UNIVERSIDADE FEDERAL DO PARANÁ

CURSO DE AGRONOMIA

CLARISSA GREBOGI BILYK

MYLLENA LACERDA DOS SANTOS

VANESSA RONCOVSKY

MONITORAMENTO DE TEMPERATURA E UMIDADE EM AMBIENTES

CURITIBA

2019

Clarissa Grebogi Bilyk (GRR20196429)

Myllena Lacerda dos Santos (GRR20194222)

Vanessa Roncovsky (GRR20194153)

MONITORAMENTO DE TEMPERATURA E UMIDADE EM AMBIENTES

Relatório apresentado à disciplina Fundamentos da Programação de Computadores do Curso de Graduação em Agronomia da Universidade Federal do Paraná.

Orientador: Prof. Jackson Antônio do Prado Lima

Curitiba, junho de 2019.

SUMÁRIO

1.INTRODUÇÃO

Baseado em sistemas não automatizados de controle de temperatura e umidade, tanto em ambientes de criação de animais quanto de cultivos em ambientes fechados, o projeto foi desenvolvido de maneira que o produtor pudesse monitorar as mudanças de temperatura e umidade de forma simples, pelo dispositivo móvel que o mesmo possua.

O sistema monitora, por meio de uma interação do sensor DHT22 (temperatura e umidade), Arduíno Mega 2560 e python, a variação de temperatura e umidade em tempo real, e produz assim um gráfico.

2.OBJETIVOS

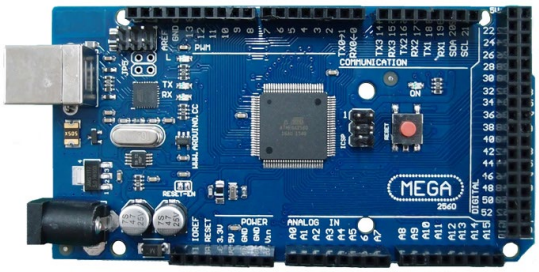
O objetivo do programa é monitorar a temperatura e umidade em tempo real, e informar o usuário quanto as variações para tanto facilitar os processos de monitoramento quanto auxiliar na produção de maneira que o produtor obtenha o lucro e o rendimento máximo do período em que o dispositivo é usado, podendo prevenir desventuras causadas pelas variações de temperatura e umidade.

3.DESENVOLVIMENTO

Para o desenvolvimento do trabalho se fez necessário a utilização de alguns materiais para o auxílio da montagem do sistema que será listado abaixo.

Arduíno Mega 2560

O **Arduino Mega 2560** é uma placa de microcontrolador baseada no ATmega2560 (datasheet). Ele possui 54 pinos de entradas/saídas digitais, 16 entradas analógicas,4 UARTs (portas seriais de hardware), um oscilador de cristal de 16 MHz, uma conexão USB, uma entrada de alimentação, uma conexão ICSP e um botão de reset. Utilizada para o funcionamento do sensor DHT22.



Fonte: https://www.makerlab-electronics.com/product/arduino-mega-2560-r3/

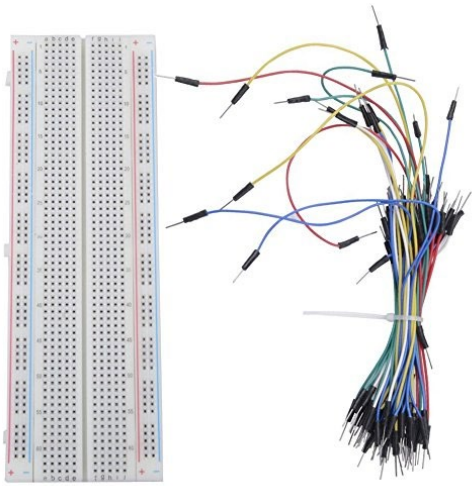
Sensor de Umidade e Temperatura DHT22

O DHT22 é um sensor de temperatura e umidade que permite fazer leituras de temperaturas entre -40 a +80 graus Celsius e umidade entre 0 a 100%, sendo muito fácil de usar com Arduino, Raspberry e outros microcontroladores pois possui apenas 1 pino com saída digital.

Fonte:https://www.filipeflop.com/produto/sensor-de-umidade-e-temperatura-am2302-dht22/

Protoboard e Jumpers

Ferramenta utilizada para montagem dos circuitos.



Fonte: https://cosmos.bluesoft.com.br/produtos/753610374589-lgdehome-mb-102-breadboard-830-tie-point-pcb-board-solderless-protoboard-kit-for-arduino-with-65pcs-m-m-flexible-breadboard-jumper-wires

Para a utilização do programa são necessários alguns softwares: Arduino IDE, Python, Biblioteca pySerial, Biblioteca MatplotLib e Biblioteca NumPy. Deve-se configurar a IDE com as informações do Arduino (que deverá estar conectado através da porta USB). Também deve-se selecionar a porta de comunicação com o Arduino.O sensor DHT22 comunica-se através de um pino digital (serial) e possui 3 pinos (+: 5 ou 3.3 Volts, out: saída serial, -: Terra).

4. SISTEMA DE BIBLIOTECA

O arquivo TemperatureHumidity.ino deve ser carregado no Arduino através da IDE. Ele é responsável por realizar a leitura do sensor e enviar os valores de umidade e temperatura, através da porta serial, para o computador.

```
#include <Adafruit_Sensor.h>

#include <DHT.h>
#include <DHT_U.h>

//Define o macro DHTPIN de acordo com o pino digital correspondente.
#define DHTPIN 7

//Selecionando o tipo de sensor (no nosso caso, o DHT22)

#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321


//Cria e inicializa os dados do sensor.
DHT dht(DHTPIN, DHTTYPE);

float umidadeAtual;
float temperaturaAtual;

// Função que executa uma vez para configurar
// variáveis e parametros

void setup() {

// Inicializa a porta serial a uma velocidade de 57600bps
Serial.begin(57600);
//Inicializa o sensor.
dht.begin();
```

```
}

// Esta função executa em loop indefinidamente após a execução da função
//setup
// mas pode ser parado por certas funções e interrupções.
void loop()
{
    // Aguarda-se 2s antes de realizar uma nova leitura do sensor.
    delay(2000);

    umidadeAtual = dht.readHumidity();
    temperaturaAtual= dht.readTemperature();

    // Checa se alguma das leituras falhou.
    if (isnan(umidadeAtual) || isnan(temperaturaAtual))
      {
        Serial.println("ERRO: Falha ao ler os dados do sensor DHT!");
        return;
      }

      // Envia através da porta serial os valores de umidade e temperatura.

      Serial.print(umidadeAtual);
      Serial.print("\t");
      Serial.print(temperaturaAtual);
      Serial.print("\n");

}
```

O arquivo TemperatureHumiditySimple.py lê os dados enviados pelo Arduino através da porta serial e configura o Bot criado para o programa no Telegram. O código coleta os dados enviados pelo arduíno, e a partir dos comandos “/umidade” ou “/temperatura” enviados pelo usuário, transmite as informações em forma de mensagem. O bot também tem como opção o comando “/plot” que utiliza as últimas informações coletadas pelo sensor (após comandos “/umidade”;”/temperatura”) e plota um gráfico pelo MatPlotLib do tipo Simple Plot.

- Importam-se as bibliotecas iniciais, Telebot, serial, serial.tools; cria-se a variável que irá conter o Token do Bot, a lista que guardará as informações coletadas pelo sensor aos comandos do usuário, e a função que irá coletar os dados do arduíno.

```python
import telebot
import serial
from serial.tools import list_ports

bot = telebot.TeleBot("859832840:AAGCqrBMfri4iC9_4prenY6Q3jML4BUfL-w")

# Salva todas as leituras de umidade/temperatura
lista = []

def le_porta():
    #Primeiramente precisamos determinar as portas seriais disponiveis na maquina
    # para isso, obtemos a lista de portas seriais e escolhemos manualmente
    # aquela com index 0.

    selectedPortIndex = 0
    selectedDevice = ""

    ports = list_ports.comports()
    #print("Avaiable ports:\n%s"%"\n".join(["\t%d: %s"%
    #(portIndex,str(ports[portIndex])) for portIndex in range(len(ports))]))

    selectedDevice = ports[selectedPortIndex].device
    #print(f"Selected device: {selectedDevice}")

    ser = serial.Serial(selectedDevice, 57600)

    # Cada execução lê uma linha da porta serial
    # e separa os dois valores (umidade e temperatura)
```

```python
    try:
        for line in ser:
            try:
                entry = line.decode("utf-8").split("\t")
                humidity = float(entry[0])
                temperature = float(entry[1])
                #print(f"T: {temperature} - H: {humidity}")
                lista.append([humidity,temperature])
                return humidity, temperature
            except ValueError as e:
                print(f"E: {line}")
                return -1, 0
            except IndexError as e:
                print(f"E: {line}")
                return -1, 0
    except KeyboardInterrupt:
        # Ao abortar a execução do programa esta exception é chamada
        # deve-se então fechar a porta serial para novas comunicações
        ser.close()
        return -1, 0
    except:
        #Caso seja um erro não especificado é importante fechar a porta
        # serial para permitir comunicação futura
        ser.close()
        return -1, 0
```

- Parte do código dirigida aos comandos do Bot.

```
    except:
        #Caso seja um erro não especificado é importante fechar a porta
        # serial para permitir comunicação futura
        ser.close()
        return -1, 0

# mensagem de Início ao comando /start do Bot
@bot.message_handler(commands=['start', 'help'])
def send_welcome(message):
    bot.reply_to(message, "Olá! Eu sou um Bot criado com a finalidade de captar as informações transmitidas pelo Sensor DHT22 integrado com um Arduíno Mega 2

# Bot informa a umidade captada pelo sensor do arduíno (ultima info passada pelo sensor) e passa para o usuário
@bot.message_handler(commands=['umidade'])
def analisa_umidade(message):
    umidade, temperatura = le_porta()

    if umidade == -1:
        bot.reply_to(message, "Erro ao obter informações")
    else:
        bot.reply_to(message,"Umidade atual: {0}".format(umidade))

# Bot informa a Temperatura captada pelo sensor do arduíno (ultima info passada pelo sensor) e passa para o usuário
@bot.message_handler(commands=['temperatura'])
def analisa_temperatura(message):
    umidade, temperatura = le_porta()

    if umidade == -1:
        bot.reply_to(message, "Erro ao obter informações")
    else:
        bot.reply_to(message,"Temperatura atual: {0}".format(temperatura))

bot.polling()
```

5. CONCLUSÃO

O programa envia os dados logo que recebe o comando, podendo assim o usuário obter um monitoramento em tempo real do local onde o sensor está.

6. REFERÊNCIAS

Multilógica: https://multilogica-shop.com/arduino-mega2560 - Acesso em 06 de junho de 2019.