

Centro Universitário Católico Salesiano Auxilium Araçatuba
Faculdade de Engenharia da Computação
Curso Superior de Engenharia da Computação

Paulo Victor Oliveira Canalle

Relatório de Projeto

Estação Meteorológica

Araçatuba/SP
2024

Paulo Victor Oliveira Canalle

Relatório de Projeto

Estação Meteorológica

Relatório parcial do projeto, apresentado como requisito parcial a obtenção do grau de Engenheiro da Computação, pelo UniSALESIANO – Araçatuba.

Orientador: Professor Amadeu Zanon Neto

Araçatuba/SP
2024

Paulo Victor Oliveira Canalle

Relatório de Projeto

Relatório parcial do projeto, apresentado como
requisito parcial a obtenção do grau de
Engenheiro de Computação, pelo
UniSALESIANO - Araçatuba

Orientador: Professor Amadeu Zanon Neto

Aprovado pela Banca Examinadora em dezembro de 2024

Prof. Me.

Prof. Me.

Prof. Me.
Araçatuba/SP
2024

RESUMO

O tema aborda uma estação meteorológica, um sistema utilizado para monitorar e coletar dados sobre as condições atmosféricas e climáticas em uma determinada região. Ao longo deste relatório técnico, exploraremos não apenas a mecânica intrincada por trás dessas estações, mas também a importância fundamental de coletar e analisar dados precisos sobre as condições atmosféricas. A jornada começa com a escolha cuidadosa dos sensores a serem empregados, cada um contribuindo de forma única para capturar informações cruciais, como temperatura, humidade, pressão atmosférica. Com isso abordará os principais aspectos relacionados à concepção, implementação e funcionamento de uma estação meteorológica, incluindo sensores utilizados, métodos de coleta e análise de dados, além de possíveis aplicações práticas dos dados coletados para diversos fins, como previsão do tempo, monitoramento ambiental e estudos científicos.

Palavras-Chave: Meteorologia, Análise Climática, Coleta de dados, Previsão do Tempo, Condições Atmosféricas

LISTA DE ILUSTRAÇÕES

Figura 1 - Arduino Uno ATmega328p.....	14
Figura 2 - Transdutor DHT11	17
Figura 3 - MCU para DHT11	18
Figura 4 -Modulo Bluetooth.....	20
Figura 5- Anemômetro	22
Figura 6 - Ky-003	24
Figura 7- Ky-003 Datasheet	25
Figura 8 - DFT de contexto	30
Figura 9 - Tela de login	31
Figura 10 - Tela de Registrar	31
Figura 11 - Tela Principal.....	32
Figura 12 - Tela dashboard.....	32
Figura 13 - Tela de consulta	33
Figura 14 - Biblioteca MySql	34
Figura 15 - Código conexão BD	35
Figura 16 - Esquema Completo.....	36
Figura 17- Modelo Pluviômetro	37
Figura 18 - Modelo Impresso (Pluviômetro).....	38
Figura 19 - Pluviômetro montado	38
Figura 20 - Dht11 datasheet	39
Figura 21 - Biblioteca DHT11	40
Figura 22 - Código dht11	41
Figura 23 - Serial monitor.....	41
Figura 24 - Esquema módulo Bluetooth.....	42
Figura 25 - Código Hc-05	43
Figura 26 - Datasheet ky-003	46
Figura 27- Código Ky-003	47
Figura 28 - Anemômetro	48
Figura 29 - Código Anemômetro	49

LISTA DE TABELAS

Tabela 1- Comparação das plataformas	13
Tabela 2 - Tamanhos de memória.	16
Tabela 3 - Pinagem Hc-05	20
Tabela 4- Configurações Ky-003	45
Tabela 5- Pinagem Ky-003	45

LISTA DE ABREVIATURAS E SIGLAS

SO – Sistema operacional

IoT – *Internet of Things* (Internet das Coisas)

PIC – Peripheral Interface Controllers (Controladores de Interface Periférica)

IDE – Integrated Development Environment (Ambiente de Desenvolvimento Integrado)

Wi-Fi – Wireless Fidelity (Fidelidade sem fio)

SRAM – Static Random Access Memory (Memória de Acesso Randômico Estática)

Kb – Kilobyte

SDIO – Secure Digital Card

JTAG – Joint Test Action Group

USB – Universal Serial Bus (Porta Serial Universal)

PC – Personal Computer (Computador Pessoal)

GPIO - General Purpose Input/Output (Entrada/Saída de Propósito Geral)

API – Application Programming Interface (Interface de Programação de Aplicativos)

Bot – Robot (Robô)

IA – Inteligência Artificial

PLN – Processamento da Linguagem Natural

JSON – JavaScript Object Notation (Notação de Objetos JavaScript)

HTTP – HyperText Transfer Protocol (Protocolo de Transferência de Hipertexto)

SUMÁRIO

1. INTRODUÇÃO	10
2. TECNOLOGIAS DE DESENVOLVIMENTO	11
2.1. <i>Hardware</i>	11
2.2. Arduino uno ATmega328	13
2.3. Descrição dos Pinos	15
2.4. Comparação entre ATmega48P, ATmega88P, ATmega168P e ATmega328P	16
2.5. Componentes	17
3. TRANSDUTOR DHT11	17
3.1. Fonte e Pin	18
3.2. Comunicação	18
4. MODULO BLUETOOTH HC-05	19
4.1.1. Frequência e Alcance	19
4.2. Funcionamento do HC-05	20
4.2.1. Modo de comando	20
4.2.2. Modo de dados	20
4.2.3. Configurações padrão do HC-05	21
5. ANEMÔMETRO	21
5.1. Tipos de anemômetro	22
5.1.1. Anemômetros de copo	22
5.1.2. Anemômetro de moinhos de vento	22
5.1.3. Anemômetro termoeletrico	22
5.1.4. Anemômetros ultrassônicos	22
5.2. Quem precisa de um anemômetro?	23
6. SENSOR HALL KY-003	24
6.1. Conceito de magnetismo	24
6.2. Como funciona?	25
6.3. Especificações	25
6.4. Aplicações	26
7. BANCO DE DADOS	26
8. API	28
8.1. Estrutura de dados	28
9. DESENVOLVIMENTO	29

9.1. Desenvolvimento do Aplicativo	30
9.1.1. Idealização	30
9.2. Software	30
9.3. Comunicação com o banco de dados	34
10. MONTAGEM DO HARDWARE	36
10.1. Pluviômetro	37
10.2. DHT11	39
10.3. HC-05 Modulo Bluetooth	42
10.4. Ky-003 Sensor Magnético	44
10.4.1 Princípio de funcionamento do pluviômetro com o sensor	46
10.5. Anemometro	47
10.5.1 Equipamento utilizado para montagem	48
10.5.2 Montagem no Arduino	48
11. RESULTADOS E ANÁLISE	49
11.1. Problemas e soluções	49
12. CONCLUSÃO	50
13. REFERÊNCIAS	52

1. INTRODUÇÃO

Nos dias de hoje, busca-se controlar o máximo de fatores possíveis que influenciam de modo direto ou indireto tanto para estudos climáticos e utilização na agricultura. Tratando desse assunto, o monitoramento do clima vem se tornando cada dia mais relevante no campo, tendo em vista que sobre posse de dados meteorológicos é possível otimizar todos os processos de produção, desde a semeadura até a colheita.

As estações meteorológicas são equipamentos utilizados para coleta de dados que posteriormente são utilizados para análise do clima de uma determinada região. Na maioria dos casos, as estações possuem vários instrumentos ou sensores capazes de registrar as possíveis variáveis meteorológicas e climáticas. Os dados meteorológicos coletados por uma estação são úteis em diversos campos da sociedade, podendo destacar o campo de prevenção de desastres e o campo de previsão do tempo, que por sua vez auxiliam em diversas outras áreas, estando entre elas a agricultura.

Nas estações meteorológicas automáticas os equipamentos podem ser classificados em três principais categorias: sensores, data logger e telemetria. Os sensores convertem eventos físicos em sinais elétricos e são responsáveis por quantificar vários parâmetros meteorológicos. Entre sensores mais comuns nas estações meteorológicas estão: pluviômetro, termo higrômetro, anemômetro, direção do vento e pressão atmosférica.

Apesar da grande importância e das diversas aplicações do monitoramento ambiental por meio das estações meteorológicas, essa ferramenta ainda é pouco utilizada no Brasil. De acordo com o Instituto Nacional de Meteorologia (2022), existem um pouco mais que 13 quatrocentas estações meteorológicas no país. Esta informação torna-se bastante preocupante quando se tem em vista que o Brasil é um dos maiores países em extensão territorial e um dos maiores produtores de grãos do mundo. Para reverter essa situação seria necessário estabelecer uma rede de estações meteorológicas mais ampla e que atenda cada região de maneira mais específica, proporcionando desta maneira um maior conhecimento das variáveis climáticas. No entanto em um cenário com um grande número de estações adotando-se o sistema institucional tradicional, ainda haveria dificuldade quanto a instalação e especialmente na manutenção das mesmas, tornando o custo proibitivamente alto.

(SILVA et al., 2015). Muitas vezes, o alto custo com a aquisição destes instrumentos conhecidos no mercado como referência tornam-se um grande impasse na utilização desta tecnologia. Tendo em vista essa situação, o objetivo desse trabalho foi desenvolver uma estação meteorológica automática utilizando sensores de baixo custo presentes no mercado e placas de prototipagem com controladores de linguagem de programação simples, para o monitoramento das mudanças climáticas.

2. TECNOLOGIAS DE DESENVOLVIMENTO

Uma estação meteorológica é um sistema que coleta e processa dados climáticos para análise e previsão do tempo. Esses dispositivos são equipados com sensores que medem grandezas atmosféricas, como temperatura, umidade relativa do ar e pressão entre outras. Ao longo dos anos com o surgimento de diversas tecnologias, surgiram no mercado novos dispositivos trazendo novas funções ou reinventar outras, com esse avanço foi sendo possível cada vez ter mais precisão na coleta de dados.

2.1. *Hardware*

Para o desenvolvimento do *hardware* foi necessário definir quais seriam os componentes a serem utilizados e mais adequados para que o objetivo do mesmo pudesse ser atingido, sem que houvesse um investimento em peças que talvez não fossem usufruídas em sua totalidade ou em peças que não atendessem completamente os requisitos para que o *hardware* conseguisse realizar tudo o que fora idealizado.

Pesquisas foram realizadas inicialmente em busca de um micro controlador, estendendo permitir a utilização de módulos, sensores e comunicação com internet e *Bluetooth*, com isso foram encontradas várias opções de componentes já renomados no mercado, como o *Arduino*, *Raspberry PI*, *PICs*, *ESPs*, etc. Colocando-os em comparação foi possível definir que os componentes *ESP* se encaixariam melhor às necessidades do projeto.

Trazendo para discussão os fatores que possibilitaram que fosse chegada à essa decisão, pode-se citar que para o desenvolvimento com os *PICs*, seria necessário muito esforço relacionado à produção do *hardware*, visto que todos os componentes necessitariam de uma placa de circuito impresso antes mesmo de iniciar o desenvolvimento e testes do protótipo, correndo o risco de não ser

adequado levando ao retrabalho, além disso, não possui nenhuma funcionalidade extra já integrado aos seus circuitos, como o *Wi-Fi* ou o *Bluetooth*.

Relacionado ao *Arduino* podemos dizer que ganha dos *PICs*, que mesmo tendo várias opções de modelos diferentes atrai o desenvolvedor ter seu micro controlador já embutido em um circuito integrado, ter sua própria *IDE* e diversas bibliotecas disponíveis para o desenvolvimento, sendo esses pontos que facilitam a prototipagem dos mesmos, em contrapartida quando o projeto precisa ser apresentado como um produto final para o uso de um público.

O microcontrolador *Arduino Uno* é uma plataforma de prototipagem eletrônica baseada em um micro controlador *ATmega328P*. É um dos modelos mais populares e amplamente utilizados da linha de produtos *Arduino*. O *Arduino Uno* é projetado para ser fácil de usar, acessível e altamente expansível. Ele possui uma série de pinos de entrada e saída digital e analógica, que podem ser usados para conectar sensores, atuadores e outros componentes eletrônicos. Além disso, o *Arduino Uno* pode ser programado usando a linguagem de programação baseada em C/C++, o que o torna uma plataforma flexível e poderosa para a criação de projetos eletrônicos.

Sobre o *Raspberry PI*, ele conta com tudo o que seria necessário para o desenvolvimento do projeto, pois possui comunicação serial facilitada, conexão *Wi-Fi* e *Bluetooth* embutidos, além de ser capaz de realizar processamentos grandes e complexos se comparado com o *Arduino*. O ponto que levou a não utilização do mesmo para a construção do hardware tem reflexo no lado financeiro, visto que o valor de uma unidade do *Raspberry PI* tem um custo mais elevado e mais complexo para aplicação.

Já o *ESP* podemos dizer também que o *Raspberry* se encaixaria no quesito de “sobra” de funcionalidades, ou seja, mesmo tendo muito a oferecer, nem tudo seria utilizado para o intuito do projeto inviabilizando-o, um dos pontos para a tomada decisão de não utilizar é por ter mais conhecimento e conforto na programação do *Arduino ATmega328*.

Tabela 1- Comparação das plataformas

	<i>Processador</i>	<i>Memória RAM</i>	<i>Conexões</i>	<i>Pinos</i>	<i>Clock</i>	<i>Custo (R\$)</i>
<i>PIC</i>	16F690	368 bytes	Serial	Digitais: 18 Analógicas: 11	8MHz	18,69
<i>Arduino UNO</i>	ATmega328	2 KB	Serial	Digitais: 14 Analógicas: 6	16MHz	64,90
<i>Raspberry PI</i>	Broadcom BCM2711	4 GB	Wi-Fi, Bluetooth, Serial	GPIO: 40	1.5GHz	599,90
<i>ESP32</i>	Xtensa	520 KB	Wi-Fi, Bluetooth, Serial	GPIO: 11	240MHz	45,00

Fonte: autores

2.2. Arduino uno ATmega328

O ATmega48P/88P/168P/328P é um micro controlador CMOS de 8 bits de baixo consumo baseado na arquitetura RISC aprimorada AVR. Ao executar instruções poderosas em um único ciclo de clock, o ATmega48P/88P/168P/ 328P atinge taxas de transferência próximas de 1 MIPS por MHz, permitindo ao projetista do sistema otimizar o consumo de energia em relação à velocidade de processamento.

O AVR ATmega48P/88P/168P/328P é compatível com um conjunto completo de ferramentas de desenvolvimento de programas e sistemas, incluindo: Compiladores C, Macro, Assemblers, Depuradores/Simuladores de Programas, Emuladores In-Circuit kits de avaliação.

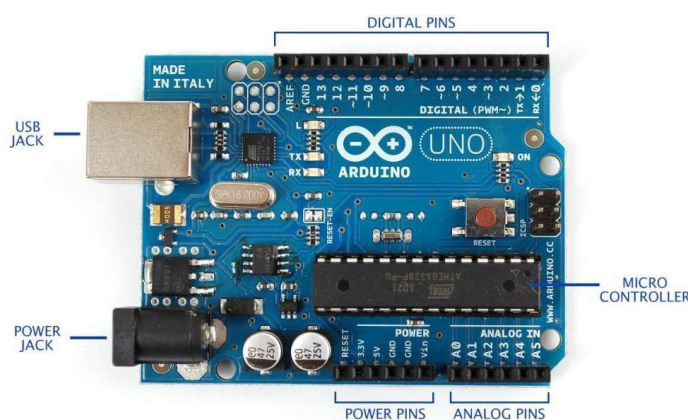


Figura 1 - Arduino Uno ATmega328p

Fonte: techsul.com.

O núcleo do AVR combina um rico conjunto de instruções com 32 registros de trabalho de uso geral. Todos os 32 registros estão diretamente conectados à Unidade Lógica Aritmética (ALU), permitindo que dois registros independentes sejam acessados em uma única instrução executada em um ciclo de clock.

A arquitetura é mais eficiente em termos de código e atinge rendimentos até dez vezes mais rápidos do que o micro controlador CISC convencionais. O ATmega48P/88P/168P/328P fornece os seguintes recursos: 4K/8K/16K/32K bytes de Flash Programável no Sistema com recursos de leitura durante gravação, EEPROM de 256/512/512/1K bytes, 512/1K/1K /2K bytes SRAM, 23 linhas de E/S de uso geral, 32 registros de trabalho de uso geral, três temporizadores/contadores flexíveis com modos de comparação, interrupções internas e externas, uma USART serial programável, uma interface serial de 2 fios orientada a bytes, um SPI porta serial, um ADC de 10 bits de 6 canais (8 canais em pacotes TQFP e QFN/MLF), um temporizador Watchdog programável com oscilador interno e cinco modos de economia de energia selecionáveis por software. O modo inativo para a CPU enquanto permite que SRAM, temporizadores/contadores, USART, interface serial de 2 fios, porta SPI e sistema de interrupção continuem funcionando.

O chip ATMEGA328p-pu é um micro controlador de alto desempenho que oferece grande flexibilidade para desenvolvimento de projetos eletrônicos. Com sua arquitetura AVR de 8 bits, ele é capaz de executar operações complexas e controlar vários dispositivos periféricos. Este chip é amplamente utilizado em projetos eletrônicos devido à sua capacidade de processamento, baixo consumo de energia e grande quantidade de memória.

O dispositivo é fabricado com tecnologia de memória não volátil de alta

densidade da Atmel. O Flash ISP no chip permite que a memória do programa seja reprogramada no sistema por meio de uma interface serial SPI, por um programador de memória não volátil convencional ou por um programa de inicialização no chip executado no núcleo AVR.

2.3. Descrição dos Pinos

VCC: Tensão de alimentação digital.

GND: Terra.

Porta B (PB7:0): A porta B é uma porta de E/S bidirecional de 8 bits com resistores pull-up internos (selecionados para cada bit). Os buffers de saída da Porta B possuem características de unidade simétricas com alta capacidade de dissipador e fonte. Como entradas, os pinos da Porta B que são puxados externamente para baixo fornecerão corrente se os resistores pull-up estiverem ativados. Os pinos da Porta B recebem três estados quando uma condição de reinicialização se torna ativa, mesmo se o relógio não estiver funcionando. Dependendo das configurações do fusível de seleção do relógio, o PB6 pode ser usado como entrada para o amplificador oscilador inversor e entrada para o circuito operacional do relógio interno. Dependendo das configurações do fusível de seleção do clock, o PB7 pode ser usado como saída do amplificador oscilador inversor. Se o oscilador RC calibrado interno for usado como fonte de clock do chip, PB7.6 será usado como entrada TOSC2.1 para o temporizador assíncrono 2 se o bit AS2 em ASSR estiver definido.

Porta C (PC5:0): A porta C é uma porta de E/S bidirecional de 7 bits com resistores pull-up internos (selecionados para cada bit). Os buffers de saída PC5.0 possuem características de unidade simétricas com alta capacidade de dissipador e fonte. Como entradas, os pinos da Porta C que são puxados externamente para baixo fornecerão corrente se os resistores pull-up estiverem ativados. Os pinos da Porta C recebem três estados quando uma condição de reinicialização se torna ativa, mesmo se o relógio não estiver funcionando.

PC6/RESET: Se o fusível RSTDISBL estiver programado, PC6 será usado como pino de E/S. Observe que as características elétricas do PC6 diferem das dos demais pinos da Porta C. Se o fusível RSTDISBL não estiver programado, PC6 será

usado como entrada de Reset. Um nível baixo neste pino por mais tempo que o comprimento mínimo do pulso gerará um Reset, mesmo se o relógio não estiver funcionando

Porta D (PD7:0) A porta D é uma porta de E/S bidirecional de 8 bits com resistores pull-up internos (selecionados para cada bit). Os buffers de saída da Porta D possuem características de unidade simétricas com alta capacidade de dissipador e fonte. Como entradas, os pinos da Porta D que são puxados externamente para baixo fornecerão corrente se os resistores pull-up estiverem ativados. Os pinos da Porta D recebem três estados quando uma condição de reinicialização se torna ativa, mesmo se o relógio não estiver funcionando.

AVCC: é o pino de tensão de alimentação para o conversor A/D, PC3:0 e AD

AREF: é o pino de referência analógico para o conversor A/D.C7:6. Deve ser conectado externamente a VCC, mesmo que o ADC não seja usado.

ADC7:6 (somente pacote TQFP e QFN/MLF): No pacote TQFP e QFN/MLF, ADC7:6 servem como entradas analógicas para o conversor A/D. Esses pinos são alimentados pela fonte analógica e servem como canais ADC de 10 bits

2.4. Comparação entre ATmega48P, ATmega88P, ATmega168P e ATmega328P

O ATmega48P, ATmega88P, ATmega168P e ATmega328P diferem apenas em tamanhos de memória, suporte ao carregador de inicialização e tamanhos de vetores de interrupção, na tabela a seguir resume os diferentes tamanhos de memória e vetores de interrupção para os três dispositivos.

Tabela 2 - Tamanhos de memória.

Dispositivo	Flash	EEPROM	RAM	Interrupção
ATmega48P	4K Bytes	256 Bytes	512 Bytes	1 palavra/vetor de instrução
ATmega88P	8K Bytes	512 Bytes	1K Bytes	1 palavra/vetor de instrução
ATmega168P	16K Bytes	512 Bytes	1K Bytes	2 palavras/vetor de instrução
ATmega328P	32K Bytes	1K Bytes	2K Bytes	2 palavras/vetor de instruções

Fonte: datasheet

Esses dispositivos suportam um mecanismo real de autoprogramação de leitura durante gravação. Há uma sessão separada do carregador de inicialização e a instrução SPM só pode ser executada a partir daí. No ATmega48P, não há suporte para leitura durante gravação e nenhuma sessão separada do carregador de inicialização. A instrução SPM pode ser executada em todo o Flash.

2.5. Componentes

Além do *Arduino Uno Atmega328p* que vai ser utilizado como peça central para desenvolvimento do projeto, muitos outros periféricos foram necessários para que o hardware fosse montado por completo, onde podemos citar como mais importantes. Cada um desses componentes tem suas peculiaridades e funcionalidades, que serão explicadas a seguir.

3. TRANSDUTOR DHT11

O sensor de temperatura e umidade DHT11 possui um sensor de temperatura e umidade complexo com uma saída de sinal digital calibrada. Usando o exclusivo sistema de aquisição de sinal digital técnica e tecnologia de detecção de temperatura e umidade, garante alta confiabilidade e excelente estabilidade a longo prazo. Este sensor inclui uma medição de umidade do tipo resistiva com medição de temperatura NTC e se conecta a um micro controlador de 8 bits de alto desempenho, oferecendo excelente qualidade, resposta rápida, anti- interferência, capacidade e custo-benefício.

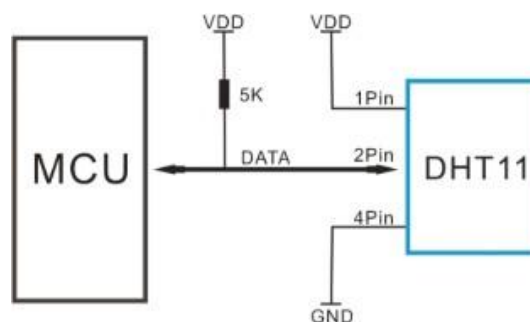
Figura 2 - Transdutor DHT11



Fonte: freanzing

Cada elemento DHT11 é rigorosamente calibrado em laboratório e é extremamente preciso, os coeficientes de calibração são armazenados como programas na memória OTP, que são usados pelo processo de detecção de sinal interno do sensor. A interface serial de fio único torna a integração do sistema rápida e fácil. Seu tamanho pequeno, baixo consumo de energia e até 20 transmissões de sinal do medidor, tornando-o a melhor escolha para diversas aplicações, incluindo aquelas os mais exigentes. O componente é um pacote de pinos de linha única de 4 pinos. É conveniente connect e pacotes especiais podem ser fornecidos de acordo com a solicitação dos usuários.

Figura 3 - MCU para DHT11



Fonte: Randon Nerd Tutorials

3.1. Fonte e Pin

A fonte de alimentação do DHT11 é de 3-5,5 VCC. Quando a energia for fornecida ao sensor, não envie nenhuma instrução para o sensor dentro de um segundo para passar o status instável. Um capacitor com valor de 100nF pode ser adicionado entre VDD e GND para filtragem de energia.

3.2. Comunicação

Quando o MCU envia um sinal de início, o DHT11 muda do modo de baixo consumo de energia para o modo de execução, aguardando que o MCU complete o sinal de partida. Depois de concluído, o DHT11 envia um sinal de resposta de dados de 40 bits que inclui informações de umidade relativa e temperatura para MCU. Os usuários podem optar por coletar alguns dados. Sem o sinal de início do MCU, DHT11 não dará o sinal de resposta ao MCU. Assim que os

dados forem coletados, o DHT11 mudará para o modo de baixo consumo de energia até receber novamente um sinal de início do MCU.

4. MODULO BLUETOOTH HC-05

O módulo Bluetooth HC-05 é um dispositivo de comunicação sem fio que utiliza a Tecnologia Bluetooth SPP (Serial Port Protocol), para transmitir dados de um dispositivo para outro, com baixo custo, e de baixa potência para sistemas eletrônicos.

Ele é projetado para suportar comunicações com Protocolo Serial (RX/TX) para envio e recebimento de dados a pequenas distâncias, e muito utilizado em projetos com micro controladores como; Arduino, PIC, Raspberry, ESP8266, ESP32, etc.

Possui uma memória interna para armazenar o firmware e outras informações importantes, como configurações e endereços MAC. A capacidade da memória interna pode variar dependendo da versão e do fabricante do módulo.

Normalmente, a memória interna do módulo HC-05 varia de 8 KB a 16 KB. Isso é suficiente para armazenar o firmware do módulo, que inclui o protocolo Bluetooth e outras funções essenciais para permitir a comunicação sem fio entre o módulo e outros dispositivos.

Dependendo do fabricante do módulo HC-05, o chip de memória flash ou EEPROM, podem ser de diversas marcas, mais, as mais conhecidas no mercado geralmente são das empresas MX "Macronix International Co., Ltd." ou da empresa SST "Silicon Storage Technology, Inc.", o a SPANSION atualmente faz parte da empresa "Cypress Semiconductor". Esses chips de memória flash são frequentemente usados em módulos HC-05 para aumentar a capacidade de armazenamento de dados e permitir o armazenamento de dados ou informações adicionais, configurações personalizadas ou logs de atividade e outros dados importantes.

4.1.1. Frequência e Alcance

O módulo Bluetooth HC-05 é projetado para funcionar na faixa de frequência de 2,4 GHz e utiliza o padrão Bluetooth v2.0 + EDR. Ele suporta um

alcance máximo de comunicação de 10 metros em ambientes internos e 30 metros em ambientes externos.

4.2. Funcionamento do HC-05

Pode ser usado em dois modos de operação: Modo de Comando e Modo de Dados.

4.2.1. Modo de comando

No Modo de Comando, podemos nos comunicar com o módulo Bluetooth por meio de Comandos AT para definir várias configurações e parâmetros do Módulo. Isso inclui as informações do firmware, alteração da taxa de transmissão, alteração do nome do módulo, etc.

Também podemos usá-lo para definir HC-05 como mestre ou escravo. Para selecionar qualquer um dos modos, precisamos ativar o Modo de Comando e enviar os Comandos AT corretos. A taxa de transmissão é de 38400bps no modo de comando.

4.2.2. Modo de dados

Neste modo, o módulo é usado para comunicação com outros dispositivos Bluetooth, ou seja, a transferência de dados ocorre neste modo. Troca de dados entre dispositivos. A taxa de transmissão é de 9600bps no modo de dados.

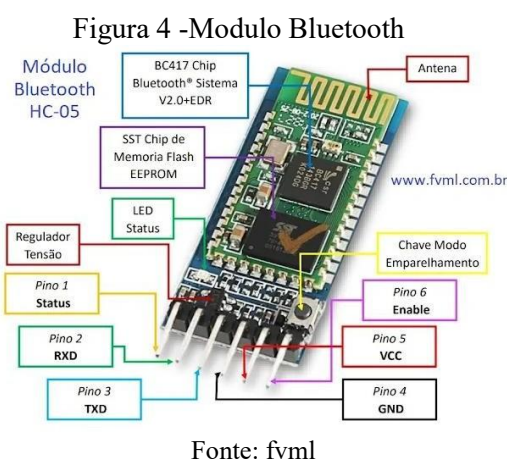


Tabela 3 - Pinagem Hc-05

Pinagem Hc-05

Pino	Definição
Enable	Habilita comando AT em nível alto

VCC	Alimentação positiva (3.6V – 6V)
GND	Alimentação negativa
TXD	Transmissor de dados
RXD	Recebedor de dados
State	Auxiliar para LED indicador

Fonte: autores

4.2.3. Configurações padrão do HC-05

- Nome padrão do Bluetooth: "HC-05"
- Senha padrão: 1234 ou 0000
- Comunicação padrão: Modo escravo
- Modo padrão: Modo de dados
- Taxa de transmissão: 9600 bps
- Taxa de transmissão do modo de comando: 38400 bps
- Configurações do modo de dados: 8 bits de dados, 1 bit de parada, sem paridade

5. ANEMÔMETRO

Anemômetro é o dispositivo meteorológico usado para medir a velocidade e intensidade dos ventos, ou de outros fluídos em movimento. São utilizados em laboratórios de hidráulica, de aerodinâmica ou em algum fluido como os gases existentes nas estrelas e planetas. Atualmente, os instrumentos são altamente precisos e fornecem dados de variadas formas.

Os anemômetros são utilizados em estações meteorológicas, tanto em lugares com temperaturas baixas como em locais mais quentes. A velocidade do vento auxilia na identificação da mudança dos padrões climáticos, tão como pode prever uma tempestade que se aproxima, o que é de suma importância para pilotos de aeronaves, engenheiros e climatologistas.

Um dispositivo muito utilizado por engenheiros e físicos são os anemômetros a laser. Esse tipo é utilizado nos experimentos de velocidade, calculando a rapidez do vento em torno dos carros e aviões, por exemplo, pela medida da taxa e direção da mudança no sentido de um objeto. Isso ajuda engenheiros a reduzir a resistência que o vento imprime contra um veículo em movimento.

Figura 5- Anemômetro



Fonte: Arduino fórum

5.1. Tipos de anemômetro

5.1.1. Anemômetros de copo

Este é o dispositivo de medição mais simples dos anemômetros. Ele é composto de um polo vertical com quatro braços horizontais unidos ao topo. Os copos estão ligados às extremidades dos braços existentes e o vento faz com que eles girem em volta do polo central. Como ele é mais propenso ao atrito, acaba tendo menor precisão que as versões avançadas.

5.1.2. Anemômetro de moinhos de vento

Este é bem semelhante a um moinho de vento, por isso o nome, uma vez que deve ser paralelo à direção do vento para funcionar corretamente. Um cata-vento é unido à cauda do próprio anemômetro, sendo impulsionado até o momento que a hélice esteja vertente ao vento. O vento irá girar a hélice, produzindo um mecanismo que calcula a velocidade do vento.

5.1.3. Anemômetro termoeletrico

Estes instrumentos de medição são muito delicados, mas precisos. Este dispositivo usa um fio fino que é aquecido a uma temperatura maior que a do ambiente. O vento esfria o fio e a eletrônica, localizada dentro do corpo do instrumento, calcula a velocidade do vento com sustentação na resistência elétrica do fio.

5.1.4. Anemômetros ultrassônicos

Estes são responsáveis por determinar a velocidade do vento, ao conduzir

ondas de som entre um par de transdutores, e indicar como é atingida a velocidade. Os anemômetros ultrassônicos medem a velocidade ou a direção do vento e são muito utilizados em aviões, navios e turbinas eólicas. Os doppler laser e os ultrassônicos são muito precisos. Os anemômetros Doppler Laser fazem a medição da velocidade do vento ao constatar a quantidade de luz do raio laser que reflete nas partículas de ar em atividade.

5.2. Quem precisa de um anemômetro?

Várias indústrias e profissionais precisam de anemômetros para medir a velocidade e, em alguns casos, a direção do vento. O uso de anemômetros é crucial em diversas áreas, incluindo:

Meteorologia: anemômetros são fundamentais para estações meteorológicas na coleta de dados sobre o vento, ajudando na previsão do tempo e na monitoração de condições climáticas extremas.

Aviação: Pilotos e controladores de tráfego aéreo usam anemômetros para avaliar condições de vento, essenciais para a decolagem, pouso e segurança do voo.

Energia Eólica: Na indústria de energia eólica, anemômetros são usados para avaliar o potencial de geração de energia de um local e para otimizar o desempenho de turbinas eólicas.

Construção e Engenharia Civil: Os engenheiros usam anemômetros para garantir que as condições de vento sejam seguras para a construção de grandes estruturas, como arranha-céus e pontes, e para realizar análises de carga de vento.

Navegação: Na navegação marítima e em esportes aquáticos como vela, os anemômetros ajudam a avaliar as condições de vento para a navegação segura e eficiente.

Pesquisa Científica e Ambiental: Anemômetros são usados em pesquisas ambientais para estudar padrões de vento e em pesquisas climáticas, especialmente em estudos relacionados às mudanças climáticas.

Agricultura: Agricultores utilizam anemômetros para monitorar condições de vento, que podem afetar a pulverização de pesticidas e a saúde das plantações.

Esportes ao Ar Livre: Em esportes como parapente, asa-delta e até mesmo golfe, os participantes usam anemômetros para avaliar as condições do vento para melhorar a segurança e o desempenho.

Segurança Industrial e Controle de Poluição: Anemômetros são usados para

monitorar ventos em ambientes industriais, especialmente em locais onde há risco de emissão de gases nocivos ou explosivos.

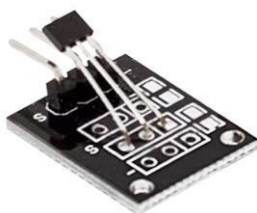
O anemômetro é um instrumento extremamente útil para determinar a velocidade do vento. Ele pode ser usado para monitorar as condições climáticas em tempo real, o que permite tomar medidas preventivas contra intempéries.

6. SENSOR HALL KY-003

O Sensor Hall KY-003 é um dispositivo compacto e altamente sensível que opera com base no princípio do Efeito Hall. Projetado para detectar campos magnéticos, possui uma ampla faixa de tensão de operação de 4,5 V a 24 V, tornando-o compatível com uma variedade de plataformas de micro controladores. Com dimensões de aproximadamente 18,5 mm x 15 mm x 5 mm, é fácil de integrar em projetos eletrônicos.

Sua saída é digital, simplificando sua utilização e interpretação pelo micro controlador. O Sensor Hall KY-003 é ideal para aplicações que exigem detecção de campo magnético, como controle de posição, detecção de metais, entre outros.

Figura 6 - Ky-003



Fonte: Eletrogate

6.1. Conceito de magnetismo

Magnetismo se refere a todo fenômeno, ou conjunto de fenômenos, de atração ou repulsão de corpos que se caracteriza como uma força dipolar, ou seja, que possui dois polos. No caso de corpos magnéticos, como ímãs, a força magnética é originada no polo Norte, em direção ao polo Sul. Essa força resulta em uma atração de corpos caso os polos sejam opostos, ou seja, o polo Norte irá atrair o polo Sul, e vice-versa. Consequentemente, para polos iguais, os corpos tendem a se

repelir.

Alguns materiais metálicos, como o próprio ferro, não são capazes de gerar uma força magnética, porém são atraídos por elas. No entanto, existem já alguns métodos para imantar esses materiais, de modo que os mesmos passem a ter uma força magnética para atração ou repulsão. Além disso, graças à eletricidade, é possível criar um campo magnético através da passagem de corrente elétrica. Este fenômeno é denominado eletromagnetismo

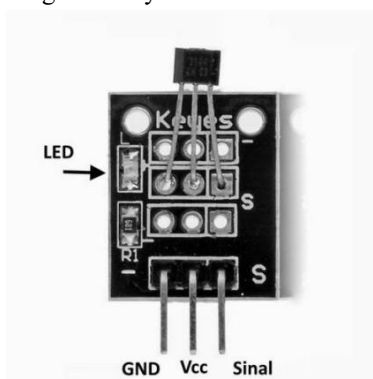
6.2. Como funciona?

O Sensor Hall KY-003 é baseado no princípio do Efeito Hall. Quando um campo magnético está presente próximo ao sensor, ele gera uma diferença de potencial (ou tensão) no chip do sensor. Essa tensão é proporcional à intensidade do campo magnético. O sensor detecta essa tensão e a converte em um sinal elétrico que pode ser interpretado pelo micro controlador, como o Arduino.

Assim, o sensor pode ser utilizado para detectar a presença e a intensidade de campos magnéticos em sua proximidade, tornando-o útil em uma variedade de aplicações, como controle de posição, detecção de metais, entre outros.

6.3. Especificações

Figura 7- Ky-003 Datasheet



Fonte: Arduinoecia

- Modelo: KY-003. Tensão de operação: 4,5V a 24V.
- Tipo de saída: Coletor aberto.
- Chip principal: A3144E.
- Indicador LED: Sim, para indicação visual de detecção.
- Proteção contra inversão de polaridade: Sim.

- Compatibilidade: Arduino, Raspberry Pi, e outros microcontroladores.
- Dimensões do módulo: Aproximadamente 18,5mm x 15mm x 10mm.

6.4. Aplicações

Aplicações de Segurança e Controle de Acesso: O sensor KY-003 pode detectar a presença de um campo magnético próximo, tornando-o útil em sistemas de segurança para detectar a aproximação de um objeto magnético, como uma chave ou um cartão magnético. Em sistemas de controle de acesso, ele pode ser utilizado para identificar a presença de um cartão ou chave magnética antes de liberar o acesso.

Tacômetros e Contadores de Rotação: O sensor KY-003 pode ser integrado a sistemas de medição de velocidade, como tacômetros e contadores de rotação, para detectar a presença de um ímã em movimento. Ao monitorar a frequência com que o campo magnético é detectado, é possível calcular a velocidade de rotação ou de movimento do objeto ao qual o ímã está fixado.

Sensorialmente em Projetos de Robótica e Automação: Em projetos de robótica, o sensor KY-003 pode fornecer informações sobre a presença ou ausência de objetos magnéticos, ajudando a controlar movimentos ou ações dos robôs. Ele pode ser usado para detectar a posição de componentes móveis ou para criar sistemas de Feedback sem contato.

Essas aplicações destacam a versatilidade do sensor KY-003, tornando-o um componente útil em diversos contextos onde a detecção de campos magnéticos e a criação de sistemas sem contato são necessários, pode se tornar um sensor bem utilizado em diversos projetos e aplicações.

7. BANCO DE DADOS

Um banco de dados, como o próprio nome sugere, é um local onde se pode armazenar uma grande quantidade de dados. Com base nele, o desenvolvedor consegue utilizá-lo para diversas aplicações diferentes, realizando buscas, inserções e alterações das informações necessárias, que ficam disponíveis para acesso a qualquer momento que a aplicação solicitar.

Para o desenvolvimento deste projeto de estação meteorológica, a escolha do MySQL se destacou como a opção mais adequada. Além de ser uma ferramenta

gratuita, o MySQL fornece uma vasta gama de informações e documentação de apoio, facilitando o processo de aprendizado e implementação.

O MySQL Workbench é uma ferramenta gráfica poderosa para trabalhar com servidores e bancos de dados MySQL, oferecendo suporte completo ao MySQL Server 8.0. Suas funcionalidades abrangem cinco tópicos principais:

SQL Development: Permite que você crie e gerencie conexões com servidores de banco de dados. O MySQL Workbench oferece um SQL Editor integrado, que facilita a execução de consultas SQL e a manipulação dos dados de forma eficiente.

Modelagem de Dados (Design): Possibilita a criação de modelos gráficos do esquema do banco de dados, facilitando a engenharia reversa e direta entre o esquema e um banco de dados ativo. O abrangente Editor de Tabelas permite a edição fácil de tabelas, colunas, índices e outros aspectos do banco.

Administração do servidor: Com o MySQL Workbench, você pode administrar instâncias do servidor MySQL, gerenciar usuários, realizar backups e recuperação de dados, além de monitorar o desempenho do servidor, garantindo a integridade e a segurança do banco.

Migração de Dados: Facilita a migração de dados de outros sistemas de gerenciamento de banco de dados para o MySQL, o que é útil caso você precise integrar dados existentes ao seu projeto.

Suporte MySQL Enterprise: Oferece suporte para produtos empresariais, como MySQL Enterprise Backup e MySQL Firewall, aumentando a segurança e a confiabilidade do sistema.

Com a adoção do MySQL, você se beneficia de um sistema otimizado para armazenar grandes volumes de dados, o que é ideal para uma estação meteorológica que registra informações ao longo do tempo. A flexibilidade do MySQL em integrar-se com diversas linguagens de programação também permite a fácil implementação em projetos que utilizam Arduino e Visual Studio.

Além disso, o MySQL proporciona segurança robusta, com opções de backup e recuperação que garantem a proteção dos dados coletados. Sua escalabilidade permite que o sistema cresça conforme a necessidade do projeto, sem comprometer o desempenho.

Em resumo, o MySQL se apresenta como uma solução prática e poderosa para o gerenciamento de dados em projetos de estação meteorológica. Sua adoção

pode otimizar o processo de coleta e análise de dados, contribuindo significativamente para a eficácia do projeto em desenvolvimento.

8. API

A previsão do tempo é uma informação crucial para uma ampla gama de aplicações, desde o planejamento de atividades ao ar livre até a logística de transporte e a gestão de desastres naturais. Com o avanço da tecnologia e o crescimento da integração de dados em tempo real, as APIs (Application Programming Interfaces) de previsão do tempo desempenham um papel fundamental em fornecer informações meteorológicas precisas e atualizadas para sistemas e aplicativos diversos.

A API de Previsão do Tempo é uma interface que permite a integração de dados meteorológicos em sistemas e aplicativos, possibilitando a obtenção de previsões detalhadas, condições atuais e históricos climáticos. Essas APIs são projetadas para oferecer acesso a uma variedade de informações meteorológicas, incluindo temperatura, umidade, velocidade do vento, precipitação e outras condições atmosféricas.

Neste projeto, exploraremos a API de Previsão do Tempo, detalhando suas principais funcionalidades, a estrutura dos dados fornecidos, e as melhores práticas para sua implementação e uso. A análise incluirá a descrição das principais características da API, os métodos disponíveis para consulta, e como essas informações podem ser integradas em aplicações para oferecer previsões meteorológicas precisas e em tempo real. Além disso, discutiremos as considerações para a escolha da API apropriada com base nas necessidades específicas do projeto e os desafios associados à utilização dessas interfaces para garantir a confiabilidade e a eficiência dos dados fornecidos.

8.1. Estrutura de dados

Na API de Previsão do Tempo, os dados referentes à consulta são organizados e retornados no parâmetro `results`. A estrutura de dados dentro deste parâmetro é projetada para fornecer uma representação clara e detalhada das condições meteorológicas solicitadas.

Compreender a estrutura dos dados é essencial para interpretar corretamente

as informações recebidas e integrá-las eficazmente em aplicações e sistemas.

Alguns parâmetros para consulta:

- temp - temperatura atual em °C
- date - data da consulta, em fuso horário do local
- time - hora da consulta, em fuso horário do local
- condition_code - código da condição de tempo atual
- description - descrição da condição de tempo atual no idioma escolhido
- currently - retorna se está de dia ou de noite no idioma escolhido
- city - nome da cidade
- humidity - umidade atual em percentual
- cloudiness - nebulosidade em percentual, de 0 a 100
- rain - volume de chuva em mm na última hora
- wind_speedy - velocidade do vento em km/h
- wind_direction - direção do vento em grau
- wind_cardinal - direção do vento
- sunrise - nascer do Sol em horário local da cidade
- sunset - pôr do Sol em horário local da cidade
- moon_phase - fase da Lua

Para obter esses dados, é necessário utilizar a URL, que pode ser obtida no site oficial da Weather API. A inserção dessa URL deve ser realizada conforme mostrado abaixo.

URL:

<http://api.weatherapi.com/v1/forecast.xml?key=46d22bedb4824b5485a172113201209&q>

Ao considerar a implementação da API, é crucial avaliar suas características, métodos de consulta e a precisão dos dados fornecidos. A escolha adequada da API e o entendimento de seus dados e funcionalidades garantirão a eficiência e a confiabilidade das previsões meteorológicas integradas nas soluções desenvolvidas.

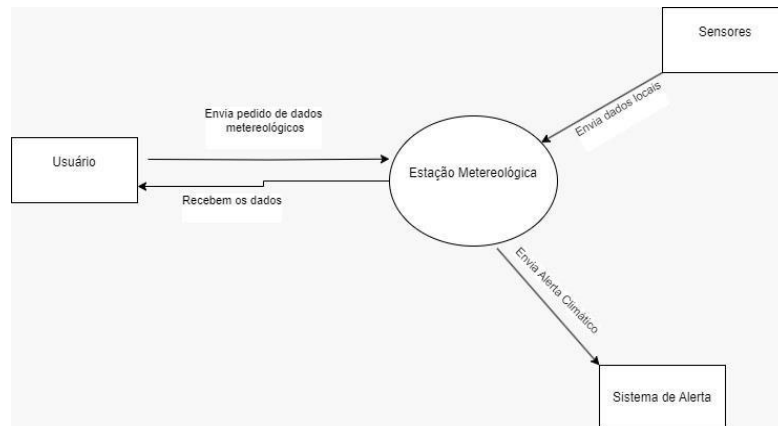
9. DESENVOLVIMENTO

O projeto final consiste em uma Estação Meteorológica que será utilizado vários sensores Api Weather com o Arduino, assim podendo obter dados referente a

funcionalidade dos sensores para a leitura e exibição dos dados, será criada uma interface utilizando Windows Forms na linguagem C#.

Para garantir o entendimento melhor do projeto foi elaborado um DFT ilustrando o fluxo de dados e seu comportamento ao longo da execução. Como podemos ver na Figura 6, um esquema interligando todas as partes do projeto.

Figura 8 - DFT de contexto



Fonte: Autores.

9.1. Desenvolvimento do Aplicativo

9.1.1. Idealização

Antes do desenvolvimento do aplicativo ser iniciado alguns mockups foram desenvolvidos para servir como guia inicial e definir as funções necessárias para a operação e integração correta do software com o Hardware.

9.2. Software

Na realização do projeto, foi feito um software implementado no IDE do Arduino e o Visual Studio. A IDE foi utilizada no desenvolvimento do código fonte como um todo, e o Visual S foi utilizado na interface do monitoramento dos dados e na comunicação via bluetooth com os sensores da que compõe a estação meteorológica.



Seja Bem Vindo!

Nome:

Senha:

☐ Mostrar a senha


ENTRAR

[Não tem uma conta?](#) **Cadastrar-se**

Figura 9 - Tela de login

Fonte: autores

Figura 10 - Tela de Registrar



Email:

Nome:

Senha:

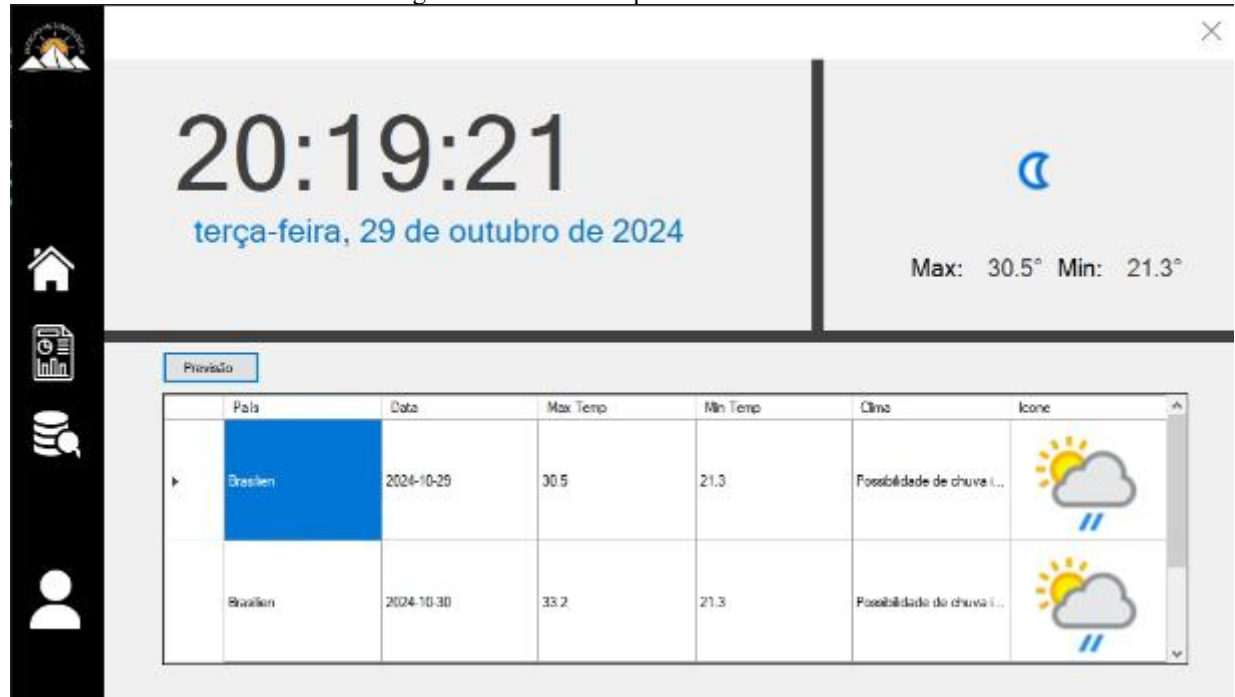
☐ Mostrar Senha

REGISTRAR

[Já tem uma conta?](#) **Fazer login**

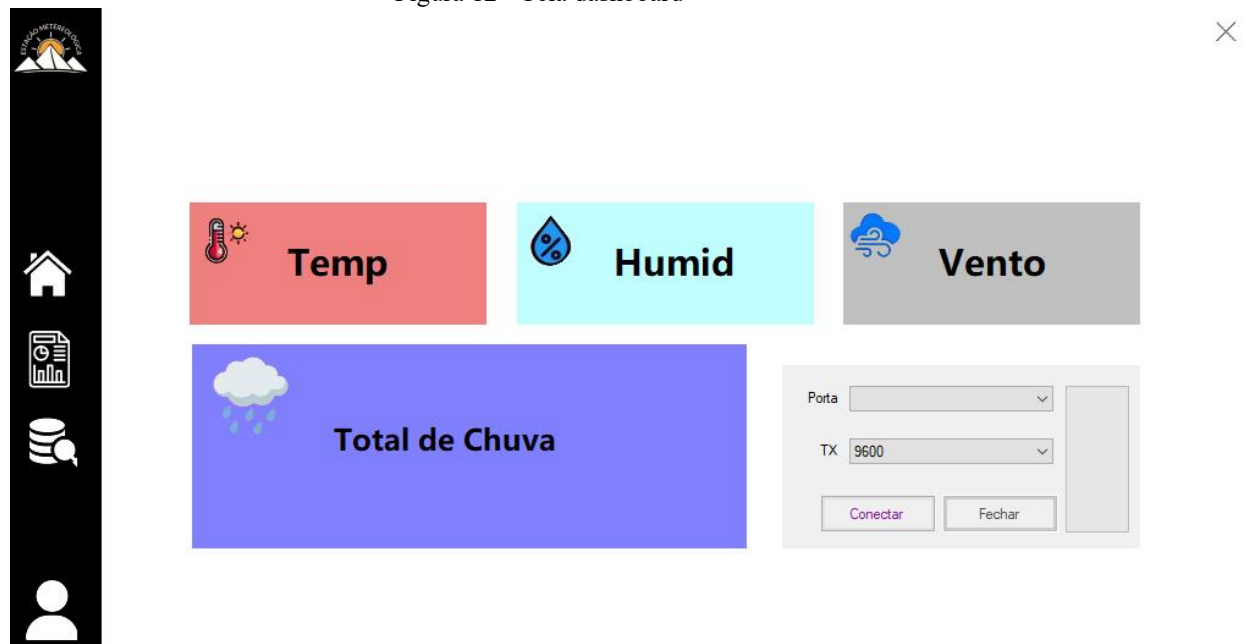
Fonte: autores

Figura 11 - Tela Principal



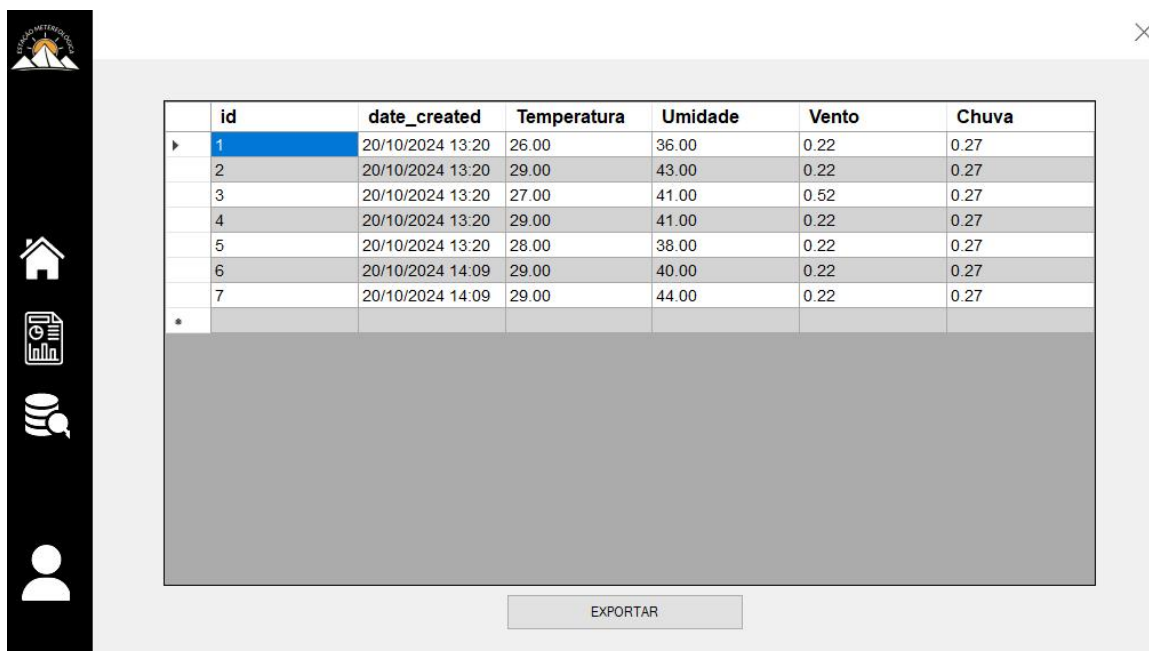
Fonte: autores

Figura 12 - Tela dashboard



Fonte: autores

Figura 13 - Tela de consulta



id	date_created	Temperatura	Umidade	Vento	Chuva
1	20/10/2024 13:20	26.00	36.00	0.22	0.27
2	20/10/2024 13:20	29.00	43.00	0.22	0.27
3	20/10/2024 13:20	27.00	41.00	0.52	0.27
4	20/10/2024 13:20	29.00	41.00	0.22	0.27
5	20/10/2024 13:20	28.00	38.00	0.22	0.27
6	20/10/2024 14:09	29.00	40.00	0.22	0.27
7	20/10/2024 14:09	29.00	44.00	0.22	0.27

EXPORTAR

Fonte: autores

As (Figuras 9 e 10) descrevem funcionalidades cruciais para a operação do sistema.

Elas são responsáveis por gerenciar o acesso à conta do usuário e pela coleta e armazenamento dos dados obtidos dos sensores no banco de dados durante todo o período de operação. Este recurso é fundamental para garantir a integridade e a disponibilidade das informações, evitando a perda de dados valiosos que podem comprometer a análise posterior.

A tela principal, ilustrada na (Figura 11), serve como um painel inicial do software. Nela, os usuários poderão visualizar informações relevantes, incluindo a previsão do tempo para os próximos dias, acessada através de uma API de clima. Essa funcionalidade não apenas fornece dados meteorológicos atualizados, mas também permite que os usuários planejem suas atividades com base nas condições climáticas previstas.

A interface dedicada à visualização dos dados dos sensores, representada na (Figura 12), foi projetada com o objetivo de oferecer uma experiência de usuário intuitiva e amigável. Nessa tela, os usuários poderão analisar os dados coletados pelo Arduino de forma clara e eficiente. Além disso, será possível gerar relatórios

detalhados que consolidam as informações dos sensores, facilitando a interpretação e a tomada de decisões informadas com base nos dados coletados.

A (Figura 13) apresenta a interface de consulta dos dados armazenados dos sensores no banco de dados MySQL. Essa tela permite ao usuário visualizar as informações coletadas em tempo real ou a partir de registros históricos, proporcionando uma visão detalhada dos dados capturados pelos sensores. Além disso, há a opção de exportar essas informações para o Excel através de um botão dedicado, facilitando a análise e o compartilhamento dos dados em outros formatos.

Essas funcionalidades foram desenvolvidas com o objetivo de proporcionar uma experiência de uso rica e eficiente, permitindo que os usuários acessem e analisem os dados meteorológicos de maneira prática e dinâmica. A integração entre o Arduino e o Visual Studio, aliada ao uso de sensores e tecnologias de conectividade, resulta em uma estação meteorológica robusta, capaz de atender às necessidades de monitoramento e análise climática.

9.3. Comunicação com o banco de dados

Para que possa ser realizado o armazenamento dos dados obtidos pelos sensores, será feito o envio para o banco de dados, neste caso, o MySQL. Após esse processo o usuário poderá fazer consultas e gerar relatórios usando as integrações feitas no Visual Studio.

Para poder trabalhar com MySQL temos que incluir uma referência a bibliotecas **MySQL.Data** no projeto usando o Nuget.

Para isso clique no menu **Tools** e a seguir selecione **Library Package Manager -> Manage Nuget Packages for Solution...**

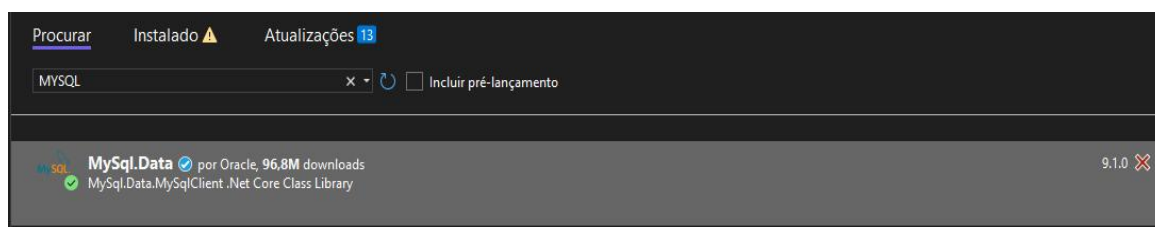


Figura 14 - Biblioteca MySQL

Fonte: autores

A classe DB implementa um padrão básico para gerenciar conexões de

banco de dados, garantindo que as conexões sejam abertas e fechadas corretamente. Com isso poderá fazer implementação de tratamento de exceções para lidar com possíveis falhas de conexão e a adoção de práticas de segurança, como o uso de senhas fortes e a configuração de permissões adequadas no banco de dados.

No menu do Project Solution clique em Add Class e informe o nome. A seguir terá que incluir o código abaixo para fazer a conexão com o banco de dados, assim podendo usar os métodos para armazenar os dados.

Figura 15 - Código conexão BD

```
class DB
{
    MySqlConnection connection = new MySqlConnection("server=localhost;port=3308;username=root;password=;database=user");

    public void openConnection()
    {
        if (connection.State == System.Data.ConnectionState.Closed)
        {
            connection.Open();
        }
    }

    public void closeConnection()
    {
        if (connection.State == System.Data.ConnectionState.Open)
        {
            connection.Close();
        }
    }

    public MySqlConnection getConnection()
    {
        return connection;
    }
}
```

Fonte: autores

A classe DB é responsável por gerenciar a conexão com um banco de dados MySQL. Ela contém um atributo que representa a conexão, que é configurada com detalhes como o servidor, a porta, o nome de usuário, a senha e o banco de dados a ser utilizado.

O método `openConnection ()` abre a conexão com o banco de dados, verificando primeiro se a conexão está fechada. Se estiver, a conexão é estabelecida. Da mesma forma, o método `closeConnection ()` fecha a conexão, mas apenas se ela estiver aberta, garantindo que os recursos sejam liberados corretamente.

Além disso, a classe oferece o método `getConnection ()`, que permite o acesso à instância da conexão para que outras partes do código possam executar

comandos SQL.

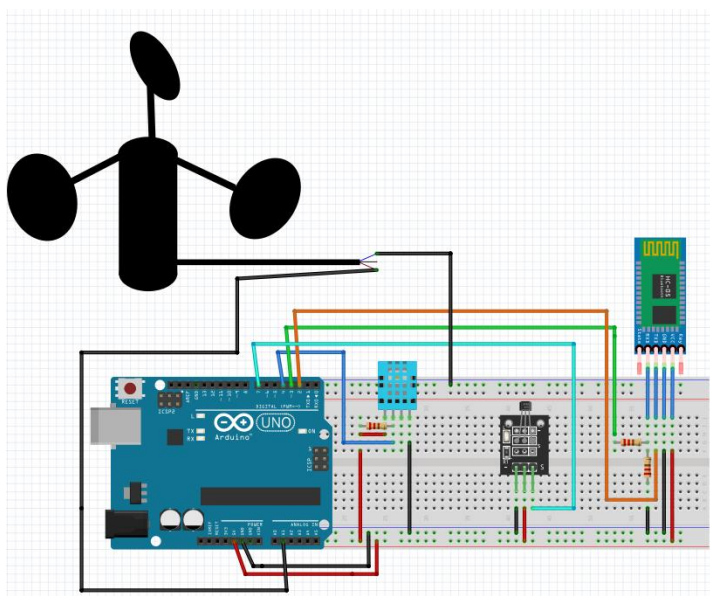
Em resumo, essa classe implementa um padrão simples e eficiente para o gerenciamento de conexões, essencial para o funcionamento adequado da aplicação, especialmente em relação ao desempenho e à segurança, pois garante que as conexões sejam sempre abertas e fechadas de maneira controlada.

10. MONTAGEM DO HARDWARE

A seguir, apresentamos uma visão geral da **estação meteorológica** desenvolvida neste projeto. O sistema foi projetado para monitorar e registrar dados climáticos em tempo real. A comunicação entre os sensores e o software foi implementada para armazenar e visualizar os dados periodicamente, proporcionando uma análise precisa das condições meteorológicas.

A imagem abaixo ilustra toda a arquitetura e funcionamento da estação meteorológica, destacando seus principais componentes e o fluxo de dados.

Figura 16 - Esquema Completo



Fonte: Autores

Para realização da montagem dos sensores foi necessário seguir um passo a passo para que cada um colete os dados, nesse projeto foi usado os seguintes hardwares para compor uma estação meteorológica.

10.1. Pluviômetro

Para construção do pluviômetro irá precisar de:

- Haste roscada de aço inoxidável M4
- Porcas autoblocantes M4
- Arruelas
- Um sensor KY-003
- Um imã

Como funciona:

Na base há uma peça formada por dois tipos de colheres, que é livre para girar. A chuva é coletada pelo funil e canalizada para o centro. A água então preenche parte da peça central primeiro. Em algum momento, a peça vira com o peso da água e a outra metade é preenchida. Este ciclo continua, e cada vez que a peça gira significa que uma certa quantidade de água passou. Um sensor magnético detecta esse movimento e, a cada vez, adiciona uma certa quantidade ao total.

Impressão 3D:

Foi impresso em 3d as várias peças que o pluviômetro é composto, foi usado PLA branco como material, foi usado os modelos de desenhos em arquivos.stl projetados no Tinkercad onde que pode manipular e desenhar formas em 3d para impressão. Abaixo estão as formas que foram feitas.

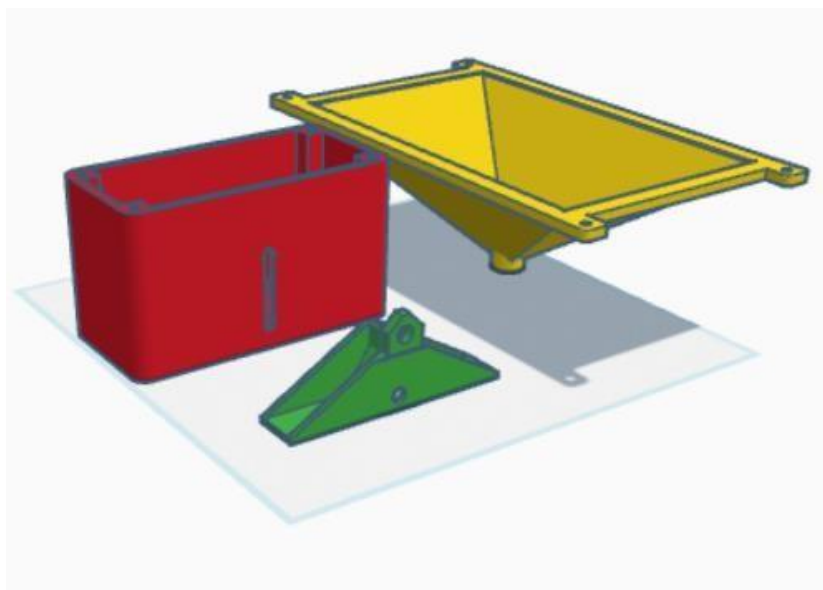


Figura 17- Modelo Pluviômetro

Fonte: Giovanni Aggiustatutto

Figura 18 - Modelo Impresso (Pluviômetro)



Fonte: Autores

Para unir a base e o funil, usarei um pedaço de acrílico, que me permite ver por dentro se tudo está funcionando. Para unir essas três peças, usei alguns pedaços de haste roscada M4, porcas autotravantes M4 e arruelas. No final, tudo deve sair como mostrado na imagem abaixo.



Figura 19 - Pluviômetro montado

Fonte: instructables

A peça central se move conforme a água passa, e usaremos um sensor de efeito Hall e um Arduino para detectar o movimento, assim podendo medir quantidade por milímetro de chuva.

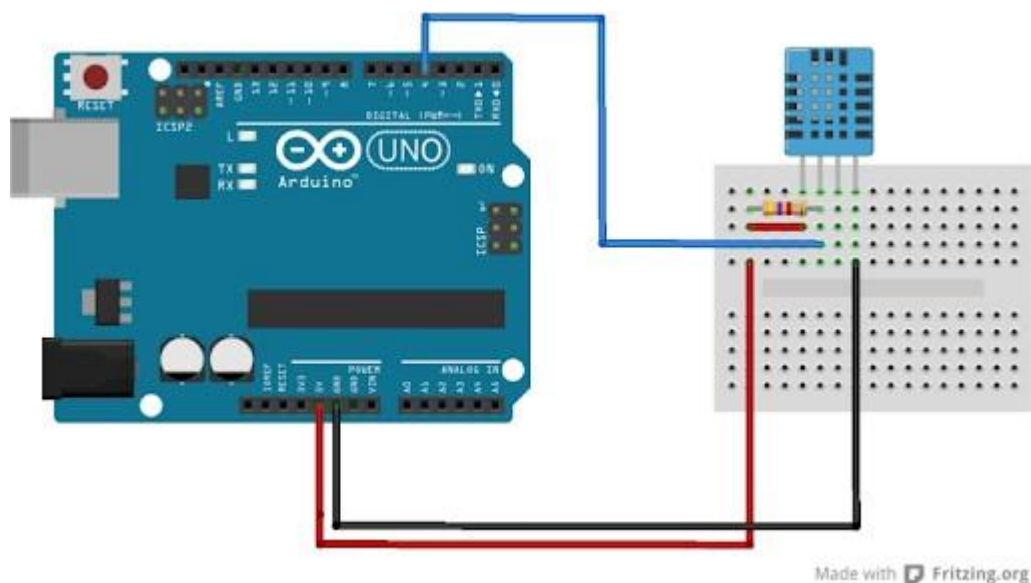
Para obter uma medição mais precisa foi feito um cálculo de quantos milímetro será necessário para mover a peça central, fazendo a medida, será obtido 2,52 milímetros por deslocamento. Já a precipitação em milímetros de altura precisa ser medido o tamanho do funil e encontrar a área. Usando a fórmula do volume inverso cada movimento da peça central corresponde a 0,173 mm de chuva.

A construção do pluviômetro foi um processo que envolveu diversas etapas, desde a seleção dos materiais até a montagem e implementação dos sensores. Utilizando uma haste roscada de aço inoxidável M4, porcas auto blocantes, arruelas e um sensor KY-003, conseguimos desenvolver um dispositivo funcional que mede a precipitação de forma eficaz. Essa precisão é essencial para garantir que os dados coletados sejam confiáveis e úteis para análises meteorológicas.

10.2. DHT11

A princípio, existem 2 tipos de sensor DHT11 no mercado, o sensor que eu vou utilizarnesse projeto, é equivalente ao que está na imagem abaixo, ele 4 pinos de ligação, sendo um pino de sinal, um positivo e um GND:

Figura 20 - Dht11 datasheet



Fonte: fritzing

Como usar o DHT11 com Arduino:

1. Primeiro irá precisar adicionar a biblioteca do sensor a sua IDE, segue imagem abaixo.

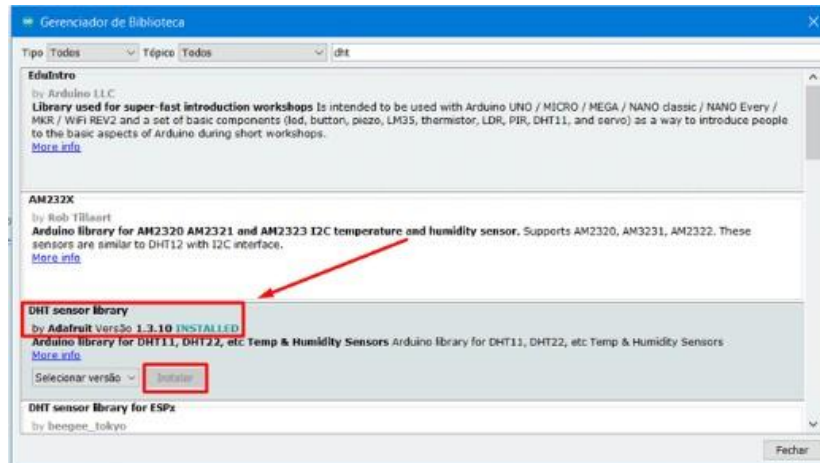


Figura 21 - Biblioteca DHT11

Fonte: autores

2. Utilizar os códigos teste da biblioteca para fazer o upload no Arduino.
3. Após o upload deve se abrir o serial monitor na velocidade de 9600.
4. Pronto, assim deve começar a ver as leituras de temperatura e humidade.

Código Fonte:

Depois do circuito montado, e a biblioteca instalada, irá rodar o seguinte algoritmo:


```

#include <dht11.h>
#define DHT11PIN 4

dht11 DHT11;

void setup()
{
  Serial.begin(9600);
}
void loop()
{
  Serial.println();

  int chk = DHT11.read(DHT11PIN);

  Serial.print("Humidity (%): ");
  Serial.println((float)DHT11.humidity, 2);

  Serial.print("Temperature (C): ");
  Serial.println((float)DHT11.temperature, 2);
  delay(2000);
}

```

Figura 22 - Código dht11

Fonte: Autores

Esse algoritmo instrui o Arduino a ler os dados de temperatura e umidade do sensor DHT11, e mostrar esses dados através da porta Serial, que o Arduino disponibiliza.

O resultado que você deve obter deve ser igual ao da imagem abaixo:

Figura 23 - Serial monitor



Fonte: autores

Por fim, assim você conseguira utilizar o sensor DHT11 de forma correta e irá coletar os dados através de conexão serial para utilizar no projeto Estação meteorológica.

10.3. HC-05 Modulo Bluetooth

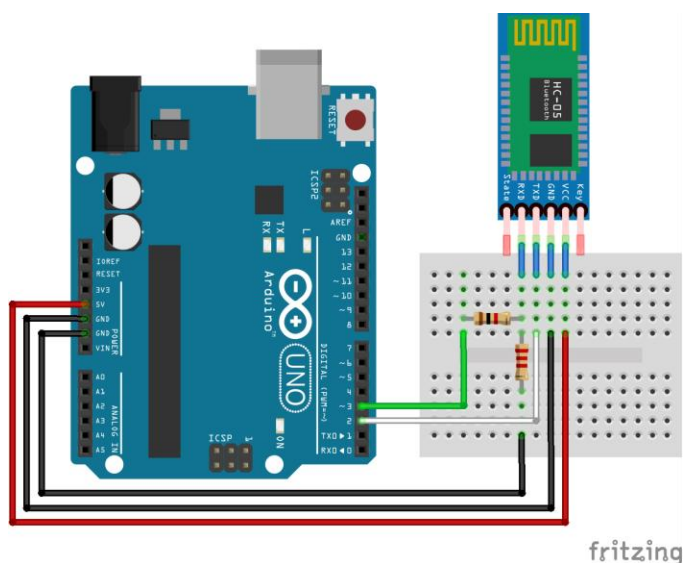
Pode comunicar-se com outros dispositivos (Smartphone, computador ou algum outro micro controlador) via conexão Bluetooth utilizando um módulo HC-05 (mestre/escravo). Desse modo, é possível conectar a sua placa Arduino a outros sistemas, para enviar e receber dados. A comunicação por Bluetooth possibilita, por exemplo, controlar o seu robô por meio de uma aplicação para Smartphone, exibir as medidas de um sensor no seu computador ou ainda permitir que outros robôs se comuniquem entre si.

O módulo Bluetooth HC-05 apresenta 6 pinos que permitem realizar a conexão.

- VCC pino de alimentação. Tipicamente conectado ao pino 5V do Arduino.
- GND terra. Tipicamente conectado ao pino GND do Arduino
- RX pino de recepção. Tipicamente conectado ao pino de transmissão (TX) do Arduino
- TX pino de transmissão. Tipicamente conectado ao pino de recepção (RX) do Arduino
- Status retorna 1 quando o módulo é conectado
- Key ou EN precisam estar alimentados para entrar no modo de configuração e não pode estar conectado para entrar em modo de comunicação.

Esquema de ligação:

Figura 24 - Esquema módulo Bluetooth



Fonte: fritzing

Como o nível lógico do módulo Bluetooth é 3,3V, recomendamos usar um divisor de tensão para reduzir a tensão de 5V do Arduino. (1k de resistência entre pin3 e RX; 2k Ohm entre RX e GND).

Configuração do modulo HC-05:

A configuração do módulo Bluetooth pode ser interessante para verificar o seu funcionamento e para modificar os seus parâmetros, sobretudo se quiser utilizá-lo no modo master. O módulo deve estar alimentado, mas não emparelhado, e o pino Key/EN deve estar alimentado a 3V quando o módulo for aceso. Quando o módulo estiver em modo de configuração, o LED se acenderá por dois segundos a intervalos de dois segundos.

Abra o monitor serial do Arduino e certifique-se de que a opção de fim de linha esteja a exibir “Nova Linha” e que a taxa de transferência esteja regulada a 9600.

Para gerir a comunicação com o módulo HC-05, utilizamos a biblioteca SoftwareSerial.h, que permite criar uma porta serial distinta da utilizada pela porta USB.

Com os módulos já configurados, poderá desconectar o pino Key ou EN, carregar os códigos abaixo nas suas placas Arduino e acender novamente o módulo HC-05 para entrar em modo de comunicação.

Aqui estão algumas das suas funções:

- Software Serial **hc05(RX, TX)** para definir a porta serial e os pinos que servirão para a comunicação.
- **hc05.begin()** para definir a taxa de transmissão.
- **hc05.available()** para verificar se os dados estão disponíveis no buffer da porta serial.
- **hc05.read()** para ler os dados da porta serial, um byte por vez.
- **hc05.print()** para enviar uma cadeia de caracteres em ASCII.
- **hc05.write()** para enviar dados, um byte por vez.
- **hc05.flush()** para esvaziar os dados do buffer.

Código fonte:

Figura 25 - Código Hc-05

```

#include <SoftwareSerial.h>

SoftwareSerial mySerial(10, 11); // RX, TX

void setup() {

  Serial.begin(9600);
  while (!Serial) {

  }

  Serial.println("Goodnight moon!");

  mySerial.begin(4800);
  mySerial.println("Hello, world?");
}

void loop() { |
  if (mySerial.available()) {
    Serial.write(mySerial.read());
  }
  if (Serial.available()) {
    mySerial.write(Serial.read());
  }
}

```

Fonte: autores

Comandos de configuração:

Além da vantagem de poder se utilizado como módulo mestre, o módulo Bluetooth HC- 05 também permite conhecer as configurações salvas.

Em geral, ao digitar no monitor serial AT+<command>? Obtém-se o valor registrado no módulo. Ao inserir a linha AT+<Command>=<Param>, é possível definir um valor para o modulo Veja abaixo alguns comandos AT básicos:

- Para testar a comunicação, digite **AT** no monitor serial da IDE do Arduino. Se tudo estiver bem, o módulo deverá responder OK.
- Para verificar o nome do módulo, digite **AT+NAME=<Param>** no módulo. Ele deverá responder OK.
- Para alterar o código PIN do módulo, digite **AT+PSWD=<Param>**. O módulo deverá responder OK.
- **AT+ROLE=<Param>** para modificar a função do módulo para mestre ou escravo.
- Para modificar a velocidade de transmissão do módulo (somente se necessário), digite **AT+UART=<Param1>,<Param2>,<Param3>**, sendo Param1, 2 e 3 as configurações de comunicação serial: o baud rate, o bit de parada e o bit de paridade, respectivamente.

A particularidade do módulo Bluetooth HC-05 é que pode ser utilizado em modo escravo ou em modo mestre, o que significa que pode conectar-se a um outro módulo Bluetooth de forma autônoma, sem ação externa.

10.4. Ky-003 Sensor Magnético

O módulo KY-003 Hall Magnetic Sensor é um switch que reage à presença de um campo magnético, ligando-se ou desligando-se. Compatível com microcontroladores populares como Arduino, Raspberry Pi e ESP32.

Este módulo consiste em um interruptor de efeito Hall, um resistor de 680Ω , um LED e 3 pinos de cabeçalho macho.

Tabela 4- Configurações Ky-003

Tensão de operação	4,5 V a 24 V
Faixa de temperatura operacional	-40°C a 85°C [-40°F a 185°F]
Dimensões da placa	18,5 mm x 15 mm [0,728 pol. x 0,591 pol.]

Fonte: arduinomodules

Conecte a linha de alimentação da placa (meio), sinal (S) ao pino 3 e o terra (-) a +5 e GND no Arduino, respectivamente.

Tabela 5- Pinagem Ky-003

KY-003	Arduino
S	Pino 3
meio	+5V
—	Terra

Fonte: arduinomodules

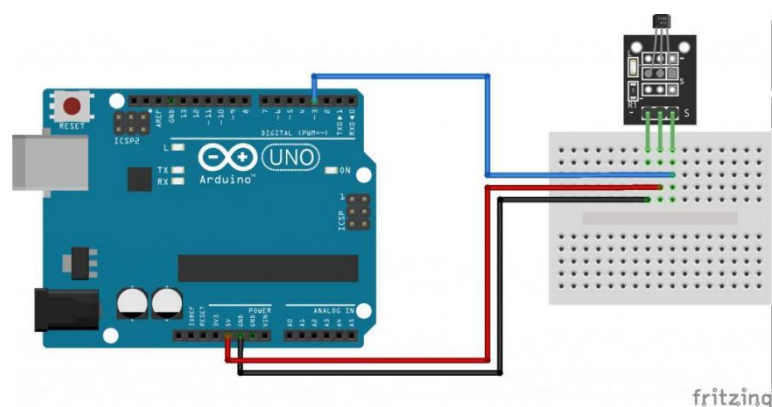


Figura 26 - Datasheet ky-003

Fonte: arduinomodules

10.4.1 Princípio de funcionamento do pluviômetro com o sensor

Um pluviômetro geralmente consiste em uma caçamba que se enche de água quando chove. Quando uma certa quantidade de água é coletada, a caçamba atinge um ponto de inclinação e se vira, esvaziando-se e gerando um pulso.

O sensor, como um interruptor de Reed, é ativado por um ímã que está ligado à caçamba. Quando a caçamba vira, o ímã se aproxima do sensor, fechando o circuito e gerando um pulso elétrico.

Cada pulso gerado indica uma quantidade específica de chuva (0,2 mm por pulso). O microcontrolador conta esses pulsos e, com base na quantidade por pulso, calcula a precipitação total.

Figura 27- Código Ky-003

```

#include <Wire.h>

const float mmPerPulse = 0.173; // mm de chuva por pulso do sensor
float mmTotali = 0;
int sensore = 0;
int statoPrecedente = 0;

void setup() {
  pinMode(9, INPUT);
  Serial.begin(9600); // Inicializa a comunicação serial

  Serial.println("Pluviometro Inicializado");
  Serial.println("Total de Chuva: 0.00 mm");
}

void loop() {
  sensore = digitalRead(9); // Lê o estado do sensor
  if (sensore == HIGH && statoPrecedente == LOW) {
    mmTotali += mmPerPulse; // Atualiza a quantidade total de chuva
    Serial.print("Total de Chuva: ");
    Serial.print(mmTotali);
    Serial.println(" mm"); // Imprime a quantidade total no Monitor Serial
  }
  statoPrecedente = sensore; // Atualiza o estado anterior
}

```

Fonte: autores

O sensor KY-003 será responsável por medir a precipitação de forma precisa. Sempre que o pluviômetro se encher de água, ele acionará o sensor, que fará a leitura em forma de pulso. Essa ação permitirá contabilizar a quantidade de mililitros de chuva de acordo com a capacidade do medidor.

O funcionamento se baseia em um cálculo que relaciona a capacidade volumétrica do pluviômetro com a contagem dos pulsos gerados pelo sensor. A cada pulso registrado, um volume específico de água correspondente ao total acumulado será computado. Dessa forma, é possível calcular a quantidade total de chuva acumulada ao longo do dia, proporcionando dados essenciais para análises climáticas e agronômicas.

10.5. Anemometro

Para realizar a verificação e o controle das rajadas de vento e das suas velocidades médias, podem proporcionar a estimativa do potencial eólico de uma dada região, evitando que acidentes causados pelos ventos causem perdas econômicas tanto em plantações quanto em construções.

Para conseguir medir a velocidade do vento são utilizados instrumentos meteorológicos conhecidos como anemômetro, eles possuem em seu corpo exclusivas conchas conectadas em um eixo que gira sempre que identificada uma rajada de vento, quanto mais voltas ele der maior será a velocidade do vento.

Internamente o Anemômetro Arduino possui um Reed Switch para detectar a velocidade do vento, este equipamento é um interruptor de lâmina com duas

lâminas flexíveis e separadas e hermeticamente seladas dentro de um tubo de vidro com atmosfera interna inerte.

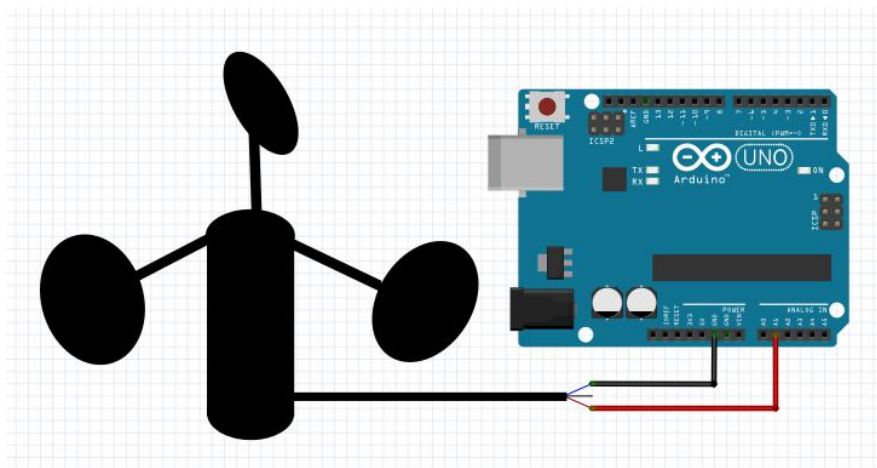
10.5.1 Equipamento utilizado para montagem

- 1 Arduino
- 1 Anemômetro
- 1 Resistor 10k
- Jumpers

10.5.2 Montagem no Arduino

A montagem física Anemômetro é extremamente simples já que trabalha com somente dois fios do anemômetro e um Arduino, em um dos fios do sensor é ligado em uma porta analógica, nesse caso no A4, e o outro pode ser no próprio pino VCC do Arduino, seguindo de acordo com a figura abaixo.

Figura 28 - Anemômetro



Fonte: autores

Para visualização dos valores obtidos precisa usar um código capaz de fazer uma visualização direta no serial do Arduino com um Baud Rate configurado para 9600, seguindo conforme a figura abaixo.

Figura 29 - Código Anemômetro

```

const float pi = 3.14159265; //Número de pi
int period = 5000; //Tempo de medida(miliseconds)
int delaytime = 2000; //Intervalo entre as amostras (miliseconds)
int radius = 147; //Raio do anemometro(mm)
unsigned int Sample = 0; //Armazena o número de amostras
unsigned int counter = 0; //Contador para o sensor
float windspeed = 0; //Velocidade do vento (km/h)

void setup()
{
  pinMode(2, INPUT); //configura o digital 2 como entrada
  digitalWrite(2, HIGH); //internall pull-up active
  Serial.begin(9600); //inicia serial em 9600 baud rate
} //end setup

void loop()
//print m/s
  WindSpeed();
  Serial.print(windspeed);
  Serial.print(" [m/s] ");
  delay(delaytime);
} //end setup
//Função para medir velocidade do vento
void windvelocity()
{
  windspeed = 0;
  attachInterrupt(0, addcount, RISING);
  unsigned long millis();
  long startTime = millis();
  while(millis() < startTime + period) {}
}

void WindSpeed()
{
  windspeed = ((4 * pi * radius * RPM)/60) / 1000; //Calcula a velocidade do vento em m/s
} //end WindSpeed

```

Fonte: autores

Neste projeto deve ser levada em consideração uma informação muito importante para o processo de instalação do Anemômetro no ambiente em que desejar, desenvolvido neste formato, o mesmo precisa ser instalado no sentido horizontal com as conchas o mais linearmente possível.

11. RESULTADOS E ANÁLISE

11.1. Problemas e soluções

Durante o desenvolvimento deste projeto, enfrentamos diversos contratempos que exigiram adaptações e soluções criativas. Embora alguns desafios fossem facilmente identificáveis e corrigíveis, a maioria exigiu a adoção de métodos inovadores, resultando em um considerável investimento de tempo.

Um dos principais obstáculos relacionados aos atrasos foi a integração dos sensores com o Arduino. Apesar de termos realizado uma extensa pesquisa sobre os sensores, encontramos dificuldades em determinar se os dispositivos adquiridos apresentavam defeitos ou se realmente eram os modelos pesquisados. O objetivo inicial era estabelecer uma comunicação eficaz com o Arduino, mas a integração não estava funcionando conforme o esperado.

Na primeira versão do projeto, optamos por integrar o sensor BMP280 diretamente ao Arduino. No entanto, devido aos contratemplos já mencionados, decidimos testar uma alternativa, o sensor BMP180. Apesar de ser um modelo menos preciso, sua implementação também não resultou em sucesso, levando ao seu descarte nas versões atuais do projeto.

Outro desafio significativo foi a escolha da melhor plataforma para a integração da estação meteorológica, com foco na comunicação entre hardware e software via Bluetooth. Inicialmente, planejamos utilizar Python, mas a falta de familiaridade com a linguagem nos levou a optar por C#. Essa decisão resultou na criação de uma interface intuitiva no Visual Studio, permitindo ao usuário visualizar as informações meteorológicas de maneira clara e acessível.

A escolha do Visual Studio se mostrou acertada, pois a plataforma oferece uma vasta gama de materiais de suporte e facilita o desenvolvimento. Sua integração com plataformas externas, como o Arduino, foi um ponto positivo que contribuiu para a eficiência do projeto.

Em resumo, apesar dos desafios enfrentados, a capacidade de adaptação e a escolha adequada das ferramentas foram cruciais para o avanço do projeto, levando à implementação bem-sucedida da estação meteorológica.

12. CONCLUSÃO

A implementação de uma estação meteorológica automática com componentes de baixo custo representa um avanço significativo no monitoramento climático no Brasil. Ao utilizar sensores acessíveis, como o DHT11, o anemômetro e um pluviômetro, juntamente com tecnologias integradas como Arduino e o módulo Bluetooth HC-05, estamos não apenas facilitando a coleta de dados em tempo real, mas também facilitando o acesso a informações meteorológicas cruciais. Essas informações são vitais para setores como a agricultura, onde decisões informadas podem impactar diretamente a produtividade e a sustentabilidade.

O uso do pluviômetro, enriquece a coleta de dados ao permitir a medição precisa da quantidade de chuva, uma informação essencial para o planejamento de irrigação e manejo de culturas. Essa integração fornece uma visão mais completa das condições climáticas locais, capacitando agricultores e comunidades a tomarem decisões mais assertivas em face das variações climáticas.

Apesar das barreiras financeiras, a proposta de uma solução acessível e eficiente visa estabelecer uma rede de monitoramento mais abrangente e colaborativa. Essa rede pode fornecer dados essenciais para a compreensão das variáveis climáticas e suas implicações nas diversas atividades econômicas e sociais.

A adoção dessa tecnologia se alinha à necessidade de adaptarmos e mitigarmos os impactos das mudanças climáticas. É imperativo que comunidades, empresas e instituições se mobilizem para investir e implementar essas soluções inovadoras. Ao avaliar e utilizar os dados coletados, será possível tomar decisões mais informadas e sustentáveis, promovendo resiliência e adaptabilidade em face das adversidades climáticas.

Este projeto não só traz uma solução técnica viável, mas também propõe uma mudança na forma como interagimos com o meio ambiente. Investir em tecnologia acessível é um passo fundamental para garantir que todos tenham acesso às informações necessárias para enfrentar os desafios climáticos de maneira eficaz e responsável. A participação coletiva e o comprometimento com essa causa podem transformar o cenário de monitoramento climático no Brasil, favorecendo o desenvolvimento sustentável e a preservação dos recursos naturais para as futuras gerações.

13.REFERÊNCIAS

alldatasheet.com. “ATMEGA328P Datasheet(PDF) - ATMEL Corporation.” Disponível em: www.alldatasheet.com/datasheetpdf/pdf/241077/ATMEL/ATMEGA328P.html. Acesso em: 19 abr 2024

ARDUINO. O que é Arduino? Disponível em: <https://www.arduino.cc/en/Guide/Introduction>. Acesso em: 4 mar 2024.

ARDUINOMODULES. KY-003 Hall Magnetic Sensor Module - ArduinoModulesInfo. Disponível em: https://arduinomodules.info/ky-003-hall-magnetic-sensor-module/#google_vignette. Acesso em: 24 set. 2024.

CIA, A. E. Detectando campos magnéticos com o Sensor Hall KY-003 - Arduino e Cia. Disponível em: <https://www.arduinoecia.com.br/sensor-hall-ky-003-arduino/>. Acesso em: 17 set. 2024.

Como Usar Com Arduino: Tutorial Módulo Bluetooth. Disponível em: <https://flaviobabos.com.br/bluetooth-arduino/>. Acesso em: 7 set 2024

Como utilizar o sensor de pressão atmosférica – Barômetro BMP180 com Arduino – Blog da Robótica. Disponível em: <https://www.blogdarobotica.com/2022/10/06/como-utilizar-o-sensor-de-pressao-atmosferica-barometro-bmp180-com-arduino/>. Acesso em: 9 abr. 2024.

ENG. JEMERSON MARQUES. Módulo Bluetooth HC-05 - Características e Especificações. Pinagem - Pinout! Disponível em: <https://www.fvml.com.br/2023/04/modulo-bluetooth-hc-05-caracteristicas.html>. Acesso em: 7 set 2024

HC-05 pinout, specifications, datasheet and HC05 Arduino connection. Disponível em: <https://www.etechnophiles.com/hc-05-pinout-specifications-datasheet/>. Acesso 3 ago 2024.

HOLZ, Edson; CAMPIGOTO, Evandro. Desenvolvimento de uma estação meteorológicamicrocontrolada para monitoramento de variáveis climáticas. 2019. Acesso 24 mar 2024.

INSTRUSUL. Para que serve o Anemômetro - Blog da Instrusul. Disponível em: <https://blog.instrusul.com.br/para-que-serve-o-anemometro/#:~:E=Este%20%C3%A9%20o%20dispositivo%20de%20medi%C3%A7%C3%A3o%20mais%20simples>. Acesso 27 ago 2024.

Instituto Nacional de Meteorologia - INMET. ([s.d.]). Gov.br. Disponível em: <https://portal.inmet.gov.br/sobre-meteorologia>. Acesso em: 3 mar.2024.

Incaper Coordenação de Meteorologia. Disponível em: <https://met>

MySQL :: MySQL Workbench Manual :: 1 General Information. Disponível em: <https://dev.mysql.com/doc/workbench/en/wb-intro.html>. Acesso em: 23 out 2024.

MASTERWALKER SHOPMASTERWALKER ELECTRONIC SHOP, 14 jan. 2019.
Disponível em: <https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-sensor-de-pressao-e-temperatura-bmp180>. Acesso em: 3 abr. 2024.

OLIVEIRA, E. Como usar com Arduino - Sensor de Pressão e Temperatura BMP180. BL

O que é Arduino: Tudo o que você precisa saber.

Disponível em: <https://www.hostgator.com.br/blog/o-que-e-arduino/>. Acesso em: 3 abr. 2024.

O que é um Anemômetro - Guia do Comprador.

Disponível em: <https://www.splabor.com.br/blog/guia-do-comprador/o-que-e-um-anemometro-confira-sua-utilizacao/>. Acesso 9 jul 2024

Rede de Estações INMET. Laboratório de Climatologia E Análise Ambiental.

Disponível em: <https://www2.ufjf.br/labcaa/rede-de-estacoes-inmet/>. Acesso em: 5 jun. 2024.

ROBOCORE. Usando o Sensor de Efeito Hall US1881 - RoboCore. Disponível

em: <https://www.robocore.net/tutoriais/usando-sensor-efeito-hall-us1881#:~:>. Acesso em: 17 jun 2024

SENSORTEC, B. Digital Pressure Sensor. Disponível em: <https://blogmasterwalkershop.com.br/arquivos/datasheet/Datasheet%20BMP280.pdf>. Acesso em: 3 abr. 2024.

SILVA, R. B. et al. Estações meteorológicas de código aberto: Um projeto de pesquisa e desenvolvimento tecnológico. Revista Brasileira de Ensino de Física, v. 37, n. 1, p. 1505, 2015. (EspaçoReservado1)

THOMAZINI, Daniel; ALBUQUERQUE, Pedro Urbano Braga de. Sensores industriais. [Digite o Local da Editora]: Editora Saraiva, 2020. E-book. ISBN 9788536533247.

Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788536533247/>. Acesso em: 11 abr. 2024.

YNOUE, Rita Yuri et al. Meteorologia: noções básicas. São Paulo: Oficina de Textos, 2017