



Universidade Federal de Alagoas - UFAL
Instituto de Computação - IC

**Implementação de estufa controlada por
Arduino**

Alunos(a): Antônio Carlos,
José Renilson,
Monique Moreira e
Rafael Augusto.

Professores orientadores:
Erick Barboza e Tiago Alves.

Maceió - AL, 2023

Universidade Federal de Alagoas - UFAL

Instituto de Computação - IC

Implementação de estufa controlada por Arduino

Relatório para composição de nota prova
AB2 das disciplinas de
Microcontroladores e Aplicação e
Instrumentação Eletrônica da
Universidade Federal de Alagoas.

Alunos(a): Antônio Carlos, José Renilson,
Monique Moreira e Rafael Augusto.
Professor orientador: Erick Barboza e
Tiago Alves.

Maceió - AL, 2023

Sumário:

| | |
|---|-----------|
| Resumo: | 4 |
| Abstract: | 4 |
| Introdução: | 5 |
| Descrição do projeto: | 6 |
| Implementação: | 7 |
| Controle de Água:..... | 9 |
| Controle de Luminosidade:..... | 10 |
| Controle de Temperatura e umidade do Ar:..... | 13 |
| Controle de Umidade do Solo:..... | 14 |
| Conclusão: | 17 |
| Anexos: | |
| 18 | |
| Referência e Links: | 20 |

Resumo:

Sistemas controlados por microcontroladores estão cada vez mais presentes no cotidiano, oferecendo soluções viáveis e confiáveis nas suas mais diversas aplicações. O que a algumas décadas atrás era um sonho devido à complexidade, o preço dos equipamentos e a confiabilidade dos itens. A cada dia que se passa, esses sistemas estão ficando cada vez mais baratos, robustos, simples e confiáveis, sendo usados em praticamente todas as áreas de conhecimento, oferecendo precisão, velocidade, versatilidade e monitoração em tempo real. Esses sistemas estão sendo usados cada vez mais na agricultura e no plantio caseiro, devido a demanda nessas áreas, controlando e monitorando a irrigação, temperatura, umidade e iluminação, para as plantas crescerem mais saudáveis e rapidamente para a colheita ou para decoração.

Palavras-chave: Sistemas microcontrolados, sensores para plantação, controle de iluminação com sensor.

Abstract:

Systems controlled by microcontrollers are increasingly present in everyday life, offering viable and reliable solutions in their most diverse applications. What a few decades ago was a dream due to the complexity, the price of the equipment and the reliability of the items. With each passing day, these systems are getting cheaper, more robust, simple and reliable, being used in practically all areas of knowledge, offering accuracy, speed, versatility and real-time monitoring. These systems are being used more and more in agriculture and home planting, due to the demand in these areas, controlling and monitoring irrigation, temperature, humidity and lighting, for plants to grow healthier and faster for harvesting or for decoration.

Keywords: Microcontrolled systems, sensors for planting, lighting control with sensor.

Introdução:

Este relatório apresenta a implementação de um projeto prático da construção de uma estufa controlada por Arduino. A estufa é um ambiente controlado projetado para fornecer condições ideais de crescimento para plantas e cultivos, onde nela podemos criar um ecossistema hermético.

Ao longo da história, a humanidade tem constantemente buscado maneiras de superar as limitações e desafios decorrentes do crescimento populacional, como a escassez de alimentos e as dificuldades no cultivo de plantas. Nesse contexto, as estufas surgem como uma solução para simular ambientes favoráveis e enfrentar o desenvolvimento de espécies em condições desfavoráveis, tais como climas adversos, solos inférteis, pragas e outros problemas. Desde sua concepção, as estufas visam principal criar condições ideais para o crescimento saudável das plantas, permitindo que elas se desenvolvam de forma otimizada, independentemente dos obstáculos externos. Com os avanços tecnológicos, como o desenvolvimento de plataformas versáteis e de baixo custo com o Arduino, tornou-se possível automatizar e monitorar de forma mais precisa os parâmetros ambientais dentro das estufas.

O objetivo deste projeto é desenvolver um sistema que permita controlar e otimizar as condições internas da estufa, como temperatura, umidade, luminosidade e ventilação. Isso proporciona um ambiente favorável para o crescimento saudável das plantas, além de facilitar o cultivo de espécies que necessitam de cuidados constantes.

Neste relatório, iremos abordar a metodologia utilizada na implementação do projeto, incluindo a seleção dos sensores e atuadores, o desenvolvimento do código Arduino e a montagem do sistema de controle.

Descrição do projeto:

O projeto consiste na implementação de uma estufa controlada por Arduino, onde, em consenso com o grupo, foi idealizada inicialmente a criação de um ambiente controlado por sensores, a fim de desenvolver uma planta sem interferência humana. No interior da estufa, serão instalados sensores, como temperatura (Sensor DHT11), umidade (Sensor Higrômetro) e iluminação (Sensor LDR), além de atuadores para controlar elementos como sistemas de aquecimento, resfriamento e umidificação.

A escolha do Arduino se deu pela necessidade de uma plataforma de controle de hardware e software que permitisse o desenvolvimento de um código capaz de receber as leituras dos sensores e ajustar os parâmetros necessários para manter a planta viva. Por meio da programação, é possível estabelecer os parâmetros desejados para cada variável ambiental e criar algoritmos que monitorem constantemente as condições internas, realizando ajustes automáticos quando necessário. Por exemplo, se a temperatura atingir um nível pré-determinado, o Arduino acionará o sistema de resfriamento para reduzi-la ou quando o solo for considerado “seco” efetuar a irrigação do recipiente.

Assim com esta implementação visamos resolver alguns problemas e oferecer benefícios, como a eliminar a necessidade de intervenção manual constante, resultando em economia de tempo e recursos. Além disso, o controle preciso dos parâmetros ambientais proporciona um ambiente ótimo para o cultivo de plantas, resultando em maior produtividade e qualidade.

Implementação:

Durante a implementação do projeto foram utilizados sensores e atuadores na construção da estufa, visando criar um ambiente controlado e otimizado para o cultivo da planta. Sensores como higrômetros e termostatos foram integrados para monitorar constantemente a umidade do ar e a temperatura, fornecendo dados precisos para a tomada de decisões.

```
#include <LiquidCrystal.h>
#include <dht.h>
#include <TimerOne.h>

#define DHT_TYPE          dht11
#define DHT_PIN           9
#define COOLER_MOTOR      11
#define HIGROMETER_1      A0
#define HIGROMETER_2      A1
#define LIGHT_INPUT_UP    A2
#define LIGHT_INPUT_FRONT A3
#define LIGHT_INPUT_RIGHT A4
#define LIGHT_INPUT_LEFT  A5
#define LIGHT_OUTPUT_UP   3
#define LIGHT_OUTPUT_RIGHT 5
#define LIGHT_OUTPUT_LEFT 6
#define WATER_MOTOR       10
#define RS_DISP           12
#define EN_DISP           8
#define D4_DISP           7
#define D5_DISP           4
#define D6_DISP           2
#define D7_DISP           1

struct variables_t {
    uint8_t light_up;           //guarda o valor analógico que vai para
os leds de cima
    uint8_t light_right;        //guarda o valor analógico que vai para
os leds da direita
    uint8_t light_left;         //guarda o valor analógico que vai para
os leds da esquerda
    uint8_t hygrometer_average;
```

```

    bool motor_status;
    float humidity;
    float temperature;
    uint8_t cooler_speed;
    bool water_the_plant;
} self = {
    .light_up          = 0,          //Inicia light_up em 0
    .light_right       = 0,          //Inicia light_right em 0
    .light_left        = 0,          //Inicia light_left em 0
    .hygrometer_average = 0,          //Inicia hygrometer_average em 0
    .motor_status      = true,
    .humidity          = 0,
    .temperature       = 0,
    .cooler_speed      = 0,
    .water_the_plant   = false,
};

float AD_RESO = (4.8/1024.0); //resolução do canal AD (pinos A0 ao A5)

LiquidCrystal lcd(RS_DISP, EN_DISP, D4_DISP, D5_DISP, D6_DISP, D7_DISP);
dht dht_sensor;//(DHT_PIN, DHT_TYPE);

void setup() {
    pinMode(COOLER_MOTOR, OUTPUT);
    pinMode(LIGHT_OUTPUT_UP, OUTPUT);
    pinMode(LIGHT_OUTPUT_RIGHT, OUTPUT);
    pinMode(LIGHT_OUTPUT_LEFT, OUTPUT);
    pinMode(WATER_MOTOR, OUTPUT);
    lcd.begin(16, 2);
}

```

Código 1: Inicialização do firmware e configurações.

Além disso, atuadores, como sistemas de irrigação e ventilação, foram incorporados para ajustar automaticamente às condições ambientais, garantindo o crescimento saudável das plantas. Podendo assim separar a implementação em 4 partes que serão responsáveis por gerir o sistema, como descrita abaixo:

Controle de Água:

A água é um componente fundamental para o desenvolvimento saudável de uma planta, sendo responsável pela fotossíntese, transporte de nutrientes, suporte estrutural e regulação da temperatura. Para que todos esses processos ocorram de maneira satisfatória, é necessária uma determinada quantidade de água, dita pelas necessidades da planta, nem mais, nem menos.

Com isso em mente, desenvolvemos um sistema de controle encabeçado por uma mini bomba de água RS-385 de 12V, capaz de impulsionar de 1500ml a 2000ml por minuto. Seu tamanho compacto, baixo peso e precisão durante a execução em conjunto com o Arduino foram os pontos que nos fizeram escolher esse dispositivo para nosso projeto.

```
#define WATER_MOTOR      10

void callback_timer_motor() {
    self.motor_status = true;
    digitalWrite(WATER_MOTOR, LOW);
}

void water_motor_control() {
    self.motor_status = false;
    Timer1.initialize(10000000);           // 10 segundos (em
microsegundos)
    Timer1.attachInterrupt(callback_timer_motor);
    digitalWrite(WATER_MOTOR, HIGH);
}
```

Código 2: Timer controlando a bomba de água.

No código do sistema temos, primeiramente, a definição da bomba de água no pino 10. Em seguida, quando a função `water_motor_control` for chamada, ela checa o estado atual da bomba, através da variável **`self.motor_status`**, que atribui um `false`, indicando que a bomba está desligada no momento, e inicializa o temporizador `Timer1` em 10 segundos, ligando a bomba durante esse período de tempo com a linha de código **`digitalWrite(WATER_MOTOR, HIGH)`**.

Após 10 segundos, ocorreu uma interrupção indicada pela função **`Timer1.attachInterrupt()`**, que associa uma outra função, a **`callback_timer_motor()`**. Esse retorno de chamada é ativado pela

interrupção do Timer1, onde ela vai indicar que o motor estava ativado, pela linha `self.motor_status = true` e irá desligá-lo em `digitalWrite(WATER_MOTOR, LOW)`.

Controle de Luminosidade:

Em estufas controladas, temos o principal objetivo de fazer a planta crescer saudável, e para isso a luminosidade interna da estufa deve ser tida como prioridade em estufas internas, para que as plantas consigam realizar a fotossíntese e obter a sua energia.

De acordo com Ted Goldammer, autor do livro “Greenhouse Management”, a Photosynthetically active radiation (PAR, radiação fotossintética ativa), é uma faixa espectral da radiação solar de 400 a 700 nm, em que as plantas conseguem um aproveitamento maior para a realização da fotossíntese, sendo esta faixa quase que totalmente visível ao olho humano. Os fótons em comprimentos de onda mais curtos da faixa PAR são mais energéticos para as plantas, pois há uma presença maior de energia.

Com isso, sabemos que as luzes com menor comprimento de onda que está nesta faixa, melhor para a planta, logo as cores ultra violeta, azul e vermelho são de extrema importância para as plantas, dando prioridade a essa ordem exposta. Porém deve-se ter em mente um estudo feito por John Whitmark e Govindjee, onde diferentes comprimentos de onda da faixa PAR não são absorvidos pela planta da mesma forma, ou de maneira linear, onde as plantas não aproveitam bem a cor amarela e verde (figura 1). De acordo com Ted Goldammer, a luz azul e ultravioleta estimula o crescimento da planta e foliar por meio do forte crescimento das raízes e intensa fotossíntese. Já a vermelha, estimula o crescimento do caule, a formação de tubérculos e bulbos, a floração e a produção de frutos e a produção de clorofila. Também ajuda a aumentar o diâmetro do caule e promove a ramificação.

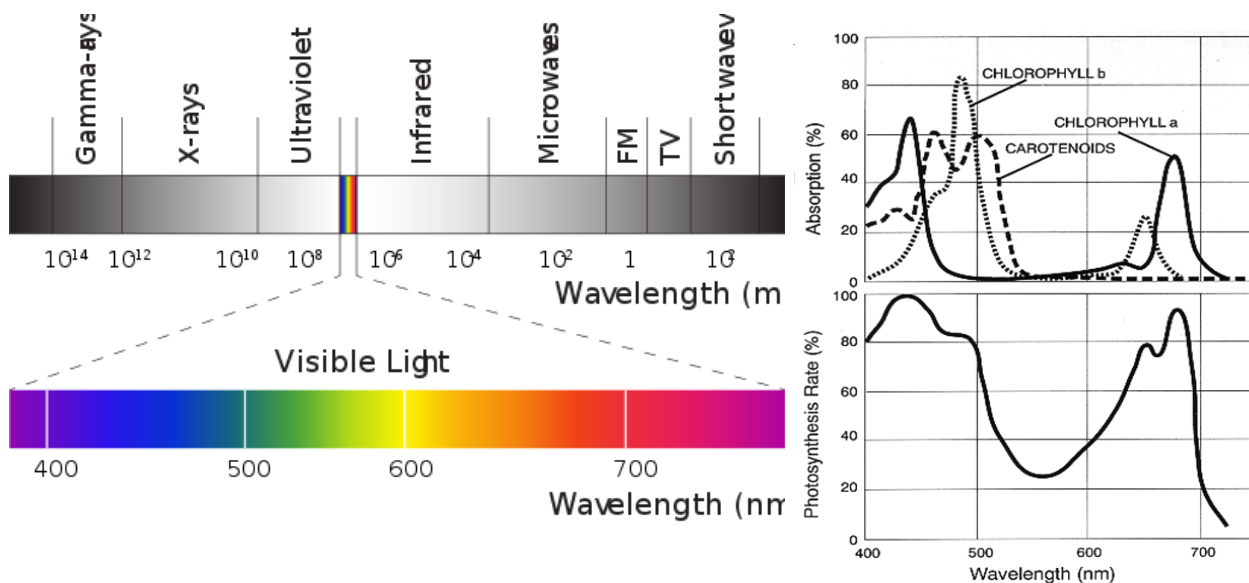


Imagem 1: luzes da faixa PAR e gráfico de absorção e fotossíntese das plantas de acordo com o comprimento de onda.

Tendo isso em mente, a luz branca emitida pelas lâmpadas como incandescente, LED e fluorescentes foram indicadas para uma estufa controlada, porém o gasto de potência e aquecimento da lâmpada deve ser considerado pelo fato de que pode matar a planta. Com a lâmpada LED sendo a forte candidata, foram usados LEDs Epistar ultravioleta, por ter um aproveitamento maior para as lâmpadas, ao liberarem fótons mais energéticos para a planta.



Imagem 2: LED Epistar ultravioleta.

Esses LEDs possuem uma potência baixa para as plantas, pois cada LED tem potência de aproximadamente 500mW, mas permitindo usar vários desses na estufa, pois seu uso é simples e o LED é pequeno. Usando 3 dispositivos com 4 LEDs em série, temos 12 LEDs com potência total de aproximadamente 6W, num comprimento de onda bom para a planta.

Para controlar esses LEDs, um controle de potência deve ser levado em consideração para evitar jogar energia demais na planta e o gasto de energia elétrica, pois não há necessidade de ligar esses LEDs se a estufa estiver em um local com uma luminosidade boa no ambiente. Com isso em mente, foram implementadas fototransistores, sensores infravermelhos, onde mesmo que a faixa de percepção do comprimento de onda do sensor seja próximo de 900nm, todos dispositivos emissores de luz, emite uma radiação infravermelho considerável. Porém, se luzes artificiais como lâmpadas a uma distância considerável da estufa pode gerar erros de leituras, logo em conjunto com os sensores, foram usados filtros passivos passa baixa, para evitar que fontes artificiais de luzes sejam percebidas pelo microcontrolador.

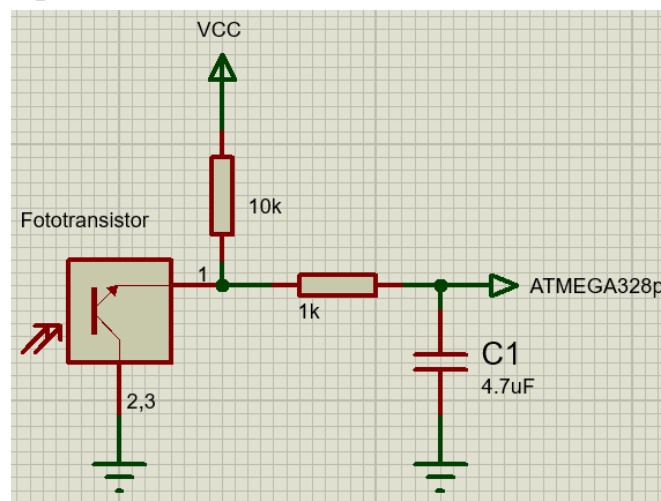


Imagem 3: Sensor infravermelho com filtro passivo passa baixa.

Considerando a fórmula de frequência de corte de um filtro passivo, temos que o filtro deixa passar frequências abaixo de:

$$f_c = \frac{1}{2\pi RC} = \frac{1}{2\pi * 1000 * 4.7 * 10^{-6}} \cong 33,86 \text{ Hz}$$

Considerando que filtro passivo atenua bastante sinais percebidos, temos então que amplificar o sinal percebido pelo microcontrolador, podendo implementar um amplificador analógico, ou amplificador no próprio microcontrolador, sendo esta última a melhor opção por ser mais simples. Foram implementados 4 desses sensores, sendo localizados na parte da frente, cima, direita e esquerda da estufa, e cada lado contribui parcialmente para cada conjunto de LEDs, permitindo ter diferentes potências nos LEDs de acordo com a luminosidade externa daquele lado. Os sensores de cima e frente tem um controle igual em todos os conjuntos de LEDs, pois a luminosidade percebida por eles

garante uma melhor distribuição de energia luminosa. Abaixo está o código correspondente implementado.

```
#define LIGHT_INPUT_UP      A2
#define LIGHT_INPUT_FRONT  A3
#define LIGHT_INPUT_RIGHT  A4
#define LIGHT_INPUT_LEFT   A5
#define LIGHT_OUTPUT_UP    3
#define LIGHT_OUTPUT_RIGHT 5
#define LIGHT_OUTPUT_LEFT  6

void light_control() {
    uint16_t lux0    = 2 * analogRead(LIGHT_INPUT_UP);
    uint16_t lux1    = 2 * analogRead(LIGHT_INPUT_FRONT);
    uint16_t lux2    = 2 * analogRead(LIGHT_INPUT_RIGHT);
    uint16_t lux3    = 2 * analogRead(LIGHT_INPUT_LEFT);
    self.light_up     = map(lux0, 0, 1023, 0, 255);
    self.light_right  = map((4*lux0) + (4*lux1) + (5*lux2) + (2*lux3),
(1024 * 6) - 1, 0, 255, 100);
    self.light_left   = map((4*lux0) + (4*lux1) + (2*lux2) + (5*lux3),
(1024 * 6) - 1, 0, 255, 100);
    analogWrite(LIGHT_OUTPUT_UP, self.light_up);
    analogWrite(LIGHT_OUTPUT_RIGHT, self.light_right);
    analogWrite(LIGHT_OUTPUT_LEFT, self.light_left);
}
```

Código 3: Leitura dos sensores com amplificação por programação e controle de luminosidade com pesos de acordo com o sensor percebido.

Controle de Temperatura e umidade do Ar:

Para fazer o controle da temperatura e umidade foi utilizado um cooler na estufa onde irá funcionar como um exaustor, com o auxílio de um sensor DHT11 para a leitura da temperatura e da umidade.

Com os valores obtidos pelo sensor, o Arduino irá processar os valores e dependendo da temperatura observada irá ligar o cooler com uma tensão específica referente a necessidade de esfriar a planta. Abaixo está o código da implementação.

```
#define DHT_TYPE      DHT11
#define DHT_PIN       9
#define COOLER_MOTOR  11
```

```
void cooler_motor_control(){
    self.humidity = dht.readHumidity();
    self.temperature = dht.readTemperature();
    if(self.temperature > 35 || self.humidity > 70){
        self.cooler_speed = map(self.temperature, 25, 35, 100, 255);
        analogWrite(COOLER_MOTOR, self.cooler_speed);
    } else {
        analogWrite(COOLER_MOTOR, 0);
    }
}
```

Código 4: Leitura do sensor DHT11 e saída de tensão para o motor.

Com as informações obtidas através:

self.humidity = dht.readHumidity()

self.temperature = dht.readTemperature()

A temperatura ideal de cultivo da rúcula é de 15 C a 18 C, o código tenta manter a planta nesse intervalo, ao observar o valor da temperatura se necessário ligar o cooler, o código através da função **map** irá passar a saída ideal para o controle.

Além disso, o sistema funcionará de forma parecida para o controle de umidade, onde caso a umidade observada seja superior a 90% o cooler também irá ligar para tentar controlar esse aumento de umidade indesejada.

Para isso foi utilizado o pino 8 com o sensor e o pino 9 o motor para o cooler.

Controle de Umidade do Solo:

Com o objetivo de monitorar e registrar a umidade do solo em tempo real, foram implementados dois sensores higrômetros compostos por duas sondas metálicas inseridas no solo que irão medir a umidade a partir da quantidade de condutividade entre as placas. Quando o solo está “úmido”, a condutividade é alta, enquanto em um solo “seco”, a condutividade é baixa.

O sensor mede a condutividade e a converte em um valor numérico que representa a umidade do solo. Esses sensores foram colocados em cada extremidade do recipiente e conectados às portas A0 e A1 do Arduino. O Arduino é responsável por coletar os dados sobre a

quantidade de água presente no solo e determinar se é necessário regar ou não a planta.

Para garantir uma medição precisa, é importante que ambos os sensores tenham leituras baixas e sejam próximas uma da outra. Isso evita variações significativas caso um lado do recipiente esteja mais úmido do que o outro. Dessa forma, a média dos valores analógicos obtidos pelos sensores é calculada por meio da variável MAH.

Aqui está o código correspondente:

```
#define HIGROMETER_1      A0
#define HIGROMETER_2      A1

void hygrometer_control() {
    uint16_t AH1 = analogRead(HIGROMETER_1); // Recebe Valor Analógico
    Higrômetro 1
    uint16_t AH2 = analogRead(HIGROMETER_2); // Recebe Valor Analógico
    Higrômetro 2
    float MAH = (AH1 + AH2)/2;

    self.hygrometer_average = map(MAH, 0, 1023, 0, 100);
    if (self.hygrometer_average < 20 && self.motor_status){
        water_motor_control();
    }
}
```

Código 5: Leitura dos sensores e calculando a média de ambos, para controlar a irrigação.

No código fornecido, a média da umidade do solo é convertida para uma escala de **0 a 100** utilizando a função **map()**, sendo armazenada na variável **self.hygrometer_average** que será mostrada no display. Em seguida, é realizada a verificação se a média é menor que **20%** e se o status do motor está ativo (**self.motor_status**). Caso ambas as condições sejam verdadeiras, isto é caso o solo seja considerado ‘seco’, a função **water_motor_control()** é chamada para acionar a bomba de água e regar a planta.

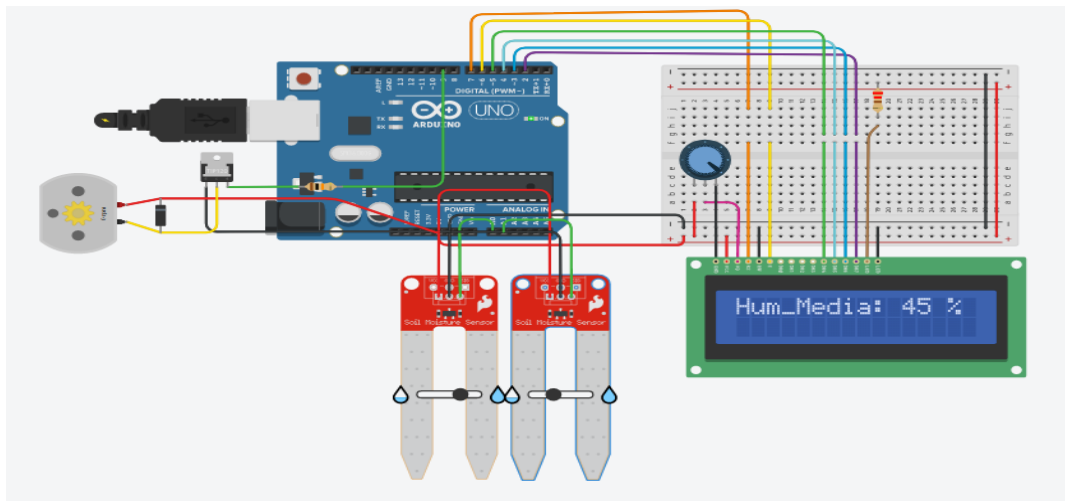


Imagem 4: Sensor Higrômetro e circuito Bomba de Água

Assim, com esse código, é possível monitorar a umidade do solo em tempo real e acionar a bomba de água precisamente, garantindo que o solo seja regado apenas quando necessário.

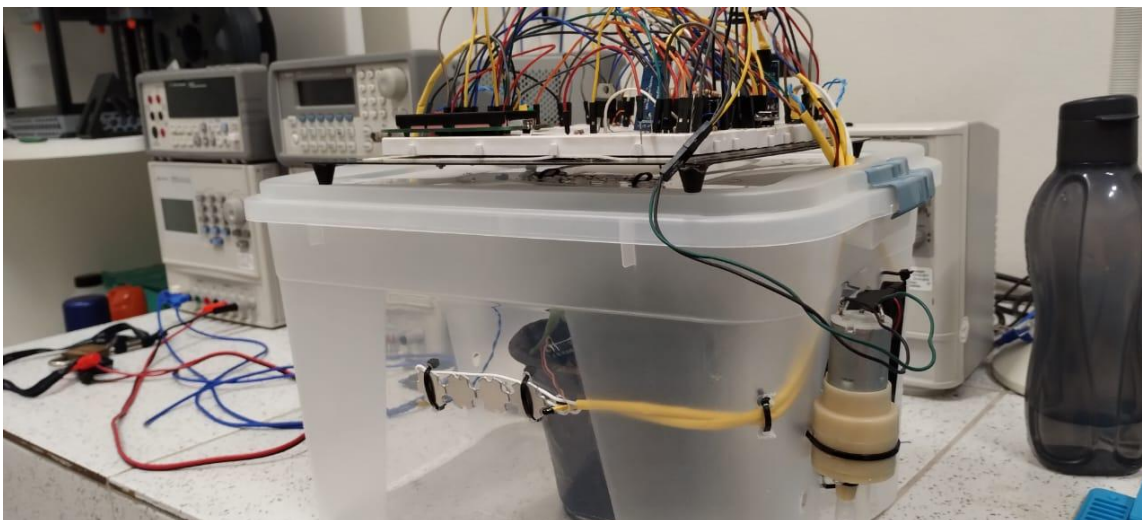
Conclusão:

Em suma, a implementação da estufa com Arduino demonstrou ser um projeto altamente eficaz e promissor para o cultivo de plantas em um ambiente controlado. A integração dos sensores de temperatura, umidade e luz, permitiu o monitoramento preciso das condições ambientais na estufa. Além disso, o controle automatizado dos sistemas de resfriamento(cooler) e irrigação(bomba de água) com base nos dados coletados proporcionou um ambiente ideal para o crescimento das plantas.

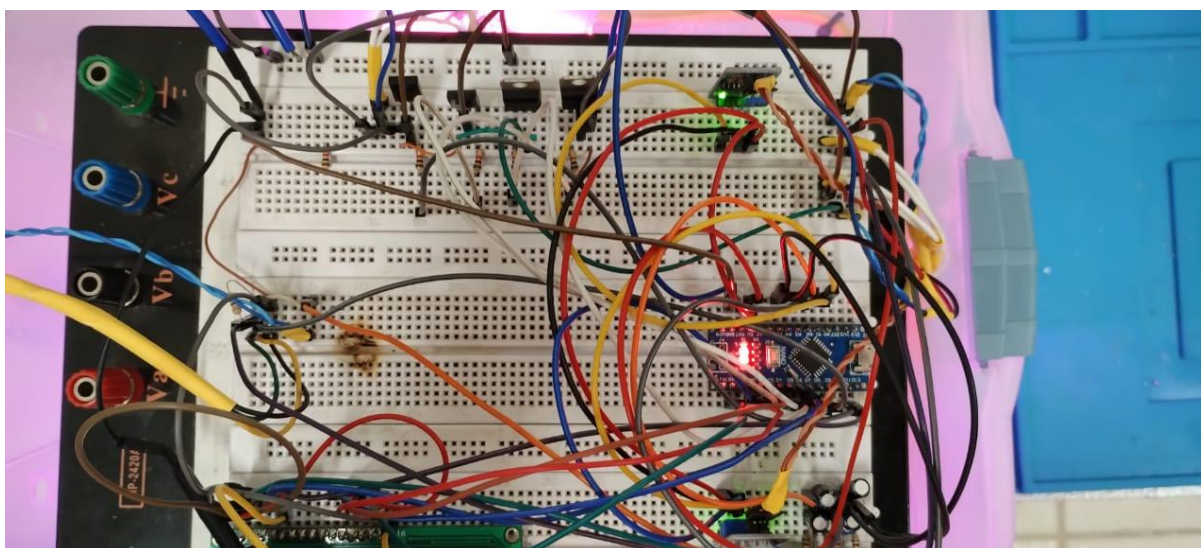
A utilização do Arduino também oferece flexibilidade e facilidade de ajuste dos parâmetros de controle, permitindo a adaptação às necessidades específicas das diferentes espécies cultivadas, como em caso de plantas que necessitam de mais luz do que outras ou plantas desérticas não podem receber tanta água. Os resultados obtidos demonstraram um aumento significativo na produtividade e qualidade das plantas, além de uma redução no consumo de recursos, como consumo de água, energia e tempo de cultivo. Futuramente o projeto pode ser implementado para ajuste automático de alguns parâmetros dependendo das necessidades da espécie ou do próprio sistema em si caso haja desgastes.

O sistema se mostra muito mais estável com um sistema de filtragem eletrônica com filtro passa baixa, onde no ambiente em que as fotos de anexo foram tiradas, o sistema demonstrou uma alteração de menos de 7% de potência percebida na fonte, e sem os filtros, a potência com as lâmpadas do laboratório acesas, o gasto era muito maior e desnecessário.

Anexos:



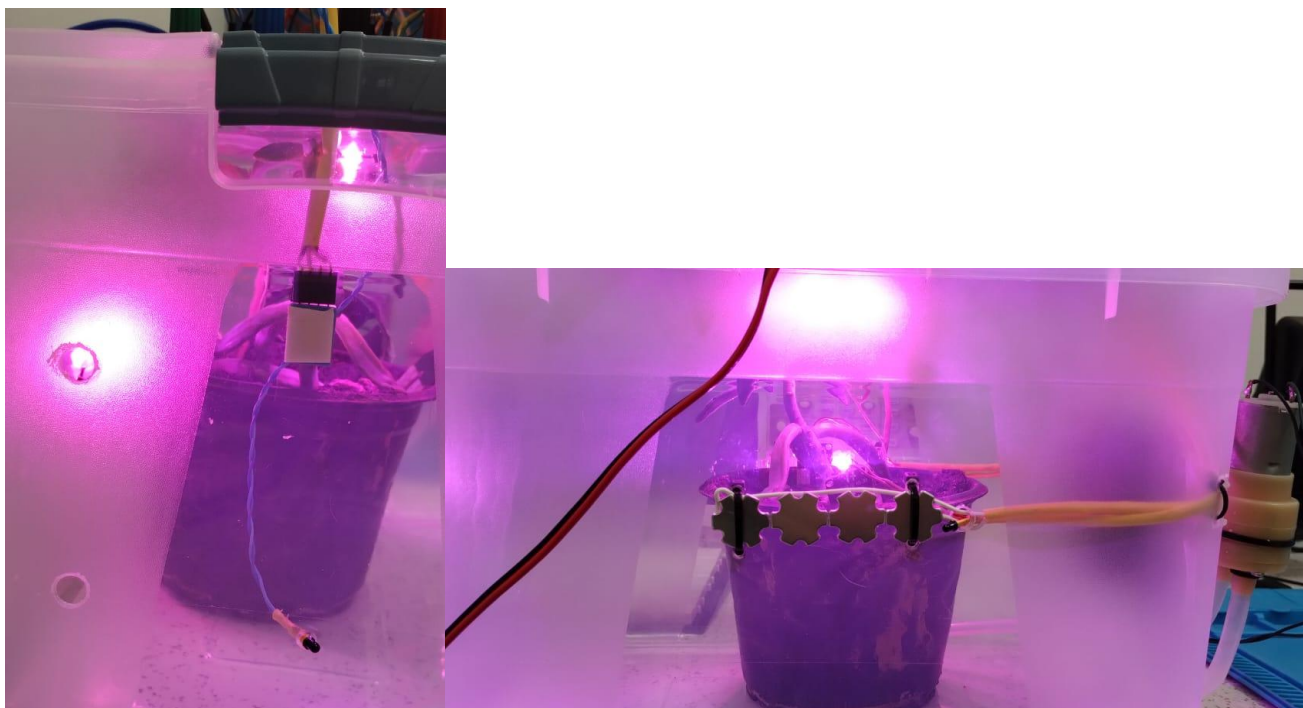
Anexo 1: Estufa desligada



Anexo 2: Circuito montado.



Anexo 3: Planta na estufa.



Anexo 4: Vista frontal e lateral da estufa.



Anexo 5: Planta na estufa.

Referência e Links:

[1] Light and Lighting Control in Greenhouses, Ted Goldammer.

Disponível em:

<https://www.greenhouse-management.com/greenhouse_management/light_lighting_control_greenhouses/light_quality.htm>. Acesso em 18/05/2023.

[2] Photosynthetically active radiation. Disponível em:

<https://en.wikipedia.org/wiki/Photosynthetically_active_radiation>.

Acesso em 18/05/2023.

[3] The photosynthetic process, John Whitmark e Govindjee. Disponível

em: <<https://www.life.illinois.edu/govindjee/paper/gov.html#61>>. Acesso em 19/05/2023.

Código no GitHub:

https://github.com/RenilsonA/Microcontroladores/blob/Renilson/estufa_controlada/Estufa_controlada.ino