International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 9 Issue 3, May-Jun 2025 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Design of A Satellite Communication System with Low Latency for Enhanced Data Transmission Enabled by AI-Driven Capabilities

Aisha Ndahi

Interplanetary Initiative, Arizona State University, United States of America

ABSTRACT

The aim of this work is to design a satellite communication system with low latency for enhanced data transmission enabled by A.I driven capabilities. This research presents the design of a low-latency satellite communication system that incorporates artificial intelligence (AI) to enhance data transmission speed, reliability, and autonomy. The design employed an experimental method, which involved the determination of path pathloss and received power after the selection of the appropriate frequency band, operation frequency, transmitter power, and antenna gain. The materials used include both software like Python and Arduino and hardware such as an Arduino board, servo motor, antenna, and Raspberry Pi. The system operates in the Ka-band (20 GHz downlink, 30 GHz uplink) using a geostationary Earth orbit (GEO) platform. We apply AI algorithms to perform real-time beamforming, dynamic frequency selection, and resource optimization, which ensure minimal latency and significantly reduce data loss. A comprehensive link budget analysis is conducted, incorporating a 100 W transmit power and high-gain antennas, alongside advanced modulation schemes (e.g., 16-APSK, 32-APSK) and LDPC coding to maximize throughput and signal integrity. The results show that the experimental project demonstrates a basic directional signal tracking system over a simple RF link using low-cost hardware, in which the integration of a signal generator with a servo-controlled antenna and RF transmission allowed for a basic simulation of signal movement and directionality. Although the 433 MHz modules don't support native analog RSSI, the workaround using an analog read circuit enabled signal strength estimation.

The system was stable in terms of signal generation and servo control, with latency readings very consistent under good conditions but varying slightly with environmental interference. It was concluded that with a comprehensive link budget analysis, the use of AI can be applied to develop a satellite communication system with a low latency for improved data transmission capabilities.

How to cite this paper: Aisha Ndahi "Design of A Satellite Communication System with Low Latency for Enhanced Data Transmission Enabled by AI-Driven Capabilities" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-9 | Issue-3, June 2025, pp.812-820, URL: www.ijtsrd.com/papers/ijtsrd81158.pdf

IJTSRD81158

KEYWORDS: *Satellite communication, low latency, artificial intelligence, beamforming, Ka-band, data transmission, predictive maintenance, anomaly detection, LDPC, modulation, Artificial intelligence*

1. INTRODUCTION

Satellite communication is the process of transmitting and receiving signals through space with the aid of satellites, which enables communication over long distances, often between two distant points on the planet, as the satellite orbits the earth. This transmission and reception of the communication signal is done through the communication link, which can be uplink, downlink, or crosslink. This link requires a special design in order to perform efficiently or optimally irrespective of various challenges. Satellite communication link design is defined as the planning and optimizing of the communication pathway between a satellite and Earth stations or other satellites as transmitter and receiver to ensure reliable and efficient transmission of data, voice, or video signals.

Satellite communication systems are foundational to numerous sectors, such as disaster management, transportation, defense, and financial services. However, these systems occasionally encounter significant challenges such as data loss and high latency, which can severely affect the safety, operational efficiency, and security. Satellite networks are one of the available methods that can solve the shortcomings of the long-range and short-range technologies, given that the advancement of satellite technology has reduced the cost of deployment and development of satellite networks as compared to the large space-based satellite networks that used bigger satellites (Mauro et al., 2015). Satellite technology is being recognised as one of the feasible solutions to connect the IoT devices scattered over the globe. Thus, the integration of IoT with satellite communication can be helpful in creating a global network by interconnecting a number of devices (Shilpi, 2012).

Aditi (2024) states that satellite communication has served as an important tool linking remote or underserved regions to the rest of the world by isolating constraints encountered in most ground-based networks. The opportunity to provide coverage and signal without interruption by geographical obstacles makes it very paramount in countering the digital divide in regions that have no or weak physical infrastructures. Apart from the communication satellite itself, Earth observation satellites use a number of technologies to collect observations, store raw data, and communicate the information to the earth stations. These techniques include direct transmission, storage, deferred transmission, and relay through communication satellites (Pooja and Gary, 2002).

Since the transmission and the reception frequencies for the satellites are different, the satellite has to communicate with two types of earth stations. The *uplink* earth station modulates the signals and radiates them to the satellite, which in most cases is a relay such that the satellite receives the signal and shifts the signal frequency, amplifies it, and then re-radiates it back to the earth, where it is received by *downlink* earth stations. This must be done or is possible if a satellite establishes a line-of-sight wireless link with the ground stations (Inglis et al., 1997).

In addition, as a result of the large distances between the satellite and the Earth, the signal, travelling at the speed of light, takes a long time to propagate to the Earth and then back to the satellite, as a complete round trip propagation delay for a satellite link between a GEO satellite and the Earth, which is approximately 36,000 km, is around 250 ms. This large delay has an adverse effect on the transmission of voice and video traffic. In the case of data communications involving transmission speeds of 10Mbps and higher, a huge amount of data transmitted by a source is temporarily in flight on the satellite link due to such a high propagation delay (Bleazard, 1985), which is a setback. According to Aditi (2024), satellite communication systems include complex technologies that are essential for the availability of international communication. Such systems include satellites that are in orbit, ground stations, and other components that entice end-to-end connectivity. In a satellite transmission link, the transmitter antenna on a satellite establishes a communication link with the receiver antennas for relaying information.

In addition, the transmission capacity depends upon the allocated satellite bandwidth for the link, and the bandwidth has to be efficiently utilized in case of multiple links (Pooja and Gary, 2002).

According to Fourati and Alouini (2021), satellite communication offers the prospect of service continuity over uncovered and under-covered areas, service ubiquity, and service scalability. However, several challenges must first be addressed to realize these benefits, as the resource management, network control, network security, spectrum management, and energy usage of satellite networks are more challenging than those of terrestrial networks. It is further reported that, in line with these trends, artificial intelligence (AI), including machine learning, deep learning, and reinforcement learning, has been steadily growing as a research field and has shown successful results in diverse applications, including wireless communication, and it has been discovered that the application of AI to a wide variety of satellite communication aspects has demonstrated excellent potential, including beam-hopping, anti-jamming, network traffic forecasting, channel modeling, telemetry mining, ionospheric scintillation detecting, interference managing, remote sensing, behavior modeling, space-air-ground integrating, and energy managing. In the views of Simon et al. (2024), low-latency communication is a major subject of investigation in the context of non-geostationary satellite orbit constellations, enabling a wide range of applications in different industries.

This research presents the design of a low-latency satellite communication system that incorporates artificial intelligence (AI) to enhance data transmission speed, reliability, and autonomy. The system operates in the Ka-band (20 GHz downlink, 30 GHz uplink) using a geostationary Earth orbit (GEO) platform. AI algorithms are applied to perform real-time beamforming, dynamic frequency selection, and resource optimization, thereby ensuring minimal latency and significantly reducing data loss.

2. Material and method

2.1. Materials:

The data or materials used in this work are primary data and secondary data. The secondary data consisted of library and online materials, while the primary data Arduino board, servo motor, antenna, Raspberry Pi, transceiver (transmitter, receiver), and WCMCU9833 signal generator were used for the work.

2.2. Method:

The research work was carried out using the experimental method. After the design of the satellite communication link, a prototype was constructed and implemented with artificial intelligence. The signal was transmitted and then received after some time to determine the duration or time taken for the signal to move from the transmitter to the receiver, which will determine or tell the delay. The system will be developed and tested using two Arduino boards. The WCMCU9833 signal generator was able to produce a clean sine wave at specified frequencies (e.g., 1 kHz), with stable output through SPI communication with Arduino A. The transmitter's servo rotated the antenna (or simulated directional signal) to track the signal. Both Arduinos will display key parameters (frequency, angle, signal strength, and latency) in real-time using I2C LCDs, enhancing usability and clarity.

3. Design processes of a satellite communication link

We have to ensure that the proposed device is compatible with all networking services.

Below are the steps to designing the above satellite communication.

Step 1: The first step is to determine the purpose of the satellite communication, and in this case, it is for real-time communication services. After determining the purpose of the satellite communication, the maximum acceptable latency and minimum required data rate for the application are then defined. In this design, the assumed acceptable latency is less than two hundred milliseconds (< 250 ms) while the minimum required data rate is 1 Gbps. In addition, one will select a suitable frequency band, such as Ka-band, K, or S-band, and this must be based on requirements as well as the communication regulations. In this work, the frequency band used is Ka-band, in which 20 GHz is for the downlink while 30 GHz is for the uplink.

Step 2: This design stage involves the choosing of the satellite orbit, such as geostationary orbit (GEO). Medium Earth Orbit (MEO) or Low Earth Orbit (LEO). But GEO (approximately 36,000 km GEO) is selected in this work even though some MEO or LEO satellite constellations provide communication services. Also, the transponder of the satellite will be selected, and here, a Ka-band transponder with 1 GHz bandwidth is selected. The design of the ground station is also done at this stage, and in this work, a 2.4-meter antenna with a Ka-band transmitter and receiver was chosen. The ground station can be a transmitting type, a receiving type, or both types, in which it has the capacity to transmit and to receive. This ground station is both receiving and transmitting; hence, the use of a 2.4-meter antenna with a Ka-band transmitter and receiver.

Step 3: This step involves the link budget calculation, or link budget design. This entails the calculation of the path loss and received signal strength based on the frequency and the transmit power, the gains of the antenna at the ground station as well as on the satellite in the orbit or the receiver. In this design, the power of the transmitter selected is 100W and the antenna gains are 40 dBi for the satellite antenna and 50 dBi for the ground station antenna.

From the parameters, the path loss calculated is 210 dB, while the received signal strength was calculated to be 120 dBm. The calculation of the received signal strength was done using transmit power, antenna gain, and path loss. These were done using the formulas

$$FPSL = 20\log^{(d)}{}_{10} + 20\log^{(f)}{}_{10} + 32.44 \quad (1)$$

Where $FPSL$ is the free space pathloss, d is the distance between transmitter and the receiver in km, f is the frequency of the signal transmission in MHZ,

$$P_r = P_t + G_{tx} + G_{rx} - FPSL - L \text{ (dB)} \quad (2)$$

Where Pt is the transmitter power in output at the antenna input. This is the amount of microwave carrier output power, usually expressed in dBm; L is losses due to the presence of atmospheric gases, vegetation, buildings, clouds, and fogs; G_{tx} = Transmit antenna gain, G_{rx} = Receive antenna gain, FSL = Free space path loss.

After this, the implementation of advanced modulation and coding schemes was carried out. The modulation schemes adopted here are advanced modulation schemes, such as 16-APSK or 32-APSK, which will help to increase data rate. This was followed by the error correction coding. And in this work, advanced error correction coding schemes, known as LDPC, were used to ensure reliable data transmission from the satellite to the ground station and vice versa.

Step 4: This step involves the application or implementation of an AI mechanism with Arduino. This deals with some optimisation methods that will help to improve the performance of the signal transmission and reception. In this work, AI-driven signal tracking using Arduino and AI-driven resource allocation is adopted. The signal tracking was adopted to enable the antenna to change from one direction or frequency to the other when the signal reception has reached the minimum acceptable value in the design. In this case, it will not be done manually but with the aid of AI. That is, implement AI-driven antenna signal tracking algorithms to optimize signal strength and reduce interference. Also, AI with Arduino can be used to dynamically allocate resources, such as bandwidth and power, and this is based on user demand and channel conditions. They have a rotational mechanism that searches for strong signals for data transmission (rotations) as they search for high signals, so the motor controls simulations, path loss, and antenna gain.

Step 5: This step deals with the implementation of AI-driven predictive maintenance and anomaly detection. It involves the application of predictive maintenance, which is made of AI-driven predictive maintenance, to forecast potential system failures or degradation in the system before knowing the decision to take automatically. This also involves the use of AI anomaly detection, in which the AI-driven mechanism helps to identify unusual patterns or behavior. In this step, the incorporation of the AI and Arduino with a Raspberry Pi for AI tasks is adopted.

Step 6: This is the integration and testing of the system stage. At this point, all components, including AI capabilities and Raspberry Pi, were integrated into the system or prototype. After the integration, the system is tested thoroughly to ensure that it is working as it should be. The testing involves the transmitting of a signal and receiving of the signal, and then determining the duration or lateness of the signal received. The quality of the signal received is also assessed along with other parameters from the experiment.

Step 7: This stage is the point where the system is deployed. The system has been tested and validated, and it is now proven to be working according to the design objectives. It moves to deploy the satellite and ground station while ensuring that all components are functioning in accordance with the design goals. The system can be monitored, and optimization done when necessary. In this design, the result of the experiment at the end of the last stage or step of the experiment will be discussed in (4.0) under results and discussion.

The architecture integrates modular hardware to simulate transmission conditions, fault detection, and latency monitoring. Preliminary results demonstrate improved system performance in both uplink and downlink scenarios. Future phases of this research will incorporate AI-driven security mechanisms to detect intrusions and safeguard communication integrity. The proposed system introduces a scalable, intelligent framework for next-generation satellite communications, particularly suitable for applications demanding high reliability and real-time responsiveness.

4. Results and discussion

4.1. Results

The system was successfully developed and tested using two Arduino boards: one as the transmitter and the other as the receiver. The key components of the operations and the results include

Signal Generation:

The WCMCU9833 signal generator was able to produce a clean sine wave at specified frequencies (e.g., 1 kHz), with stable output through SPI communication with Arduino A.

433 MHz Wireless Transmission:

The transmitter module reliably sent digital pulses synchronized with servo movements, allowing the receiver to capture pulse events as indications of angle changes.

Servo Motor Alignment:

The transmitter's servo rotated the antenna (or simulated directional signal) from 0° to 180° in steps, which was reflected in the receiver as inferred signal source directions. This was to enable the receiver to search for the higher signal, which will enable stable and improved data, which is the aim of the work.

Signal Strength Monitoring:

The receiver reads analog signal levels via a custom circuit or analog tap from the RX module, providing a crude but functional representation of signal strength (RSSI).

Latency Measurement:

Using timestamp comparison at signal receipt, the system could estimate latency between transmitted and received pulses with millisecond precision.

LCD Display:

Both Arduinos displayed key parameters (frequency, angle, signal strength, and latency) in real-time using I2C LCDs, enhancing usability and clarity.

4.2. Discussion

This experimental project demonstrates a basic directional signal tracking system over a simple RF link using low-cost hardware. Below are the key discussion points:

1. Functionality

The integration of a signal generator with a servo-controlled antenna and RF transmission allowed for a basic simulation of signal movement and directionality. Although the 433 MHz modules don't support native analog RSSI, the workaround using an analog read circuit enabled signal strength estimation.

2. Accuracy and Reliability

The system was stable in terms of signal generation and servo control. Signal reception was somewhat sensitive to noise and physical obstruction. In other words, the antenna needs a clear line of sight between the transmitter and the receiver for effective and improved signal reception. Latency readings were consistent under good conditions but varied slightly with environmental interference.

3. Challenges

The 433 MHz receiver modules are inherently noisy and low-fidelity, especially in urban environments. Accurate RSSI measurement required an external circuit (peak detector or amplifier). But the improvement can be achieved by replacing the 433 MHz modules with NRF24L01 for better reliability and built-in signal strength indicators. The use of actual data encoding (e.g., Manchester encoding) to transmit exact angle or frequency values instead of simple pulses will also be important. In addition, further results regarding the signal strength variation could not be carried out due to budget challenge as that would need an oscilloscope and other signal strength meters.

5. Conclusion

The design of a satellite communication system with low latency for enhanced data transmission enabled by AI capabilities has been carried out. This research presents the design of a low-latency satellite communication system that incorporates artificial intelligence (AI) to enhance data transmission speed, reliability, and autonomy. The design employed an experimental method and comprises the determination of loss and received power using frequency band, operation frequency, transmitter power, and the antenna gain. AI algorithms used in the work were to perform real-time beamforming, dynamic frequency selection, and resource optimization, thereby ensuring minimal latency and significantly reducing data loss. The results show that the experimental project demonstrates a basic directional signal tracking system over a simple RF link using low-cost hardware, in which the integration of a signal generator with a servo-controlled antenna and RF transmission allowed for a basic simulation of signal movement and directionality. Although the 433 MHz modules don't support native analog RSSI, the workaround using an analog read circuit enabled signal strength estimation. The system was stable in terms of signal generation and servo control, with latency readings very consistent under good conditions but varying slightly with environmental interference. From the result, it can be concluded that with a comprehensive link budget analysis, the use of AI can be adopted to develop a satellite communication system with a low latency for improved data transmission capabilities. Additional features of this design will include the integration of data protection mechanisms through advanced cybersecurity firmware to safeguard information against unauthorized access during transmission. Furthermore, the system will undergo rigorous performance testing to ensure optimal efficiency, reliability, and compliance with the advanced standards and operational requirements of both commercial and defense sectors. Below is the appendix presenting the simulation results of the initial prototype design conducted in the laboratory.

References

[1] Aditi Ranjit Kumar Verma (2024): Satellite Communication: Bridging the Gap to the Farthest Places on Earth and Enhancing 911 Emergency Calls. International Journal for Multidisciplinary Research. Volume 6, Issue 6, pp 1 – 10.

[2] Bleazard, G. B., Introducing Satellite Communications, NCC Publications, England, 1985.

[3] Fares Fourati and Mohamed-Slim Alouini (2021): Artificial intelligence for satellite communication: A review. Intelligent and Converged Networks. 2(3): 213–244

[4] Inglis, Andrew F., Luther, Arch C., *Satellite Technology: An Introduction,* Focal Press Publications, U.S.A, Second Edition, 1997.

[5] Mauro De Sanctis, Ernestina Cianca, Giuseppe Araniti, Igor Bisio, Ramjee Prasad (2015): Satellite Communications Supporting Internet of Remote Things, IEEE Internet of Things Journal

[6] Pooja J. Wagh and Gary Minden (2002): Design for a Satellite Communication Link in a Space Based Internet Emulation System. Technical Report. The University of Kansas Center for Research, Inc.

[7] Shilpi Singhal (2012): Applications and Security of Satellite based IoT Network. IJFANS International Journal of Food and Nutritional Sciences. Volume 11(2), pp 787 – 798

[8] Simon Heine, Christian Hofmann, and Andreas Knopp (2024): End-To-End Latency Evaluation Of Leo Satellite IoT Systems. https://www.researchgate.net/publication/381511092

APPENDIX I

Code for the programming

DOWNLINK ARDUINO CODE:

```
#include <SPI.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <RF24.h>
#include <Servo.h>

#define RED_LED     3
#define GREEN_LED   4
#define SERVO_PIN   6
#define CE_PIN      9
#define CSN_PIN     8

RF24 radio(CE_PIN, CSN_PIN);
LiquidCrystal_I2C lcd(0x27, 16, 2);
Servo antenna;

struct SignalPacket {
  float freq;
  unsigned long timestamp;
};

unsigned long lastReceived = 0;
unsigned long lastSweepUpdate = 0;
int sweepAngle = 0;
bool sweepingRight = true;
bool signalFound = false;

void setup() {
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  pinMode(RED_LED, OUTPUT);
  pinMode(GREEN_LED, OUTPUT);
  digitalWrite(RED_LED, HIGH);
  digitalWrite(GREEN_LED, LOW);
  antenna.attach(SERVO_PIN);
  antenna.write(sweepAngle);
  radio.begin();
  radio.setPALevel(RF24_PA_LOW);
  radio.setDataRate(RF24_1MBPS);
  radio.openReadingPipe(1, 0xF0F0F0F0E1LL);
  radio.startListening();
  lcd.clear();
}

void loop() {
  if (radio.available()) {
    SignalPacket packet;
    radio.read(&packet, sizeof(packet));
    unsigned long latency_ms = millis() - packet.timestamp;
    float latency_sec = latency_ms / 1000.0;
    int angle = constrain(map(packet.freq, 1000, 5000, 0, 180), 0, 180);
```

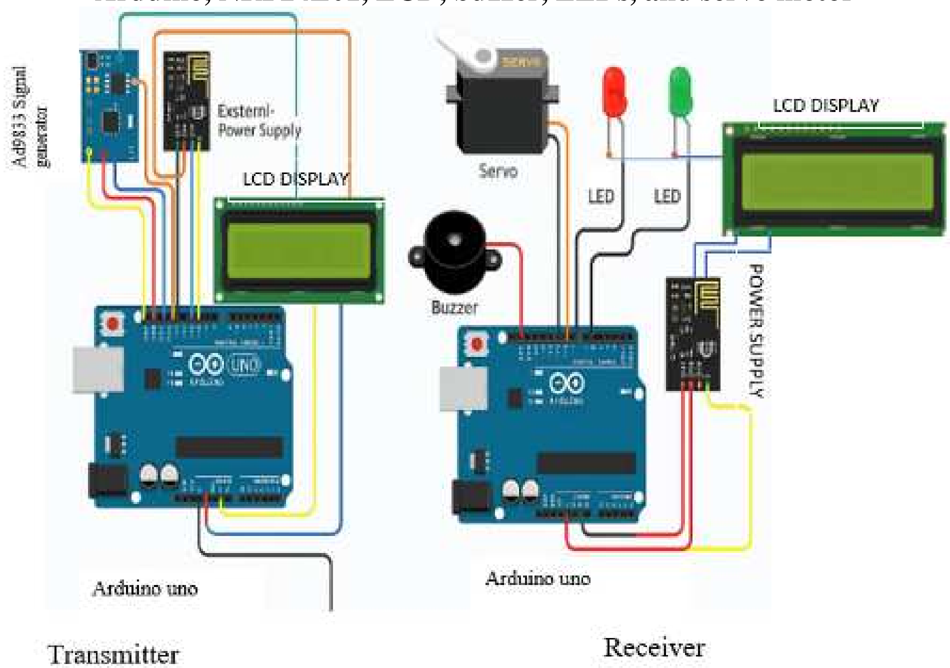
```
    antenna.write(angle);
    lastReceived = millis();
    signalFound = true;
    digitalWrite(GREEN_LED, HIGH);
    digitalWrite(RED_LED, LOW);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("lat:"); lcd.print(latency_sec, 2); lcd.print("s");
    lcd.setCursor(0, 1);
    lcd.print("ang:"); lcd.print(angle); lcd.print(" Rx:"); lcd.print(packet.freq, 0);
  }

  if (millis() - lastReceived > 2000) {
    signalFound = false;
    digitalWrite(GREEN_LED, LOW);
    digitalWrite(RED_LED, HIGH);
    if (millis() - lastSweepUpdate > 100) {
      lastSweepUpdate = millis();
      antenna.write(sweepAngle);
      if (sweepingRight) {
        sweepAngle += 5;
        if (sweepAngle >= 180) sweepingRight = false;
      } else {
        sweepAngle -= 5;
        if (sweepAngle <= 0) sweepingRight = true;
      }
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Searching...");
      lcd.setCursor(0, 1);
      lcd.print("ang:");
      lcd.print(sweepAngle);
    }
  }
}
```

UPLINK CODE (Arduino + NRF24L01 + AD9833 + LCD)

```
#include <SPI.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <RF24.h>
#include <AD9833.h>

// Pins
#define FSYNC 10  // AD9833 FSYNC
#define CE_PIN 9
#define CSN_PIN 8

AD9833 gen(FSYNC);  // Will Hickmott's AD9833 library
RF24 radio(CE_PIN, CSN_PIN);
LiquidCrystal_I2C lcd(0x27, 16, 2);  // Adjust if needed

struct SignalPacket {
  float freq;
```

```
  unsigned long timestamp;
};

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

  // LCD
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Uplink Init...");

  // Signal Generator
  gen.Begin();
  gen.setMode(MODE_SINE);
  gen.setFrequency(1000);  // Initial

  // NRF24L01
  radio.begin();
  radio.setPALevel(RF24_PA_LOW);
  radio.setDataRate(RF24_1MBPS);
  radio.openWritingPipe(0xF0F0F0F0E1LL);
  radio.stopListening();

  delay(1000);
  lcd.clear();
}

void loop() {
  float freq = random(1000, 5000);  // Random freq in Hz
  gen.setFrequency(freq);

  SignalPacket packet = { freq, millis() };
  bool success = radio.write(&packet, sizeof(packet));

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Tx Freq: ");
  lcd.print(freq, 0);
  lcd.setCursor(0, 1);
  lcd.print(success ? "Sent OK" : "Send FAIL");

  delay(1000);
}
```

Appendix II: circuit diagram showing the full connection of all hardware components including Arduino, NRF24L01, LCD, buzzer, LEDs, and servo motor



Appendix III: The transmitter and the receiver components of the AI enabled satellite communication system