

COMUNICAÇÕES PROJETO DE MINI DRONE DE BAIXO CUSTO PARA APLICAÇÕES MILITARES

1º TEN JEFFERSON ADINIZ BORGES FERREIRA

2º SGT MICAEL REBOUÇAS PEREIRA

3º SGT MARCUS AUGUSTO DE BRITO SOUSA

RESUMO

O projeto RONDRONE: DESENVOLVIMENTO DE UM DRONE DE BAIXO CUSTO PARA APLICAÇÕES MILITARES NO ÂMBITO DO EXÉRCITO BRASILEIRO, desenvolvido no Curso Avançado de Eletrônica 2025, visou criar um protótipo funcional de drone de baixo custo para observação tática, sendo controlado via controle remoto e com transmissão de vídeo em tempo real. A iniciativa é válida pela necessidade de oferecer uma alternativa viável, acessível e eficiente, apoiando, de maneira positiva, atividades de ISR (Inteligência, Vigilância e Reconhecimento), em face do elevado custo dos Veículos Aéreos Não Tripulados (VANTs) militares. O sistema, baseado na plataforma Arduino, demonstrou, através de testes realizados durante o seu desenvolvimento, funcionamento satisfatório, com resposta eficiente aos comandos e transmissão de vídeo com baixa latência. Embora os testes de voo ainda não tenham sido executados, o protótipo demonstra ser uma solução promissora para missões que exigem autonomia, mobilidade e baixo custo operacional.

Palavras-chave: drones; Arduino; defesa; baixo custo; vigilância.

ABSTRACT

The project RONDRONE: DESENVOLVIMENTO DE UM DRONE DE BAIXO CUSTO PARA APLICAÇÕES MILITARES NO ÂMBITO DO EXÉRCITO BRASILEIRO, developed during the Curso Avançado de Eletrônica 2025, aimed to create a functional prototype of a low-cost drone for tactical observation, remotely controlled and capable of real-time video transmission. The initiative is justified by the need to provide a viable, accessible, and efficient alternative to positively

support ISR (intelligence, surveillance, and reconnaissance) activities, given the high cost of military Unmanned Aerial Vehicles (UAVs). The system, based on the Arduino platform, demonstrated, through tests carried out during its development, satisfactory operation, with efficient response to commands and low-latency video transmission. Although flight tests have not yet been performed, the prototype proves to be a promising solution for missions requiring autonomy, mobility, and low operational cost.

Keywords: drones; Arduino; defense; low cost; surveillance.

1. INTRODUÇÃO

O avanço na tecnologia dos drones (Veículo Aéreo Não Tripulado – VANT) representa uma das inovações mais relevantes das últimas décadas, mudando a maneira como são conduzidas as operações militares e de vigilância. Operações recentes, como a “Operação Teia de Aranha”, ataque secreto ao território russo com 117 drones realizado pelo Serviço de Segurança da Ucrânia (O GLOBO, 2025), demonstraram o potencial dos VANTs em redefinir estratégias de guerra e inteligência.

No contexto brasileiro, observa-se um crescente interesse das Forças Armadas na incorporação dessas tecnologias. Desde 2010, a Força Aérea Brasileira passou a operar drones israelenses dos modelos RQ-450 e RQ-900, empregados principalmente em atividades de sensoriamento remoto, com foco em missões ISR. Em 2022, a Marinha do Brasil incorporou o modelo Scan Eagle, também destinado a operações de vigilância (Silva et al., 2025). Contudo, o elevado



custo dos equipamentos militares limita sua adoção em larga escala. Exemplo disso é a aquisição, pelo Exército Brasileiro, de três unidades do modelo NAURU 1000C, em 2022, pelo valor de R\$ 15,7 milhões (Silva et al., 2025).

Diante desse cenário, o projeto RONDRONE surge como uma proposta inovadora e de baixo custo, voltada para o desenvolvimento nacional de drones de observação tática. A utilização de plataformas acessíveis, como o Arduino, e de sensores de fácil obtenção, viabiliza a criação de soluções eficientes, autônomas e financeiramente viáveis.

2. DESENVOLVIMENTO

O Projeto RONDRONE é uma iniciativa de pesquisa aplicada que busca o desenvolvimento de um quadricóptero funcional, priorizando a simplicidade construtiva e a viabilidade técnica e econômica. O protótipo se enquadra na Classe 3 (até 25 kg), com peso total inferior a 250 gramas, garantindo maior liberdade operacional e simplicidade regulatória, uma vez que já são considerados como devidamente licenciados (ANAC, 2023).

2.1. Aspectos técnicos do projeto

O protótipo foi concebido utilizando componentes de prateleira e open-source (código aberto), e em sua arquitetura se destacam quatro subsistemas principais: controle de voo, propulsão, comunicação e energia. O projeto adota a arquitetura de um quadricóptero, por apresentar simplicidade mecânica, estabilidade de voo e facilidade de replicação.

Controle de Voo

O controle de voo utiliza a plataforma Arduino como unidade central, sendo o "cérebro" da aeronave.

- **Microcontrolador:** O drone emprega o Arduino Pro Mini (3,3 V), escolhido por ser compacto e leve. O controle remoto utiliza o Arduino Nano.

- **Sensores:** Para estabilidade e manobrabilidade, é utilizado o sensor MPU-6050 (IMU), que integra acelerômetro e giroscópio, comunicando-se via protocolo I²C. Sua leitura é indispensável para o sistema de estabilização.

- **Firmware:** O controle dinâmico da aeronave utiliza o firmware MultiWii, um software open-source em linguagem C/C++, amplamente empregado em projetos de drones amadores e acadêmicos. Ele utiliza os sinais do sensor MPU6050 para controlar dinamicamente o giro dos motores. Foi implementada blindagem de cobre sob o microcontrolador para reduzir interferências eletromagnéticas.

- **Limitação:** A principal limitação técnica foi a finalização parcial do código de controle de voo, o que comprometeu o acionamento efetivo dos motores e a realização de testes práticos capazes de avaliar a estabilidade do equipamento.

Propulsão

O subsistema de propulsão fornece a tração necessária para o voo.

- **Motores:** Foram selecionados quatro motores coreless de 7 mm, ideais para micro quadricópteros devido à leveza, resposta rápida e baixo custo.

- **Controle dos Motores (ESCs):** Foi desenvolvido um circuito específico de controle de micromotores DC com MOSFETs AO3402 (transistores de canal N), que atuam como chaves eletrônicas para modular a velocidade (PWM) dos motores. Diodos Schottky SS14 foram adicionados para proteção contra picos de tensão (flyback).

- **Hélices:** As hélices 30 x 4 mm foram escolhidas por oferecer equilíbrio entre leveza e tração.

- **Estrutura:** O frame (estrutura) do drone (58,79 x 58,79 x 10,80 mm) foi fabricado por meio de impressão 3D em PLA, garantindo leveza (aproximadamente 4 gramas) e resistência mecânica.



Figura 1 - Projeção 3D da estrutura do drone - Elaborado pelos alunos do curso Avançado de Eletrônica

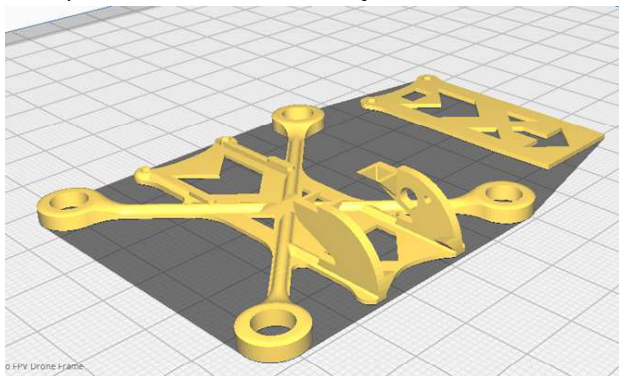


Figura 2 - Protótipo da estrutura física do drone - Elaborado pelos alunos do curso Avançado de Eletrônica



Figura 3 - Projeção 3D do controle - Elaborado pelos alunos do curso Avançado de Eletrônica

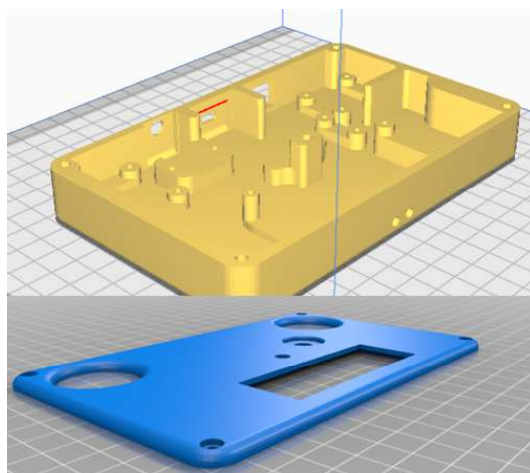


Figura 4 - Protótipo da estrutura física do controle - Elaborado pelos alunos do curso Avançado de Eletrônica



- **FPV (Visão em Primeira Pessoa):** Para observação tática, o projeto integrou a câmera WT05 Micro AIO (*All-in-One*) 600TVL, que transmite vídeo em 5,8 GHz, evitando interferência com o controle de voo (2,4 GHz), a 200 mW de potência. As imagens são recebidas por um receptor OTG (*On-The-Go*) conectado a um smartphone, proporcionando transmissão com baixa latência.

Energia

O subsistema de energia determina o tempo de voo e sustenta os demais sistemas.

- **Bateria:** O drone é alimentado por uma bateria Li-Po de 3,7 V e 220 mAh, escolhida por sua alta densidade energética, baixo peso e tamanho.

- **Câmera FPV:** A câmera foi conectada diretamente aos terminais da bateria Li-Po, evitando sobrecarga no regulador interno do Arduino Pro Mini.

- **Controle Remoto:** O controle remoto utiliza uma bateria Power Unit de 9 V.

- **Monitoramento:** Um divisor de tensão permite o monitoramento do nível de carga da bateria principal.

2.2. Aplicações e vantagens operacionais

O RONDRONE se destaca pela viabilidade econômica e eficiência operacional.

- **Custo-Benefício:** O custo total do protótipo funcional é inferior a R\$1.000,00 (drone: R\$779,94; controle remoto: R\$193,92), valor significativamente menor do que soluções comerciais e militares existentes. Essa acessibilidade torna o projeto escalável e replicável.

- **Vantagens Táticas:** O mini drone é ideal para missões de vigilância, reconhecimento tático, patrulhamento de fronteiras e apoio logístico.

- **Performance Tática:** O modelo quadricóptero (asa rotativa) é mais adequado ao ambiente urbano, pois possibilita o voo estacionário (*hovering*), manobras precisas e operação em baixa velocidade. O que possibilita substituir o uso de helicópteros em determinadas situações (SILVA, 2008).

- **Segurança:** A ausência de tripulação elimina o risco humano, e o pequeno porte confere baixa assinatura radar, dificultando a detecção (SILVA, 2008).

2.3. Impactos éticos e industriais

O desenvolvimento do RONDRONE possui implicações estratégicas claras para a soberania nacional, a autonomia tecnológica e a modernização das Forças Armadas.

- **Soberania Tecnológica:** O projeto, baseado em sistemas abertos e de baixo custo, contribui

para a redução da dependência de importações estratégicas.

Figura 5 - Esquemático do drone desenhado no programa Fritzing - Elaborado pelos alunos do curso Avançado de Eletrônica

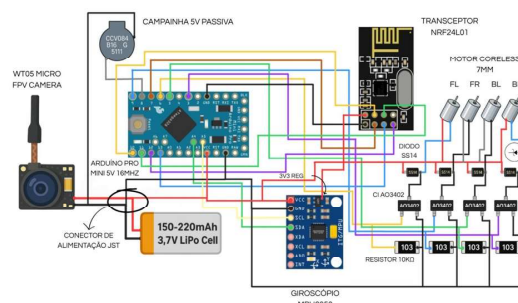
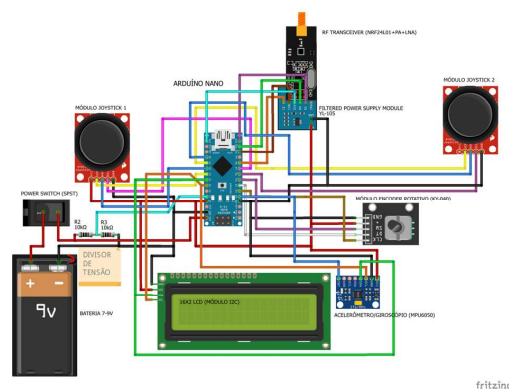


Figura 6 - Esquemático do controle desenhado no programa Fritzing - Elaborado pelos alunos do curso Avançado de Eletrônica



- **Inovação Industrial:** Fortalece a Base Industrial de Defesa (BID), estimulando a pesquisa aplicada, a inovação tecnológica e a formação de profissionais qualificados.

- **Aspectos Regulatórios:** O peso inferior a 250 gramas isenta o drone de registro junto à ANAC, facilitando seu uso experimental. Contudo, sua aplicação militar deve obedecer à Portaria COTER/C Ex nº 333/2023, que regulamenta o uso de VANTs pelo Exército Brasileiro.

- **Aprimoramentos:** Embora o baixo custo seja a prioridade, o projeto estabelece uma base sólida para futuras melhorias, como a integração de sensores adicionais — incluindo GPS, barômetro,

magnetômetro, câmeras infravermelhas ou de visão noturna, LiDAR e sensores ultrassônicos — que viabilizariam funcionalidades de voo autônomo, maior precisão na navegação e detecção de obstáculos. Melhorias no software, como algoritmos avançados de controle de motores, fail-safe, retorno automático (return-to-home) e geofencing, bem como aprimoramentos na transmissão de vídeo digital e redundância na comunicação, poderiam aumentar a confiabilidade e a versatilidade operacional do sistema. Além disso, ajustes na estrutura, modularidade, eficiência energética e capacidade de carga dariam suporte a operações mais prolongadas e seguras, mantendo a filosofia de baixo custo e acessibilidade do protótipo. (Silva et al., 2025)

Figura 7 - Protótipo final do Drone - Elaborado pelos alunos do curso Avançado de Eletrônica



Figura 8 - Protótipo final do controle - Elaborado pelos alunos do curso Avançado de Eletrônica



3. CONCLUSÃO

O desenvolvimento do Projeto RONDRONE demonstrou a viabilidade técnico-econômica de criar um drone voltado para ISR e de baixo custo, baseado em uma tecnologia acessível e componentes amplamente disponíveis. A integração dos sistemas de controle, propulsão, comunicação e vídeo demonstrou coerência e eficiência nos testes de bancada, validando o conceito proposto.

O custo total inferior a R\$1.000,00 valida a proposta de um Veículo Aéreo Não Tripulado (VANT) economicamente sustentável, que representa uma alternativa estratégica às soluções comerciais e militares de alto custo. O protótipo desenvolvido, pelo Curso Avançado de Eletrônica 2025, apresenta potencial significativo para ampliar as capacidades de inteligência, reconhecimento e vigilância do Exército Brasileiro, especialmente em operações de Garantia da Lei e da Ordem (GLO) (SILVA, 2008), nas quais a versatilidade e o baixo risco operacional são essenciais.

Embora a limitação de software tenha impedido a realização de testes de voo, a base técnica estabelecida é consistente e oferece um ponto de partida sólido para futuras melhorias, como o aprimoramento do controle dinâmico, a integração de sensores adicionais e o avanço rumo ao voo autônomo.

A continuidade da pesquisa poderá consolidar o RONDRONE como uma ferramenta estratégica indispensável à modernização tecnológica e à capacidade operacional das Forças Armadas, reforçando o compromisso com a autonomia nacional e a inovação, bem como o fortalecimento da Base Industrial de Defesa (BID).

4. REFERÊNCIAS

SILVA, W. F. C. D.; et al. **Rondrone: desenvolvimento de um drone de baixo custo para aplicações militares no âmbito do Exército Brasileiro**. Curso Avançado de Eletrônica. Brasília, 2025.

SILVA, R. M. B. da. **O emprego do veículo aéreo não-tripulado (VANT) em operações de garantia da lei e da ordem.** Escola de Artilharia de Costa e Antiaérea. Rio de Janeiro, 2008.


O Globo. **Operação ‘Teia de Aranha’: o que se sabe sobre o mega-ataque a drone da Ucrânia na Rússia e seu impacto para a guerra.** Disponível em: <https://oglobo.globo.com/mundo/noticia/2025/06/02/operacao-teia-de-aranha-o-que-se-sabe-sobre-o-mega-ataque-a-drone-da-ucrania-na-russia-e-seu-o-impacto-para-a-guerra.ghhtml>. Acesso em 02 nov. 2025.

Agência Nacional De Aviação Civil (ANAC). **Regulamento Brasileiro da Aviação Civil Especial – RBAC-E nº 94: Requisitos gerais para aeronaves não tripuladas de uso civil. Emenda nº 03. Brasília: ANAC, 03 abr. 2023.** Disponível em: <https://www.anac.gov.br/assuntos/legislacao/legislacao-1/rbha-e-rbac/rbac/rbac-e-94>. Acesso em: 02 nov. 2025.



ANEXO A - CUSTO DO PROTÓTIPO DO DRONE

Tabela 1 - Custo do protótipo do drone.

COMPONENTE ELETRÔNICO	QTD	FIGURA	VALOR
ARDUINO PRO MINI (3,3V)	01		R\$ 59,31
ACELERÔMETRO E GIROSCÓPIO (MPU6050)	01		R\$ 16,11
TRANSCEPTOR NRF24L01	01		R\$ 8,01
CÂMERA FPV 5.8g 200mw + Receptor Otg	01		R\$ 559,00
MOTOR CORELESS (7 mm)	04		R\$ 64,90
MOSFET SMD (TRANSISTOR CIAO3402)	04		R\$ 5,60
RESISTOR SMD (10K)	04		R\$ 0,44
DIODO SCHOTTKY SMD (SS14)	04		R\$ 8,00
CONECTOR JST 1 macho + 1 fêmea	01		R\$ 24,60
BATERIA LIPO (3,7V 220 mAh)	01		R\$ 4,45
HÉLICES PARA DRONE CHEERSON CX10 (30 x 4 mm)	04		R\$ 15,00
PROTOBOARD (PLACA PERFURADA)	01		R\$ 3,51

COMPONENTE ELETRÔNICO	QTD	FIGURA	VALOR
FOLHA DE COBRE (30X18mm)	5cm		R\$ 1,00
FIO DE SOLDA 0,5MM ESTANHO	20g		R\$ 5,76
FIO DE SILICONE (30AWG)	200cm		R\$ 4,00
CAPACITOR ELETROLÍTICO 10uF x 25V	01		R\$ 0,25
VALOR TOTAL			R\$ 779,94

ANEXO B - CUSTO DO PROTÓTIPO DO CONTROLE

Tabela 2 - Custo do protótipo do controle.

COMPONENTE ELETRÔNICO	QTD	FIGURA	VALOR
ARDUINO NANO	01		R\$ 44,91
MÓDULO WIRELESS NRF24L01 LNA	01		R\$ 29,61
POWER UNIT BATERIA 9V PILHA	01		R\$ 22,40
ACELERÔMETRO E GIROSCÓPIO (MPU6050)	01		R\$ 16,11
ALAVANCAS ANALÓGICAS (JOYSTICKS)	02		R\$ 13,50
MÓDULO ENCODER ROTACIONAL	01		R\$ 11,12
CONECTOR 9V PARA BATERIA OU PILHA	01		R\$ 1,03
SWITCH (CHAVE GANGORRA)	01		R\$ 0,71
LCD (16X2)	01		R\$ 34,90
MÓDULO ADAPTADOR PARA NRF24L01	01		R\$ 12,00
PORCA DE INSERÇÃO M3	04		R\$ 3,20

COMPONENTE ELETRÔNICO	QTD	FIGURA	VALOR
FIO DE SILICONE (30AWG)	200cm		R\$ 4,00
RESISTOR 10K Ohm	02		R\$ 0,18
CAPACITOR ELETROLÍTICO 10uF x 25V	01		R\$ 0,25
VALOR TOTAL			R\$ 193,92

ANEXO C – CÓDIGO DE PROGRAMAÇÃO UTILIZADO NO MICROCONTROLADOR

Este anexo apresenta o código-fonte do sistema embarcado desenvolvido para a operação do drone, com base na plataforma MultiWii.

Considerando que grande parte da base de código da plataforma MultiWii é padronizada e amplamente documentada na literatura e nos repositórios oficiais, optou-se por incluir neste anexo apenas os trechos de código que foram modificados ou desenvolvidos especificamente para o projeto RONDODRONE, tanto na unidade embarcada (drone) quanto na unidade transmissora (controle remoto).

Os arquivos originais, que permaneceram inalterados, podem ser consultados diretamente no repositório oficial do firmware MultiWii.

C.1 Código de programação utilizado no Drone

C.1.1 NRF24_RX.cpp

```
#include "Arduino.h" // BIBLIOTECA PADRÃO DO ARDUINO
#include "config.h" // ARQUIVO DE CONFIGURAÇÃO DO PROJETO
#include "def.h" // BIBLIOTECA COM DEFINIÇÕES GERAIS
#include "types.h" // BIBLIOTECA COM TIPOS
#include "MultiWii.h" // BIBLIOTECA PRINCIPAL DO MULTIWII (CONTROLE DE VOO)
#include <RF24.h> // BIBLIOTECA OFICIAL DO NRF24L01 PARA COMUNICAÇÃO SEM FIO
#include "NRF24_RX.h" // ARQUIVO DE CABEÇALHO PARA A IMPLEMENTAÇÃO DO RECEPTOR NRF24.

#if defined(NRF24_RX) // INCLUI O BLOCO SE E SOMENTE SE A FLAG NRF24_RX ESTIVER DEFINIDA

int16_t nrf24_rcData[RC_CHANS]; // ARRAY PARA ARMAZENAR OS VALORES RECEBIDOS DOS CANAIS DE CONTROLE REMOTO
static const uint64_t pipe = 0xE8E8F0F0E1LL; // ENDEREÇO DE COMUNICAÇÃO ENTRE O CONTROLE E O RECEPTOR NRF24
RF24 radio(7, 10); // INSTANCIA UM OBJETO PARA O MÓDULO NRF24 COM PINOS CE = D7 E CSN = D10
RF24Data MyData; // ESTRUTURA USADA PARA ARMAZENAR OS DADOS RECEBIDOS DO TRANSMISSOR
```



```

void resetRF24Data() // FUNÇÃO PARA ZERAR OS VALORES DE ENTRADA EM CENTRO OU DESLIGADO
{
MyData.throttle = 0;
MyData.yaw = 128;
MyData.pitch = 128;
MyData.roll = 128;
MyData.AUX1 = 0;
MyData.AUX2 = 0;
MyData.AUX3 = 0;
}

void NRF24_Init() // FUNÇÃO PARA INICIALIZAR A COMUNICAÇÃO COM O MÓDULO NRF24
{
radio.begin(); // INICIALIZA O MÓDULO NRF
radio.setChannel(90); // DEFINE O CANAL DE COMUNICAÇÃO
radio.setDataRate(RF24_250KBPS); // DEFINE A TAXA DE DADOS MAIS LENTA PARA MAIOR ALCANCE E ESTABILIDADE
radio.setAutoAck(true); // ATIVA RECONHECIMENTO AUTOMÁTICO DE PACOTES RECEBIDOS
radio.openReadingPipe(1,pipe); // ABRE O CANAL DE RECEPÇÃO USANDO O PIPE DEFINIDO
radio.setPALevel(RF24_PA_HIGH); // DEFINE O NÍVEL DE POTÊNCIA DE TRANSMISSÃO
radio.startListening(); // COLOCA O MÓDULO EM MODO RECEPÇÃO
}

void NRF24_Read_RC() // FUNÇÃO PARA LER OS DADOS DO CONTROLE REMOTO A CADA LOOP
{
static unsigned long lastRecvTime = 0; // VARIÁVEL PARA RASTREAR O ÚLTIMO PACOTE RECEBIDO
unsigned long now = millis(); // OBTÉM O TEMPO ATUAL
while (radio.available()) // VERIFICA SE HÁ DADOS DISPONÍVEIS
{
radio.read(&MyData, sizeof(RF24Data)); // LÊ OS DADOS RECEBIDOS E ARMAZENA
lastRecvTime = now; // ATUALIZA O TEMPO DO ÚLTIMO PACOTE
}
if ( now - lastRecvTime > 50000 ) // SE PASSAR ESSE TEMPO SEM RECEBER DADOS
resetRF24Data(); // CONSIDERA SINAL PERDIDO E ZERA OS COMANDOS
// CONVERTE OS VALORES RECEBIDOS (0–255) PARA O INTERVALO USADO PELO MULTIWII (1000–2000)
nrf24_rcData[THROTTLE] = map(MyData.throttle, 0, 255, 2000, 1000);
nrf24_rcData[PITCH] = map(MyData.pitch, 0, 255, 2000, 1000);
nrf24_rcData[YAW] = map(MyData.yaw, 0, 255, 2000, 1000);
nrf24_rcData[AUX1] = map(MyData.AUX1, 0, 1, 2000, 1000);
nrf24_rcData[AUX2] = map(MyData.AUX2, 0, 1, 2000, 1000);
nrf24_rcData[AUX3] = map(MyData.AUX3, 0, 1, 2000, 1000);
}
#endif

```



C.1.2 NRF24_RX.h

```
// INÍCIO DO SISTEMA DE INCLUSÃO CONDICIONAL (INCLUDE GUARD), PARA EVITAR MÚLTIPLA INCLUSÃO DO HEADER
#ifndef NRF24_RX_H_
#define NRF24_RX_H_

#include "config.h" // INCLUI O ARQUIVO DE CONFIGURAÇÕES GERAIS

#if defined(NRF24_RX) // VERIFICA SE A FLAG NRF24_RX ESTÁ DEFINIDA
struct RF24Data // DEFINE UMA STRUCT QUE REPRESENTA OS DADOS ENVIADOS PELO TRANSMISSOR
{
byte throttle;
byte yaw;
byte pitch;
byte roll;
byte AUX1;
byte AUX2;
byte AUX3;
};

struct RF24AckPayload // DEFINE UMA STRUCT PARA DADOS DE RESPOSTA QUE O DRONE ENVIA PARA O TRANSMISSOR
{
float lat;
float lon;
int16_t heading;
int16_t pitch;
int16_t roll;
int32_t alt;
byte flags;
};

extern RF24Data nrf24Data; // DECLARAÇÃO DE VARIÁVEL EXTERNA DEFINIDA EM OUTRO ARQUIVO
extern RF24AckPayload nrf24AckPayload; // DECLARAÇÃO DE VARIÁVEL EXTERNA DEFINIDA EM OUTRO ARQUIVO
extern int16_t nrf24_rcData[RC_CHANS]; // DECLARAÇÃO DE ARRAY EXTERNO DEFINIDO EM OUTRO ARQUIVO

void NRF24_Init(); // FUNÇÃO QUE INICIALIZA O MÓDULO NRF
void NRF24_Read_RC(); // FUNÇÃO QUE LÊ DADOS E ENVIA AO MÓDULO NRF

#endif
#endif
```



C.2 Código de programação utilizado no Controle

C.2.1 NRF24_6CH_Radio_Controller_Code.h

```
#include <SPI.h> // BIBLIOTECA PARA COMUNICAÇÃO SPI PADRÃO NRF24L01
#include <nRF24L01.h> // BIBLIOTECA ESPECÍFICA PARA CONTROLE DO NRF
#include <RF24.h> // BIBLIOTECA ESPECÍFICA PARA CONTROLE DO NRF
#include <LiquidCrystal_I2C.h> // BIBLIOTECA PARA CONTROLE DO DISPLAY LCD COM I2C

float vol = 0; // VARIÁVEL PARA ARMAZENAR LEITURA CONVERTIDA DA TENSÃO
int input = 0; // VARIÁVEL PARA ARMAZENAR LEITURA BRUTA DA TENSÃO
int vdividerPin = A7; // VARIÁVEL DO DIVISOR DE TENSÃO
const uint64_t pipeOut = 0xE8E8F0F0E1LL; // CÓDIGO DE CONEXÃO, O MESMO DO DRONE
LiquidCrystal_I2C lcd(0x3F, 16, 2); // DEFINIÇÃO DO TAMANHO DO LCD 16X2
RF24 radio(7, 10); // SELEÇÃO DOS PINOS CE E CSN
struct MyData // CRIAÇÃO DE "BIBLIOTECA DE VARIÁVEIS PARA ENVIO AO NRF
{
byte throttle; // CRIAÇÃO DE VARIÁVEL PARA COMUNICAÇÃO RF
byte yaw; // CRIAÇÃO DE VARIÁVEL PARA COMUNICAÇÃO RF
byte pitch; // CRIAÇÃO DE VARIÁVEL PARA COMUNICAÇÃO RF
byte roll; // CRIAÇÃO DE VARIÁVEL PARA COMUNICAÇÃO RF
byte AUX1; // CRIAÇÃO DE VARIÁVEL PARA COMUNICAÇÃO RF
byte AUX2; // CRIAÇÃO DE VARIÁVEL PARA COMUNICAÇÃO RF
byte AUX3; // CRIAÇÃO DE VARIÁVEL PARA COMUNICAÇÃO RF
};

MyData data; // BIBLIOTECA DATA CRIADA

unsigned long lcdUpdateTime = 0; // VARIÁVEL PARA ATUALIZAÇÃO DO LCD
const unsigned long lcdUpdateInterval = 100; // ATUALIZAÇÃO DE DADOS EM 100 MILISEGUNDOS

// DEFINIÇÃO DA FUNÇÃO RESERTDATA - INICIALIZAÇÃO DOS VALORES NRF ARMAZENADOS
void resetData()
{
data.throttle = 0; // ZERAR VARIÁVEL DA VELOCIDADE DO MOTOR PARA PARAR
data.yaw = 127; // ZERAR VARIÁVEIS CENTRO (0-255)
data.pitch = 127; // ZERAR VARIÁVEIS CENTRO (0-255)
data.roll = 127; // ZERAR VARIÁVEIS CENTRO (0-255)
data.AUX1 = 0; // HABILITAR CONTROLE PARA ATIVAR MOTORES COM SEGURANÇA QUANDO HIGH
data.AUX2 = 0; // TELIMETRIA DO LCD VIA CONTROLE
data.AUX3 = 0; // VARIÁVEL VAZIA
}
```



```

void setup() // PARTE DE INICIALIZAÇÃO DO CÓDIGO
{
radio.begin(); // INICIALIZA O MÓDULO DO RÁDIO NRF
radio.setChannel(90); // DEFINE O CANAL DE COMUNICAÇÃO
radio.setAutoAck(true); // HABILITA O RECONHECIMENTO AUTOMÁTICO DE PACOTES RECEBIDOS
radio.setDataRate(RF24_250KBPS); // DEFINE A TAXA DE DADOS
radio.setPALevel(RF24_PA_HIGH); // DEFINE O STATUS DO NRF NO MÁXIMO DE POTENCIA
radio.openWritingPipe(pipeOut); // ABRE UM CANAL PARA ENVIAR DADOS AO ENDEREÇO REFERENCIADO
lcd.init(); // INICIALIZA O LCD
lcd.backlight(); // LIGA A LUZ DE FUNDO DO LCD

// EXIBE NA TELA AS INFORMAÇÕES DE INICIALIZAÇÃO E AGUARDA 3 SEGUNDOS
lcd.setCursor(1, 0); // SETAR POSIÇÃO DO CURSOR LCD
lcd.clear(); // LIMPAR O LCD
lcd.print("Iniciando..."); // ESCREVER NO LCD
lcd.setCursor(0, 1); // SETAR POSIÇÃO DO CURSOR LCD
lcd.print("AGUARDE"); // ESCREVER NO LCD
delay(3000); // ESPERA EM MILISSEGUNDOS

// EXIBE NA TELA AS INFORMAÇÕES DE CONFECÇÃO DO DRONE
lcd.setCursor(1, 0); // SETAR POSIÇÃO DO CURSOR LCD
lcd.clear(); // LIMPAR O LCD
lcd.print("MINI FPV DRONE"); // ESCREVER NO LCD
lcd.setCursor(0, 1); // SETAR POSIÇÃO DO CURSOR LCD
lcd.print("Feito por:"); // ESCREVER NO LCD
delay(1600); // ESPERA EM MILISSEGUNDOS
lcd.setCursor(1, 0); // SETAR POSIÇÃO DO CURSOR LCD
lcd.clear(); // LIMPAR O LCD
lcd.print("SGT CABRAL SILVA"); // ESCREVER NO LCD
lcd.setCursor(0, 1); // SETAR POSIÇÃO DO CURSOR LCD
lcd.print("SGT ADALBERTO"); // ESCREVER NO LCD
delay(1600); // ESPERA EM MILISSEGUNDOS
lcd.setCursor(1, 0); // SETAR POSIÇÃO DO CURSOR LCD
lcd.clear(); // LIMPAR O LCD
lcd.print("SGT GABRIELY"); // ESCREVER NO LCD
lcd.setCursor(0, 1); // SETAR POSIÇÃO DO CURSOR LCD
lcd.print("SGT CERQUEIRA"); // ESCREVER NO LCD
delay(1600); // ESPERA EM MILISSEGUNDOS
lcd.setCursor(1, 0); // SETAR POSIÇÃO DO CURSOR LCD
lcd.clear(); // LIMPAR O LCD
lcd.print("SGT BALIEIRO"); // ESCREVER NO LCD
lcd.setCursor(0, 1); // SETAR POSIÇÃO DO CURSOR LCD
lcd.print("SGT VELLASCO"); // ESCREVER NO LCD
delay(1600); // ESPERA EM MILISSEGUNDOS
lcd.setCursor(1, 0); // SETAR POSIÇÃO DO CURSOR LCD

```



```

lcd.clear(); // LIMPAR O LCD
lcd.print("RONDRONE"); // ESCREVER NO LCD
lcd.setCursor(0, 1); // SETAR POSIÇÃO DO CURSOR LCD
lcd.print("Boas-Vindas"); // ESCREVER NO LCD
delay(2200); // ESPERA EM MILISSEGUNDOS
lcd.clear(); // LIMPAR O LCD

pinMode(vdividerPin, INPUT); // DEFINE O PINO DO DIVISOR DE TENSÃO COMO ENTRADA
Serial.begin(9600); // INICIALIZA A COMUNICAÇÃO COM O MONITOR SERIAL
pinMode(5, INPUT_PULLUP); // DEFINE O PINO 5 COMO ENTRADA E COM RESISTORES DE PULLUP
pinMode(3, INPUT_PULLUP); // DEFINE O PINO 3 COMO ENTRADA E COM RESISTORES DE PULLUP
}

// MAPEAMENTO DOS VALORES ANALÓGICOS (0-255) DO CONTROLE

int mapJoystickValues(int val, int lower, int middle, int upper, bool reverse)
{
val = constrain(val, lower, upper); if ( val < middle )
val = map(val, lower, middle, 0, 128); else
val = map(val, middle, upper, 128, 255); return ( reverse ? 255 - val : val );
}

void loop() // ESTRUTURA DE REPETIÇÃO
{
unsigned long currentMillis = millis(); // VARIÁVEL DE ATUALIZAÇÃO DO LCD

if (currentMillis - lcdUpdateTime >= lcdUpdateInterval) // ESTRUTURA CONDICIONAL PARA ATUALIZAR O LCD EM
INTERVALOS DEFINIDOS
{
updateLCD(); // ATUALIZAR LCD
lcdUpdateTime = currentMillis; //ATUALIZAR A VARIÁVEL DE ATUALIZAÇÃO DO LCD
}

// LEITURA DOS VALORES DOS ANALÓGICOS E BOTÃO ENCODER
data.throttle = mapJoystickValues(analogRead(A2), 0, 528, 1022, true); // LEITURA VRY esquerda ACELERAÇÃO
data.yaw = mapJoystickValues(analogRead(A1), 0, 515, 1023, true); // LEITURA VRY direita DIREITA E ESQUERDA
data.pitch = mapJoystickValues(analogRead(A3), 0, 526, 1023, true); // LEITURA VRX esquerda Mergulhar PITCH
data.roll = mapJoystickValues(analogRead(A0), 0, 497, 1022, true); // LEITURA VRX direita ROLAGEM
if (digitalRead(5) == LOW) // CONDICIONAL PARA FORÇAR VALOR DE AUX1
{
data.AUX1 = 2000;
}

else
{
data.AUX1 = 1000;
}
}

```



```
}  
data.AUX2 = digitalRead(3); // LEITURA ENCODER ROTAÇÃO  
data.AUX3 = 0; // VARIÁVEL VAZIA  
radio.write(&data, sizeof(MyData)); // ENVIO DA BIBLIOTECA PREENCHIDA AO DRONE  
}  
void updateLCD() // FUNÇÃO DE ATUALIZAÇÃO DO TEXTO EXIBIDO NO LCD  
{  
input = analogRead(vdividerPin); // LEITURA DO DIVISOR DE TENSÃO  
vol = (input * 10.0) / 1024.0; // CONVERSÃO DO VALOR LIDO  
  
lcd.clear(); // LIMPAR O LCD  
lcd.setCursor(0, 0); // SETAR POSIÇÃO DO CURSOR LCD  
lcd.print("Tensao:"); // ESCREVER NO LCD  
lcd.setCursor(0, 1); // SETAR POSIÇÃO DO CURSOR LCD  
lcd.print(vol); // ESCREVER A VARIÁVEL NO LCD  
lcd.print("V"); // ESCREVER NO LCD
```

