LoRaTransceiver Library Documentation

Overview

The LoRaTransceiver library provides an easy-to-use interface for sending and receiving messages over LoRa using the Arduino platform. It includes error detection and automatic retransmission of messages and allows the user to customize the maximum number of retransmission attempts and the timeout period.

Constructor

LoRaTransceiver(int ss, int rst, int dio0, int maxNackCount = 5, int timeout = 2000)

The constructor for the LoRaTransceiver class. Initializes a new LoRaTransceiver object with the specified pin connections, maximum NACK count, and timeout period.

Functions

void begin(long frequency)

Initializes the LoRa module with the specified frequency.

void send(String message)

Sends a message over LoRa. If the message is not acknowledged within the timeout period, it will be automatically retransmitted up to the maximum NACK count.

String receive()

Checks for incoming messages over LoRa and returns the most recently received message. If no message has been received, returns an empty string.

bool checkForAck()

Checks for an acknowledgement of the most recently sent message. Returns **true** if an acknowledgement has been received, and **false** otherwise.

Usage

```
#include "LoRaTransceiver.h"

// Create a new LoRaTransceiver object
// SS pin, RST pin, DIO0 pin
LoRaTransceiver lora(10, 9, 2);

void setup() {
  Serial.begin(9600);
  while (!Serial);
  // Initialize the LoRa module with a frequency of 915 MHz
  lora.begin(915E6);
}

void loop() {
  // Send a message
  lora.send("Hello, world!");
  // Wait for an acknowledgement
  if (lora.checkForAck()) {
    Serial.println("Message acknowledged");
  } else {
    Serial.println("Message not acknowledged");
  }
  // Check for incoming messages
  String message = lora.receive();
  if (message != "") {
    Serial.println("Received message: " + message);
  }
  delay(1000);
}
```

Error Handling

The LoRaTransceiver library includes built-in error detection and handling. Each message is sent with a checksum, and if the checksum does not match on the receiving end, a NACK message is sent back. If a NACK message is received, the message is automatically retransmitted up to the maximum NACK count. If the maximum NACK count is reached without receiving an acknowledgement, the library gives up on the message and returns **false** from the **checkForAck()** function.

Limitations

The LoRaTransceiver library does not support addressing or sequencing of messages. Therefore, it is best suited for simple point-to-point communication where only one message is being sent at a time. For more complex communication scenarios, additional features may need to be implemented.