Altair – 4

CanSat Project – Miniature Satellite Simulation

Team NanoCurrent

- Mst Farhana Sadika – 0112330368
- Bijoy Das Gupta – 0112330355
- Md. Mahmudul Hasan – 0112330536
- Sadiya Sultana – 0112331056

Course: Electronics Laboratory

Instructor: Md. Muhaiminul Islam Nafi

Abstract - The CanSat project is an educational experiment aimed at bringing realworld satellite engineering concepts- into a simplified, small-scale model that fits within the size constraints of a soft drink can. This project focuses on designing, assembling, and programming a fully functional miniature satellite capable of collecting, transmitting, and analyzing environmental data. The CanSat is equipped with multiple sensors—such as a barometric pressure sensor for altitude measurement, an accelerometer for motion detection, and a temperature sensor to monitor environmental conditions. All collected data are processed by an Arduino microcontroller and transmitted wirelessly via a LoRa communication module to a ground receiving station in real-time. Beyond technical implementation, the project emphasizes teamwork, problem-solving, and system design integration. It challenges participants to balance constraints such as mass, power efficiency, and communication reliability while ensuring operational stability. Ultimately, the CanSat project demonstrates that complex aerospace principles can be practically implemented through low-cost, accessible technology, thereby serving as a valuable educational platform for students and researchers interested in space systems and embedded design.

Project Overview: The purpose of the CanSat project is to simulate major satellite functionalities on a miniature and manageable scale. A real satellite requires substantial investment, advanced manufacturing processes, and space launching capabilities, making hands-on learning nearly impossible at the student level. The CanSat bridges this gap by providing a ground-based, scalable model that operates as a micro-satellite for experimental and academic purposes. In this project, the CanSat acts as an autonomous sensing device capable of monitoring parameters like air pressure, altitude, acceleration, and temperature. The onboard Arduino Nano serves as the central control unit, executing data acquisition from sensors and coordinating LoRa communication with the ground station. This setup allows continuous telemetry, which is critical for assessing the satellite's operational performance and environmental interaction. The system is powered by a compact lithium-ion battery configured to sustain optimal performance during the acquisition and transmission phases. Additionally, modular design principles allow customization of payloads for specific research needs—such as atmospheric observation or flight trajectory analysis. From an educational perspective, this project provides an integrated learning approach that strengthens students' understanding of electronics, coding, wireless communication, and project management. The outcomes not only demonstrate the working principles of satellite subsystems but also encourage innovation in low-cost satellite development techniques for academic and research advancements.

Objectives:

1. Design & Build a CanSat Prototype – Construct a fully functional miniature satellite.

2. Collect Sensor Data – Gather environmental and motion data in real-time.

3. Transmit Data via LoRa – Establish a wireless connection between the CanSat and ground station.

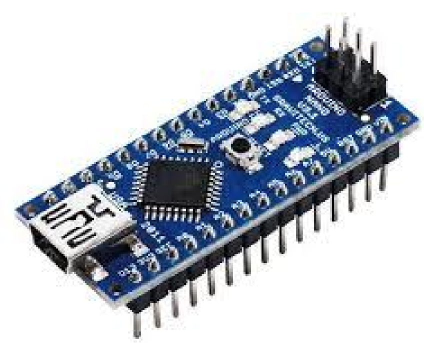
4. Analyze Transmitted Data – Interpret sensor readings for meaningful insights and performance evaluation.

Key Components :

- Arduino Nano R3: Main controller
- BMP280 temperature, pressure sensor
- Lipo Battery : Power supply
- Breadboard : Circuit connections
- DHT11 Temperature & Humidity Sensor
- ADXL345 Accelerometer: Measures acceleration
- SX1278 LoRa Module: Wireless data transmission
- LM-2596 Buck converter

-**1 .Arduino Nano – Main Controller**

The Arduino Nano is the central control unit of the CanSat. It reads data from all the connected sensors, processes this data, and sends it to the LoRa module for wireless transmission. Its small size and low cost make it ideal for student projects.



2 . BMP280 Temperature & Pressure Sensor

This sensor measures atmospheric pressure and temperature. The data collected is used to calculate the altitude of the CanSat during its flight.

It is compact, accurate, and energy efficient.

3. LiPo Battery – Power Supply

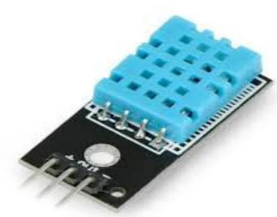
The LiPo battery provides power to the Arduino Nano, sensors, and the LoRa module. It is rechargeable, lightweight, and provides sufficient energy for the CanSat to function during the mission.

4. Breadboard – Circuit Connections

A breadboard is used to make temporary electrical connections between the microcontroller, sensors, and other modules. It allows for easy prototyping and testing before final assembly.

5. DHT11 Temperature & Humidity Sensor

This sensor measures ambient temperature and humidity. It provides additional environmental data, complementing the BMP280 sensor.



6. ADXL345 Accelerometer

The ADXL345 measures acceleration along the X, Y and Z axes. This helps to track the CanSat's movement, detect motion events, and analyze its flight trajectory.

7. SX1278 LoRa Module – Wireless Data Transmission

The SX1278 module transmits sensor data wirelessly to the ground station using LoRa communication.

It supports long-range, low-power data transfer, making it ideal for real-time telemetry.

8. **LM-2596 Buck Converter**

The LM-2596 buck converter regulates the battery voltage to a stable level required by the components. It ensures that sensors and modules receive safe and consistent power during operation.

9. Project Picture

10. Code Implementation

Reciver_MainCode.ino

```
#include <SPI.h>
#include <LoRa.h>

// LoRa pins
#define SS   10
#define RST  9
#define DIO0 2

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

  LoRa.setPins(SS, RST, DIO0);
  if (!LoRa.begin(433E6)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
  Serial.println("LoRa Receiver ready!");
}

void loop() {
  int packetSize = LoRa.parsePacket();
  if (packetSize) {
    String incoming = "";
    while (LoRa.available()) {
      incoming += (char)LoRa.read();
    }
```

Reciver main code

Trasmitter_MainCode.ino

```
#include <Wire.h>
#include <SPI.h>
#include <LoRa.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP280.h>
#include <Adafruit_ADXL345_U.h>
#include <DHT.h>

// --- DHT11 setup ---
#define DHTPIN 3        // DHT11 data pin connected to D3
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

// --- LoRa pins ---
#define SS   10
#define RST  9
#define DIO0 2

Adafruit_BMP280 bmp;
Adafruit_ADXL345_Unified accel = Adafruit_ADXL345_Unified(12345);

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

  // Init BMP280
  if (!bmp.begin(0x76)) {
    Serial.println("Could not find BMP280!");
    while (1);
  }
```

Transmitter code

Trasmitter_MainCode.ino

```
    while (1);
  }
  accel.setRange(ADXL345_RANGE_16_G);

  // Init DHT11
  dht.begin();

  // Init LoRa
  LoRa.setPins(SS, RST, DIO0);
  if (!LoRa.begin(433E6)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }

  Serial.println("LoRa Transmitter ready!");
}

void loop() {
  // --- Read BMP280 (Temperature, Pressure, Altitude) ---
  float temperature = bmp.readTemperature();
  float pressure    = bmp.readPressure() / 100.0F;
  float altitude    = bmp.readAltitude(1013.25);

  // --- Read ADXL345 (X, Y, Z) ---
  sensors_event_t event;
  accel.getEvent(&event);

  // --- Read DHT11 (Humidity only) ---
  float h = dht.readHumidity();
```

Trasmitter_MainCode.ino

```
  float temperature = bmp.readTemperature();
  float pressure    = bmp.readPressure() / 100.0F;
  float altitude    = bmp.readAltitude(1013.25);

  // --- Read ADXL345 (X, Y, Z) ---
  sensors_event_t event;
  accel.getEvent(&event);

  // --- Read DHT11 (Humidity only) ---
  float h = dht.readHumidity();

  // Build message (Only one Temperature now)
  String message = "Temp:" + String(temperature) + "C, " +
                   "Humidity:" + String(h) + "%, " +
                   "Pressure:" + String(pressure) + "hPa, " +
                   "Alt:" + String(altitude) + "m, " +
                   "X:" + String(event.acceleration.x) +
                   " Y:" + String(event.acceleration.y) +
                   " Z:" + String(event.acceleration.z);

  // Send via LoRa
  LoRa.beginPacket();
  LoRa.print(message);
  LoRa.endPacket();

  Serial.println("Sent: " + message);
  delay(2000); // send every 2 sec
}
```

Conclution : This CanSat project encourages students to design and build a fully functional miniature satellite at a low cost. By understanding and using these components, students gain hands-on experience with real satellite systems, learn data collection and wireless communication, and get inspired to explore aerospace and space research in the future.

References

[1] RBD-0032,Aurdino Nano

[2] RBD-0687,BMP-280 Sensor

[3] RBD-0664,ADXL-345 Sensor

[4] DHT-11 Sensor

[5] RBD-1647,SX-1278 Lora Module