

ESCOLA ESTADUAL DE EDUCAÇÃO PROFISSIONAL DR. SOLON TAVARES

CURSO TÉCNICO EM ELETROELETRÔNICA

Vinicius Moraes Boeira Pereira

ARDURESCUE

Guaíba

2020

Vinicius Moraes Boeira Pereira

ARDURESCUE

Trabalho de Conclusão de Curso
apresentado à Escola Estadual de
Educação Profissional Dr. Solon Tavares
como requisito parcial para a obtenção do
título de Técnico em Eletroeletrônica.

Orientador: Márcio Anderson de Souza
Mota

Guaíba

2020

Vinicius Moraes Boeira Pereira

ARDURESCUE

Trabalho de Conclusão apresentado à Escola Estadual de Educação Profissional Dr. Solon Tavares como requisito parcial para obtenção do título de Técnico em Eletroeletrônica.

Aprovado em ____/____/____

BANCA EXAMINADORA

Prof^a. Scheila Cristina Silva – Web Design e Programação/Unisul

Prof. Márcio Anderson de Souza Mota - Licenciatura em Física/ULBRA

Prof^a. Carmen Silvia Propp de Jesus Medeiros - Licenciatura em Pedagogia/Ulbra

Dedico este trabalho aos meus colegas de
escola, a professora Scheila, professor
Marcio e ex professor Claiton.

“O Problema que a maioria de nós tem é
que preferimos ser arruinados por elogios
do que salvos por críticas”

RESUMO

Neste trabalho será apresentado todos os passos que segui para construir o projeto ArduRescue, que basicamente é um drone construído a partir de um microcontrolador AVR Atmega328P, nativamente utilizado nas placas de prototipagem Arduino. A programação do projeto foi realizada toda em linguagem C++ (também linguagem nativa da IDE do Arduino). A montagem da placa de controle do projeto é feita totalmente de placas externas a principal do drone, como o módulo nRF24l01 para o envio e recebimento de dados de ambas as partes de comunicação, também o módulo MPU6050, utilizado para captar a movimentação realizada pelo drone e calcular através de software os movimentos que ele faria. O software do drone foi utilizado um conjunto de algoritmos para filtragem de dados como por exemplo o filtro de média móvel, para estabilização do drone, foi utilizado um algoritmo de PID, que detecta um erro de inclinação através do giroscópio do modulo MPU, que é enviado a um algoritmo de distribuição de potencias que então é enviado aos ESC's (Eletronic Speed Controller) que são repassadas as bobinas dos motores brushless responsáveis pelo voo.

Palavras chaves: DRONE, RESGATE, SOFTWARE, QUADCÓPTERO

ABSTRACT

In this work, all the steps I followed to build the ArduRescue project will be presented, which basically is a drone built from an AVR Atmega328P microcontroller, natively used in Arduino prototyping boards. The programming of the project was done entirely in C ++ (also native language of the Arduino IDE). The assembly of the project control board is made entirely of external boards, the main of the drone, such as the module nRF24I01 for sending and receiving data from both communication parts, also the module MPU6050, used to capture the movement carried out by the drone and calculate through software the movements he would make. The drone software used a set of algorithms for data filtering, such as the moving average filter, for stabilizing the drone, a PID algorithm was used, which detects an inclination error through the gyroscope of the MPU module, which is sent to a power distribution algorithm that is then sent to ESC's (Electronic Speed Controller) which are passed on the coils of the brushless motors responsible for the flight.

Keywords: DRONE, RECUE, SOFTWARE, QUADCOPTER

LISTA DE ILUSTRAÇÕES

Figura 1 - Buzz Bomb V-1	13
Figura 2 - Drone Amber e Abraham.....	14
Figura 3 - Controle PID	16
Figura 4 - Exemplo	17
Figura 5 - Filtro de Média Móvel	18
Figura 6 - Valor Média Móvel	18
Figura 7 - Arduino.....	19
Figura 8 - Módulo nRF24I91	20
Figura 9 - Imagem ilustrativa do módulo MPU-6050	21
Figura 10 - Diagrama Esquemático VANT.....	25
Figura 11 - Diagrama Esquemático Controlador	25
Figura 12 - Construção do Controlador de voo (bagunça total)	34
Figura 13 - Materiais utilizados na construção do frame	34
Figura 14 - Pintura dos canos de PVC para hastes e pés do frame	35
Figura 15 - Shield acoplada ao drone.....	35
Figura 16 - Shield acoplada ao dispositivo do controlador	36
Figura 17 - Frame do drone finalizado.....	36

LISTA DE TABELAS

Tabela 1 - Preços	24
-------------------------	----

SUMÁRIO

1 INTRODUÇÃO	11
2 REFERENCIAL TEÓRICO.....	13
2.1 Uma Breve História dos Drones Militares	13
2.1.2 Desastres Ambientais	14
2.2 Recursos ou Componentes Utilizados.....	16
2.2.1 Controle PID (Proporcional, Integral, Derivativo).....	16
2.2.2 Filtro de Média Móvel.....	17
2.3.3 Arduino	18
2.4.4 nRF24I01 + PA + LNA.....	19
2.4.5 MPU-6050	20
3 METODOLOGIA.....	22
3.1 Análise do Problema	22
3.2 Solução do Problema.....	22
3.3 Método de Funcionamento	22
3.4 Aplicação.....	23
3.5 Características do Projeto.....	23
3.6 Tabela de Preços	23
3.7 Custo Benefício.....	24
3.8 Resultados Esperados	24
4 CONSTRUÇÃO DO PROJETO	25
4.1 Construção Teórica	25
4.1.1 Diagramas Esquemático	25
4.1.1.1 Diagrama Esquemático VANT	25
4.1.1.2 Diagrama Esquemático Controlador.....	25
4.1.2 Softwares	26

4.1.2.1 Software VANT.....	26
4.1.2.2 Software Controlador	31
4.2 Construção Prática.....	33
4.2.1 Construção do Frame	34
4.2.2 Fabricação das PCB's.....	35
4.2.2.1 VANT PCB	35
4.2.2.2 Controlador PCB	36
4.2.3 Resultados Finais.....	36
5 ANÁLISE DOS RESULTADOS.....	37
5.1 Resultados Obtidos	37
5.2 Possíveis Melhorias	37
6 CONSIDERAÇÕES FINAIS	38
REFERÊNCIAS.....	39

1 INTRODUÇÃO

Tragédias como as que aconteceram em Mariana em 2015 e o desastre de Brumadinho no início de 2019 mostraram o quanto nosso país é ineficiente nos resgates em zonas remotas. De acordo com dados coletados pelo G1 no mês de maio de 2019, foram identificadas 244 mortes e 26 pessoas ainda desaparecidas. Isso mostra a ineficiência das equipes de resgate em áreas de difícil acesso, pois estas sofrem com uma falta de investimento tecnológico em seus equipamentos de auxílio nas buscas, e que assim, diminuem sua produtividade e aumentam seus riscos fatais.

O projeto Ardurescue trata-se de um VANT (veículo aéreo não tripulado), com o objetivo de sobrevoar a área de busca, capturar imagens e enviá-las a um piloto, que poderá fazer a identificação do território de busca procurando vítimas, assim então, facilitando o trabalho dos bombeiros, diminuindo seus riscos e aumentando a produtividade das equipes de resgate.

Nos dias de hoje, as equipes de resgate brasileiras geralmente usam helicópteros na busca de pessoas e no reconhecimento de território, o que é de grande ajuda, pois estes veículos tripulam pessoas, e oferecem um campo de visão maior do local de atuação. Contudo, por se tratar de um veículo aéreo tripulado, sempre há riscos de uma pane de sistemas, o que pode ocasionar um acidente fatal. Já um drone, por se tratar de um veículo aéreo não tripulado, não há este risco, além de ser um serviço com menor custo, manutenções menos frequentes, treinamento de pilotos por um valor muito menor e risco de acidentes fatais zero.

Pensando em todas essas questões, para a construção do projeto serão utilizados os principais componentes: um módulo NRF24L01 com antena e etapa amplificadora para comunicação do veículo com o Rádio controlador, a escolha deste módulo foi feita pelo fato de ele ser de baixo custo e pelo bom range e estabilidade de sinal. Para sensores, será usado apenas uma MPU6050, este sensor trata-se de um acelerômetro e giroscópio, a escolha deste sensor é devida o baixo custo. O microcontrolador escolhido será o Atmega328p, devido apenas a facilidade de compra, uso e montagem. Com software, segundo as orientações dadas em fóruns de aeromodelismo, será utilizado apenas um algoritmo PID para fazer a estabilização do aeromodelo, e o filtro de média móvel para disponibilizar uma onda mais estável das leituras da MPU6050. A linguagem utilizada será a nativa do Arduino.

Com o projeto concluído pretende-se alcançar um veículo de exploração militar de alto nível, capaz de realizar imagens de qualidade, semelhantes a aeromodelos famosos do mercado, assim então, oferecendo um meio mais seguro e ágil para o corpo de bombeiros em suas buscas e resgates.

2 REFERENCIAL TEÓRICO

2.1 Uma Breve História dos Drones Militares

Drones, embora tenham se popularizado na década passada, é uma tecnologia muito antiga, datada desde a Segunda Guerra Mundial, pela Alemanha em lançamentos de bombas. Estes drones eram conhecidos por Buzz Bomb.

Este tipo de drone, apesar de poder voar apenas em linhas retas e em velocidade constante, obteve um sucesso considerável, mais de 1.000 lançamentos bem sucedidos pela Alemanha com o modelo V-1, que no decorrer da Segunda Guerra, criou o modelo V-2

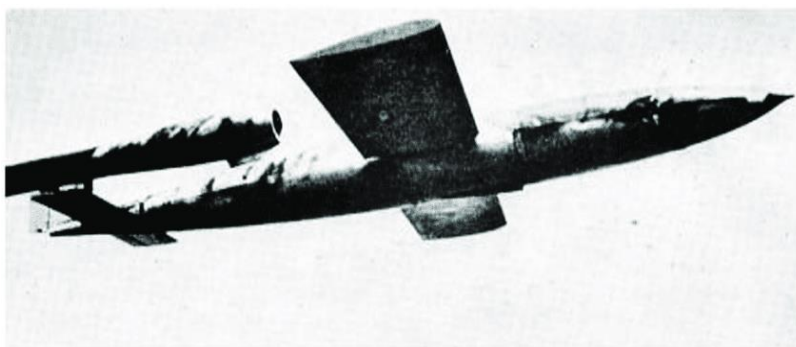


Figura 1 - Buzz Bomb V-1

O drone conhecido por todos hoje em dia foi inventado pelo Israelita Abraham Karem, engenheiro espacial. Segundo Abe (seu apelido) seu desejo com a Invenção destes VANT's era apresentar um veículo com os mesmos padrões já popularizados pelos aviões tripulados.

“eu só queria que os veículos aéreos não tripulados operassem com os mesmos padrões de confiabilidade e desempenho que os aviões tripulados” ~Abraham Karem.

O primeiro drone do engenheiro espacial chegou aos Estados Unidos em 1977. O drone conhecido por “O Aquila”, eram necessárias 30 pessoas para controlá-lo e voava apenas por alguns minutos, mesmo tendo autonomia por 20 horas de voo.

Após este “fracasso”, Abe fundou uma empresa, a Landing System, e utilizando poucos recursos tecnológicos como: fibra de vidro caseira e restos de madeira, deu origem ao Albatross. Este modelo apresentou melhorias surpreendentes com relação

ao modelo anterior, sendo as principais delas: Voos muito mais longos sem recarga de baterias (em torno de 56 horas no ar) e apenas 3 pessoas necessárias para operá-lo. Com este sucesso do modelo, o engenheiro recebeu financiamento da DARPA (Defense Advanced Research Projects Agency) para os aprimoramentos necessários de seu protótipo. Com este financiamento, Abe apresentou seu mais novo modelo, o famoso Amber, o VANT foi projetado e desenvolvido para operações militares do exército que ofereciam riscos à vida humana.



Figura 2 - Drone Amber e Abraham

Desde então, a tecnologia dos drones vem sendo aprimorada rapidamente. Hoje em dia, drones são usados nas mais diversas atividades possíveis como: lazer, filmagens de eventos, monitoramento de áreas, pulverização de plantações, entregas de produtos e no assunto que o projeto é focado, no auxílio de buscas e resgates.

2.1.2 Desastres Ambientais

Nos últimos anos, o mundo está sofrendo com diversos desastres, tanto ambientais como queimadas nas florestas por conta do clima árido, furacões nas costas dos mares principalmente nos Estados Unidos, terremotos, etc. Também, sofrendo com desastres de natureza humana, como as recentes quebras de barragens de Mariana e Brumadinho.

Todos estes desastres citados, tem uma coisa em comum, pessoas perdidas. De acordo com dados do G1, houveram 259 mortes e ainda 11 desaparecidos do desastre de Brumadinho (matéria de dezembro de 2019). Isto, mostra uma tremenda ineficácia das equipes de resgate na busca de desaparecidos, pois estes estão

desprovidos de equipamentos que poderiam aumentar seu desempenho nas buscas, podendo assim, salvar mais vidas.

2.1.3 Comprovação da Eficiência e Eficácia dos Drones de Auxílio ao Resgate

De acordo com o site “droneshowla”, em uma matéria de 9 de abril de 2019, 910 agências públicas nos Estados Unidos já estão usando drones diariamente. Esta mesma matéria mostra um caso de uma cidade dos Estados Unidos. A cidade chamada Mesa, fica na zona leste de Phoenix, é cercada pelo deserto de Sonora e conhecida por suas atividades de entretenimento ao ar livre. Contudo, são ocorridos cerca de 5 mil acidentes todos os meses no deserto próximo a esta metrópole, onde pessoas se perdem. Então, em 2015, o departamento de bombeiros e médicos de Mesa tomou a decisão que os ajudariam no monitoramento das trilhas do deserto e nas suas buscas.

“em 2015, nosso departamento foi um dos primeiros a implementar definitivamente um programa com drones no Arizona. Drones simplificam nosso trabalho e é uma grande ajuda” ~Bryan Kotsur, Vice Xerife e comandante de operações do Corpo de Bombeiros e Departamento Médico de Mesa.

Outra matéria no mínimo curiosa é a apresentada pelo site “Meio Bit”, há 6 anos atrás o caso que aconteceu em Wisconsin, onde, um oftalmologista de 82 anos que sofre de alguns problemas mentais e quase surdo desapareceu na cidade de Fitchburg. A Equipe de resgate desde senhor envolveu centenas de voluntários, cães farejadores e até mesmo um helicóptero, não conseguiram achar o idoso em 3 dias de buscas. Após os 3 dias, um operador de drone amador entrou no time de busca com seu drone. O operador varreu 81 hectares a cerca de 60 metros de altura e encontrou o senhor em apenas 20 minutos.

Estas são apenas algumas matérias que mostram a eficácia de drones de resgate em áreas extensas ou de difícil acesso.

2.2 Recursos ou Componentes Utilizados

2.2.1 Controle PID (Proporcional, Integral, Derivativo)

O controle PID trata-se de um algoritmo matemático, que tem por função o controle de uma variável em um sistema, permitindo ao sistema operar de forma mais estável no ponto de ajuste desejado, mesmo que ocorra variações ou distúrbios que afetam a estabilidade do algoritmo inteiro. Este algoritmo pode ser empregado com as mais diversas funcionalidades, uma delas na estabilização de drones.

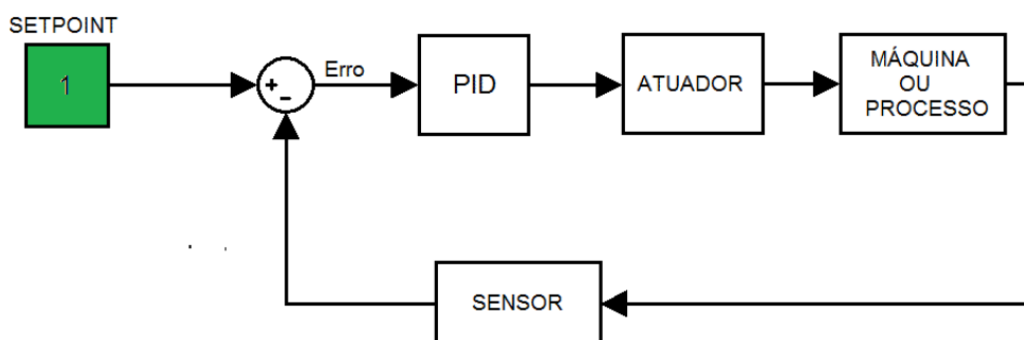


Figura 3 - Controle PID

Vamos a um exemplo prático. Imaginemos que temos uma barra horizontal de movimento livre assegurada por um eixo no meio a ponta de outra barra vertical. Contudo, a barra horizontal possui 2 motores com hélices, um em cada extremo, que serão responsáveis por equilibrar a barra horizontal e um sensor que detecta quantos graus de inclinação a barra está posicionada. A barra fica equilibrada por alguns minutos, porém, surge uma onda forte de vento que bate contra a barra e faz com que ela sofra uma inclinação de 15 graus.

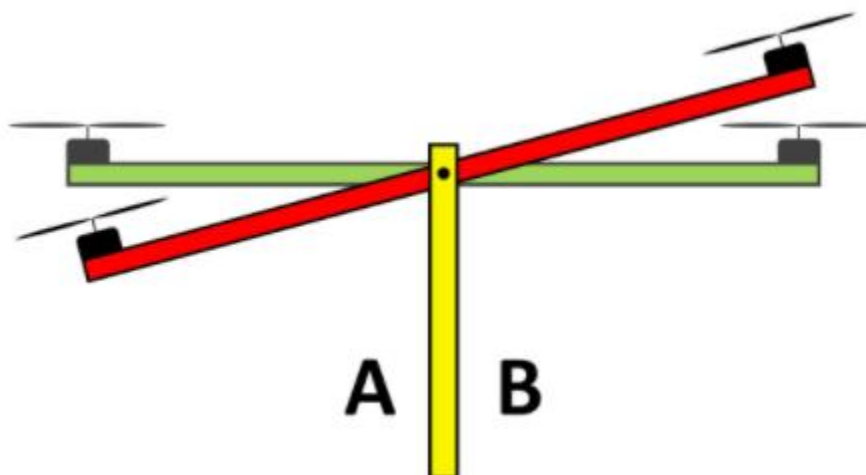


Figura 4 - Exemplo

Neste caso, o algoritmo PID será responsável por receber dados do sensor que indica o erro, e ajustar a potência dos dois motores gradualmente conforme as leituras do sensor (aumentando a potência do motor A e diminuindo do motor B), até que o sensor indique que não há mais inclinação.

2.2.2 Filtro de Média Móvel

O filtro de Média Móvel é um dos filtros digitais mais fáceis de serem atualizados e podem ser implementados nos mais diversos ambientes possíveis, desde bolsa de valores para previsão de valores futuros, até em leituras de sinais digitais para suavizar uma onda, assim então, deixando um sinal mais limpo e sem muitas incongruências causadas por ruídos (dados defeituosos lidos por um sensor).

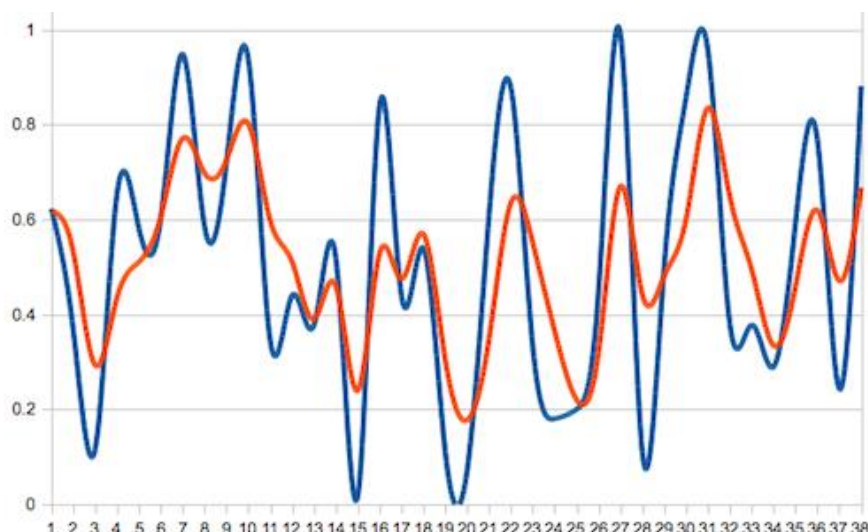


Figura 5 - Filtro de Média Móvel

O filtro de Média Móvel no drone tem o seguinte funcionamento: ele armazena seus valores em um vetor de N posições. Conforme o algoritmo for recebendo dados mais recentes ele irá atualizando o vetor mantendo apenas as últimas N leituras, deslocando-os a próxima posição, sempre mantendo o valor mais recente na primeira posição e o mais antigo na última, então os somando e realizando a média entre estes. O valor pela Média Móvel será o valor já filtrado.

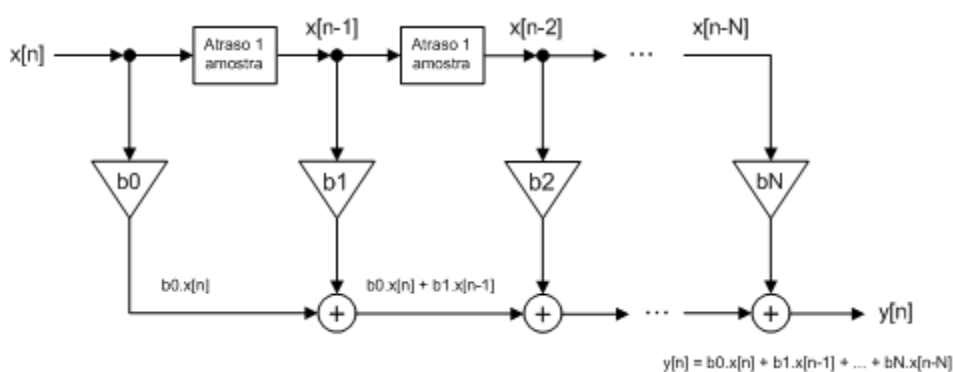


Figura 6 - Valor Média Móvel

2.3.3 Arduino

O Projeto Arduino trata-se de uma plataforma open-source e open-hardware de prototipagem. Controlada por um microcontrolador AVR Atmel, esta placa recebe instruções através de uma porta serial e as armazena no AVR, que poderá executar o software recebido. Não irei usar um Arduino completo no projeto final, onde será

usado o chamado Arduino Stand Alone, que se trata apenas dos componentes principais para a formação de um Arduino (fonte de alimentação, cristal de 16Mhz e capacitores de desacoplamento). O Atmel utilizado no projeto final e protótipo será o Atmega328p, o mais conhecido, presente na placa Arduino Uno que será usada no protótipo para enviar todo o software para o microcontrolador. O Atmel será o cérebro do projeto, que irá controlar os dados recebidos pelos sensores e processá-los e mandar os sinais aos atuadores.

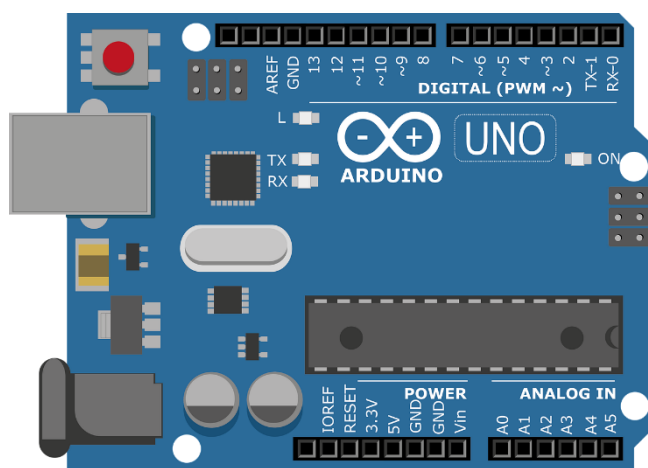


Figura 7 - Arduino

2.4.4 nRF24I01 + PA + LNA

O módulo NRF24L91, trata-se de um transceptor RF, capaz de comunicar plataformas de prototipagem através de uma conexão de 2.4Ghz a uma extensão de 250 metros sem obstáculos numa taxa de transmissão de 250Kbps a 2Mbps. O PA e LNA citados no título são os circuitos capazes de amplificar o sinal do módulo original a uma distância de aproximadamente 1km sem obstáculos. Serão usados 2 módulos destes no projeto, um acoplado no VANT e outro no controlador. Estes serão responsáveis por mandar e receber os dados TX-RX e RX-TX, o que dará os comandos ao drone e sinais no controlador.

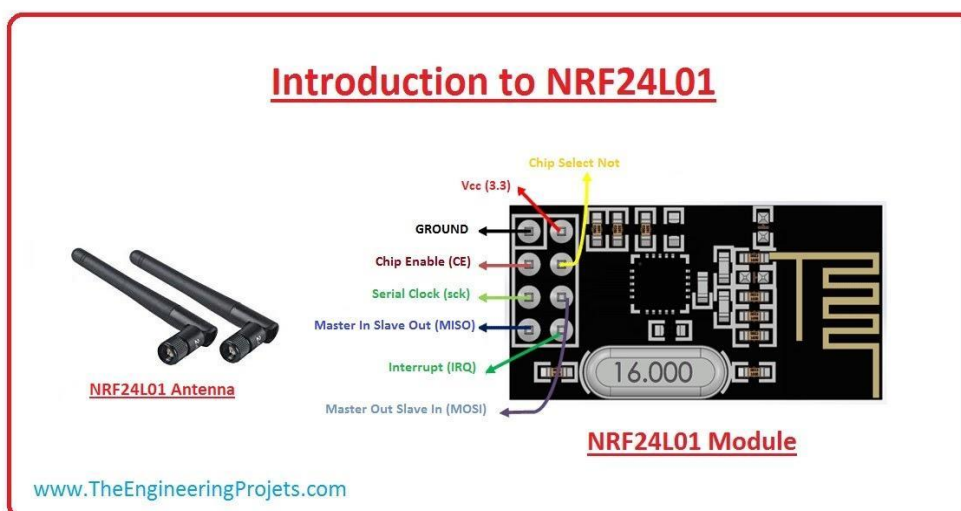


Figura 8 - Módulo nRF24l91

2.4.5 MPU-6050

O módulo MPU-6050 trata-se de uma placa que contém o chip GY-521. Este chip trata-se de um sensor de 3 eixos de giroscópio e acelerômetro, presente geralmente em robôs equilibristas. Não sendo o bastante, a placa também apresenta um sensor de temperatura, podendo fazer medições de -40 a 80°C.

Cada canal do módulo possui um conversor de analógico digital de 16 Bits, o que dá a capacidade do sensor de capturar os canais X, Y e Z ao mesmo tempo, trazendo-o assim, uma alta precisão na captura dos seus dados.

No projeto, ele será responsável por capturar dados do acelerômetro e giroscópio para o microcontrolador, que então irá calcular a inclinação e manter o sempre o drone no setpoint (posição) definido pelo controle.

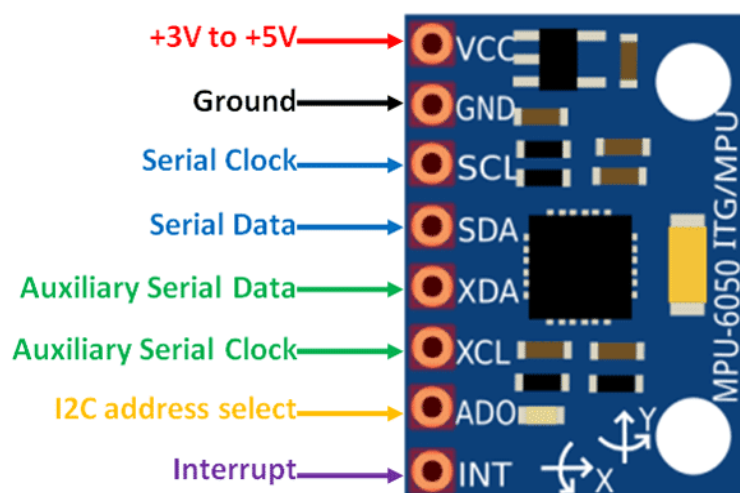


Figura 9 - Imagem ilustrativa do módulo MPU-6050

3 METODOLOGIA

3.1 Análise do Problema

Hoje em dia, o mercado não oferece um sistema específico de rastreamento ao o corpo de bombeiros e polícia militar, para localização de possíveis vítimas de desastres em áreas de difícil acesso, o que dificulta seus trabalhos de busca e resgate, diminui sua produtividade e aumentam os riscos de surgir possíveis acidentes aéreos, ocasionando assim mais vítimas.

3.2 Solução do Problema

O Ardurescue é um VANT (veículo aéreo não tripulado) de precisão, específico em busca e resgate em locais remotos, capaz de fazer imagens aéreas para localizar vítimas de uma maneira prática, fácil, eficiente e rápida, assim também reduzindo os riscos de acidentes aéreos fatais das equipes de buscas e resgate, aumentando sua produtividade e então fornecendo um auxílio mais eficiente, seguro e com menor custo comparado aos helicópteros utilizados hoje em dia.

3.3 Método de Funcionamento

O projeto funcionará com base em uma PCI controlada por um Atmega328P, programado em uma placa arduino. Além da placa principal do projeto, também será usado um MPU6050, um módulo GY NEO6Mv2, nRF24I01, um ou mais CI 's registradores de deslocamento 74HC585 e um optoacoplador 4n25.

O processo de voo do VANT será o seguinte: o MPU6050 e o módulo GPS irão fazer a leitura dos dados utilizados pelo drone, onde passará por um filtro de média móvel de “n” posições para fazer a filtragem dos dados lidos e ruídos dos sensores. Após a filtragem, os dados coletados irão passar por um algoritmo PID, que fará o controle dos 4 motores dos eixos X e Y, então aplicando as grandezas aos motores. Após os passos ditos, o algoritmo do drone fará a leitura da telemetria (controle) do veículo. Caso haja uma conexão e o drone esteja no ar, ele funcionará normalmente, caso não haja, o drone ficará parado no ar esperando comunicação do controle.

3.4 Aplicação

O projeto está sendo destinado ao auxílio no resgate de pessoas em locais remotos, porém, por se tratar de um drone de alta versatilidade, poderá ser usado nas mais diversas funcionalidades como o monitoramento de trânsito.

3.5 Características do Projeto

Com certeza, a principal característica do projeto em si é o software, pois este envolve diversos algoritmos que não são ensinadas em cursos técnicos, como por exemplo o uso de um registrador de deslocamento para expansão de portas digitais dos AVR's.

Outra característica marcante do projeto é o valor final dele, um drone com tantas capacidades de expansão de área, hoje em dia no Brasil custa em torno de sessenta mil reais, enquanto o custo de protótipo do drone não chegou a mil reais.

3.6 Tabela de Preços

Componente	Valor Unitário (R\$)	Valor Total (R\$)
Arduino Uno R3	R\$37,00	R\$37,00
Kit Arduino Uno	R\$15,00	R\$30,00
nRF24I01 + PA + LNA	R\$32,00	R\$64,00
MPU6050	R\$15,00	R\$15,00
Modulo de GPS	R\$74,00	R\$74,00
Fenolite de Dupla Face	R\$1,50	R\$3,00
Barra de Tubos de PVC	R\$12,00	R\$12,00
Joelhos de PVC	R\$1,50	R\$6,00
Emendas de PVC	R\$1,00	R\$4,00
Caps de PVC	RS1,00	R\$4,00
Filamento de PLA 1KG	R\$130,00	R\$130,00
74HC595	R\$1,50	R\$1,50
Sprays para acabamentos	R\$21,00	R\$84,00

Kit motor brushless + ESC's + Propellers	R\$437,00	R\$437,00
	TOTAL	R\$901,50

Tabela 1 - Preços

3.7 Custo Benefício

Bom, tratando-se de custo benefício, podemos citar de imediato que a vida humana não tem preço, cada segundo pode fazer diferença no resgate de uma pessoa que está em apuros. Mas tratando-se de números, drones específicos para busca de pessoas tem o custo em média de R\$50.000,00 a R\$85.000,00 (breve pesquisa no google sobre), o projeto final eu estimo que o drone terá o custo em média de 8 mil reais, com todas as tecnologias que podem ser aplicadas nele, então acho que é uma diferença bem considerável. Se falando de prazo de retorno, creio que seja de imediato, pois os combustíveis de aviação têm o preço muito elevado, e em resgates na procura de pessoas um helicóptero tem que sobrevoar áreas muito extensas para achar a pessoa, então creio que é de retorno imediato.

3.8 Resultados Esperados

Como dito anteriormente, os resultados esperados no fim do projeto serão de um drone de baixo custo, confiável e capaz de realizar imagens aéreas de locais que seriam difíceis de serem acessados por veículos maiores, mostrando qualidade e dando um auxílio a equipes de resgate ou de fiscalização desses locais.

4 CONSTRUÇÃO DO PROJETO

4.1 Construção Teórica

4.1.1 Diagramas Esquemático

4.1.1.1 Diagrama Esquemático VANT

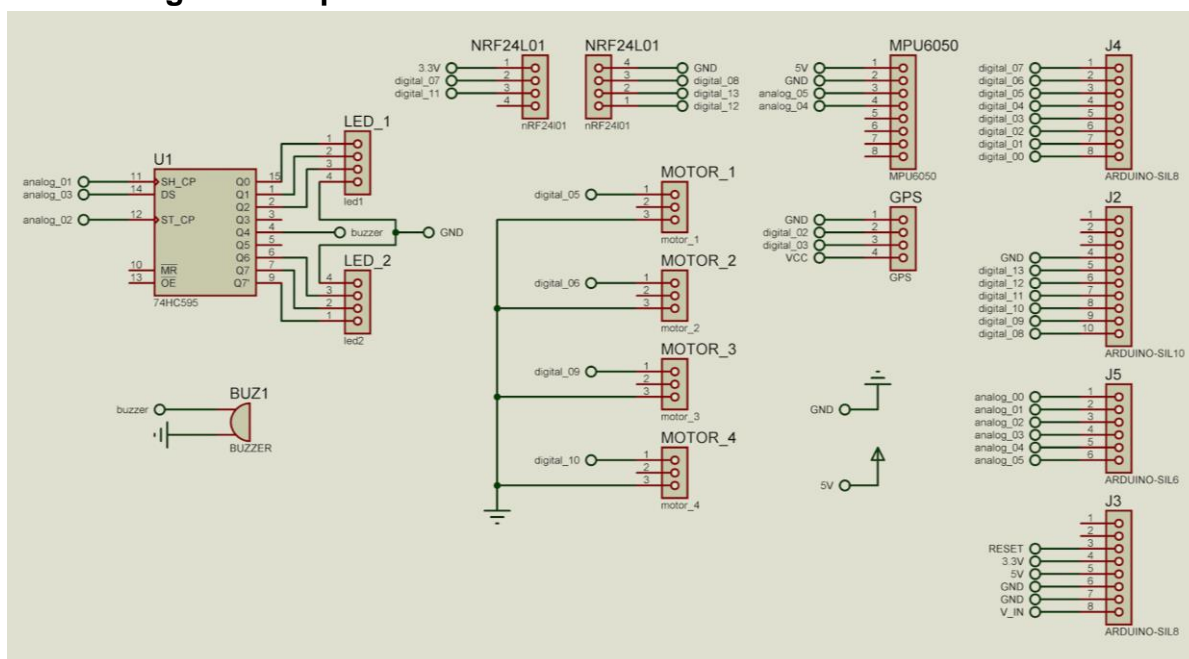


Figura 10 - Diagrama Esquemático VANT

4.1.1.2 Diagrama Esquemático Controlador

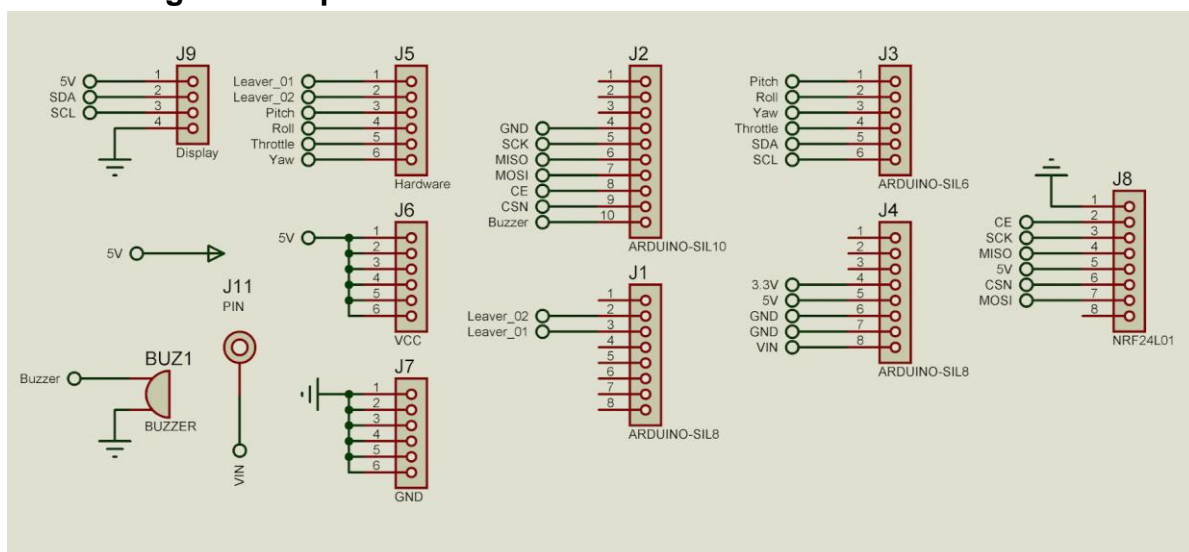


Figura 11 - Diagrama Esquemático Controlador

4.1.2 Softwares

4.1.2.1 Software VANT

```
// _____INCLUSÃO DE BIBLIOTECAS_____

#include <SPI.h>
#include <Wire.h>
#include <TinyGPS.h>
#include <SoftwareSerial.h>
#include <Servo.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <RF24_config.h>

// _____INSTANCIANDO OBJETOS_____

RF24 radio (7, 8);
const int MPU = 0x68;
SoftwareSerial serialGPS(3, 2);
TinyGPS gps;
Servo motor1;
Servo motor2;
Servo motor3;
Servo motor4;

// _____MAPEAMENTO DE HARDWARE_____

#define bateria A0
#define clk      A1
#define latch    A2
#define data     A3

// _____VARIÁVEIS DA TELIMETRIA_____

byte enderecos[][6] = {"1node", "2node"};
bool estadoAnteriorRadio, alertaRadio;
unsigned long timePerda, timeAlerta;
struct estruturaR {
    int throttleValue;
    int yawValue;
    int pitchValue;
    int rollValue;
    bool travamentoValue;
    bool modoValue;
```

```

};
struct estruturaE {
    int battery;
    float latitudeGPS;
    float longitudeGPS;
    unsigned short satellitesGPS;
};
typedef estruturaE dadosE;
typedef estruturaR dadosR;
dadosR dadosRecebidos;
dadosE dadosEnviados;

//_____VARIAVEIS PARA ESTABILIZAÇÃO_____
const int n = 15;
int valueAcgy[7][n];
float PIDx, PIDy, errorX, errorY, previous_errorX, previous_errorY, timed;
float  px = 0,      py = 0,
       ix = 0,      iy = 0,
       dx = 0,      dy = 0;
double kpx = 1.0,   kpy = 1.0,
       kix = 0.0,   kiy = 0.0,
       kdx = 0.0,   kdy = 0.0;
float setpointX = 0.0;
float setpointY = 0.0;
int motorA, motorB, motorC, motorD;
int conjunto1, conjunto2, conjunto3, conjunto4;

void setup() {
    //_____INICIANDO PERIFÉRICOS_____
    Serial.begin(9600);
    radio.begin();
    radio.openWritingPipe(enderecos[1]);
    radio.openReadingPipe(1, enderecos[0]);
    radio.startListening();
    Wire.begin();
    Wire.beginTransmission(MPU);
    Wire.write(0x6B);
    Wire.write(0);
    Wire.endTransmission(true);
    serialGPS.begin(9600);

```

```

// _____SETANDO PINOS_____
pinMode(bateria, INPUT);
pinMode(clk, OUTPUT);
pinMode(latch, OUTPUT);
pinMode(data, OUTPUT);
}

void loop() {
// _____COLORAÇÃO DOS LEDS_____
if (alertaRadio) {
    if ((millis() - timeAlerta) < 500) {
        shiftOut(data, clk, LSBFIRST, 0x66);
        attShift();
    }
    if ((millis() - timeAlerta) > 500) {
        shiftOut(data, clk, LSBFIRST, 0xD0);
        attShift();
    }
    if ((millis() - timeAlerta) >= 1000) {
        timeAlerta = millis();
    }
} else {
    if ((millis() - timeAlerta) < 250) {
        shiftOut(data, clk, LSBFIRST, 0xA8);
        attShift();
    }
    if ((millis() - timeAlerta) > 250) {
        shiftOut(data, clk, LSBFIRST, 0xE0);
        attShift();
    }
    if ((millis() - timeAlerta) >= 1000) {
        timeAlerta = millis();
        attShift();
    }
}

// _____TRABALHANDO COM DADOS DO GPS_____
bool gpsRecebido = false;
while (serialGPS.available()) {
    char gpsIN = serialGPS.read();
    gpsRecebido = gps.encode(gpsIN);
}

```

```

}

unsigned long idadeInformacaoGPS;
gps.f_get_position(&dadosEnviados.latitudeGPS,
  &dadosEnviados.longitudeGPS, idadeInformacaoGPS);
dadosEnviados.satellitesGPS = gps.satellites();

// _____MEDINDO E FAZENDO A MEDIA MOVEL DA MPU_____
for (int i = 0; i < 7; i++) {
  for (int ii = n - 1; ii > 0; ii--) {
    valueAcgy[i][ii] = valueAcgy[i][ii - 1];
  };
}

Wire.beginTransaction(MPU);
Wire.write(0x3B);
Wire.endTransmission(false);
Wire.requestFrom(MPU, 14, true);
valueAcgy[0][0] = Wire.read() << 8 | Wire.read();
valueAcgy[1][0] = Wire.read() << 8 | Wire.read();
valueAcgy[2][0] = Wire.read() << 8 | Wire.read();
valueAcgy[3][0] = Wire.read() << 8 | Wire.read();
valueAcgy[4][0] = Wire.read() << 8 | Wire.read();
valueAcgy[5][0] = Wire.read() << 8 | Wire.read();
valueAcgy[6][0] = Wire.read() << 8 | Wire.read();
float moving_avarage[7] = {0, 0, 0, 0, 0, 0};
for (int i = 0; i < 7; i++) {
  for (int ii = 0; ii < n; ii++) {
    moving_avarage[i] += valueAcgy[i][ii];
  }
}
for(int i = 0; i < 7; i++) moving_avarage[i] = moving_avarage[i] / n;
moving_avarage[3] = moving_avarage[3] / 340.00 + 36.53;
double pitch = atan(moving_avarage[0] / sqrt(moving_avarage[1] *
  moving_avarage[1] + moving_avarage[2] * moving_avarage[2])) * RAD_TO_DEG;
double roll = atan(moving_avarage[1] / sqrt(moving_avarage[0] *
  moving_avarage[0] + moving_avarage[2] * moving_avarage[2])) * RAD_TO_DEG;

// _____CALCULO PID DOS MOTORES_____
double errorX = setpointX - pitch;
double errorY = setpointY - roll;

```

```

float delta = (millis() - timed) / 1000.0;
timed = millis();
py = kpy * errorY;
px = kpx * errorX;
iy += (kiy * errorY) * delta;
ix += (kix * errorX) * delta;
dy = errorY * kdy / delta;
dx = errorX * kdx / delta;
PIDx = px + ix + dx;
PIDy = py + iy + dy;
motorA = dadosRecebidos.throttleValue + PIDx + PIDy;
motorB = dadosRecebidos.throttleValue - PIDx + PIDy;
motorC = dadosRecebidos.throttleValue + PIDx - PIDy;
// _____ESTRUTURA PRINCIPAL_____
radio.startListening();
if (radio.available()) {
    radio.read(&dadosRecebidos, sizeof(dadosR));
    estadoAnteriorRadio = HIGH;
} else {
    if (estadoAnteriorRadio == HIGH) {
        estadoAnteriorRadio = LOW;
        timePerda = millis();
    }
}
if (((millis() - timePerda) > 3000) && (estadoAnteriorRadio == LOW)) {
    alertaRadio = 1;
    return 0;
} else {
    alertaRadio = 0;
}
delay(21);
radio.stopListening();
radio.write(&dadosEnviados, sizeof(dadosE));
delay(21);
}

void attShift() {
    digitalWrite(latch, LOW);
    digitalWrite(latch, HIGH);
    digitalWrite(latch, LOW);
    return 0;
}

```

```
}
```

4.1.2.2 Software Controlador

```
// _____ INCLUSÃO DE BIBLIOTECAS _____
#include <RF24_config.h>
#include <RF24.h>
#include <nRF24L01.h>
#include <SPI.h>

// _____ INSTANCIANDO OBJETOS _____
RF24 radio (7, 8);

// _____ MAPEAMENTO DE HARDWARE _____
#define throttle    A4
#define yaw         A5
#define pitch       A2
#define roll        A3
#define travamento 3
#define modo        4

// _____ VARIÁVEIS DA TELIMETRIA _____
byte enderecos[][6] = {"1node", "2node"};
bool estadoAnteriorRadio, alertaRadio;
unsigned long timePerda, timeAlerta;
struct estruturaE {
    int throttleValue;
    int yawValue;
    int pitchValue;
    int rollValue;
    bool travamentoValue;
    bool modoValue;
};
struct estruturaR {
    int battery;
    float latitudeGPS;
    float longitudeGPS;
    float satellitesGPS;
};
typedef estruturaE dadosE;
typedef estruturaR dadosR;
```



```

dadosR dadosRecebidos;
dadosE dadosEnviados;

void setup() {
    // _____ INICIANDO PERIFÉRICOS _____
    Serial.begin(9600);
    radio.begin();
    radio.openWritingPipe(enderecos[0]);
    radio.openReadingPipe(1, enderecos[1]);
    radio.startListening();
}

void loop() {
    // _____ COLETANDO DADOS _____
    dadosEnviados.throttleValue = analogRead(throttle);
    dadosEnviados.throttleValue = 50;
    dadosEnviados.yawValue      = analogRead(yaw);
    dadosEnviados.pitchValue    = analogRead(pitch);
    dadosEnviados.rollValue     = analogRead(roll);
    dadosEnviados.travamentoValue = digitalRead(travamento);
    dadosEnviados.modosValue    = digitalRead(modos);

    radio.stopListening();
    radio.write(&dadosEnviados, sizeof(dadosE));
    delay(20);
    radio.startListening();
    if (radio.available()) {
        radio.read(&dadosRecebidos, sizeof(dadosR));
        Serial.println("DADOS RECEBIDOS");
        estadoAnteriorRadio = HIGH;
    } else {
        Serial.println("DADOS PERDIDOS");
        if (estadoAnteriorRadio == HIGH) {
            estadoAnteriorRadio = LOW;
            timePerda = millis();
        }
    }
    if (((millis() - timePerda) > 3000) && (estadoAnteriorRadio == LOW)) {
        alertaRadio = 1;
        return 0;
    } else {
        alertaRadio = 0;
    }
}

```

```
}  
delay(20);  
}
```

4.2 Construção Prática

A construção do projeto no início, embora conturbado deu tudo certo, pois comecei pela parte que mais iria causar problemas, os softwares. No desenvolvimento dos softwares eu comecei a escrever primeiramente a parte de recebimento e envio de dados através do módulo nRF24I01, demorou um pouco para eu conseguir, pois procurando na internet eu achava apenas programas de como enviar um tipo de dado, e no drone eu preciso enviar 3 tipos (float, int e bool), então tive que procurar muito para conseguir aprender como fazer tudo, mas consegui.

Acabando isto comecei a construção dos códigos da MPU, pois é o principal do drone, foi muito complicado achar o conteúdo que eu precisava também, pois até agora foram poucas pessoas que fizeram drones utilizando o Arduino como controlador, e os que fizeram usaram um firmware de uma startup americana, mas, depois de muito perturbar o professor e pedir explicações, consegui estabilizar o drone através de um tipo de balança para testes que eu fiz.

Após isto foram apenas modificações para maior comodidade, como a adição do GPS que foi fácil de programar. Os módulos controlados pelo 74HC595 que deu um pouco de trabalho, mas consegui também. O controlador eu não precisei programar, pois eu já tinha feito isto a um tempo por conta de outro projeto que eu tinha feito.

Após a construção do software então parti para a construção física do drone, cortes, pinturas e montagem dos canos que formam o protótipo, que não foi finalizado por várias complicações, mas é certo que iria funcionar.

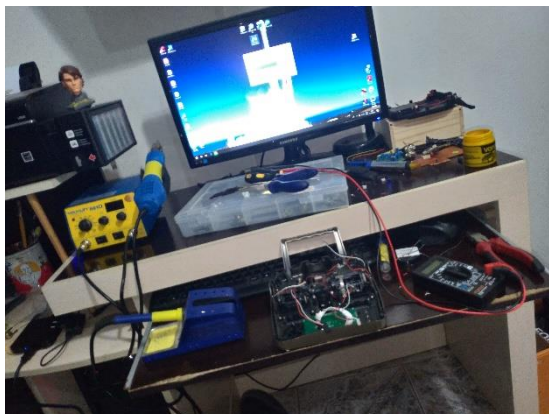


Figura 12 - Construção do Controlador de voo (bagunça total)

4.2.1 Construção do Frame

Pelo fato de o protótipo ter sido feito de PVC, os únicos trabalhos que tive que fazer foram cortar os tubos, isolar as pontas para não pegar tinta, pintar e encaixar tudo, demorou alguns dias para ser feito pois tentei fazer os cortes com maior precisão o possível, e a pintura fiz em camadas muito finas para que não criasse bolhas e ficasse uniforme. A pintura fiz com tinta spray, 2 camadas de primer PU por peça, 3 camadas de tinta para cobrir a peça por completo e por fim duas camadas de verniz, para que a tinta não descasque com facilidade.



Figura 13 - Materiais utilizados na construção do frame



Figura 14 - Pintura dos canos de PVC para hastes e pés do frame

4.2.2 Fabricação das PCB's

A Fabricação das duas PCB's do drone não foram exatamente o esperado, até o dia da apresentação, pois pelo fato do shield ter muitas trilhas para terminais de ligação de componentes, as trilhas tiveram que ficar muito estreitas, o que acarretou uma tonelada de problemas na fabricação, pois tentei fazer do modo transferência térmica. Poucas tentativas as trilhas grudavam na placa, e nas que grudavam geralmente mais da metade da PCB saia borrada. Então não consegui fabricar a PCB até o dia da apresentação e tive que fazer uma montagem na protoboard, mas, vou deixar aqui o design em 3D das PCB's

4.2.2.1 VANT PCB

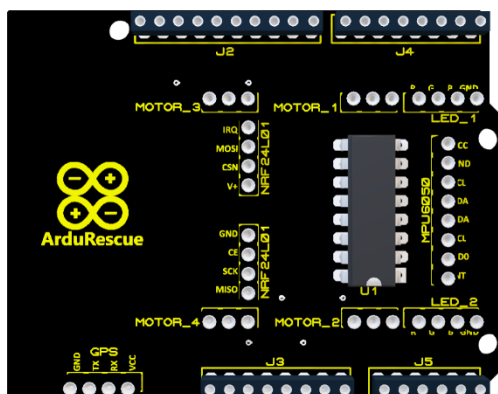


Figura 15 - Shield acoplada ao drone

5 ANÁLISE DOS RESULTADOS

5.1 Resultados Obtidos

Os resultados obtidos até a entrega deste projeto não foram exatamente os esperados. Ao longo desse projeto, ocorreram vários problemas na construção do mesmo, desde a parte do software que deu muito trabalho do começo ao fim para fazê-lo, até a parte da estrutura que teve problemas na montagem e pintura de tudo.

Fazer um drone não é uma tarefa fácil, precisa de muito estudo, planejamento do começo ao fim de tudo, cada detalhe precisa ser pensado e repensado antes de finalmente concluir a montagem final. Aprendi isso da pior maneira, porém estou contente com o resultado que tive, tendo em mente que este é o meu primeiro modelo de drone, acho que tive sucesso na maior parte do projeto.

5.2 Possíveis Melhorias

A principal melhoria para o projeto final é o desenvolvimento total da placa controladora de voo, pois este protótipo foi feito apenas uma shield para o arduino, para o projeto final seria construída uma controladora de voo do zero, pois além de ser mais compacto e ocupar menos espaço no drone, eu teria mais liberdade para adição de mais sensores sem me preocupar com espaço ocupado dentro do drone com os sensores off-board.

A próxima melhoria é a mais óbvia em si, a maior parte da estrutura é feita de PVC, no projeto final eu faria as hastes dos motores com tubos de fibra de carbono, que embora seja muito mais cara que o PVC, oferece uma resistência muito superior ao PVC. Os suportes para o motor ao invés de usar o material base de PLA, eu usaria o ABS, que embora seja um material mais difícil de se trabalhar, é mais fácil de dar acabamento, e como o suporte não precisa ser muito resistente pois não receberá impactos nem irá exercer muita força, o material é mais que suficiente. A escolha dele é simplesmente pelo valor da obra prima, o rolo de 1Kg custa em média de R\$80,00, enquanto o PLA custa em torno de R\$110,00.

6 CONSIDERAÇÕES FINAIS

Como dito anteriormente, um drone não é um projeto fácil, exige conhecimento nas mais diversas áreas para se fazer a construção, no começo eu não sabia exatamente o quão complicado seria fazer um, agora que tenho um pouco mais de experiência vejo que no fim o projeto inteiro está cheio de pequenas falhas que serão corrigidas no futuro.

Não me arrependi nem um pouco da escolha do projeto, pois embora tenha me dado muita dor de cabeça, desde a escolha do que seria feito até a hora da apresentação, foi algo que me trouxe as mais diversas experiências de como funciona a montagem de um produto, olhando para trás vejo que faria a mesma escolha, pois foi um projeto que me desafiou do começo ao fim, claro que agora faria quase tudo diferente, mas a base de tudo seria a mesma.

REFERÊNCIAS

Arduino. Disponível em <<https://pixabay.com/pt/illustrations/arduino-arduino-uno-tecnologia-2168193/>> Acesso em 2020.

Borges Corporation. Disponível em <<http://borgescorporation.blogspot.com/2013/05/filtro-de-media-movel.html>> Acesso em 2020.

Embarcados. Disponível em <<https://www.embarcados.com.br/controle-pid-em-sistemas-embarcados/>> Acesso em 2020.

EPXX. Disponível em <https://epxx.co/artigos/firfilter_pt.html> Acesso em 2020.

Maniarduino. Disponível em <<https://maniarduino.blogspot.com/p/mpu6050-gyroscope-interfacing.html>> Acesso em 2020.

Researchgate. Disponível em <https://www.researchgate.net/figure/German-V-1-buzz-bomb-2_fig8_316880797> Acesso em 2020.

Wyethdrone Technology. Disponível em <<https://wyethdronetechnology.weebly.com/the-history-of-drone-technology.html>> Acesso em 2020.