MIDIRouter Reference Manual

Generated by Doxygen 1.8.18

Sun Apr 26 2020 23:16:45

1 MIDIRouter Library 1

2 Hierarchical Index 3

2.1 Class Hierarchy . . . 3

3 Data Structure Index 5

3.1 Data Structures . . . 5

4 File Index 7

4.1 File List . . . 7

5 Data Structure Documentation 9

5.1 MIDIRouter::ByteBitLocation Struct Reference . . . 9

5.1.1 Detailed Description . . . 9

5.2 MIDIRouter::MidiFilter Class Reference . . . 9

5.2.1 Detailed Description . . . 10

5.2.2 Member Function Documentation . . . 10

5.2.2.1 controllersFiltered() . . . 10

5.2.2.2 midiclockFiltered() . . . 10

5.2.2.3 notesFiltered() . . . 11

5.2.2.4 realtimeFiltered() . . . 11

5.2.2.5 setControllers() . . . 11

5.2.2.6 setMidiclock() . . . 12

5.2.2.7 setNotes() . . . 12

5.2.2.8 setRealtime() . . . 12

5.2.2.9 setSysex() . . . 13

5.2.2.10 sysexFiltered() . . . 13

5.3 MIDIRouter::MidiFilterType Class Reference . . . 13

5.3.1 Detailed Description . . . 14

5.3.2 Member Function Documentation . . . 14

5.3.2.1 get() . . . 14

5.3.2.2 set() . . . 14

5.4 MIDIRouter::MIDIRouter_Lib Class Reference . . . 15

5.4.1 Detailed Description . . . 15

5.4.2 Member Function Documentation . . . 15

5.4.2.1 encoder() . . . 16

5.4.2.2 encPush() . . . 16

5.4.2.3 inputAt() . . . 16

5.4.2.4 outputAt() . . . 16

5.4.2.5 SetupEncoder() . . . 17

5.5 MIDIRouter::MIDIRouterSetup Struct Reference . . . 17

5.5.1 Detailed Description . . . 18

5.6 MIDIRouter::MRInputPort Class Reference . . . 18

5.6.1 Detailed Description . . . 19

5.6.2 Constructor & Destructor Documentation 19

5.6.2.1 MRInputPort() 19

5.7 MIDIRouter::MRIO_MidiHardwareSerialInterface< SerialPort > Class Template Reference 19

5.7.1 Detailed Description 20

5.7.2 Constructor & Destructor Documentation 21

5.7.2.1 MRIO_MidiHardwareSerialInterface() 21

5.8 MIDIRouter::MRIO_MidiUSBClientInterface Class Reference 21

5.8.1 Detailed Description 22

5.9 MIDIRouter::MRIO_MidiUSBHostInterface Class Reference 22

5.9.1 Detailed Description 23

5.10 MIDIRouter::MRIOInterface Class Reference 23

5.10.1 Detailed Description 24

5.10.2 Constructor & Destructor Documentation 24

5.10.2.1 MRIOInterface() 24

5.11 MIDIRouter::MROutputPort Class Reference 24

5.11.1 Detailed Description 25

5.11.2 Constructor & Destructor Documentation 25

5.11.2.1 MROutputPort() 26

5.12 MIDIRouter::MRPort Class Reference 26

5.12.1 Detailed Description 27

5.12.2 Constructor & Destructor Documentation 27

5.12.2.1 MRPort() 27

5.12.3 Member Function Documentation 27

5.12.3.1 active() 27

5.12.4 Field Documentation 28

5.12.4.1 location 28

5.13 RGBColor Class Reference 28

5.13.1 Detailed Description 28

5.13.2 Constructor & Destructor Documentation 28

5.13.2.1 RGBColor() 28

5.13.3 Member Function Documentation 29

5.13.3.1 asUint16() 29

6 File Documentation 31

6.1 MIDIRouter.ino File Reference 31

6.1.1 Detailed Description 40

6.1.2 Function Documentation 41

6.1.2.1 bmpDraw() 41

6.1.2.2 color565() 41

6.1.2.3 csvReadDouble() 41

6.1.2.4 csvReadFloat() 42

6.1.2.5 csvReadInt16() 42

6.1.2.6 csvReadInt32() 43
6.1.2.7 csvReadText() 43
6.1.2.8 csvReadUint16() 44
6.1.2.9 csvReadUint32() 44
6.1.2.10 CVnoteCal() 44
6.1.2.11 CVparamCal() 45
6.1.2.12 dPrint() 45
6.1.2.13 drawHomeScreen() 45
6.1.2.14 drawPiano() 46
6.1.2.15 dWrite() 46
6.1.2.16 filtRoute() 46
6.1.2.17 flashIn() 47
6.1.2.18 getTouchCol() 47
6.1.2.19 getTouchRow() 47
6.1.2.20 hub1() 48
6.1.2.21 hub2() 48
6.1.2.22 hub3() 48
6.1.2.23 hub4() 48
6.1.2.24 knob_calCV() 49
6.1.2.25 knobFull() 49
6.1.2.26 knobSet() 49
6.1.2.27 matchSysExID() 49
6.1.2.28 midi01() 49
6.1.2.29 midi02() 50
6.1.2.30 midi03() 50
6.1.2.31 midi04() 50
6.1.2.32 midi05() 50
6.1.2.33 midi06() 51
6.1.2.34 midi07() 51
6.1.2.35 midi08() 51
6.1.2.36 midi09() 51
6.1.2.37 midi10() 52
6.1.2.38 newColor() 52
6.1.2.39 read16() 53
6.1.2.40 read32() 53
6.1.2.41 reOrderR() 54
6.1.2.42 setDAC() 54
6.1.2.43 transmitMIDI() 54
6.1.2.44 transmitSysEx() 55
6.1.2.45 withinBox() 55
6.1.3 Variable Documentation 55
6.1.3.1 midilist 56

6.1.3.2 myusb 56
6.1.3.3 routing 56
6.1.3.4 SD 56
6.1.3.5 SysCsvFile 56
6.2 MIDIRouter_Library.h File Reference 57
6.2.1 Detailed Description 58

Index **59**

Chapter 1

MIDIRouter Library

MIDI Router based on Teensy 3.6
Library

Developed with embedXcode+: https://embedXcode.weebly.com

Author

Kurt Arnlund

Timeline85 / Ingenious Arts and Technologies LLC

Date

4/25/20 6:04 PM

Version

0.0.1

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:

- MIDIRouter::ByteBitLocation 9
- MIDIRouter::MidiFilter 9
- MIDIRouter::MidiFilterType 13
- MIDIRouter::MIDIRouter_Lib 15
- MIDIRouter::MIDIRouterSetup 17
- MIDIRouter::MRIOInterface 23
  - MIDIRouter::MRIO_MidiHardwareSerialInterface< SerialPort > 19
  - MIDIRouter::MRIO_MidiUSBClientInterface 21
  - MIDIRouter::MRIO_MidiUSBHostInterface 22
- MIDIRouter::MRPort 26
  - MIDIRouter::MRInputPort 18
  - MIDIRouter::MROutputPort 24
- RGBColor 28

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

MIDIRouter::ByteBitLocation
Structure that describes a specific byte and bit location 9

MIDIRouter::MidiFilter
This object contains common midi message filter types and methods to set and get them on port locations 9

MIDIRouter::MidiFilterType
This object is intended to wrap bit flags for a single bit-based feature flag 13

MIDIRouter::MIDIRouter_Lib
MIDI Router Library object 15

MIDIRouter::MIDIRouterSetup
Structure that describes the pins used for a midi router setup 17

MIDIRouter::MRInputPort
MIDI Router Input Port 18

MIDIRouter::MRIO_MidiHardwareSerialInterface< SerialPort >
MIDI Router Input/Output Interface For Serial Connected MIDI Ports 19

MIDIRouter::MRIO_MidiUSBClientInterface
DescriptionMIDI Router Input/Output Interface For USB Connected MIDI Ports 21

MIDIRouter::MRIO_MidiUSBHostInterface
DescriptionMIDI Router Input/Output Interface For USB Connected Host Interface MIDI Ports . 22

MIDIRouter::MRIOInterface
This object ties and input and an ouput together 23

MIDIRouter::MROutputPort
DescriptionMIDI Router Output Port 24

MIDIRouter::MRPort
MIDI Router Port Base Class 26

RGBColor
RGBColor object so that color descriptions can be allocated only once 28

Chapter 4

File Index

4.1 File List

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

MIDIRouter.ino
Main sketch Project MIDIRouter Library 31
MIDIRouter_Library.h
Library header Project MIDIRouter Library 57

Chapter 5

Data Structure Documentation

5.1 MIDIRouter::ByteBitLocation Struct Reference

Structure that describes a specific byte and bit location.

```
#include <MidiFilter.h>
```

Data Fields

- unsigned int byte

    byte location
- unsigned int bit

    bit location

5.1.1 Detailed Description

Structure that describes a specific byte and bit location.

The documentation for this struct was generated from the following file:

- MidiFilter.h

5.2 MIDIRouter::MidiFilter Class Reference

This object contains common midi message filter types and methods to set and get them on port locations.

```
#include <MidiFilter.h>
```

Public Member Functions

- void setRealtime (ByteBitLocation location, bool state)
- void setNotes (ByteBitLocation location, bool state)
- void setControllers (ByteBitLocation location, bool state)
- void setSysex (ByteBitLocation location, bool state)
- void setMidiclock (ByteBitLocation location, bool state)
- bool realtimeFiltered (ByteBitLocation location)
- bool notesFiltered (ByteBitLocation location)
- bool controllersFiltered (ByteBitLocation location)
- bool sysexFiltered (ByteBitLocation location)
- bool midiclockFiltered (ByteBitLocation location)

5.2.1 Detailed Description

This object contains common midi message filter types and methods to set and get them on port locations.

MIDI Filter object

5.2.2 Member Function Documentation

5.2.2.1 controllersFiltered()

```
bool MidiFilter::controllersFiltered (
             ByteBitLocation location )
```

Get controllers filter state

Parameters

location	location of feature flag

Returns

boolean true/false status, true = filtered

5.2.2.2 midiclockFiltered()

```
bool MidiFilter::midiclockFiltered (
             ByteBitLocation location )
```

Get midi clock filter state

Parameters

location	location of feature flag

Returns

boolean true/false status, true = filtered

5.2.2.3 notesFiltered()

```
bool MidiFilter::notesFiltered (
            ByteBitLocation location )
```

Get notes filter state

Parameters

location	location of feature flag

Returns

boolean true/false status, true = filtered

5.2.2.4 realtimeFiltered()

```
bool MidiFilter::realtimeFiltered (
            ByteBitLocation location )
```

Get system realtime filter state

Parameters

location	location of feature flag

Returns

boolean true/false status, true = filtered

5.2.2.5 setControllers()

```
void MidiFilter::setControllers (
            ByteBitLocation location,
            bool state )
```

Set controllers filter state

Parameters

location	port location
state	feature state

5.2.2.6 setMidiclock()

```
void MidiFilter::setMidiclock (
            ByteBitLocation location,
            bool state )
```

Set midi clock filter state

Parameters

location	port location
state	feature state

5.2.2.7 setNotes()

```
void MidiFilter::setNotes (
            ByteBitLocation location,
            bool state )
```

Set note filter state

Parameters

location	port location
state	feature state

5.2.2.8 setRealtime()

```
void MidiFilter::setRealtime (
            ByteBitLocation location,
            bool state )
```

Set system realtime filter state

Parameters

location	port location
state	feature state

5.2.2.9 setSysex()

```
void MidiFilter::setSysex (
            ByteBitLocation location,
            bool state )
```

Set sysex filter state

Parameters

location	port location
state	feature state

5.2.2.10 sysexFiltered()

```
bool MidiFilter::sysexFiltered (
            ByteBitLocation location )
```

Get sysex filter state

Parameters

location	location of feature flag

Returns

boolean true/false status, true = filtered

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

- MidiFilter.h
- MidiFilter.hpp

5.3 MIDIRouter::MidiFilterType Class Reference

this object is intended to wrap bit flags for a single bit-based feature flag

```
#include <MidiFilter.h>
```

Public Member Functions

- void set (ByteBitLocation location, bool state)
- bool get (ByteBitLocation location)

5.3.1 Detailed Description

this object is intended to wrap bit flags for a single bit-based feature flag

MIDI filter object

5.3.2 Member Function Documentation

5.3.2.1 get()

```
bool MidiFilterType::get (
            ByteBitLocation location )
```

Get feature flag state

Parameters

location	location of feature flag

Returns

boolean true/false status, true = filtered

5.3.2.2 set()

```
void MidiFilterType::set (
            ByteBitLocation location,
            bool state )
```

Set feature flag to a state

Parameters

location	location of feature flag
state	boolean state to set it to

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

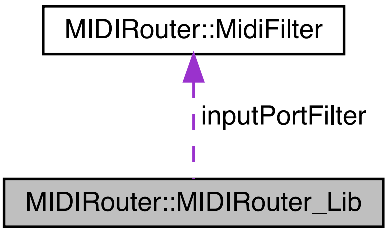
- MidiFilter.h
- MidiFilter.hpp

5.4 MIDIRouter::MIDIRouter_Lib Class Reference

MIDI Router Library object.

```
#include <MIDIRouter_Library.h>
```

Collaboration diagram for MIDIRouter::MIDIRouter_Lib:



Public Member Functions

- MRInputPort ∗ inputAt (int index)
- MROutputPort ∗ outputAt (int index)
- void SetupEncoder (uint8_t encpin1, uint8_t encpin2, uint8_t enc_button_pin)

  [KURT] this may possibly be changed in the future to us a MIDIRouterSetup structure for input
- Encoder & encoder ()
- Bounce & encPush ()

Static Public Attributes

- static MidiFilter inputPortFilter = MidiFilter()

  Input port filter object.

5.4.1 Detailed Description

MIDI Router Library object.

5.4.2 Member Function Documentation

5.4.2.1 encoder()

```
Encoder & MIDIRouter_Lib::encoder ( )
```

Encoder object

Returns

Encoder object reference

5.4.2.2 encPush()

```
Bounce & MIDIRouter_Lib::encPush ( )
```

Encoder push button object

Returns

Bounce button debouncing object reference

5.4.2.3 inputAt()

```
MRInputPort * MIDIRouter_Lib::inputAt (
             int index )
```

Locate a midi input port

Parameters

index	index of port

Returns

pointer to MRInputPort object

5.4.2.4 outputAt()

```
MROutputPort * MIDIRouter_Lib::outputAt (
             int index )
```

Locate a midi output port

Parameters

index	index of port

Returns

pointer to MROutputPort object

5.4.2.5 SetupEncoder()

```
void MIDIRouter_Lib::SetupEncoder (
             uint8_t encpin1,
             uint8_t encpin2,
             uint8_t enc_button_pin )
```

[KURT] this may possibly be changed in the future to us a MIDIRouterSetup structure for input

Setup an encoder

Parameters

encpin1	encoder pin 1 input
encpin2	encoder pin 2 input
enc_button_pin	encoder push button pin input

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

- MIDIRouter_Library.h
- MIDIRouter_Library.hpp

5.5 MIDIRouter::MIDIRouterSetup Struct Reference

Structure that describes the pins used for a midi router setup.

```
#include <MIDIRouter_Library.h>
```

Data Fields

- uint8_t encpin1

  Encoder pin 1.
- uint8_t encpin2

  Encoder pin 2.
- uint8_t enc_button_pin

  Encoder button pin.

5.5.1 Detailed Description

Structure that describes the pins used for a midi router setup.

The documentation for this struct was generated from the following file:

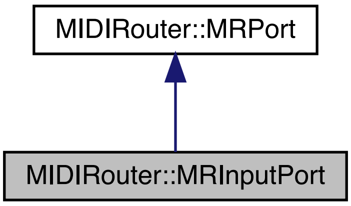
- MIDIRouter_Library.h

5.6 MIDIRouter::MRInputPort Class Reference

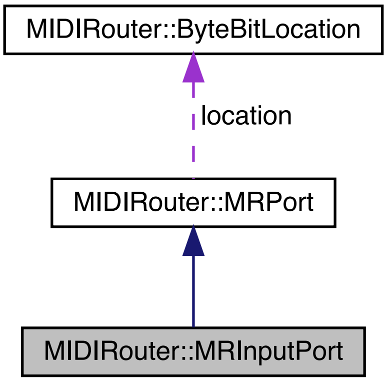
MIDI Router Input Port.

```
#include <MRPorts.h>
```

Inheritance diagram for MIDIRouter::MRInputPort:



Collaboration diagram for MIDIRouter::MRInputPort:



Public Member Functions

- MRInputPort (const char ∗displayName)

Additional Inherited Members

5.6.1 Detailed Description

MIDI Router Input Port.

5.6.2 Constructor & Destructor Documentation

5.6.2.1 MRInputPort()

```
MRInputPort::MRInputPort (
            const char * displayName )
```

Constructor

Parameters

displayName	the name to display for the port

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

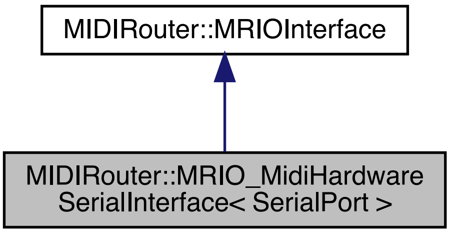
- MRPorts.h
- MRPorts.hpp

5.7 MIDIRouter::MRIO_MidiHardwareSerialInterface< SerialPort > Class Template Reference

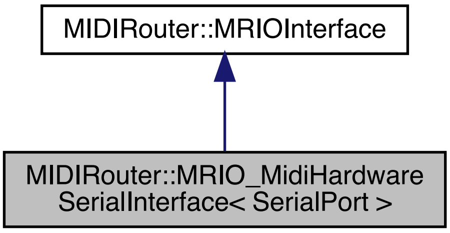
MIDI Router Input/Output Interface For Serial Connected MIDI Ports.

```
#include <MRIOInterface.h>
```

Inheritance diagram for MIDIRouter::MRIO_MidiHardwareSerialInterface< SerialPort >:


MIDIRouter::MRIOInterface
MIDIRouter::MRIO_MidiHardware SerialInterface< SerialPort >

Collaboration diagram for MIDIRouter::MRIO_MidiHardwareSerialInterface< SerialPort >:


MIDIRouter::MRIOInterface
MIDIRouter::MRIO_MidiHardware SerialInterface< SerialPort >

Public Member Functions

- MRIO_MidiHardwareSerialInterface (unsigned char inPort, unsigned char outPort, SerialPort &inSerial)

Data Fields

- midi::MidiInterface< HardwareSerial > interface

  The Hardware Serial MIDI Interface.

5.7.1 Detailed Description

template<class SerialPort>
class MIDIRouter::MRIO_MidiHardwareSerialInterface< SerialPort >

MIDI Router Input/Output Interface For Serial Connected MIDI Ports.

5.7.2 Constructor & Destructor Documentation

5.7.2.1 MRIO_MidiHardwareSerialInterface()

```
template<class SerialPort >
MRIO_MidiHardwareSerialInterface::MRIO_MidiHardwareSerialInterface (
             unsigned char inPort,
             unsigned char outPort,
             SerialPort & inSerial )
```

Constructor

Parameters

inPort	input port index
outPort	output port index
inSerial	serial port to create an interface for

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

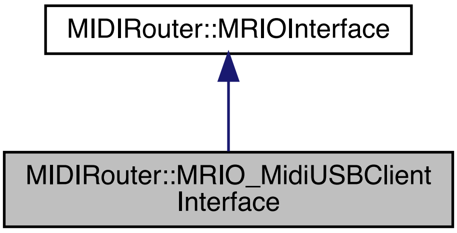
- MRIOInterface.h
- MRIOInterface.hpp

5.8 MIDIRouter::MRIO_MidiUSBClientInterface Class Reference

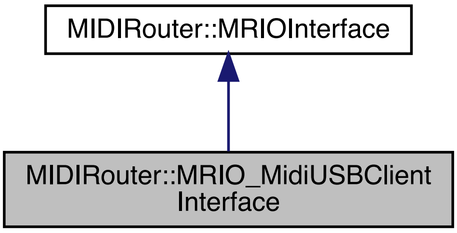
DescriptionMIDI Router Input/Output Interface For USB Connected MIDI Ports.

```
#include <MRIOInterface.h>
```

Inheritance diagram for MIDIRouter::MRIO_MidiUSBClientInterface:


MIDIRouter::MRIOInterface
MIDIRouter::MRIO_MidiUSBClient Interface

Collaboration diagram for MIDIRouter::MRIO_MidiUSBClientInterface:


MIDIRouter::MRIOInterface
MIDIRouter::MRIO_MidiUSBClient Interface

Additional Inherited Members

5.8.1 Detailed Description

DescriptionMIDI Router Input/Output Interface For USB Connected MIDI Ports.

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

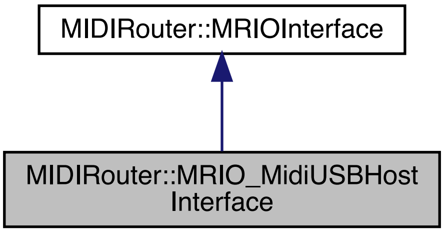
- MRIOInterface.h
- MRIOInterface.hpp

5.9 MIDIRouter::MRIO_MidiUSBHostInterface Class Reference

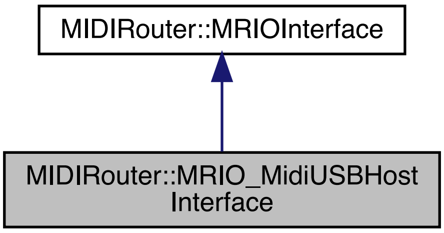
DescriptionMIDI Router Input/Output Interface For USB Connected Host Interface MIDI Ports.

```
#include <MRIOInterface.h>
```

Inheritance diagram for MIDIRouter::MRIO_MidiUSBHostInterface:


MIDIRouter::MRIOInterface
MIDIRouter::MRIO_MidiUSBHost Interface

Collaboration diagram for MIDIRouter::MRIO_MidiUSBHostInterface:


MIDIRouter::MRIOInterface
MIDIRouter::MRIO_MidiUSBHost Interface

Additional Inherited Members

5.9.1 Detailed Description

DescriptionMIDI Router Input/Output Interface For USB Connected Host Interface MIDI Ports.

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

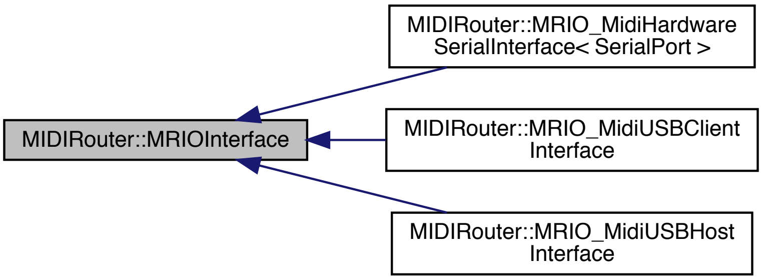
- MRIOInterface.h
- MRIOInterface.hpp

5.10 MIDIRouter::MRIOInterface Class Reference

This object ties and input and an ouput together.

```
#include <MRIOInterface.h>
```

Inheritance diagram for MIDIRouter::MRIOInterface:


MIDIRouter::MRIO_MidiHardware SerialInterface< SerialPort >
MIDIRouter::MRIOInterface
MIDIRouter::MRIO_MidiUSBClient Interface
MIDIRouter::MRIO_MidiUSBHost Interface

Public Member Functions

- MRIOInterface (unsigned char inPort, unsigned char outPort)

Data Fields

- unsigned char input

    Index if the input port.
- unsigned char output

    Index if the output port.

5.10.1 Detailed Description

This object ties and input and an ouput together.

MIDI Router Input/Output Interface Base Class

5.10.2 Constructor & Destructor Documentation

5.10.2.1 MRIOInterface()

```
BEGIN_MIDIROUTER_NAMESPACE MRIOInterface::MRIOInterface (
             unsigned char inPort,
             unsigned char outPort )
```

Constructor

Parameters

inPort	index of the input port
outPort	index of the output port

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

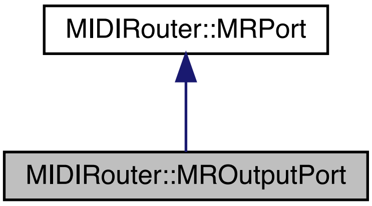
- MRIOInterface.h
- MRIOInterface.hpp

5.11 MIDIRouter::MROutputPort Class Reference

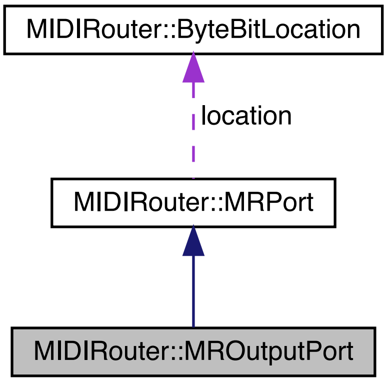
DescriptionMIDI Router Output Port.

```
#include <MRPorts.h>
```

Inheritance diagram for MIDIRouter::MROutputPort:



Collaboration diagram for MIDIRouter::MROutputPort:



Public Member Functions

- MROutputPort (const char ∗displayName)

Additional Inherited Members

5.11.1 Detailed Description

DescriptionMIDI Router Output Port.

5.11.2 Constructor & Destructor Documentation

5.11.2.1 MROutputPort()

```
MROutputPort::MROutputPort (
            const char * displayName )
```

Constructor

Parameters

displayName	the name to display for the port

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

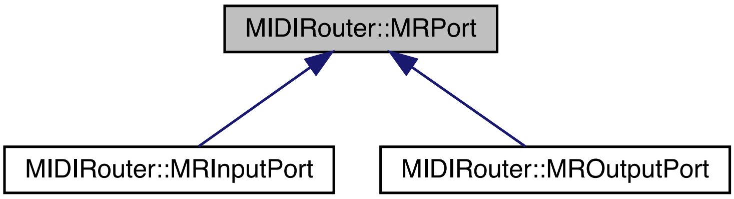
- MRPorts.h
- MRPorts.hpp

5.12 MIDIRouter::MRPort Class Reference

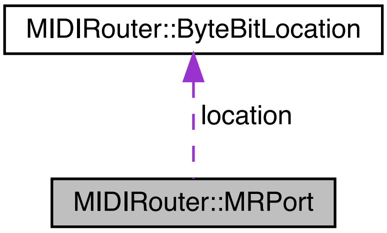
MIDI Router Port Base Class.

`#include <MRPorts.h>`

Inheritance diagram for MIDIRouter::MRPort:



Collaboration diagram for MIDIRouter::MRPort:



Public Member Functions

- MRPort (const char ∗displayName, unsigned int newindex)
- bool active ()

Data Fields

- char name [9]

    port name storage
- unsigned int index

    port index
- ByteBitLocation location

    This can be used to access port feature flag bits.

5.12.1 Detailed Description

MIDI Router Port Base Class.

5.12.2 Constructor & Destructor Documentation

5.12.2.1 MRPort()

```
BEGIN_MIDIROUTER_NAMESPACE MRPort::MRPort (
                const char * displayName,
                unsigned int newindex )
```

Port Constructor

[KURT] add details explainging the newIndex

Parameters

displayName	the name to display for the port
newindex	index of the port

5.12.3 Member Function Documentation

5.12.3.1 active()

```
bool MRPort::active ( )
```

Is the port active

Returns

boolean true/false is the port active. Inactive ports as used as filler.

5.12.4 Field Documentation

5.12.4.1 location

```
ByteBitLocation MIDIRouter::MRPort::location
```

This can be used to access port feature flag bits.

byte and bit location for this port

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

- MRPorts.h
- MRPorts.hpp

5.13 RGBColor Class Reference

RGBColor object so that color descriptions can be allocated only once.

```
#include <ColorCalc.h>
```

Public Member Functions

- RGBColor (uint8_t r, uint8_t g, uint8_t b)
- uint16_t asUint16 ()

5.13.1 Detailed Description

RGBColor object so that color descriptions can be allocated only once.

5.13.2 Constructor & Destructor Documentation

5.13.2.1 RGBColor()

```
RGBColor::RGBColor (
            uint8_t r,
            uint8_t g,
            uint8_t b )  [inline]
```

Constructor

Parameters

r	red 0-255
g	green 0-255
b	blue 0-255

5.13.3 Member Function Documentation

5.13.3.1 asUint16()

```
uint16_t RGBColor::asUint16 ( )  [inline]
```

Returns

16 bit color

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

- ColorCalc.h

Chapter 6

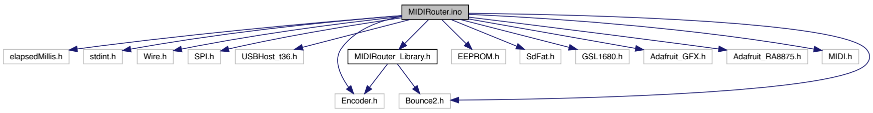
File Documentation

6.1 MIDIRouter.ino File Reference

Main sketch Project MIDIRouter Library.

```
#include <elapsedMillis.h>
#include <stdint.h>
#include <Wire.h>
#include <SPI.h>
#include <USBHost_t36.h>
#include <Encoder.h>
#include <Bounce2.h>
#include <EEPROM.h>
#include "SdFat.h"
#include "GSL1680.h"
#include "Adafruit_GFX.h"
#include "Adafruit_RA8875.h"
#include "MIDI.h"
#include "MIDIRouter_Library.h"
#include "ColorCalc.h"
#include "MR_TOUCH.h"
#include "MR_DRAW.h"
#include "MR_MIDI.h"
#include "MR_UTIL.h"
#include "MR_DAC.h"
#include "MR_TXT.h"
#include "MR_EEPROM.h"
```

Include dependency graph for MIDIRouter.ino:



Macros

- #define MIDI_SERIAL_SUPPORT

  MIDI_SERIAL_SUPPORT = add hardware serial support for midi ports.
- #define MIDI_USB_SUPPORT

  MIDI_USB_SUPPORT = add usb support for midi ports.
- #define SDCARD_SUPPORT

  SDCARD_SUPPORT = add SD card support.
- #define TFT_DISPLAY

  TFT_DISPLAY = add TFT display support.
- #define STARTUP_PICTURE

  STARTUP_PICTURE = show the startup picture loaded from the sd card.
- #define CSV_DELIM ','

  CSV_DELIM CSV file field delimiter.
- #define EncA 26

  Encoder pin 1 input.
- #define EncB 27

  Encoder pin 2 input.
- #define EncSwitch 28

  Encoder push button pin input.
- #define dacA B00010100

  DAC 1 address.
- #define dacB B00010010

  DAC 2 address.
- #define dacC B00010110

  DAC 3 address.
- #define dacD B00010000

  DAC 4 address.
- #define dALL B00110100

  DAC ALL address.
- #define CS 43

  SPI chip select pin.
- #define dac5 A22

  DAC 5 pin.
- #define dac6 A21

  DAC 6 pin.
- #define dig1 3

  DOUT 1 pin.
- #define dig2 4

  DOUT 2 pin.
- #define dig3 5

  DOUT 3 pin.
- #define dig4 6

  DOUT 4 pin.
- #define dig5 22

  DOUT 5 pin.
- #define dig6 21

  DOUT 6 pin.
- #define adc1 A9

  Analog to Digital Converter 1 pin.
- #define adc2 A6

Analog to Digital Converter 2 pin.

- #define INTERVALMIDI 250

  how often (in microseconds) we call routeMidi()
- #define RA8875_INT 15

  graphic interrupt
- #define RA8875_CS 14

  chip select
- #define RA8875_RESET 35

  reset
- #define WAKE 16

  wakeup! (is this used?)
- #define INTRPT 17

  touch interrupt
- #define SPEED 4

  [Eric] -document me

Functions

- MIDI_CREATE_INSTANCE (HardwareSerial, Serial1, MIDI1)

  Create MIDI 1 interace instance.
- MIDI_CREATE_INSTANCE (HardwareSerial, Serial2, MIDI2)

  Create MIDI 2 interace instance.
- MIDI_CREATE_INSTANCE (HardwareSerial, Serial3, MIDI3)

  Create MIDI 3 interace instance.
- MIDI_CREATE_INSTANCE (HardwareSerial, Serial4, MIDI4)

  Create MIDI 4 interace instance.
- MIDI_CREATE_INSTANCE (HardwareSerial, Serial5, MIDI5)

  Create MIDI 5 interace instance.
- MIDI_CREATE_INSTANCE (HardwareSerial, Serial6, MIDI6)

  Create MIDI 6 interace instance.
- USBHub hub1 (myusb)
- USBHub hub2 (myusb)
- USBHub hub3 (myusb)
- USBHub hub4 (myusb)
- MIDIDevice midi01 (myusb)
- MIDIDevice midi02 (myusb)
- MIDIDevice midi03 (myusb)
- MIDIDevice midi04 (myusb)
- MIDIDevice midi05 (myusb)
- MIDIDevice midi06 (myusb)
- MIDIDevice midi07 (myusb)
- MIDIDevice midi08 (myusb)
- MIDIDevice midi09 (myusb)
- MIDIDevice midi10 (myusb)
- uint16_t newColor (uint8_t r, uint8_t g, uint8_t b)
- void saveEEPROM ()

  save to eeprom
- void loadEEPROM ()

  load from eeprom
- void setDAC (int dac, uint32_t data)
- void touchIO ()

perform touch i/o
- void drawTouchPos ()

  [Eric] -document me
- void evaltouch ()

  [Eric] -document me
- void drawMenu_Routing ()

  [Eric] -document me
- void refMenu_Routing ()

  [Eric] -document me
- void refMenu_Calibrate ()

  [Eric] -document me
- void drawMenu_Calibrate ()

  [Eric] -document me
- void drawMenu_Calibrate_udcv ()

  [Eric] -document me
- void readKnob ()

  [Eric] -document me
- void knobZero ()

  [Eric] -document me
- void knobFull ()
- void knobSet (int v)
- void knob_calCV ()
- boolean withinBox (int x, int y, int bx, int by, int bw, int bh)
- int getTouchCol (long x)
- int getTouchRow (long y)
- void drawBox ()

  [Eric] -document me
- void drawColumns ()

  [Eric] -document me
- void drawRows ()

  [Eric] -document me
- void drawRouting ()

  [Eric] -document me
- void drawGLines ()

  [Eric] -document me
- void drawBGs ()

  [Eric] -document me
- void drawHomeScreen ()
- void drawPiano (int c, int r)
- void flashIn (int inp, int state)
- void bmpDraw (const char ∗filename, int x, int y)
- uint16_t read16 (File &f)
- uint32_t read32 (File &f)
- uint16_t color565 (uint8_t r, uint8_t g, uint8_t b)
- void routeMidi ()

  [Eric] -document me
- void transmitMIDI (int t, int d1, int d2, int ch, byte inPort)
- void transmitSysEx (unsigned int len, const uint8_t ∗sysexarray, byte inPort)
- float CVnoteCal (int note, int dac)
- float CVparamCal (int data, int dac)
- void showADC ()

  [Eric] -document me

- void profileInstruments ()

  [Eric] -document me
- bool filtRoute (int t, int f)
- void matchSysExID (int16_t b1, int16_t b2, int16_t b3)
- void printMatch ()

  [Eric] -document me
- int csvReadText (File ∗SysCsvFile, char ∗str, size_t size, char delim)
- int csvReadInt32 (File ∗SysCsvFile, int32_t ∗num, char delim)
- int csvReadInt16 (File ∗SysCsvFile, int16_t ∗num, char delim)
- int csvReadUint32 (File ∗SysCsvFile, uint32_t ∗num, char delim)
- int csvReadUint16 (File ∗SysCsvFile, uint16_t ∗num, char delim)
- int csvReadDouble (File ∗SysCsvFile, double ∗num, char delim)
- int csvReadFloat (File ∗SysCsvFile, float ∗num, char delim)
- void csvClose ()

  Close CSV File.
- int reOrderR (int r)
- void dPrint (String s, int sz=fSize)
- void dWrite (unsigned char c, unsigned int s)
- void setup ()

  ! Add setup code
- void loop ()

  ! Add loop code

Variables

- SdFatSdioEX SD

  [KURT] this will be moving into the MIDI Router library
- File SysCsvFile

  [KURT] this will be moving into the MIDI Router library
- USBHost myusb
- MIDIDevice ∗ midilist [10]

  a list of usb midi devices for easy array access
- IntervalTimer callMIDI

  Interval timer for midi io scheduling.
- Adafruit_RA8875 tft = Adafruit_RA8875(14 , 35)

  TFT Display Driver Instance.
- GSL1680 TS = GSL1680()

  Touch screen driver.
- boolean clockPulse = 0

  clock pulse boolean
- int startCount = 0

  clock pulse start count
- int led = 13

  LED pin.
- int backLight = 255

  backlight intensity
- long oldPosition = 0

  encoder old position
- long newPosition = 0

  encoder new position
- int knobVal = 0

encoder knob value

- int oldKnobVal = 0

  encoder old knob value
- bool knobDir = 0

  encoder knob direction 0 = CCW, 1 = CW
- bool knobAccelEnable = 0

  encoder knob acceleration
- unsigned long knobTimer = millis()

  encoder knob timer
- unsigned long knobSlowdown = 2

  wait this many ms before checking the knob value
- int knobSpeedup = 3

  threshold for difference between old and new value to cause a speed up
- float knobSpeedRate = 2.8

  factor (exponent) to speed up by
- int knobMin = 0

  encoder knob min value
- int knobMax = 8

  encoder knob max value
- float cvee = 0

  [Eric] - document me
- long cveeKnobOffset = 0

  [Eric] - document me
- long dacNeg [6]

  [Eric] - document me
- long dacPos [6]

  [Eric] - document me
- long dacOffset [120]

  [Eric] - document me
- int eeprom_addr_offset = 0

  Create address offset so Array2 is located after dacOffset in EEPROM.
- uint16_t tbColor = RGBColor(0, 150, 0).asUint16()

  tempo/clock box
- uint16_t hbColor = RGBColor(102, 102, 102).asUint16()

  setup/home button
- uint16_t ibColor = RGBColor(0, 0, 150).asUint16()

  input page box
- uint16_t insColor = RGBColor(0, 0, 100).asUint16()

  inputs box color
- uint16_t insColFlash = RGBColor(0, 0, 255).asUint16()

  input flash color
- uint16_t obColor = RGBColor(150, 0, 0).asUint16()

  output page box
- uint16_t outsColor = RGBColor(100, 0, 0).asUint16()

  putputs box color
- uint16_t outColFlash = RGBColor(255, 0, 0).asUint16()

  output flash color
- uint16_t gridColor = RGBColor(102, 102, 102).asUint16()

  grid
- uint16_t linClr = RGBColor(0, 0, 0).asUint16()

  lines

- uint16_t txColor = RGBColor(255, 255, 255).asUint16()

  text
- uint16_t routColor = RGBColor(255, 255, 255).asUint16()

  routing
- uint16_t actFieldBg = RGBColor(0, 0, 255).asUint16()

  Active Field color.
- uint16_t fieldBg = RGBColor(50, 50, 50).asUint16()

  inactive field color
- uint16_t posCol

  for CV calib
- uint16_t negCol

  for CV calib
- long fingers = 0

  [Eric] -document me
- long curFing = 0

  [Eric] -document me
- long x = 0

  [Eric] -document me
- long y = 0

  [Eric] -document me
- int WIDE = 799

  [Eric] -document me
- int TALL = 479

  [Eric] -document me
- int curRot = 2

  [Eric] -document me
- int rows = 6

  [Eric] -document me
- int columns = 6

  [Eric] -document me
- uint8_t sysexIDReq [ ] = {240, 126, 127, 6, 1, 247}

  [Eric] -document me
- int menu = 0

  which menu are we looking at? 0 = routing, 1 = CV calibration
- int actField = 1

  which data entry field on the page is active?
- boolean rdFlag = 0

  flag to redraw screen
- int inPages = 6

  [Eric] -document me
- int outPages = 7

  [Eric] -document me
- int pgOut = 0

  [Eric] -document me
- int pgIn = 0

  [Eric] -document me
- elapsedMillis elapseIn

  [Eric] -document me
- elapsedMillis elapseIn1

  [Eric] -document me
- elapsedMillis elapseIn2

[Eric] -document me

- elapsedMillis elapseIn3

  [Eric] -document me
- elapsedMillis elapseIn4

  [Eric] -document me
- elapsedMillis elapseIn5

  [Eric] -document me
- elapsedMillis elapseIn6

  [Eric] -document me
- unsigned int flashTime = 1000

  delay in milliseconds between activity flashes
- int inFlag [5]

  [Eric] -document me
- uint16_t fColor = RA8875_WHITE

  [Eric] -document me
- uint16_t fBG = 0

  [Eric] -document me
- int fSize = 3

  [Eric] -document me
- int fWidth = 18

  [Eric] -document me
- int fHeight = 25

  [Eric] -document me
- uint16_t curX = 20

  [Eric] -document me
- uint16_t curY = 20

  [Eric] -document me
- int tBord = 5

  buffer/border from edge of screen to beginning of text
- int rOffset = 119

  was 152 - [Eric] -document me
- int rHeight = (TALL - rOffset) / rows

  60 - [Eric] -document me
- int tROffset = (rHeight/2)-(fHeight/2)

  text vertical offset in rows
- int cOffset = 199

  was 238 - [Eric] -document me
- int cWidth = (WIDE - cOffset) / columns

  100 - [Eric] -document me
- int tCOffset = (cWidth/2)-(fHeight/5)

  text horizontal offset in rows
- float tbWidth = cOffset/2

  [Eric] -document me
- float tbHeight = rOffset/2

  [Eric] -document me
- float tbOX = (cOffset - tbWidth)

  origin X
- float tbOY = (rOffset - tbHeight)

  origin Y
- int tbText = 60

  [Eric] -document me

- int tempo = 120

  [Eric] -document me
- float hbWidth = (cOffset - tbWidth)

  [Eric] -document me
- float hbHeight = (rOffset - tbHeight)

  [Eric] -document me
- int hbOX = 0

  origin X
- int hbOY = 0

  origin Y
- int menuCV_butDacNeg5_x = 150

  [Eric] -document me
- int menuCV_butDacNeg5_y = rOffset+100

  [Eric] -document me
- int menuCV_butDacNeg5_w = 125

  [Eric] -document me
- int menuCV_butDacNeg5_h = 50

  [Eric] -document me
- int menuCV_butDacPos5_x = 460

  [Eric] -document me
- int menuCV_butDacPos5_y = rOffset+100

  [Eric] -document me
- int menuCV_butDacPos5_w = 125

  [Eric] -document me
- int menuCV_butDacPos5_h = 50

  [Eric] -document me
- int CVcalSelect = 0

  [Eric] -document me
- char ystr [16]

  [Eric] -document me
- char xstr [16]

  [Eric] -document me
- long touchX = 0

  [Eric] -document me
- long touchY = 0

  [Eric] -document me
- long lastPress = 0

  [Eric] -document me
- long newX = 0

  [Eric] -document me
- long newY = 0

  [Eric] -document me
- int difX = 0

  [Eric] -document me
- int difY = 0

  [Eric] -document me
- unsigned long touchShort = 300

  (ms) must touch this long to trigger
- int tMargin = 5

  pixel margin to filter out duplicate triggers for a single touch
- float clearRouting = 0

[Eric] -document me

- float pi = 3.141592

  [Eric] -document me
- int curRoute = 0

  storage for current routing/filter value
- int curCol = 0

  [Eric] -document me
- int curRow = 0

  [Eric] -document me
- uint8_t routing [50][50]
- char syIdHex [20]

  [Eric] -document me
- char mfg [80]

  [Eric] -document me
- int16_t idLen

  [Eric] -document me
- int16_t idB1

  [Eric] -document me
- int16_t idB2

  [Eric] -document me
- int16_t idB3

  [Eric] -document me
- USING_NAMESPACE_MIDIROUTER MIDIRouter_Lib router = MIDIRouter_Lib()

  Create MIDI Router Object.

6.1.1 Detailed Description

Main sketch Project MIDIRouter Library.

Main MIDI Router INO implementation
Developed with `embedXcode+`

Author

Eric Bateman and Kurt Arnlund

Timeline85 / Ingenious Arts and Technologies LLC

Date

4/25/20 6:04 PM

Version

1.0.0

Copyright

(c) 2020 Eric Bateman and Kurt Arnlund

GNU General Public Licence

See also

ReadMe.txt for references

6.1.2 Function Documentation

6.1.2.1 bmpDraw()

```
void bmpDraw (
             const char * filename,
             int x,
             int y )
```

[Eric] -document me

Parameters

filename	[Eric] -document me
x	[Eric] -document me
y	[Eric] -document me

6.1.2.2 color565()

```
uint16_t color565 (
             uint8_t r,
             uint8_t g,
             uint8_t b )
```

create a 565 color

Parameters

r	red 0-255
g	green 0-255
b	blue 0-255

Returns

uint16_t 16 bit 565 color value

6.1.2.3 csvReadDouble()

```
int csvReadDouble (
             File * SysCsvFile,
             double * num,
             char delim )
```

[Eric] -document me

Parameters

SysCsvFile	[Eric] -document me
num	[Eric] -document me
delim	[Eric] -document me

Returns

int value

6.1.2.4 csvReadFloat()

```
int csvReadFloat (
              File * SysCsvFile,
              float * num,
              char delim )
```

[Eric] -document me

Parameters

SysCsvFile	[Eric] -document me
num	[Eric] -document me
delim	[Eric] -document me

Returns

int value

6.1.2.5 csvReadInt16()

```
int csvReadInt16 (
              File * SysCsvFile,
              int16_t * num,
              char delim )
```

[Eric] -document me

Parameters

SysCsvFile	[Eric] -document me
num	[Eric] -document me
delim	[Eric] -document me

Returns

int value

6.1.2.6 csvReadInt32()

```
int csvReadInt32 (
             File * SysCsvFile,
             int32_t * num,
             char delim )
```

[Eric] -document me

Parameters

SysCsvFile	[Eric] -document me
num	[Eric] -document me
delim	[Eric] -document me

Returns

int value

6.1.2.7 csvReadText()

```
int csvReadText (
             File * SysCsvFile,
             char * str,
             size_t size,
             char delim )
```

[Eric] -document me

Parameters

SysCsvFile	[Eric] -document me
str	[Eric] -document me
size	[Eric] -document me
delim	[Eric] -document me

Returns

int value

6.1.2.8 csvReadUint16()

```
int csvReadUint16 (
            File * SysCsvFile,
            uint16_t * num,
            char delim )
```

[Eric] -document me

Parameters

SysCsvFile	[Eric] -document me
num	[Eric] -document me
delim	[Eric] -document me

Returns

int value

6.1.2.9 csvReadUint32()

```
int csvReadUint32 (
            File * SysCsvFile,
            uint32_t * num,
            char delim )
```

[Eric] -document me

Parameters

SysCsvFile	[Eric] -document me
num	[Eric] -document me
delim	[Eric] -document me

Returns

int value

6.1.2.10 CVnoteCal()

```
float CVnoteCal (
            int note,
            int dac )
```

[Eric] -document me

Parameters

note	[Eric] -document me
dac	[Eric] -document me

Returns

float value

6.1.2.11 CVparamCal()

```
float CVparamCal (
             int data,
             int dac )
```

[Eric] -document me

Parameters

data	[Eric] -document me
dac	[Eric] -document me

Returns

float value

6.1.2.12 dPrint()

```
void dPrint (
             String s,
             int sz = fSize )
```

debug print

Parameters

s	[Eric] -document me
sz	[Eric] -document me

6.1.2.13 drawHomeScreen()

```
void drawHomeScreen ( )
```

[Eric] -document me

6.1.2.14 drawPiano()

```
void drawPiano (
             int c,
             int r )
```

[Eric] -document me

Parameters

c	[Eric] -document me
r	[Eric] -document me

6.1.2.15 dWrite()

```
void dWrite (
             unsigned char c,
             unsigned int s )
```

debug write

Parameters

c	[Eric] -document me
s	[Eric] -document me

6.1.2.16 filtRoute()

```
bool filtRoute (
             int t,
             int f )
```

[Eric] -document me

Parameters

t	[Eric] -document me
f	[Eric] -document me

Returns

bool true/false

6.1.2.17 flashIn()

```
void flashIn (
            int inp,
            int state )
```

[Eric] -document me

Parameters

inp	[Eric] -document me
state	[Eric] -document me

6.1.2.18 getTouchCol()

```
int getTouchCol (
            long x )
```

using location x determine what column it is in

Parameters

x	location x

Returns

column

6.1.2.19 getTouchRow()

```
int getTouchRow (
            long y )
```

using location x determine what row it is in

Parameters

y	location y

Returns

row

6.1.2.20 hub1()

```
USBHub hub1 (
            myusb  )
```

Returns

USBHub USB Hub 1

6.1.2.21 hub2()

```
USBHub hub2 (
            myusb  )
```

Returns

USBHub USB Hub 2

6.1.2.22 hub3()

```
USBHub hub3 (
            myusb  )
```

Returns

USBHub USB Hub 3

6.1.2.23 hub4()

```
USBHub hub4 (
            myusb  )
```

Returns

USBHub USB Hub 4

6.1.2.24 knob_calCV()

```
void knob_calCV ( )
```

[Eric] -document me

6.1.2.25 knobFull()

```
void knobFull ( )
```

[Eric] -document me

6.1.2.26 knobSet()

```
void knobSet (
             int v )
```

Parameters

v	[Eric] = comment on what V is

6.1.2.27 matchSysExID()

```
void matchSysExID (
             int16_t b1,
             int16_t b2,
             int16_t b3 )
```

matchSysExID

Parameters

b1	[Eric] -document me
b2	[Eric] -document me
b3	[Eric] -document me

6.1.2.28 midi01()

```
MIDIDevice midi01 (
             myusb )
```

Returns

MIDIDevice USB MIDI 1

6.1.2.29 midi02()

```
MIDIDevice midi02 (
             myusb  )
```

Returns

MIDIDevice USB MIDI 2

6.1.2.30 midi03()

```
MIDIDevice midi03 (
             myusb  )
```

Returns

MIDIDevice USB MIDI 3

6.1.2.31 midi04()

```
MIDIDevice midi04 (
             myusb  )
```

Returns

MIDIDevice USB MIDI 4

6.1.2.32 midi05()

```
MIDIDevice midi05 (
             myusb  )
```

Returns

MIDIDevice USB MIDI 5

6.1.2.33 midi06()

```
MIDIDevice midi06 (
             myusb   )
```

Returns

MIDIDevice USB MIDI 6

6.1.2.34 midi07()

```
MIDIDevice midi07 (
             myusb   )
```

Returns

MIDIDevice USB MIDI 7

6.1.2.35 midi08()

```
MIDIDevice midi08 (
             myusb   )
```

Returns

MIDIDevice USB MIDI 8

6.1.2.36 midi09()

```
MIDIDevice midi09 (
             myusb   )
```

Returns

MIDIDevice USB MIDI 9

6.1.2.37 midi10()

```
MIDIDevice midi10 (
             myusb  )
```

Returns

MIDIDevice USB MIDI 10

6.1.2.38 newColor()

```
uint16_t newColor (
             uint8_t r,
             uint8_t g,
             uint8_t b )
```

Color Calculation

Parameters

r	red 0-255
g	green 0-255
b	blue 0-255

Returns

uint16_t 16 bit color

6.1.2.39 read16()

```
uint16_t read16 (
            File & f )
```

[Eric] -document me

Parameters

f	[Eric] -document me

Returns

uint16_t 16 bit value

6.1.2.40 read32()

```
uint32_t read32 (
            File & f )
```

[Eric] -document me

Parameters

f	[Eric] -document me

Returns

uint32_t 32 bit value

6.1.2.41 reOrderR()

```
int reOrderR (
             int r )
```

[Eric] -document me

Parameters

r	[Eric] -document me

Returns

int value

6.1.2.42 setDAC()

```
void setDAC (
             int dac,
             uint32_t data )
```

set DAC output value

Parameters

dac	DAC identifier
data	32 bit output value

6.1.2.43 transmitMIDI()

```
void transmitMIDI (
             int t,
             int d1,
             int d2,
             int ch,
             byte inPort )
```

[Eric] -document me

Parameters

t	[Eric] -document me
d1	[Eric] -document me
d2	[Eric] -document me
ch	[Eric] -document me
inPort	[Eric] -document me

6.1.2.44 transmitSysEx()

```
void transmitSysEx (
            unsigned int len,
            const uint8_t * sysexarray,
            byte inPort )
```

[Eric] -document me

Parameters

len	[Eric] -document me
sysexarray	[Eric] -document me
inPort	[Eric] -document me

6.1.2.45 withinBox()

```
boolean withinBox (
            int x,
            int y,
            int bx,
            int by,
            int bw,
            int bh )
```

Parameters

x	x location
y	y location
bx	box x location
by	box y location
bw	box width
bh	box height

Returns

boolean true if x,y location is inside the box

6.1.3 Variable Documentation

6.1.3.1 midilist

```
MIDIDevice* midilist[10]
```

Initial value:

```
= {
    &midi01, &midi02, &midi03, &midi04, &midi05, &midi06, &midi07, &midi08, &midi09, &midi10
}
```

a list of usb midi devices for easy array access

6.1.3.2 myusb

```
USBHost myusb
```

Returns

USBHost USB Host

6.1.3.3 routing

```
uint8_t routing[50][50]
```

Initial routing matrix for routing

Each byte in the routing matrix is decoded thusly:

bit 0 = keyboard (note on/off, pitchbend, aftertouch, etc) bit 1 = parameters (CC, NRPN/RPN, Sysex parameters etc) bit 3 = transport (clock, start/stop) bit 4 = global channel flag (0 = pass all channels, 1 = filter using bits 5-8) bits 5-8 = channel filter (only pass events matching this channel)

6.1.3.4 SD

```
SdFatSdioEX SD
```

[KURT] this will be moving into the MIDI Router library

SdFatSdioEX - SD card object

6.1.3.5 SysCsvFile

```
File SysCsvFile
```

[KURT] this will be moving into the MIDI Router library

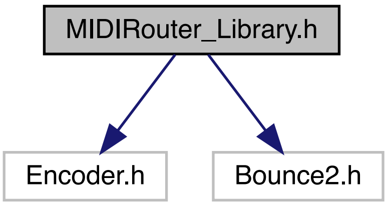
File - SysCsvFile csv file object

6.2 MIDIRouter_Library.h File Reference

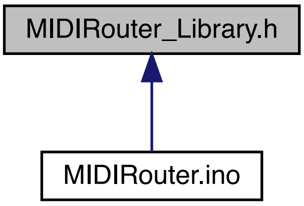
Library header Project MIDIRouter Library.

```
#include "MIDIRouter_Library_Defs.h"
#include "MRPorts.h"
#include "MidiFilter.h"
#include <Encoder.h>
#include <Bounce2.h>
#include "MIDIRouter_Library.hpp"
```

Include dependency graph for MIDIRouter_Library.h:



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



Data Structures

- struct MIDIRouter::MIDIRouterSetup

  Structure that describes the pins used for a midi router setup.

- class MIDIRouter::MIDIRouter_Lib

  MIDI Router Library object.

6.2.1 Detailed Description

Library header Project MIDIRouter Library.

MIDI Router Library

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

Author

Kurt Arnlund

Timeline85 / Ingenious Arts and Technologies LLC

Date

4/25/20 6:04 PM

Version

0.0.1

Copyright

(c) Kurt Arnlund, 2020

GNU General Public Licence

See also

ReadMe.txt for references

Index

active
    MIDIRouter::MRPort, 27
asUint16
    RGBColor, 29

bmpDraw
    MIDIRouter.ino, 41

color565
    MIDIRouter.ino, 41
controllersFiltered
    MIDIRouter::MidiFilter, 10
csvReadDouble
    MIDIRouter.ino, 41
csvReadFloat
    MIDIRouter.ino, 42
csvReadInt16
    MIDIRouter.ino, 42
csvReadInt32
    MIDIRouter.ino, 43
csvReadText
    MIDIRouter.ino, 43
csvReadUint16
    MIDIRouter.ino, 43
csvReadUint32
    MIDIRouter.ino, 44
CVnoteCal
    MIDIRouter.ino, 44
CVparamCal
    MIDIRouter.ino, 45

dPrint
    MIDIRouter.ino, 45
drawHomeScreen
    MIDIRouter.ino, 45
drawPiano
    MIDIRouter.ino, 46
dWrite
    MIDIRouter.ino, 46

encoder
    MIDIRouter::MIDIRouter_Lib, 15
encPush
    MIDIRouter::MIDIRouter_Lib, 16

filtRoute
    MIDIRouter.ino, 46
flashIn
    MIDIRouter.ino, 47

get
    MIDIRouter::MidiFilterType, 14
getTouchCol
    MIDIRouter.ino, 47
getTouchRow
    MIDIRouter.ino, 47

hub1
    MIDIRouter.ino, 48
hub2
    MIDIRouter.ino, 48
hub3
    MIDIRouter.ino, 48
hub4
    MIDIRouter.ino, 48

inputAt
    MIDIRouter::MIDIRouter_Lib, 16

knob_calCV
    MIDIRouter.ino, 48
knobFull
    MIDIRouter.ino, 49
knobSet
    MIDIRouter.ino, 49

location
    MIDIRouter::MRPort, 28

matchSysExID
    MIDIRouter.ino, 49
midi01
    MIDIRouter.ino, 49
midi02
    MIDIRouter.ino, 49
midi03
    MIDIRouter.ino, 50
midi04
    MIDIRouter.ino, 50
midi05
    MIDIRouter.ino, 50
midi06
    MIDIRouter.ino, 50
midi07
    MIDIRouter.ino, 51
midi08
    MIDIRouter.ino, 51
midi09
    MIDIRouter.ino, 51
midi10
    MIDIRouter.ino, 51
midiclockFiltered

MIDIRouter::MidiFilter, 10
midilist
MIDIRouter.ino, 55
MIDIRouter.ino, 31
bmpDraw, 41
color565, 41
csvReadDouble, 41
csvReadFloat, 42
csvReadInt16, 42
csvReadInt32, 43
csvReadText, 43
csvReadUint16, 43
csvReadUint32, 44
CVnoteCal, 44
CVparamCal, 45
dPrint, 45
drawHomeScreen, 45
drawPiano, 46
dWrite, 46
filtRoute, 46
flashIn, 47
getTouchCol, 47
getTouchRow, 47
hub1, 48
hub2, 48
hub3, 48
hub4, 48
knob_calCV, 48
knobFull, 49
knobSet, 49
matchSysExID, 49
midi01, 49
midi02, 49
midi03, 50
midi04, 50
midi05, 50
midi06, 50
midi07, 51
midi08, 51
midi09, 51
midi10, 51
midilist, 55
myusb, 56
newColor, 52
read16, 53
read32, 53
reOrderR, 53
routing, 56
SD, 56
setDAC, 54
SysCsvFile, 56
transmitMIDI, 54
transmitSysEx, 55
withinBox, 55
MIDIRouter::ByteBitLocation, 9
MIDIRouter::MidiFilter, 9
controllersFiltered, 10
midiclockFiltered, 10
notesFiltered, 11
realtimeFiltered, 11
setControllers, 11
setMidiclock, 12
setNotes, 12
setRealtime, 12
setSysex, 13
sysexFiltered, 13
MIDIRouter::MidiFilterType, 13
get, 14
set, 14
MIDIRouter::MIDIRouter_Lib, 15
encoder, 15
encPush, 16
inputAt, 16
outputAt, 16
SetupEncoder, 17
MIDIRouter::MIDIRouterSetup, 17
MIDIRouter::MRInputPort, 18
MRInputPort, 19
MIDIRouter::MRIO_MidiHardwareSerialInterface< SerialPort >, 19
MRIO_MidiHardwareSerialInterface, 21
MIDIRouter::MRIO_MidiUSBClientInterface, 21
MIDIRouter::MRIO_MidiUSBHostInterface, 22
MIDIRouter::MRIOInterface, 23
MRIOInterface, 24
MIDIRouter::MROutputPort, 24
MROutputPort, 25
MIDIRouter::MRPort, 26
active, 27
location, 28
MRPort, 27
MIDIRouter_Library.h, 57
MRInputPort
MIDIRouter::MRInputPort, 19
MRIO_MidiHardwareSerialInterface
MIDIRouter::MRIO_MidiHardwareSerialInterface< SerialPort >, 21
MRIOInterface
MIDIRouter::MRIOInterface, 24
MROutputPort
MIDIRouter::MROutputPort, 25
MRPort
MIDIRouter::MRPort, 27
myusb
MIDIRouter.ino, 56

newColor
MIDIRouter.ino, 52
notesFiltered
MIDIRouter::MidiFilter, 11

outputAt
MIDIRouter::MIDIRouter_Lib, 16

read16
MIDIRouter.ino, 53
read32

MIDIRouter.ino, 53
realtimeFiltered
MIDIRouter::MidiFilter, 11
reOrderR
MIDIRouter.ino, 53
RGBColor, 28
asUint16, 29
RGBColor, 28
routing
MIDIRouter.ino, 56

SD
MIDIRouter.ino, 56
set
MIDIRouter::MidiFilterType, 14
setControllers
MIDIRouter::MidiFilter, 11
setDAC
MIDIRouter.ino, 54
setMidiclock
MIDIRouter::MidiFilter, 12
setNotes
MIDIRouter::MidiFilter, 12
setRealtime
MIDIRouter::MidiFilter, 12
setSysex
MIDIRouter::MidiFilter, 13
SetupEncoder
MIDIRouter::MIDIRouter_Lib, 17
SysCsvFile
MIDIRouter.ino, 56
sysexFiltered
MIDIRouter::MidiFilter, 13

transmitMIDI
MIDIRouter.ino, 54
transmitSysEx
MIDIRouter.ino, 55

withinBox
MIDIRouter.ino, 55