

Alan Nascimento de Paula

**Implementação de um Sistema de
Sensoriamento em VANTs para a Manutenção
Preditiva e Atuação em Emergências no
Contexto da Indústria 4.0**

Uberlândia-MG

2024

Alan Nascimento de Paula

Implementação de um Sistema de Sensoriamento em VANTs para a Manutenção Preditiva e Atuação em Emergências no Contexto da Indústria 4.0

Projeto de conclusão de curso de graduação apresentado ao colegiado do curso de Engenharia Mecatrônica da Faculdade de Engenharia Mecânica da Universidade Federal de Uberlândia, como parte dos requisitos a obtenção do título de Bacharel em Engenharia Mecatrônica.

Universidade Federal de Uberlândia
Faculdade de Engenharia Mecânica - FEMEC
Graduação em Engenharia Mecatrônica

Orientadora
Prof Dra^o.: Gabriela Vieira Lima

Uberlândia-MG

2024

Alan Nascimento de Paula

Implementação de um Sistema de Sensoriamento em VANTs para a Manutenção Preditiva e Atuação em Emergências no Contexto da Indústria 4.0/ Alan Nascimento de Paula. – Uberlândia-MG, 2024-

120 p. : il. (algumas color.); 30 cm.

Orientadora

Prof Dra^o.: Gabriela Vieira Lima

Trabalho de Conclusão de Curso – Universidade Federal de Uberlândia

Faculdade de Engenharia Mecânica - FEMEC

Graduação em Engenharia Mecatrônica, 2024.

1. Veículo aéreo não tripulado. 2. Sensoriamento remoto. 3. Sistemas embarcados. 4. Indústria 4.0. I. Gabriela Vieira Lima. II. Universidade Federal de Uberlândia. III. Faculdade de Engenharia Mecânica. IV. Laboratório de automação, sistemas eletrônicos e controle. V. Implementação de um sistema de sensoriamento em vants para a manutenção preditiva e atuação em emergências no contexto da indústria 4.0.

CDU 02:141:005.7

Alan Nascimento de Paula

Implementação de um Sistema de Sensoriamento em VANTs para a Manutenção Preditiva e Atuação em Emergências no Contexto da Indústria 4.0

Projeto de conclusão de curso de graduação apresentado ao colegiado do curso de Engenharia Mecatrônica da Faculdade de Engenharia Mecânica da Universidade Federal de Uberlândia, como parte dos requisitos a obtenção do título de Bacharel em Engenharia Mecatrônica.

Prof Dra^o.: Gabriela Vieira Lima
Orientadora

Professor Dr^o.:
José Jean-Paul Zanlucchi de Souza Tavares

Professor Me^o.:
Éder Alves de Moura

Uberlândia-MG
2024

Dedico este trabalho a minha mãe Raquel Silva, ao meu irmão Luis Gustavo e a todos aqueles que promovem a ciência e a tecnologia, que são amantes da descoberta, que contemplam os números e veem lógica na complexidade.

Agradecimentos

Este projeto sinaliza o término da minha segunda graduação. Iniciar outra graduação do “zero” em um curso de graduação rigoroso como a Engenharia Mecatrônica já tendo feito outro curso bastante exigente trouxe sentimentos de angústia e incerteza sobre o caminho que eu estava trilhando. A minha trajetória acadêmica no curso de Engenharia Mecatrônica não foi fácil. O “gás” já não era o mesmo de outrora, somado a isso, enfrentar novamente toda a burocracia de um curso superior, pandemia e os dilemas da vida adulta fizeram com que a minha jornada acadêmica fosse bastante desafiadora. Entretanto, eu tive pessoas ao meu lado que sempre me apoiaram e fizeram com que esse percurso pudesse ser percorrido e finalizado com êxito. Neste sentido, eu quero agradecer àqueles que contribuíram para que isso fosse possível.

Agradeço . . .

. . . à minha mãe [Raquel Silva] por sempre me incentivar a estudar, desde que eu me entendo por gente. Pela paciência, apoio emocional (e financeiro). Ao meu irmão [Luis Gustavo] pelo convívio, companheirismo e incentivo na minha trajetória universitária em Uberlândia. Você é um cara brilhante! Espero um dia poder retribuir toda a ajuda recebida por vocês. Se eu tive força para continuar, devo isso muito a vocês!

. . . aos amigos que obtive ao longo da minha jornada.

. . . ao [Herquinho] por toda prontidão e ajuda dada a mim quando eu precisei.

. . . aos [professores] que contribuíram para a minha formação escolar e acadêmica.

. . . a [Equipe de Desenvolvimento em Robótica Móvel] (EDROM) pela oportunidade concedida a mim para que eu pudesse trabalhar com robótica.

. . . ao [Laboratório de Automação, Sistemas Eletrônicos e Controle] (LASEC) pela oportunidade de realização de um projeto de iniciação científica e por fornecer as condições necessárias para que esse trabalho fosse realizado.

. . . à comunidade de desenvolvedores que contribuem na criação de *softwares* e *hardwares open source*, cujo trabalho possibilita que projetos como esse possam ser realizados.

. . . a todos àqueles que direta ou indiretamente “caminharam” para que hoje eu

pudesse “correr”.

. . . a [UFU] por ter sido a minha “casa” e por ter me ajudado a crescer como pessoa e intelectualmente.

. . . à minha orientadora [Gabriela Vieira Lima] pela orientação no projeto de iniciação científica, no estágio e neste projeto de monografia. Obrigado pela ajuda!

. . . e o apoio financeiro fornecido pela Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) para o projeto APQ-00594-22, intitulado “Investigação do uso de VANTs para manutenção preditiva e atuação em emergências no contexto da Indústria 4.0.”

. . . *De qualquer modo, nos preocupamos mais com o que acontece agora que com o que acontecerá no futuro. O futuro sempre será resultado dos acontecimentos presentes*^{*} . . .

.....
^{*} Pequena fração do texto “A Definição da Consciência Negra”, escrito por Bantu Steve Biko, em dezembro de 1971. Disponível na *webpage*: [A Definição da Consciência Negra](#). Acesso: 31 de outubro de 2023.

“For millions of years, mankind lived just like the animals. Then something happened which unleashed the power of our imagination. We learned to talk and we learned to listen. Speech has allowed the communication of ideas, enabling human beings to work together to build the impossible. Mankind’s greatest achievements have come about by talking, and its greatest failures by not talking. It doesn’t have to be like this. Our greatest hopes could become reality in the future. With the technology at our disposal, the possibilities are unbounded. All we need to do is make sure we keep talking.” (Stephen Hawking)

Resumo

As inovações exigidas pela indústria 4.0 irão transformar a maneira como as indústrias trabalham e se relacionam com o consumidor final. De fato, ao passo que novas tecnologias surgem e as demandas do mercado se alteram ocorrem transformações profundas em todos os segmentos da sociedade. Neste cenário, áreas promissoras como sistemas embarcados, sensoriamento remoto e a utilização de veículos aéreos não tripulados (VANTs) em diferentes contextos despontam visando atender as necessidades e os desafios impostos aos agentes inseridos nesse processo. Esse ecossistema tecnológico visa impulsionar o desenvolvimento e melhoria dos processos produtivos possibilitando modificar substancialmente os setores industriais. A utilização de sistemas inteligentes, soluções em nuvem e a análise de dados mostram-se bastante promissoras para o setor de manutenção. Busca-se a conectividade, a integração dos sistemas, a eficiência operacional, entre outros, com o mínimo de interferência humana. Neste sentido, este projeto apresenta uma solução que abrange o tema. Para isso, foi desenvolvido um sistema de monitoramento da temperatura e da umidade de um ambiente *indoor* de interesse, onde a unidade sensorial **AHT10** foi embarcada em um nano quadricóptero modelo *Crazyflie 2.1*. Para garantir a interatividade utilizando o protocolo de comunicação I^2C do sensor com o drone foi desenvolvido e implementado um *firmware* embutido no *hardware* do veículo, sendo a aplicação de baixo nível do sistema. As operações de controle de voo, a captura dos *logs* relevantes para a criação de uma base de dados, o tratamento das informações capturadas e a criação da aplicação *web* foram feitas via *scripts* na linguagem de programação *Python*. Foram realizados três ensaios de voo obtendo-se as temperaturas médias de **29.07 °C**, **31.09 °C**, **31.64 °C** e umidade relativa do ar de **49.90 %**, **44.48 %**, **43.00 %** respectivamente. Pôde-se demonstrar, mesmo em um local restrito, o comportamento típico da atmosfera, os valores da temperatura e da umidade relativa do ar são modificados com a variação da altitude. Para alturas menores a temperatura é maior e diminui ao passo que a altitude aumenta, em contraste, a umidade relativa do ar tende a aumentar com a altura, pois para temperaturas mais elevadas o ar fica mais seco. Podemos destacar ainda, devido ao tamanho reduzido da aeronave as rotinas são mais eficientes, rápidas e seguras quando comparado as maneiras tradicionais de coleta de dados, com isso o veículo pode ser um grande aliado para o setor de manutenção preditiva. Contudo, os resultados obtidos na monografia podem ser expandidos visando a utilização de sensores mais complexos, como os desenvolvidos para a coleta de fumaça, gases como o etanol, entre outros, onde os ensaios são mais restritos e regulados visando evitar acidentes.

Palavras-chaves: Veículos aéreos não tripulados, sensoriamento remoto, sistemas embarcados, indústria 4.0.

Abstract

The innovations required by Industry 4.0 will transform how industries operate and interact with the end consumer. As new technologies emerge and market demands change, profound transformations take place in all segments of society. In this scenario, promising areas such as embedded systems, remote sensing, and uncrewed aerial vehicles (UAVs) in different contexts arise to meet the needs and challenges imposed on agents inserted in this process. This technological ecosystem aims to enhance the development and improvement of production processes, enabling substantial modifications to industrial sectors. Intelligent systems, cloud solutions, and data analysis are up-and-coming for the maintenance sector. We seek connectivity, systems integration, and operational efficiency, among others benefits, with minimal human intervention. In this sense, this project presents a solution that covers the theme. For this purpose, a system was developed to monitor the temperature and humidity of an indoor environment of interest utilizing the **AHT10** sensory unit on a nano quadcopter model, specifically the Crazyflie 2.1. To ensure interactivity using I^2C communication protocol between the sensor and the drone, firmware was embedded in the vehicle's hardware and implemented, serving as the low-level application of the system. The flight control operations, log capture, information processing, and web application creation were performed using scripts written in the Python programming language. Three flight tests were performed, obtaining the average temperatures of **29.07 °C**, **31.09 °C**, **31.64 °C**, and relative humidity of the air of **49.90 %**, **44.48 %**, **43.00 %** respectively. It could be demonstrated that even in a restricted location, the atmosphere's typical behavior, temperature, and relative humidity are modified with altitude variation. For lower altitudes, the temperature is higher and decreases as the altitude increases. In contrast, humidity-relative air tends to increase with height because, for higher temperatures, the air gets drier. We can also highlight that, due to the reduced size of the aircraft, the routines are more efficient, faster, and safer when compared to traditional methods of data collection. With this, the vehicle can be an excellent ally for the predictive maintenance sector. However, the results obtained in the monograph can be expanded to the use of more complex sensors, such as those developed for collecting smoke and gases, including ethanol, among others, where the tests are more restricted and regulated, aiming to avoid accidents.

Keywords: Unmanned aerial vehicles, remote sensing, embedded systems, industry 4.0.

Lista de ilustrações

Figura 1 – Diagrama simples de um sistema embarcado contendo um microcontrolador monitorando o ambiente.	33
Figura 2 – Aplicações tecnológicas que utilizam sistemas embarcados.	34
Figura 3 – Ecossistema de tecnologias digitais que utilizam sistemas embarcados. O microcontrolador e/ou o microprocessador é utilizado para processar as informações dos dispositivos eletrônicos. O <i>firmware</i> realiza as instruções computacionais via código escrito em alguma linguagem de programação.	35
Figura 4 – Em (a) tem-se o microprocessador modelo Apple M1 . Em (b) tem-se o microcontrolador modelo ATmega1609 de 8 <i>bits</i>	36
Figura 5 – Sensoriamento remoto aéreo realizado por diferentes plataformas, tais como satélite, veículos aéreos tripulados e não tripulados.	37
Figura 6 – Panorama do processo de industrialização ao longo dos séculos.	38
Figura 7 – Representações digitais de “1s” e “0s”.	44
Figura 8 – Exemplo de barramento compartilhado que faz uso do protocolo de comunicação I^2C	45
Figura 9 – <i>Frame</i> de dados do protocolo I^2C	46
Figura 10 – Exemplo de barramento que utiliza o protocolo de comunicação I^2C . Na via de dados existem dois dispositivos controladores da via e dois equipamentos alvos da transmissão de informação. A comunicação é bidirecional, <i>half-duplex</i> , tanto os controladores quanto os alvos podem enviar e receber os dados.	47
Figura 11 – Envio de mensagem no protocolo de comunicação I^2C	48
Figura 12 – Panorama dos <i>frames</i> utilizados no protocolo de comunicação I^2C	48
Figura 13 – Ilustração de um sensor como um sistema de uma única variável.	49
Figura 14 – Comportamento típico realizado na operação de instrumentos de medição. Em (a) temos o comportamento em malha aberta, não há o uso de realimentação. Em (b) temos o comportamento em malha fechada, onde a realimentação atua no meio.	50
Figura 15 – Unidade sensorial AHT10 de temperatura e umidade.	51

Figura 16 – Diagrama de tempo digital do sensor AHT10 . Na via SDA as linhas mais espessas são controladas pelo sensor, as linhas pontilhadas na posição horizontal são arbitradas pelo microcontrolador STM32F405 da aeronave <i>Crazyflie 2.1</i> . A explicação sobre os <i>inputs/outputs</i> é apresentada na Tabela (3).	52
Figura 17 – Nano quadricóptero modelo <i>Crazyflie 2.1</i>	55
Figura 18 – Eixos de controle utilizados na manobrabilidade da aeronave <i>Crazyflie 2.1</i>	57
Figura 19 – Aspecto de funcionamento do nano quadricóptero modelo <i>Crazyflie 2.1</i>	58
Figura 20 – Sistema de posicionamento local utilizado na localização espacial do <i>Crazyflie 2.1</i>	59
Figura 21 – Posicionamento das âncoras no ambiente. Dimensões do laboratório de ensaio.	60
Figura 22 – Elementos utilizados no sistema de orientação espacial via rádio. Em (a) tem-se o receptor. Em (b) tem-se a âncora.	61
Figura 23 – Em (a) tem-se o <i>Crazyradio</i> ; em (b) tem-se o <i>Crazyflie Client</i>	62
Figura 24 – Componentes do nano quadricóptero <i>Crazyflie 2.1</i>	64
Figura 25 – Momento de inércia de massa de um corpo rígido bidimensional qualquer.	66
Figura 26 – <i>CAD</i> da aeronave <i>Crazyflie 2.0</i>	67
Figura 27 – Arrasto e empuxo agindo em um perfil bidimensional.	68
Figura 28 – Simulação via <i>CAD</i> para obtenção do coeficiente de arrasto.	69
Figura 29 – Representação gráfica do ensaio experimental para determinar o empuxo da aeronave.	70
Figura 30 – Experimento para a aquisição do coeficiente de empuxo da aeronave.	71
Figura 31 – Esquema utilizado na soldagem dos pinos correspondentes da aeronave <i>Crazyflie 2.1</i> e do sensor AHT10 responsáveis pela comunicação serial <i>I²C</i>	74
Figura 32 – Aeronave embarcada com o sensor AHT10	75
Figura 33 – Diagrama de gestão de <i>tasks</i> no <i>FreeRTOS</i>	76
Figura 34 – Camada de aplicação do projeto.	78
Figura 35 – Carregador de bateria sem fio da aeronave <i>Crazyflie 2.1</i>	79
Figura 36 – Aplicação <i>web</i> desenvolvida para o projeto, lado do cliente.	80
Figura 37 – Comportamento típico da unidade sensorial quando os motores do drone estão desligados. A coleta dos dados foram feitas em um ambiente <i>indoor</i>	81

- Figura 38 – **Ensaio 1:** A fig.(38a) mostra a temperatura ($^{\circ}\text{C}$) e a umidade (%) em função do horário. A temperatura ($^{\circ}\text{C}$) é reduzida à medida que o drone aumenta a altitude do voo, em contraste, a umidade relativa do ar (%) aumenta. A fig.(38b) mostra a trajetória percorrida pela aeronave *Crazyflie 2.1*, e as projeções no plano x-z. O valor médio da temperatura foi de 29.07°C e a média da umidade relativa do ar foi de 49.90% . . . 84
- Figura 39 – **Ensaio 2:** A fig.(39a) mostra a temperatura ($^{\circ}\text{C}$) e a umidade (%) em função do horário. A temperatura ($^{\circ}\text{C}$) é reduzida à medida que o drone aumenta a altitude do voo, em contraste, a umidade relativa do ar (%) aumenta. A fig.(39b) mostra a trajetória percorrida pela aeronave *Crazyflie 2.1*, e as projeções no plano x-z. O valor médio da temperatura foi de 31.09°C e a média da umidade relativa do ar foi de 44.48% . . . 85
- Figura 40 – **Ensaio 3:** A fig.(40a) mostra a temperatura ($^{\circ}\text{C}$) e a umidade (%) em função do horário. A temperatura ($^{\circ}\text{C}$) é reduzida à medida que o drone aumenta a altitude do voo, em contraste, a umidade relativa do ar (%) aumenta. A fig.(40b) mostra a trajetória percorrida pela aeronave *Crazyflie 2.1*, e as projeções no plano x-z. O valor médio da temperatura foi de 31.64°C e a média da umidade relativa do ar foi de 43.00% . . . 86

Lista de tabelas

Tabela 1	– Taxa de transferência na comunicação para o protocolo I^2C .	45
Tabela 2	– Endereços I^2C úteis para a comunicação com o sensor AHT10 .	51
Tabela 3	– Especificação de tempo típico na comunicação I^2C para <i>input/output</i> da unidade sensorial AHT10 .	52
Tabela 4	– Medidas físicas do <i>Crazyflie 2.1</i> .	63
Tabela 5	– Arranjo da massa do nano quadricóptero <i>Crazyflie 2.1</i> .	65
Tabela 6	– Ensaio de empuxo da aeronave <i>Crazyflie 2.0</i> .	70
Tabela 7	– Ensaio de voo da aeronave <i>Crazyflie 2.1</i> .	81

Lista de abreviaturas e siglas

VANT	Veículo Aéreo não Tripulado
<i>CPU</i>	Unidade Central de Processamento (<i>Central Processing Unit</i>)
<i>PC</i>	Computador Pessoal (<i>Personal Computer</i>)
<i>Software</i>	Conjunto lógico de programas, dados e processos que possibilitam a execução de tarefas em computadores e dispositivos eletrônicos
<i>Hardware</i>	Conjunto físico de componentes interno ou externo de dispositivos eletrônicos
Conversor A/D	Conversor Analógico-Digital
Conversor D/A	Conversor Digital-Analógico
<i>PROM</i>	Memória programável somente de leitura (<i>Programmable Read-Only Memory</i>)
<i>EPROM</i>	Memória programável apagável somente de leitura (<i>Erasable Programmable Read-Only Memory</i>)
CI	Circuito Integrado
<i>Bit</i>	Unidade de informação computacional que pode ser salva ou comunicada. Assume dois valores: 0 ou 1
<i>PCB</i>	<i>Printed Circuit Board</i>
PCI	Placas de Circuito Impresso
<i>AVR</i>	<i>Advanced Virtual RISC</i> ou <i>Reduced Instruction Set Computing</i> . Arquitetura <i>RISC</i> de microcontrolador
<i>UAVs</i>	<i>Unmanned Aerial Vehicles</i>
TI	Tecnologias de Informação
<i>Payload</i>	Capacidade de Carga

<i>IMU</i>	Unidade de Medição Universal (<i>Inertial Measurement Unit</i>)
<i>Shield</i>	Placa Utilizada para Posicionamento
<i>Wireless</i>	Comunicação ou transmissão de informação sem a utilização física de cabos ou fios.
<i>CAD</i>	Projeto ou desenho assistido por computador (<i>Computer-Aided Design</i>)
SO	Sistema Operacional
<i>Thrust</i>	Em português é conhecido como empuxo, é responsável por ajustar a altitude ou a altura do drone
<i>Roll</i>	Em português é conhecido como rolamento, realiza a rotação em torno do eixo horizontal do drone, uma linha imaginária que passa da parte de trás para frente no drone
<i>Pitch</i>	Em português é conhecido como arfagem, realiza a rotação em torno do eixo horizontal que atravessa o drone da esquerda para a direita
<i>Yaw</i>	Em português é conhecido como guinada, realiza a rotação em torno do eixo vertical do drone
<i>Internet of Things (IoT)</i>	Conjunto de equipamentos conectados na rede que compartilham dados e informações
<i>Smart Factory</i>	Rede interconectada de máquinas com o intuito de automatizar os processos de maneira mais eficiente em uma indústria utilizando tecnologias avançadas
<i>FreeRTOS</i>	<i>Kernel</i> de um sistema operacional de tempo real para sistemas embarcados
<i>Indoor</i>	Ambiente interno fechado
<i>MSB</i>	<i>Most Significant Bit</i> : é o <i>bit</i> de maior valor na representação binária de um número
<i>LSB</i>	<i>Least Significant Bit</i> : é o <i>bit</i> de menor valor na representação binária de um número
<i>SDA</i>	<i>Serial data line</i>
<i>SCL</i>	<i>Serial clock line</i>
<i>VIN</i>	<i>Input voltage</i>

V_{DD}	<i>Supply voltage</i>
GND	<i>Power ground</i>
ACK	<i>Acknowledge</i>
I^2C	<i>Inter-Integrated Circuit</i>
CAN	<i>Controller Area Network</i>
$OLED$	<i>Organic light-emitting diode</i>
URI	<i>Uniform Resource Identifier</i>
$CRTP$	<i>Crazyradio Real-Time Protocol</i>
$HTML$	<i>Hypertext Markup Language</i>
CSS	<i>Cascading Style Sheets</i>
Fig.	Figura
Pg.	Página
Eq.	Equação
$Localhost$	Servidor local

Lista de símbolos

$I_x = \int_m y^2 dm$ Momento de inércia de massa em torno do eixo x

$I_y = \int_m x^2 dm$ Momento de inércia de massa em torno do eixo y

$I_z = I_x + I_y$ Momento de inércia de massa em torno do eixo z

$C_D = \frac{F_D}{\frac{1}{2}\rho V^2 A}$ Coeficiente de arrasto teórico

$b = \frac{mg}{4\Omega^2}$ Coeficiente de empuxo experimental

$J = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix}$ Matriz de inércia de um quadricóptero

Sumário

1	INTRODUÇÃO	31
1.1	Sistemas Embarcados	32
1.1.1	<i>Firmware</i> , Microprocessador & Microcontrolador	34
1.2	Sensoriamento Remoto via Dados	36
1.3	Indústria 4.0	37
1.4	Estado da Arte	39
1.5	Justificativas	40
1.6	Objetivos	40
1.6.1	Objetivos Específicos	41
1.7	Organização da Monografia	41
2	GERENCIAMENTO DE INFORMAÇÕES EM DISPOSITIVOS DIGITAIS	43
2.1	Protocolo de Comunicação <i>I²C</i>	44
2.2	Unidade Sensorial AHT10 de Temperatura e Umidade	49
3	AERONAVE <i>CRAZYFLIE 2.1</i>	55
3.1	Estrutura de Funcionamento do Drone	56
3.2	Orientação Espacial do Drone	58
3.3	Transmissão de Dados do Drone	61
3.4	Informações Físicas do Drone	62
3.4.1	Massa	63
3.4.2	Distância Entre o Centro de Massa e os Rotores	63
3.4.3	Momento de Inércia	65
3.4.4	Coeficiente de Arrasto	67
3.4.5	Coeficiente de Empuxo	68
4	METODOLOGIA UTILIZADA NO PROJETO	73
4.1	Adequação do <i>Hardware</i> da Aeronave <i>Crazyflie 2.1</i>	73
4.2	Implementação do <i>Software</i> Embarcado & <i>Scripts</i> na Aeronave <i>Crazyflie 2.1</i>	74
4.2.1	<i>Firmware</i>	75
4.2.2	Acionamento da Aeronave, Coleta de <i>Logs</i> & Camada de Aplicação	76
4.2.3	Ensaios	80
4.2.4	Caracterização do Sensor AHT10	80

5	RESULTADOS OBTIDOS	83
6	CONCLUSÕES & CONSIDERAÇÕES FINAIS	87
	REFERÊNCIAS	89
	 APÊNDICES	 93
APÊNDICE A	– FERRAMENTAS COMPUTACIONAIS ÚTEIS AO DESENVOLVIMENTO DO PROJETO	95
APÊNDICE B	– <i>FIRMWARE</i> DESENVOLVIDO PARA A COMUNICAÇÃO ENTRE A UNIDADE SENSORIAL E A AERONAVE <i>CRAZYFLIE 2.1</i>	99
	 ANEXOS	 103
ANEXO A	– <i>DATASHEET</i> DO SENSOR AHT10	105
ANEXO B	– CONECTOR DE EXPANSÃO MULTIPLEXADOR DA AERONAVE	119

CAPÍTULO 1

INTRODUÇÃO

O constante estímulo na busca por produtividade, processos mais baratos e eficientes impulsiona o desenvolvimento e aprimoramento das relações profissionais. Vivencia-se uma era de grandes transformações que estão alterando as formas de atividade laboral e as convenções produtivas em todo o mundo. A difusão digital e a busca pela maturação de tecnologias complexas como robótica avançada, internet das coisas, inteligência artificial, computação em nuvem, entre outras, modifica o setor industrial, econômico, social e político e traz consigo controvérsias e desafios.

A quarta revolução industrial, denominada de indústria 4.0 incorpora essas inovações. Nesta, procura-se a conectividade, o uso de sistemas inteligentes, soluções em nuvem e a análise de dados, de modo a garantir a eficiência e a confiabilidade das operações industriais, com o mínimo de intervenção humana. A implementação dessas técnicas propicia novas possibilidades para a cadeia produtiva e tem grande potencial de impulsionar o ecossistema de manutenção fabril.

De extrema importância em aplicações civis e militares, os veículos aéreos não tripulados (VANTs) vem instigando o interesse de pesquisadores. A ascensão de técnicas como coleta e análise de dados, monitoramento de grandes áreas, filmagens, entre outros, provê um leque robusto de aplicações tecnológicas [1]. De fato, sistemas inteligentes embarcados em drones contribuem para a automatização de processos e podem ser utilizados em parques industriais monitorando variáveis como vazamento de gases, temperatura e umidade, sendo um instrumento valioso na execução de tarefas complicadas, custosas e de grande risco para os seres humanos [2–5].

À luz da indústria 4.0 esses veículos podem ser grandes aliados. A integração de veículos aéreos não tripulados ao sistema de gerenciamento e manutenção de uma planta industrial gera ganhos significativos. Na manutenção preditiva, por exemplo, há a possibilidade da realização do voo de drones de maneira distribuída e cooperativa provendo rotinas mais eficientes, rápidas e seguras na inspeção dos equipamentos e processos.

Portanto, o uso dessas aeronaves na execução de tarefas é bastante valioso para a

indústria moderna. Agrega-se produtividade, destreza e minimiza-se o risco em atividades repetitivas e laboriosas aos seres humanos. Intrinsecamente, leva-se a economia de custos e aumento da capacidade de gerenciamento do negócio contribuindo-se para a valorização do produto final.

Nesta monografia investigamos a implementação de um sistema de detecção e análise de dados coletados de temperatura e umidade via o sensor **AHT10** embarcado em um drone modelo *Crazyflie 2.1*. Organizamos o texto da seguinte maneira, na introdução apresento resumidamente (i) as características de um sistema embarcado; (ii) *firmware*, microprocessador & microcontrolador; (iii) sensoramento remoto via dados; (iv) indústria 4.0; (v) estado da arte; (vi) justificativas; (vii) objetivos e (viii) objetivos específicos. Em seguida, apresento o conteúdo da monografia e os resultados obtidos.

1.1 Sistemas Embarcados

Os equipamentos eletrônicos necessitam de um dispositivo que realiza as funções de cálculo e tomada de decisões. Estes componentes em grande maioria, tem como base tecnológica o microprocessador que reúne funções em uma unidade central de processamento (*CPU*) em um circuito integrado. A *CPU* recebe informações digitais de entrada, processa com base nos comandos gravados em sua memória e provê os resultados como saída. Adicionalmente, há a possibilidade de conexão desse componente a outros blocos do sistema [6].

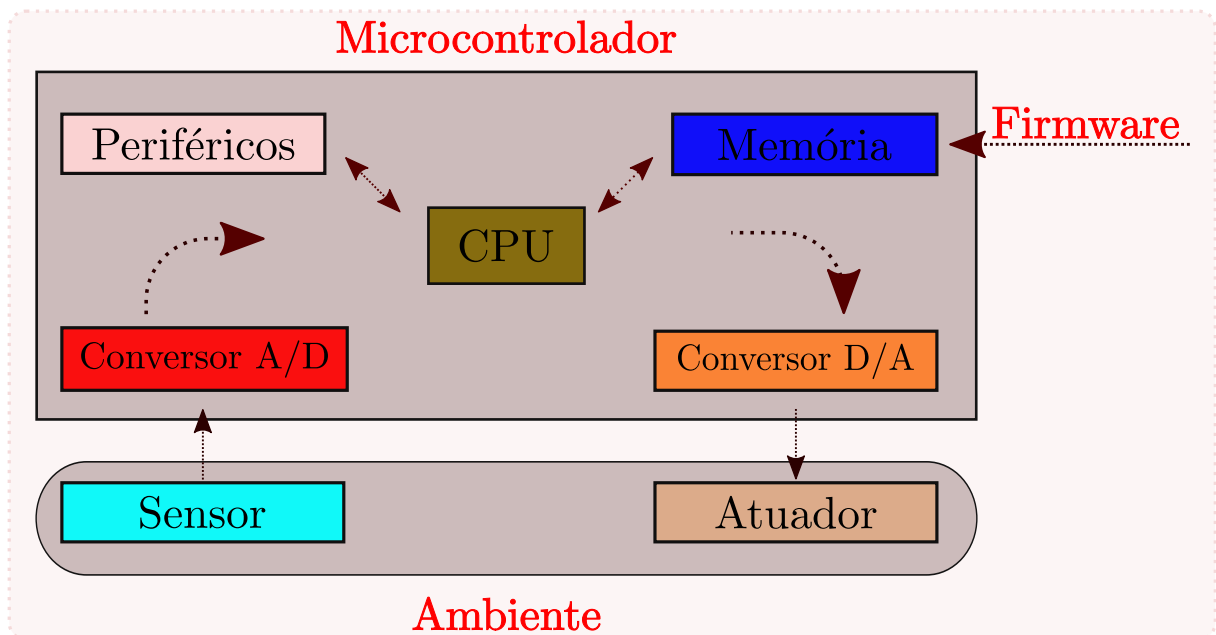
Neste contexto, temos os microcomputadores de uso múltiplo, como *PCs*, são computadores que realizam uma infinidade de tarefas com base em *softwares*. E os computadores dedicados, que executam tarefas específicas, como operar o sistema de freios de um automóvel, operar o forno de micro-ondas, entre outros. Esses, não podem ser programados pelo usuário final, eles apenas cumprem a rotina de controle para a qual foram criados e são, por isso, chamados de microcontroladores. São chamados também de controladores embutidos ou embarcados, devido ao ecossistema maior no qual estão inseridos.

Os computadores tradicionais possuem sistemas operacionais e *softwares* para diversas funcionalidades, em contraste os sistemas embarcados são projetados para algum tipo de propósito específico, como os citados previamente. Desta forma, não existe a maleabilidade de *software* ou *hardware* para a execução de outras tarefas nas quais não foram projetados previamente. A única exceção ocorre quando há atualizações para novas versões, melhorando e corrigindo problemas nesse sistema.

Assim, um sistema embarcado tem a finalidade de interagir continuamente via sensores e atuadores ao ambiente a sua volta lendo sinais externos dos dispositivos de

captação, realizando os processos com as atividades que deverão ser cumpridas e enviando os resultados desejados aos equipamentos de execução. As informações coletadas, podem ser transmitidas para outros dispositivos, como para um computador via comunicação serial [6]. A fig.(1) ilustra um exemplo de um diagrama simples de um sistema embarcado coletando informações externas.

Figura 1 – Diagrama simples de um sistema embarcado contendo um microcontrolador monitorando o ambiente.

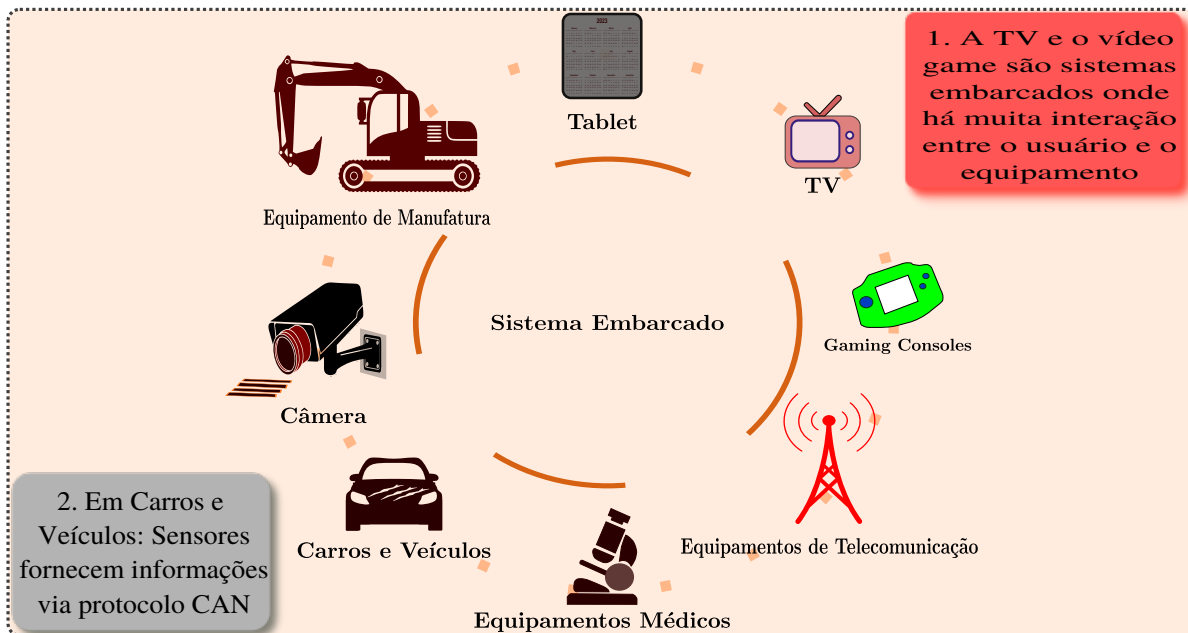


Fonte: Adaptado de [7].

Com o avanço da tecnologia, os sistemas embarcados estão paulatinamente mais frequentes em nosso dia a dia e desempenham um papel crucial em nosso mundo moderno. De fato, eles estão presentes em uma vasta gama de indústrias e aplicações tecnológicas, como por exemplo, no setor de automação industrial e robótica. A fig.(2) apresenta um conjunto de equipamentos e setores industriais que utilizam esse sistema de computação especializada.

Em geral, um sistema embarcado é projetado contendo um microcontrolador e um *software*, chamado de *firmware*. Em seu desenvolvimento busca-se otimizar as dimensões físicas, aperfeiçoar o consumo de energia elétrica, aumentar a resistência e a durabilidade do componente. Com isso, garante-se a eficiência e a confiabilidade dos dispositivos onde eles serão inseridos.

Figura 2 – Aplicações tecnológicas que utilizam sistemas embarcados.



Fonte: Adaptado da *webpage*: [Embedded Systems](#). Acesso: 02 de novembro de 2023.

1.1.1 Firmware, Microprocessador & Microcontrolador

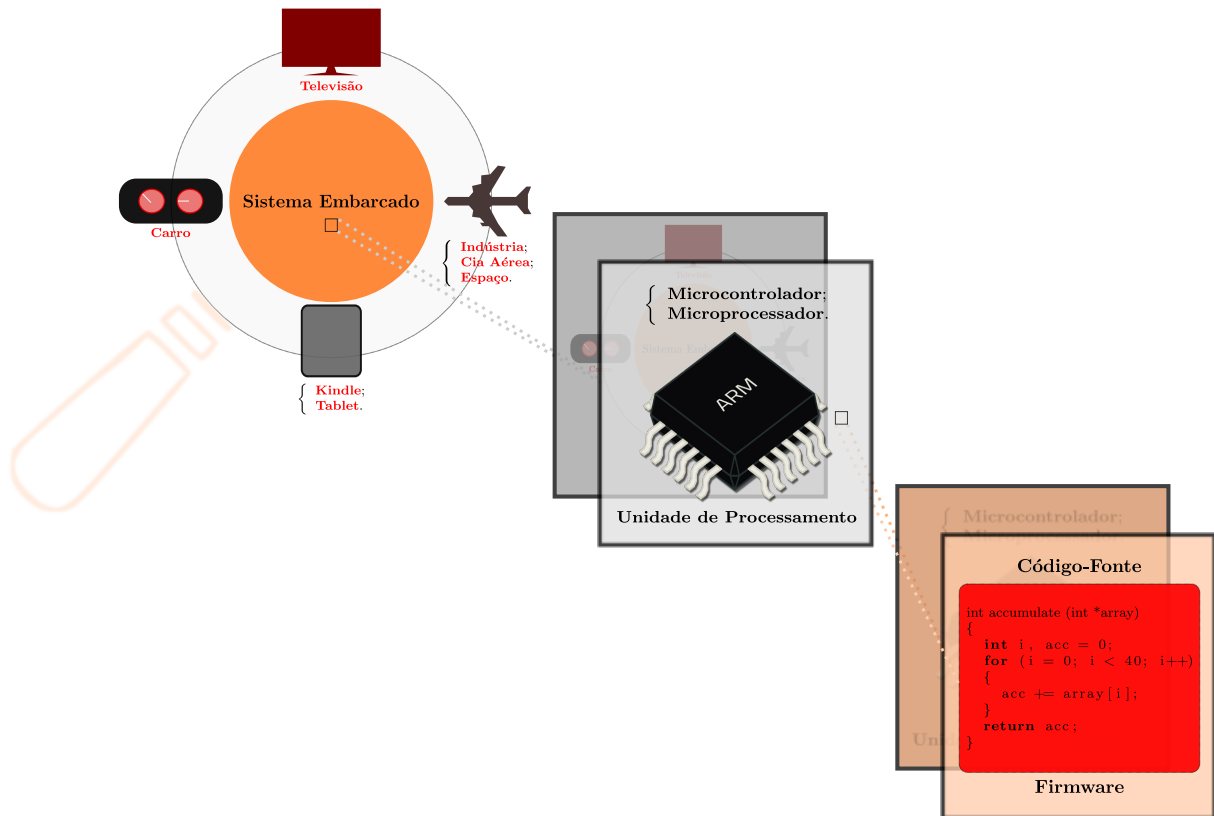
Em um sistema embarcado há uma unidade capaz de processar as informações via *software*. Ela é programada diretamente no *hardware* e fornece o conjunto de instruções computacionais necessárias para a inicialização e execução dos dispositivos eletrônicos. Este *software* embarcado é chamado de *firmware*. A fig.(3) apresenta alguns elementos de um sistema embarcado.

O *firmware* é um elemento crítico para a execução adequada das atividades nos dispositivos eletrônicos, ele fornece as instruções necessárias para o funcionamento do sistema. Por meio de sua configuração ele assegura que as funções básicas dos equipamentos sejam imutáveis e inalteradas[†]. Em geral, ele é desenvolvido para continuar estável e com a mesma essência durante o tempo de vida útil do dispositivo, diferente de um *software* convencional. Para isso, o *firmware* é armazenado no chip de memória do *hardware* nas memórias programável somente de leitura (*PROM*) e na programável apagável somente de leitura (*EPROM*).

De maneira complementar, temos os microprocessadores e os microcontroladores que estão inseridos em dispositivos digitais. Um microprocessador é um circuito integrado (CI)

[†] O *firmware* pode ser atualizado para a correção de erros e melhorias no equipamento, mas as funções básicas do dispositivo se mantêm as mesmas.

Figura 3 – Ecossistema de tecnologias digitais que utilizam sistemas embarcados. O microcontrolador e/ou o microprocessador é utilizado para processar as informações dos dispositivos eletrônicos. O *firmware* realiza as instruções computacionais via código escrito em alguma linguagem de programação.



Fonte: Adaptado da *webpage*: [Embedded Systems](#) e de [7]. Acesso: 01 de novembro de 2023.

que efetua cálculos lógicos ou aritméticos. Esse *chip* comporta portas lógicas que executam comandos via *bits*. Ele utiliza equipamentos periféricos para o seu funcionamento, como memórias, interfaces de comunicação, entre outros, desenvolvidos em placas de circuito impresso (*PCB* ou *PCI*) que dialogam entre si. São utilizados para aplicações que requerem maior poder de processamento. De maneira distinta, nos microcontroladores os periféricos são fabricados em um único encapsulamento, assim este dispositivo é reservado para aplicações que executam atividades mais modestas [6, 8]. A fig.(4) apresenta um exemplo de cada dispositivo mencionado previamente.

O microprocessador modelo Apple M1 encontra-se presente no MacBook Pro e MacBook Air por exemplo. Já o microcontrolador ATmega1609[‡] pertence à família de microcontroladores AVR[§] da *Microchip Technology*. Geralmente, esse grupo de dispositivos

[‡] Para informações adicionais, consultar o *datasheet* via a *webpage*: [ATmega808/809/1608/1609](#). Acesso: 09 de novembro de 2023.

[§] Microcontrolador com base na arquitetura *RISC*. Para informações adicionais, consultar o livro [8].

Figura 4 – Em (a) tem-se o microprocessador modelo **Apple M1**. Em (b) tem-se o microcontrolador modelo **ATmega1609** de 8 *bits*.



Fonte: Adaptado da *webpage*: [Comparison of Microprocessor and Integrated Circuit e Microchip Atmega1609](#). Acesso: 05 de novembro de 2023.

são utilizados em diversas aplicações.

1.2 Sensoriamento Remoto via Dados

De modo atual, a tecnologia é imprescindível para a execução de diversas atividades profissionais e realiza a mediação entre as relações humanas com as máquinas. É a nomeada era da hiperconectividade. Face a essa realidade, essa interação gera uma vasta quantidade de dados que são úteis para avaliar informações de diferentes processos. Essa comunicação analisada e processada é uma poderosa ferramenta para nortear estratégias, decodificar problemas e propor soluções valiosas ao ambiente no qual se está inserido. Diante desta caracterização, percebe-se que os dados são essenciais para a tomada de decisões e são vistos como a matéria-prima do conhecimento moderno impulsionando novas áreas como inteligência artificial, ciência de dados, e assim por diante.

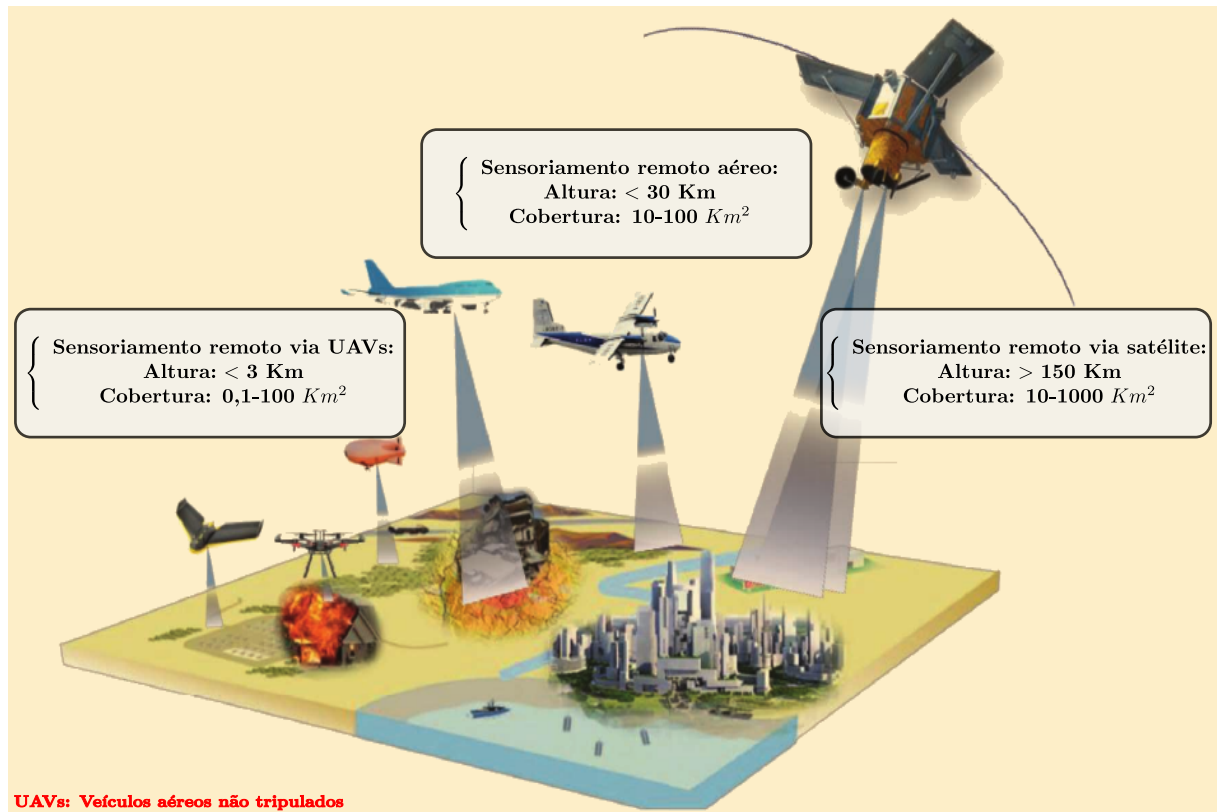
As implicações tecnológicas são imensas. A captação de dados de maneira eficiente, clara e apropriada beneficia vários setores e têm impulsionado campos como a agricultura de precisão, mapeamento e planejamento urbano, gestão ambiental, entre outros.

No âmbito da coleta de dados aérea, testemunhamos várias transformações. Houve um vasto desenvolvimento no uso de VANTs em atividades civis, como fotogrametria[¶], aquisição de geoinformações e sensoriamento remoto [1,9]. Os ganhos no uso de veículos aéreos não tripulados frente a aeronaves tripuladas e satélites são significativos. Destacam-se, por exemplo, o baixo custo, a flexibilidade e a facilidade de operação. A fig.(5) apresenta

¶ Utilização da fotografia para os levantamentos topográficos.

esses equipamentos que são capazes de realizar a captura de informações externas.

Figura 5 – Sensoriamento remoto aéreo realizado por diferentes plataformas, tais como satélite, veículos aéreos tripulados e não tripulados.



Fonte: Adaptado de [1].

Portanto, nota-se que soluções inovadoras e acessíveis são necessárias para suprir a demanda crescente por coleta, processamento e análise de dados.

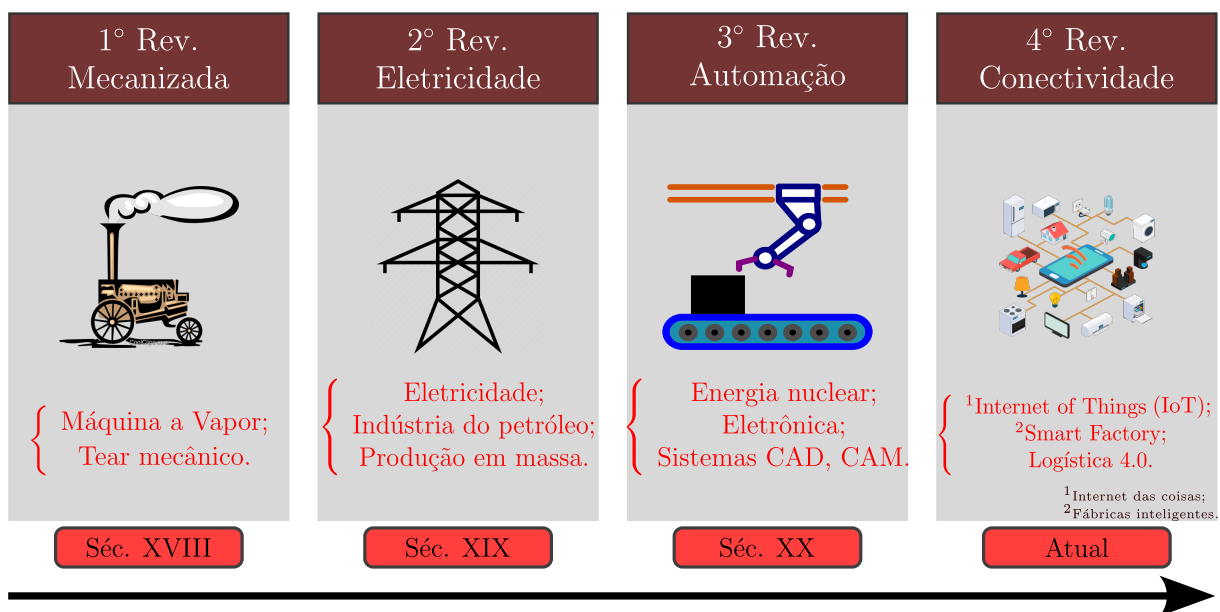
Em alinhamento com o que foi dito, este trabalho investiga os dados coletados do sensor **AHT10** embarcado no nano quadricóptero modelo *Crazyflie 2.1*. Esta tarefa possibilita que a aquisição da informação da unidade sensorial seja executada de maneira dinâmica e flexível quando comparada a sistemas de detecção fixos. Desta forma, os resultados obtidos provê informações úteis sobre a temperatura e a umidade do ambiente ao qual foi feita a captura dos dados.

1.3 Indústria 4.0

Ao longo dos séculos a industrialização e os seus desdobramentos exerceram um papel crucial nas transformações vividas nas sociedades. A revolução industrial dividida

em fases, marca os avanços e aperfeiçoamentos técnicos de produção que culminaram em inovações e maior produtividade aos países inseridos nesse processo. A fig.(6) apresenta um resumo temporal dessa perspectiva de desenvolvimento industrial.

Figura 6 – Panorama do processo de industrialização ao longo dos séculos.



Fonte: Adaptado de [10].

A primeira revolução industrial datada do final do século XVIII sinaliza a passagem de uma sociedade baseada em atividades artesanais e na produção de mercadorias em pequenas escalas para atividades de fabricação mecanizadas. A partir desse momento, novos níveis de industrialização foram atingidos, os quais possibilitaram o surgimento de novas indústrias, melhoria tecnológica e alteração na organização do trabalho.

O incremento vivenciado da capacidade produtiva provocou mudanças na economia e na vida cotidiana das pessoas. Como consequência, houve modificações nos hábitos e no modelo de consumo da sociedade, mudou-se a interação entre o homem e a natureza, impulsionou melhoria em diversos campos do conhecimento e assim sucessivamente. Desta forma, em cada fase de industrialização, consolidou-se novas maneiras de organização da sociedade oriundo do desenvolvimento alcançado até então.

Nas últimas décadas, devido ao avanço de tecnologias de informação (TI) incorporada ao ecossistema fabril permitiu que as organizações melhorassem seus processos produtivos e estratégicos. Essa visão abrangente da cadeia de valor^{||} da companhia possibi-

.....
^{||} Método utilizado para gerenciar processos e confeccionar quais serão as vantagens competitivas de uma corporação.

litou fornecer melhores serviços aos clientes, trazendo vantagens a organização [10]. As atividades desenvolvidas ficaram mais interligadas, alavancando o desempenho da empresa.

O progresso tecnológico impulsionou a produtividade industrial, diminuindo as despesas de produção fornecendo respostas mais eficientes para atender os clientes com rapidez, qualidade e melhores preços. Neste contexto, a produção apoiada na Internet melhora a interação entre fabricantes, clientes e fornecedores e possibilita o surgimento de novos modelos de negócios.

Essa forma de indústria está sendo referida sob a nomenclatura de indústria 4.0. Nesta, procura-se promover a digitalização das atividades industriais na busca de produtos mais complexos, de maior qualidade, personalizáveis, entre outros. Essa evolução dos sistemas industriais gera desafios e oportunidades aos agentes inseridos nesse contexto. Por isso tudo, esse novo modelo de desenvolvimento e administração das organizações despertam o interesse de pesquisadores e países visando soluções melhores que possam atender as necessidades e anseios existentes da sociedade.

1.4 Estado da Arte

Os avanços em *hardware*, *software* e tecnologias correlatas alavancaram a utilização de sistemas inteligentes em diferentes cenários. O aperfeiçoamento de áreas como comunicação, conectividade, autonomia, eficiência energética, entre outras, fizeram com que áreas dinâmicas como sensoriamento remoto realizado via veículos aéreos não tripulados fosse beneficiada. Nos últimos anos, trabalhos disponíveis na literatura apresentaram o panorama e as tendências correlatas a utilização de drones para automatizar diversos processos [1, 11, 12].

Neste cenário, destacam-se, por exemplo, propostas de aplicações industriais como a identificação de fonte de gás via nano quadricóptero [2, 13], levantamento topográfico via drones [14], mapeamento de focos de reprodução de *Aedes Aegypti* [15], trabalhos com aplicações em geotecnologias [4, 16], entre outros.

Essas aplicações evidenciam o potencial da utilização de VANTs em diferentes contextos sendo um grande aliado dos pesquisadores no estudo de diversas áreas, provendo dados de qualidade para a coleta de informações e pesquisa de maneira mais rápida, precisa e eficiente.

Adicionalmente, o advento da chamada indústria 4.0 tem contribuído para o desenvolvimento de áreas como sistemas embarcados, onde busca-se sistemas dedicados para a realização de tarefas específicas. Neste sentido, campos como internet das coisas (IoT), aprendizado de máquina, etc, estão evoluindo ao passo que novas tecnologias surgem e as exigências do mercado se alteram.

1.5 Justificativas

Em alinhamento com o que foi discutido nas seções anteriores nota-se o grande interesse, a importância e a extensão do uso de VANTs em diferentes esferas tecnológicas. Dada a maneira mais eficiente, rápida, descentralizada e precisa essa plataforma vem revolucionando diversas áreas, fornecendo informações mais detalhadas e melhores contribuindo para a tomada de decisão mais assertiva dos atores inseridos nesse contexto.

Percebe-se, desta maneira, que a utilização de drones melhora a rotina industrial e contribui para uma vasta gama de aplicações tecnológicas. De fato, a popularização do uso desse equipamento coopera para o avanço tecnológico gerando novas possibilidades e inovações para diversos segmentos.

A utilização de VANTs no setor de sensoriamento remoto provê benefícios quando comparado a coleta de dados via métodos tradicionais. A acessibilidade, flexibilidade, redução de riscos e segurança a integridade humana, entre outros, desse aparelho torna-o ideal para diversas operações e tarefas cotidianas.

Além disso, de acordo com as inovações exigidas pela indústria 4.0 e com trabalhos acessíveis na literatura o setor industrial pode ser significativamente beneficiado. Por exemplo, os dados coletados por VANTs podem compor bancos de dados no sistema de monitoramento industrial. Com isso, fábricas inteligentes podem ser criadas, onde *softwares* decidiriam o momento mais adequado para o acionamento do drone para a construção de uma rotina de manutenção preditiva. Desta forma, as inspeções na indústria podem ser feitas de maneira mais eficiente e precisa atendendo a requisitos preestabelecidos.

Portanto, é compreensível e justificável a ascensão no interesse do estudo dessas aeronaves. Neste sentido, este trabalho busca apresentar uma proposta de aplicação tecnológica que abrange o tema. Para isso, será desenvolvido uma solução utilizando um nano quadricóptero embarcado com um sensor de temperatura e umidade coletando informações de um ambiente *indoor* de interesse. Adicionalmente, buscar-se-á contribuir para a literatura referente ao tema da indústria 4.0, onde os dados coletados da unidade sensorial serão disponibilizados na nuvem.

1.6 Objetivos

O objetivo geral desta pesquisa é, de acordo com o que foi apresentado anteriormente, desenvolver uma solução de sistema embarcado utilizando um nano quadricóptero, de forma que seja possível coletar os dados de temperatura e umidade do ambiente de interesse.

1.6.1 Objetivos Específicos

Para o desenvolvimento da pesquisa serão seguidos os seguintes objetivos específicos:

- Alteração do *hardware* e *firmware* da aeronave *Crazyflie 2.1* para a instalação do sensor de temperatura e umidade **AHT10** encarregado da coleta de dados do ambiente de interesse;
- Implementação do sistema supervisor responsável por acionar a aeronave;
- Disponibilização dos dados coletados na nuvem;
- Realização de testes em ambientes controlados para verificar a acurácia do sistema desenvolvido.

1.7 Organização da Monografia

Nesta monografia investigamos a implementação de uma unidade sensorial para a coleta de dados de temperatura e umidade embarcado em um nano quadricóptero. Para isso, foram efetuadas alterações no *hardware* e no *software* da aeronave para permitir a aquisição das informações de interesse. Esse processo será exibido da seguinte maneira:

- **Capítulo 2:** Mostra o protocolo de comunicação I^2C e as características do sensor de temperatura e umidade **AHT10**.
- **Capítulo 3:** Apresenta o modelo da aeronave utilizada neste trabalho, bem como as suas características e especificidades.
- **Capítulo 4:** Exibe a metodologia desenvolvida no trabalho.
- **Capítulo 5:** Mostra os resultados obtidos.
- **Capítulo 6:** Apresenta a conclusão e considerações finais do projeto.

GERENCIAMENTO DE INFORMAÇÕES EM DISPOSITIVOS DIGITAIS

Quando pensamos em tecnologias de tempo real uma função necessária é o monitoramento. Consiste na análise dos fenômenos do sistema, possibilitando a validação de requisitos em tempo de execução. Nestes, a comunicação entre o microcontrolador e os diferentes tipos de dispositivos periféricos podem ser realizadas por uma ampla variedade de tecnologias, como redes de comunicação ou via algum tipo de protocolo digital.

Essa tecnologia é concebida para executar dados e a reagir a eventos dentro de um intervalo de tempo preestabelecido. Para isso, o sistema é desenvolvido para realizar tarefas específicas onde são fundamentais garantir restrição de tempo, preempção**, comunicação em tempo real, entre outros. Dada a essa robustez, estes sistemas são empregados em setores tecnológicos onde a confiabilidade é essencial, tais como controle de tráfego aéreo, sistemas de controle industrial, sistemas de automação de fábricas e assim por diante.

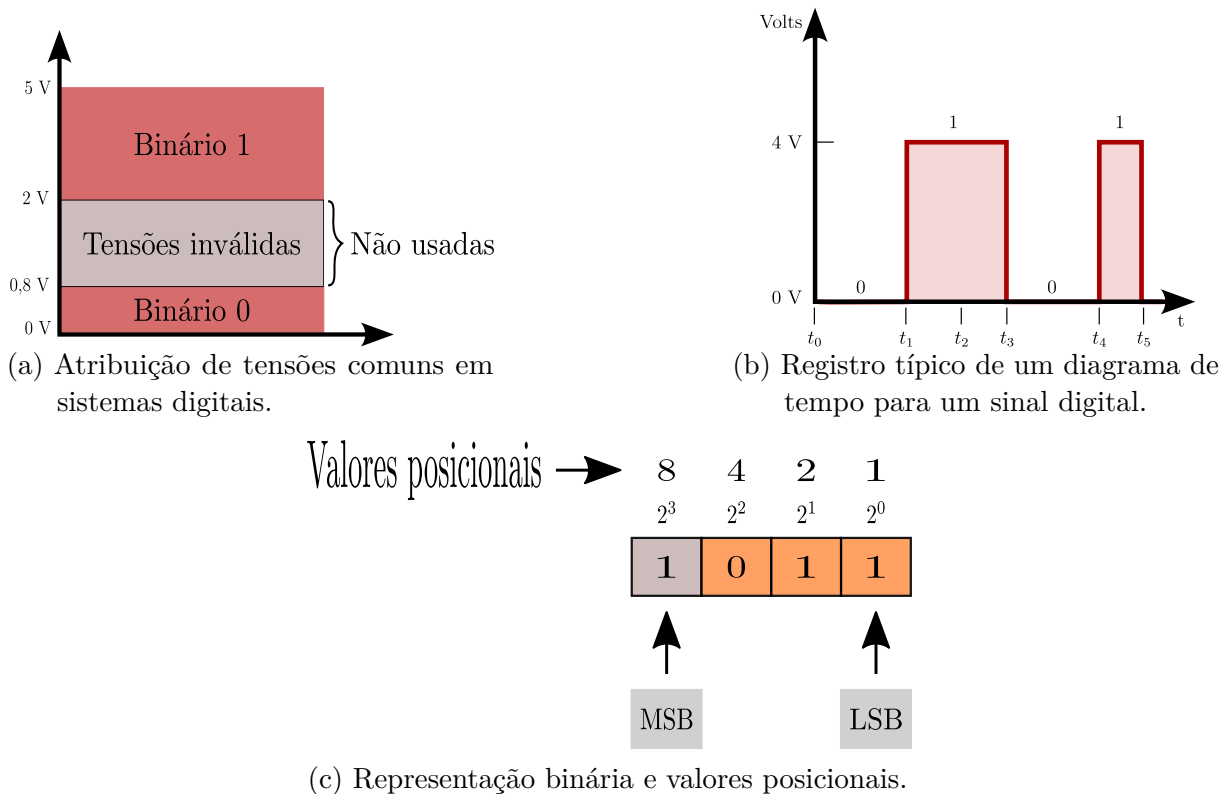
Neste meio, uma possibilidade para a transferência de dados eletronicamente ocorre de duas maneiras. Sendo elas, em série ou em paralelo. Os *bits* são enviados em conjunto e cada *bit* utiliza uma via de conexão na transmissão paralela, em contraste, a transmissão em série utiliza apenas uma via de comunicação, na qual o *bit* é enviado em série, em outras palavras, um de cada vez.

A mensagem é enviada na configuração binária e, comumente, é descrita por meio de tensões no circuito transmissor e receptor, conectados a saída e a entrada do sistema respectivamente. A fig.(7) ilustra um exemplo de sinal amostrado digitalmente, onde a fig.(7a) mostra as tensões de acordo com o respectivo valor binário, a fig.(7b) apresenta o

.....
** Na computação, refere-se à interrupção temporária de tarefas realizada pelo escalonador preemptivo, onde a execução de uma tarefa privilegiada de maior prioridade é executada primeiro. Para maiores informações, consultar o livro [17].

diagrama de tempo e a fig.(7c) ilustra uma representação binária.

Figura 7 – Representações digitais de “1s” e “0s”.



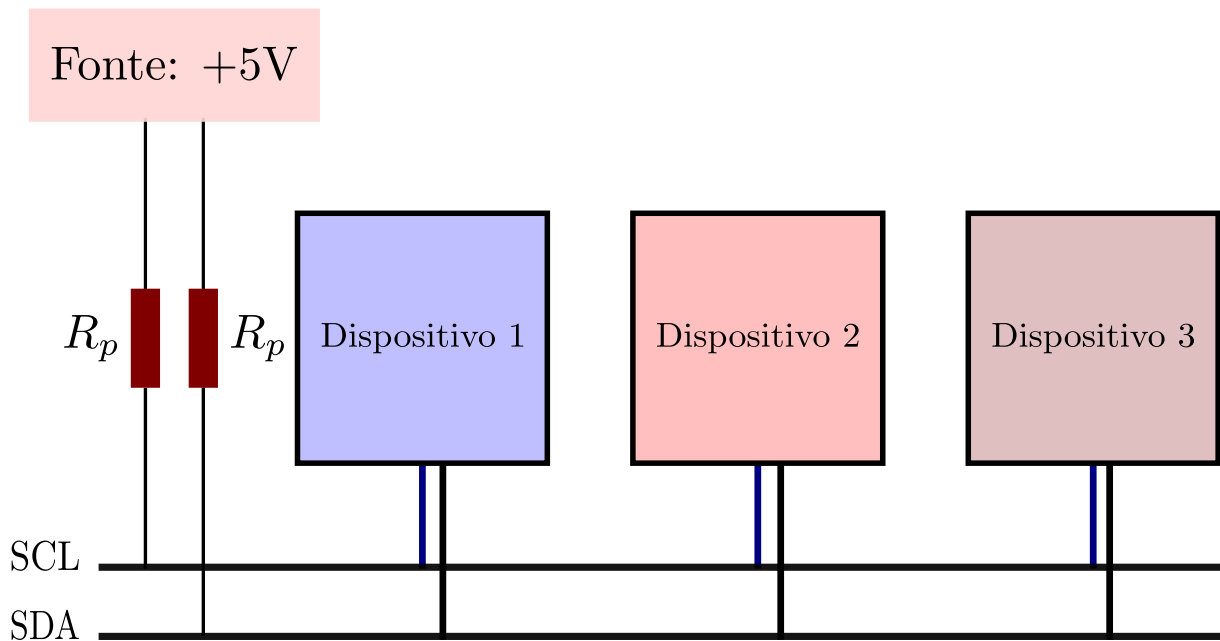
Fonte: Adaptado de [6].

2.1 Protocolo de Comunicação I^2C

O protocolo de comunicação I^2C é uma maneira versátil, rápida e simples para a transmissão serial de dados amplamente utilizada em diversos dispositivos eletrônicos, como *displays*, sensores, computadores, microcontroladores, conexão de periféricos, entre outros. Por intermédio de um barramento compartilhado os diferentes tipos de equipamentos digitais comunicam-se e compartilham informações. A fig.(8) ilustra um exemplo de barramento I^2C .

Este método de comunicação e compartilhamento de dados foi criado pela *Philips Semiconductor*, atualmente *NXP Semiconductor* em 1982. Para o funcionamento do protocolo, a unidade de controle e a de periféricos precisam possuir implementação e suporte I^2C , seja via *hardware* ou *software*.

Figura 8 – Exemplo de barramento compartilhado que faz uso do protocolo de comunicação I^2C .



Fonte: Adaptado de [18].

O barramento de comunicação I^2C possui dois fios com resistências do tipo *pull up*, as vias **SDA** (*serial data line*) e **SCL** (*serial clock line*). A alimentação do sistema geralmente é feita utilizando fontes V_{DD} de 3,3 V ou 5 V. A transmissão de informação é enviada em pacotes de *bytes* para um único endereço de cada dispositivo alvo.

O protocolo de comunicação I^2C funciona com diferentes modos de velocidade, como pode ser visualizado na Tabela (1) a seguir.

Tabela 1 – Taxa de transferência na comunicação para o protocolo I^2C .

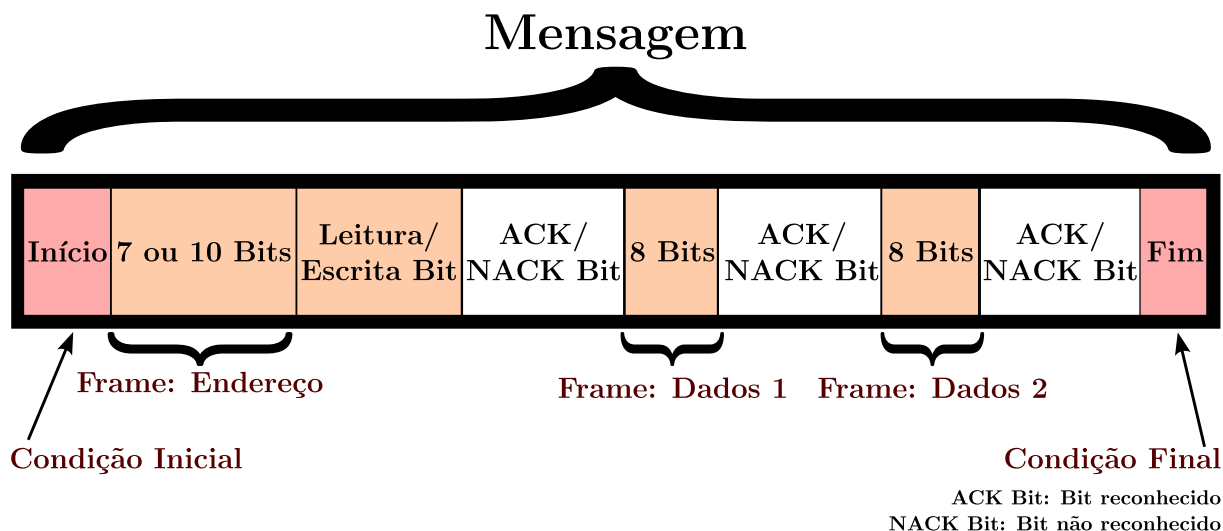
Método I^2C	Taxa Máxima de Transferência do <i>Bit</i>
Modo Padrão	100 kbps
Modo Rápido	400 kbps
Modo Rápido <i>Plus</i>	1 Mbps
Modo Alta Velocidade	3,4 Mbps
Modo Ultrarrápido	5 Mbps

Fonte: Extraído de [19].

Quando trabalhamos com o protocolo de comunicação I^2C encontramos limitações na quantidade de componentes que se conectam ao barramento de compartilhamento de

dados. Tais restrições podem ser provenientes do tamanho do endereçamento no dispositivo, que podem ser de 7 *bits*, 10 *bits* e até 16 *bits*. Adicionalmente, há restrição de espaço na via de comunicação de informação, visto que ela é reduzida para garantir que o meio funcione da maneira adequada. A fig.(9) apresenta um *frame* de dados desse protocolo.

Figura 9 – *Frame* de dados do protocolo *I²C*.



Fonte: Adaptado de [20].

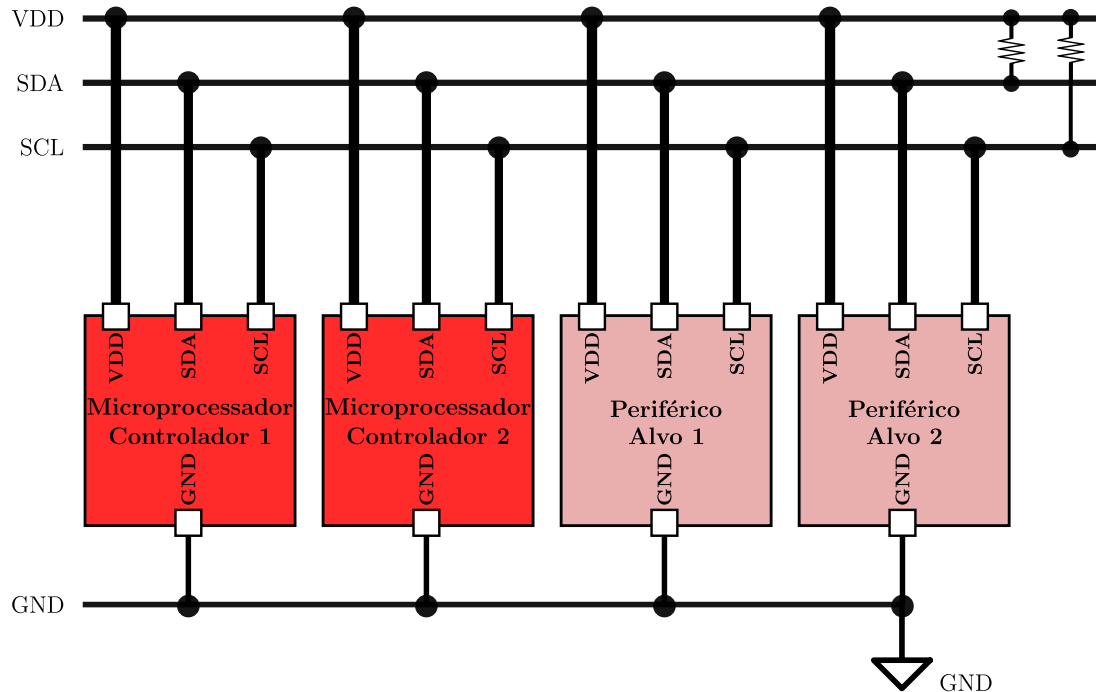
A comunicação *I²C* é bidirecional, *half-duplex*^{**}. Em essência, este protocolo utiliza dois tipos de equipamentos. Um deles é o responsável por arbitrar a via de comunicação de dados, enquanto o outro é o alvo da informação. No jargão computacional, são chamados de *master* (mestre) e *slave* (escravo)^{††}. O mestre coordena todos os periféricos conectados ao barramento, ou seja, os escravos. A via **SCL** é encarregada do controle de tempo (*clock*) do barramento e a via **SDA** é a responsável pela transmissão dos dados, como pode ser visualizado na fig.(10).

Enquanto a transmissão da informação ainda não foi inicializada o barramento *I²C* é mantido com o valor digital alto para as duas vias de comunicação. Iniciado o envio da mensagem, **SDA** muda o estado da via para o valor digital baixo, isto é feito pelo controlador do barramento. A fig.(11) ilustra esse comportamento. A escrita da informação

^{**} Os dois dispositivos conectados a via, o controlador e o alvo, podem transmitir e receber as informações. Entretanto, o envio dos dados é realizado em uma única direção por vez.

^{††} Essa nomenclatura é utilizada no jargão computacional, em resumo ela descreve os dispositivos que controlam o envio e o recebimento dos dados, denominado de mestre. Os demais equipamentos conectados na via de transmissão da informação são chamados de escravo, pois são os elementos alvo da comunicação.

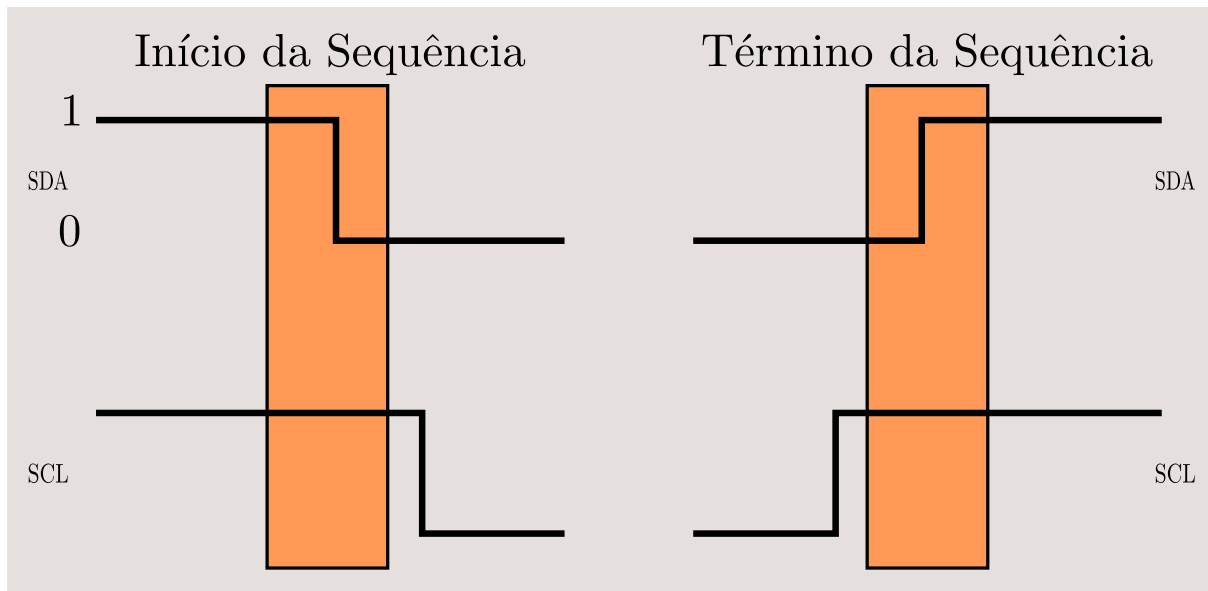
Figura 10 – Exemplo de barramento que utiliza o protocolo de comunicação I^2C . Na via de dados existem dois dispositivos controladores da via e dois equipamentos alvos da transmissão de informação. A comunicação é bidirecional, *half-duplex*, tanto os controladores quanto os alvos podem enviar e receber os dados.



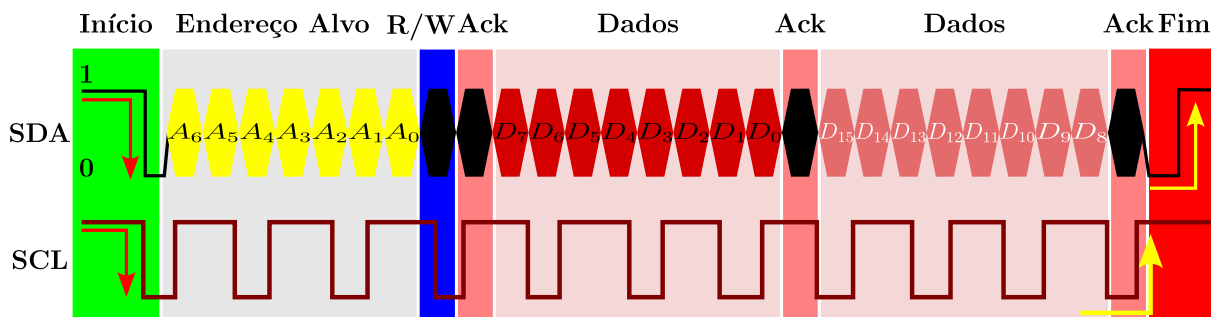
Fonte: Adaptado de [19].

no barramento é realizada com a pulsação de **SCL**. Cada pulso realizado é lido como um *bit* por **SDA**, este *bit* começa do valor mais alto transmitido **MSB** (*most significant bit*), fig.(7c).

Quando **SDA** é trazido para o nível digital baixo, o endereço do dispositivo ao qual o controlador quer se comunicar é escrito e o equipamento conectado no barramento responde com um **ACK** (*acknowledge*) que é um pulso de sinal na via **SCL**. Feito isto, o envio da informação é inicializado. O controlador do barramento então envia o endereço da comunicação para o registrador do dispositivo alvo que ele deseja ler ou escrever a mensagem. A fig.(12) exemplifica esse comportamento.

Figura 11 – Envio de mensagem no protocolo de comunicação I^2C .

Fonte: Adaptado de [18].

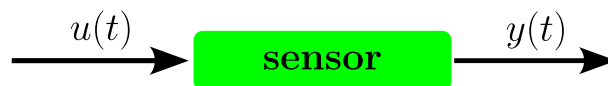
Figura 12 – Panorama dos *frames* utilizados no protocolo de comunicação I^2C .

Fonte: Adaptado de [21].

2.2 Unidade Sensorial AHT10 de Temperatura e Umidade

Uma análise simplificada de um sensor pode ser feita com base em um sistema que possui uma entrada, $u(t)$, e uma saída, $y(t)$, dependente do tempo. O mensurando^{††} é a entidade que fornece a informação ao sensor, ou seja, a entrada do sistema. A saída é a propriedade assinalada pelo sensor, que pode ser a expansão do mercúrio devido a elevação da temperatura, números exibidos em um *display oled* e assim por diante. A saída deve caracterizar convenientemente a entrada e possuir uma ligação com ela, ou seja, $y(t) = f[u(t)]$ [22]. A fig.(13) ilustra essa representação.

Figura 13 – Ilustração de um sensor como um sistema de uma única variável.



Fonte: Adaptado de [22].

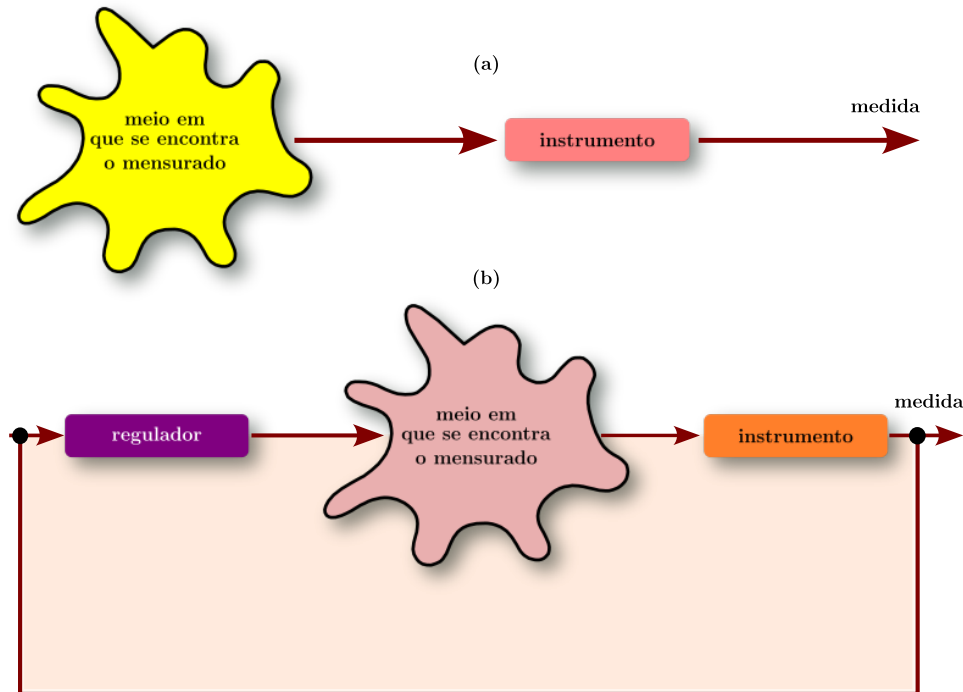
Ao reagir a um estímulo físico, químico ou biológico um sensor emite um determinado sinal que pode ser transformado em alguma propriedade “palpável” de medição. Desta maneira, podemos interpretar a entidade física coletada observando e monitorando algum fenômeno mensurável do sistema de interesse. Dispositivos como circuitos, cabos, visores, equações, programas de computador, etc, compõem um sistema de medição, onde busca-se investigar e coletar os dados do mensurando.

Em um sistema de medição ao medirmos alguma grandeza para investigar o seu comportamento utilizamos instrumentos*. O uso desse equipamento é realizado de duas maneiras. Uma alternativa é a utilização do aparato de medição em malha aberta, onde a medida realizada não interage com o meio. Isso ocorre, por exemplo, em averiguação de grandezas para fins de monitoração, análise e projeto de sistemas. Por outro lado, em malha fechada, a saída medida é utilizada para tomar decisões e atuar no sistema com base no dado coletado. Por exemplo, o acompanhamento da temperatura de uma geladeira doméstica. O estado “quente” pré-ajustado se mantém constante embora possa ocorrer perturbações ao sistema. A fig.(14) exemplifica esse comportamento.

.....
^{††} Nome dado a grandeza que se deseja medir.

* Refere-se a um sistema que inclui um sensor e algum tipo de processamento na saída desse equipamento. Para informações adicionais, consultar o livro [22].

Figura 14 – Comportamento típico realizado na operação de instrumentos de medição. Em (a) temos o comportamento em malha aberta, não há o uso de realimentação. Em (b) temos o comportamento em malha fechada, onde a realimentação atua no meio.

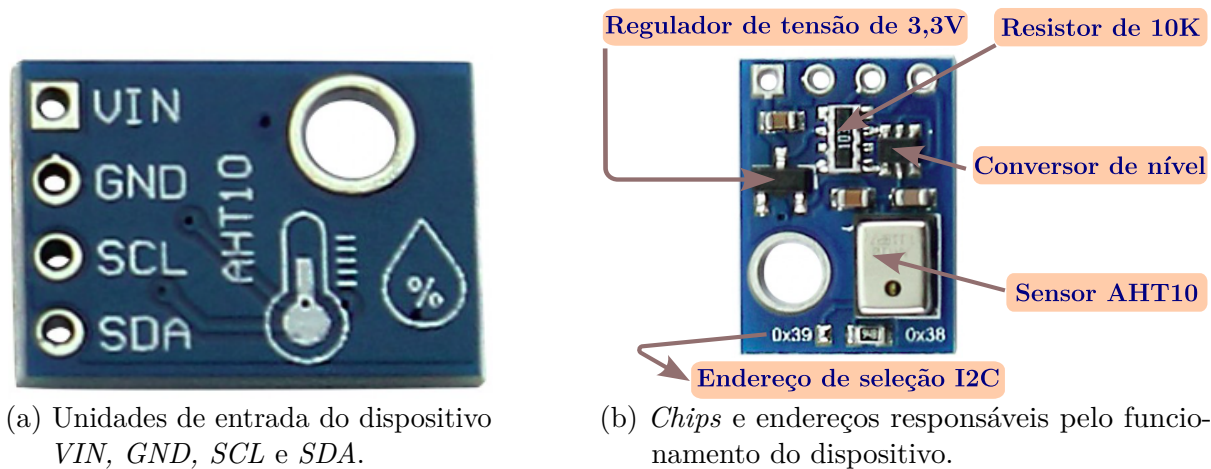


Fonte: Adaptado de [22].

A coleta de informações de forma dinâmica, flexível e versátil de um ambiente gera oportunidades de inovações para os instrumentos de medição. Uma possibilidade, são o uso de sensores embarcados em drones. Ganha-se produtividade, mobilidade e agilidade na aquisição dos dados quando comparado a coleta convencional. Neste cenário, podemos utilizar em conjunto a aeronave *Crazyflie 2.1* e o sensor **AHT10**. O microcontrolador **STM32F405** do drone é o dispositivo regulador da informação e a unidade sensorial é o equipamento alvo dos dados. Ambos os equipamentos possuem suporte e implementação I^2C e podem ser utilizados em aplicações que requerem a coleta de informações sobre a temperatura e a umidade do ambiente a sua volta. A unidade sensorial é apresentada na fig.(15).

A comunicação entre o controlador do barramento I^2C e a unidade sensorial **AHT10** é realizada utilizando-se o endereço de transmissão de dados do sensor. Esse localizador da informação possui 7 *bits* e é representado na forma binária pelo número **0111000x**. A operação de escrita de dados do controlador da via de comunicação é realizada adicionando-se um *bit* "0" no número menos significativo do endereço do dispositivo alvo. A direção da informação é da aeronave para o sensor. Em contraste, a operação de

Figura 15 – Unidade sensorial **AHT10** de temperatura e umidade.



Fonte: Elaborado pelo autor.

leitura de dados é feita quando é acrescentado um *bit* “1” no valor menos significativo do endereço do sensor. Desta forma, os dados são enviados do sensor para o microcontrolador **STM32F405** do drone. O processo de escrita e leitura de informações transforma o endereço do dispositivo alvo em um sinal de 8 *bits*. A Tabela (2) apresenta os valores de localização do sensor utilizados para compartilhar informações entre o elemento alvo e o controlador da via de dados.

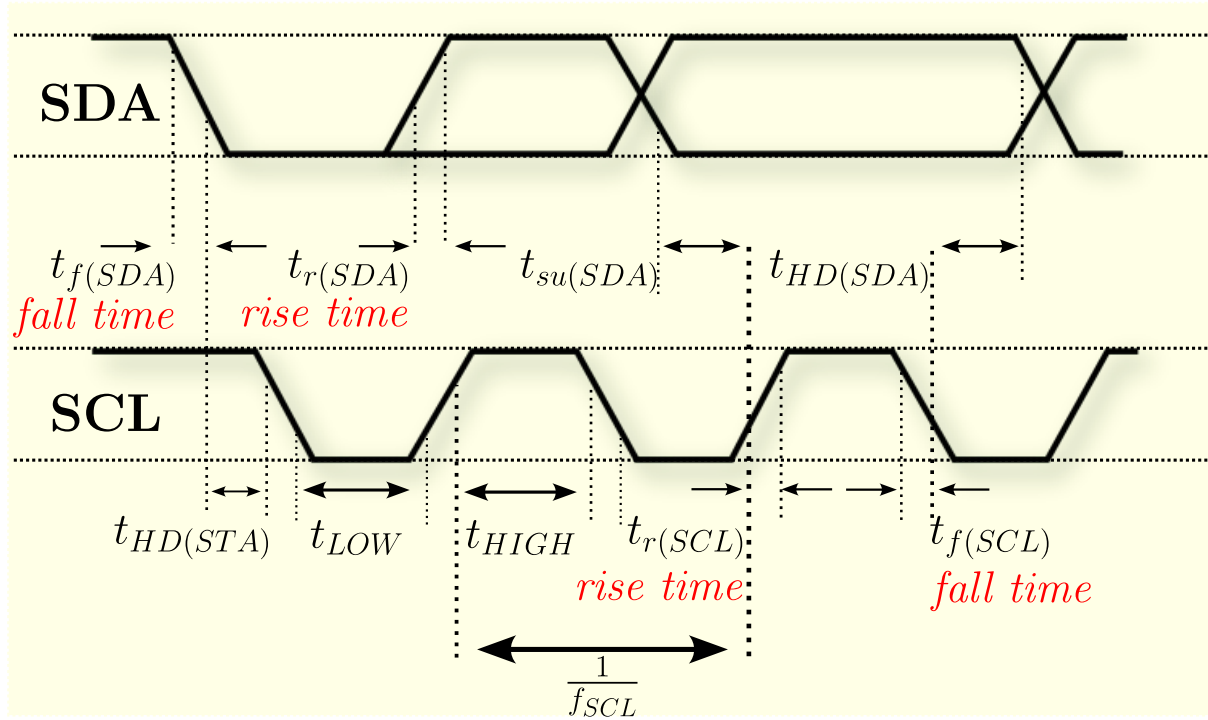
Tabela 2 – Endereços *I²C* úteis para a comunicação com o sensor **AHT10**.

Comando	Endereço <i>I²C</i> em Hexadecimal	Endereço <i>I²C</i> em Binário
Dispositivo	0x38	Escrita: 01110000
		0111000⊗↗
Inicialização	0xBE	Leitura: 01110001
		10111110
Trigger	0xAC	10101100

Fonte: Extraído de [23, 24].

Um diagrama de tempo apresenta uma interpretação visual dos eventos que ocorrem no sistema em um determinado período. Desta forma, dentro do tempo de vida da atividade é possível visualizar as mudanças de estados e as situações necessárias para a sua ocorrência. Essa representação é bastante útil em aplicações que requerem limitação de tempo. A sequência gráfica formada no compartilhamento de dados entre a unidade sensorial e o microcontrolador **STM32F405** do drone é apresentado na fig.(16). Adicionalmente, a Tabela (3) resume as informações típicas para o modo de transferência rápido dos dados no sensor **AHT10**.

Figura 16 – Diagrama de tempo digital do sensor **AHT10**. Na via **SDA** as linhas mais espessas são controladas pelo sensor, as linhas pontilhadas na posição horizontal são arbitradas pelo microcontrolador **STM32F405** da aeronave *Crazyfly* 2.1. A explicação sobre os *inputs/outputs* é apresentada na Tabela (3).



Fonte: Adaptado de [23,24].

Tabela 3 – Especificação de tempo típico na comunicação I^2C para *input/output* da unidade sensorial **AHT10**.

Parâmetro	Símbolo	Modo Típico Mín	Máx	Modo Vel. Máxima Mín	Máx	Unidade
Frequência do clock I^2C	f_{SCL}	0	100	0	400	khz
Tempo de espera: condição inicial	$t_{HD,STA}$					μs
Período <i>high</i> de SCL	t_{HIGH}	4,7		1,3		μs
Período <i>low</i> de SCL	t_{LOW}	4,0		0,6		μs
Tempo de espera de SDA	$t_{HD,DAT}$	0,09	3,45	0,02	0,9	μs
Tempo de configuração de SDA	$t_{SU,DAT}$	250		100		μs

Os tempos I^2C seriais acima são determinados por *delays* internos ao sensor. *Input*: 100ns *Output*: 200 ns.

Fonte: Extraído de [23,24].

Em uma unidade sensorial os dados coletados brutos precisam ser tratados para transformá-los em um valor que faça sentido fisicamente. No sensor **AHT10** a saída da via **SDA** fornece os dados “crus” da umidade relativa S_{RH} e da temperatura S_T . Os valores reais são obtidos via o cálculo de expressões matemáticas. Assim, a eq.(2.1) calcula a

umidade relativa e a eq.(2.2) calcula a temperatura em graus Celsius do ambiente a sua volta com base nos valores digitais do barramento de dados.

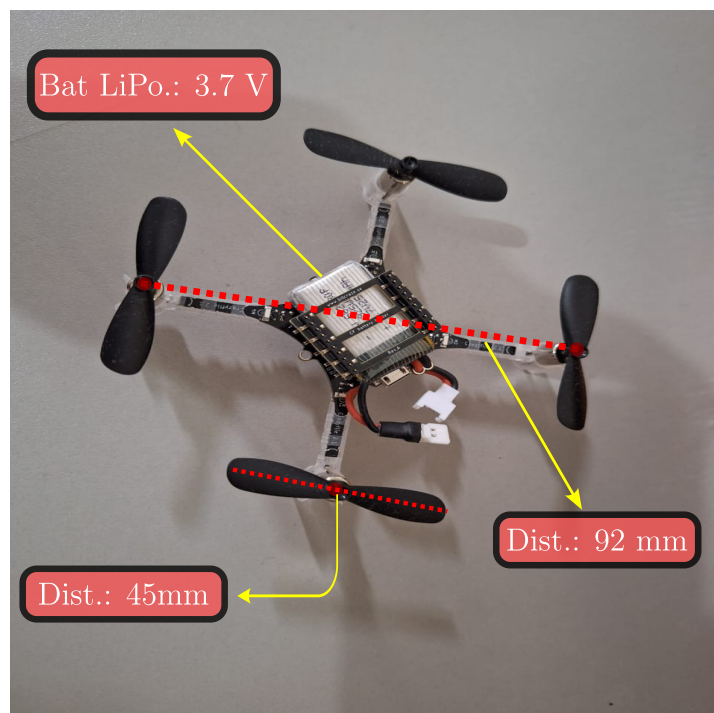
$$RH[\%] = \left(\frac{S_{RH}}{2^{20}} \right) \cdot 100\%, \quad (2.1)$$

$$T(^{\circ}C) = \left(\frac{S_T}{2^{20}} \right) \cdot 200 - 50. \quad (2.2)$$

AERONAVE *CRAZYFLIE 2.1*

A aeronave utilizada para o desenvolvimento desse projeto foi o nano quadricóptero modelo *Crazyflie 2.1*. Esse equipamento é uma pequena aeronave não tripulada que faz parte da classe de quadricópteros. A fig.(17) apresenta uma ilustração do veículo. O equipamento é fabricado pela empresa sueca *Bitcraze* e destaca-se por possuir código fonte aberto, suporte do fabricante e dimensões reduzidas. Estas características, são ideais para voos de curta distância em relação a seres humanos, voos em espaços limitados e em ambientes internos.

Figura 17 – Nano quadricóptero modelo *Crazyflie 2.1*.



Fonte: Elaborado pelo autor.

O drone possui quatro rotores para propulsão e estabilidade de voo. Têm como características essenciais o tamanho compacto e a massa reduzida, sendo possível apoiá-lo na palma da mão. Além disso, ele traz uma vasta maleabilidade em seu *software* permitindo o acesso ao *firmware* da aeronave, ao sistema de comunicação de rádio, entre outros, tornando-o adequado para diversos projetos de pesquisas e aplicações tecnológicas [2, 3].

Adicionalmente, no drone temos que o tamanho entre eixos dos motores na posição diagonal é de 92 mm, com uma massa de aproximadamente 27 gramas e aptidão de carga (*payload*) de 15 gramas. O fornecimento de energia ao veículo é feito via uma bateria de LiPo de 3,7 V (250 mAh), que provê uma autonomia de voo de aproximadamente 7 minutos ininterruptos [25].

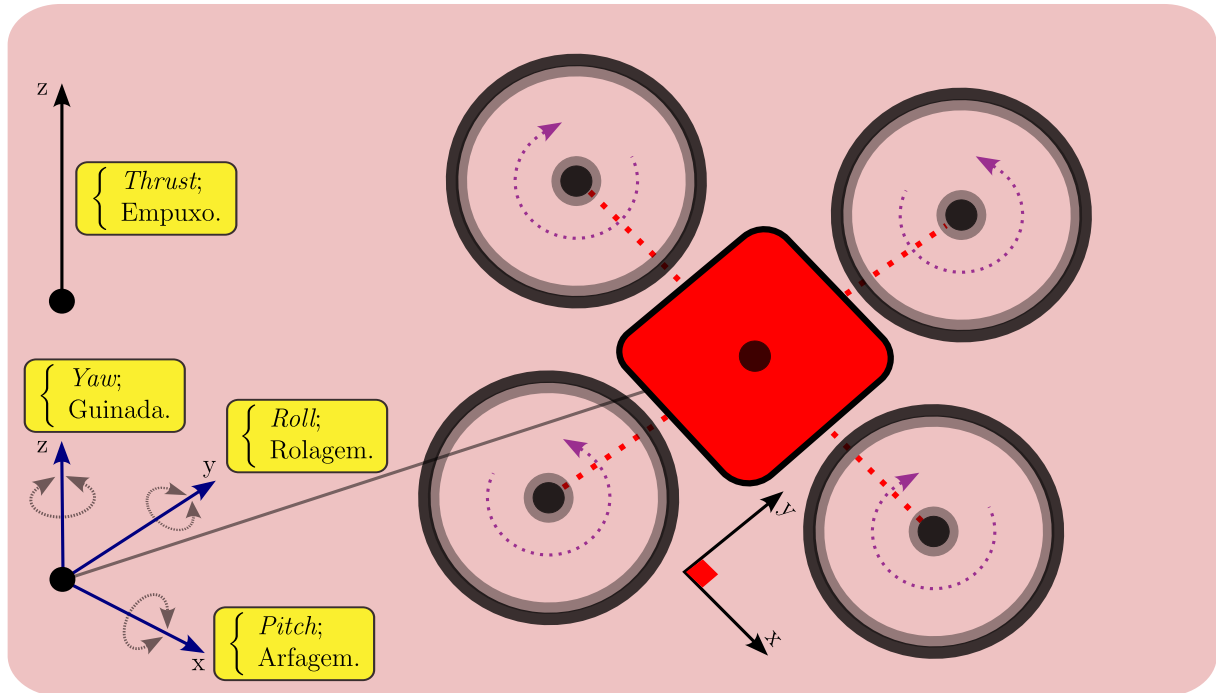
A manobrabilidade do voo na aeronave é associada a três eixos coordenados de controle. O movimento em torno do eixo lateral é chamado de arfagem (*pitch*), o eixo é uma linha imaginária horizontal que cruza o veículo lateralmente, da esquerda para a direita, essa inclinação possibilita o movimento de “gangorra”, ou seja, o drone se desloca para a frente e para trás. O deslocamento em torno do eixo vertical, é conhecido como guinada (*yaw*), esse eixo é uma linha imaginária que atravessa o drone de baixo para cima, a rotação feita pelo veículo é realizada da esquerda para a direita ou da direita para a esquerda. Esse movimento possibilita alterar a direção do voo posicionando a frente do drone em diferentes posições. O deslocamento em torno do eixo longitudinal é conhecido como rolagem (*roll*), o eixo é uma linha imaginária que atravessa o drone da parte traseira até a parte dianteira, o movimento permite que o drone se mova para a esquerda ou para a direita. Além disso, há o movimento de empuxo (*thrust*) que é responsável por adaptar a altitude ou a altura do *Crazyflie*. A fig.(18) apresenta o comportamento descrito utilizado no controle de voo do veículo.

3.1 Estrutura de Funcionamento do Drone

O *Crazyflie 2.1* utiliza dois modelos embarcados de microcontroladores. O protótipo principal é o *chip STM32F405*^{††} que trabalha primordialmente com as funções de sensoriamento, estimação dos estados, no controle e no acionamento dos motores da aeronave. E o *chip nRF51822*^{‡‡} que é utilizado para administrar a comunicação via rádio e a alimentação de energia do drone. A fig.(19) apresenta a estrutura de funcionamento da

^{††} O *chip* possui as seguintes características: *Microcontroller unit (MCU) (Cortex-M4, 168 MHz, 192 kb SRAM, 1 Mb flash)*.

^{‡‡} Esse microcontrolador tem as seguintes particularidades: *Microcontroller unit (MCU) (Cortex-M0, 32 Mhz, 16 kb SRAM, 128 kb flash)*.

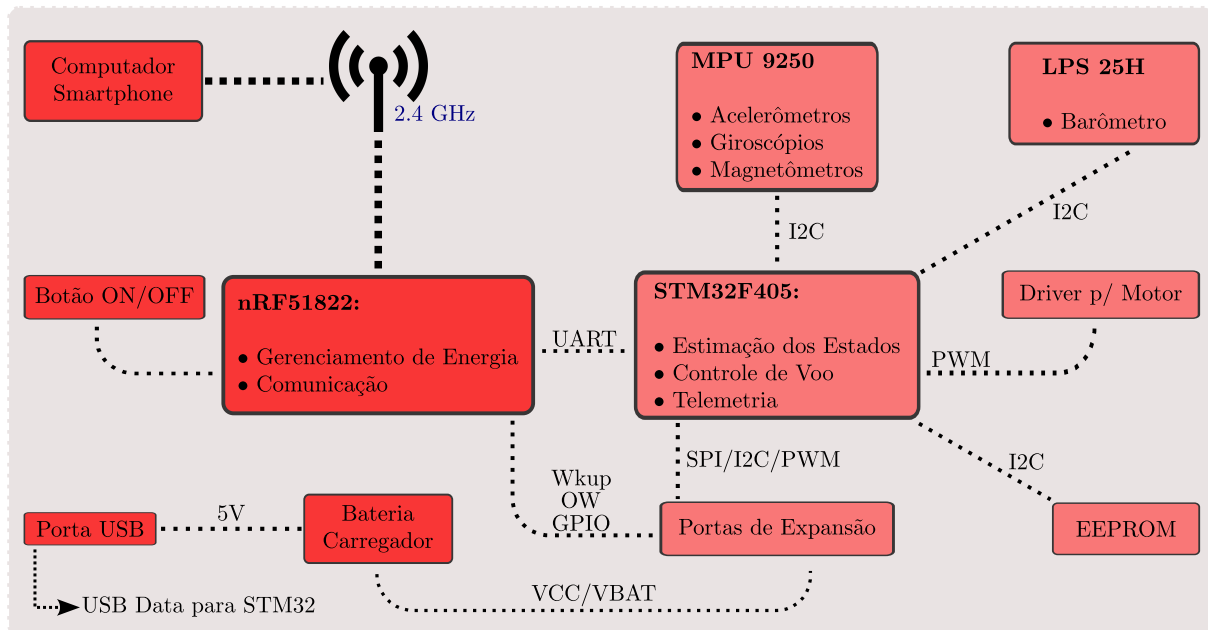
Figura 18 – Eixos de controle utilizados na manobrabilidade da aeronave *Crazyflie 2.1*.

Fonte: Elaborado pelo autor.

aeronave.

A leitura da orientação da aeronave é realizada via o *chip* **MPU-9250**. Esse sensor é uma unidade de medição inercial (*IMU*) que engloba giroscópios, acelerômetros e magnetômetros de três eixos. O sensor **LPS25H** é utilizado para captar a pressão e estima a altura da aeronave. Em sua versão mais simples, o *Crazyflie 2.1* necessita da utilização de placas secundárias (*shields*) para a realização do voo de forma autônoma.

Temos ainda que os quatros motores do *Crazyflie 2.1* são de corrente contínua (*DC*) escovados sem núcleo, com velocidade de rotação de cerca de 25000 rpm. As hélices da aeronave são fabricadas de plástico e possui 45 mm de diâmetro. O driver para o motor é feito via *MOSFET pull down* com controle realizado via modulação de largura de pulso (*PWM: Pulse Width Modulation*). Não há controle eletrônico de velocidade (*ESC: Eletronic Speed Control*) na aeronave. Para evitar a degradação da queda de tensão na bateria é realizado um ajuste via *software* de compensação.

Figura 19 – Aspecto de funcionamento do nano quadricóptero modelo *Crazyflie 2.1*.

STM32F405 Microcontrolador principal;

nRF51822 Microcontrolador secundário;

MPU Unidade de medição inercial;

LPS Sensor de pressão;

EEPROM Memória não volátil, retém as informações sem o fornecimento de energia;

SPI/I2C/UART Formas de comunicação serial;

PWM Pulso com modulação, usado para controle de tensão;

Wkup *Input/Output pin*;

GPIO *General Purpose Input/Output*;

OW *OneWire* para comunicação de dados;

VBAT *Battery voltage*.

Fonte: Adaptado de [3].

3.2 Orientação Espacial do Drone

Para a realização de navegação autônoma, voo estável, não colisão com obstáculos e outras funcionalidades avançadas quadricópteros utilizam algum sistema de posicionamento para compreender a sua localização no espaço tridimensional. Neste sentido, destacam-se, os seguintes métodos e sensores que fornecem maneiras eficazes para a implementação de um sistema de posicionamento:

- Giroscópios e acelerômetros;
- Sistema de posicionamento global (*GPS*);
- Sensores de visão (câmeras);

- Sensores ultrassônicos ou de distância a laser (Lidar);
- Unidade de medição inercial (*IMU*);
- Sensores de fluxo óptico.

Contudo, a aeronave *Crazyflie 2.1* possui certas particularidades, como por exemplo, a atuação em ambientes restritos e fechados. Nesse cenário, a localização tridimensional realizada via os sensores no veículo não possui a precisão adequada para garantir a autonomia em voo. É necessário a utilização de uma ferramenta adicional para assegurar essa funcionalidade. Assim, é utilizado um sistema de localização via rádio. O equipamento utiliza banda ultralarga (*UWB: Ultra-wideband*), baseia-se no componente *Decawave DWM1000* e possui o padrão de comunicação IEEE 802.15.4a* com ± 10 cm de precisão. Esse equipamento é conhecido como sistema de posicionamento local, ao contrário do *GPS* que é global, mas que não serve para ambientes *indoor*. A fig.(20) apresenta essa tecnologia.

Figura 20 – Sistema de posicionamento local utilizado na localização espacial do *Crazyflie 2.1*.



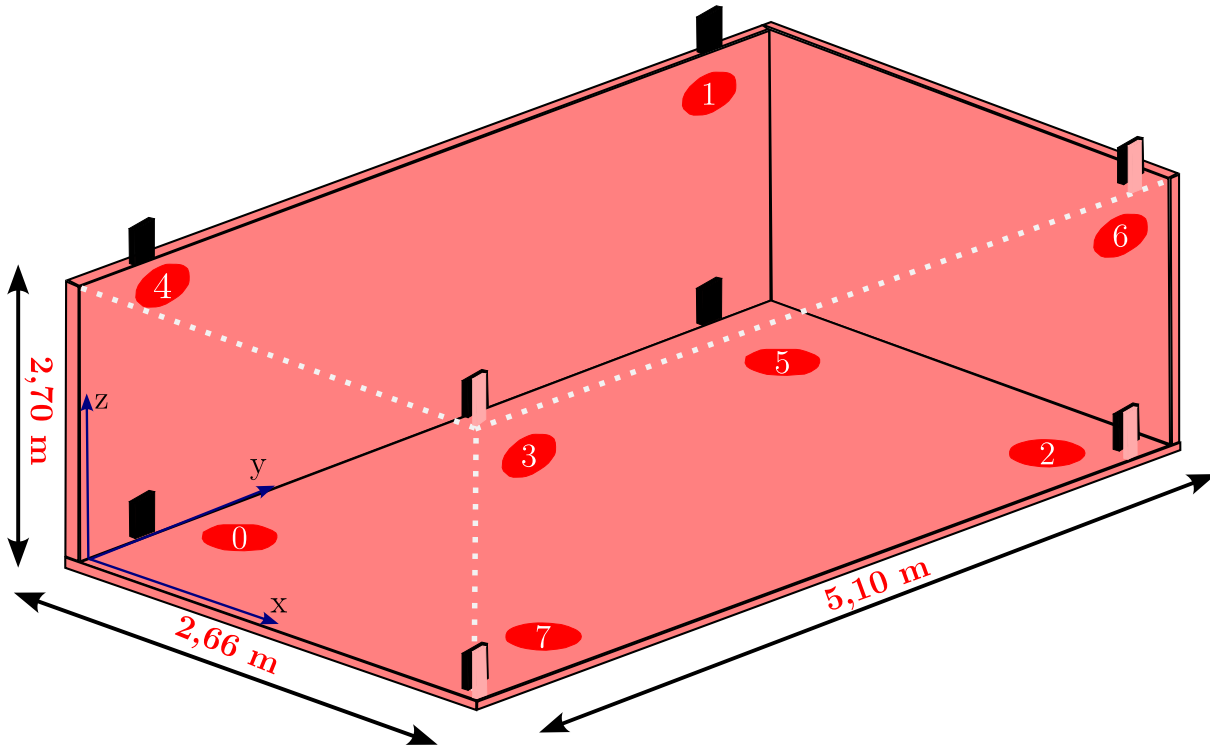
Fonte: Extraído de [3].

A localização tridimensional é feita via o compartilhamento de mensagens de rádio entre receptores na aeronave e âncoras espalhadas pelo ambiente. Para evitar interferência reflexiva o fabricante recomenda posicionar as âncoras com afastamento de 15 cm da

* Padrão de rede IEEE de comunicação sem fio conhecido como *ZigBee*. Possui menos potência, menor taxa de dados e menor ciclo de trabalho quando comparado ao *Bluetooth*. Para informações adicionais consultar o livro [26].

parede, teto ou algum tipo de obstáculo. O ambiente cujo as âncoras são anexadas, deve não ser curvo, conforme apresentado na fig.(21) a seguir.

Figura 21 – Posicionamento das âncoras no ambiente. Dimensões do laboratório de ensaio.

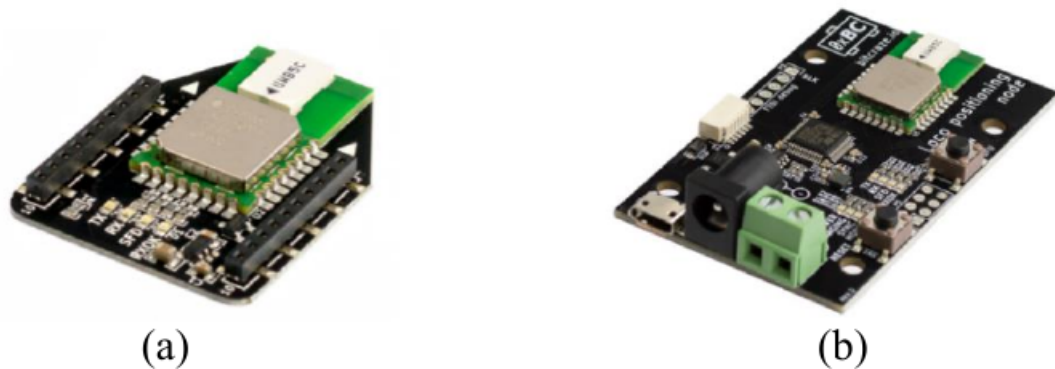


Fonte: Adaptado da *webpage*: [Loco Positioning System](#). Acesso: 22 de novembro de 2023.

O fornecimento de energia para as âncoras é feito via uma fonte externa de 5 V a 12 V. Cada âncora consome até 180 mA de corrente. O módulo de recepção é fixado na aeronave que será localizada no espaço. A fig.(22) ilustra um exemplo de receptor e de uma âncora, respectivamente.

Quando é realizada a montagem do ensaio, a posição precisa de cada âncora deve ser medida e salva via *software*. A precisão desse dado é de duas casas decimais. Existem duas possibilidades de estimar a posição do drone usando o sistema de posicionamento local. A primeira é feita com base no *TWR* (*Two Way Ranging*). O módulo receptor envia uma mensagem às âncoras sequencialmente, isso possibilita mensurar a distância da aeronave até as âncoras. Este método de localização é preciso e possibilita que a aeronave saia do espaço restrito das âncoras. O segundo método é executado via *TDofA* (*Time Difference of Arrive*). O conjunto de âncoras no ambiente encaminha de modo contínuo pacotes de sincronização. O receptor, na aeronave, ao receber esses dados calcula a distância relativa até duas âncoras, aferindo a variação de horário de chegada dos pacotes.

Figura 22 – Elementos utilizados no sistema de orientação espacial via rádio. Em (a) tem-se o receptor. Em (b) tem-se a âncora.



Fonte: Extraído de [3].

Com essa informação, é obtido a localização espacial. Essa técnica exige maior poder computacional, a precisão é menor quando comparado ao *TWR* e a aeronave não pode sair do espaço demarcado pelas âncoras. Todavia, o receptor, na aeronave, ouve essas informações passivamente. Neste cenário, é possível utilizar outros drones *Crazyfly* em voo no mesmo ambiente.

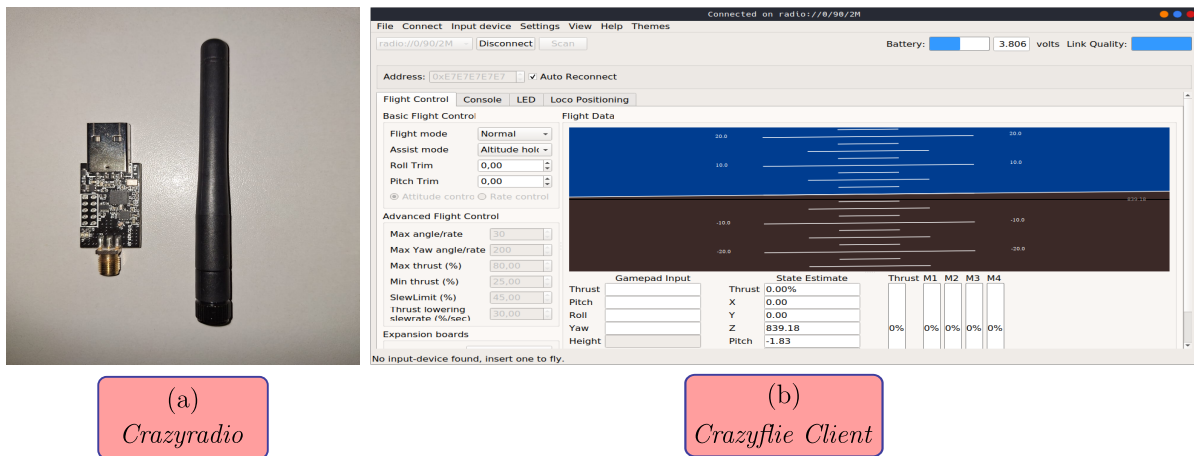
3.3 Transmissão de Dados do Drone

O monitoramento das informações recebidas da aeronave pode ser visualizado em computadores e em dispositivos móveis. Essa comunicação é feita com base no protocolo de rádio e não utiliza fios (*wireless*). Para o computador utiliza-se *ESB* (*Enhanced ShockBurst*) com um *dongle* conectado à porta *USB* do dispositivo, chamado (*Crazyradio*). A aplicação no computador é conhecida como *Crazyfly Client*. Em dispositivos móveis a comunicação é feita por *Bluetooth*. A fig.(23) apresenta o *Crazyradio* que é conectado ao computador e o *Crazyfly Client* respectivamente.

Para o projeto foi utilizado o *Crazyradio*. Ele tem características como baixa latência e comodidade de uso no computador. Adicionalmente, a troca de informação da aeronave com o *Crazyradio* é feita de forma bidirecional com base em um transceptor *Nordic Semiconductor* de 2,4 GHz, com taxa de até 2 Mb/s. A comunicação do *Crazyradio* é ampliada em sua saída em 20 dBm, para distâncias maiores do que 1 Km.

A troca de mensagens com a aeronave é feita via a transmissão de pacotes por um determinado canal. O *Crazyradio* aguarda a confirmação de recebimento da mensagem

Figura 23 – Em (a) tem-se o *Crazyradio*; em (b) tem-se o *Crazyflie Client*.



Fonte: Elaborado pelo autor.

pela aeronave. Se não ocorrer erros na comunicação, a aeronave confirma o recebimento da mensagem fazendo com que o *Crazyradio* envie o próximo pacote. Caso ocorra erro na confirmação da transmissão, a mesma mensagem é enviada de maneira contínua sempre que houver uma aeronave detectada.

Essa técnica é capaz de utilizar aproximadamente 80 canais diferentes, cada mensagem de dado é enviada com uma localização de acesso. Desta forma, há a possibilidade de outra aeronave compartilhar o mesmo canal. O período entre o início de transmissão dos dados e a sua resposta é estimada em 2 ms, sendo 1 ms para o *USB* e 1 ms para o rádio com frequência de 2 Mbps.

O *firmware* da aeronave é escrito na linguagem de programação C e é baseado no sistema operacional de tempo real *FreeRTOS*. A aplicação de troca de mensagem da aeronave com o computador utiliza rotinas escritas na linguagem de programação *Python*.

Existe uma biblioteca fornecida pela fabricante da aeronave que é possível ajustar os *setpoints*, receber as informações sobre o sistema de sensoramento e modificar características no *firmware* do drone.

3.4 Informações Físicas do Drone

As informações físicas de quadricópteros são importantes ferramentas para a compreensão e monitoramento do seu comportamento. Assim, os parâmetros físicos são utilizados para modelar equações matemáticas que descrevam a dinâmica da aeronave fornecendo características úteis para o controle de voo. Dessa forma, a Tabela (4) apresenta

os parâmetros físicos via ensaio e literatura da aeronave *Crazyflie 2.1*.

Tabela 4 – Medidas físicas do *Crazyflie 2.1*.

Variáveis	Unidade de Medida	Ensaio	Literatura
Massa (m)	kg	0,0282	0,0284
Distância entre o centro de massa e os rotores (l)	m	0,0461	0,0467
Momento de inércia em torno do eixo x (I_{xx})	kgm^2	$1,473 \times 10^{-5}$	$1,657 \times 10^{-5}$
Momento de inércia em torno do eixo y (I_{yy})	kgm^2	$1,479 \times 10^{-5}$	$1,665 \times 10^{-5}$
Momento de inércia em torno do eixo z (I_{zz})	kgm^2	$2,847 \times 10^{-5}$	$2,926 \times 10^{-5}$
Coefficiente de arrasto dos rotores (d)	kgm^2/rad^2	$1,217 \times 10^{-10}$	$1,688 \times 10^{-10}$
Coefficiente de empuxo dos rotores (b)	kgm/rad^2	$2,364 \times 10^{-8}$	$2,220 \times 10^{-8}$
Velocidade de rotação máxima dos rotores ($\Omega^{máx}$)	rad/s	2617	2500

Fonte: Extraído de [3].

3.4.1 Massa

A massa é uma característica essencial que impacta a dinâmica de voo, a força de sustentação, entre outros em veículos aéreos. No nano quadricóptero *Crazyflie 2.1* a massa é partilhada pelos seus componentes como pode ser visualizado na fig.(24) a seguir. Os itens que compõem o drone incluem placa eletrônica fig.(24a), bateria de LiPo fig.(24b), suporte para a bateria fig.(24c), motores elétricos fig.(24d), suporte para os motores fig.(24e) e hélices fig.(24f).

A coleta da massa dos componentes do drone ilustrado na fig.(24) foi feita utilizando uma balança de precisão de capacidade máxima de 220 gramas e resolução de 0,0001 gramas. A Tabela (5) apresenta a distribuição de massa dos componentes do drone.

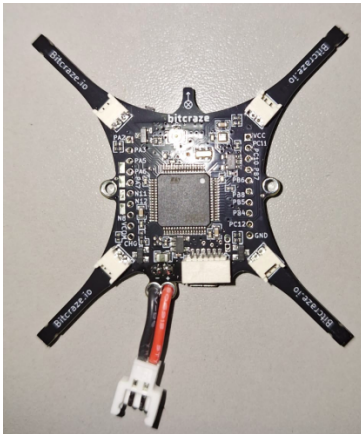
3.4.2 Distância Entre o Centro de Massa e os Rotores

O valor da distância entre o centro de massa dos rotores na aeronave é de 0,0461 m. Esse valor foi obtido utilizando-se um paquímetro analógico, que possui capacidade máxima de medida de 200 mm e resolução de 0,05 mm.

* Na configuração final o drone utiliza quatro motores. A massa de cada motor é 0,6442 gramas.

† A massa de cada suporte de motor é 0,0981.

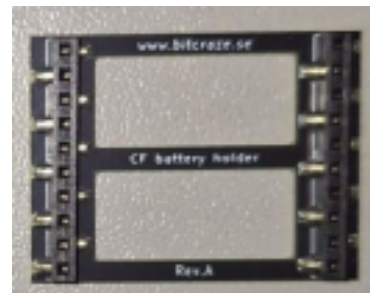
‡ A massa de cada hélice é 0,0668.

Figura 24 – Componentes do nano quadricóptero *Crazyflie 2.1*.

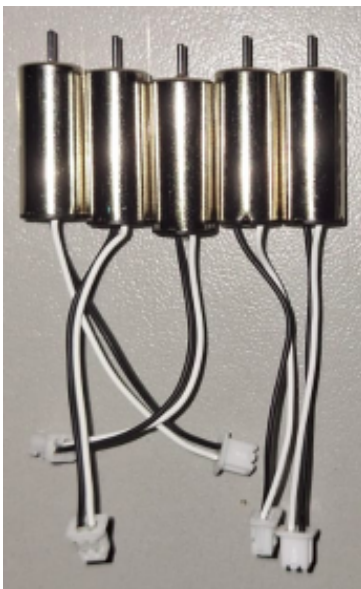
(a) Placa eletrônica.



(b) Bateria de LiPo de 3,7 V (250 mAh).



(c) Suporte para a bateria.



(d) Motores elétricos. São utilizados quatro motores na configuração final.



(e) Suporte para os motores elétricos. São utilizados quatro suportes na configuração final.



(f) Hélices. São utilizadas quatro hélices na configuração final.

Fonte: Elaborado pelo autor.

Tabela 5 – Arranjo da massa do nano quadricóptero *Crazyflie 2.1*.

Componente	Massa (g)
Placa eletrônica	6,7307
Bateria de LiPo	7,1667
Suporte para a bateria	1,4418
Motor*	2,5768
Suporte para o motor†	0,3922
Hélice†	0,2673
Total	28,2844

Fonte: Extraído de [3].

3.4.3 Momento de Inércia

Quando estudamos a dinâmica de corpos rígidos em rotação em torno de um eixo fixo deparamos com uma propriedade física que descreve o grau de dificuldade em se alterar a velocidade angular desse objeto. Esse comportamento é conhecido como momento de inércia e exerce uma função semelhante ao da massa em um corpo em translação quando este muda a sua velocidade linear, ou seja, se a intensidade do momento de inércia de um corpo for elevada será mais difícil girá-lo ou mudar a sua rotação [27, 28].

De maneira teórica, o momento de inércia de um corpo rígido bidimensional é obtido do cálculo da distribuição de massa em relação a algum eixo de referência via integração. É realizado a multiplicação de cada componente de massa dm do corpo por r^2 . Onde r é a distância de dm ao eixo de rotação. A fig.(25) ilustra um exemplo dessa propriedade para um corpo rígido qualquer.

O momento de inércia é representado pela letra I , em seu cálculo necessita-se do corpo e do eixo ao qual será executada a rotação. A sua definição matemática para corpos rígidos bidimensionais é apresentado na eq.(3.1) a seguir.

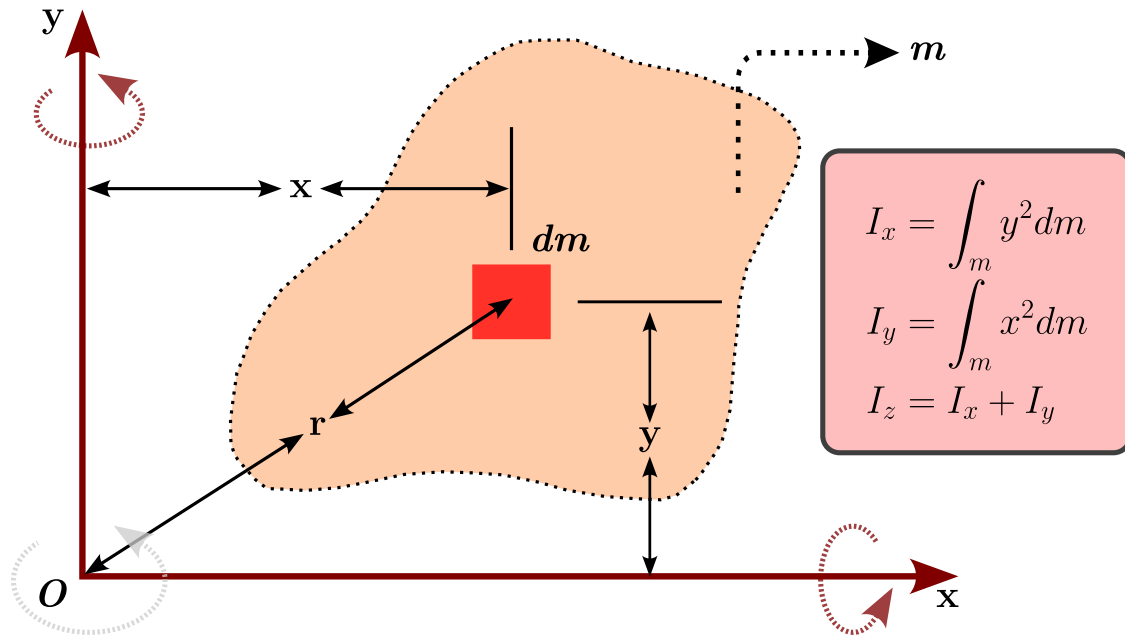
$$I = \int \overbrace{r^2}^{\text{Momento de Inércia}} \cdot \underbrace{dm}_{\text{infinitesimal de massa}} \quad (3.1)$$

distância ao quadrado

Um quadricóptero é um corpo rígido tridimensional, o momento de inércia[§] do

.....
[§] Nesse caso, também pode ser chamado de tensor de inércia.

Figura 25 – Momento de inércia de massa de um corpo rígido bidimensional qualquer.



Fonte: Adaptado de [27,28].

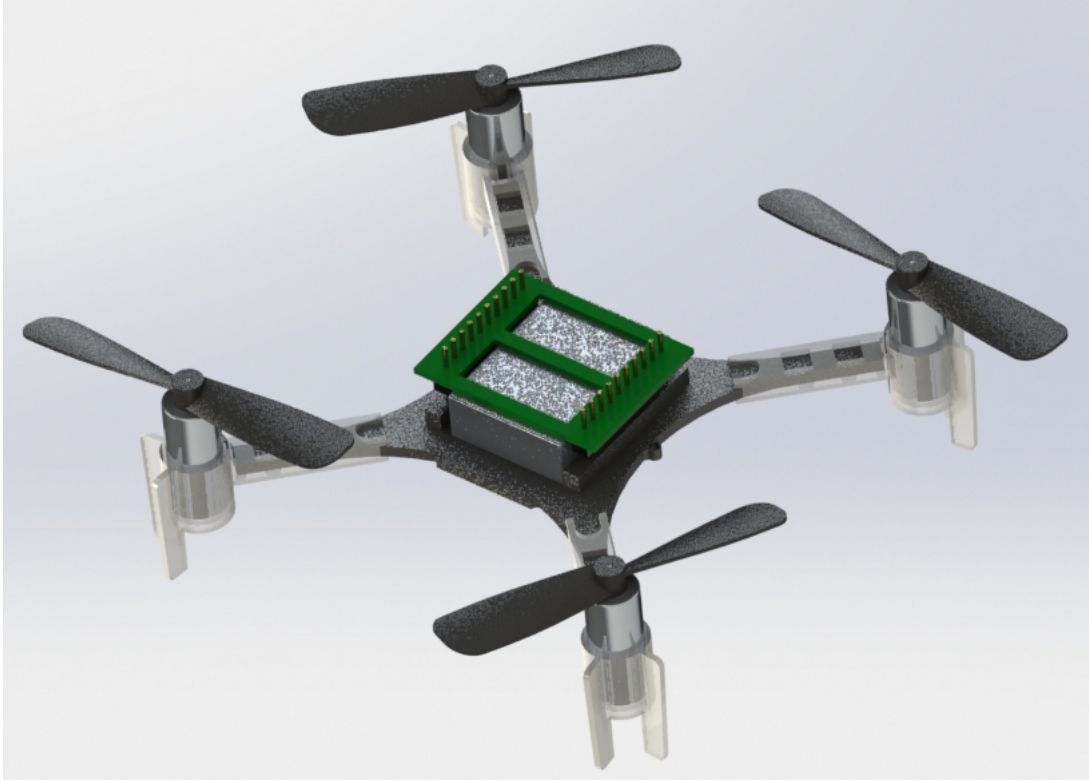
equipamento é escrito na representação matricial de acordo com a eq.(3.2) a seguir.

$$J = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix}. \quad (3.2)$$

O corpo do drone é simétrico em relação aos seus eixos. Os únicos elementos não nulos na eq.(3.2) são os componentes da diagonal principal. Podemos obter o momento de inércia da aeronave *Crazyflie 2.1* ao redor dos eixos de rotação utilizando o *software SolidWorks*. Com base no CAD (*computer-aided design*) da aeronave *Crazyflie 2.0** ilustrado na fig.(26) obtemos os valores de I_{xx} , I_{yy} e I_{zz} respectivamente.

Para a realização do cálculo do momento de inércia de massa via o *software SolidWorks* é necessário adicionar os valores da massa de cada componente do drone, apresentado previamente na Tabela (5) pg.(65). Os valores obtidos para o momento de inércia de massa em torno dos eixos x , y e z são apresentados a seguir.

* As dimensões da aeronave *Crazyflie 2.0* não difere da aeronave *Crazyflie 2.1*.

Figura 26 – CAD da aeronave *Crazyflie 2.0*.

Fonte: Adaptado de [3].

$$I_{xx} = 1,473 \times 10^{-5} \text{kgm}^2, \quad (3.3)$$

$$I_{yy} = 1,479 \times 10^{-5} \text{kgm}^2, \quad (3.4)$$

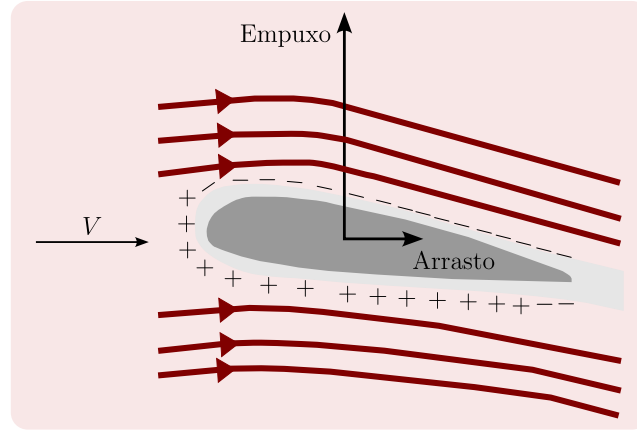
$$I_{zz} = 2,847 \times 10^{-5} \text{kgm}^2. \quad (3.5)$$

3.4.4 Coeficiente de Arrasto

Um corpo submetido ao deslocamento através de um fluido, especialmente um líquido enfrenta resistência. Assim, o corpo experimenta forças e momentos devido ao fluido. Na direção do escoamento o fluido em movimento exerce uma força chamada de arrasto [29]. A densidade ρ do fluido, a velocidade a montante V , o tamanho, a forma e a orientação do corpo, entre outros exercem influência na força de arrasto. A fig.(27) ilustra esse comportamento.

Para caracterizar essa oposição exercida no corpo pelo fluido é conveniente trabalharmos com um número adimensional conhecido como coeficiente de arrasto C_D apresentado

Figura 27 – Arrasto e empuxo agindo em um perfil bidimensional.



Fonte: Adaptado de [29].

na eq.(3.6) a seguir:

$$C_D = \frac{F_D}{\frac{1}{2} \cdot \rho \cdot V^2 \cdot A}, \quad (3.6)$$

onde A é a área frontal, ou seja, é a área traçada sob um plano normal à direção do escoamento, $\frac{1}{2}\rho V^2$ é a pressão dinâmica e F_D é a força de arrasto.

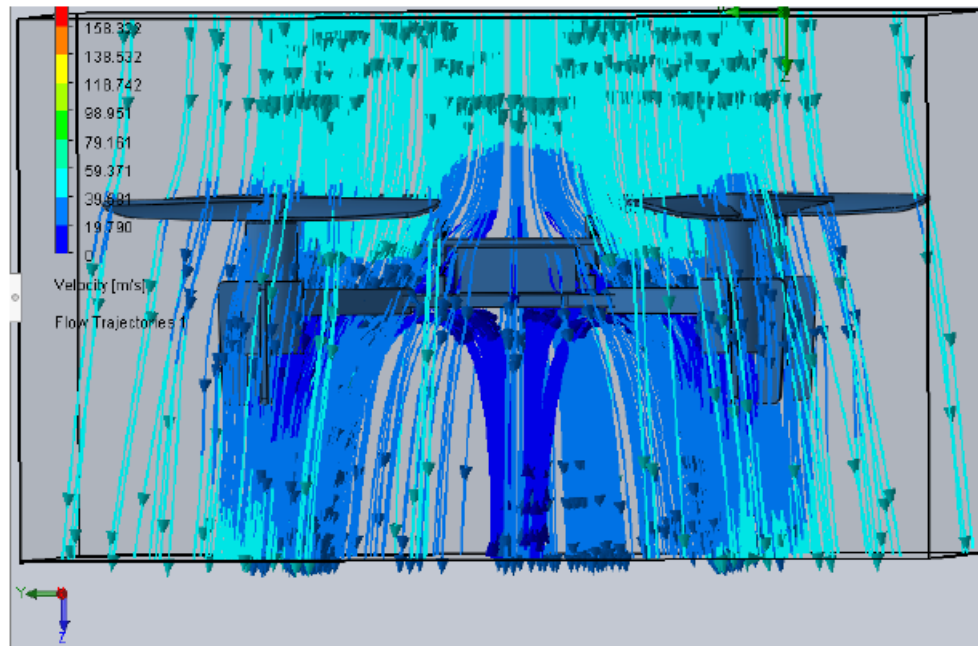
Para obter o coeficiente de arrasto da aeronave *Crazyflie 2.1* foi utilizado o CAD do drone no software *SolidWorks* via o pacote *Flow Simulation* conforme apresentado na fig.(28). Para a realização da simulação foram adotados os seguintes valores: $V = 45,535$ m/s e $\rho = 1,29$ kg/m³. Desta forma, foi obtido o valor de $C_D = 0,2056$ para o coeficiente de arrasto da aeronave.

3.4.5 Coeficiente de Empuxo

A realização do movimento ascendente e descendente do quadricóptero é realizado devido a ação da força de empuxo. Esse estímulo é obtido da soma das forças criadas por cada motor do drone e manifesta-se devido ao equipamento ocupar o local no espaço do fluido, ou seja, do ar. Essa propriedade física é uma grandeza vetorial e é oposta a força peso do veículo. Para a obtenção dessa grandeza é conveniente trabalharmos com o coeficiente de empuxo que é uma entidade adimensional que representa a capacidade de um objeto gerar empuxo ou sustentação quando sujeito a um fluido.

Para a obtenção experimental do coeficiente de empuxo dos motores da aeronave foram utilizados os seguintes itens:

Figura 28 – Simulação via *CAD* para obtenção do coeficiente de arrasto.



Fonte: Extraído de [3].

- Balança de 500 gramas de capacidade máxima e resolução de 0,01 g;
- Tacômetro óptico com resolução de 0,01 rpm;
- Aeronave *Crazyflie 2.0*[¶];
- Garrafa com água;
- Fitas adesivas e reflexivas.

O tacômetro óptico foi utilizado para obter a velocidade angular dos motores e a balança estimou a força produzida pela aeronave. A velocidade angular dos motores foi ampliada em 16 passos via a alteração do valor de *duty cycle*. O experimento foi realizado três vezes e os valores médios são apresentados na Tabela (6) a seguir.

Com os dados da tabela foi possível obter a representação gráfica do experimento. A fig.(29) apresenta os resultados. A fig.(29a) mostra o empuxo em função do *duty cycle* e a fig.(29b) apresenta o empuxo em função da velocidade angular da aeronave.

Para a realização do ensaio foi montado o arranjo experimental apresentado na fig.(30) a seguir.

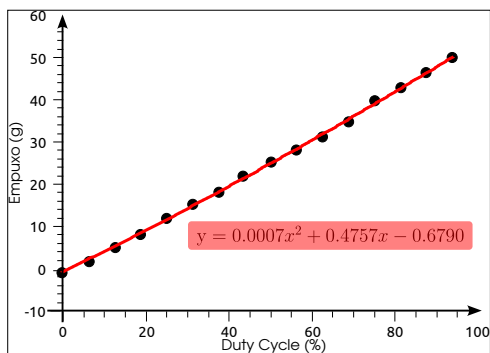
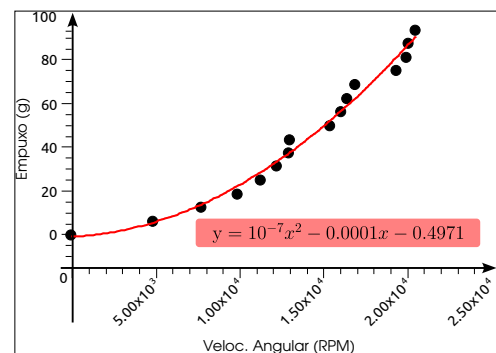
.....
[¶] A aeronave *Crazyflie 2.1* não difere significativamente da aeronave *Crazyflie 2.0*.

Tabela 6 – Ensaio de empuxo da aeronave *Crazyflie 2.0*.

Empuxo (g)	PWM (%)	Vel. Angular (rpm)
0,00	0,00	0,00
1,60	6,25	4804,33
5,10	12,50	7643,67
8,20	18,75	9796,00
11,90	25,00	11177,33
15,20	31,25	12134,67
18,10	37,50	12871,00
21,90	43,25	12945,00
25,30	50,00	15349,00
28,00	56,25	15990,33
31,10	62,50	16330,67
34,75	68,76	16840,00
39,70	75,00	19304,33
42,75	81,25	19910,00
46,45	87,50	20035,33
49,90	93,75	20449,00

Fonte: Extraído de [3].

Figura 29 – Representação gráfica do ensaio experimental para determinar o empuxo da aeronave.

(a) Empuxo em função do *duty cycle*.

(b) Empuxo em função da velocidade angular.

Fonte: Adaptado de [3].

Figura 30 – Experimento para a aquisição do coeficiente de empuxo da aeronave.



Fonte: Extraído de [3].

Analisando o gráfico apresentado na fig.(29b) nota-se que o empuxo da aeronave é proporcional ao quadrado da velocidade de rotação dos rotores (Ω) e fisicamente é oposto a força peso do drone. Neste sentido, considerando os quatro rotores do veículo o cálculo do coeficiente de empuxo da aeronave pode ser aproximado pela eq.(3.7). Assim, substituindo os dados experimentais na equação obtêm-se o valor do coeficiente de empuxo igual a $b = 2,364 \times 10^{-8} \text{ kgm/rad}^2$.

$$b = \frac{mg}{4\Omega^2}. \quad (3.7)$$

CAPÍTULO 4

METODOLOGIA UTILIZADA NO PROJETO

Embarcar a unidade sensorial no nano quadricóptero modelo *Crazyflie 2.1* para a realização da coleta dos dados de temperatura e umidade envolveu algumas etapas e considerações visando assegurar que o sensor fosse instalado da maneira correta e funcionasse de forma adequada. Foram realizadas modificações no *hardware* da aeronave, cuja placa eletrônica possui implementação e suporte para a comunicação serial *I²C* e foi desenvolvido o *firmware*, embutido no drone, escrito em linguagem C com recursos e funcionalidades do sistema operacional de tempo real *FreeRTOS*. O *software* criado foi o responsável em garantir a função de comunicação da unidade sensorial com a aeronave, sendo executado de maneira contínua, em outras palavras, o drone estando ligado capta as informações de temperatura e umidade do ambiente a sua volta. Adicionalmente, foram desenvolvidos *scripts* na linguagem *Python* para acionar o drone, coletar os *logs* de interesse do veículo para tratá-los e enviá-los a nuvem. Esse processo é descrito mais detalhadamente a seguir.

4.1 Adequação do *Hardware* da Aeronave *Crazyflie 2.1*

O compartilhamento de dados de temperatura e umidade da unidade sensorial **AHT10** é feita com base no protocolo de comunicação serial *I²C*. A placa eletrônica da aeronave *Crazyflie 2.1*, fig.(24a) pg.(64), possui funcionalidades similares a de um multiplexador, dispondo de suporte e implementação *I²C* em seu *hardware*. Desta forma, para embarcar o sensor no veículo foi realizada a soldagem dos pinos correspondentes dos dispositivos, necessários para a troca de informações de interesse, conforme apresentado na fig.(31) a seguir.

Figura 32 – Aeronave embarcada com o sensor **AHT10**.

Fonte: Elaborado pelo autor.

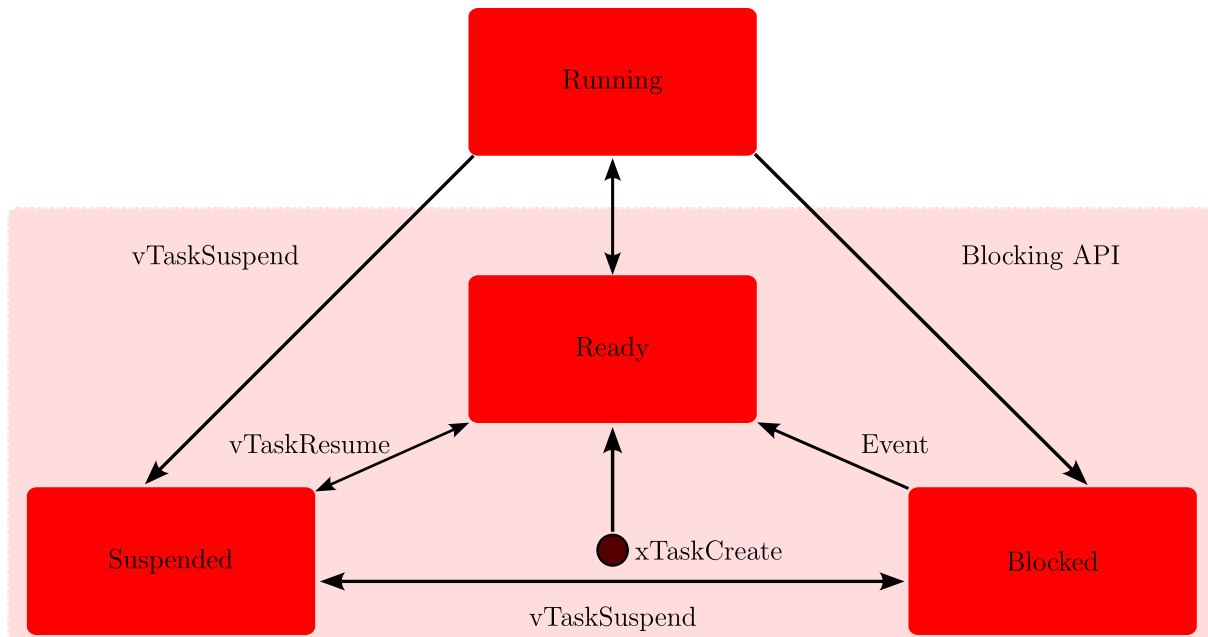
4.2.1 *Firmware*

Para controlar as funcionalidades de comunicação em baixo nível entre a unidade sensorial e a aeronave foi desenvolvido um *firmware*. O *software* foi embutido no *hardware* do veículo *Crazyflie 2.1* e funciona de maneira ininterrupta. O programa foi escrito em linguagem C com recursos do sistema operacional de tempo real *FreeRTOS*. Esse SO, em geral, é utilizado em projetos que envolvam sistemas embarcados e provê recursos como confiabilidade, portabilidade, segurança, entre outros, sendo a “base” da aplicação. A fig.(33) apresenta o diagrama de administração de *tasks* feita pelo *FreeRTOS*.

O *FreeRTOS* como outros sistemas operacionais possui a capacidade de executar diversas tarefas ou *tasks* em conjunto. Para o projeto o *firmware* desenvolvido opera a tarefa de inicializar o barramento I^2C e gerenciar a operação de escrita e leitura de dados no barramento em *loop* infinito. A coleta de informações sobre a temperatura e a umidade do sensor **AHT10** são armazenados em *logs** que podem ser utilizados posteriormente na camada de aplicação do sistema.

Para que o drone consiga realizar a leitura e a escrita de dados no sensor, foi

* Refere-se as mensagens registradas quando o *software* está sendo executado.

Figura 33 – Diagrama de gestão de *tasks* no *FreeRTOS*.

Fonte: Adaptado de [31].

utilizado os endereços de comunicação da unidade sensorial conforme apresentado na Tabela (2) pg(51) baseado no *datasheet* do dispositivo.

No processo de comunicação, inicialmente é definido um ponteiro para o barramento *I²C* do sensor e é escrito 1 *byte* no endereço **0x38** utilizando o *buffer* **0xBE**, forçando o barramento a permanecer ligado. Em seguida, é utilizado o *trigger* da unidade sensorial para permitir que o *firmware* acesse os dados coletados do sensor. É enviado 1 *byte* para o endereço **0x38** utilizando o *buffer* **0xAC**, após isso é dado um *delay* de 200 ms para que o sensor armazene as informações coletadas. A leitura dos dados é realizada enviando 6 *bytes* para o endereço de localização do sensor, armazenando-os em um vetor. Os dados colhidos nesse momento, são chamados de “*crus*”. A obtenção dos valores reais de temperatura e de umidade é feito via operações *bit a bit* com os valores salvos no vetor de dados e utilizando as operações de conversão, apresentadas nas eqs.(2.1, 2.2). O pseudocódigo (1) a seguir do *firmware* responsável em administrar o compartilhamento de informações entre a aeronave *Crazyflie 2.1* e o sensor **AHT10** resume esse cenário.

4.2.2 Acionamento da Aeronave, Coleta de *Logs* & Camada de Aplicação

As funções de aplicação do projeto, como por exemplo o acionamento da aeronave, a coleta dos *logs* de interesse, análise dos dados coletados e a interatividade com a nuvem

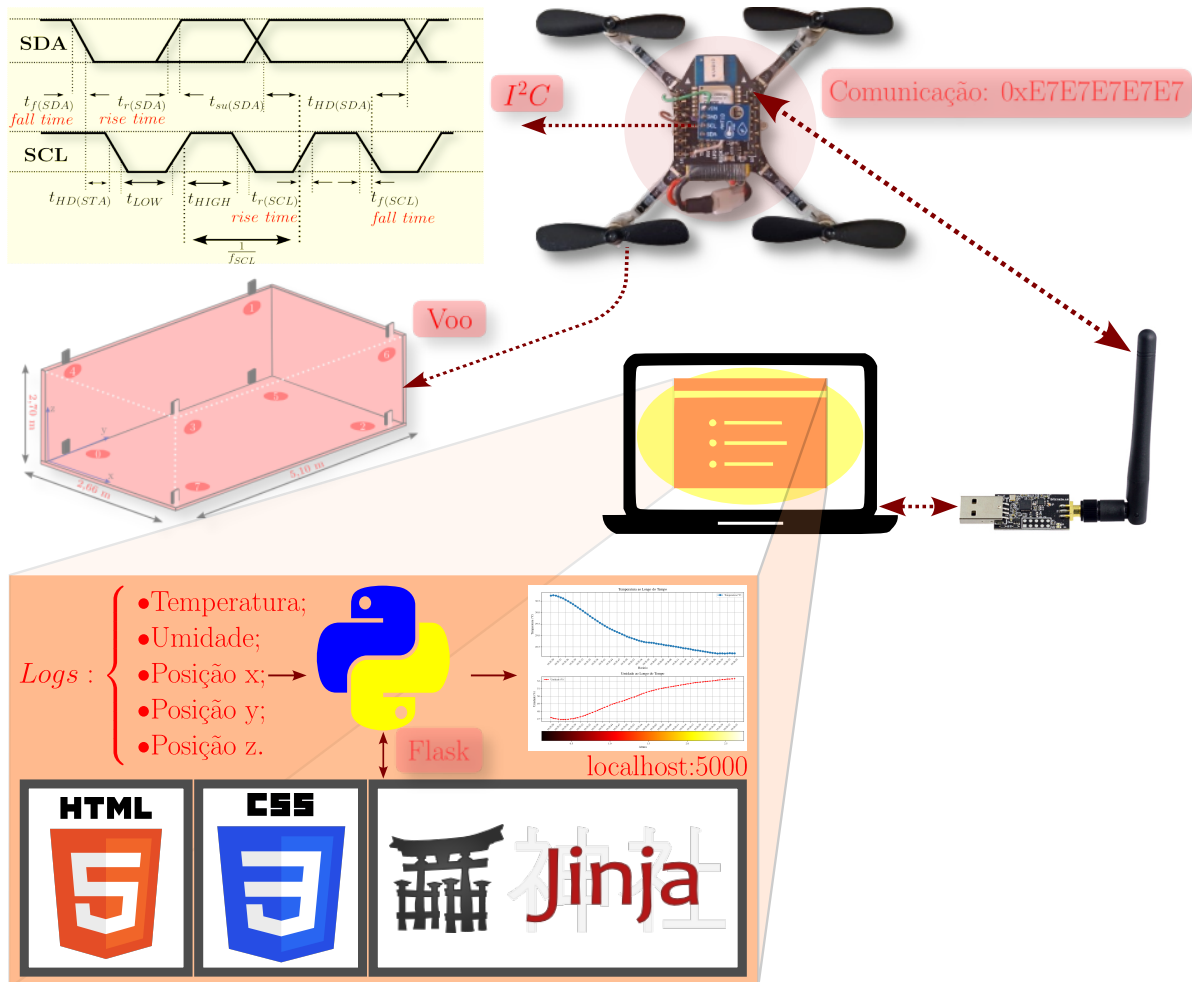
Algoritmo 1: Pseudocódigo do *firmware* desenvolvido

Entrada: Endereço I^2C do sensor: **0x38**
Endereço de inicialização do sensor: **0xBE**
Endereço do *trigger*: **0xAC**
Booleano: *isInit*
float: temperatura & umidade
void: *aht10Task*

- 1 Função: *aht10Init()*
- 2 Inicia o *deck driver* com ponteiro para o periférico I^2C
- 3 Inicia o barramento do sensor **AHT10** usando o endereço de *buffer*: **0xBE**
// Escreve 1 *byte* em **0x38**
- 4 Função *appMain()*
- 5 Criação da *task* com os parâmetros do sensor **AHT10**
- 6 Função: *aht10Test()*
- 7 Verifica se o *deck driver* iniciou corretamente
- 8 se sim, retorna verdadeiro
- 9 Função Principal: *aht10Task()*
- Data:** *data*[6]
- 10 Inicializa o sistema principal
- 11 **while** *verdadeiro* **do**
- 12 Escreve 1 **byte** no endereço: **0x38** usando o *buffer*: **0xAC**
- 13 Lê 6 **bytes** do barramento de dados, endereço: **0x38** e armazena no vetor **data**
- 14 **Realiza operações *bit a bit* com os valores do vetor *data***
- 15 **uint32_t** *dados_crus_temp* = (*data*[3] & 0x0F) // 0x0F: máscara; &
 bitwise and
- 16 *dados_crus_temp* «= 8 // *left shift*: desloca 8 *bits* para a
 esquerda
- 17 *dados_crus_temp* |= *data*[4] // | *bitwise or*
- 18 *dados_crus_temp* «= 8
- 19 *dados_crus_temp* |= *data*[5]
- 20 **Obtém o valor da temperatura:**
- 21 *temperatura* = $\left(\frac{\textit{dados_crus_temp}}{2^{20}}\right) \cdot 200.0 - 50.0$
- 22 **uint32_t** *dados_crus_umidade* = *data*[1]
- 23 *dados_crus_umid* «= 8
- 24 *dados_crus_umid* |= *data*[2]
- 25 *dados_crus_umid* «= 4
- 26 *dados_crus_umid* |= *data*[3] » 4
- 27 **Obtém o valor da umidade:**
- 28 *umidade* = $\left(\frac{\textit{dados_crus_umid}}{2^{20}}\right) \cdot 100.0$
- 29 **end**
- Saída** : *Logs* de temperatura e umidade

foram feitos utilizando *scripts* via a linguagem de programação *Python*. A fig.(34) apresenta a camada de aplicação de alto nível do sistema.

Figura 34 – Camada de aplicação do projeto.



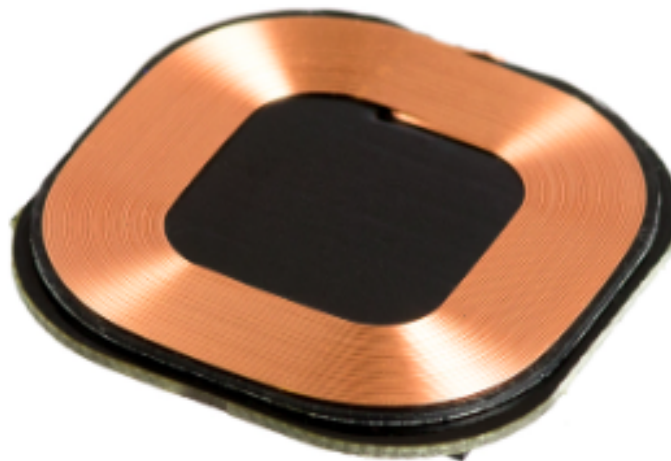
Fonte: Elaborado pelo autor.

O voo do drone é realizado em um ambiente *indoor* com o auxílio do sistema de posicionamento local. As âncoras são posicionadas no ambiente e o receptor dos dados da comunicação é anexado na aeronave *Crazyfly 2.1*, fig(22) pg(61). A trajetória realizada pelo veículo no ambiente, a coleta dos *logs* de interesse, entre outros, são inseridos no *script* responsável pelo controle de voo da aeronave. A troca de informações do drone com o código é feita via rádio utilizando o *Crazyradio Real-Time Protocol (CRTP)* desenvolvido pela *Bitcraze*. O identificador da transmissão (*URI*) é único, sendo da seguinte forma: **radio://0/80/2M/E7E7E7E7E7**, onde **radio://0/** representa o meio da comunicação, **80** caracteriza a quantidade de canais possíveis para a realização da transmissão, **2M** é a velocidade da troca de dados e **E7E7E7E7E7** é o endereço de identificação do drone.

Quando o drone finaliza o voo os dados de interesse captados são salvos em um arquivo **.csv** para compor a base de dados do sistema. Esses dados são tratados e analisados para a avaliação do ensaio.

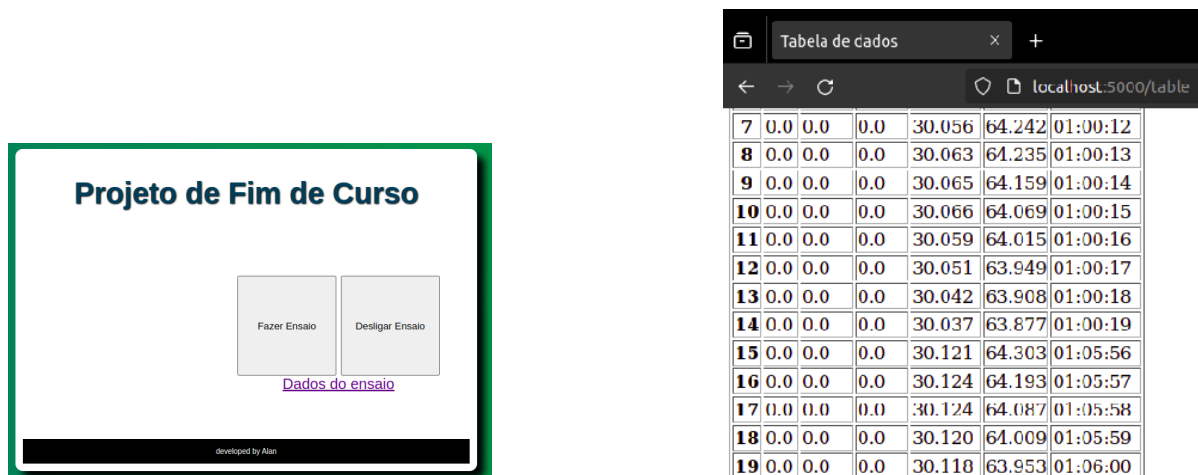
A aplicação *web* possibilita iniciar e encerrar o *software* que faz o controle de voo do drone e permite a visualização da base de dados **.csv** *on-line*. A autonomia do veículo é garantida utilizando-se o *deck* para a realização da recarga sem fio do drone, apresentado na fig(35) a seguir. Após o término da trajetória programada para o *Crazyflie 2.1*, é dado um *delay* para o drone pousar sob o carregador indutivo e encher a sua bateria. Feito isto, a trajetória é retomada repetindo o ciclo em um *loop* infinito. Caso ocorra algum problema há a possibilidade de encerrar o voo clicando no botão “desligar ensaio” na aplicação *web*.

Figura 35 – Carregador de bateria sem fio da aeronave *Crazyflie 2.1*.



Fonte: Extraído de *Qi 1.2 charger deck*.

O desenvolvimento da aplicação *web* baseou-se no *framework Flask*, encarregado de renderizar a página e definir as rotas de informações. Foi utilizado um servidor local (*localhost*) escutando a porta 5000. Além disso, foi utilizado *HTML* para a criação do *site* e para possibilitar a interatividade da aplicação no lado do cliente. A apresentação visual foi feita com base em *CSS*, responsável em modificar os estilos dos elementos *HTML*. Os conteúdos dinâmicos da página *web* basearam-se em *Jinja*. A fig.(36a) apresenta o *layout* da aplicação e a fig.(36b) mostra a base de dados acessada *on-line*.

Figura 36 – Aplicação *web* desenvolvida para o projeto, lado do cliente.

Fonte: Elaborado pelo autor.

4.2.3 Ensaios

Os ensaios de voo com o auxílio do sistema de posicionamento local foram repetidos três vezes com as trajetórias apresentadas na Tabela (7) a seguir. Para a criação da base de dados, foram coletadas as posições do drone nos eixos x, y e z, os valores de temperatura e de umidade do ambiente durante a realização do voo.

4.2.4 Caracterização do Sensor AHT10

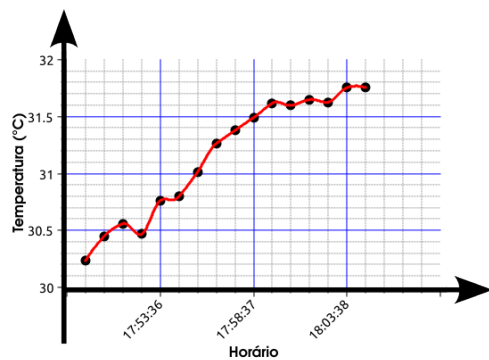
O comportamento típico do sensor quando os motores da aeronave *Crazyflie 2.1* não estão em funcionamento e a coleta é realizada em um ambiente *indoor* é apresentado na fig.(37). De acordo com o *datasheet* do dispositivo o equipamento já vem calibrado. Não foram observados valores *outliers*[†] ao longo da captura dos dados, conforme pode ser visualizado na figura.

[†] Valores que se distanciam consideravelmente do restante dos dados em um conjunto.

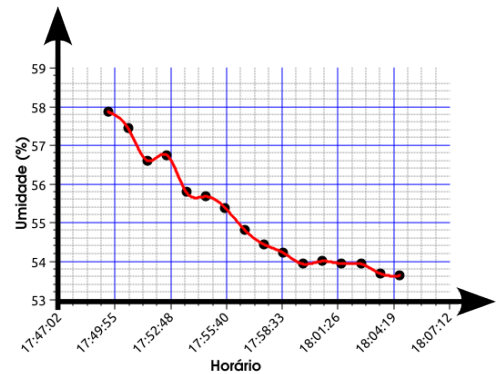
Tabela 7 – Ensaio de voo da aeronave *Crazyflie 2.1*.

Ensaio 1			Ensaio 2			Ensaio 3		
x	y	z	x	y	z	x	y	z
0.4	0.4	1.3	0.4	0.4	1.3	0.4	0.4	1.3
1.4	0.4	1.3	1.4	0.4	1.3	1.4	0.4	1.3
2.5	0.4	1.3	2.5	0.4	1.3	2.5	0.4	1.3
2.5	1.6	1.3	2.5	1.6	1.3	2.5	1.6	1.3
2.5	2.8	1.3	2.5	2.8	1.3	2.5	2.8	1.3
1.4	2.8	1.3	1.4	2.8	1.3	1.4	2.8	1.3
0.4	2.8	1.3	0.4	2.8	1.3	0.4	2.8	1.3
0.4	1.6	1.3	0.4	1.6	1.3	0.4	1.6	1.3
0.4	0.4	1.3	0.4	0.4	1.3	0.4	0.4	1.3
0.4	0.4	2.6	0.4	0.4	2.6	0.4	0.4	2.6
1.4	0.4	2.6	1.4	0.4	2.6	1.4	0.4	2.6
2.5	0.4	2.6	2.5	0.4	2.6	2.5	0.4	2.6
2.5	1.6	2.6	2.5	1.6	2.6	2.5	1.6	2.6
2.5	2.8	2.6	2.5	2.8	2.6	2.5	2.8	2.6
1.4	2.8	2.6	1.4	2.8	2.6	1.4	2.8	2.6
0.4	2.8	2.6	0.4	2.8	2.6	0.4	2.8	2.6
0.4	1.6	2.6	0.4	1.6	2.6	0.4	1.6	2.6
0.4	0.4	2.6	0.4	0.4	2.6	0.4	0.4	2.6
0.4	0.4	1.3	0.4	0.4	1.3	0.4	0.4	1.3
0.4	0.4	0.5	0.4	0.4	0.5	0.4	0.4	0.5

Fonte: Elaborado pelo autor.

Figura 37 – Comportamento típico da unidade sensorial quando os motores do drone estão desligados. A coleta dos dados foram feitas em um ambiente *indoor*.

(a) Temperatura (°C) em função do horário da coleta.



(b) Umidade relativa (%) em função do horário da coleta.

Fonte: Elaborado pelo autor.

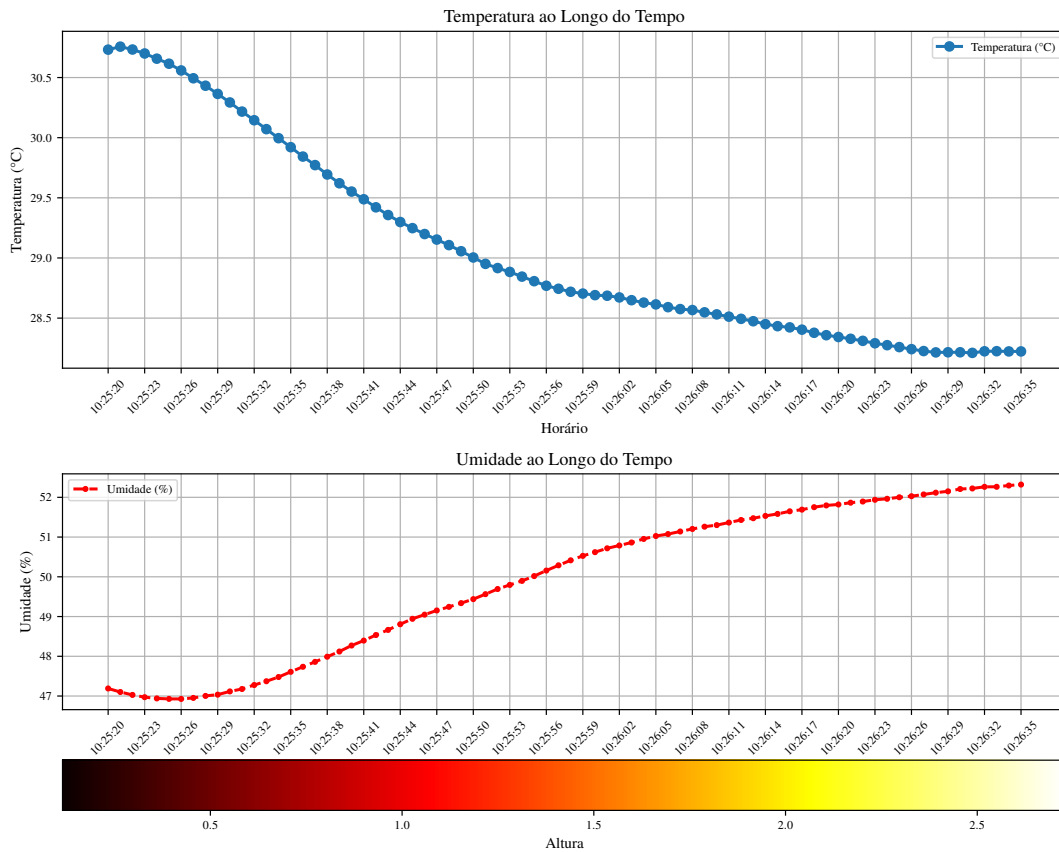
RESULTADOS OBTIDOS

Os fenômenos relacionados a temperatura e ao calor são propriedades de estudo da termodinâmica. Sistemas em equilíbrio estão ligados a caracterização da temperatura, onde as formas de transferência de calor permanecem estáveis em relação ao meio [32, 33].

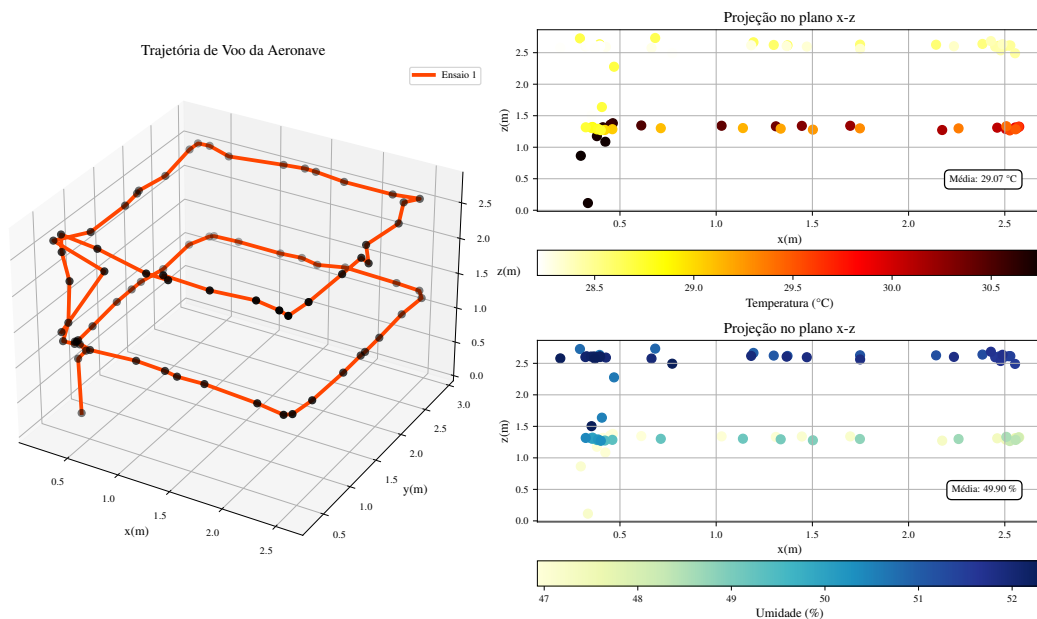
A temperatura mensura a quantidade de calor encontrado na atmosfera fornecendo informações sobre o clima. Por exemplo, a medida que aumentamos a altitude, a densidade do ar abaixa e a temperatura tende a diminuir. Em contraste, a umidade do ar é um sinalizador desse elemento na forma de vapor na atmosfera. Assim, a umidade relativa do ar, é ampliada com a diminuição da temperatura. Isso ocorre, pois para temperaturas mais elevadas o ar torna-se mais seco e com menos umidade.

A unidade sensorial utilizada no projeto captura esses dois elementos. A realização do voo foi feita em diferentes altitudes. Neste sentido, foi possível visualizar o comportamento típico de variação da temperatura e da umidade para diferentes alturas. Foi observado que para altitudes menores a temperatura é maior e a umidade relativa do ar é menor, invertendo esse comportamento à medida que o drone elevava a sua altitude de voo. Foram realizados três ensaios em um ambiente *indoor*. Os pontos da trajetória percorridos pela aeronave *Crazyflie 2.1* são apresentados na Tabela (7). Os resultados dos ensaios podem ser visualizados nos gráficos a seguir. Durante a realização do ensaio 1 o ventilador do laboratório estava ligado, a temperatura do ambiente é menor e a umidade relativa é maior quando comparado aos demais ensaios.

Figura 38 – **Ensaio 1:** A fig.(38a) mostra a temperatura ($^{\circ}\text{C}$) e a umidade (%) em função do horário. A temperatura ($^{\circ}\text{C}$) é reduzida à medida que o drone aumenta a altitude do voo, em contraste, a umidade relativa do ar (%) aumenta. A fig.(38b) mostra a trajetória percorrida pela aeronave *Crazyflie 2.1*, e as projeções no plano x-z. O valor médio da temperatura foi de 29.07°C e a média da umidade relativa do ar foi de 49.90% .



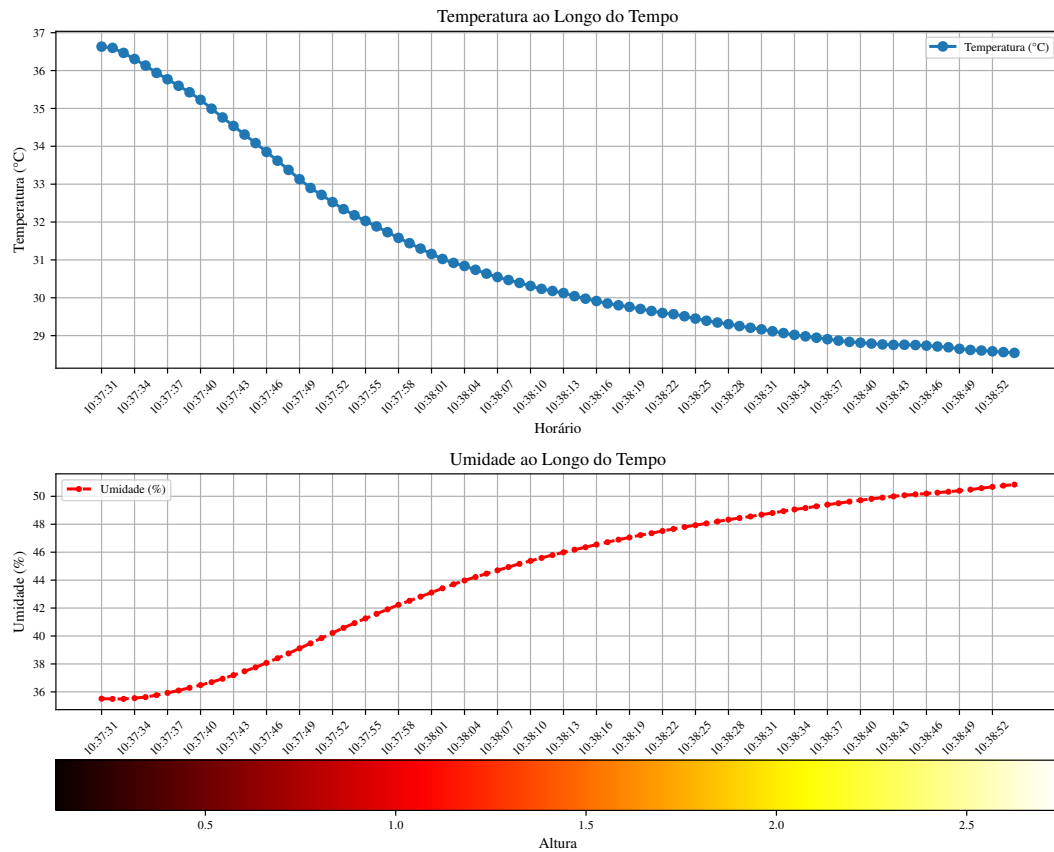
(a) Temperatura ($^{\circ}\text{C}$) e umidade (%) em função do horário da coleta.



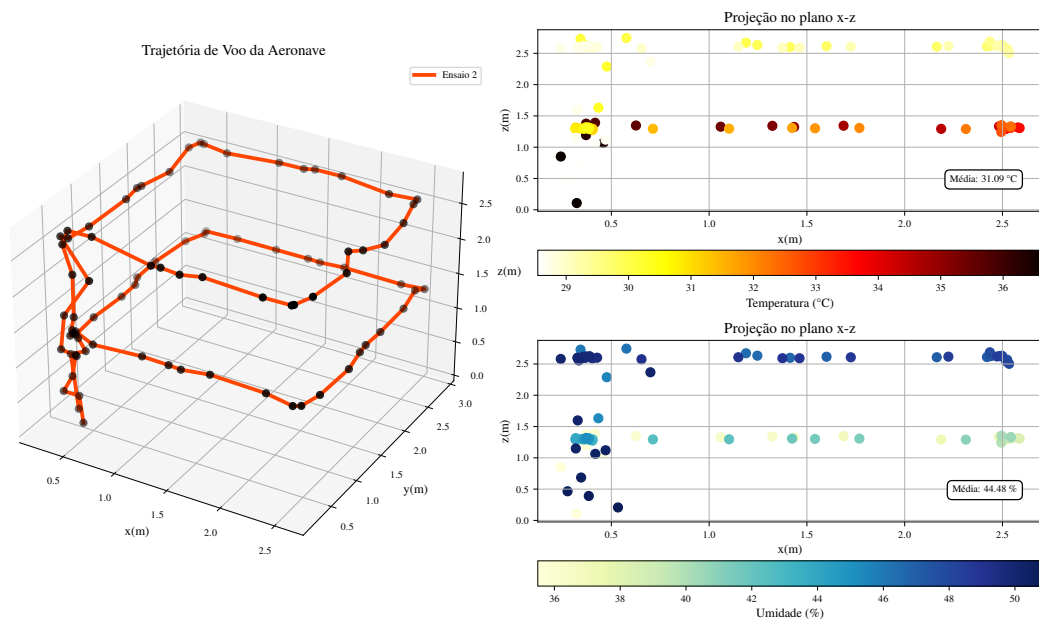
(b) Trajetória percorrida pelo drone.

Fonte: Elaborado pelo autor.

Figura 39 – **Ensaio 2:** A fig.(39a) mostra a temperatura ($^{\circ}\text{C}$) e a umidade (%) em função do horário. A temperatura ($^{\circ}\text{C}$) é reduzida à medida que o drone aumenta a altitude do voo, em contraste, a umidade relativa do ar (%) aumenta. A fig.(39b) mostra a trajetória percorrida pela aeronave *Crazyflie 2.1*, e as projeções no plano x-z. O valor médio da temperatura foi de 31.09°C e a média da umidade relativa do ar foi de 44.48% .



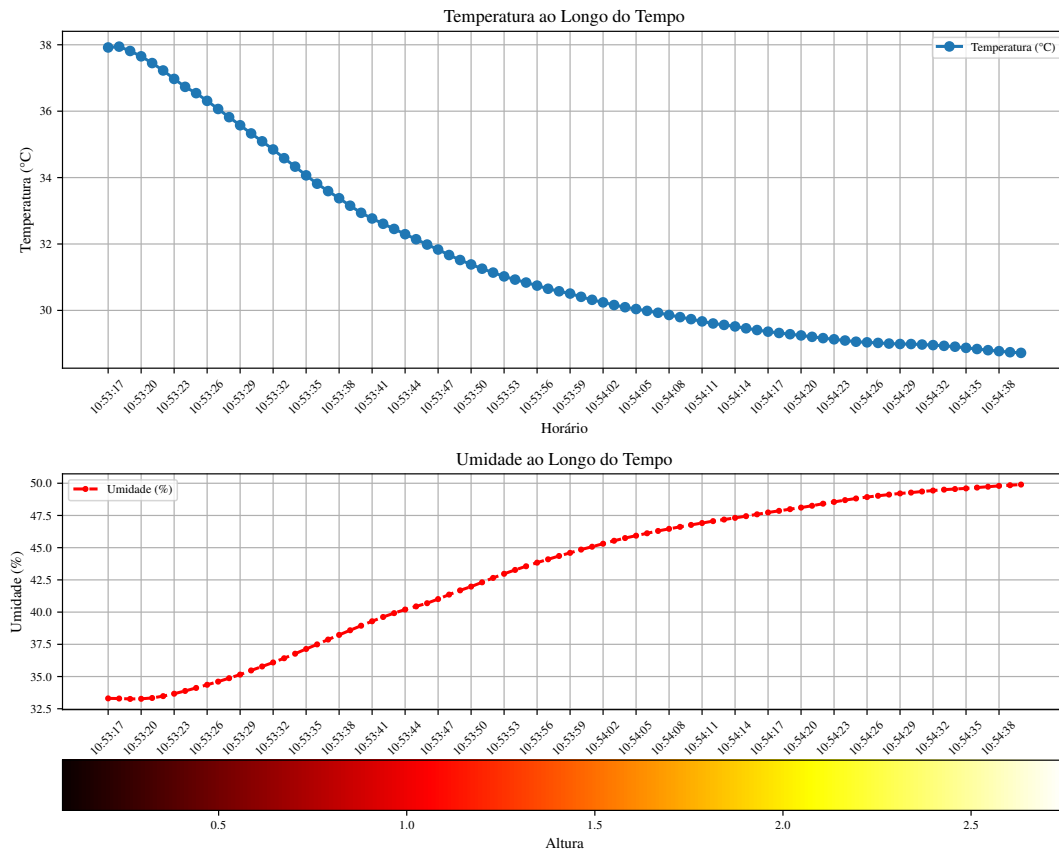
(a) Temperatura ($^{\circ}\text{C}$) e umidade (%) em função do horário da coleta.



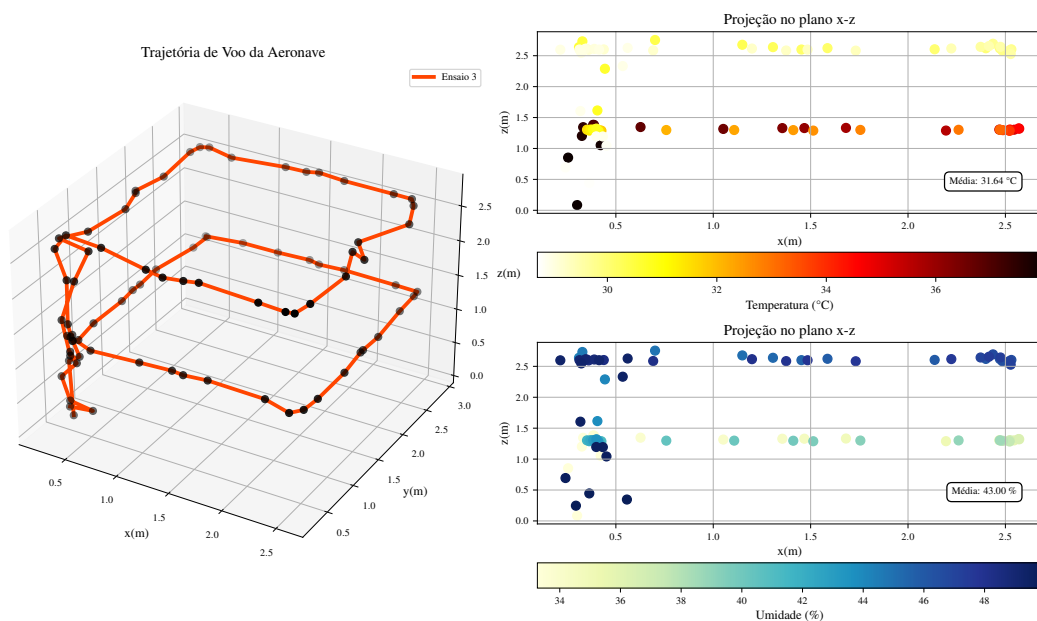
(b) Trajetória percorrida pelo drone.

Fonte: Elaborado pelo autor.

Figura 40 – **Ensaio 3:** A fig.(40a) mostra a temperatura ($^{\circ}\text{C}$) e a umidade (%) em função do horário. A temperatura ($^{\circ}\text{C}$) é reduzida à medida que o drone aumenta a altitude do voo, em contraste, a umidade relativa do ar (%) aumenta. A fig.(40b) mostra a trajetória percorrida pela aeronave *Crazyflie 2.1*, e as projeções no plano x-z. O valor médio da temperatura foi de 31.64°C e a média da umidade relativa do ar foi de 43.00% .



(a) Temperatura ($^{\circ}\text{C}$) e umidade (%) em função do horário da coleta.



(b) Trajetória percorrida pelo drone.

Fonte: Elaborado pelo autor.

CONCLUSÕES & CONSIDERAÇÕES FINAIS

Apresentamos nesta monografia o desenvolvimento de um sistema embarcado contendo a unidade sensorial **AHT10** e a aeronave *Crazyflie 2.1*. Para isso, foi desenvolvido e implementado o *firmware* no *hardware* do drone, sendo a aplicação de baixo nível do sistema, responsável em gerenciar a interatividade do sensor com o veículo utilizando o protocolo de comunicação *I²C*. Para a camada de aplicação foi utilizado rotinas na linguagem *Python* visando automatizar o voo da aeronave, a criação da base de dados dos *logs* de interesse coletados, o tratamento das informações e a criação da página *web* no lado do cliente.

Os resultados obtidos no trabalho representam a fundamentação de algo maior. O sensor utilizado no projeto capta as informações do ambiente de maneira mais acessível, recebendo os dados de forma passiva quando os ensaios são realizados. Entretanto, os resultados apresentados podem ser ampliados, utilizando sensores mais complexos como os desenvolvidos para a detecção de fumaça, gases como o etanol, entre outros, onde os ensaios são mais delicados e regulados visando evitar acidentes. Adicionalmente, a base de dados desenvolvida pode ser utilizada para integrar informações de interesse de outros drones, ampliando o conhecimento sobre o ambiente.

Em suma, o trabalho demonstrou o comportamento típico da variação da temperatura e da umidade relativa do ar em um ambiente *indoor* devido a alteração da altitude. Com isso, foi possível, mesmo em um local restrito, visualizar uma característica presente da atmosfera bastante importante em pesquisas climáticas. Em última análise, o trabalho contribuiu para o tema da indústria 4.0, onde busca-se modificar os processos industriais com a inserção de tecnologias avançadas para a criação de sistemas autônomos com o mínimo de interferência humana.

Referências

- 1 XIANG, T.-Z.; XIA, G.-S.; ZHANG, L. Mini-unmanned aerial vehicle-based remote sensing: Techniques, applications, and prospects. *IEEE geoscience and remote sensing magazine*, IEEE, v. 7, n. 3, p. 29–63, 2019.
- 2 SOUZA, R. M. J. A. de. *Sistema de Identificação de Fonte de Gás Utilizando Nanoquadricóptero*. Tese (Doutorado) — Universidade Federal de Uberlândia, 2022.
- 3 LIMA, G. V. *Planejamento de Trajetórias para Quadricópteros em Tarefas de Perseguição*. Tese (Doutorado) — Universidade Federal de Uberlândia, 2019.
- 4 LONGHITANO, G. A. *VANTs para sensoriamento remoto: aplicabilidade na avaliação e monitoramento de impactos ambientais causados por acidentes com cargas perigosas. 2010. 148 f.* Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo., 2010.
- 5 PAULA, A. N. de. *Implementação de um Sistema de Sensoriamento em Vants para a Manutenção Preditiva e Atuação em Emergências no Contexto da Indústria 4.0*. Universidade Federal de Uberlândia: Relatório de Iniciação Científica, 2023.
- 6 TOCCI, R. J. *Sistemas digitais: princípios e aplicações*. São Paulo, Brasil: Pearson Prentice Hall, 2011.
- 7 CHASE, O.; ALMEIDA, F. Sistemas embarcados. *Mídia Eletrônica. Página na internet: <www.sbajovem.org/chase>, capturado em*, v. 10, n. 11, p. 13, 2007.
- 8 STALLINGS, W. *Computer Organization and Architecture: Designing for Performance*. One Lake Street Upper Saddle River, NJ United States: Prentice Hall Press, 2013.
- 9 SOUSA, H. L. de. Sensoriamento remoto com vants: uma nova possibilidade para a aquisição de geoinformações. *Revista Brasileira de Geomática*, v. 5, n. 3, p. 326–342, 2017.
- 10 SANTOS, B. P. et al. Indústria 4.0: desafios e oportunidades. *Revista Produção e Desenvolvimento*, v. 4, n. 1, p. 111–124, 2018.
- 11 FERNÁNDEZ-CARAMÉS, T. M. et al. Towards an autonomous industry 4.0 warehouse: A uav and blockchain-based system for inventory and traceability applications in big data-driven supply chain management. *Sensors*, MDPI, v. 19, n. 10, p. 2394, 2019.
- 12 ROSSI, M.; BRUNELLI, D. Autonomous gas detection and mapping with unmanned aerial vehicles. *IEEE Transactions on Instrumentation and Measurement*, IEEE, v. 65, n. 4, p. 765–775, 2015.
- 13 OLIVEIRA, A. M. et al. Detection of volatile organic compounds (vocs) in indoor environments using nano quadcopter. *Drones*, MDPI, v. 7, n. 11, p. 660, 2023.

- 14 RODRIGUES, M. T. et al. Levantamento topográfico por meio de veículo aéreo não tripulado (vant). *Energia na Agricultura*, v. 33, n. 4, p. 367–372, 2018.
- 15 DINIZ, M. T. M.; MEDEIROS, J. B. de. Mapeamento de focos de reprodução de *aedes aegypti* na cidade de caicó/rn com o auxílio de veículo aéreo não tripulado (mapping of breeding sites of *aedes aegypti* in caicó/rn city with use of unmanned aerial vehicle). *Revista GeoNordeste*, n. 2, p. 196–207, 2018.
- 16 BARCELOS, A. C. *O Uso de Veículo Aéreo não Tripulado (VANT) em Monitoramentos de Campo: Aplicabilidades e Viabilidades*. Universidade Federal de Uberlândia: Monografia, 2017.
- 17 S, A.; BOS, H. *Sistemas Operacionais Modernos*. Bookman, 2024. ISBN 9788582606179. Disponível em: <<https://books.google.com.br/books?id=F3LuEAAQBAJ>>.
- 18 CAMARA, R. *Resumo do Protocolo de Comunicação I2C*. 2013. Disponível em: [Protocolo de Comunicação I2C](#). Acesso: 01 de setembro de 2023.
- 19 VALDEZ, J. B. J. *A Basic Guide to I2C*. 2015. Disponível em: [Understanding the I2C Bus](#). Acesso: 01 de setembro de 2023.
- 20 CAMPBELL, S. *BASICS OF THE I2C COMMUNICATION PROTOCOL*. 2016. Disponível em: [Basic of the I2C](#). Acesso: 14 de março de 2024.
- 21 SCHWARZ, R. *Understanding I2C*. 2023. Disponível em: [Videoaula: Understanding I2C](#). Acesso: 14 de março de 2024.
- 22 AGUIRRE, L. *Fundamentos de Instrumentação*. Pearson Universidades, 2013. ISBN 9788581431833. Disponível em: <<https://books.google.com.br/books?id=R4xf0AEACAAJ>>.
- 23 ASAIR. *AHT10 Technical Manual*. Disponível em: [Datasheet: Sensor AHT10](#). Acesso: 25 de março de 2024.
- 24 ASAIR. *AHT20 Technical Manual*. Disponível em: [Datasheet: Sensor AHT20](#). Acesso: 25 de março de 2024.
- 25 BITCRAZE. *Crazyflie 2.1*. 2023. Disponível em: [Bitcraze.io](#). Acesso: 20 de setembro de 2023.
- 26 KUROSE, J. F.; ROSS, K. W. *Redes de Computadores e a Internet: uma abordagem top-down*. São Paulo, Brasil: Pearson Education do Brasil, 2013.
- 27 NUSSENZVEIG, H. M. *Curso de Física Básica*. São Paulo, Brasil: Edgard Blücher, 2002.
- 28 HIBBELER, R. C. *Estática: mecânica para engenharia*. São Paulo, Brasil: Pearson Prentice Hall, 2011.
- 29 ÇENGEL, J. M. C. Y. A. *Mecânica dos fluidos: fundamentos e aplicações*. [S.l.]: AMGH Editora, 2015. ISBN 9788580554915.
- 30 BITCRAZE. *Crazyflie 2.1 Technical Manual*. Disponível em: [Datasheet: Crazyflie 2.1](#). Acesso: 03 de abril de 2024.

- 31 ALMEIDA, R. M. A. de. *FreeRTOS: Gestão de Tasks*. 2021. Disponível em: [Videoaula: FreeRTOS](#). Acesso: 27 de fevereiro de 2024.
- 32 NUSSENZVEIG, H. M. *Curso de Física Básica: fluidos, oscilações e ondas, calor*. [S.l.]: Editora Blucher, 2018. v. 2.
- 33 INCROPERA, F. et al. *Fundamentos De Transferência De Calor E De Massa*. LTC, 2014. ISBN 9788521625049. Disponível em: <https://books.google.com.br/books?id=HOUfvgAACAAJ>.

Apêndices

APÊNDICE A

FERRAMENTAS COMPUTACIONAIS ÚTEIS AO DESENVOLVIMENTO DO PROJETO

Para o desenvolvimento do projeto foi necessário a instalação do repositório da aeronave *Crazyflie 2.1* em um sistema operacional. Foi utilizado uma distribuição Linux, chamada Ubuntu, versão 20.04.6 *LTS* de 64 *bits*. A escolha desse SO foi feita devido a facilidade em seu uso e ao controle eficaz dos processos do *software* na comunicação via rádio entre o computador e a aeronave *Crazyflie 2.1*. Adicionalmente, o Ubuntu possui suporte, compatibilidade com as ferramentas necessárias para a execução do projeto e uma comunidade ativa para sanar quaisquer eventuais dúvidas sobre o sistema operacional.

A instalação do *Crazyflie client* foi necessário para o desenvolvimento do projeto. Isso exigia a versão 3 da linguagem de programação *Python*, *git*, *pip* e a biblioteca de interface gráfica Qt[†]. Desta forma, foi utilizado os comandos a seguir, executados no terminal do Ubuntu:

- 1) `sudo apt install git python3-pip libxcb-xinerama0`
- 2) `pip3 install - --upgrade pip`
- 3) `pip3 install cfclient`

Para clonar localmente o repositório via *git*, foi executado o comando a seguir:

- 4) `git clone https://github.com/bitcraze/crazyflie-clients-python`

A instalação do *Crazyflie client* pode ser feita via o pacote *snap*, porém recomendasse

[†]
Consultar as seguintes *webpages* para mais informações sobre *Python* e *git*.


```

#
# Copyright (C) 2014 Bitcraze AB
#
# Crazyflie Nano Quadcopter Client
#
# This program is free software; you can redistribute it and/or
# modify it under the terms of the GNU General Public License
# as published by the Free Software Foundation; either version 2
# of the License, or (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <https://www.gnu.org/licenses/>.
"""
Simple example that connects to the first Crazyflie found, ramps up/down
the motors and disconnects.
"""
import logging
import time
from threading import Thread

import cflib
from cflib.crazyflie import Crazyflie
from cflib.utils import uri_helper

uri = uri_helper.uri_from_env(default='radio://0/90/2M/E7E7E7E7E7')

logging.basicConfig(level=logging.ERROR)

class MotorRampExample:
    """Example that connects to a Crazyflie and ramps the motors up/down and
    the disconnects"""

    def __init__(self, link_uri):
        """ Initialize and run the example with the specified link_uri """

        self._cf = Crazyflie(rw_cache='./cache')

        self._cf.connected.add_callback(self._connected)
        self._cf.disconnected.add_callback(self._disconnected)
        self._cf.connection_failed.add_callback(self._connection_failed)
        self._cf.connection_lost.add_callback(self._connection_lost)

        self._cf.open_link(link_uri)

        print('Connecting to %s' % link_uri)

    def _connected(self, link_uri):
        """ This callback is called form the Crazyflie API when a Crazyflie
        has been connected and the TOCs have been downloaded."""

        # Start a separate thread to do the motor test.

```

```

        # Do not hijack the calling thread!
        Thread(target=self._ramp_motors).start()

    def _connection_failed(self, link_uri, msg):
        """Callback when connection initial connection fails (i.e no
        Crazyflie
        at the specified address)"""
        print('Connection to %s failed: %s' % (link_uri, msg))

    def _connection_lost(self, link_uri, msg):
        """Callback when disconnected after a connection has been made (i.e
        Crazyflie moves out of range)"""
        print('Connection to %s lost: %s' % (link_uri, msg))

    def _disconnected(self, link_uri):
        """Callback when the Crazyflie is disconnected (called in all cases)
        """
        print('Disconnected from %s' % link_uri)

    def _ramp_motors(self):
        thrust_mult = 1
        thrust_step = 500
        thrust = 20000
        pitch = 0
        roll = 0
        yawrate = 0

        # Unlock startup thrust protection
        self._cf.commander.send_setpoint(0, 0, 0, 0)

        while thrust >= 20000:
            self._cf.commander.send_setpoint(roll, pitch, yawrate, thrust)
            time.sleep(0.1)
            if thrust >= 25000:
                thrust_mult = -1
                thrust += thrust_step * thrust_mult
            self._cf.commander.send_setpoint(0, 0, 0, 0)
            # Make sure that the last packet leaves before the link is closed
            # since the message queue is not flushed before closing
            time.sleep(0.1)
            self._cf.close_link()

if __name__ == '__main__':
    # Initialize the low-level drivers
    cflib.crtp.init_drivers()

    le = MotorRampExample(uri)

```

APÊNDICE B

FIRMWARE DESENVOLVIDO PARA A COMUNICAÇÃO ENTRE A UNIDADE SENSORIAL E A AERONAVE *CRAZYFLIE 2.1*

```

#define DEBUG_MODULE "aht10Deck" // Macro (objeto) para o deckdriver

// Bibliotecas
#include "debug.h"
#include "i2cdev.h"
#include "deck.h"
#include "log.h"
#include "FreeRTOS.h"
#include "task.h"
#include "system.h"
#include "param.h"
#include "pm.h"
#include "math.h"

// Endereços e constantes para o sensor AHT10
#define AHT10I2CAddr 0x38 // Endereço de Identificação I2C do Sensor AHT10
#define AHT10I2CAddr_Alternate 0x39 // Endereço Alternativo I2C do Sensor
    AHT10
#define AHT10_CMD_CALIBRATE 0xE1 // Comando de Calibração
#define AHT10_INIT_CMD 0xBE // Comando de inicialização do sensor
#define AHT10_CMD_TRIGGER 0xAC // Comando de leitura TRIGGER
#define AHT10_CMD_SOFTRESET 0xBA // Comando Soft reset
#define AHT10_STATUS_BUSY 0x80 // Status de bit ocupado
#define AHT10_STATUS_CALIBRATED 0x08 // Status de bit para calibração
#define AHT10_TASK_STACKSIZE (2 * configMINIMAL_STACK_SIZE) // Tamanho da
    Pilha
#define AHT10_TASK_PRI 3 // Prioridade da Task
#define AHT10_TASK_NAME "AHT10" // Nome de Identificação da Task

// Variáveis globais

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

```

static bool isInit;
static float temp; // temperatura
static float humidity; // umidade
void aht10Task(void* arg);
static uint8_t INIT_CMD = AHT10_INIT_CMD; // Endereço de inicialização do
    sensor AHT10
static uint8_t triggerCommand = AHT10_CMD_TRIGGER; // Endereço do trigger
static uint8_t data[6]; // Número inteiro sem sinal: Vetor de 6 posições

// Deck driver init funtion
static void aht10Init()
{
    if(isInit)
        return;

    DEBUG_PRINT("\nIniciando AHT10...\n");
    i2cdevInit(I2C1_DEV); // Ponteiro para o periférico I2C
    isInit = true;
    if(i2cdevWriteReg8(I2C1_DEV, (uint8_t)AHT10I2CAddr, I2CDEV_NO_MEM_ADDR, 1,
        &INIT_CMD)){
        DEBUG_PRINT("Barramento Inicializado\n");
    }else{
        DEBUG_PRINT("Falha na Inicialização do Barramento I2C\n");
    }
    DEBUG_PRINT("Inicialização completa: AHT10!\n");
    vTaskDelay(M2T(10));
}

void appMain(void* arg) // Criação da Task com parâmetros do sensor AHT10
{
    // Retorna uma struct
    xTaskCreate(aht10Task, // Ponteiro para a função que implementa a Task
        AHT10_TASK_NAME, // Nome (string) da Task
        AHT10_TASK_STACKSIZE, // Tamanho (words) da stack alocada para a Task
        NULL, // Parâmetro passado para a Task na inicialização
        AHT10_TASK_PRI, // Prioridade da Task
        NULL); // Handle (Guia para Task "identificador")
}

static bool aht10Test()
{
    DEBUG_PRINT("AHT10 test passed!\n");
    return true;
}

void aht10Task(void* arg) // Função Principal
{
    systemWaitStart(); // Inicialização
    TickType_t xLastWakeTime; // Contador de tempo do FreeRTOS (TickType_t ->
        último registro de tempo)

    xLastWakeTime = xTaskGetTickCount();

    while(1) // Loop Infinito -> A Task fica presa nesse loop
    {

```

```

    i2cdevWriteReg8(I2C1_DEV, AHT10I2CAddr, I2CDEV_NO_MEM_ADDR, 1, &
triggerCommand)
    vTaskDelayUntil(&xLastWakeTime, M2T(200)); // Delay sincronizado (200ms):
    Aguarda esse tempo para o sensor coletar os dados

    i2cdevReadReg8(I2C1_DEV, AHT10I2CAddr, I2CDEV_NO_MEM_ADDR, (uint8_t)6,
data);

    // Extrair a temperatura e a umidade dos dados crus
    // Operação bit a bit

    uint32_t raw_temp = (data[3] & 0x0F); // and e máscara
    raw_temp <= 8; // deslocamento para esquerda
    raw_temp |= data[4]; // or
    raw_temp <= 8;
    raw_temp |= data[5];
    temp = ((raw_temp * 200.0) / 1048576.0) - 50.0; // datasheet do sensor

    uint32_t raw_humidity = data[1];
    raw_humidity <= 8;
    raw_humidity |= data[2];
    raw_humidity <= 4;
    raw_humidity |= data[3] >> 4;
    humidity = (raw_humidity * 100.0) / 1048576.0; // datasheet do sensor

    // Crazyflie Client
    DEBUG_PRINT("Temperatura: [%.1f], umidade relativa: [%.1f]\n", (double)
temp, (double)humidity);

    // Aguardar um intervalo antes da próxima leitura
    vTaskDelay(M2T(1000)); // Delay
}
}

static const DeckDriver aht10Driver = {
    .name = "myaht10",
    .init = aht10Init,
    .test = aht10Test,
};

// Logs de interesse
DECK_DRIVER(aht10Driver);
LOG_GROUP_START(AHT10)
LOG_ADD(LOG_FLOAT, temperature, &temp)
LOG_ADD(LOG_FLOAT, humidity, &humidity)
LOG_GROUP_STOP(AHT10)

```


Anexos

DATASHEET DO SENSOR AHT10

AHT10 Technical Manual

Temperature and humidity sensor

- Fully calibrated
- Digital output, I²C interface
- Excellent long-term stability
- SMD package reflow solderable
- Fast response and strong anti-interference ability

ASAIR®



Product overview

AHT10, the new generation of temperature and humidity sensors sets a new standard in size and intelligence: it is embedded for reflow soldering. The dual-row flat leadless SMD package has a 4x5mm bottom and a height of 1.6mm. The sensor outputs a calibrated digital signal in standard I2C format. The AHT10 is equipped with a newly designed ASIC- specific chip, an improved MEMS semiconductor capacitive humidity sensing element and a standard on-chip temperature sensing element. Its performance has been greatly improved beyond the reliability level of the previous generation of sensors. The first generation of temperature and humidity sensors have been improved to make them more stable in harsh environments.

Each sensor is calibrated and tested with a product lot number printed on the surface of the product. Thanks to improved and miniaturized sensors, it is more cost-effective and ultimately all equipment will benefit from cutting-edge energy-saving operating modes.

Application range

HVAC, dehumidifiers, testing and testing equipment, consumer goods, automobiles, automatic control, data logger, weather home appliances, humidity regulation, medical and other related temperature and humidity detection and control.

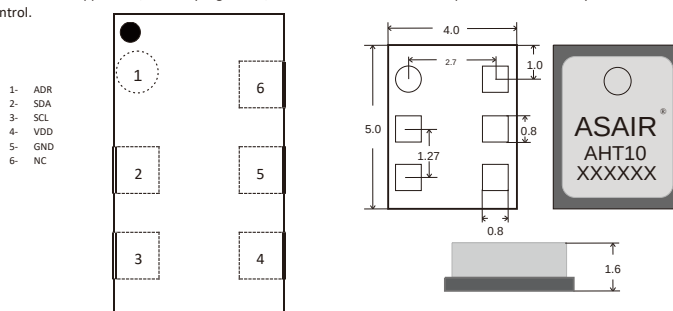


Figure 1: AHT10 sensor package diagram (unit: mm tolerance: 0.1mm)

Sensor performance

Relative humidity

parameter	condtn	min	typ	max	unit
Resolution	typical		0.024		%RH
Accuracy	typical		±2		%RH
	maximum	See Figure 2			%RH
Repeatability			±0.1		%RH
Hysteresis			±1		%RH
Nonlinear			<0.1		%RH
Response	t 63%		8		S
scope	extended	0		100	%RH
Drift	normal		<0.5		%RH/yr

table 1 humidity characteristics table $\Delta RH(\%RH)$

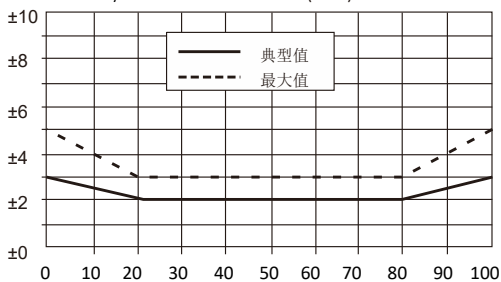


Figure 2 The maximum error of relative humidity at 25 °C

1. This accuracy is the test accuracy of the sensor at a factory voltage of 3.3V at 25 °C. This value does not include hysteresis and nonlinearity and only applies to Non-condensing conditions. ² The time required to achieve a first-order response of 63% at 25 °C and 1 m / s airflow.
2. Normal operating range: 0-80% RH, beyond this range, sensor readings will be biased (drift <3% RH after 200 hours at 90% RH humidity). The scope of work is further limited to -40~ 80 °C.
3. If there are volatile solvents, irritating tapes, adhesives, and packaging materials around the sensor, the readings may be high. Please refer to the relevant documents for detailed instructions.
4. The minimum and maximum values of supply current and power are based on VDD = 3.3V and T < 60 °C conditions. The average is the value measured once every two seconds.

Temperature

parameter	condtn	min	typ	max	unit
Resolution	typical		0.01		°C
Accuracy	typical		±0.3		°C
	maximum	See Figure 3			°C
Repeatability			±0.1		°C
Hysteresis			±0.1		°C
Response	t 63%	5		30	S
scope	extended ¹	-40		85	°C
drift			<0.04		°C/yr

table 3 temperature characteristics table

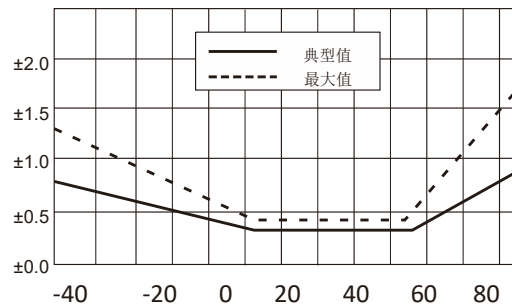


Figure 3 Typical temperature error and maximum error

5. The response time depends on the thermal conductivity of the sensor substrate.
6. Contact surface means The metal layer on the PCB where the SMD pads are soldered.
7. Solder mask The top layer of the PCB covers the insulating layer on the connecting line.
8. The type of solder is related to the size of the internal particles of the solder. Type 3 sizes range from 25 – 45 μm powder.
9. 75% RH can be easily saturated NaCl is generated.

electrical characteristics

parameter	condition	min	typ	max	unit
Supply voltage	typical	1.8	3.3	3.6	V
Supply current,I _{DD}	Sleep	-		0.25	μA
	measuring		23		μA
Power	Sleep	-		0.9	μW
	measuring		0.07		mW
	average	-	3.3	-	μW
communication	Two-wire digital interface , standard I ² C protocol				

Table 2 Electrical characteristics

packaging information

Sensor model	package	Quantity
AHT10	Tape and reel packaging	4000PCS/ volume

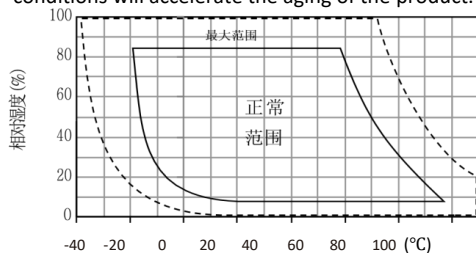
Table 4 package information.

AHT10 User Guide

11 Expansion performance

1.1 Working conditions

The sensor has stable performance within the recommended operating range, see Figure 4 . Long-term exposure to conditions outside the normal range, especially at humidity >80 %, may result in temporary signal drift (drift after 60hours +3% RH). When returning to normal operating conditions, the sensor will slowly self-recover to the calibration state. Refer to Section 2.3, “ Recovery Processing ” to speed up the recovery process. Prolonged use under abnormal conditions will accelerate the aging of the product.



1.2 RH accuracy at different temperatures

The RH accuracy at 25 °C is defined in Figure 2 , and the maximum humidity error in other temperature segments is shown in Figure 5 .

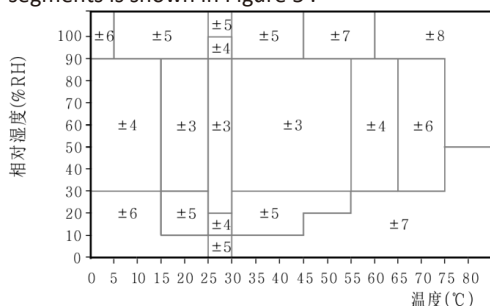


Figure The maximum humidity error in the range of 0~80 °C, unit : (%RH) Please note : The above error is the maximum error (excluding hysteresis) of the reference instrument test with high precision dew point meter . The maximum error of $\pm 3\%$ RH typical range of error $\pm 2\%$ RH, in other ranges, typically 1/2 of the maximum error value.

1.3 Electrical characteristics

The power consumption given in Table 1 is related to the temperature and supply voltage VDD . About power estimation and see Figure 6 7 . Please pay attention to the picture 6 and The curve in 7 is a typical natural characteristic and there may be deviations.

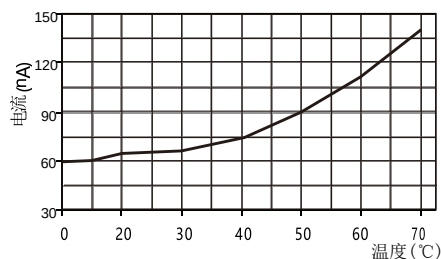


Figure 6 VDD = 3.3V , typical supply current vs. temperature (sleep mode).

Please note that these data and display values are approximately $\pm 25\%$ deviation.

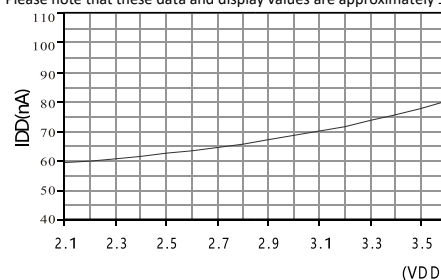


Figure 7 shows the relationship between typical supply current and supply voltage at a temperature of 25 °C . Curve (sleep mode). Please note that these data deviate from the displayed values. Can reach $\pm 50\%$ of the displayed value . At 60 °C , the coefficient is about 15

(Compared with Table 2).

2 Application information

2.1 Welding instructions

SMD 's I/O pads are made of copper leadframe planar substrates, except that these pads are exposed to the outside for mechanical and electrical connections. When used, both the I/O pad and the exposed pad need to be soldered to the PCB . To prevent oxidation and optimize soldering, the solder joints on the bottom of the sensor are plated with Ni/Au.

On the PCB, I / O contact surface length should / O pads package is larger than the AHT10 I 0.2mm, to the portion on the inner side of the I / O pads to match the shape, and the width of the lead width ratio SMD pads of the package 1:1, see Figure 8 .

For stencil and solder mask design [7] It is recommended to use a copper foil definition pad (SMD) with a solder mask opening larger than the metal pad .

For the SMD pad, if the gap between the copper foil pad and the solder resist layer is $60\text{ }\mu\text{m}$ - $75\text{ }\mu\text{m}$, the solder mask opening size should be larger than the pad size $120\text{ }\mu\text{m}$ - $150\text{ }\mu\text{m}$. The rounded portion of the package pad is matched to the corresponding round solder mask opening to ensure that there is sufficient solder mask area (especially at the corners) to prevent solder from intersecting. Each pad must have its own solder mask opening to form a solder mask network around the adjacent pads.

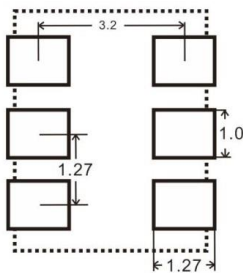


Figure 8 recommends the AHT10 PCB design dimensions (unit: mm), peripheral The dotted line is the outer dimensions of the SMD package.

For solder printing, it is recommended to use a laser-cut stainless-steel mesh with an electronically polished trapezoidal wall. The recommended mesh thickness is 0.125mm . The stencil size for the pad portion must be 0.1 mm longer than the PCB pad .

And placed at a distance of 0.1mm from the center of the package . The stencil of the exposed pad is covered

70%-90% of the pad area — that is, $1.4\text{mm} \times 2.3\text{mm}$ at the center of the heat sink area . Due to the low placement height of SMD , it is recommended to use no-clean type 3 solder. and purify with nitrogen during reflow.

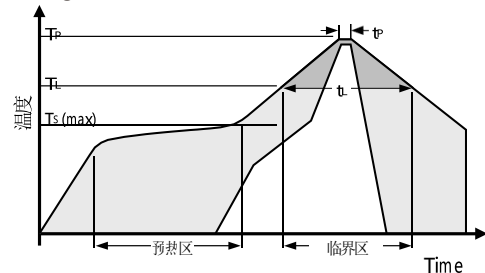


Figure 9 JEDEC standard welding process diagram, $T_p \leq 260\text{ }^\circ\text{C}$, $t_p < 30\text{ sec}$, lead-free soldering Pick up. $T_L < 220\text{ }^\circ\text{C}$, $t_L < 150\text{ sec}$, the rate of temperature rise and fall during welding Should be $< 5\text{ }^\circ\text{C/sec}$.

The AHT10 can be soldered using a standard reflow oven . The sensor is fully compliant with the IPC/JEDEC J-STD-020D soldering standard. The contact time should be less than 40 seconds at temperatures up to $260\text{ }^\circ\text{C}$ (see Figure 9) ; the ultimate soldering temperature that the sensor can withstand is $260\text{ }^\circ\text{C}$.

Note: After reflow soldering, the sensor should be stored in a $>75\%$ RH environment for at least 12 hours to ensure rehydration of the polymer. Failure to do so will cause the sensor reading to drift. It is also possible to place the sensor in a natural environment ($>40\%$ RH) for more than 5 days to rehydrate it. Use low temperature reflow soldering

(Example : $180\text{ }^\circ\text{C}$) can reduce hydration time.

Do not flush the board after soldering. Therefore, customers are advised to use " no-clean " solder paste. If the sensor is used in corrosive gases or if condensed water is produced (eg, in a high-humidity environment), both the lead pad and the PCB need to be sealed (eg, using a conformal coating) to avoid poor contact or short circuit.

2.2 Storage conditions and operating instructions

Humidity sensitivity level (MSL) is 1, based on

IPC/JEDEC J-STD-020 standard. Therefore, it is recommended to use it within one year after shipment.

Humidity sensors are not ordinary electronic components and require careful protection. This must be taken seriously by users. Prolonged exposure to high concentrations of chemical vapor will cause the sensor reading to drift. It is recommended to store the sensors ESD original package comprises a sealed bag, and meet the following conditions:



Sensors should be avoided during production and transportation

High concentrations of chemical solvents and prolonged exposure.

Avoid contact with volatile glue, tape, stickers or volatile packaging materials. Such as foam, foam and so on. The production area should be well ventilated.

2.3 recovery processing

As described above, if the sensor is exposed to extreme conditions or chemical vapors, readings generated drift. It can be restored to the calibration state by the following processing.

Drying: maintained at 80-85 °C and <5% RH humidity for 10 hours;

Rehydration: 12 hours at 20-30 °C and >75% RH humidity.

2.4 temperature effects

The relative humidity of a gas depends to a large extent on temperature. Therefore, when measuring humidity, it should be ensured that all sensors measuring the same humidity work at the same temperature. When testing, ensure that the sensor under test and the reference sensor are at the same temperature and then compare the humidity readings.

If the sensor is on the same printed circuit board as the heat-generating electronic components, measures should be taken to minimize the effects of heat transfer when designing the circuit.

Temperature range 10 °C -50 °C (0-85 deg.] C for a limited time); humidity of 20-60% RH (no ESD Package Sensor). For sensors that have been removed from the original packaging, we recommend storing them in the inner metal.

Anti-static bag made of PET/AL/CPE material.

In addition, when the measurement frequency is too high, the sensor's own temperature will rise and affect the measurement accuracy. If it is to ensure that its own temperature rise is less than 0.1 °C, the activation time of AHT10 should not exceed the measurement time. 10% - recommended every 2 seconds measurement data once.

2.5 Materials for sealing and packaging

Many materials absorb moisture and act as a buffer, which increases response time and hysteresis. Therefore, the material around the sensor should be carefully selected. The recommended materials are: metal, LCP, POM (Delrin), PTFE (Teflon), PE, PEEK, PP, PB, PPS, PSU, PVDF, PVF.

For sealing and

gluing (the conservative recommendation): recommended package, or an electronic component using a silicone resin-filled epoxy. The gases released by these materials may also contaminate AHT10 (see 2.2). Therefore, the sensor should be assembled last and placed in a well-ventilated area or dried in an environment > 50 °C for 24 hours to allow the release of contaminated gases prior to packaging.

2.6 wiring rules and signal integrity

If the SCL and SDA signal lines are parallel and very close to each other, it may cause signal crosstalk and communication failure. The solution is to place VDD and / or GND between

For example, to maintain good ventilation of the outer casing, the copper plating of the AHT10 and other parts of the printed circuit board should be as small as possible, or a gap should be left between the two. (See Figure 10).

the two signal lines, separate the signal lines, and use shielded cables. In addition, lower The SCL frequency may also improve the integrity of the signal transmission. A100nF decoupling capacitor must be added between the power supply pins (VDD , GND)for filtering. This capacitor should be as close as possible to the sensor. See the next chapter.

3 Interface definition

Pin	name	Interpretation
1	ADR	Power ground
2	SDA	Serial data
3	SCL	Serial clock
4	VDD	Supply voltage
5	GND	Power ground
6	NC	Stay vacant

Table 5 AHT10 pin distribution (top view).

3.1 Power supply pin (VDD, GND)

AHT10 power supply range of 1.8-3.6V, recommended voltage is 3.3V. A 100nF decoupling capacitor must be connected between the power supply (VDD) and ground (GND) and the capacitor should be placed as close as possible to the sensor - see Figure 11.

3.2 Serial clock SCL

SCL is used for communication synchronization between the microprocessor and the AHT10. Since the interface contains completely static logic, there is no minimum SCL frequency.

3.3 serial data SDA

The SDA pin is used for data input and output of the sensor. SDA is active on the rising edge of the serial clock (SCL) when a command is sent to the sensor, and SDA must remain stable when SCL is high. After the falling edge of SCL, the SDA value can be changed. To ensure communication security, the effective time of the SDA should be extended to TSU and THO before and after the rising edge of SCL - see Figure 12. When reading data from the sensor, SDA is low after SCL active (TV), and maintained until the falling edge of SCL.

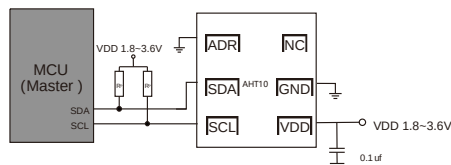


Figure 11 shows a typical application circuit that includes a pull-up resistor R_P and a decoupling capacitor between VDD and GND.

Note : 1. The power supply voltage of the host MCU must be consistent with the sensor when the product is in use.

2, To further improve the reliability of the system, the sensor power can be controlled.

3, Only a single AHT10 can be connected to the I2C bus and no other I2C devices can be connected.

To avoid signal collisions, the microprocessor (MCU) must only drive SDA and SCL low. An external pull-up resistor (eg 10k Ω) is required to pull the signal high. Pull-up resistors are usually already included in the microprocessor In the I/O circuit. Reference table 7 and table 8 can get detailed information about sensor input / output characteristics.

4 Electrical characteristics

4.1 Absolute maximum rating

The electrical characteristics of AHT10 are defined in Table 1. The absolute maximum ratings given in Table 6 are stress ratings only and provide more information. Under such conditions, it is not advisable for the device to perform functional operations. Prolonged exposure to absolute maximum ratings may affect sensor reliability.

parameter	Minimum	maximum	unit
VDD to GND	-0.3	3.6	V
Digital I/O pin(SDA, SCL) toGND	-0.3	VDD +0.3	V
Input current per pin	-10	10	mA

Table 6 Electrical Absolute Maximum Ratings

ESD electrostatic discharge meets JEDEC JESD22-A114 standard (Human Body Mode $\pm 4kV$), JEDEC JESD22-A115. (machine mode $\pm 200V$). If the test conditions exceed the nominal limit, the sensor requires an additional protection circuit.

4.2 Input / output characteristics

Electrical characteristics, such as power consumption, high and low voltages of the input and output, etc., depend on the power supply voltage. In order to make the sensor communication smooth, it is important to ensure that the signal design is strictly limited to the ranges given in Tables 7, 8, and 12.

parameter	condition	Minimum	typical	maximum	unit
Output low voltage VOL	VDD = 3.3V, -4mA < IOL < 0mA	0	-	0.4	V
Output high voltage VOH		70% VDD	-	VDD	V
Output sink current IOL		-	-	-4	mA
Input low voltage VIL		0	-	30% VDD	V
Input high voltage VIH		70% VDD	-	VDD	V
Input Current	VDD = 3.6 V, VIN=0V to 3.6V	-	-	±1	uA

Table 7 DC characteristics of the digital input and output pads, if there is no special statement, VDD = 1.8 V to 3.6 V, T = -40 ° C to 85 ° C.

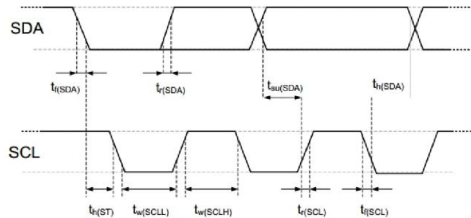


Figure 12 shows the timing diagram and abbreviations of the digital input / output terminals in Table 8.

Explanation. The thicker SDA line is controlled by the sensor, the normal SDA line

Controlled by a single chip microcomputer. Please note that the effective read time of SDA is from the previous one.

Triggered by the falling edge of the conversion.

parameter	Label	I ² C typical mode		I ² C high speed mode		unit
		MIN	MAX	MIN	MAX	
I ² C clock frequency	fSCL	0	100	0	400	KHz
Start signal time	tHDST A					μs
SCL clock high level width	tHIGH	4.7		1.3		μs
SCL clock low width	tLOW	4.0		0.6		μs
Data retention time relative to the SCL SDA edge	tHDDA T	0.09	3.45	0.02	0.9	μs
Data setup time relative to the SCL SDA edge	tSUDA T	250		100		μs

Note : Measurements for both pins are from 0.2VDD and 0.8VDD.

Note : The above I²C timing is determined by the following internal delay:

- (1) The internal SDA input pin is delayed relative to the SCK pin, typically 100ns
- (2) The internal SDA output pin is delayed relative to the SCK falling edge, typically 200ns

Table 8 Timing characteristics of the I²C fast mode digital input / output. The specific meaning is shown in Figure 12. Unless otherwise stated.

5 Sensor communication

The AHT10 communicates using the standard I²C protocol. For information on the I²C protocol other than the following sections, please refer to the following website : www.aosong.com provides a sample program reference.

5.1 Start sensor

The first step is to power up the sensor and the voltage is selected. VDD supply voltage (range between 1.8V and 3.6V). After power-on, the sensor needs at most 20 milliseconds time SCL is high) to reach the idle state, ready to receive commands sent by the host (MCU).

5.2 Start / stop timing

Each transmission sequence is The Start state is started and ends with the Stop state, as shown in the figure. 13 and map 14

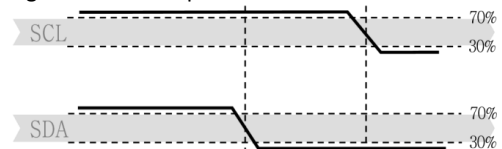


Figure 13 Start Transfer Status (S) - When SCL is high, SDA transitions from high to low. The start state is a special bus state controlled by the host, indicating the start of the slave transfer (after the Start , the BUS bus is generally considered to be busy)

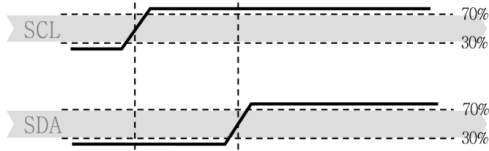


Figure 14 Stop Transfer Status (P) - When SCL is high, the SDA line transitions from low to high. The stop state is a special bus state controlled by the master that indicates the end of the slave transfer (after the Stop , the BUS bus is generally considered to be idle).

5.3 send command

After the transmission is initiated, the subsequently transmitted I²C first byte includes the 7-bit I²C device address 0x38 and one SDA direction (read R : '1', write W : '0'). After the first falling edge of the SCL clock 8, by pulling the SDA pin (ACK bit), indicating proper reception of the sensor. After issuing the initialization command ('1110'0001' represents initialization, '1010'1100' stands for temperature and humidity measurement), the MCU must wait for the measurement to be completed. The basic commands are summarized in Table 9 . Table 10 shows the status bit descriptions returned by the slave.

command	Interpretation	Code
Initialization command	Keep the host	1110'0001
Trigger measurement	Keep the host	1010'1100
Soft reset		1011'1010

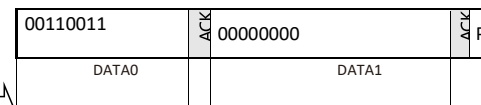
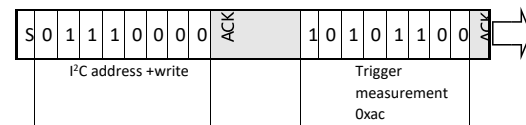
Table 9 basic command set

Bit	significance	description
Bit[7]	Busy indication	1 -- The device is busy, 0 -- the device is in the measurement state, and the device is idle.
Bit[6:5]	Current working mode (Mode Status)	00 Currently NOR mode 01 Currently CYC mode 1x Currently CMD mode
Bit[4]	Reserved	Reserved
Bit[3]	Calibration enable bit CAL Enable	1-calibrated 0-- not calibrated

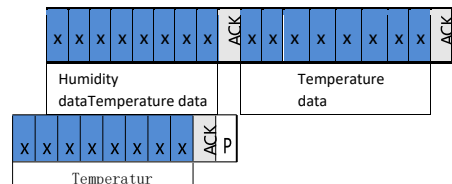
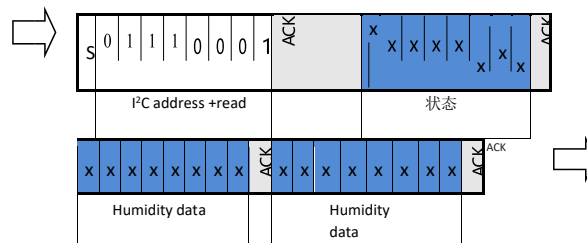
Bit[2: 0]	Reserved	Reserved
-----------	----------	----------

Table 10 status bit description

Trigger measurement data



读取温湿度数据



主机到从机 从机到主机 S Start P Stop

Note: The sensor takes time to collect . After the host sends a measurement command (0xac) , the delay is over 75ms and the converted data is read and the returned status bit is normal. If the status bit [Bit7 of] is 0 for the data can be read normally, the sensor 1 is the busy state, the host needs to wait for the data processing is completed.

5.4 Soft reset

This command (see Table 9) is used to restart the sensor system without turning the power off and on again. After receiving this command, the sensor system begins to reinitialize and restores the default settings. The soft reset takes less than 20ms

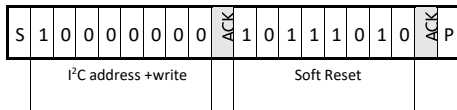


Figure 17 soft reset – the gray part consists of AHT10 control.

6 Signal conversion

6.1 Relative humidity conversion

Relative humidity RH Can be based on SDA
Output relative humidity signal S_{RH} Calculated
by the following formula (the result is %RH
indicates).

$$RH[\%] = \left(\frac{S_{RH}}{2^{20}} \right) * 100\%$$

6.2 Temperature conversion

Temperature T Can output the signal by the
temperature S_T Substitute into the formula below
to calculate (Results are expressed in temperature
°C):

$$T(^{\circ}C) = \left(\frac{S_T}{2^{20}} \right) * 200 - 50$$

7 Environmental stability

If the sensor is used in equipment or machinery, make sure that the sensor used for the measurement and the sensor used for the reference perceive the same conditions of temperature and humidity. If the sensor is placed in the equipment, the reaction time will be extended, so it is necessary to ensure sufficient measurement time in the programming. The AHT10 sensor is tested according to the AOSONG temperature sensor. The performance of the sensor under other test conditions is not guaranteed and cannot be used as part of the sensor's performance. In particular, no commitment is made to the specific circumstances requested by the user.

8 package

The AHT10 is available in an SMD package (similar to QFN), and the SMD is a two-sided, leadless flat package. The sensor chip is made of Ni/

Made of Au 's copper lead frame. The sensor weighs about 63mg.

8.1 tracking information

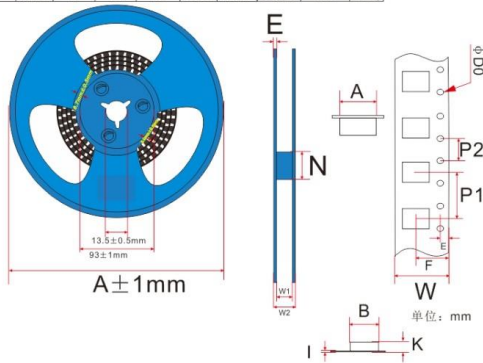
All AHT10 sensor surfaces are laser marked. See picture 18



Figure 18 sensor laser identification.

Labels are also placed on the reels, as shown in Figure 19, and other tracking information is provided. .

Model	A	E	W1	W2	N	Unit	Tolerance	Quantity	Weight
AHT10	330	2	12	16	100	mm	±0.5	4000	500/g



Model	A	B	φD0	K	W	P1	P2	I	F	E
AHT10	4.5±0.1	5.8±0.1	1.5±0.1	2.2±0.1	12.0±0.1	8.0±0.1	4.0±0.1	0.25±0.05	7.0±0.1	1.75±0.1

Figure 20 packaging tape and sensor positioning map



图19：卷轴上的标签。

8.2 Transport Packaging

The AHT10 is packaged in tape and reel and sealed in an antistatic ESD bag. The standard package size is 4000 pieces per roll. For the AHT10 package, the 440mm (55 sensor capacity) and front 200mm (25 sensor capacity) portions of each reel are empty.

The package diagram with sensor positioning is shown in Figure 20. The reel is placed in an anti-static pocket.

Version Information

date	version	page number	change
2018/11	V1.0	1-10	Initial version
2018/12	V1.1	1-10	Added protocol description, command parameters.
2019/05	V1.1	1-13	Draft UNOFFICIAL English translation

This manual is subject to change without notice.

Precautions

Warning, personal injury

Do not apply this product to safety devices or emergency stop devices, and to any other application that may result in personal injury due to malfunction of the product. This product may not be used unless it has a special purpose or authorization to use it. Refer to the product data sheet and application guide before installing, handling, using, or maintaining this product. Failure to follow this advice can result in death and serious personal injury.

If the buyer is to purchase or use AOSONG products without any application license and authorization, the buyer will bear all compensation for personal injury and death, and will be exempt from the AOSONG company managers and employees and affiliated subsidiaries. Any claims that may arise from agents, distributors, etc., including: various costs, compensation, legal fees, etc.

ESD protection

Due to the inherent design of the component, it is sensitive to static electricity. To prevent damage from static electricity or to reduce product performance, take the necessary antistatic measures when applying this product.

Quality Assurance

The company provides a 12 -month (one- year) warranty on the direct purchaser of its products (calculated from the date of shipment), based on the technical specifications in the data sheet of the product published by AOSONG. If the product proves to be defective during the warranty period, the company will provide free repair or replacement. The user must meet the following conditions:

The product is found defective within 14 days written notice to the Company; ● This product defect helps to discover the company's design and materials ● Insufficient materials and processes;

This product should be returned to the company by the purchaser; ● This product should be in the warranty period. ●

The company is only responsible for products that are defective in applications that meet the technical requirements of the product. The company does not make any guarantees, guarantees or written statements about the application of its products in those special applications.

At the same time, the company does not make any commitment to the reliability of its products applied to products or circuits.

Copyright © 2018, ASAIR ®

ANEXO B

CONECTOR DE EXPANSÃO MULTIPLEXADOR DA AERONAVE

