**ÁREA DE CONHECIMENTO: DESAFIOS CIÊNCIAS DA NATUREZA**

**CATEGORIA: MASTER**

**E. E. PROF DR ARIOVALDO DA FONSECA – DE TAQURITINGA – SP**

***Estudante(s): Alan Jonas F. Nascimento, Kaique Avanci da Silva, Pedro Henrique T. Guimaraes***

***Professor: Claudio Eduardo Aravechia de Sa, Caio R. Demiciano***

**Projeto Estratosfera**

**1. Resumo**

O projeto tem como objetivo o desenvolvimento e lançamento de um satélite meteorológico, com o auxílio de um balão que ao atingir determinada altitude será liberado, para a obtenção de dados e posterior estudo destes dados aplicando conceitos matemáticos. O satélite será equipado com sensores de pressão, temperatura e umidade, módulos de GPS, de gravação de cartão e de relógio para medir a temperatura, a umidade, a pressão e outras propriedades da atmosfera em diferentes altitudes. O Anexo I apresenta de forma detalhada esses sensores e módulos.

**2. Introdução**

Um satélite meteorológico é um dispositivo para medir vários parâmetros meteorológicos, como temperatura, pressão atmosférica, umidade relativa, velocidade e direção do vento, entre outros. Essas informações são coletadas através de instrumentos que são fixados ao balão, que pode atingir altitudes de até 40 km. O Satélite será lançado a atmosfera por um balão e ao atingir determinada altitude ele será liberado.

Os balões são normalmente feitos de látex ou de materiais sintéticos mais resistentes, como o polietileno, e são inflados com um gás leve, como o hélio. Eles são lançados em locais abertos e longe de obstáculos, como árvores ou edifícios, para garantir que possam subir livremente na atmosfera.

Inicialmente a altura máxima estipulada de 2 quilômetros será para testar a eficiência do protótipo e para que o raio de queda seja próximo ao município afim de verificar os dados coletados e recolher o protótipo.

O projeto final prevê que o satélite atinja a estratosfera até o balão de hélio estoure, com o intuito de coletar imagens e dados para a aplicação em ciências referente aos tópicos de estratosfera e curvatura da terra

**3. Justificativa**

O projeto tem como objetivo demonstrar a importância do jovem protagonista e o poder que o mesmo tem em desenvolver suas ideias, demonstrando que os conteúdos estudados têm valor prático na resolução de situações-problemas através de atividades interdisciplinares. Outro foco do projeto é a cultura “*maker*”, envolvendo a análise e elaboração de tabelas e gráficos, a construção do protótipo, bem como o entendimento teórico de situações apresentadas pela ciência, haja vista o nível de camadas que o projeto agrega, podendo ser trabalhado desde o 6° ano do ensino fundamental até o 1ªsérie do ensino médio, engajando os estudantes a pensarem em tecnologia e matemática não como disciplinas obrigatórias, mas ferramentas úteis no dia a dia.

**Pessoas em frente a mesa com bolo

Descrição gerada automaticamente**

Figura 1: Equipe do projeto Estratosfera.

**4. Questão Problema**

Diante do cenário tecnológico que estamos inseridos e a necessidade de aguçar o conhecimento pelas novas tecnologias foi desenvolvido o projeto Estratosfera. Como os estudantes podem associar esse projeto com a resolução de problemas?

**5. Metodologia**

Em cima do escopo inicial, foram elaboradas propostas pedagógicas, figura 2, para definir a capacidade e a complexidade do projeto.

Texto, Carta

Descrição gerada automaticamente Texto, Carta

Descrição gerada automaticamente

Figura 2: Esboco do projeto

Na área de matemática os dados coletados serão utilizados para a elaboração de gráficos os quais poderão ser interpretados pelos estudantes, figura 3.

A partir dos dados serão calculados a velocidade média do protótipo no percurso, podendo esses dados serem aplicados tanto na disciplina de matemática quanto na disciplina de física, figura 4.

Em uma versão complementar, pode-se utilizar de gravações visuais para demonstrar a curvatura do planeta de acordo com a altura atingida pelo protótipo no trajeto, podendo criar debates e observações nas disciplinas de ciências.

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

Figura 3: Exemplo de tabela e gráfico

Fonte: Autor

Texto

Descrição gerada automaticamente

Figura 4: Exemplo de cálculo de velocidade média

Fonte: Autor.

O escopo final prevê o lançamento de um protótipo de satélite com um balão de hélio e um sistema de paraquedas onde, ao atingir a altura máxima definida, o balão será desacoplado do satélite e o sistema de paraquedas auxiliará na descida para que o protótipo não estrague a ponto de perder os dados coletados que serão utilizados em sala de aula.

Inicialmente a altura máxima estipulada de 2 quilômetros será para testar a eficiência do protótipo e para que o raio de queda seja próximo ao município afim de verificar os dados coletados e recolher o protótipo.

O projeto final prevê que o satélite atinja a estratosfera até o balão de hélio estoure, com o intuito de coletar imagens e dados para a aplicação em ciências referente aos tópicos de estratosfera e curvatura da terra.

A estrutura física do satélite foi modelada através da ferramenta Sketchup e a impressão 3D foi realizada na impressora 3D.

Com a estrutura desenvolvida, foi utilizada a plataforma arduído composto por sensores e módulos para o desenvolvimento eletrônico do satélite.

Foi utilizado, também, a IDE arduído, Anexo VI, na elaboração do código na linguagem C++, Anexo VII, com o objetivo de desenvolver o código necessário para o funcionamento e coleta de dados dos sensores do sistema eletrônico desenvolvido.

**6. Expectativa de Resultado ou resultados**

Até o presente momento foram elaborados a parte física e eletrônica, apresentados nos Anexos I e II, sendo testado o sistema de paraquedas e a de coleta de dados (Hora, Temperatura, Altitude, Pressão, Umidade, Latitude, Longitude, número de Satélites), apresentado no Anexo III.

Com os dados coletados nos testes foi possível ver que os sensores funcionaram de forma adequada. O Anexo IV mostra o trajeto que foi realizado para o teste do GPS. O teste consistiu no deslocamento em um veículo do satélite entre duas cidades.

Também foram realizados teste de abertura do paraquedas que será utilizado no sistema, Anexo V.

O escopo final prevê o lançamento de um protótipo de satélite com um balão de hélio e um sistema de paraquedas onde, ao atingir a altura máxima definida, o balão será desacoplado do satélite e o sistema de paraquedas auxiliará na descida para que o protótipo não estrague a ponto de perder os dados coletados que serão utilizados em sala de aula.

**7. Contrapartida Social**

Devido a altura que o satélite atingirá no lançamento, não será possível lançar na unidade escolar, sendo necessário um local aberto e adequado pra tal prática.

Devido a isso, serão gravados desde o lançamento até o salvamento do satélite após a queda para postagem nas redes sociais da unidade escolar e apresentação a comunidade. Caso seja possível no local, será elaborado uma “*live*” para acompanhamento em tempo real do lançamento.

**8. Considerações Finais**

Após os testes foram feitos melhoramentos no sistema de soltura do satélite, figura 4, que se mostraram necessários para uma melhor eficiência do sistema. Todos os demais sistemas testados apresentaram o funcionamento desejado. Para próxima etapa do projeto está previsto o lançamento do satélite até uma altitude de 2.000 metros com o objetivo de testar de uma forma real os sistemas desenvolvidos. No Anexo II são apresentadas as figuras da montagem do satélite.

Em uma etapa posterior o satélite será lançado até a estratosfera objetivo final deste projeto.

**9. Referências**

* Currículo em Ação Matemática, Ciências da Natureza e Ciências Humanas - Ano Ensino Fundamental Anos Finais Caderno do Estudante.
* Arduino IDE 2.0.1.. Disponível em: <https://www.arduino.cc/en/software>. Acesso em: 12 out. 2022.
* Converter XLS para KML on-line. Disponível em: <https://mygeodata.cloud/converter/xls-to-kml>. Acesso em: 12 out. 2022.
* Previsões da trajetória do balão. Disponível em: <http://weather.uwyo.edu/upperair/balloon_traj.html>. Acesso em: 12 out. 2022.
* Disponível em: <https://blog.maua.br/2018/05/equipe-hab-imt-lanca-balao-meteorologico-no-espaco/> [BLOG DA MAUÁ](https://blog.maua.br/). Acesso em: 12 out. 2022.
* Projeto Estratosfera - E E Prof Dr Ariovaldo da Fonseca, 1º Encontro de Práticas Pedagógicas Inspiradoras. Disponível em: <https://youtu.be/XmQsE3AOKCY>. Acesso em: 12 out. 2022.

**11. Anexos**

**Anexo I**

**COMPONENTES UTILIZADOS**

Foram utilizados os sensores de pressão, temperatura e umidade BMP 180 e DTH11 para a obtenção destes parâmetros. O modulo de GPS GY-NEO-6M para a obtenção da latitude e longitude do dispositivo durante o seu trajeto, o módulo de gravação de cartão e o modulo DS1307 para a obtenção da data e hora dos parâmetros desejados.

|  |  |
| --- | --- |
| Placa Nano V 3.0 R3 Atmega328 - Sem Cabo USB compatível para Arduino | |
| Figura 5: Arduino Nano:  é uma placa pequena, baseada no ATmega328, completa e fácil de usar. Ele tem as mesmas funcionalidades do Arduino Uno em uma configuração compacta e destinada ao uso em Protoboards ou diretamente em Placas de Circuito Impresso. Ela tem uma entrada de alimentação DC e também funciona com um cabo USB Mini-B. | |
| Módulo Sensor de Temperatura DHT11 P/ Arduino ARDUÍNO Módulos Achei  Componentes Eletrônicos - Loja Virtual Módulos | Como usar com Arduino - Módulo Sensor de Umidade e Temperatura DHT11 - BLOG  MASTERWALKER SHOP |
| Figura 6: Sensor temperatura e umidade DTH11: é um sensor de temperatura e umidade que permite fazer leituras de temperaturas entre 0 a 50 Celsius e umidade entre 20 a 90%. | |
| Módulo Cartão Micro Sd Arduino Automação Pic Leitor Gravador em Promoção |  Ofertas na Americanas | Modulo Leitor Cartão Micro Sd Card (leitura/escrita) Arduino | Parcelamento  sem juros |
| Figura 7: Módulo cartão micro SD: possibilita efetuar a gravação de leitura de dados, a partir de um microcontrolador com interface SPI, em um cartão SD, suportando formatos de arquivo FAT16 e FAT32. | |
| Módulo Relógio RTC DS1307 Com Bateria - PISCALED | Arduino | Componentes  eletrônicos | Motor de passo | Sensores | Tela de jogo de vídeo game  Descrição gerada automaticamente com confiança média |
| Figura 8: Módulo DS1307: é um relógio em tempo real (RTC) que possui calendário completo e conta com 56 Bytes de SRAM. O mesmo é capaz de fornecer informações de hora, minutos, segundos, dia, data, mês e ano. | |
| Tela de computador com texto preto sobre fundo branco  Descrição gerada automaticamente com confiança média | Tela de computador com texto preto sobre fundo branco  Descrição gerada automaticamente com confiança média |
| Figura 9: Módulo de GPS GY-NEO-6M: é um sistema de posicionamento global que remete a utilização de satélites que tem como função fornecer informações sobre localização de um determinado dispositivo no globo terrestre. | |
| Électronique en amateur: Mesure de la pression dans un liquide avec BMP180  et Arduino | Controlando temperatura e pressão com o BMP180 - FilipeFlop.pdf - Avast Secure Browser |
| Figura 10: Sensor de pressão BMP 180:  um sensor barométrico utilizado em diversos projetos onde é necessário coletar as informações de pressão e temperatura de um ambiente com precisão. | |

Uma imagem contendo mesa

Descrição gerada automaticamente

Figura 11: Dispositivo de soltura de carga com servo motor MG 966 R.

Circuito eletrônico com fios

Descrição gerada automaticamente

Figura 12: Módulo I2c Pwm Servo Motor 16 Canais Pca9685 Arduino para controlar o dispositivo de soltura de carga.

**Anexo II**

|  |  |
| --- | --- |
| **FOTOS DA MONTAGEM DO SATÉLITE**  Câmera fotográfica preta  Descrição gerada automaticamente com confiança baixa  Figura 13: Impressão das plataformas na impressora 3D. | Figura 14: Satélite parcialmente desmontado. |
| Figura 15: Montagem inicial do protótipo do satélite. | Uma imagem contendo no interior, mesa, pequeno, computer  Descrição gerada automaticamente  Figura 16: Montagem inicial do protótipo do satélite. |
| Figura 17: Visualização do sistema de soltura do satélite | Figura 18: Sistema de soltura do satélite com corda acoplada. |
| Figura 19: Montagem final do satélite. | Figura 20: Montagem final do satélite. |

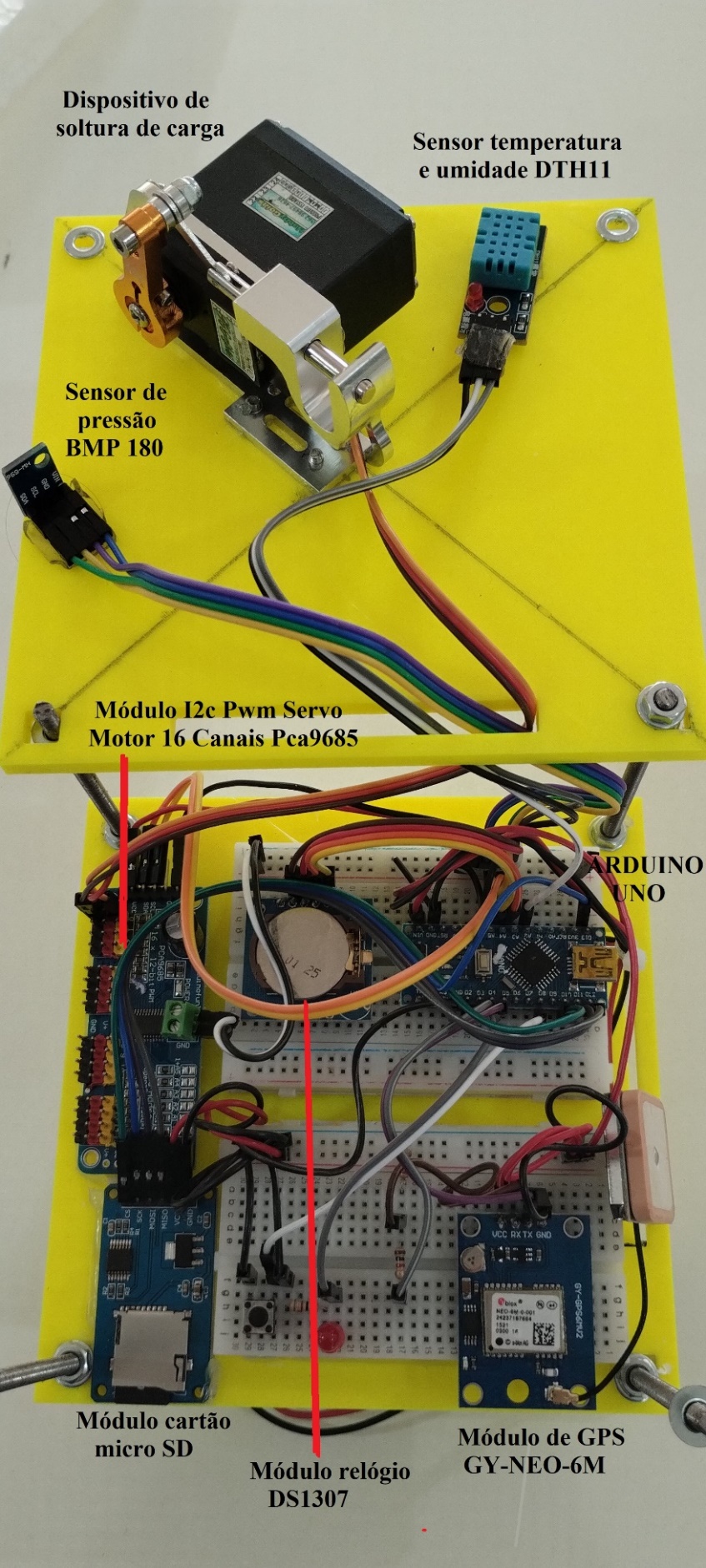


Figura 21: Indicação dos sensores e módulos utilizados no satélite.

**Anexo III**

**ARQUIVO DE DADOS OBTIDO PELO SATÉLITE**

Os dados obtidos, figura 22, são armazenados em um cartão para posterior análise dos dados obtidos. São armazenados os seguintes parâmetros: hora, temperatura, altitude, pressão atmosférica, umidade, latitude, longitude e números de satélites que estão conectados para a obtenção da longitude e latitude.

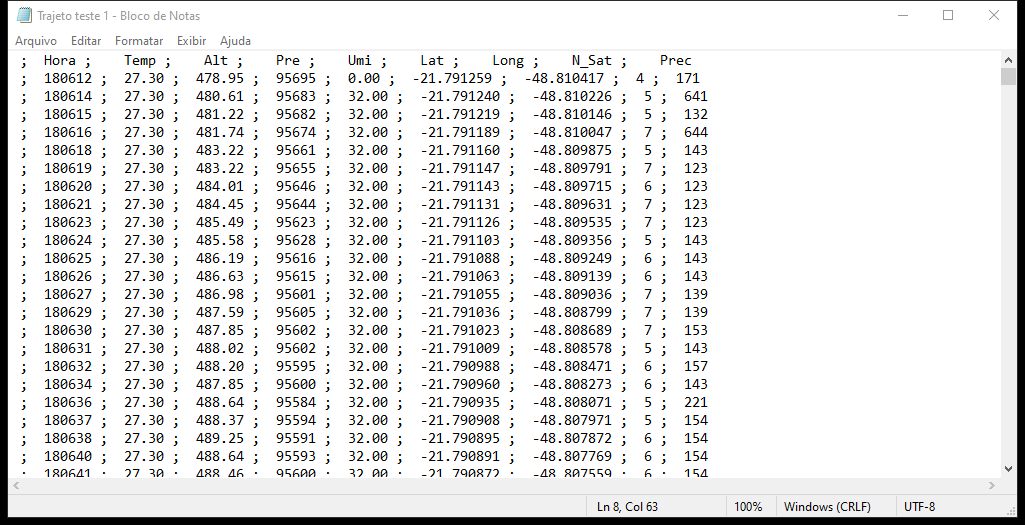


Figura 22: Exemplo dos dados obtidos.

**Anexo IV**

**TESTE DA COLETA DE DADOS DO GPS**

Para o teste da coleta de dados da longitude e latitude o satélite foi colocado em um carro e foi realizado um trajeto para coleta dos dados. Após isso os dados obtidos foram enviados para o Google Earth, figuras 23 e 24, para verificar se os dados condiziam com o trajeto realizado.

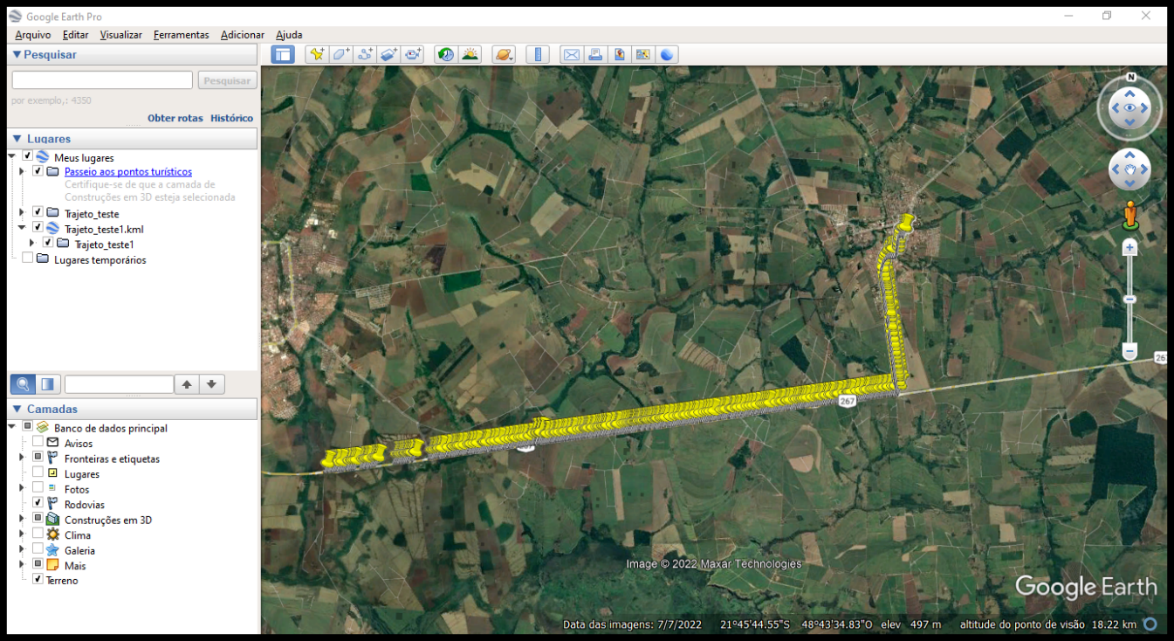


Figura 23: Trajetória visualizada no Google Earth do teste do GPS do sistema.

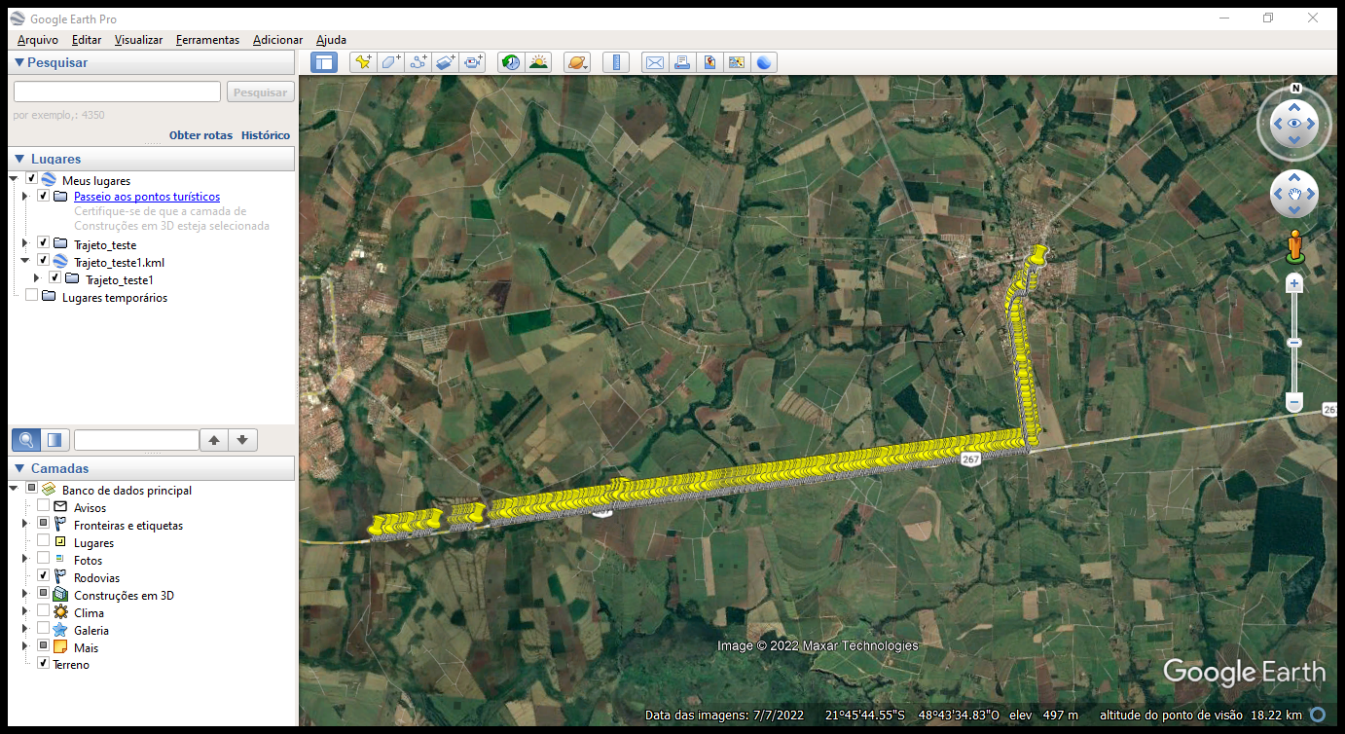


Figura 24: Trajetória visualizada no Google Earth do teste do GPS do sistema

**Anexo V**

**TESTE DA ABERTURA DO PARAQUEDAS**

As Figuras abaixo mostram os testes realizados para a abertura do paraquedas que será utilizado no sistema.

|  |  |
| --- | --- |
| Figura 25: Teste do paraquedas do sistema. | Figura 26: Teste do paraquedas do sistema. |
| Figura 27: Teste do paraquedas do sistema. | Figura 28: Teste do paraquedas do sistema. |

**Anexo VI**

**IDE ARDUINO**

A IDE do Arduino, figura 29, pode ser utilizada com as linguagens C/C++, e ela tem como utilidade programar a própria placa Arduino. Com ela, os programas são escritos, compilados e gravados na placa.

Tela de computador com fundo preto

Descrição gerada automaticamente

Figura 29: IDE Arduino.

**Anexo VII**

**CODIGO FONTE**

Código fonte em C++ para implementação do sistema de coleta de dados dos sensores utilizado no sistema arduino.

#include <SD.h>

#include <Adafruit\_BMP085.h>

#include <Adafruit\_GFX.h>

#include "dht.h"

#include "RTClib.h"  //INCLUSÃO DA BIBLIOTECA

#include <Adafruit\_PWMServoDriver.h>

#include <SoftwareSerial.h>

#include <TinyGPS.h>

#define GPS\_RX 4

#define GPS\_TX 3

#define GPS\_Serial\_Baud 9600

TinyGPS gps;

SoftwareSerial gpsSerial(GPS\_RX, GPS\_TX);

// INSTANCIANDO OBJETOS

Adafruit\_PWMServoDriver pwm = Adafruit\_PWMServoDriver();

// DECLARAÇÃO DE FUNÇÕES

void SoltarBalao();

void beginServos();

void writeServos(int nServo, int posicao);

void GravarDadosCartao();

byte pinoCS = 10;  //Pino CS don= modulo de gravacao de cartao

File myFile;

bool IniciaArquivo = true;

int Contador = 0;

int GravaCartao = 1;

float flat, flon;

unsigned long age;

// pushbutton para iniciar a gravacao do cartao

const int button = 8;  // entrada digital - pushbutton

const int ledPin = 2;  // saída digital -  LED

float AltitudeAbertura = 520;  // teste de abertura do servo

Adafruit\_BMP085 bmp180;

//PINO ANALÓGICO UTILIZADO PELO DHT11

const int pinoDHT11 = A2;

dht DHT;                   //VARIÁVEL DO TIPO DHT

RTC\_DS1307 rtc;  //OBJETO DO TIPO RTC\_DS1307

//DECLARAÇÃO DOS DIAS DA SEMANA

char daysOfTheWeek[7][12] = { "Domingo", "Segunda", "Terça", "Quarta", "Quinta", "Sexta", "Sábado" };

void setup() {

  // INICIALIZA O SERVO

  beginServos();

  delay(300);

  //COLOCA O SERVO NA POSICAO 90 GRAUS

  writeServos(0, 90);

  Serial.begin(GPS\_Serial\_Baud);

  gpsSerial.begin(GPS\_Serial\_Baud);

  // INICIALIZA O SENSOR BMP180

  if (!bmp180.begin()) {

    Serial.println("Sensor nao encontrado !!");

    while (1) {}

  }

  // INICIALIZA O RELOGIO DO SISTEMA

  if (!rtc.begin()) {                         // SE O RTC NÃO FOR INICIALIZADO, FAZ

    Serial.println("DS1307 não encontrado");  //IMPRIME O TEXTO NO MONITOR SERIAL

    while (1)

      ;  //SEMPRE ENTRE NO LOOP

  }

  if (!rtc.isrunning()) {               //SE RTC NÃO ESTIVER SENDO EXECUTADO, FAZ

    Serial.println("DS1307 rodando!");  //IMPRIME O TEXTO NO MONITOR SERIAL

    //rtc.adjust(DateTime(2018, 7, 5, 15, 33, 15)); //(ANO), (MÊS), (DIA), (HORA), (MINUTOS), (SEGUNDOS)

  }

  rtc.adjust(DateTime(F(\_\_DATE\_\_), F(\_\_TIME\_\_)));  //CAPTURA A DATA E HORA EM QUE O SKETCH É COMPILADO

  delay(100);                                      //INTERVALO DE 100 MILISSEGUNDOS

  pinMode(pinoCS, OUTPUT);  //Define o pinoSS como saida

  if (SD.begin())           //Inicializa o SD Card

  {

    Serial.println("SD Card pronto para uso.");  //Imprime na tela

  } else {

    Serial.println("Falha na inicialização do SD Card.");

    return;

  }

  // define pinos de saída e entrada

  pinMode(ledPin, OUTPUT);  //Define ledPin  como saída

  pinMode(button, INPUT\_PULLUP);

  digitalWrite(ledPin, LOW);

}

void loop() {

  if (digitalRead(button) == 0) {

    GravaCartao = 0;

    digitalWrite(ledPin, HIGH);

    Serial.println(" inicio");

    delay(150);

  }

  bool newData = false;

  unsigned long chars;

  // For one second we parse GPS data and report some key values

  for (unsigned long start = millis(); millis() - start < 1000;) {

    while (gpsSerial.available()) {

      char c = gpsSerial.read();

      // Serial.write(c); //apague o comentario para mostrar os dados crus

      if (gps.encode(c))  // Atribui true para newData caso novos dados sejam recebidos

        newData = true;

    }

  }

  if (newData) {

    SoltarBalao();

    ImprimirTela();

    if (GravaCartao == 0) {

      //     float flat, flon;

      //     unsigned long age;

      DHT.read11(pinoDHT11);  //LÊ AS INFORMAÇÕES DO SENSOR

      DateTime now = rtc.now();

      Serial.println(Contador);

      if (IniciaArquivo == true) {

        myFile = SD.open("gps.txt", FILE\_WRITE);  //Cria e abre o arquivo

        delay(100);

        myFile.print(" ;  ");

        myFile.print("Hora");

        myFile.print(" ;  ");

        myFile.print("  Temp");

        myFile.print(" ;  ");

        myFile.print("  Alt");

        myFile.print(" ;  ");

        myFile.print("  Pre");

        myFile.print(" ;  ");

        myFile.print("  Umi");

        myFile.print(" ;  ");

        myFile.print("  Lat");

        myFile.print(" ;  ");

        myFile.print("  Long");

        myFile.print(" ;  ");

        myFile.print("  N\_Sat");

        myFile.print(" ;  ");

        myFile.println("  Prec");

        myFile.print(" ;  ");

        IniciaArquivo = false;

        Contador = 0;

      }

      if (now.hour() > 10) {

        myFile.print(now.hour(), DEC);

      } else {

        myFile.print("0");

        myFile.print(now.hour(), DEC);

      }

      if (now.minute() < 10) {

        myFile.print("0");

        myFile.print(now.minute(), DEC);

      } else {

        myFile.print(now.minute(), DEC);

      }

      if (now.second() < 10) {

        myFile.print("0");

        myFile.print(now.second(), DEC);

      } else {

        myFile.print(now.second(), DEC);

      }

      myFile.print(" ;  ");

      // Temperatura

      if (bmp180.readTemperature() < 10) {

        myFile.print(bmp180.readTemperature(), 2);

        myFile.print(" ;  ");

      } else {

        myFile.print(bmp180.readTemperature(), 2);

        myFile.print(" ;  ");

      }

      //Altitude

      myFile.print(bmp180.readAltitude(), 2);

      myFile.print(" ;  ");

      // Pressao

      myFile.print(bmp180.readPressure());

      myFile.print(" ;  ");

      // Umidade

      myFile.print(DHT.humidity, 2);

      myFile.print(" ;  ");

      gps.f\_get\_position(&flat, &flon, &age);

      myFile.print(flat == TinyGPS::GPS\_INVALID\_F\_ANGLE ? 0.0 : flat, 6);  //LAT

      myFile.print(" ;  ");

      myFile.print(flon == TinyGPS::GPS\_INVALID\_F\_ANGLE ? 0.0 : flon, 6);  //LON

      myFile.print(" ;  ");

      myFile.print(gps.satellites() == TinyGPS::GPS\_INVALID\_SATELLITES ? 0 : gps.satellites());  //SAT

      myFile.print(" ;  ");

      myFile.println(gps.hdop() == TinyGPS::GPS\_INVALID\_HDOP ? 0 : gps.hdop());  //PREC

      myFile.print(" ;  ");

      Contador = Contador + 1;

      if (Contador == 100) {

        myFile.close();

        delay(500);

        IniciaArquivo = true;

        GravaCartao = 1;  // NAO GRAVA NO CARTAO

        Contador = 0;

        digitalWrite(ledPin, LOW);

      }

    }

  }

}

// abre o dispositivo para soltar o balao do satelite

void SoltarBalao() {

  // abertura do servo

  AltitudeAbertura = AltitudeAbertura + 10.0;

  Serial.print(AltitudeAbertura);

  if (AltitudeAbertura > 650.0) {

    //if (AltitudeAbertura > bmp180.readAltitude()) {

    writeServos(0, 0);

    AltitudeAbertura = 520;

    delay(1000);

    writeServos(0, 90);

  }

}

// imprime os valores dos sensores na tela

void ImprimirTela() {

  // le os outros sensores

  DHT.read11(pinoDHT11); //LÊ AS INFORMAÇÕES DO SENSOR

  DateTime now = rtc.now();

  Serial.print("Data: "); //IMPRIME O TEXTO NO MONITOR SERIAL

  Serial.print(now.day(), DEC); //IMPRIME NO MONITOR SERIAL O DIA

  Serial.print('/'); //IMPRIME O CARACTERE NO MONITOR SERIAL

  Serial.print(now.month(), DEC); //IMPRIME NO MONITOR SERIAL O MÊS

  Serial.print('/'); //IMPRIME O CARACTERE NO MONITOR SERIAL

  Serial.print(now.year(), DEC); //IMPRIME NO MONITOR SERIAL O ANO

  Serial.print('-');

  if (now.hour() > 10) {

    Serial.print(now.hour(), DEC); //IMPRIME NO MONITOR SERIAL A HORA

  }

  else

  {

    Serial.print("0"); //IMPRIME NO MONITOR SERIAL A HORA

    Serial.print(now.hour(), DEC); //IMPRIME NO MONITOR SERIAL A HORA

  }

  Serial.print(':'); //IMPRIME O CARACTERE NO MONITOR SERIAL

  if (now.minute() < 10) {

    Serial.print("0"); //IMPRIME NO MONITOR SERIAL OS MINUTOS

    Serial.print(now.minute(), DEC); //IMPRIME NO MONITOR SERIAL OS MINUTOS

  }

  else

  {

    Serial.print(now.minute(), DEC); //IMPRIME NO MONITOR SERIAL OS MINUTOS

  }

  Serial.print(':'); //IMPRIME O CARACTERE NO MONITOR SERIAL

  if (now.second() < 10) {

    Serial.print("0"); //IMPRIME NO MONITOR SERIAL OS SEGUNDOS

    Serial.print(now.second(), DEC); //IMPRIME NO MONITOR SERIAL OS SEGUNDOS

  }

  else {

    Serial.print(now.second(), DEC); //IMPRIME NO MONITOR SERIAL OS SEGUNDOS

  }

  Serial.println(); //QUEBRA DE LINHA NA SERIAL

  Serial.print("Temperatura : ");

  if ( bmp180.readTemperature() < 10)

  {

    Serial.print(bmp180.readTemperature());

    Serial.println(" C");

  }

  else

  {

    Serial.print(bmp180.readTemperature(), 1);

    Serial.println(" C");

  }

  Serial.print("Altitude : ");

  Serial.print(bmp180.readAltitude());

  Serial.println(" m");

  Serial.print("Pressao : ");

  Serial.print(bmp180.readPressure());

  Serial.println(" Pa");

  Serial.print("UMI. : ");

  Serial.print(DHT.humidity);

  Serial.println(" %");

  float flat, flon;

  unsigned long age;

  gps.f\_get\_position(&flat, &flon, &age);

  Serial.print("LAT=");

  Serial.print(flat == TinyGPS::GPS\_INVALID\_F\_ANGLE ? 0.0 : flat, 6);

  Serial.print(" LON=");

  Serial.print(flon == TinyGPS::GPS\_INVALID\_F\_ANGLE ? 0.0 : flon, 6);

  Serial.print(" SAT=");

  Serial.print(gps.satellites() == TinyGPS::GPS\_INVALID\_SATELLITES ? 0 : gps.satellites());

  Serial.print(" PREC=");

  Serial.print(gps.hdop() == TinyGPS::GPS\_INVALID\_HDOP ? 0 : gps.hdop());

  Serial.println();

  Serial.println();

}

// inicializaco dos servos

void beginServos() {

#define Frequencia 50  // VALOR DA FREQUENCIA DO SERVO

  pwm.begin();                 // INICIA O OBJETO PWM

  pwm.setPWMFreq(Frequencia);  // DEFINE A FREQUENCIA DE TRABALHO DO SERVO

}

void writeServos(int nServo, int posicao) {

#define SERVOMIN 125  // VALOR PARA UM PULSO MAIOR QUE 1 mS

#define SERVOMAX 525  // VALOR PARA UM PULSO MENOR QUE 2 mS

  int pos = map(posicao, 0, 180, SERVOMIN, SERVOMAX);

  Serial.print("ANGULO: ");

  Serial.println(pos);

  pwm.setPWM(nServo, 0, pos);

}