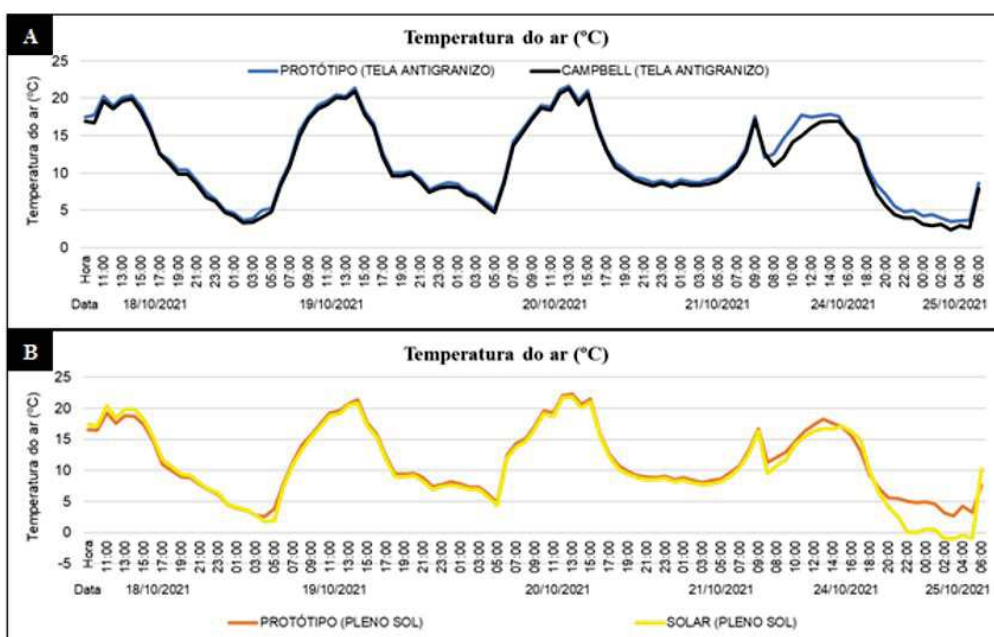


Figura 3. Temperatura do ar em pomar de macieira cultivado sob tela antigranizo (A) e a pleno sol (B) registrada por duas estações meteorológicas comerciais (Campbell e Solar) e o equipamento DIY desenvolvido.

ção instalada no meio de filas em uma área sem a cobertura de tela, e outra em um local com cobertura. Para a determinação do percentual de sombreamento imposta pela tela juntamente à estação instalada nesse ambiente, foi montado um radiômetro construído em plataforma Arduino. O radiômetro requeria dois sensores de luz para fazer leituras simultâneas, acima e abaixo da tela antigranizo (Figura 4), permitindo a leitura da radiação fotossinteticamente ativa entre 400 nm e 700 nm.



Figura 4. Montagem dos sensores de luz a campo para leituras simultâneas de luminosidade.

Para isso, foi utilizado um controlador Arduino Uno, alimentado pela tomada USB 5 V disponível no controlador de carga da estação meteorológica. Junto ao controlador foi adicionado um *shield datalogger* Adafruit v1.0 e um cartão de memória, em que a cada 15 minutos eram registrados em um arquivo de texto (.txt) a data, a hora e os valores de luminância (Lux) do sensor 1 e sensor 2 na mesma linha do texto.

A parte física (hardware) do radiômetro foi construída utilizando-se caixas plásticas Gerbox, com tampas perfuradas por um ferro de solda para a fixação com cola quente do sensor de luz BH1750, e coberta com uma metade de bolas de natal de acrílico transparente vedadas nas bordas também com cola quente. A conexão entre cada sensor e o Arduino foi feita com cabo de alarme de quatro vias (preto, vermelho, verde e amarelo). No sensor localizado acima da tela (pleno sol), a ligação dos fios foi feita na seguinte sequência: fio vermelho foi ligado à saída VCC do sensor, fio preto ao GND, amarelo à saída SCL, e verde à saída SDA. No sensor instalado abaixo da tela antigranizo, foram feitas as mesmas ligações, porém na saída ADDR do sensor foi conectado mais um fio vermelho.

Em uma placa de circuito impresso, foram soldadas quatro fileiras de pinos para a conexão dos fios, sendo uma fileira correspondente para cada saída dos sensores. Para as saídas SDA, SCL e GND, foram deixados três pinos, e para VCC foram deixados quatro pinos, pois um deles foi utilizado pela saída ADDR do sensor instalado abaixo da tela antigranizo. O primeiro pino de cada fileira de pinos do circuito impresso foi usado para ligar ambos sensores ao Arduino. O fio vermelho foi ligado à saída 3,3 V; o preto à GND; o verde à A4; e o amarelo à A5 do Arduino Uno, conforme mostrado na Figura 5.

O sensor de luz BH1750 tem como característica se comunicar com o controlador via I2C e possuir dois endereços, sendo o primeiro 0x23 e o segundo 0x5C, através do pino ADDR, para que na mesma entrada seja possível conectar dois sensores e assim fazer duas leituras simultâneas e independentes. Para que isso fosse possível, a programação usada foi retirada de um site colaborativo de programação (Ewald, 2022), tornando possível a utilização de dois sensores BH1750 simultaneamente; ajustar o intervalo de leitura através da variável “measuringTimeFactor” para 0.45 Lux; e permitir que o pico de leitura atinja até 121557 Lux, gerando leituras mais realistas a céu aberto (Anexo B).

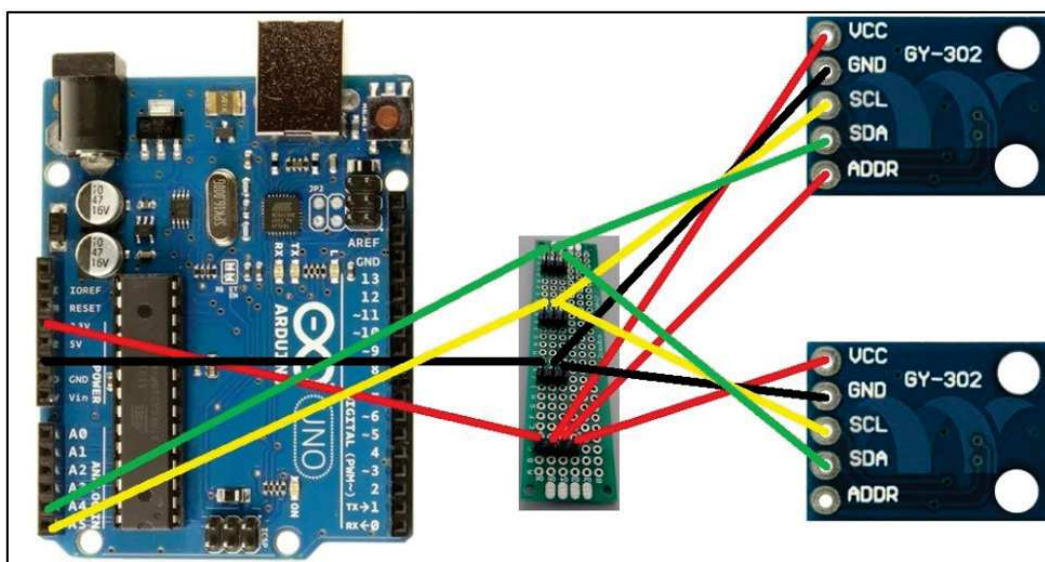


Figura 5. Esquema de ligação de dois sensores de luz BH1750 em um Arduino Uno.

4 CONSIDERAÇÕES FINAIS

Com a experiência adquirida durante o projeto, é possível afirmar que a falta de instrumentação disponível na área comercial não é necessariamente um fator inteiramente limitante. Apesar de a agricultura de precisão ser muito mais do que a existência de equipamentos, estes auxiliam sobremaneira no trabalho de coleta de dados e não devem ser descartados, caso haja condições de obtê-los de maneira simples e barata, mas confiável e funcional. A experiência do “Faça você mesmo” na fruticultura de precisão tem demonstrado que esta é uma prática útil e de grande valor, até que o mercado possa oferecer a demanda de sensores e equipamentos que o setor produtivo exige.

REFERÊNCIAS

- ARMANI, G.; GALVANI, E. Avaliação do desempenho de um abrigo meteorológico de baixo custo. **Revista Brasileira de Agrometeorologia**, v. 14, n. 1, p. 116-122, 2006.
- BANZI, M.; SHILOH, M. **Primeiros passos com o Arduino**. São Paulo: Novatec, 2011.
- EWALD, W. **BH1750FVI (GY-30, 302) ambient light sensor**. 2022. Disponível em: <https://wolles-elektronikkiste.de/en/bh1750fvi-gy-30-302-ambient-light-sensor>. Acesso em: 25 maio 2022.
- GBLER, S. Do-It-Yourself: constructing, repairing and maintaining domestic masculinity. **American Quarterly**, v. 49, n. 1, p. 66-112, 1997. DOI: <http://doi.org/10.1353/aq.1997.0007>.
- INSTITUTO NACIONAL DE METEOROLOGIA. **Sobre meteorologia**. 2024. Disponível em: <https://portal.inmet.gov.br/sobre-meteorologia>. Acesso em: 25 abr. 2024.
- LEITE JUNIOR, M. J. **Protótipo de microestação meteorológica de baixo custo para monitoramento de variáveis microclimáticas**. 2022. 66 f. Dissertação (Mestrado) – Universidade Federal de Santa Maria, Santa Maria.
- KRISHNAMURTHI, K.; THAPA, S.; KOTHARI, L.; PRAKASH, A. Arduino based weather monitoring system. **International Journal of Engineering Research and General Science**, v. 3, n. 2, p. 452-458, 2015.
- MICHAEL, P. **A conversion guide: solar irradiance and lux illuminance**. IEEE Dataport; 2019. DOI: <http://doi.org/10.21227/mxr7-p365>.
- MINTEL. **DIY Review 2005**. Mintel International Group Ltd, 2005.
- MONTEIRO, J. E. B. A. **Agrometeorologia dos Cultivos: o fator meteorológico na produção agrícola**. Brasília, DF: INMET, 2009.
- PAULA, R. C. M.; SILVA A. G.; COSTA E.; BINOTTI F. F. S. Monitoramento de variáveis micrometeorológicas em diferentes ambientes protegidos no período de inverno. **Revista de Agricultura Neotropical**, v. 4, p. 103-109, 2017. DOI: <http://doi.org/10.32404/rean.v4i5.2210>.
- PEARCE, J. M. Building research equipment with free, open-source hardware. **Science**, v. 337, n. 6100, p. 1303-1304, 2012.
- VIANNA, L. F. D. N.; PANDOLFO, C.; BLAINSKI, É.; VIEIRA, H. J. Avaliação da distribuição espacial das estações de observação climática de superfície de Santa Catarina ativas em 2020, segundo as orientações da organização mundial de meteorologia (OMM). **Revista Brasileira de Meteorologia**, v. 36, n. 3, p. 457-469, 2021. DOI: <http://doi.org/10.1590/0102-7786363001021>.

```
// --- Bibliotecas ---
#include <Arduino.h>
#include <Wire.h>
#include <BH1750.h> // Luminosidade
#include <RTCLib.h> // Relógio em Tempo Real (RTC)
#include <SPI.h> // Módulo e Cartão SD
#include <SD.h> // Módulo de Cartão SD
#include <DHT.h> // DHT22
BH1750 lightMeter (0x5c); // Endereço I2C do Sensor
de Luminosidade
// --- Variáveis do Sensor de Índice UV ---
int pino_sensor = A1; // Pino referente ao Sensor UV
int valor_sensor = 0; // Valor referente ao Sensor UV
String UV_index = "0"; // Armazena o valor do
Índice UV
// --- Variáveis do Sensor de Molhamento ---
int pino_d = 6; // Pino conectado ao Sensor de
Molhamento
int pino_a = A2; // Pino conectado ao Sensor de
Molhamento
int val_d = 0; // Armazena o valor lido do pino di-
gital do Sensor de Molhamento
int val_a = 0; // Armazena o valor lido do pino ana-
lógico do Sensor de Molhamento
// --- Variáveis do Sensor de Umidade e Temperatura ---
#define DHTPIN 7 // Pino conectado ao sensor DHT
#define DHTTYPE DHT22 // Tipo de sensor DHT
(DHTWW)
DHT dht(DHTPIN, DHTTYPE);
// --- Variáveis do Sensor de Umidade do Solo ---
#define L1 20 //
#define L2 40 //
#define L3 60 //
#define L4 80 //
#define L5 100 //
#define analogi A0 //
#define digital 11 //
#define gnd_pin 10 //
#define vcc_pin 9 //
void read_humidity(int level1, int level2, int level3,
int level4, int level5);
int adc_value = 0x00;
const int chipSelect = 10;
// --- Variáveis do Módulo de Cartão SD e RTC ---
File myFile; // Cria um arquivo para armazenar os
dados
RTC_DS1307 rtc; // RTC
void setup() {
  pinMode(pino_sensor, INPUT); // Chama o pino
do sensor UV
```

```
Serial.begin(9600);
Serial.println(F("Iniciando os sensores..."));
pinMode(analogi, INPUT);
pinMode(digital, INPUT);
pinMode(gnd_pin, OUTPUT);
pinMode(vcc_pin, OUTPUT);
digitalWrite(gnd_pin, LOW);
digitalWrite(vcc_pin, HIGH);
pinMode(pino_d, INPUT); // Referente ao sensor
de molhamento
pinMode(pino_a, INPUT); // Referente ao sensor
de molhamento
Wire.begin();
lightMeter.begin();
dht.begin();
// --- Configuração do RTC ---
if (!rtc.begin()) {
  Serial.println("Não foi possível encontrar o RTC");
  while (1);
}
else {
  rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
// Define a hora a partir da compilação do sketch
}
if (!rtc.isrunning()) {
  Serial.println("RTC não está funcionando!");
}
// --- Configuração do Módulo de Cartão SD ---
Serial.print("Inicializando cartão SD... ");
if (!SD.begin(chipSelect)) {
  Serial.println("Falha na inicialização!");
  return;
}
Serial.println("Inicialização OK.");
myFile = SD.open("DATA.txt", FILE_WRITE);
if (myFile) {
  Serial.println("Arquivo 'DADOS.TXT' aberto com
sucesso!");
  // myFile.println("Data;Hora;Temperatura;Umid
ade;Umidade_do_Solo;Indice_UV;Luminosidade"); }
myFile.close();
}
// --- Gravação dos dados do RTC no Cartão SD ---
void loggingTime() {
  DateTime now = rtc.now();
  myFile = SD.open("DATA.txt", FILE_WRITE);
  if (myFile) {
    myFile.print(now.day(), DEC);
    myFile.print("/");
    myFile.print(now.month(), DEC);
    myFile.print("/");
    myFile.print(now.year(), DEC);
    myFile.print(";");
```

```

myFile.print(now.hour(), DEC);
myFile.print(':');
myFile.print(now.minute(), DEC);
myFile.print(':');
myFile.print(now.second(), DEC);
myFile.print(",");
}
Serial.print(now.day(), DEC);
Serial.print('/');
Serial.print(now.month(), DEC);
Serial.print('/');
Serial.print(now.year(), DEC);
Serial.print(" ");
Serial.print(now.hour(), DEC);
Serial.print(':');
Serial.print(now.minute(), DEC);
Serial.print(':');
Serial.print(now.second(), DEC);
Serial.print(" ");
myFile.close();
}
void loggingTemperature() {
float t = dht.readTemperature();
float h = dht.readHumidity();
uint16_t lux = lightMeter.readLightLevel();
if (isnan(h) || isnan(t) ) {
Serial.println("Falha na leitura do sensor DHT!");
return;
}
// --- Gravação dos dados dos sensores no Cartão
SD ---
myFile = SD.open("DATA.txt", FILE_WRITE);
if (myFile) {
myFile.print(t); // Temperatura
myFile.print(",");
myFile.print(h); // Umidade
myFile.print(",");
myFile.print(adc_value); // Umidade do solo
myFile.print(",");
myFile.print(UV_index); // Índice UV
myFile.print(",");
myFile.println(lux); // Luminosidade
}
myFile.close();
}
void loop() {
loggingTime();
loggingTemperature();
// (2000);
uint16_t lux = lightMeter.readLightLevel(); // Lê os
valores do sensor de luminosidade
read_humidity(L1, L2, L3, L4, L5); // Função que lê
Sensor de Umidade do Solo

```

```

// --- Configuração do Sensor de Radiação Ultravioleta
(UV) ---
valor_sensor = analogRead(pino_sensor);
int tensao = (valor_sensor * (5.0 / 1023.0)) * 1000;
// Calcula tensão em milivolts
// Compara com valores tabela de Índice UV
if (tensao > 0 && tensao < 50)
{
UV_index = "0";
}
else if (tensao > 50 && tensao <= 227)
{
UV_index = "0";
}
else if (tensao > 227 && tensao <= 318)
{
UV_index = "1";
}
else if (tensao > 318 && tensao <= 408)
{
UV_index = "2";
}
else if (tensao > 408 && tensao <= 503)
{
UV_index = "3";
}
else if (tensao > 503 && tensao <= 606)
{
UV_index = "4";
}
else if (tensao > 606 && tensao <= 696)
{
UV_index = "5";
}
else if (tensao > 696 && tensao <= 795)
{
UV_index = "6";
}
else if (tensao > 795 && tensao <= 881)
{
UV_index = "7";
}
else if (tensao > 881 && tensao <= 976)
{
UV_index = "8";
}
else if (tensao > 976 && tensao <= 1079)
{
UV_index = "9";
}
else if (tensao > 1079 && tensao <= 1170)
{
UV_index = "10";
}

```

```

    }
    else if (tensao > 1170)
    {
        UV_index = "11";
    }
    // --- Configuração do Sensor de Temperatura e
    Umidade ---
    delay(2000); // No mínimo 2000 milissegundos
    para iniciar o DHT22
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    float f = dht.readTemperature(true);
    if (isnan(h) || isnan(t) || isnan(f)) {
        Serial.println(F("Falha na leitura do sensor DHT"));
        return;
    }
    // --- Informações apresentadas no Monitor Serial ---
    Serial.print(F("Temperatura: "));
    Serial.print(t);
    Serial.print(F("°C "));
    Serial.print(F("Umidade do ar: "));
    Serial.print(h);
    Serial.print(F("% "));
    Serial.print("Umidade do solo: ");
    Serial.print(adc_value);
    Serial.print("% ");
    Serial.print("Índice UV: ");
    Serial.print(UV_index);
    Serial.print(" Luminosidade: ");
    Serial.print(lux);
    Serial.print(" Lux ");
    // --- Configuração do Sensor de Molhamento ---
    val_d = digitalRead(pino_d); // Lê e armazena o va-
    lor do pino digital do sensor de chuva
    val_a = analogRead(pino_a); // Lê e armazena o va-
    lor do pino analógico do sensor de chuva
    // Serial.print("Valor analogico: "); // Envia as in-
    formações do sensor de molhamento MS
    // Serial.print(val_a); // Envia as informações do
    valor sensor de molhamento para o MS
    if (val_a > 600 && val_a < 1024) {
        Serial.println(" Planta seca");
    }
    if (val_a > 400 && val_a < 600) {
        Serial.println(" Planta moderadamente molhada");
    }
    if (val_a > 0 && val_a < 400) {
        Serial.println(" Planta molhada");
    }
    // delay(0000);
}
void read_humidity(int level1, int level2, int level3,
int level4, int level5)
{
    adc_value = map(analogRead(analogi), 1023, 0, 0,
100); // Lê valor do sinal analógico em %
}
// --- FIM DO CÓDIGO ---

```

ANEXO B

```
#include <Wire.h>
#include <BH1750_WE.h>
#include <DHT.h>; //INCLUSÃO DE BIBLIOTECA
// RTC
#include <Arduino.h>
#include "RTCLib.h"
#include <SPI.h> // Módulo e Cartão SD
#include <SD.h> // Módulo de Cartão SD
RTC_DS1307 RTC;
const int chipSelect = 10;
// --- Variáveis do Módulo de Cartão SD e RTC ---
File myFile; // Cria um arquivo para armazenar os
dados
//RTC_DS1307 rtc; // RTC
#define BH1750_ADDRESS 0x23
#define DHTPIN 2 //PINO DIGITAL UTILIZADO
PELO DHT22
#define DHTTYPE DHT22 //DEFINE O MODELO
DO SENSOR (DHT22 / AM2302)
DHT dht(DHTPIN,DHTTYPE);//PASSAOSPARÂMETROS
PARA A FUNÇÃO
BH1750_WE myBH1750(BH1750_ADDRESS);
// You may also pass a TwoWire object like wire2
//BH1750_WE myBH1750(&wire2,BH1750_ADDRESS);
//Variables
int chk;
float hum; //Stores humidity value
float temp; //Stores temperature value
void setup() {
  Serial.begin(9600);
  Wire.begin();
  myBH1750.init(); // sets default values: mode =
CHM, measuring time factor = 1.0
  //myBH1750.setMode(CLM); // uncomment if you
want to change default values
  myBH1750.setMeasuringTimeFactor(0.45); // un-
comment for selection of value between 0.45 and 3.68
  dht.begin(); //INICIALIZA A FUNÇÃO
  delay(2000); //INTERVALO DE 2 SEGUNDO ANTES
DE INICIAR
  Serial.println(F("Iniciando os sensores..."));
  RTC.begin();
  if (! RTC.isrunning())
  {
    RTC.adjust(DateTime(__DATE__, __TIME__));
    Serial.println("RTC is NOT running!");
  }
  // --- Configuração do Módulo de Cartão SD ---
  Serial.print("Inicializando cartão SD... ");
  if (!SD.begin(chipSelect)) {
```

```
    Serial.println("Falha na inicialização!");
    return;
  }
  Serial.println("Inicialização OK.");
  myFile = SD.open("DATA.txt", FILE_WRITE);
  if (myFile) {
    Serial.println("Arquivo 'DADOS.TXT' aberto com
sucesso!");
    // myFile.println("Data;Hora;Temperatura"); //
Cabeçalho dos dados
  }
  myFile.close();
}
void loop() {
  float lightIntensity = myBH1750.getLux();
  // Sensor readings may also be up to 2 seconds 'old'
  (its a very slow sensor)
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  delay(2000); //INTERVALO DE 2 SEGUNDOS * NÃO
DIMINUIR ESSE VALOR
  // Compute heat index in Celsius (isFahreheit = false)
  float hic = dht.computeHeatIndex(t, h, false);
  DateTime now = RTC.now();
  Serial.print(now.day(), DEC);
  Serial.print("/");
  Serial.print(now.month(), DEC);
  Serial.print("/");
  Serial.print(now.year(), DEC);
  Serial.print(" ");
  Serial.print(now.hour(), DEC);
  Serial.print(":");
  Serial.print(now.minute(), DEC);
  Serial.print(":");
  Serial.print(now.second(), DEC);
  Serial.print(F(" Luminosidade: "));
  Serial.print(lightIntensity);
  Serial.print(F(" Lux "));
  Serial.print(F(" UR: "));
  Serial.print(h);
  Serial.print(F(" % Temperatura: "));
  Serial.print(t);
  Serial.println(F(" °C "));
  // --- Gravação dos dados dos sensores no Cartão
SD ---
  myFile = SD.open("DATA.txt", FILE_WRITE);
  if (myFile) {
    DateTime now = RTC.now();
    myFile.print(now.day(), DEC);
    myFile.print("/");
    myFile.print(now.month(), DEC);
    myFile.print("/");
```

```
myFile.print(now.year(), DEC);
myFile.print(' ');
myFile.print(now.hour(), DEC);
myFile.print(':');
myFile.print(now.minute(), DEC);
myFile.print(':');
myFile.print(now.second(), DEC);
myFile.print(" ");
myFile.print(" Luminosidade: ");
myFile.print(lightIntensity);
myFile.print(" LUX ");
myFile.print(",");
myFile.print(" UR: ");
myFile.print(h);
myFile.print(",");
myFile.print("% Temperatura: ");
myFile.print(t);
myFile.println(" °C ");
}
myFile.close();
}
```