Arduino Wireless Audio Transmission Library For Guitar/Bass with effects

Generated by Doxygen 1.8.6

Tue Aug 23 2016 16:39:45

Contents

Chapter 1

Arduino Wireless Audio Transmission Library For Guitar/Bass with effects

This library allows streaming of audio from analog inputs via NRF24L01 radio modules See the `documentation section` for more info about the libraries used to generate the library RF24AudioJ

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Class Documentation

3.1 RF24AudioJ Class Reference

`#include <RF24AudioJ.h>`

Public Member Functions

- RF24AudioJ (RF24 &_radio, byte radioNum)
- void begin ()
- void begin_Tx ()
- void transmit ()
- void receive ()
- void broadcast (byte radioID)
- uint64_t getAddress (byte addressNo)

3.1.1 Detailed Description

Zebra193 2016 - RF24AudioJ: Arduino Realtime Audio Streaming library for Guitar/Bass (with effects)

This class implements an Audio Streaming library using nRF24L01(+) radios driven by the Optimized RF24 library (https://github.com/TMRh20/RF24) and the RF24Audio library (http://tmrh20.github.io/)

3.1.2 Constructor & Destructor Documentation

3.1.2.1 RF24AudioJ::RF24AudioJ (RF24 & *_radio,* byte *radioNum*)

Setup the radio and radio identifier

Note

Changing radioNum is only required if utilizing private node-to-node communication as opposed to broadcasting to the entire radio group

```
RF24 radio(48,49);                // Initialize the radio driver
RF24AudioJ rfAudioJ(radio,0);        // Initialize the audio driver
```

Parameters

_radio	The underlying radio driver instance
radioNum	The radio identifier

3.1.3 Member Function Documentation

3.1.3.1 void RF24AudioJ::begin ( )

Initialize the radio and audio library

Generally called in setup to initialize the radio as a Rx

```
rfAudioJ.begin();
```

3.1.3.2 void RF24AudioJ::begin_Tx ( )

Initialize the radio and audio library

Generally called in setup to initialize the radio as a Tx

```
rfAudioJ.begin_Tx();
```

3.1.3.3 void RF24AudioJ::broadcast (byte *radioID*)

Control of Private or Public Communication

Call this function to establish private communication between nodes in a radio group, or to switch back to public transmission.

Note

Using a radioID of 255 will disable private communication and broadcast to all nodes

```
rfAudioJ.broadcast(1);   // Only transmit audio to radio number 1
rfAudioJ.broadcast(255);  // Transmit audio to all radios in the group
```

Parameters

radioID	Set the radioID of the radio to communicate privately with.

3.1.3.4 uint64_t RF24AudioJ::getAddress (byte *addressNo*)

Get any of the preset radio addresses

Useful for listening nodes who wish to create private or additional radio groups The library has 14 predefined radio addreses. All radios listen/write on the first two addresses (0,1), and engage a private channel based on the radio number. Radio 0 listens on address 2, Radio 1 on address 3, etc.

```
uint64_t newAddress = rfAudioJ.getAddress(3);    // Gets the 3rd defined radio address
OR
radio.openReadingPipe(0,rfAudioJ.getAddress(7)); // Listens on the 7th defined radio address
```

Parameters

addressNo	Numbers 0 through 14 to access any part of the defined address array

Returns

RadioAddress: Returns the requested predefined radio address

3.1.3.5 void RF24AudioJ::receive ( )

Stop transmission through code

```
rfAudioJ.receive(); // Stop audio streaming
```

Call this function to stop transmission

3.1.3.6 void RF24AudioJ::transmit ( )

Control transmission through code

```
rfAudioJ.transmit(); // Begin realtime audio streaming
```

Call this function to begin transmission

The documentation for this class was generated from the following files:

- RF24AudioJ.h
- RF24AudioJ.cpp

Index