riseRFM95 Documentation

The `RFM95Communication` class enables communication over LoRa using the RFM95 module. It provides methods for sending and receiving data using LoRa.

Constructor

`RFM95Communication(uint8_t address)`

The constructor takes one argument `address` which is the address of the device. The address is used to distinguish between devices in a LoRa network.

Public Methods

`void setup()`

The `setup` method initializes the RFM95 module and sets the LoRa parameters such as frequency, spreading factor, bandwidth, and coding rate.

`void send(uint8_t toAddress, const SensorReading& reading)`

The `send` method takes two arguments: `toAddress` and `reading`. `toAddress` specifies the address of the device to which the data is being sent. `reading` is a `SensorReading` struct that contains the data to be sent.

The method encrypts the data using AES-256 and uses the CSMA/CA algorithm to avoid collisions with other devices on the LoRa network. It waits for a random time interval and checks if the channel is clear before transmitting the packet.

`bool receive(SensorReading& reading, uint8_t& fromAddress, uint8_t desiredAddress)`

The `receive` method takes three arguments: `reading`, `fromAddress`, and `desiredAddress`. `reading` is a `SensorReading` struct that contains the received data. `fromAddress` is the address of the device from which the data was received. `desiredAddress` is the address of the device that is intended to receive the data.

The method waits for a random time interval and checks if there is any data available on the channel. If there is data available, it decrypts the data using AES-256 and copies it to the `reading` struct. If the data is intended for the device (`desiredAddress` matches the device's address), it returns `true`. If there is no data available, it checks if the channel is clear. If the channel is clear, it returns `false`.

Private Variables

`uint8_t address_`

The `address_` variable is the address of the device.

`RH_RF95 rf95_`

The `rf95_` variable is an instance of the `RH_RF95` class, which is used to communicate with the RFM95 module.

`uint8_t receiveBuffer_[RH_RF95_MAX_MESSAGE_LEN]`

The `receiveBuffer_` variable is a buffer that is used to receive data from the LoRa network.

`byte key[32]` and `byte iv[32]`

The `key` and `iv` variables are used to encrypt and decrypt the data using AES-256. The `key` is a 256-bit key and the `iv` is a 128-bit initialization vector.

Example :

```
#include <RiseTelemetry.h>

RFM95Communication rfm95Communication(MY_ADDRESS);

void setup() {
  Serial.begin(9600);
  rfm95Communication.setup();
}

void loop() {
  // Create a sensor reading to send
  SensorReading reading;
  reading.temperature = 25;
  reading.humidity = 50;
  reading.lightLevel = 1000;

  // send it
  rfm95Communication.send(OTHER_ADDRESS, reading);
  delay(1000);

  // this struct will be updated to be the received struct
  SensorReading receivedReading;

  // after the receive function this uint_8 will contain the address of the sender of the received packet
  uint8_t fromAddress;
  if (rfm95Communication.receive(receivedReading, fromAddress, OTHER_ADDRESS)) {
    Serial.print("Received sensor reading from address ");
    Serial.print(fromAddress);
    Serial.print(": ");
    Serial.print(receivedReading.temperature);
    Serial.print(" ");
    Serial.print(receivedReading.humidity);
    Serial.print(" ");
    Serial.println(receivedReading.lightLevel);
  }
}
```