sub1GHz_RX_Layer Reference Manual

Generated by Doxygen 1.8.16

Fri Nov 8 2019 13:49:18

Chapter 1

sub1GHz_RX_Layer

Additional layer for EasyLink

Developed with `embedXcode+`

Author

Rei Vilo
`https://embeddedcomputing.weebly.com`

Date

05 Nov 2019 10:56

Version

103

Copyright

See also

ReadMe.txt for references

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 5

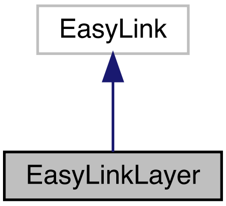
Data Structure Documentation

5.1 EasyLinkLayer Class Reference

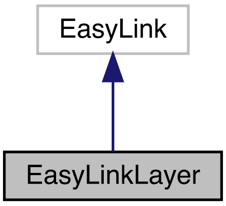
EasyLinkLayer.

```
#include <EasyLinkLayer.h>
```

Inheritance diagram for EasyLinkLayer:



Collaboration diagram for EasyLinkLayer:



Public Member Functions

- EasyLinkLayer (bool flagAddressFiltering=false)

  Constructor.
- EasyLink_Status begin ()

  Initialise and start the radio.
- EasyLink_Status transmit (const void ∗payload, size_t size)

  Send a message.
- EasyLink_Status receive (void ∗payload, size_t size, uint32_t ms=2000)

  Receive a message.
- EasyLink_Status setAddressFilter (uint8_t slot, AddressIEEE_t address)

  Set address to addresses filter.
- EasyLink_Status getAddressFilter (uint8_t slot, AddressIEEE_t ∗address)

  Get address to addresses filter.
- EasyLink_Status getAddressLocal (AddressIEEE_t ∗ieeeAddress)

  Get local address.
- void setAddressDestinationTX (AddressIEEE_t ieeeAddress)

  Set destination for TX message.
- void getAddressDestinationRX (AddressIEEE_t ∗ieeeAddress)

  Get destination from RX message.
- uint8_t getRSSIRX ()

  Get RSSI from RX message.

5.1.1 Detailed Description

EasyLinkLayer.

Additional layer with address filtering for EasyLink

Note

Basic usage: begin(), transmit() and receive()

5.1.2 Constructor & Destructor Documentation

5.1.2.1 EasyLinkLayer()

```
EasyLinkLayer::EasyLinkLayer (
            bool flagAddressFiltering = false )
```

Constructor.

Parameters

in	*flagAddressFiltering*	Enable address filtering, default=false

Note

Basic usage: no parameter, no filtering

Warning

Both RX and TX need to have the same configuration

5.1.3 Member Function Documentation

5.1.3.1 begin()

```
EasyLink_Status EasyLinkLayer::begin ( )
```

Initialise and start the radio.

Note

Basic usage: begin()

Returns

EasyLink_Status_Success or EasyLink_Status_Param_Error

5.1.3.2 getAddressDestinationRX()

```
void EasyLinkLayer::getAddressDestinationRX (
             AddressIEEE_t * ieeeAddress )
```

Get destination from RX message.

Parameters

out	*ieeeAddress*	destination address, uint8_t[8]

Warning

Not part of basic usage

5.1.3.3 getAddressFilter()

```
EasyLink_Status EasyLinkLayer::getAddressFilter (
            uint8_t slot,
            AddressIEEE_t * address )
```

Get address to addresses filter.

Parameters

in	*slot*	0, 1 or 2
out	*address*	IEEE address read from filters

Returns

EasyLink_Status_Success or EasyLink_Status_Param_Error

Note

The filter contains up to three addresses.

Warning

Not part of basic usage

5.1.3.4 getAddressLocal()

```
EasyLink_Status EasyLinkLayer::getAddressLocal (
            AddressIEEE_t * ieeeAddress )
```

Get local address.

IEEE address uint8_t[8]

Parameters

out	*ieeeAddress*	pointer to uint8_t[8]

Returns

EasyLink_Status

5.1.3.5 getRSSIRX()

```
uint8_t EasyLinkLayer::getRSSIRX ( )
```

Get RSSI from RX message.

Returns

RSSI level

5.1.3.6 receive()

```
EasyLink_Status EasyLinkLayer::receive (
            void * payload,
            size_t size,
            uint32_t ms = 2000 )
```

Receive a message.

Parameters

out	*payload*	pointer to the payload
out	*size*	size of the payload
in	*ms*	period to receive, default=2 seconds, time-out after

Returns

EasyLink_Status_Success or EasyLink_Status_Param_Error

Note

Maximum payload size is EASYLINK_MAX_DATA_LENGTH=128

Basic usage: receive(&payload, &size)

5.1.3.7 setAddressDestinationTX()

```
void EasyLinkLayer::setAddressDestinationTX (
            AddressIEEE_t ieeeAddress )
```

Set destination for TX message.

Parameters

in	*ieeeAddress*	destination address, uint8_t[8]

Warning

Not part of basic usage

5.1.3.8 setAddressFilter()

```
EasyLink_Status EasyLinkLayer::setAddressFilter (
            uint8_t slot,
            AddressIEEE_t address )
```

Set address to addresses filter.

Parameters

in	*slot*	0, 1 or 2
in	*address*	IEEE address to add to filters

Returns

EasyLink_Status_Success or EasyLink_Status_Param_Error

Note

Up to three addresses can be added to the filter.
Recommended allocation of addresses

- **Hub**
  - not used
  - specific IEEE address of the hub
  - generic address for commissioning nodes to hub
- **Node**
  - general broadcast from hub to all nodes
  - specific IEEE address of the node
  - not used

Warning

Not part of basic usage

5.1.3.9 transmit()

```
EasyLink_Status EasyLinkLayer::transmit (
            const void * payload,
            size_t size )
```

Send a message.

Parameters

in	*payload*	pointer to the payload
in	*size*	size of the payload

Returns

EasyLink_Status_Success or EasyLink_Status_Param_Error

Note

Maximum payload size is EASYLINK_MAX_DATA_LENGTH=128

Basic usage: transmit(payload, size)

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

- EasyLinkLayer.h
- EasyLinkLayer.cpp

Chapter 6

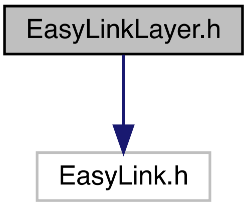
File Documentation

6.1 EasyLinkLayer.h File Reference

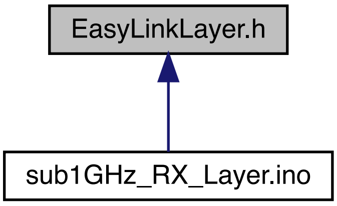
Library header.

```
#include "EasyLink.h"
```

Include dependency graph for EasyLinkLayer.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class EasyLinkLayer

  EasyLinkLayer.

Macros

- #define EASYLINKLAYER_RELEASE

  Release.

Typedefs

- typedef uint8_t AddressIEEE_t[8]

  IEEE address.
- typedef AddressIEEE_t AddressFilter_t[EASYLINK_MAX_ADDR_SIZE]

  Addresses filter array.

6.1.1 Detailed Description

Library header.

Additional layer for EasyLink

Project sub1GHz_TX_Layer
Developed with embedXcode+: `https://embedXcode.weebly.com`

Author

Rei Vilo

Rei Vilo

Date

05 Nov 2019 11:00

Version

103

Copyright

See also

ReadMe.txt for references

6.1.2 Typedef Documentation

6.1.2.1 AddressFilter_t

```
typedef AddressIEEE_t AddressFilter_t[EASYLINK_MAX_ADDR_SIZE]
```

Addresses filter array.

EASYLINK_MAX_ADDR_SIZE set to 3

6.1.2.2 AddressIEEE_t

```
typedef uint8_t AddressIEEE_t[8]
```

IEEE address.

uint8_t[8]

Note

CC13xx is little endian

- (uint8_t[8])00.12.4B.00.0A.27.CD.6A
- but (uint64_t)6ACD270A004B1200

See also

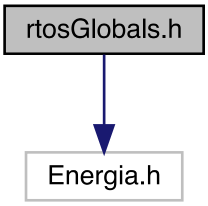
http://www.ti.com/lit/ug/swcu117h/swcu117h.pdf

6.2 rtosGlobals.h File Reference

Global variables for Energia MT project.

```
#include "Energia.h"
```

Include dependency graph for rtosGlobals.h:



6.2.1 Detailed Description

Global variables for Energia MT project.

<#details#>
Developed with embedXcode+: `https://embedXcode.weebly.com`

Author

Rei Vilo

Rei Vilo

Date

05 Nov 2019 10:56

Version

103

Copyright

(c) Rei Vilo, 2019

CC = BY SA NC

See also

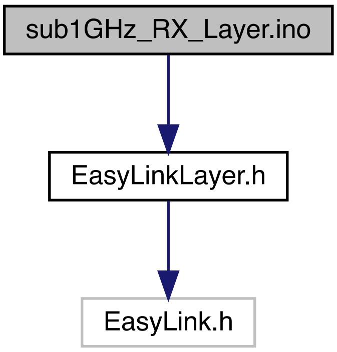
ReadMe.txt for references

6.3 sub1GHz_RX_Layer.ino File Reference

Main sketch.

```
#include "EasyLinkLayer.h"
```

Include dependency graph for sub1GHz_RX_Layer.ino:



Macros

- #define WITH_ADDRESS_FILTERING 1

  Address filtering.
- #define WITHOUT_ADDRESS_FILTERING 0

  desacticated
- #define ADDRESS_FILTERING WITH_ADDRESS_FILTERING

  mode

Functions

- void printAddress (AddressIEEE_t address, bool prefix=false)

  Print IEEE address.
- void printFilter (String title="Filter")

  Print the addresses of the filter.
- void **setup** ()
- void **loop** ()

Variables

- EasyLinkLayer **myLink** (true)
- AddressIEEE_t **addressHub** = { 0x00, 0x12, 0x4B, 0x00, 0x0B, 0xCA, 0x27, 0x82 }
- AddressIEEE_t **addressNode** = { 0x00, 0x12, 0x4B, 0x00, 0x0A, 0x27, 0xCD, 0x6A }
- AddressIEEE_t **addressLocal**
- uint16_t **value**

6.3.1 Detailed Description

Main sketch.

RX example for EasyLink additional layer
Developed with `embedXcode+`

Author

Rei Vilo

`https://embeddedcomputing.weebly.com`

Date

05 Nov 2019 10:56

Version

103

Copyright

(c) Rei Vilo, 2019

CC = BY SA NC

See also

ReadMe.txt for references

6.3.2 Macro Definition Documentation

6.3.2.1 WITH_ADDRESS_FILTERING

```
#define WITH_ADDRESS_FILTERING 1
```

Address filtering.

activated

6.3.3 Function Documentation

6.3.3.1 printAddress()

```
void printAddress (
             AddressIEEE_t address,
             bool prefix = false )
```

Print IEEE address.

Parameters

address	IEEE address
prefix	default=false, true=add 0x

6.3.3.2 printFilter()

```
void printFilter (
             String title = "Filter" )
```

Print the addresses of the filter.

Parameters

title	default="Filter"

Index