edison ctrlmod

Generated by Doxygen 1.8.6

Thu Aug 27 2015 18:47:19

Contents

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 mraa Namespace Reference

5.2 std Namespace Reference

Functions

- template<class T >
  T round (const T &n)

5.2.1 Function Documentation

5.2.1.1 template<class T > T std::round (const T & *n*)

Chapter 6

Class Documentation

6.1 Command Struct Reference

`#include <edcomm_system.h>`

Public Member Functions

- Command ()

Public Attributes

- union {
  struct {
  uint32_t hash_id
  uint32_t cmd_data
  double cmd_data_d
  double cmd_data_d2
  double cmd_data_d3
  double cmd_data_d4
  double cmd_data_d5
  double cmd_data_d6
  double cmd_data_d7
  double cmd_data_d8
  }
  uint8_t data [COMMAND_BYTE_SIZE]
  };

6.1.1 Constructor & Destructor Documentation

6.1.1.1 Command::Command ( )

6.1.2 Member Data Documentation

6.1.2.1 union { ... }

6.1.2.2 uint32_t Command::cmd_data

6.1.2.3 double Command::cmd_data_d

6.1.2.4 double Command::cmd_data_d2

6.1.2.5 double Command::cmd_data_d3

6.1.2.6 double Command::cmd_data_d4

6.1.2.7 double Command::cmd_data_d5

6.1.2.8 double Command::cmd_data_d6

6.1.2.9 double Command::cmd_data_d7

6.1.2.10 double Command::cmd_data_d8

6.1.2.11 uint8_t Command::data[COMMAND_BYTE_SIZE]

6.1.2.12 uint32_t Command::hash_id

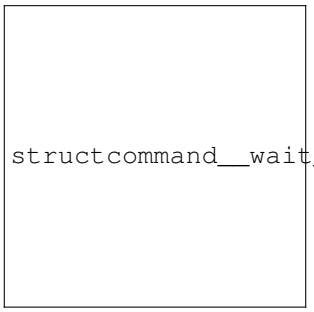
The documentation for this struct was generated from the following files:

- /home/dprandle/Documents/code/ctrlmod/include/edcomm_system.h
- /home/dprandle/Documents/code/ctrlmod/src/edcomm_system.cpp

6.2 command_wait_callback Struct Reference

`#include <edthreaded_fd.h>`

Inheritance diagram for command_wait_callback:



Public Member Functions

- command_wait_callback (edthreaded_fd *_handle)
- void exec ()

Public Attributes

- edthreaded_fd * handle

6.2.1 Constructor & Destructor Documentation

6.2.1.1 command_wait_callback::command_wait_callback (edthreaded_fd * *_handle*) [inline]

6.2.2 Member Function Documentation

6.2.2.1 void command_wait_callback::exec ( ) [inline],[virtual]

Reimplemented from wait_ready_callback.

6.2.3 Member Data Documentation

6.2.3.1 edthreaded_fd* command_wait_callback::handle

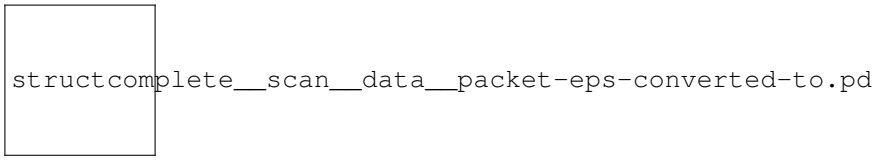
The documentation for this struct was generated from the following file:

- /home/dprandle/Documents/code/ctrlmod/include/edthreaded_fd.h

6.3 complete_scan_data_packet Struct Reference

`#include <edrplidar_packets.h>`

Inheritance diagram for complete_scan_data_packet:



Public Member Functions

- complete_scan_data_packet ()
- virtual std::string toString ()
- std::string type ()
- virtual uint32_t size ()
- virtual uint8_t & operator[] (uint32_t index)
- virtual uint8_t * dataptr ()

Static Public Member Functions

- static std::string Type ()
- static uint32_t Size ()

Public Attributes

- scan_data_packet data [360]

6.3.1 Constructor & Destructor Documentation

6.3.1.1 complete_scan_data_packet::complete_scan_data_packet ( )

6.3.2 Member Function Documentation

6.3.2.1 virtual uint8_t* complete_scan_data_packet::dataptr ( ) [inline], [virtual]

Implements data_packet.

6.3.2.2 virtual uint8_t& complete_scan_data_packet::operator[] (uint32_t *index*) [inline], [virtual]

Implements data_packet.

6.3.2.3 virtual uint32_t complete_scan_data_packet::size ( ) [inline], [virtual]

Implements data_packet.

6.3.2.4 static uint32_t complete_scan_data_packet::Size ( ) [inline], [static]

6.3.2.5 std::string complete_scan_data_packet::toString ( ) [virtual]

Implements data_packet.

6.3.2.6 std::string complete_scan_data_packet::type ( ) [inline], [virtual]

Implements data_packet.

6.3.2.7 static std::string complete_scan_data_packet::Type ( ) [inline], [static]

6.3.3 Member Data Documentation

6.3.3.1 scan_data_packet complete_scan_data_packet::data[360]

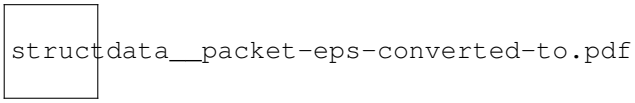
The documentation for this struct was generated from the following files:

- /home/dprandle/Documents/code/ctrlmod/include/edrplidar_packets.h
- /home/dprandle/Documents/code/ctrlmod/src/edrplidar_packets.cpp

6.4 data_packet Struct Reference

`#include <edrplidar_packets.h>`

Inheritance diagram for data_packet:



Public Member Functions

- data_packet ()
- virtual ~data_packet ()
- virtual std::string toString ()=0
- virtual std::string type ()=0
- virtual uint32_t size ()=0
- virtual uint8_t & operator[] (uint32_t index)=0
- virtual uint8_t * dataptr ()=0

6.4.1 Constructor & Destructor Documentation

6.4.1.1 data_packet::data_packet ( ) [inline]

6.4.1.2 virtual data_packet::~data_packet ( ) [inline], [virtual]

6.4.2 Member Function Documentation

6.4.2.1 virtual uint8_t* data_packet::dataptr ( ) [pure virtual]

Implemented in firmware_data_packet, info_data_packet, health_data_packet, complete_scan_data_packet, and scan_data_packet.

6.4.2.2 virtual uint8_t& data_packet::operator[] (uint32_t *index*) [pure virtual]

Implemented in firmware_data_packet, info_data_packet, health_data_packet, complete_scan_data_packet, and scan_data_packet.

6.4.2.3 virtual uint32_t data_packet::size ( ) [pure virtual]

Implemented in firmware_data_packet, info_data_packet, health_data_packet, complete_scan_data_packet, and scan_data_packet.

6.4.2.4 virtual std::string data_packet::toString ( ) [pure virtual]

Implemented in firmware_data_packet, info_data_packet, health_data_packet, complete_scan_data_packet, and scan_data_packet.

6.4.2.5 virtual std::string data_packet::type ( ) [pure virtual]

Implemented in firmware_data_packet, info_data_packet, health_data_packet, complete_scan_data_packet, and scan_data_packet.

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

- /home/dprandle/Documents/code/ctrlmod/include/edrplidar_packets.h

6.5 eduart::DataFormat Struct Reference

```
#include <eduart.h>
```

Public Member Functions

- DataFormat ()

Public Attributes

- DataBits db
- Parity p
- StopBits sb

6.5.1 Constructor & Destructor Documentation

6.5.1.1 eduart::DataFormat::DataFormat ( ) [inline]

6.5.2 Member Data Documentation

6.5.2.1 DataBits eduart::DataFormat::db

6.5.2.2 Parity eduart::DataFormat::p

6.5.2.3 StopBits eduart::DataFormat::sb

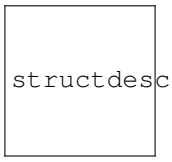
The documentation for this struct was generated from the following file:

- /home/dprandle/Documents/code/ctrlmod/include/eduart.h

6.6 descriptor_packet Struct Reference

`#include <edrplidar_packets.h>`

Inheritance diagram for descriptor_packet:



Public Member Functions

- descriptor_packet (uint8_t drlen0_=0x00, uint8_t drlen1_=0x00, uint8_t drlen2_=0x00, uint8_t drlen3_smode_=0x00, uint8_t datatype_=0x00)
- virtual ~descriptor_packet ()
- virtual std::string type ()=0
- uint32_t size ()
- uint8_t & operator[] (uint32_t index)

Static Public Member Functions

- static uint32_t Size ()

Public Attributes

- union {
  struct {
  uint8_t s1
  uint8_t s2
  uint8_t drlen0
  uint8_t drlen1
  uint8_t drlen2
  uint8_t drlen3_smode
  uint8_t datatype
  }

```
    uint8_t data [7]
};
```

6.6.1 Constructor & Destructor Documentation

6.6.1.1 descriptor_packet::descriptor_packet (uint8_t *drlen0_* = 0x00, uint8_t *drlen1_* = 0x00, uint8_t *drlen2_* = 0x00, uint8_t *drlen3_smode_* = 0x00, uint8_t *datatype_* = 0x00) [inline]

6.6.1.2 virtual descriptor_packet::~descriptor_packet ( ) [inline], [virtual]

6.6.2 Member Function Documentation

6.6.2.1 uint8_t& descriptor_packet::operator[] (uint32_t *index*) [inline]

6.6.2.2 uint32_t descriptor_packet::size ( ) [inline]

6.6.2.3 static uint32_t descriptor_packet::Size ( ) [inline], [static]

6.6.2.4 virtual std::string descriptor_packet::type ( ) [pure virtual]

Implemented in device_health_descriptor, device_info_descriptor, and scan_descriptor.

6.6.3 Member Data Documentation

6.6.3.1 union { ... }

6.6.3.2 uint8_t descriptor_packet::data[7]

6.6.3.3 uint8_t descriptor_packet::datatype

6.6.3.4 uint8_t descriptor_packet::drlen0

6.6.3.5 uint8_t descriptor_packet::drlen1

6.6.3.6 uint8_t descriptor_packet::drlen2

6.6.3.7 uint8_t descriptor_packet::drlen3_smode

6.6.3.8 uint8_t descriptor_packet::s1

6.6.3.9 uint8_t descriptor_packet::s2

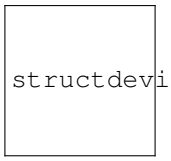
The documentation for this struct was generated from the following file:

- /home/dprandle/Documents/code/ctrlmod/include/edrplidar_packets.h

6.7 device_health_descriptor Struct Reference

```
#include <edrplidar_packets.h>
```

Inheritance diagram for device_health_descriptor:



Public Member Functions

- device_health_descriptor ()
- virtual std::string type ()

Additional Inherited Members

6.7.1 Constructor & Destructor Documentation

6.7.1.1 device_health_descriptor::device_health_descriptor ( ) [inline]

6.7.2 Member Function Documentation

6.7.2.1 virtual std::string device_health_descriptor::type ( ) [inline], [virtual]

Implements descriptor_packet.

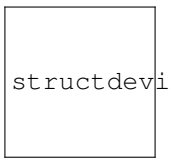
The documentation for this struct was generated from the following file:

- /home/dprandle/Documents/code/ctrlmod/include/edrplidar_packets.h

6.8 device_health_request Struct Reference

`#include <edrplidar_packets.h>`

Inheritance diagram for device_health_request:



Public Member Functions

- device_health_request ()

Additional Inherited Members

6.8.1 Constructor & Destructor Documentation

6.8.1.1 device_health_request::device_health_request ( ) [inline]

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

- /home/dprandle/Documents/code/ctrlmod/include/edrplidar_packets.h

6.9 device_info_descriptor Struct Reference

`#include <edrplidar_packets.h>`

Inheritance diagram for device_info_descriptor:

Public Member Functions

- device_info_descriptor ()
- virtual std::string type ()

Additional Inherited Members

6.9.1 Constructor & Destructor Documentation

6.9.1.1 device_info_descriptor::device_info_descriptor ( ) [inline]

6.9.2 Member Function Documentation

6.9.2.1 virtual std::string device_info_descriptor::type ( ) [inline], [virtual]

Implements descriptor_packet.

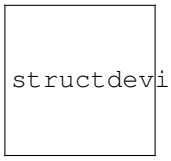
The documentation for this struct was generated from the following file:

- /home/dprandle/Documents/code/ctrlmod/include/edrplidar_packets.h

6.10 device_info_request Struct Reference

`#include <edrplidar_packets.h>`

Inheritance diagram for device_info_request:



Public Member Functions

- device_info_request ()

Additional Inherited Members

6.10.1 Constructor & Destructor Documentation

6.10.1.1 device_info_request::device_info_request ( ) [inline]

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

- /home/dprandle/Documents/code/ctrlmod/include/edrplidar_packets.h

6.11 edcallback Struct Reference

`#include <edcallback.h>`

Inheritance diagram for edcallback:

Public Member Functions

- virtual ~edcallback ()
- virtual void exec ()=0

6.11.1 Constructor & Destructor Documentation

6.11.1.1 virtual edcallback::~edcallback ( ) [inline], [virtual]

6.11.2 Member Function Documentation

6.11.2.1 virtual void edcallback::exec ( ) [pure virtual]

Implemented in command_wait_callback, instruction_callback, edpl_callback, and wait_ready_callback.

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

- /home/dprandle/Documents/code/ctrlmod/include/edcallback.h

6.12 edcomm_system Class Reference

`#include <edcomm_system.h>`

Inheritance diagram for edcomm_system:

Public Member Functions

- edcomm_system ()
- virtual ~edcomm_system ()
- virtual void init ()
- virtual void release ()
- virtual bool process (edmessage *msg)
- uint16_t port ()
- void set_port (uint16_t port_)
- virtual void update ()
- uint32_t recvFromClients (uint8_t *data, uint32_t max_size)
- void sendToClients (uint8_t *data, uint32_t size)
- virtual std::string typestr ()

Static Public Member Functions

- static std::string TypeString ()

6.12.1 Constructor & Destructor Documentation

6.12.1.1 edcomm_system::edcomm_system ( )

6.12.1.2 edcomm_system::~edcomm_system ( ) [virtual]

6.12.2 Member Function Documentation

6.12.2.1 void edcomm_system::init ( ) [virtual]

Implements edsystem.

6.12.2.2 uint16_t edcomm_system::port ( )

6.12.2.3 bool edcomm_system::process (edmessage * *msg*) [virtual]

Implements edsystem.

6.12.2.4 uint32_t edcomm_system::recvFromClients (uint8_t * *data*, uint32_t *max_size*)

6.12.2.5 void edcomm_system::release ( ) [virtual]

Implements edsystem.

6.12.2.6 void edcomm_system::sendToClients (uint8_t * *data*, uint32_t *size*)

6.12.2.7 void edcomm_system::set_port (uint16_t *port_*)

6.12.2.8 virtual std::string edcomm_system::typestr ( ) [inline], [virtual]

Implements edsystem.

6.12.2.9 static std::string edcomm_system::TypeString ( ) [inline], [static]

6.12.2.10 void edcomm_system::update ( ) [virtual]

Implements edsystem.

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

- /home/dprandle/Documents/code/ctrlmod/include/edcomm_system.h
- /home/dprandle/Documents/code/ctrlmod/src/edcomm_system.cpp

6.13 edi2c Class Reference

edi2c

`#include <edi2c.h>`

Inheritance diagram for edi2c:

Public Member Functions

- edi2c (uint32_t adapterNum=1)
- ~edi2c ()
- bool command_read (uint8_t reg, uint32_t bytes_to_read)

  command_read
- void enable_smbus (bool enable)

  enaable_smbus
- uint8_t read_byte ()

  read_byte
- uint16_t read_word ()

  read_word
- uint8_t read_reg_byte (uint8_t reg)

  read_reg_byte
- int16_t read_reg_word (uint8_t reg)

  read_reg_word
- void read_reg_bytes (uint8_t reg, uint8_t *buffer, uint32_t size)

  read_reg_bytes
- uint16_t read_delay ()

  read_delay

- uint16_t write_delay ()

  write_delay
- void set_read_delay (uint16_t ms)

  set_read_delay
- void set_write_delay (uint16_t ms)
- void set_target_address (int32_t addr)

  set_target_address
- bool smbus_enabled ()

  smbus_enables
- bool start ()

  start
- int32_t target_address ()
- bool write_byte (uint8_t byte)
- bool write_word (uint16_t word)
- bool write_reg_byte (uint8_t reg, uint8_t byte)
- bool write_reg_word (uint8_t reg, int16_t word)
- bool write_reg_bytes (uint8_t reg, uint8_t *bytes, uint32_t size)

Additional Inherited Members

6.13.1 Detailed Description

edi2c

Creates a new thread to run all communication transactions using i2c protocol. The way to use this class is just like any other "edthreaded_fd" subclass - with a main exception: The read_* functions are all blocking. The read function itself is not blocking. If you want to read a value from a register in a non blocking fashion then you must use command_read instead of read_* functions. The command_read function takes how many bytes to read as a parameter - and you can access theses bytes with "read".

6.13.2 Constructor & Destructor Documentation

6.13.2.1 edi2c::edi2c (uint32_t *adapterNum* = `1`)

6.13.2.2 edi2c::~edi2c ( )

6.13.3 Member Function Documentation

6.13.3.1 bool edi2c::command_read (uint8_t *reg*, uint32_t *bytes_to_read*)

command_read

Read from a register in a non-blocking fashion - once the bytes have been read they will be available through read function. Nothing more will be written to the i2c device until bytes_to_read bytes have been read or the timeout period of time has been reached.

Parameters

reg	register to read bytes from
bytes_to_read	amount of bytes to read

Returns

Whether the command was successful or not. Will not be, for example, if no device is connected.

6.13.3.2 void edi2c::enable_smbus (bool *enable*)

enaable_smbus

Disabled by default, this will enable the smbus functions. Smbus supports more advanced styles of messaging between devices but not all devices support the smbus protocol.

Parameters

enable	Enable (true) or disable (false)

6.13.3.3 uint8_t edi2c::read_byte ( )

read_byte

Blocks until one byte has been read or the maximum wait timeout has been reached.

Returns

byte that has been read

6.13.3.4 uint16_t edi2c::read_delay ( )

read_delay

Returns

number of milliseconds to delay the thread after each read command.

6.13.3.5 uint8_t edi2c::read_reg_byte (uint8_t *reg*)

read_reg_byte

Blocking read until 1 byte is read from a register.

Parameters

reg	Register to read from

Returns

Byte value read from register

6.13.3.6 void edi2c::read_reg_bytes (uint8_t *reg,* uint8_t * *buffer,* uint32_t *size*)

read_reg_bytes

Blocking read until size bytes has been read. This is a convenience function which is the same as calling command-_read(reg, size); int cnt = 0; while (cnt != size) cnt += read(buffer+cnt, size-cnt);

Parameters

reg	Register to read bytes from

buffer	Buffer to store read in bytes - bounds are not checked so make sure it is big enough
size	Number of bytes to read

6.13.3.7 int16_t edi2c::read_reg_word (uint8_t *reg*)

read_reg_word

Blocking read until 1 word is read from a register

Parameters

reg	Register to read from

Returns

2 byte value read from register

6.13.3.8 uint16_t edi2c::read_word ( )

read_word

Blocks until one word has been read or the maximum wait timeout has been reached.

Returns

16 bit word

6.13.3.9 void edi2c::set_read_delay (uint16_t *ms*)

set_read_delay

Set the thread read delay - how many milliseconds to delay the thread after each read

Parameters

ms	Number of milliseconds

6.13.3.10 void edi2c::set_target_address (int32_t *addr*)

set_target_address

Set the target device address - all reads and writes will be done using this address

Parameters

addr	usually 7 bit device address for the slave (can be 10 bit)

6.13.3.11 void edi2c::set_write_delay (uint16_t *ms*)

Set the thread write delay - how many milliseconds to delay the thread after each write

Parameters

ms	Number of milliseconds

6.13.3.12 bool edi2c::smbus_enabled ( )

smbus_enables

Returns

Is smbus mode enabled?

6.13.3.13 bool edi2c::start ( ) [virtual]

start

Starts a new thread for communication with the device. It also opens the file descriptor using the bus number supplied in the constructor. By default this is bus 1. If a thread has already been started, or a file descriptor is already open, this function will fail.

Returns

true for success and false for fail - check error code on fail

Reimplemented from edthreaded_fd.

6.13.3.14 int32_t edi2c::target_address ( )

6.13.3.15 bool edi2c::write_byte (uint8_t *byte*)

6.13.3.16 uint16_t edi2c::write_delay ( )

write_delay

Returns

number of milliseconds to delay the thread after each write command.

6.13.3.17 bool edi2c::write_reg_byte (uint8_t *reg*, uint8_t *byte*)

6.13.3.18 bool edi2c::write_reg_bytes (uint8_t *reg*, uint8_t * *bytes*, uint32_t *size*)

6.13.3.19 bool edi2c::write_reg_word (uint8_t *reg*, int16_t *word*)

6.13.3.20 bool edi2c::write_word (uint16_t *word*)

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

- /home/dprandle/Documents/code/ctrlmod/include/edi2c.h
- /home/dprandle/Documents/code/ctrlmod/src/edi2c.cpp

6.14 edimu_system Class Reference

`#include <edimu_system.h>`

Inheritance diagram for edimu_system:

Public Types

- enum g_scale { G_SCALE_245DPS, G_SCALE_500DPS, G_SCALE_2000DPS }

  possible ranges of the gyroscope
- enum a_scale {
  A_SCALE_2G, A_SCALE_4G, A_SCALE_6G, A_SCALE_8G,
  A_SCALE_16G }

  Possible FSR's of the accelerometer.
- enum m_scale { M_SCALE_2GS, M_SCALE_4GS, M_SCALE_8GS, M_SCALE_12GS }

  Possible FSR's of the magnetometer.
- enum g_odr {
  G_ODR_95_BW_125 = 0x0, G_ODR_95_BW_25 = 0x1, G_ODR_190_BW_125 = 0x4, G_ODR_190_BW_-
  25 = 0x5,
  G_ODR_190_BW_50 = 0x6, G_ODR_190_BW_70 = 0x7, G_ODR_380_BW_20 = 0x8, G_ODR_380_BW_-
  25 = 0x9,
  G_ODR_380_BW_50 = 0xA, G_ODR_380_BW_100 = 0xB, G_ODR_760_BW_30 = 0xC, G_ODR_760_BW-
  _35 = 0xD,
  G_ODR_760_BW_50 = 0xE, G_ODR_760_BW_100 = 0xF }

  Possible data rate/bandwidth combos of the gyro.
- enum a_odr {
  A_POWER_DOWN, A_ODR_3125, A_ODR_625, A_ODR_125,
  A_ODR_25, A_ODR_50, A_ODR_100, A_ODR_200,
  A_ODR_400, A_ODR_800, A_ODR_1600 }

  Possible output data rates of the accelerometer.
- enum a_abw { A_ABW_773, A_ABW_194, A_ABW_362, A_ABW_50 }

  Possible anti-aliasing filter rates of the accelerometer.
- enum m_odr {
  M_ODR_3125, M_ODR_625, M_ODR_125, M_ODR_25,
  M_ODR_50, M_ODR_100 }

  Possible output data rates of the magnetometer.

Public Member Functions

- edimu_system ()
- ~edimu_system ()
- a_abw accel_aa ()
- a_scale accel_scale ()
- a_odr accel_datarate ()
- void calibrate ()
- g_scale gyro_scale ()
- g_odr gyro_datarate ()

- m_scale mag_scale ()
- m_odr mag_datarate ()
- void init ()
- bool process (edmessage *msg)
- void release ()
- void set_accel_aa (a_abw antialiasing)
- void set_accel_scale (a_scale scale)
- void set_accel_datarate (a_odr datarate)
- void set_gyro_scale (g_scale scale)
- void set_gyro_datarate (g_odr datarate)
- void set_mag_scale (m_scale scale)
- void set_mag_datarate (m_odr datarate)
- void update ()
- std::string typestr ()

Static Public Member Functions

- static std::string TypeString ()

6.14.1 Member Enumeration Documentation

6.14.1.1 enum edimu_system::a_abw

Possible anti-aliasing filter rates of the accelerometer.

Enumerator

A_ABW_773 773 Hz (0x0)

A_ABW_194 194 Hz (0x1)

A_ABW_362 362 Hz (0x2)

A_ABW_50 50 Hz (0x3)

6.14.1.2 enum edimu_system::a_odr

Possible output data rates of the accelerometer.

Enumerator

A_POWER_DOWN Power-down mode (0x0)

A_ODR_3125 3.125 Hz (0x1)

A_ODR_625 6.25 Hz (0x2)

A_ODR_125 12.5 Hz (0x3)

A_ODR_25 25 Hz (0x4)

A_ODR_50 50 Hz (0x5)

A_ODR_100 100 Hz (0x6)

A_ODR_200 200 Hz (0x7)

A_ODR_400 400 Hz (0x8)

A_ODR_800 800 Hz (9)

A_ODR_1600 1600 Hz (0xA)

6.14.1.3 enum edimu_system::a_scale

Possible FSR's of the accelerometer.

Enumerator

A_SCALE_2G 2g

A_SCALE_4G 4g

A_SCALE_6G 6g

A_SCALE_8G 8g

A_SCALE_16G 16g

6.14.1.4 enum edimu_system::g_odr

Possible data rate/bandwidth combos of the gyro.

Enumerator

G_ODR_95_BW_125 95 12.5

G_ODR_95_BW_25 95 25

G_ODR_190_BW_125 190 12.5

G_ODR_190_BW_25 190 25

G_ODR_190_BW_50 190 50

G_ODR_190_BW_70 190 70

G_ODR_380_BW_20 380 20

G_ODR_380_BW_25 380 25

G_ODR_380_BW_50 380 50

G_ODR_380_BW_100 380 100

G_ODR_760_BW_30 760 30

G_ODR_760_BW_35 760 35

G_ODR_760_BW_50 760 50

G_ODR_760_BW_100 760 100

6.14.1.5 enum edimu_system::g_scale

possible ranges of the gyroscope

Enumerator

G_SCALE_245DPS 245 degrees per second

G_SCALE_500DPS 500 dps

G_SCALE_2000DPS 2000 dps

6.14.1.6 enum edimu_system::m_odr

Possible output data rates of the magnetometer.

Enumerator

M_ODR_3125 3.125 Hz (0x00)
M_ODR_625 6.25 Hz (0x01)
M_ODR_125 12.5 Hz (0x02)
M_ODR_25 25 Hz (0x03)
M_ODR_50 50 (0x04)
M_ODR_100 100 Hz (0x05)

6.14.1.7 enum edimu_system::m_scale

Possible FSR's of the magnetometer.

Enumerator

M_SCALE_2GS 2Gs
M_SCALE_4GS 4Gs
M_SCALE_8GS 8Gs
M_SCALE_12GS 12Gs

6.14.2 Constructor & Destructor Documentation

6.14.2.1 edimu_system::edimu_system ( )

6.14.2.2 edimu_system::~edimu_system ( )

6.14.3 Member Function Documentation

6.14.3.1 edimu_system::a_abw edimu_system::accel_aa ( )

6.14.3.2 edimu_system::a_odr edimu_system::accel_datarate ( )

6.14.3.3 edimu_system::a_scale edimu_system::accel_scale ( )

6.14.3.4 void edimu_system::calibrate ( )

6.14.3.5 edimu_system::g_odr edimu_system::gyro_datarate ( )

6.14.3.6 edimu_system::g_scale edimu_system::gyro_scale ( )

6.14.3.7 void edimu_system::init ( ) [virtual]

Implements edsystem.

6.14.3.8 edimu_system::m_odr edimu_system::mag_datarate ( )

6.14.3.9 edimu_system::m_scale edimu_system::mag_scale ( )

6.14.3.10 bool edimu_system::process (edmessage * *msg*) [virtual]

Implements edsystem.

6.14.3.11 void edimu_system::release ( ) [virtual]

Implements edsystem.

6.14.3.12 void edimu_system::set_accel_aa (a_abw *antialiasing*)

6.14.3.13 void edimu_system::set_accel_datarate (a_odr *datarate*)

6.14.3.14 void edimu_system::set_accel_scale (a_scale *scale*)

6.14.3.15 void edimu_system::set_gyro_datarate (g_odr *datarate*)

6.14.3.16 void edimu_system::set_gyro_scale (g_scale *scale*)

6.14.3.17 void edimu_system::set_mag_datarate (m_odr *datarate*)

6.14.3.18 void edimu_system::set_mag_scale (m_scale *scale*)

6.14.3.19 std::string edimu_system::typestr ( ) [inline], [virtual]

Implements edsystem.

6.14.3.20 static std::string edimu_system::TypeString ( ) [inline], [static]

6.14.3.21 void edimu_system::update ( ) [virtual]

Implements edsystem.

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

- /home/dprandle/Documents/code/ctrlmod/include/edimu_system.h
- /home/dprandle/Documents/code/ctrlmod/src/edimu_system.cpp

6.15 edlogging_system Class Reference

`#include <edlogging_system.h>`

Inheritance diagram for edlogging_system:

Public Member Functions

- edlogging_system ()
- virtual ~edlogging_system ()
- virtual void init ()
- virtual void release ()
- virtual bool process (edmessage *msg)
- virtual void update ()
- virtual std::string typestr ()

Static Public Member Functions

- static std::string TypeString ()

6.15.1 Constructor & Destructor Documentation

6.15.1.1 edlogging_system::edlogging_system ( ) [inline]

6.15.1.2 virtual edlogging_system::~edlogging_system ( ) [inline], [virtual]

6.15.2 Member Function Documentation

6.15.2.1 void edlogging_system::init ( ) [virtual]

Implements edsystem.

6.15.2.2 bool edlogging_system::process (edmessage * *msg*) [virtual]

Implements edsystem.

6.15.2.3 void edlogging_system::release ( ) [virtual]

Implements edsystem.

6.15.2.4 virtual std::string edlogging_system::typestr ( ) [inline], [virtual]

Implements edsystem.

6.15.2.5 static std::string edlogging_system::TypeString ( ) [inline], [static]

6.15.2.6 void edlogging_system::update ( ) [virtual]

Implements edsystem.

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

- /home/dprandle/Documents/code/ctrlmod/include/edlogging_system.h
- /home/dprandle/Documents/code/ctrlmod/src/edlogging_system.cpp

6.16 edmctrl Class Reference

`#include <edmctrl.h>`

Public Member Functions

- edmctrl ()
- virtual ~edmctrl ()
- template<class T >
  T * add_sys ()
- bool running ()
- void init ()

- void release ()
- edmessage_dispatch * message_dispatch ()
- void start ()
- void stop ()
- edtimer * sys_timer ()
- void update ()
- template<class T >
  void rm_sys ()
- void rm_sys (const std::string &sysname)
- template<class T >
  T * sys ()
- edsystem * sys (const std::string &sysname)

Static Public Member Functions

- static edmctrl & inst ()
- static void quit (void)

6.16.1 Constructor & Destructor Documentation

6.16.1.1 edmctrl::edmctrl ( )

6.16.1.2 edmctrl::~edmctrl ( ) [virtual]

6.16.2 Member Function Documentation

6.16.2.1 template<class T > T* edmctrl::add_sys ( ) [inline]

6.16.2.2 void edmctrl::init ( )

6.16.2.3 edmctrl & edmctrl::inst ( ) [static]

6.16.2.4 edmessage_dispatch * edmctrl::message_dispatch ( )

6.16.2.5 void edmctrl::quit (void) [static]

6.16.2.6 void edmctrl::release ( )

6.16.2.7 template<class T > void edmctrl::rm_sys ( ) [inline]

6.16.2.8 void edmctrl::rm_sys (const std::string & *sysname*)

6.16.2.9 bool edmctrl::running ( )

6.16.2.10 void edmctrl::start ( )

6.16.2.11 void edmctrl::stop ( )

6.16.2.12 template<class T > T* edmctrl::sys ( ) [inline]

6.16.2.13 edsystem * edmctrl::sys (const std::string & *sysname*)

6.16.2.14 edtimer * edmctrl::sys_timer ( )

6.16.2.15 void edmctrl::update ( )

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

- /home/dprandle/Documents/code/ctrlmod/include/edmctrl.h
- /home/dprandle/Documents/code/ctrlmod/src/edmctrl.cpp

6.17 edmessage Struct Reference

`#include <edmessage.h>`

Inheritance diagram for edmessage:

Public Member Functions

- virtual ~edmessage ()
- virtual std::string type ()=0

Public Attributes

- uint32_t ref_count

6.17.1 Constructor & Destructor Documentation

6.17.1.1 virtual edmessage::~edmessage ( ) [inline], [virtual]

6.17.2 Member Function Documentation

6.17.2.1 virtual std::string edmessage::type ( ) [pure virtual]

Implemented in rplidar_firmware_message, rplidar_health_message, rplidar_info_message, rplidar_error_message, rplidar_scan_message, nav_system_request, rplidar_request, nav_message, and pulsed_light_message.

6.17.3 Member Data Documentation

6.17.3.1 uint32_t edmessage::ref_count

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

- /home/dprandle/Documents/code/ctrlmod/include/edmessage.h

6.18 edmessage_dispatch Class Reference

Class edmessage_dispatch.

`#include <edmessage_dispatch.h>`

Public Types

- typedef std::map< std::string, std::set< edsystem * > > listener_map
- typedef std::map< edsystem *, std::deque< edmessage * > > listener_queue

Public Member Functions

- edmessage_dispatch ()
- virtual ~edmessage_dispatch ()
- template<class MessageType >
  void register_listener (edsystem *sys)
- template<class MessageType >
  void unregister_listener (edsystem *sys)
- template<class MessageType >
  MessageType * push ()
- template<class MessageType >
  MessageType * push_front ()
- edmessage * next (edsystem *sys)
- void pop (edsystem *sys)
- void process_all (edsystem *sys)

6.18.1 Detailed Description

Class edmessage_dispatch.

A system can register its interest in certain message types, and any time a message of that type is created it will be added to that system's message queue. This queue is FIFO, and messages will not be deleted until they have been removed from every system's message queue.

Systems can process all messages in their queue by calling process_all(system*) where system* is a pointer to whatever system messages should be processed for (likely "this" pointer). Messages are processed by calling the

respective system's process function over and over until all messages in the system's message que are gone. If process returns false at any point, no more messages will be processed and process_all will return.

You can also process one message at a time by calling next to get the oldest message, and pop to remove that message.

6.18.2 Member Typedef Documentation

6.18.2.1 typedef std::map< std::string, std::set<edsystem*> > edmessage_dispatch::listener_map

This maps message type names to sets of systems. Any system that registers with a message type will be added to the system set corresponding to that message type.

6.18.2.2 typedef std::map<edsystem*, std::deque<edmessage*> > edmessage_dispatch::listener_queue

Listener queue holds a map of system pointers to deques of messages. This is FIFO setup - when a message is added to the queue it is appended to the back and when one is taken, it is taken from the front. This does not actually actually delete the message - the message is not deleted until it is no longer in any of the queues. A reference count is kept within the message itself.

6.18.3 Constructor & Destructor Documentation

6.18.3.1 edmessage_dispatch::edmessage_dispatch ( )

6.18.3.2 edmessage_dispatch::~edmessage_dispatch ( ) [virtual]

6.18.4 Member Function Documentation

6.18.4.1 edmessage * edmessage_dispatch::next (edsystem * *sys*)

6.18.4.2 void edmessage_dispatch::pop (edsystem * *sys*)

6.18.4.3 void edmessage_dispatch::process_all (edsystem * *sys*)

6.18.4.4 template<class MessageType > MessageType* edmessage_dispatch::push ( ) [inline]

6.18.4.5 template<class MessageType > MessageType* edmessage_dispatch::push_front ( ) [inline]

6.18.4.6 template<class MessageType > void edmessage_dispatch::register_listener (edsystem * *sys*) [inline]

6.18.4.7 template<class MessageType > void edmessage_dispatch::unregister_listener (edsystem * *sys*) [inline]

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

- /home/dprandle/Documents/code/ctrlmod/include/edmessage_dispatch.h
- /home/dprandle/Documents/code/ctrlmod/src/edmessage_dispatch.cpp

6.19 ednav_system Class Reference

`#include <ednavsystem.h>`

Inheritance diagram for ednav_system:

Public Member Functions

- ednav_system ()
- virtual ~ednav_system ()
- virtual void init ()
- virtual void release ()
- virtual bool process (edmessage *msg)
- virtual void update ()
- double interval ()
- void set_interval (double ms)
- virtual std::string typestr ()

Static Public Member Functions

- static std::string TypeString ()

Friends

- struct instruction_callback

6.19.1 Constructor & Destructor Documentation

6.19.1.1 ednav_system::ednav_system ( )

6.19.1.2 ednav_system::~ednav_system ( ) [virtual]

6.19.2 Member Function Documentation

6.19.2.1 void ednav_system::init ( ) [virtual]

Implements edsystem.

6.19.2.2 double ednav_system::interval ( )

6.19.2.3 bool ednav_system::process (edmessage * *msg*) [virtual]

Implements edsystem.

6.19.2.4 void ednav_system::release ( ) [virtual]

Implements edsystem.

6.19.2.5 void ednav_system::set_interval (double *ms*)

6.19.2.6 virtual std::string ednav_system::typestr ( ) [inline], [virtual]

Implements edsystem.

6.19.2.7 static std::string ednav_system::TypeString ( ) [inline], [static]

6.19.2.8 void ednav_system::update ( ) [virtual]

Implements edsystem.

6.19.3 Friends And Related Function Documentation

6.19.3.1 friend struct instruction_callback [friend]

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

- /home/dprandle/Documents/code/ctrlmod/include/ednavsystem.h
- /home/dprandle/Documents/code/ctrlmod/src/ednavsystem.cpp

6.20 edpid_controller< T > Class Template Reference

`#include <edpid_controller.h>`

Classes

- struct output_range

Public Member Functions

- edpid_controller ()
- ~edpid_controller ()
- void enable_complex_derivitive (bool enable)
- void enable_anti_reset_windup (bool enable)
- bool anti_reset_windup ()
- bool complex_derivitive ()
- const vec3 & gain ()
- double offset ()
- const output_range & range ()
- double ramp_limit ()
- void set_gain (const vec3 &pid_)
- void set_gain (double P, double I, double D)
- void set_gain_P (double P)
- void set_gain_I (double I)
- void set_gain_D (double D)
- void set_offset (double offset_)
- void set_ramp_limit (double percent)
- void set_range (const T &min, const T &max)
- void set_target (const T &target_)
- T loop (const T &input, double dt)
- const T & target ()

6.20.1 Constructor & Destructor Documentation

6.20.1.1 template<class T > edpid_controller< T >::edpid_controller ( )

6.20.1.2 template<class T > edpid_controller< T >::~edpid_controller ( )

6.20.2 Member Function Documentation

6.20.2.1 template<class T > bool edpid_controller< T >::anti_reset_windup ( )

6.20.2.2 template<class T > bool edpid_controller< T >::complex_derivitive ( )

6.20.2.3 template<class T > void edpid_controller< T >::enable_anti_reset_windup (bool *enable*)

6.20.2.4 template<class T > void edpid_controller< T >::enable_complex_derivitive (bool *enable*)

6.20.2.5 template<class T > const vec3 & edpid_controller< T >::gain ( )

6.20.2.6 template<class T> T edpid_controller< T >::loop (const T & *input,* double *dt*)

6.20.2.7 template<class T > double edpid_controller< T >::offset ( )

6.20.2.8 template<class T > double edpid_controller< T >::ramp_limit ( )

6.20.2.9 template<class T > const edpid_controller< T >::output_range & edpid_controller< T >::range ( )

6.20.2.10 template<class T > void edpid_controller< T >::set_gain (const vec3 & *pid_*)

6.20.2.11 template<class T > void edpid_controller< T >::set_gain (double *P,* double *I,* double *D*)

6.20.2.12 template<class T > void edpid_controller< T >::set_gain_D (double *D*)

6.20.2.13 template<class T > void edpid_controller< T >::set_gain_I (double *I*)

6.20.2.14 template<class T > void edpid_controller< T >::set_gain_P (double *P*)

6.20.2.15 template<class T > void edpid_controller< T >::set_offset (double *offset_*)

6.20.2.16 template<class T > void edpid_controller< T >::set_ramp_limit (double *percent*)

6.20.2.17 template<class T> void edpid_controller< T >::set_range (const T & *min,* const T & *max*)

6.20.2.18 template<class T> void edpid_controller< T >::set_target (const T & *target_*)

6.20.2.19 template<class T > const T & edpid_controller< T >::target ( )

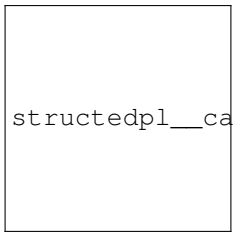
The documentation for this class was generated from the following file:

- /home/dprandle/Documents/code/ctrlmod/include/edpid_controller.h

6.21 edpl_callback Struct Reference

`#include <edplsystem.h>`

Inheritance diagram for edpl_callback:



Public Member Functions

- edpl_callback (edpl_system::pl_gpio *ceil, edpl_system::pl_gpio *floor)
- void exec ()

Public Attributes

- edpl_system::pl_gpio * pl_ceil
- edpl_system::pl_gpio * pl_floor

6.21.1 Constructor & Destructor Documentation

6.21.1.1 edpl_callback::edpl_callback (edpl_system::pl_gpio * *ceil*, edpl_system::pl_gpio * *floor*) [inline]

6.21.2 Member Function Documentation

6.21.2.1 void edpl_callback::exec ( ) [virtual]

Implements edcallback.

6.21.3 Member Data Documentation

6.21.3.1 edpl_system::pl_gpio* edpl_callback::pl_ceil

6.21.3.2 edpl_system::pl_gpio* edpl_callback::pl_floor

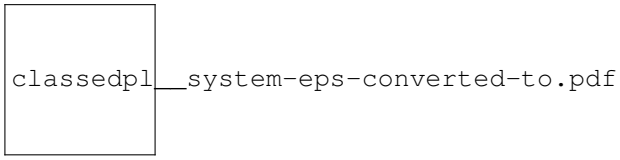
The documentation for this struct was generated from the following files:

- /home/dprandle/Documents/code/ctrlmod/include/edplsystem.h
- /home/dprandle/Documents/code/ctrlmod/src/edplsystem.cpp

6.22 edpl_system Class Reference

`#include <edplsystem.h>`

Inheritance diagram for edpl_system:



Classes

- struct pl_gpio

Public Types

- typedef std::map< uint32_t, pl_gpio * > plmap

Public Member Functions

- edpl_system ()
- virtual ~edpl_system ()
- pl_gpio * add_pl (uint32_t mraa_pin, double c_offset=0.0, const vec3 &pos_offset=vec3(), const quat &orient_offset=quat())
- pl_gpio * get_pl (uint32_t mraa_pin)
- void rm_pl (uint32_t mraa_pin)
- bool pl_pin_taken (uint32_t mraa_pin)
- void pl_set_pos (uint32_t mraa_pin, const vec3 &pos_)
- void pl_set_orientation (uint32_t mraa_pin, const quat &orient_)
- void pl_set_cal_offset (uint32_t mraa_pin, double offset)
- virtual void init ()
- virtual void release ()
- virtual bool process (edmessage *msg)
- virtual void update ()
- virtual std::string typestr ()

Static Public Member Functions

- static std::string TypeString ()

6.22.1 Member Typedef Documentation

6.22.1.1 typedef std::map<uint32_t, pl_gpio*> edpl_system::plmap

6.22.2 Constructor & Destructor Documentation

6.22.2.1 edpl_system::edpl_system ( )

6.22.2.2 edpl_system::~edpl_system ( ) [virtual]

6.22.3 Member Function Documentation

6.22.3.1 edpl_system::pl_gpio * edpl_system::add_pl (uint32_t *mraa_pin*, double *c_offset* = 0.0, const vec3 & *pos_offset* = vec3(), const quat & *orient_offset* = quat())

6.22.3.2 edpl_system::pl_gpio * edpl_system::get_pl (uint32_t *mraa_pin*)

6.22.3.3 void edpl_system::init ( ) [virtual]

Implements edsystem.

6.22.3.4 bool edpl_system::pl_pin_taken (uint32_t *mraa_pin*)

6.22.3.5 void edpl_system::pl_set_cal_offset (uint32_t *mraa_pin,* double *offset*)

6.22.3.6 void edpl_system::pl_set_orientation (uint32_t *mraa_pin,* const quat & *orient_*)

6.22.3.7 void edpl_system::pl_set_pos (uint32_t *mraa_pin,* const vec3 & *pos_*)

6.22.3.8 bool edpl_system::process (edmessage * *msg*) [virtual]

Implements edsystem.

6.22.3.9 void edpl_system::release ( ) [virtual]

Implements edsystem.

6.22.3.10 void edpl_system::rm_pl (uint32_t *mraa_pin*)

6.22.3.11 std::string edpl_system::typestr ( ) [virtual]

Implements edsystem.

6.22.3.12 static std::string edpl_system::TypeString ( ) [inline], [static]

6.22.3.13 void edpl_system::update ( ) [virtual]

Implements edsystem.

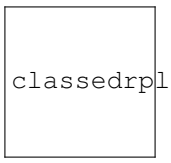
The documentation for this class was generated from the following files:

- /home/dprandle/Documents/code/ctrlmod/include/edplsystem.h
- /home/dprandle/Documents/code/ctrlmod/src/edplsystem.cpp

6.23 edrplidar_system Class Reference

`#include <edrplidar_system.h>`

Inheritance diagram for edrplidar_system:



Public Types

- enum ExchangeType {
  Scan, Info, Health, Reset,
  None }

Public Member Functions

- edrplidar_system ()
- ~edrplidar_system ()
- void init ()
- void release ()
- bool process (edmessage *msg)
- void update ()
- std::string typestr ()

Static Public Member Functions

- static std::string TypeString ()

Protected Member Functions

- bool startScan ()
- bool forceScan ()
- bool stopScan ()
- bool reset ()
- bool requestInfo ()
- bool requestHealth ()

6.23.1 Member Enumeration Documentation

6.23.1.1 enum edrplidar_system::ExchangeType

Enumerator

Scan

Info

Health

Reset

None

6.23.2 Constructor & Destructor Documentation

6.23.2.1 edrplidar_system::edrplidar_system ( )

6.23.2.2 edrplidar_system::~edrplidar_system ( )

6.23.3 Member Function Documentation

6.23.3.1 bool edrplidar_system::forceScan ( ) [protected]

6.23.3.2 void edrplidar_system::init ( ) [virtual]

Implements edsystem.

6.23.3.3 bool edrplidar_system::process (edmessage * *msg*) [virtual]

Implements edsystem.

6.23.3.4 void edrplidar_system::release ( ) [virtual]

Implements edsystem.

6.23.3.5 bool edrplidar_system::requestHealth ( ) [protected]

6.23.3.6 bool edrplidar_system::requestInfo ( ) [protected]

6.23.3.7 bool edrplidar_system::reset ( ) [protected]

6.23.3.8 bool edrplidar_system::startScan ( ) [protected]

6.23.3.9 bool edrplidar_system::stopScan ( ) [protected]

6.23.3.10 std::string edrplidar_system::typestr ( ) [inline], [virtual]

Implements edsystem.

6.23.3.11 static std::string edrplidar_system::TypeString ( ) [inline], [static]

6.23.3.12 void edrplidar_system::update ( ) [virtual]

Implements edsystem.

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

- /home/dprandle/Documents/code/ctrlmod/include/edrplidar_system.h
- /home/dprandle/Documents/code/ctrlmod/src/edrplidar_system.cpp

6.24 edsocket Class Reference

`#include <edsocket.h>`

Inheritance diagram for edsocket:

Public Member Functions

- edsocket (uint32_t socket_fd)
- ~edsocket ()

Additional Inherited Members

6.24.1 Constructor & Destructor Documentation

6.24.1.1 edsocket::edsocket (uint32_t *socket_fd*)

6.24.1.2 edsocket::~edsocket ( )

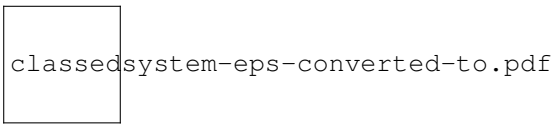
The documentation for this class was generated from the following files:

- /home/dprandle/Documents/code/ctrlmod/include/edsocket.h
- /home/dprandle/Documents/code/ctrlmod/src/edsocket.cpp

6.25 edsystem Class Reference

`#include <edsystem.h>`

Inheritance diagram for edsystem:



Public Member Functions

- edsystem ()
- virtual ~edsystem ()
- virtual void init ()=0
- virtual void release ()=0
- virtual bool process (edmessage *msg)=0
- virtual void update ()=0
- virtual std::string typestr ()=0

6.25.1 Constructor & Destructor Documentation

6.25.1.1 edsystem::edsystem ( ) [inline]

6.25.1.2 virtual edsystem::~edsystem ( ) [inline], [virtual]

6.25.2 Member Function Documentation

6.25.2.1 virtual void edsystem::init ( ) [pure virtual]

Implemented in edimu_system, edpl_system, ednav_system, edcomm_system, edrplidar_system, and edlogging_system.

6.25.2.2 virtual bool edsystem::process (edmessage * *msg*) [pure virtual]

Implemented in edimu_system, edpl_system, ednav_system, edcomm_system, edrplidar_system, and edlogging_system.

6.25.2.3 virtual void edsystem::release ( ) [pure virtual]

Implemented in edimu_system, edpl_system, ednav_system, edcomm_system, edrplidar_system, and edlogging_system.

6.25.2.4 virtual std::string edsystem::typestr ( ) [pure virtual]

Implemented in edimu_system, edpl_system, edcomm_system, ednav_system, edrplidar_system, and edlogging_system.

6.25.2.5 virtual void edsystem::update ( ) [pure virtual]

Implemented in edimu_system, edpl_system, ednav_system, edcomm_system, edrplidar_system, and edlogging_system.

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

- /home/dprandle/Documents/code/ctrlmod/include/edsystem.h

6.26 edthreaded_fd Class Reference

`#include <edthreaded_fd.h>`

Inheritance diagram for edthreaded_fd:

Classes

- struct Error
- struct WriteVal

Public Types

- enum ErrorVal {
  NoError, ConnectionClosed, DataOverwrite, InvalidRead,
  InvalidWrite, ThreadCreation, OpenFileDescriptor, Configuration,
  AlreadyRunning, FDAlreadyOpen, CommandNoResponse }

Public Member Functions

- edthreaded_fd (uint32_t readbuf_size=DEFAULT_FD_READ_BUFFER_SIZE, uint32_t writebuf_size=DEFAULT_FD_WRITE_BUFFER_SIZE)
- virtual ~edthreaded_fd ()
- virtual uint32_t read (uint8_t *buffer, uint32_t max_size)
- virtual uint32_t write (uint8_t *buffer, uint32_t size, int32_t response_size=0)
- virtual Error error ()
- bool running ()
- virtual bool start ()
- int32_t fd ()
- bool set_fd (int32_t fd_)
- virtual void stop ()

Protected Member Functions

- virtual int32_t _raw_read (uint8_t *buffer, uint32_t max_size)=0
- virtual int32_t _raw_write (uint8_t *buffer, uint32_t max_size)=0
- virtual void _do_read ()
- virtual void _do_write ()
- virtual void _exec ()
- void _setError (ErrorVal err_val, int32_t _errno)

Static Protected Member Functions

- static void * thread_exec (void *)

Protected Attributes

- int32_t m_fd
- uint32_t m_read_rawindex
- uint32_t m_read_curindex
- uint32_t m_write_rawindex
- uint32_t m_write_curindex
- std::vector< WriteVal > m_write_buffer
- std::vector< uint8_t > m_read_buffer
- Error m_err
- bool m_running
- uint32_t m_current_wait_for_byte_count
- edtimer * m_wait_timer
- pthread_mutex_t m_send_lock
- pthread_mutex_t m_recv_lock
- pthread_mutex_t m_error_lock
- pthread_mutex_t m_running_lock
- pthread_t m_thread

Friends

- struct command_wait_callback

6.26.1 Member Enumeration Documentation

6.26.1.1 enum edthreaded_fd::ErrorVal

Enumerator

NoError

ConnectionClosed

DataOverwrite

InvalidRead

InvalidWrite

ThreadCreation

OpenFileDescriptor

Configuration

AlreadyRunning

FDAlreadyOpen

CommandNoResponse

6.26.2 Constructor & Destructor Documentation

6.26.2.1 edthreaded_fd::edthreaded_fd (uint32_t *readbuf_size* = DEFAULT_FD_READ_BUFFER_SIZE, uint32_t *writebuf_size* = DEFAULT_FD_WRITE_BUFFER_SIZE)

6.26.2.2 edthreaded_fd::~edthreaded_fd ( ) [virtual]

6.26.3 Member Function Documentation

6.26.3.1 void edthreaded_fd::_do_read ( ) [protected], [virtual]

6.26.3.2 void edthreaded_fd::_do_write ( ) [protected], [virtual]

6.26.3.3 void edthreaded_fd::_exec ( ) [protected], [virtual]

6.26.3.4 virtual int32_t edthreaded_fd::_raw_read (uint8_t * *buffer*, uint32_t *max_size*) [protected], [pure virtual]

6.26.3.5 virtual int32_t edthreaded_fd::_raw_write (uint8_t * *buffer*, uint32_t *max_size*) [protected], [pure virtual]

6.26.3.6 void edthreaded_fd::_setError (ErrorVal *err_val*, int32_t *_errno*) [protected]

6.26.3.7 edthreaded_fd::Error edthreaded_fd::error ( ) [virtual]

6.26.3.8 int32_t edthreaded_fd::fd ( )

6.26.3.9 uint32_t edthreaded_fd::read (uint8_t * *buffer*, uint32_t *max_size*) [virtual]

6.26.3.10 bool edthreaded_fd::running ( )

6.26.3.11 bool edthreaded_fd::set_fd (int32_t *fd_*)

6.26.3.12 bool edthreaded_fd::start ( ) [virtual]

Reimplemented in edi2c, and eduart.

6.26.3.13 void edthreaded_fd::stop ( ) [virtual]

6.26.3.14 void * edthreaded_fd::thread_exec (void * *_this*) [static], [protected]

6.26.3.15 uint32_t edthreaded_fd::write (uint8_t * *buffer*, uint32_t *size*, int32_t *response_size* = 0) [virtual]

6.26.4 Friends And Related Function Documentation

6.26.4.1 friend struct command_wait_callback [friend]

6.26.5 Member Data Documentation

6.26.5.1 uint32_t edthreaded_fd::m_current_wait_for_byte_count [protected]

6.26.5.2 Error edthreaded_fd::m_err [protected]

6.26.5.3 pthread_mutex_t edthreaded_fd::m_error_lock [protected]

6.26.5.4 int32_t edthreaded_fd::m_fd [protected]

6.26.5.5 std::vector<uint8_t> edthreaded_fd::m_read_buffer [protected]

6.26.5.6 uint32_t edthreaded_fd::m_read_curindex [protected]

6.26.5.7 uint32_t edthreaded_fd::m_read_rawindex [protected]

6.26.5.8 pthread_mutex_t edthreaded_fd::m_recv_lock [protected]

6.26.5.9 bool edthreaded_fd::m_running [protected]

6.26.5.10 pthread_mutex_t edthreaded_fd::m_running_lock [protected]

6.26.5.11 pthread_mutex_t edthreaded_fd::m_send_lock [protected]

6.26.5.12 pthread_t edthreaded_fd::m_thread [protected]

6.26.5.13 edtimer* edthreaded_fd::m_wait_timer [protected]

6.26.5.14 std::vector<WriteVal> edthreaded_fd::m_write_buffer [protected]

6.26.5.15 uint32_t edthreaded_fd::m_write_curindex [protected]

6.26.5.16 uint32_t edthreaded_fd::m_write_rawindex [protected]

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

- /home/dprandle/Documents/code/ctrlmod/include/edthreaded_fd.h
- /home/dprandle/Documents/code/ctrlmod/src/edthreaded_fd.cpp

6.27 edtimer Class Reference

class edtimer

`#include <edtimer.h>`

Public Types

- enum cb_mode { no_shot, single_shot, continous_shot }

Public Member Functions

- edtimer ()
- ~edtimer ()
- void start ()
- void update ()
- edtimer_callback * callback ()
- cb_mode callback_mode ()
- double callback_delay ()
- void cont ()
- void stop ()
- void set_callback (edtimer_callback *cb)
- void set_callback_mode (cb_mode mode)

- void set_callback_delay (double ms)
- double dt ()
- bool running ()
- double elapsed ()

6.27.1 Detailed Description

class edtimer

This class keeps track of time allowing you to start, stop, and continue the timer. It also has a "dt" functionality to allow you to see how much time has elapsed since the last update call. This could be useful for various things.

You can also set up the timer to execute a callback every so often (every so many milliseconds) or you can set it to execute once after some delay.

6.27.2 Member Enumeration Documentation

6.27.2.1 enum edtimer::cb_mode

Enumerator

no_shot

single_shot

continous_shot

6.27.3 Constructor & Destructor Documentation

6.27.3.1 edtimer::edtimer ( )

6.27.3.2 edtimer::~edtimer ( )

6.27.4 Member Function Documentation

6.27.4.1 edtimer_callback * edtimer::callback ( )

6.27.4.2 double edtimer::callback_delay ( )

6.27.4.3 edtimer::cb_mode edtimer::callback_mode ( )

6.27.4.4 void edtimer::cont ( )

6.27.4.5 double edtimer::dt ( )

6.27.4.6 double edtimer::elapsed ( )

6.27.4.7 bool edtimer::running ( )

6.27.4.8 void edtimer::set_callback (edtimer_callback * *cb*)

6.27.4.9 void edtimer::set_callback_delay (double *ms*)

6.27.4.10 void edtimer::set_callback_mode (cb_mode *mode*)

6.27.4.11 void edtimer::start ( )

6.27.4.12 void edtimer::stop ( )

6.27.4.13 void edtimer::update ( )

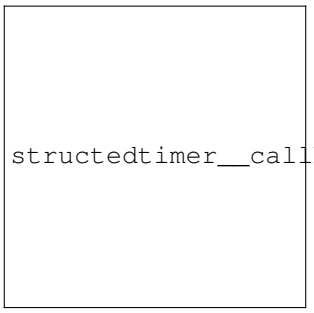
The documentation for this class was generated from the following files:

- /home/dprandle/Documents/code/ctrlmod/include/edtimer.h
- /home/dprandle/Documents/code/ctrlmod/src/edtimer.cpp

6.28 edtimer_callback Struct Reference

`#include <edcallback.h>`

Inheritance diagram for edtimer_callback:



Public Attributes

- edtimer * timer

Additional Inherited Members

6.28.1 Member Data Documentation

6.28.1.1 edtimer* edtimer_callback::timer

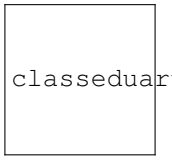
The documentation for this struct was generated from the following file:

- /home/dprandle/Documents/code/ctrlmod/include/edcallback.h

6.29 eduart Class Reference

`#include <eduart.h>`

Inheritance diagram for eduart:



Classes

- struct DataFormat

Public Types

- enum SerialPort { Uart1, Uart2 }
- enum BaudRate {
  b50 =B50, b75 =B75, b110 =B110, b134 =B134,
  b150 =B150, b200 =B200, b300 =B300, b600 =B600,
  b1200 =B1200, b1800 =B1800, b2400 =B2400, b4800 =B4800,
  b9600 =B9600, b19200 =B19200, b38400 =B38400, b57600 =B57600,
  b115200 =B115200, b230400 =B230400, b460800 =B460800, b500000 =B500000,
  b576000 =B576000, b921600 =B921600, b1000000 =B1000000, b1152000 =B1152000,
  b1500000 =B1500000, b2000000 =B2000000, b2500000 =B2500000, b3000000 =B3000000,
  b3500000 =B3500000, b4000000 =B4000000 }
- enum Parity { None = 0, Odd = (PARENB | PARODD), Even = (PARENB) }
- enum StopBits { One = 0, Two = CSTOPB }
- enum DataBits { d5 =CS5, d6 =CS6, d7 =CS7, d8 =CS8 }

Public Member Functions

- eduart (SerialPort uart_num)
- ~eduart ()
- const std::string & device_path ()
- void set_baud (BaudRate baud)
- BaudRate baud ()
- bool start ()
- void set_format (DataBits db, Parity p, StopBits sb)
- void set_format (const DataFormat &data_format)
- const DataFormat & format ()

Additional Inherited Members

6.29.1 Member Enumeration Documentation

6.29.1.1 enum eduart::BaudRate

Enumerator

b50

b75

b110

b134

b150

b200

b300

b600

b1200

b1800

b2400

b4800

b9600
b19200
b38400
b57600
b115200
b230400
b460800
b500000
b576000
b921600
b1000000
b1152000
b1500000
b2000000
b2500000
b3000000
b3500000
b4000000

6.29.1.2 enum eduart::DataBits

Enumerator

d5
d6
d7
d8

6.29.1.3 enum eduart::Parity

Enumerator

None
Odd
Even

6.29.1.4 enum eduart::SerialPort

Enumerator

Uart1
Uart2

6.29.1.5 enum eduart::StopBits

Enumerator

One
Two

6.29.2 Constructor & Destructor Documentation

6.29.2.1 eduart::eduart (SerialPort *uart_num*)

6.29.2.2 eduart::~eduart ( )

6.29.3 Member Function Documentation

6.29.3.1 eduart::BaudRate eduart::baud ( )

6.29.3.2 const std::string & eduart::device_path ( )

6.29.3.3 const eduart::DataFormat & eduart::format ( )

6.29.3.4 void eduart::set_baud (BaudRate *baud*)

6.29.3.5 void eduart::set_format (DataBits *db*, Parity *p*, StopBits *sb*)

6.29.3.6 void eduart::set_format (const DataFormat & *data_format*)

6.29.3.7 bool eduart::start ( ) [virtual]

Reimplemented from edthreaded_fd.

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

- /home/dprandle/Documents/code/ctrlmod/include/eduart.h
- /home/dprandle/Documents/code/ctrlmod/src/eduart.cpp

6.30 edthreaded_fd::Error Struct Reference

`#include <edthreaded_fd.h>`

Public Member Functions

- Error ()

Public Attributes

- ErrorVal err_val
- int32_t _errno

6.30.1 Constructor & Destructor Documentation

6.30.1.1 edthreaded_fd::Error::Error ( ) [inline]

6.30.2 Member Data Documentation

6.30.2.1 int32_t edthreaded_fd::Error::_errno

6.30.2.2 ErrorVal edthreaded_fd::Error::err_val

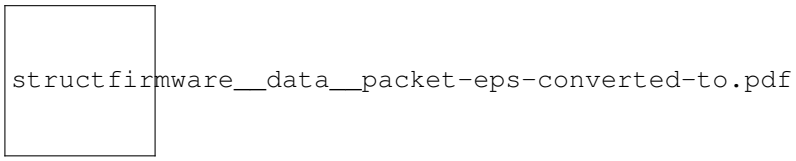
The documentation for this struct was generated from the following file:

- /home/dprandle/Documents/code/ctrlmod/include/edthreaded_fd.h

6.31 firmware_data_packet Struct Reference

`#include <edrplidar_packets.h>`

Inheritance diagram for firmware_data_packet:



Public Member Functions

- firmware_data_packet ()
- virtual std::string toString ()
- virtual std::string type ()
- virtual uint32_t size ()
- virtual uint8_t & operator[] (uint32_t index)
- virtual uint8_t * dataptr ()

Static Public Member Functions

- static std::string Type ()
- static uint32_t Size ()

Public Attributes

- union {
  struct {
  uint8_t line1 [18]
  uint8_t line2 [29]
  uint8_t line3 [9]
  }
  uint8_t data [56]
  };

6.31.1 Constructor & Destructor Documentation

6.31.1.1 firmware_data_packet::firmware_data_packet ( )

6.31.2 Member Function Documentation

6.31.2.1 virtual uint8_t* firmware_data_packet::dataptr ( ) [inline], [virtual]

Implements data_packet.

6.31.2.2 virtual uint8_t& firmware_data_packet::operator[] (uint32_t *index*) [inline], [virtual]

Implements data_packet.

6.31.2.3 virtual uint32_t firmware_data_packet::size ( ) [inline], [virtual]

Implements data_packet.

6.31.2.4 static uint32_t firmware_data_packet::Size ( ) [inline], [static]

6.31.2.5 std::string firmware_data_packet::toString ( ) [virtual]

Implements data_packet.

6.31.2.6 virtual std::string firmware_data_packet::type ( ) [inline], [virtual]

Implements data_packet.

6.31.2.7 static std::string firmware_data_packet::Type ( ) [inline], [static]

6.31.3 Member Data Documentation

6.31.3.1 union { ... }

6.31.3.2 uint8_t firmware_data_packet::data[56]

6.31.3.3 uint8_t firmware_data_packet::line1[18]

6.31.3.4 uint8_t firmware_data_packet::line2[29]

6.31.3.5 uint8_t firmware_data_packet::line3[9]

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

- /home/dprandle/Documents/code/ctrlmod/include/edrplidar_packets.h
- /home/dprandle/Documents/code/ctrlmod/src/edrplidar_packets.cpp

6.32 force_scan_request Struct Reference

`#include <edrplidar_packets.h>`

Inheritance diagram for force_scan_request:

Public Member Functions

- force_scan_request ()

Additional Inherited Members

6.32.1 Constructor & Destructor Documentation

6.32.1.1 force_scan_request::force_scan_request ( ) [inline]

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

- /home/dprandle/Documents/code/ctrlmod/include/edrplidar_packets.h

6.33 health_data_packet Struct Reference

`#include <edrplidar_packets.h>`

Inheritance diagram for health_data_packet:

Public Member Functions

- health_data_packet ()
- virtual std::string toString ()
- virtual std::string type ()
- virtual uint32_t size ()
- virtual uint8_t & operator[] (uint32_t index)
- virtual uint8_t * dataptr ()

Static Public Member Functions

- static std::string Type ()
- static uint32_t Size ()

Public Attributes

- union {
  struct {
  uint8_t status
  uint8_t error_code7to0
  uint8_t error_code15to8
  }
  uint8_t data [3]
  };

6.33.1 Constructor & Destructor Documentation

6.33.1.1 health_data_packet::health_data_packet ( )

6.33.2 Member Function Documentation

6.33.2.1 virtual uint8_t* health_data_packet::dataptr ( ) [inline], [virtual]

Implements data_packet.

6.33.2.2 virtual uint8_t& health_data_packet::operator[] (uint32_t *index*) [inline], [virtual]

Implements data_packet.

6.33.2.3 virtual uint32_t health_data_packet::size ( ) [inline], [virtual]

Implements data_packet.

6.33.2.4 static uint32_t health_data_packet::Size ( ) [inline], [static]

6.33.2.5 std::string health_data_packet::toString ( ) [virtual]

Implements data_packet.

6.33.2.6 virtual std::string health_data_packet::type ( ) [inline], [virtual]

Implements data_packet.

6.33.2.7 static std::string health_data_packet::Type ( ) [inline], [static]

6.33.3 Member Data Documentation

6.33.3.1 union { ... }

6.33.3.2 uint8_t health_data_packet::data[3]

6.33.3.3 uint8_t health_data_packet::error_code15to8

6.33.3.4 uint8_t health_data_packet::error_code7to0

6.33.3.5 uint8_t health_data_packet::status

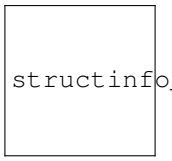
The documentation for this struct was generated from the following files:

- /home/dprandle/Documents/code/ctrlmod/include/edrplidar_packets.h
- /home/dprandle/Documents/code/ctrlmod/src/edrplidar_packets.cpp

6.34 info_data_packet Struct Reference

`#include <edrplidar_packets.h>`

Inheritance diagram for info_data_packet:



Public Member Functions

- info_data_packet ()
- virtual std::string toString ()
- virtual std::string type ()
- virtual uint32_t size ()
- virtual uint8_t & operator[] (uint32_t index)
- virtual uint8_t * dataptr ()

Static Public Member Functions

- static std::string Type ()
- static uint32_t Size ()

Public Attributes

- union {
  struct {
  uint8_t model
  uint8_t firmware_minor
  uint8_t firmware_major
  uint8_t hardware
  uint8_t serialnumber [16]
  }
  uint8_t data [20]
  };

6.34.1 Constructor & Destructor Documentation

6.34.1.1 info_data_packet::info_data_packet ( )

6.34.2 Member Function Documentation

6.34.2.1 virtual uint8_t* info_data_packet::dataptr ( ) [inline], [virtual]

Implements data_packet.

6.34.2.2 virtual uint8_t& info_data_packet::operator[] (uint32_t *index*) [inline], [virtual]

Implements data_packet.

6.34.2.3 virtual uint32_t info_data_packet::size ( ) [inline], [virtual]

Implements data_packet.

6.34.2.4 static uint32_t info_data_packet::Size ( ) [inline], [static]

6.34.2.5 std::string info_data_packet::toString ( ) [virtual]

Implements data_packet.

6.34.2.6 virtual std::string info_data_packet::type ( ) [inline], [virtual]

Implements data_packet.

6.34.2.7 static std::string info_data_packet::Type ( ) [inline], [static]

6.34.3 Member Data Documentation

6.34.3.1 union { ... }

6.34.3.2 uint8_t info_data_packet::data[20]

6.34.3.3 uint8_t info_data_packet::firmware_major

6.34.3.4 uint8_t info_data_packet::firmware_minor

6.34.3.5 uint8_t info_data_packet::hardware

6.34.3.6 uint8_t info_data_packet::model

6.34.3.7 uint8_t info_data_packet::serialnumber[16]

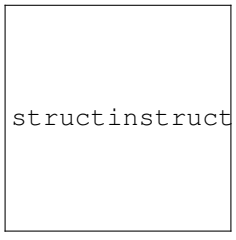
The documentation for this struct was generated from the following files:

- /home/dprandle/Documents/code/ctrlmod/include/edrplidar_packets.h
- /home/dprandle/Documents/code/ctrlmod/src/edrplidar_packets.cpp

6.35 instruction_callback Struct Reference

`#include <ednavsystem.h>`

Inheritance diagram for instruction_callback:



Public Member Functions

- instruction_callback (ednav_system *system)
- void exec ()

Public Attributes

- ednav_system * m_nav_sys

6.35.1 Constructor & Destructor Documentation

6.35.1.1 instruction_callback::instruction_callback (ednav_system * *system*) [inline]

6.35.2 Member Function Documentation

6.35.2.1 void instruction_callback::exec ( ) [virtual]

Implements edcallback.

6.35.3 Member Data Documentation

6.35.3.1 ednav_system* instruction_callback::m_nav_sys

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

- /home/dprandle/Documents/code/ctrlmod/include/ednavsystem.h
- /home/dprandle/Documents/code/ctrlmod/src/ednavsystem.cpp

6.36 nav_message Struct Reference

```
#include <edmessage.h>
```

Inheritance diagram for nav_message:

Public Member Functions

- uint32_t size ()
- std::string type ()

Static Public Member Functions

- static std::string Type ()

Public Attributes

- union {
  struct {
  int16_t throttle
  int16_t pitch
  int16_t roll

```
    int16_t yaw
    double rvec_raw [2]
    double rvec_corrected [2]
  }
  uint8_t data [40]
};
```

6.36.1 Member Function Documentation

6.36.1.1 uint32_t nav_message::size ( ) [inline]

6.36.1.2 std::string nav_message::type ( ) [inline], [virtual]

Implements edmessage.

6.36.1.3 static std::string nav_message::Type ( ) [inline], [static]

6.36.2 Member Data Documentation

6.36.2.1 union { ... }

6.36.2.2 uint8_t nav_message::data[40]

6.36.2.3 int16_t nav_message::pitch

6.36.2.4 int16_t nav_message::roll

6.36.2.5 double nav_message::rvec_corrected[2]

6.36.2.6 double nav_message::rvec_raw[2]

6.36.2.7 int16_t nav_message::throttle

6.36.2.8 int16_t nav_message::yaw

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

- /home/dprandle/Documents/code/ctrlmod/include/edmessage.h

6.37 nav_system_request Struct Reference

```
#include <edmessage.h>
```

Inheritance diagram for nav_system_request:

Public Member Functions

- virtual std::string type ()

Static Public Member Functions

- static std::string Type ()

Public Attributes

- vec3 pid
- double ramp_limit
- vec2 bias_vec
- double g_factor
- double bias_threshold_dist
- bool complex_der
- bool anti_reset_winding
- bool threshold_dropout

6.37.1 Member Function Documentation

6.37.1.1 virtual std::string nav_system_request::type ( ) [inline], [virtual]

Implements edmessage.

6.37.1.2 static std::string nav_system_request::Type ( ) [inline], [static]

6.37.2 Member Data Documentation

6.37.2.1 bool nav_system_request::anti_reset_winding

6.37.2.2 double nav_system_request::bias_threshold_dist

6.37.2.3 vec2 nav_system_request::bias_vec

6.37.2.4 bool nav_system_request::complex_der

6.37.2.5 double nav_system_request::g_factor

6.37.2.6 vec3 nav_system_request::pid

6.37.2.7 double nav_system_request::ramp_limit

6.37.2.8 bool nav_system_request::threshold_dropout

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

- /home/dprandle/Documents/code/ctrlmod/include/edmessage.h

6.38 NSBoundingBox Struct Reference

```
#include <nsmath.h>
```

Public Types

- enum Face {
  None, Bottom, Top, Left,
  Right, Back, Front }

Public Member Functions

- NSBoundingBox (const std::vector< fvec3 > &pVertices=std::vector< fvec3 >())
- void calculate (const std::vector< fvec3 > &pVertices, const fmat4 &pTransform=fmat4())
- fvec3 center (const Face &pFace=None)
- void clear ()
- float dx ()
- float dy ()
- float dz ()
- void extend (const std::vector< fvec3 > &pVertices, const fmat4 &pTransform=fmat4())
- void set (const fvec3 &pMin, const fvec3 pMax)
- float volume ()

Public Attributes

- fvec3 mMin
- fvec3 mMax
- fvec3 mVerts [8]

6.38.1 Member Enumeration Documentation

6.38.1.1 enum NSBoundingBox::Face

Enumerator

None

Bottom

Top

Left

Right

Back

Front

6.38.2 Constructor & Destructor Documentation

6.38.2.1 NSBoundingBox::NSBoundingBox (const std::vector< fvec3 > & *pVertices* = `std::vector<fvec3>()`)

Calculates the box given a set of vertices.. if no vertices are given then will set everything to zero.

6.38.3 Member Function Documentation

6.38.3.1 void NSBoundingBox::calculate (const std::vector< fvec3 > & *pVertices*, const fmat4 & *pTransform* = fmat4 `()`)

Find the min and max of a set of vertices and use that to make the bounding box.

6.38.3.2 fvec3 NSBoundingBox::center (const Face & *pFace* = None)

Returns the center of the box or the center of a face of the box if pFace is specified as something other than None.

6.38.3.3 void NSBoundingBox::clear ( )

Clears the verts and the min/max to 0.

6.38.3.4 float NSBoundingBox::dx ( )

Length of box in x direction.

6.38.3.5 float NSBoundingBox::dy ( )

Length of box in y direction.

6.38.3.6 float NSBoundingBox::dz ( )

Length of box in z direction.

6.38.3.7 void NSBoundingBox::extend (const std::vector< fvec3 > & *pVertices,* const fmat4 & *pTransform* = fmat4 ())

6.38.3.8 void NSBoundingBox::set (const fvec3 & *pMin,* const fvec3 *pMax*)

Set the min and max - will update the verts based on this new min and max.

6.38.3.9 float NSBoundingBox::volume ( )

The volume in whatever units the world is represented in. The cartesian coordinate x = 1, y = 1, z = 1 would represent a point that is 1 unit away from each axis and 1.41 units away from the origin.

6.38.4 Member Data Documentation

6.38.4.1 fvec3 NSBoundingBox::mMax

6.38.4.2 fvec3 NSBoundingBox::mMin

6.38.4.3 fvec3 NSBoundingBox::mVerts[8]

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

- /home/dprandle/Documents/code/ctrlmod/include/nsmath.h
- /home/dprandle/Documents/code/ctrlmod/src/nsmath.cpp

6.39 nsmat2< T > Struct Template Reference

`#include <nsmat2.h>`

Public Member Functions

- nsmat2 ()
- nsmat2 (const T &val)
- nsmat2 (const nsmat2 &copy)
- nsmat2 (const T &a, const T &b, const T &c, const T &d)
- nsmat2 (const NSVec2< T > &row1, const NSVec2< T > &row2)
- T * dataPtr ()
- T determinant () const
- nsmat2< T > & invert ()
- nsmat2< T > & rotationFrom (T angle, bool rads=false)
- nsmat2< T > & roundToZero ()
- nsmat2< T > & scalingFrom (const NSVec2< T > &scale)
- nsmat2< T > & scalingFrom (const nsmat2< T > &transform2d)
- nsmat2< T > & scalingFrom (const nsmat3< T > &transform2d)
- nsmat2< T > & set (const T &val)
- nsmat2< T > & set (const T &a, const T &b, const T &c, const T &d)
- nsmat2< T > & set (const NSVec2< T > &row1, const NSVec2< T > &row2)
- nsmat2< T > & setColumn (const uint32_t &i, const T &x, const T &y)
- nsmat2< T > & setColumn (const uint32_t &i, const NSVec2< T > &col)
- nsmat2< T > & setIdentity ()
- nsmat2< T > & transpose ()
- std::string toString (bool newline=true) const
- nsmat2< T > operator* (const nsmat2< T > &rhs) const
- nsmat2< T > operator/ (const nsmat2< T > &rhs) const
- nsmat2< T > operator% (const nsmat2< T > &rhs) const
- NSVec2< T > operator* (const NSVec2< T > &rhs) const
- nsmat2< T > operator% (const NSVec2< T > &rhs) const
- nsmat2< T > operator/ (const NSVec2< T > &rhs) const
- nsmat2< T > operator* (const T &rhs) const
- nsmat2< T > operator/ (const T &rhs) const
- nsmat2< T > operator+ (const nsmat2< T > &rhs) const
- nsmat2< T > operator- (const nsmat2< T > &rhs) const
- bool operator== (const nsmat2< T > &rhs) const
- bool operator!= (const nsmat2< T > &rhs) const
- nsmat2< T > & operator= (const nsmat2< T > &rhs)
- nsmat2< T > & operator*= (const nsmat2< T > &rhs)
- nsmat2< T > & operator%= (const nsmat2< T > &rhs)
- nsmat2< T > & operator/= (const nsmat2< T > &rhs)
- nsmat2< T > & operator%= (const NSVec2< T > &rhs)
- nsmat2< T > & operator/= (const NSVec2< T > &rhs)
- nsmat2< T > & operator*= (const T &rhs)
- nsmat2< T > & operator/= (const T &rhs)
- nsmat2< T > operator++ (int32_t)
- nsmat2< T > operator-- (int32_t)
- nsmat2< T > & operator++ ()
- nsmat2< T > & operator-- ()
- nsmat2< T > & operator+= (const nsmat2< T > &rhs)
- nsmat2< T > & operator-= (const nsmat2< T > &rhs)
- const NSVec2< T > & operator[] (const uint32_t &pVal) const
- NSVec2< T > & operator[] (const uint32_t &pVal)
- NSVec2< T > operator() (const uint32_t &pVal)

6.39.1 Constructor & Destructor Documentation

6.39.1.1 template<class T> nsmat2< T >::nsmat2 ( ) [inline]

6.39.1.2 template<class T> nsmat2< T >::nsmat2 (const T & *val*) [inline]

6.39.1.3 template<class T> nsmat2< T >::nsmat2 (const nsmat2< T > & *copy*) [inline]

6.39.1.4 template<class T> nsmat2< T >::nsmat2 (const T & *a,* const T & *b,* const T & *c,* const T & *d*) [inline]

6.39.1.5 template<class T> nsmat2< T >::nsmat2 (const NSVec2< T > & *row1,* const NSVec2< T > & *row2*) [inline]

6.39.2 Member Function Documentation

6.39.2.1 template<class T> T* nsmat2< T >::dataPtr ( ) [inline]

6.39.2.2 template<class T> T nsmat2< T >::determinant ( ) const [inline]

6.39.2.3 template<class T> nsmat2<T>& nsmat2< T >::invert ( ) [inline]

6.39.2.4 template<class T> bool nsmat2< T >::operator!= (const nsmat2< T > & *rhs*) const [inline]

6.39.2.5 template<class T> nsmat2<T> nsmat2< T >::operator% (const nsmat2< T > & *rhs*) const [inline]

6.39.2.6 template<class T> nsmat2<T> nsmat2< T >::operator% (const NSVec2< T > & *rhs*) const [inline]

6.39.2.7 template<class T> nsmat2<T>& nsmat2< T >::operator%= (const nsmat2< T > & *rhs*) [inline]

6.39.2.8 template<class T> nsmat2<T>& nsmat2< T >::operator%= (const NSVec2< T > & *rhs*) [inline]

6.39.2.9 template<class T> NSVec2<T> nsmat2< T >::operator() (const uint32_t & *pVal*) [inline]

6.39.2.10 template<class T> nsmat2<T> nsmat2< T >::operator* (const nsmat2< T > & *rhs*) const [inline]

6.39.2.11 template<class T> NSVec2<T> nsmat2< T >::operator* (const NSVec2< T > & *rhs*) const [inline]

6.39.2.12 template<class T> nsmat2<T> nsmat2< T >::operator* (const T & *rhs*) const [inline]

6.39.2.13 template<class T> nsmat2<T>& nsmat2< T >::operator*= (const nsmat2< T > & *rhs*) [inline]

6.39.2.14 template<class T> nsmat2<T>& nsmat2< T >::operator*= (const T & *rhs*) [inline]

6.39.2.15 template<class T> nsmat2<T> nsmat2< T >::operator+ (const nsmat2< T > & *rhs*) const [inline]

6.39.2.16 template<class T> nsmat2<T> nsmat2< T >::operator++ (int32_t) [inline]

6.39.2.17 template<class T> nsmat2<T>& nsmat2< T >::operator++ ( ) [inline]

6.39.2.18 template<class T> nsmat2<T>& nsmat2< T >::operator+= (const nsmat2< T > & *rhs*) [inline]

6.39.2.19 template<class T> nsmat2<T> nsmat2< T >::operator- (const nsmat2< T > & *rhs*) const [inline]

6.39.2.20 template<class T> nsmat2<T> nsmat2< T >::operator-- (int32_t) [inline]

6.39.2.21 template<class T> nsmat2<T>& nsmat2< T >::operator-- ( ) [inline]

6.39.2.22 template<class T> nsmat2<T>& nsmat2< T >::operator-= (const nsmat2< T > & *rhs*) [inline]

6.39.2.23 template<class T> nsmat2<T> nsmat2< T >::operator/ (const nsmat2< T > & *rhs*) const [inline]

6.39.2.24 template<class T> nsmat2<T> nsmat2< T >::operator/ (const NSVec2< T > & *rhs*) const [inline]

6.39.2.25 template<class T> nsmat2<T> nsmat2< T >::operator/ (const T & *rhs*) const [inline]

6.39.2.26 template<class T> nsmat2<T>& nsmat2< T >::operator/= (const nsmat2< T > & *rhs*) [inline]

6.39.2.27 template<class T> nsmat2<T>& nsmat2< T >::operator/= (const NSVec2< T > & *rhs*) [inline]

6.39.2.28 template<class T> nsmat2<T>& nsmat2< T >::operator/= (const T & *rhs*) [inline]

6.39.2.29 template<class T> nsmat2<T>& nsmat2< T >::operator= (const nsmat2< T > & *rhs*) [inline]

6.39.2.30 template<class T> bool nsmat2< T >::operator== (const nsmat2< T > & *rhs*) const [inline]

6.39.2.31 template<class T> const NSVec2<T>& nsmat2< T >::operator[] (const uint32_t & *pVal*) const [inline]

6.39.2.32 template<class T> NSVec2<T>& nsmat2< T >::operator[] (const uint32_t & *pVal*) [inline]

6.39.2.33 template<class T> nsmat2<T>& nsmat2< T >::rotationFrom (T *angle,* bool *rads* = false) [inline]

6.39.2.34 template<class T> nsmat2<T>& nsmat2< T >::roundToZero ( ) [inline]

6.39.2.35 template<class T> nsmat2<T>& nsmat2< T >::scalingFrom (const NSVec2< T > & *scale*) [inline]

6.39.2.36 template<class T> nsmat2<T>& nsmat2< T >::scalingFrom (const nsmat2< T > & *transform2d*) [inline]

6.39.2.37 template<class T> nsmat2<T>& nsmat2< T >::scalingFrom (const nsmat3< T > & *transform2d*) [inline]

6.39.2.38 template<class T> nsmat2<T>& nsmat2< T >::set (const T & *val*) [inline]

6.39.2.39 template<class T> nsmat2<T>& nsmat2< T >::set (const T & *a,* const T & *b,* const T & *c,* const T & *d*) [inline]

6.39.2.40 template<class T> nsmat2<T>& nsmat2< T >::set (const NSVec2< T > & *row1,* const NSVec2< T > & *row2*) [inline]

6.39.2.41 template<class T> nsmat2<T>& nsmat2< T >::setColumn (const uint32_t & *i,* const T & *x,* const T & *y*) [inline]

6.39.2.42 template<class T> nsmat2<T>& nsmat2< T >::setColumn (const uint32_t & *i,* const NSVec2< T > & *col*) [inline]

6.39.2.43 template<class T> nsmat2<T>& nsmat2< T >::setIdentity ( ) [inline]

6.39.2.44 template<class T> std::string nsmat2< T >::toString (bool *newline* = true) const [inline]

6.39.2.45 template<class T> nsmat2<T>& nsmat2< T >::transpose ( ) [inline]

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

- /home/dprandle/Documents/code/ctrlmod/include/nsmat2.h

6.40 nsmat3< T > Struct Template Reference

`#include <nsmat3.h>`

Public Member Functions

- nsmat3 ()
- nsmat3 (const T &val)
- nsmat3 (const nsmat3 &copy)
- nsmat3 (const T &a, const T &b, const T &c, const T &d, const T &e, const T &f, const T &g, const T &h, const T &i)
- nsmat3 (const NSVec3< T > &row1, const NSVec3< T > &row2, const NSVec3< T > &row3)
- nsmat3 (const nsmat2< T > &basis)
- nsmat2< T > basis () const
- T * dataPtr ()
- T determinant () const
- nsmat3< T > & invert ()
- NSVec3< T > right () const
- nsmat3< T > & rotation2dFrom (const T angle, bool rads=false)
- nsmat3< T > & rotation2dFrom (const nsmat3< T > &transform2d)
- nsmat3< T > & rotation2dFrom (const nsmat2< T > &transform2d)
- nsmat3< T > & rotationFrom (const NSVec4< T > &axisAngle, bool rads=false)
- nsmat3< T > & rotationFrom (const NSVec3< T > &euler, typename NSVec3< T >::EulerOrder order, bool rads=false)
- nsmat3< T > & rotationFrom (const nsquat< T > &orientation)
- nsmat3< T > & rotationFrom (const nsmat3< T > &transform)
- nsmat3< T > & rotationFrom (const nsmat4< T > &transform)
- nsmat3< T > & rotationFrom (const NSVec3< T > &vec, const NSVec3< T > &toVec)
- nsmat3< T > & roundToZero ()
- nsmat3< T > & scaling2dFrom (const NSVec2< T > &scale)
- nsmat3< T > & scaling2dFrom (const nsmat2< T > &transform2d)
- nsmat3< T > & scaling2dFrom (const nsmat3< T > &transform2d)
- nsmat3< T > & scalingFrom (const NSVec3< T > &scale)
- nsmat3< T > & scalingFrom (const nsmat3< T > &transform)
- nsmat3< T > & scalingFrom (const nsmat4< T > &transform)
- nsmat3< T > & set (const T &a, const T &b, const T &c, const T &d, const T &e, const T &f, const T &g, const T &h, const T &i)
- nsmat3< T > & set (const T &val)
- nsmat3< T > & set (const NSVec3< T > &row1, const NSVec3< T > &row2, const NSVec3< T > &row3)
- nsmat3< T > & set (const nsmat2< T > &basis)
- nsmat3< T > & setColumn (const uint32_t &i, const T &x, const T &y, const T &z)
- nsmat3< T > & setColumn (const uint32_t &i, const NSVec3< T > &col)
- nsmat3< T > & setIdentity ()
- NSVec3< T > target () const
- nsmat3< T > & translation2dFrom (const NSVec3< T > &v3)
- nsmat3< T > & translation2dFrom (const NSVec2< T > &v2)
- nsmat3< T > & transpose ()
- std::string toString (bool newline=true) const
- NSVec3< T > up () const
- nsmat3< T > operator* (const nsmat3< T > &rhs) const
- nsmat3< T > operator/ (const nsmat3< T > &rhs) const

- nsmat3< T > operator% (const nsmat3< T > &rhs) const
- NSVec3< T > operator∗ (const NSVec3< T > &rhs) const
- nsmat3< T > operator% (const NSVec3< T > &rhs) const
- nsmat3< T > operator/ (const NSVec3< T > &rhs) const
- nsmat3< T > operator∗ (const T &rhs) const
- nsmat3< T > operator/ (const T &rhs) const
- nsmat3< T > operator+ (const nsmat3< T > &rhs) const
- nsmat3< T > operator- (const nsmat3< T > &rhs) const
- bool operator== (const nsmat3< T > &rhs) const
- bool operator!= (const nsmat3< T > &rhs) const
- nsmat3< T > & operator= (const nsmat3< T > &rhs)
- nsmat3< T > & operator∗= (const nsmat3< T > &rhs)
- nsmat3< T > & operator/= (const nsmat3< T > &rhs)
- nsmat3< T > & operator%= (const nsmat3< T > &rhs)
- nsmat3< T > & operator%= (const NSVec3< T > &rhs)
- nsmat3< T > & operator/= (const NSVec3< T > &rhs)
- nsmat3< T > & operator∗= (const T &rhs)
- nsmat3< T > & operator/= (const T &rhs)
- nsmat3< T > operator++ (int32_t)
- nsmat3< T > operator-- (int32_t)
- nsmat3< T > & operator++ ()
- nsmat3< T > & operator-- ()
- nsmat3< T > & operator+= (const nsmat3< T > &rhs)
- nsmat3< T > & operator-= (const nsmat3< T > &rhs)
- const NSVec3< T > & operator[] (const uint32_t &pVal) const
- NSVec3< T > & operator[] (const uint32_t &pVal)
- NSVec3< T > operator() (uint32_t pVal) const

6.40.1 Constructor & Destructor Documentation

6.40.1.1 template<class T> nsmat3< T >::nsmat3 ( ) [inline]

6.40.1.2 template<class T> nsmat3< T >::nsmat3 (const T & *val*) [inline]

6.40.1.3 template<class T> nsmat3< T >::nsmat3 (const nsmat3< T > & *copy*) [inline]

6.40.1.4 template<class T> nsmat3< T >::nsmat3 (const T & *a*, const T & *b*, const T & *c*, const T & *d*, const T & *e*, const T & *f*, const T & *g*, const T & *h*, const T & *i*) [inline]

6.40.1.5 template<class T> nsmat3< T >::nsmat3 (const NSVec3< T > & *row1*, const NSVec3< T > & *row2*, const NSVec3< T > & *row3*) [inline]

6.40.1.6 template<class T> nsmat3< T >::nsmat3 (const nsmat2< T > & *basis*) [inline]

6.40.2 Member Function Documentation

6.40.2.1 template<class T> nsmat2<T> nsmat3< T >::basis ( ) const [inline]

6.40.2.2 template<class T> T∗ nsmat3< T >::dataPtr ( ) [inline]

6.40.2.3 template<class T> T nsmat3< T >::determinant ( ) const [inline]

6.40.2.4 template<class T> nsmat3<T>& nsmat3< T >::invert ( ) [inline]

6.40.2.5 template<class T> bool nsmat3< T >::operator!= (const nsmat3< T > & *rhs*) const [inline]

6.40.2.6 template<class T> nsmat3<T> nsmat3< T >::operator% (const nsmat3< T > & *rhs*) const [inline]

6.40.2.7 template<class T> nsmat3<T> nsmat3< T >::operator% (const NSVec3< T > & *rhs*) const [inline]

6.40.2.8 template<class T> nsmat3<T>& nsmat3< T >::operator%= (const nsmat3< T > & *rhs*) [inline]

6.40.2.9 template<class T> nsmat3<T>& nsmat3< T >::operator%= (const NSVec3< T > & *rhs*) [inline]

6.40.2.10 template<class T> NSVec3<T> nsmat3< T >::operator() (uint32_t *pVal*) const [inline]

6.40.2.11 template<class T> nsmat3<T> nsmat3< T >::operator* (const nsmat3< T > & *rhs*) const [inline]

6.40.2.12 template<class T> NSVec3<T> nsmat3< T >::operator* (const NSVec3< T > & *rhs*) const [inline]

6.40.2.13 template<class T> nsmat3<T> nsmat3< T >::operator* (const T & *rhs*) const [inline]

6.40.2.14 template<class T> nsmat3<T>& nsmat3< T >::operator*= (const nsmat3< T > & *rhs*) [inline]

6.40.2.15 template<class T> nsmat3<T>& nsmat3< T >::operator*= (const T & *rhs*) [inline]

6.40.2.16 template<class T> nsmat3<T> nsmat3< T >::operator+ (const nsmat3< T > & *rhs*) const [inline]

6.40.2.17 template<class T> nsmat3<T> nsmat3< T >::operator++ (int32_t) [inline]

6.40.2.18 template<class T> nsmat3<T>& nsmat3< T >::operator++ ( ) [inline]

6.40.2.19 template<class T> nsmat3<T>& nsmat3< T >::operator+= (const nsmat3< T > & *rhs*) [inline]

6.40.2.20 template<class T> nsmat3<T> nsmat3< T >::operator- (const nsmat3< T > & *rhs*) const [inline]

6.40.2.21 template<class T> nsmat3<T> nsmat3< T >::operator-- (int32_t) [inline]

6.40.2.22 template<class T> nsmat3<T>& nsmat3< T >::operator-- ( ) [inline]

6.40.2.23 template<class T> nsmat3<T>& nsmat3< T >::operator-= (const nsmat3< T > & *rhs*) [inline]

6.40.2.24 template<class T> nsmat3<T> nsmat3< T >::operator/ (const nsmat3< T > & *rhs*) const [inline]

6.40.2.25 template<class T> nsmat3<T> nsmat3< T >::operator/ (const NSVec3< T > & *rhs*) const [inline]

6.40.2.26 template<class T> nsmat3<T> nsmat3< T >::operator/ (const T & *rhs*) const [inline]

6.40.2.27 template<class T> nsmat3<T>& nsmat3< T >::operator/= (const nsmat3< T > & *rhs*) [inline]

6.40.2.28 template<class T> nsmat3<T>& nsmat3< T >::operator/= (const NSVec3< T > & *rhs*) [inline]

6.40.2.29 template<class T> nsmat3<T>& nsmat3< T >::operator/= (const T & *rhs*) [inline]

6.40.2.30 template<class T> nsmat3<T>& nsmat3< T >::operator= (const nsmat3< T > & *rhs*) [inline]

6.40.2.31 template<class T> bool nsmat3< T >::operator== (const nsmat3< T > & *rhs*) const [inline]

6.40.2.32 template<class T> const NSVec3<T>& nsmat3< T >::operator[] (const uint32_t & *pVal*) const [inline]

6.40.2.33 template<class T> NSVec3<T>& nsmat3< T >::operator[] (const uint32_t & *pVal*) [inline]

6.40.2.34 template<class T> NSVec3<T> nsmat3< T >::right ( ) const [inline]

6.40.2.35 template<class T> nsmat3<T>& nsmat3< T >::rotation2dFrom (const T *angle,* bool *rads =* false) [inline]

6.40.2.36 template<class T> nsmat3<T>& nsmat3< T >::rotation2dFrom (const nsmat3< T > & *transform2d*) [inline]

6.40.2.37 template<class T> nsmat3<T>& nsmat3< T >::rotation2dFrom (const nsmat2< T > & *transform2d*) [inline]

6.40.2.38 template<class T> nsmat3<T>& nsmat3< T >::rotationFrom (const NSVec4< T > & *axisAngle,* bool *rads =* false) [inline]

6.40.2.39 template<class T> nsmat3<T>& nsmat3< T >::rotationFrom (const NSVec3< T > & *euler,* typename NSVec3< T >::EulerOrder *order,* bool *rads =* false) [inline]

6.40.2.40 template<class T> nsmat3<T>& nsmat3< T >::rotationFrom (const nsquat< T > & *orientation*) [inline]

6.40.2.41 template<class T> nsmat3<T>& nsmat3< T >::rotationFrom (const nsmat3< T > & *transform*) [inline]

6.40.2.42 template<class T> nsmat3<T>& nsmat3< T >::rotationFrom (const nsmat4< T > & *transform*) [inline]

6.40.2.43 template<class T> nsmat3<T>& nsmat3< T >::rotationFrom (const NSVec3< T > & *vec,* const NSVec3< T > & *toVec*) [inline]

6.40.2.44 template<class T> nsmat3<T>& nsmat3< T >::roundToZero ( ) [inline]

6.40.2.45 template<class T> nsmat3<T>& nsmat3< T >::scaling2dFrom (const NSVec2< T > & *scale*) [inline]

6.40.2.46 template<class T> nsmat3<T>& nsmat3< T >::scaling2dFrom (const nsmat2< T > & *transform2d*) [inline]

6.40.2.47 template<class T> nsmat3<T>& nsmat3< T >::scaling2dFrom (const nsmat3< T > & *transform2d*) [inline]

6.40.2.48 template<class T> nsmat3<T>& nsmat3< T >::scalingFrom (const NSVec3< T > & *scale*) [inline]

6.40.2.49 template<class T> nsmat3<T>& nsmat3< T >::scalingFrom (const nsmat3< T > & *transform*) [inline]

6.40.2.50 template<class T> nsmat3<T>& nsmat3< T >::scalingFrom (const nsmat4< T > & *transform*) [inline]

6.40.2.51 template<class T> nsmat3<T>& nsmat3< T >::set (const T & *a,* const T & *b,* const T & *c,* const T & *d,* const T & *e,* const T & *f,* const T & *g,* const T & *h,* const T & *i*) [inline]

6.40.2.52 template<class T> nsmat3<T>& nsmat3< T >::set (const T & *val*) [inline]

6.40.2.53 template<class T> nsmat3<T>& nsmat3< T >::set (const NSVec3< T > & *row1,* const NSVec3< T > & *row2,* const NSVec3< T > & *row3*) [inline]

6.40.2.54 template<class T> nsmat3<T>& nsmat3< T >::set (const nsmat2< T > & *basis*) [inline]

6.40.2.55 template<class T> nsmat3<T>& nsmat3< T >::setColumn (const uint32_t & *i,* const T & *x,* const T & *y,* const T & *z*) [inline]

6.40.2.56 template<class T> nsmat3<T>& nsmat3< T >::setColumn (const uint32_t & *i,* const NSVec3< T > & *col*) [inline]

6.40.2.57 template<class T> nsmat3<T>& nsmat3< T >::setIdentity ( ) [inline]

6.40.2.58 template<class T> NSVec3<T> nsmat3< T >::target ( ) const [inline]

6.40.2.59 template<class T> std::string nsmat3< T >::toString (bool *newline* = true) const [inline]

6.40.2.60 template<class T> nsmat3<T>& nsmat3< T >::translation2dFrom (const NSVec3< T > & *v3*) [inline]

6.40.2.61 template<class T> nsmat3<T>& nsmat3< T >::translation2dFrom (const NSVec2< T > & *v2*) [inline]

6.40.2.62 template<class T> nsmat3<T>& nsmat3< T >::transpose ( ) [inline]

6.40.2.63 template<class T> NSVec3<T> nsmat3< T >::up ( ) const [inline]

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

- /home/dprandle/Documents/code/ctrlmod/include/nsmat3.h

6.41 nsmat4< T > Struct Template Reference

`#include <nsmat4.h>`

Public Member Functions

- nsmat4 ()
- nsmat4 (const nsmat4 &copy)
- nsmat4 (const T &val)
- nsmat4 (const T &a, const T &b, const T &c, const T &d, const T &e, const T &f, const T &g, const T &h, const T &i, const T &j, const T &k, const T &l, const T &m, const T &n, const T &o, const T &p)
- nsmat4 (const NSVec4< T > &row1, const NSVec4< T > &row2, const NSVec4< T > &row3, const NSVec4< T > &row4)
- nsmat4 (const nsmat3< T > &basis)
- nsmat3< T > basis () const
- T * dataPtr ()
- T determinant () const
- nsmat4< T > & invert ()
- nsmat4< T > & orthoFrom (const T &left, const T &right, const T &top, const T &bottom, const T &near, const T &far)
- nsmat4< T > & perspectiveFrom (const T &fovAngle, const T &aspectRatio, const T &zNear, const T &zFar)
- NSVec3< T > right () const
- nsmat4< T > & rotationFrom (const NSVec4< T > &axisAngle, bool rads=false)

- nsmat4< T > & rotationFrom (const NSVec3< T > &euler, typename NSVec3< T >::EulerOrder order, bool rads=false)
- nsmat4< T > & rotationFrom (const nsquat< T > &orientation)
- nsmat4< T > & rotationFrom (const nsmat3< T > &transform)
- nsmat4< T > & rotationFrom (const nsmat4< T > &transform)
- nsmat4< T > & rotationFrom (const NSVec3< T > &vec, const NSVec3< T > &toVec)
- nsmat4< T > & roundToZero ()
- nsmat4< T > & scalingFrom (const NSVec3< T > &scale)
- nsmat4< T > & scalingFrom (const nsmat3< T > &transform)
- nsmat4< T > & scalingFrom (const nsmat4< T > &transform)
- nsmat4< T > & set (const T &val)
- nsmat4< T > & set (const T &a, const T &b, const T &c, const T &d, const T &e, const T &f, const T &g, const T &h, const T &i, const T &j, const T &k, const T &l, const T &m, const T &n, const T &o, const T &p)
- nsmat4< T > & set (const NSVec4< T > &row1, const NSVec4< T > &row2, const NSVec4< T > &row3, const NSVec4< T > &row4)
- nsmat4< T > & set (const nsmat3< T > &basis)
- nsmat4< T > & setColumn (const uint32_t &i, const T &x, const T &y, const T &z, const T &w)
- nsmat4< T > & setColumn (const uint32_t &i, const NSVec4< T > &col)
- nsmat4< T > & setIdentity ()
- NSVec3< T > target () const
- nsmat4< T > & translationFrom (const NSVec3< T > &pos)
- nsmat4< T > & translationFrom (const NSVec4< T > &posw)
- nsmat4< T > & translationFrom (const nsmat4< T > &transform)
- nsmat4< T > & transpose ()
- std::string toString (bool newline=true) const
- NSVec3< T > up () const
- nsmat4< T > operator* (const nsmat4< T > &rhs) const
- nsmat4< T > operator/ (const nsmat4< T > &rhs) const
- nsmat4< T > operator% (const nsmat4< T > &rhs) const
- NSVec4< T > operator* (const NSVec4< T > &rhs) const
- nsmat4< T > operator% (const NSVec4< T > &rhs) const
- nsmat4< T > operator/ (const NSVec4< T > &rhs) const
- nsmat4< T > operator* (const T &rhs) const
- nsmat4< T > operator/ (const T &rhs) const
- nsmat4< T > operator+ (const nsmat4< T > &rhs) const
- nsmat4< T > operator- (const nsmat4< T > &rhs) const
- bool operator== (const nsmat4< T > &rhs) const
- bool operator!= (const nsmat4< T > &rhs) const
- nsmat4< T > & operator= (const nsmat4< T > &rhs)
- nsmat4< T > & operator*= (const nsmat4< T > &rhs)
- nsmat4< T > & operator/= (const nsmat4< T > &rhs)
- nsmat4< T > & operator%= (const nsmat4< T > &rhs)
- nsmat4< T > & operator%= (const NSVec4< T > &rhs)
- nsmat4< T > & operator/= (const NSVec4< T > &rhs)
- nsmat4< T > & operator*= (const T &rhs)
- nsmat4< T > & operator/= (const T &rhs)
- nsmat4< T > operator++ (int32_t)
- nsmat4< T > operator-- (int32_t)
- nsmat4< T > & operator++ ()
- nsmat4< T > & operator-- ()
- nsmat4< T > & operator+= (const nsmat4< T > &rhs)
- nsmat4< T > & operator-= (const nsmat4< T > &rhs)
- const NSVec4< T > & operator[] (const uint32_t &pVal) const
- NSVec4< T > & operator[] (const uint32_t &pVal)
- NSVec4< T > operator() (const uint32_t &pVal) const

6.41.1 Constructor & Destructor Documentation

6.41.1.1 template<class T> nsmat4< T >::nsmat4 ( ) [inline]

6.41.1.2 template<class T> nsmat4< T >::nsmat4 (const nsmat4< T > & *copy*) [inline]

6.41.1.3 template<class T> nsmat4< T >::nsmat4 (const T & *val*) [inline]

6.41.1.4 template<class T> nsmat4< T >::nsmat4 (const T & *a,* const T & *b,* const T & *c,* const T & *d,* const T & *e,* const T & *f,* const T & *g,* const T & *h,* const T & *i,* const T & *j,* const T & *k,* const T & *l,* const T & *m,* const T & *n,* const T & *o,* const T & *p*) [inline]

6.41.1.5 template<class T> nsmat4< T >::nsmat4 (const NSVec4< T > & *row1,* const NSVec4< T > & *row2,* const NSVec4< T > & *row3,* const NSVec4< T > & *row4*) [inline]

6.41.1.6 template<class T> nsmat4< T >::nsmat4 (const nsmat3< T > & *basis*) [inline]

6.41.2 Member Function Documentation

6.41.2.1 template<class T> nsmat3<T> nsmat4< T >::basis ( ) const [inline]

6.41.2.2 template<class T> T* nsmat4< T >::dataPtr ( ) [inline]

6.41.2.3 template<class T> T nsmat4< T >::determinant ( ) const [inline]

6.41.2.4 template<class T> nsmat4<T>& nsmat4< T >::invert ( ) [inline]

6.41.2.5 template<class T> bool nsmat4< T >::operator!= (const nsmat4< T > & *rhs*) const [inline]

6.41.2.6 template<class T> nsmat4<T> nsmat4< T >::operator% (const nsmat4< T > & *rhs*) const [inline]

6.41.2.7 template<class T> nsmat4<T> nsmat4< T >::operator% (const NSVec4< T > & *rhs*) const [inline]

6.41.2.8 template<class T> nsmat4<T>& nsmat4< T >::operator%= (const nsmat4< T > & *rhs*) [inline]

6.41.2.9 template<class T> nsmat4<T>& nsmat4< T >::operator%= (const NSVec4< T > & *rhs*) [inline]

6.41.2.10 template<class T> NSVec4<T> nsmat4< T >::operator() (const uint32_t & *pVal*) const [inline]

6.41.2.11 template<class T> nsmat4<T> nsmat4< T >::operator* (const nsmat4< T > & *rhs*) const [inline]

6.41.2.12 template<class T> NSVec4<T> nsmat4< T >::operator* (const NSVec4< T > & *rhs*) const [inline]

6.41.2.13 template<class T> nsmat4<T> nsmat4< T >::operator* (const T & *rhs*) const [inline]

6.41.2.14 template<class T> nsmat4<T>& nsmat4< T >::operator*= (const nsmat4< T > & *rhs*) [inline]

6.41.2.15 template<class T> nsmat4<T>& nsmat4< T >::operator*= (const T & *rhs*) [inline]

6.41.2.16 template<class T> nsmat4<T> nsmat4< T >::operator+ (const nsmat4< T > & *rhs*) const [inline]

6.41.2.17 template<class T> nsmat4<T> nsmat4< T >::operator++ (int32_t) [inline]

6.41.2.18 template<class T> nsmat4<T>& nsmat4< T >::operator++ ( ) [inline]

6.41.2.19 template<class T> nsmat4<T>& nsmat4< T >::operator+= (const nsmat4< T > & *rhs*) [inline]

6.41.2.20 template<class T> nsmat4<T> nsmat4< T >::operator- (const nsmat4< T > & *rhs*) const [inline]

6.41.2.21 template<class T> nsmat4<T> nsmat4< T >::operator-- (int32_t) [inline]

6.41.2.22 template<class T> nsmat4<T>& nsmat4< T >::operator-- ( ) [inline]

6.41.2.23 template<class T> nsmat4<T>& nsmat4< T >::operator-= (const nsmat4< T > & *rhs*) [inline]

6.41.2.24 template<class T> nsmat4<T> nsmat4< T >::operator/ (const nsmat4< T > & *rhs*) const [inline]

6.41.2.25 template<class T> nsmat4<T> nsmat4< T >::operator/ (const NSVec4< T > & *rhs*) const [inline]

6.41.2.26 template<class T> nsmat4<T> nsmat4< T >::operator/ (const T & *rhs*) const [inline]

6.41.2.27 template<class T> nsmat4<T>& nsmat4< T >::operator/= (const nsmat4< T > & *rhs*) [inline]

6.41.2.28 template<class T> nsmat4<T>& nsmat4< T >::operator/= (const NSVec4< T > & *rhs*) [inline]

6.41.2.29 template<class T> nsmat4<T>& nsmat4< T >::operator/= (const T & *rhs*) [inline]

6.41.2.30 template<class T> nsmat4<T>& nsmat4< T >::operator= (const nsmat4< T > & *rhs*) [inline]

6.41.2.31 template<class T> bool nsmat4< T >::operator== (const nsmat4< T > & *rhs*) const [inline]

6.41.2.32 template<class T> const NSVec4<T>& nsmat4< T >::operator[] (const uint32_t & *pVal*) const [inline]

6.41.2.33 template<class T> NSVec4<T>& nsmat4< T >::operator[] (const uint32_t & *pVal*) [inline]

6.41.2.34 template<class T> nsmat4<T>& nsmat4< T >::orthoFrom (const T & *left,* const T & *right,* const T & *top,* const T & *bottom,* const T & *near,* const T & *far*) [inline]

6.41.2.35 template<class T> nsmat4<T>& nsmat4< T >::perspectiveFrom (const T & *fovAngle,* const T & *aspectRatio,* const T & *zNear,* const T & *zFar*) [inline]

6.41.2.36 template<class T> NSVec3<T> nsmat4< T >::right ( ) const [inline]

6.41.2.37 template<class T> nsmat4<T>& nsmat4< T >::rotationFrom (const NSVec4< T > & *axisAngle,* bool *rads* = false) [inline]

6.41.2.38 template<class T> nsmat4<T>& nsmat4< T >::rotationFrom (const NSVec3< T > & *euler,* typename NSVec3< T >::EulerOrder *order,* bool *rads* = false) [inline]

6.41.2.39 template<class T> nsmat4<T>& nsmat4< T >::rotationFrom (const nsquat< T > & *orientation*) [inline]

6.41.2.40 template<class T> nsmat4<T>& nsmat4< T >::rotationFrom (const nsmat3< T > & *transform*) [inline]

6.41.2.41 template<class T> nsmat4<T>& nsmat4< T >::rotationFrom (const nsmat4< T > & *transform*) [inline]

6.41.2.42 template<class T> nsmat4<T>& nsmat4< T >::rotationFrom (const NSVec3< T > & *vec,* const NSVec3< T > & *toVec*) [inline]

6.41.2.43 template<class T> nsmat4<T>& nsmat4< T >::roundToZero ( ) [inline]

6.41.2.44 template<class T> nsmat4<T>& nsmat4< T >::scalingFrom (const NSVec3< T > & *scale*) [inline]

6.41.2.45 template<class T> nsmat4<T>& nsmat4< T >::scalingFrom (const nsmat3< T > & *transform*) [inline]

6.41.2.46 template<class T> nsmat4<T>& nsmat4< T >::scalingFrom (const nsmat4< T > & *transform*) [inline]

6.41.2.47 template<class T> nsmat4<T>& nsmat4< T >::set (const T & *val*) [inline]

6.41.2.48 template<class T> nsmat4<T>& nsmat4< T >::set (const T & *a,* const T & *b,* const T & *c,* const T & *d,* const T & *e,* const T & *f,* const T & *g,* const T & *h,* const T & *i,* const T & *j,* const T & *k,* const T & *l,* const T & *m,* const T & *n,* const T & *o,* const T & *p*) [inline]

6.41.2.49 template<class T> nsmat4<T>& nsmat4< T >::set (const NSVec4< T > & *row1,* const NSVec4< T > & *row2,* const NSVec4< T > & *row3,* const NSVec4< T > & *row4*) [inline]

6.41.2.50 template<class T> nsmat4<T>& nsmat4< T >::set (const nsmat3< T > & *basis*) [inline]

6.41.2.51 template<class T> nsmat4<T>& nsmat4< T >::setColumn (const uint32_t & *i,* const T & *x,* const T & *y,* const T & *z,* const T & *w*) [inline]

6.41.2.52 template<class T> nsmat4<T>& nsmat4< T >::setColumn (const uint32_t & *i,* const NSVec4< T > & *col*) [inline]

6.41.2.53 template<class T> nsmat4<T>& nsmat4< T >::setIdentity ( ) [inline]

6.41.2.54 template<class T> NSVec3<T> nsmat4< T >::target ( ) const [inline]

6.41.2.55 template<class T> std::string nsmat4< T >::toString (bool *newline* = true) const [inline]

6.41.2.56 template<class T> nsmat4<T>& nsmat4< T >::translationFrom (const NSVec3< T > & *pos*) [inline]

6.41.2.57 template<class T> nsmat4<T>& nsmat4< T >::translationFrom (const NSVec4< T > & *posw*) [inline]

6.41.2.58 template<class T> nsmat4<T>& nsmat4< T >::translationFrom (const nsmat4< T > & *transform*) [inline]

6.41.2.59 template<class T> nsmat4<T>& nsmat4< T >::transpose ( ) [inline]

6.41.2.60 template<class T> NSVec3<T> nsmat4< T >::up ( ) const [inline]

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

- /home/dprandle/Documents/code/ctrlmod/include/nsmat4.h

6.42 nsquat< T > Struct Template Reference

`#include <nsquat.h>`

Public Member Functions

- nsquat (const nsquat< T > &copy)

- nsquat ()
- nsquat (const T &pX, const T &pY, const T &pZ, const T &pW)
- nsquat< T > & conjugate ()
- nsquat< T > & from (const NSVec4< T > &axisAngle, bool pRads=false)
- nsquat< T > & from (const NSVec3< T > &euler, typename NSVec3< T >::EulerOrder order, bool pRads=false)
- nsquat< T > & from (const nsmat3< T > &rotationMat3)
- nsquat< T > & from (const nsmat4< T > &transform)
- nsquat< T > & from (const NSVec3< T > &vec, const NSVec3< T > &toVec)
- nsquat< T > & invert ()
- T length () const
- T lengthSq () const
- nsquat< T > & normalize ()
- NSVec3< T > right () const
- nsquat< T > & roundToZero ()
- nsquat< T > & set (const T &pX, const T &pY, const T &pZ, const T &pW)
- nsquat< T > & setIdentity ()
- nsquat< T > & slerp (const nsquat< T > &second, const T &scalingFactor)
- NSVec3< T > target () const
- std::string toString ()
- NSVec3< T > up () const
- nsquat< T > operator+ (const nsquat< T > &rhs) const
- nsquat< T > operator- (const nsquat< T > &rhs) const
- nsquat< T > operator* (const nsquat< T > &rhs) const
- nsquat< T > operator/ (const nsquat< T > &rhs) const
- nsquat< T > operator% (const nsquat< T > &rhs) const
- NSVec3< T > operator* (const NSVec3< T > &rhs) const
- nsquat< T > operator* (const T &rhs) const
- nsquat< T > operator/ (const T &rhs) const
- nsquat< T > & operator= (const nsquat< T > &rhs)
- nsquat< T > operator++ (int32_t)
- nsquat< T > operator-- (int32_t)
- nsquat< T > & operator++ ()
- nsquat< T > & operator-- ()
- nsquat< T > & operator+= (const nsquat< T > &rhs)
- nsquat< T > & operator-= (const nsquat< T > &rhs)
- nsquat< T > & operator*= (const nsquat< T > &rhs)
- nsquat< T > & operator/= (const nsquat< T > &rhs)
- nsquat< T > & operator%= (const nsquat< T > &rhs)
- nsquat< T > & operator*= (const T &rhs)
- nsquat< T > & operator/= (const T &rhs)
- bool operator== (const nsquat< T > &rhs) const
- bool operator!= (const nsquat< T > &rhs) const
- bool operator== (const T &rhs) const
- bool operator!= (const T &rhs) const
- const T & operator[] (const uint32_t &pVal) const
- T & operator[] (const uint32_t &pVal)

Public Attributes

- union {
  T data [4]
  struct {
  T x
  T y
  T z
  T w
  }
  struct {
  T b
  T c
  T d
  T a
  }
  };

6.42.1 Constructor & Destructor Documentation

6.42.1.1 template<class T> nsquat< T >::nsquat (const nsquat< T > & *copy*) [inline]

6.42.1.2 template<class T> nsquat< T >::nsquat ( ) [inline]

6.42.1.3 template<class T> nsquat< T >::nsquat (const T & *pX*, const T & *pY*, const T & *pZ*, const T & *pW*) [inline]

6.42.2 Member Function Documentation

6.42.2.1 template<class T> nsquat<T>& nsquat< T >::conjugate ( ) [inline]

6.42.2.2 template<class T> nsquat<T>& nsquat< T >::from (const NSVec4< T > & *axisAngle*, bool *pRads* = false) [inline]

6.42.2.3 template<class T> nsquat<T>& nsquat< T >::from (const NSVec3< T > & *euler*, typename NSVec3< T >::EulerOrder *order*, bool *pRads* = false) [inline]

6.42.2.4 template<class T> nsquat<T>& nsquat< T >::from (const nsmat3< T > & *rotationMat3*) [inline]

6.42.2.5 template<class T> nsquat<T>& nsquat< T >::from (const nsmat4< T > & *transform*) [inline]

6.42.2.6 template<class T> nsquat<T>& nsquat< T >::from (const NSVec3< T > & *vec*, const NSVec3< T > & *toVec*) [inline]

6.42.2.7 template<class T> nsquat<T>& nsquat< T >::invert ( ) [inline]

6.42.2.8 template<class T> T nsquat< T >::length ( ) const [inline]

6.42.2.9 template<class T> T nsquat< T >::lengthSq ( ) const [inline]

6.42.2.10 template<class T> nsquat<T>& nsquat< T >::normalize ( ) [inline]

6.42.2.11 template<class T> bool nsquat< T >::operator!= (const nsquat< T > & *rhs*) const [inline]

6.42.2.12 template<class T> bool nsquat< T >::operator!= (const T & *rhs*) const [inline]

6.42.2.13 template<class T> nsquat<T> nsquat< T >::operator% (const nsquat< T > & *rhs*) const [inline]

6.42.2.14 template<class T> nsquat<T>& nsquat< T >::operator%= (const nsquat< T > & *rhs*) [inline]

6.42.2.15 template<class T> nsquat<T> nsquat< T >::operator* (const nsquat< T > & *rhs*) const [inline]

6.42.2.16 template<class T> NSVec3<T> nsquat< T >::operator* (const NSVec3< T > & *rhs*) const [inline]

6.42.2.17 template<class T> nsquat<T> nsquat< T >::operator* (const T & *rhs*) const [inline]

6.42.2.18 template<class T> nsquat<T>& nsquat< T >::operator*= (const nsquat< T > & *rhs*) [inline]

6.42.2.19 template<class T> nsquat<T>& nsquat< T >::operator*= (const T & *rhs*) [inline]

6.42.2.20 template<class T> nsquat<T> nsquat< T >::operator+ (const nsquat< T > & *rhs*) const [inline]

6.42.2.21 template<class T> nsquat<T> nsquat< T >::operator++ (int32_t) [inline]

6.42.2.22 template<class T> nsquat<T>& nsquat< T >::operator++ ( ) [inline]

6.42.2.23 template<class T> nsquat<T>& nsquat< T >::operator+= (const nsquat< T > & *rhs*) [inline]

6.42.2.24 template<class T> nsquat<T> nsquat< T >::operator- (const nsquat< T > & *rhs*) const [inline]

6.42.2.25 template<class T> nsquat<T> nsquat< T >::operator-- (int32_t) [inline]

6.42.2.26 template<class T> nsquat<T>& nsquat< T >::operator-- ( ) [inline]

6.42.2.27 template<class T> nsquat<T>& nsquat< T >::operator-= (const nsquat< T > & *rhs*) [inline]

6.42.2.28 template<class T> nsquat<T> nsquat< T >::operator/ (const nsquat< T > & *rhs*) const [inline]

6.42.2.29 template<class T> nsquat<T> nsquat< T >::operator/ (const T & *rhs*) const [inline]

6.42.2.30 template<class T> nsquat<T>& nsquat< T >::operator/= (const nsquat< T > & *rhs*) [inline]

6.42.2.31 template<class T> nsquat<T>& nsquat< T >::operator/= (const T & *rhs*) [inline]

6.42.2.32 template<class T> nsquat<T>& nsquat< T >::operator= (const nsquat< T > & *rhs*) [inline]

6.42.2.33 template<class T> bool nsquat< T >::operator== (const nsquat< T > & *rhs*) const [inline]

6.42.2.34 template<class T> bool nsquat< T >::operator== (const T & *rhs*) const [inline]

6.42.2.35 template<class T> const T& nsquat< T >::operator[] (const uint32_t & *pVal*) const [inline]

6.42.2.36 template<class T> T& nsquat< T >::operator[] (const uint32_t & *pVal*) [inline]

6.42.2.37 template<class T> NSVec3<T> nsquat< T >::right ( ) const [inline]

6.42.2.38 template<class T> nsquat<T>& nsquat< T >::roundToZero ( ) [inline]

6.42.2.39 template<class T> nsquat<T>& nsquat< T >::set (const T & *pX,* const T & *pY,* const T & *pZ,* const T & *pW*) [inline]

6.42.2.40 template<class T> nsquat<T>& nsquat< T >::setIdentity ( ) [inline]

6.42.2.41 template<class T> nsquat<T>& nsquat< T >::slerp (const nsquat< T > & *second*, const T & *scalingFactor*) [inline]

6.42.2.42 template<class T> NSVec3<T> nsquat< T >::target ( ) const [inline]

6.42.2.43 template<class T> std::string nsquat< T >::toString ( ) [inline]

6.42.2.44 template<class T> NSVec3<T> nsquat< T >::up ( ) const [inline]

6.42.3 Member Data Documentation

6.42.3.1 union { ... }

6.42.3.2 template<class T> T nsquat< T >::a

6.42.3.3 template<class T> T nsquat< T >::b

6.42.3.4 template<class T> T nsquat< T >::c

6.42.3.5 template<class T> T nsquat< T >::d

6.42.3.6 template<class T> T nsquat< T >::data[4]

6.42.3.7 template<class T> T nsquat< T >::w

6.42.3.8 template<class T> T nsquat< T >::x

6.42.3.9 template<class T> T nsquat< T >::y

6.42.3.10 template<class T> T nsquat< T >::z

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

- /home/dprandle/Documents/code/ctrlmod/include/nsquat.h

6.43 NSVec2< T > Struct Template Reference

`#include <nsvec2.h>`

Public Member Functions

- NSVec2 (const NSVec2< T > &pCopy)
- NSVec2 (const T &pInit=static_cast< T >(0))
- NSVec2 (const T &pX, const T &pY)
- NSVec2< T > & abs ()
- T angle (bool pRads=false) const
- T angleTo (const NSVec2< T > &pVec, bool pRads=false) const
- NSVec2< T > & ceil ()
- NSVec2< T > & clamp (const T &pMin=static_cast< T >(0), const T &pMax=static_cast< T >(1))
- T distanceTo (const NSVec2< T > &pVec) const
- NSVec2< T > & floor ()
- NSVec2< T > & fract ()
- T length () const
- T lengthSq () const

- template<class T2 >
  NSVec2< T > & lerp (const NSVec2< T > &vec, const T2 &scalingFactor)
- T min () const
- NSVec2< T > & minimize (const NSVec2< T > &rhs)
- T max () const
- NSVec2< T > & maximize (const NSVec2< T > &rhs)
- NSVec2< T > & normalize ()
- NSVec2< T > polar (bool pRads=false) const
- NSVec2< T > & projectOn (const NSVec2< T > &vec)
- NSVec2< T > & projectOnPlane (const NSVec2< T > &planeNormal)
- NSVec2< T > & reflect (const NSVec2< T > &normal)
- template<class T2 >
  NSVec2< T > & rotate (const T2 &angle_)
- NSVec2< T > & round ()
- NSVec2< T > & roundToZero ()
- NSVec2< T > & scalingFrom (const nsmat2< T > &transform2d)
- NSVec2< T > & scalingFrom (const nsmat3< T > &transform2d)
- NSVec2< T > & set (const T &pVal)
- NSVec2< T > & set (const T &pX, const T &pY)
- NSVec2< T > & setFromPolar (const T &pMag, T angle_, bool pRads=false)
- NSVec2< T > & setFromPolar (const NSVec2< T > &pVec, bool pRads=false)
- NSVec2< T > & setLength (const T &len)
- NSVec2< T > & translationFrom (const nsmat3< T > &transform2d)
- std::string toString (bool pPolar=false)
- NSVec2< T > operator+ (const NSVec2< T > &pRHS) const
- NSVec2< T > operator- (const NSVec2< T > &pRHS) const
- T operator∗ (const NSVec2< T > &pRHS) const
- nsmat2< T > operator^ (const NSVec2< T > &pRHS) const
- NSVec2< T > operator% (const NSVec2< T > &pRHS) const
- NSVec2< T > operator/ (const NSVec2< T > &pRHS) const
- NSVec2< T > operator∗ (const T &pRHS) const
- NSVec2< T > operator/ (const T &pRHS) const
- NSVec2< T > & operator= (const NSVec2< T > &pRHS)
- NSVec2< T > operator++ (int32_t)
- NSVec2< T > operator-- (int32_t)
- NSVec2< T > & operator++ ()
- NSVec2< T > & operator-- ()
- NSVec2< T > & operator+= (const NSVec2< T > &pRHS)
- NSVec2< T > & operator-= (const NSVec2< T > &pRHS)
- NSVec2< T > & operator%= (const NSVec2< T > &pRHS)
- NSVec2< T > & operator/= (const NSVec2< T > &pRHS)
- NSVec2< T > & operator∗= (const T &pRHS)
- NSVec2< T > & operator/= (const T &pRHS)
- bool operator== (const NSVec2< T > &pRHS) const
- bool operator!= (const NSVec2< T > &pRHS) const
- bool operator== (const T &pRHS) const
- bool operator!= (const T &pRHS) const
- const T & operator[] (const uint32_t &pVal) const
- T & operator[] (const uint32_t &pVal)
- NSVec2< T > xx () const
- NSVec2< T > yx () const
- NSVec2< T > yy () const
- NSVec2< T > ss () const
- NSVec2< T > ts () const
- NSVec2< T > tt () const

- NSVec2< T > uu () const
- NSVec2< T > vu () const
- NSVec2< T > vv () const
- NSVec2< T > ww () const
- NSVec2< T > hw () const
- NSVec2< T > hh () const

Public Attributes

- union {
  T data [2]
  struct {
  T x
  T y
  }
  struct {
  T w
  T h
  }
  struct {
  T s
  T t
  }
  struct {
  T u
  T v
  }
  };

6.43.1 Constructor & Destructor Documentation

6.43.1.1 template<class T> NSVec2< T >::NSVec2 (const NSVec2< T > & *pCopy*) [inline]

6.43.1.2 template<class T> NSVec2< T >::NSVec2 (const T & *pInit* = static_cast<T>(0)) [inline]

6.43.1.3 template<class T> NSVec2< T >::NSVec2 (const T & *pX*, const T & *pY*) [inline]

6.43.2 Member Function Documentation

6.43.2.1 template<class T> NSVec2<T>& NSVec2< T >::abs ( ) [inline]

6.43.2.2 template<class T> T NSVec2< T >::angle (bool *pRads* = false) const [inline]

6.43.2.3 template<class T> T NSVec2< T >::angleTo (const NSVec2< T > & *pVec*, bool *pRads* = false) const [inline]

6.43.2.4 template<class T> NSVec2<T>& NSVec2< T >::ceil ( ) [inline]

6.43.2.5 template<class T> NSVec2<T>& NSVec2< T >::clamp (const T & *pMin* = static_cast<T>(0), const T & *pMax* = static_cast<T>(1)) [inline]

6.43.2.6 template<class T> T NSVec2< T >::distanceTo (const NSVec2< T > & *pVec*) const [inline]

6.43.2.7 template<class T> NSVec2<T>& NSVec2< T >::floor ( ) [inline]

6.43.2.8 template<class T> NSVec2<T>& NSVec2< T >::fract ( ) [inline]

6.43.2.9 template<class T> NSVec2<T> NSVec2< T >::hh ( ) const [inline]

6.43.2.10 template<class T> NSVec2<T> NSVec2< T >::hw ( ) const [inline]

6.43.2.11 template<class T> T NSVec2< T >::length ( ) const [inline]

6.43.2.12 template<class T> T NSVec2< T >::lengthSq ( ) const [inline]

6.43.2.13 template<class T> template<class T2 > NSVec2<T>& NSVec2< T >::lerp (const NSVec2< T > & *vec,* const T2 & *scalingFactor*) [inline]

6.43.2.14 template<class T> T NSVec2< T >::max ( ) const [inline]

6.43.2.15 template<class T> NSVec2<T>& NSVec2< T >::maximize (const NSVec2< T > & *rhs*) [inline]

6.43.2.16 template<class T> T NSVec2< T >::min ( ) const [inline]

6.43.2.17 template<class T> NSVec2<T>& NSVec2< T >::minimize (const NSVec2< T > & *rhs*) [inline]

6.43.2.18 template<class T> NSVec2<T>& NSVec2< T >::normalize ( ) [inline]

6.43.2.19 template<class T> bool NSVec2< T >::operator!= (const NSVec2< T > & *pRHS*) const [inline]

6.43.2.20 template<class T> bool NSVec2< T >::operator!= (const T & *pRHS*) const [inline]

6.43.2.21 template<class T> NSVec2<T> NSVec2< T >::operator% (const NSVec2< T > & *pRHS*) const [inline]

6.43.2.22 template<class T> NSVec2<T>& NSVec2< T >::operator%= (const NSVec2< T > & *pRHS*) [inline]

6.43.2.23 template<class T> T NSVec2< T >::operator∗ (const NSVec2< T > & *pRHS*) const [inline]

6.43.2.24 template<class T> NSVec2<T> NSVec2< T >::operator∗ (const T & *pRHS*) const [inline]

6.43.2.25 template<class T> NSVec2<T>& NSVec2< T >::operator∗= (const T & *pRHS*) [inline]

6.43.2.26 template<class T> NSVec2<T> NSVec2< T >::operator+ (const NSVec2< T > & *pRHS*) const [inline]

6.43.2.27 template<class T> NSVec2<T> NSVec2< T >::operator++ (int32_t) [inline]

6.43.2.28 template<class T> NSVec2<T>& NSVec2< T >::operator++ ( ) [inline]

6.43.2.29 template<class T> NSVec2<T>& NSVec2< T >::operator+= (const NSVec2< T > & *pRHS*) [inline]

6.43.2.30 template<class T> NSVec2<T> NSVec2< T >::operator- (const NSVec2< T > & *pRHS*) const [inline]

6.43.2.31 template<class T> NSVec2<T> NSVec2< T >::operator-- (int32_t) [inline]

6.43.2.32 template<class T> NSVec2<T>& NSVec2< T >::operator-- ( ) [inline]

6.43.2.33 template<class T> NSVec2<T>& NSVec2< T >::operator-= (const NSVec2< T > & *pRHS*) [inline]

6.43.2.34 template<class T> NSVec2<T> NSVec2< T >::operator/ (const NSVec2< T > & *pRHS*) const [inline]

6.43.2.35 template<class T> NSVec2<T> NSVec2< T >::operator/ (const T & *pRHS*) const [inline]

6.43.2.36 template<class T> NSVec2<T>& NSVec2< T >::operator/= (const NSVec2< T > & *pRHS*) [inline]

6.43.2.37 template<class T> NSVec2<T>& NSVec2< T >::operator/= (const T & *pRHS*) [inline]

6.43.2.38 template<class T> NSVec2<T>& NSVec2< T >::operator= (const NSVec2< T > & *pRHS*) [inline]

6.43.2.39 template<class T> bool NSVec2< T >::operator== (const NSVec2< T > & *pRHS*) const [inline]

6.43.2.40 template<class T> bool NSVec2< T >::operator== (const T & *pRHS*) const [inline]

6.43.2.41 template<class T> const T& NSVec2< T >::operator[] (const uint32_t & *pVal*) const [inline]

6.43.2.42 template<class T> T& NSVec2< T >::operator[] (const uint32_t & *pVal*) [inline]

6.43.2.43 template<class T> nsmat2<T> NSVec2< T >::operator^ (const NSVec2< T > & *pRHS*) const [inline]

6.43.2.44 template<class T> NSVec2<T> NSVec2< T >::polar (bool *pRads* = false) const [inline]

6.43.2.45 template<class T> NSVec2<T>& NSVec2< T >::projectOn (const NSVec2< T > & *vec*) [inline]

6.43.2.46 template<class T> NSVec2<T>& NSVec2< T >::projectOnPlane (const NSVec2< T > & *planeNormal*) [inline]

6.43.2.47 template<class T> NSVec2<T>& NSVec2< T >::reflect (const NSVec2< T > & *normal*) [inline]

6.43.2.48 template<class T> template<class T2 > NSVec2<T>& NSVec2< T >::rotate (const T2 & *angle_*) [inline]

6.43.2.49 template<class T> NSVec2<T>& NSVec2< T >::round ( ) [inline]

6.43.2.50 template<class T> NSVec2<T>& NSVec2< T >::roundToZero ( ) [inline]

6.43.2.51 template<class T> NSVec2<T>& NSVec2< T >::scalingFrom (const nsmat2< T > & *transform2d*) [inline]

6.43.2.52 template<class T> NSVec2<T>& NSVec2< T >::scalingFrom (const nsmat3< T > & *transform2d*) [inline]

6.43.2.53 template<class T> NSVec2<T>& NSVec2< T >::set (const T & *pVal*) [inline]

6.43.2.54 template<class T> NSVec2<T>& NSVec2< T >::set (const T & *pX,* const T & *pY*) [inline]

6.43.2.55 template<class T> NSVec2<T>& NSVec2< T >::setFromPolar (const T & *pMag,* T *angle_,* bool *pRads* = false) [inline]

6.43.2.56 template<class T> NSVec2<T>& NSVec2< T >::setFromPolar (const NSVec2< T > & *pVec,* bool *pRads* = false) [inline]

6.43.2.57 template<class T> NSVec2<T>& NSVec2< T >::setLength (const T & *len*) [inline]

6.43.2.58 template<class T> NSVec2<T> NSVec2< T >::ss ( ) const [inline]

6.43.2.59 template<class T> std::string NSVec2< T >::toString (bool *pPolar* = false) [inline]

6.43.2.60 template<class T> NSVec2<T>& NSVec2< T >::translationFrom (const nsmat3< T > & *transform2d*) [inline]

6.43.2.61 template<class T> NSVec2<T> NSVec2< T >::ts ( ) const [inline]

6.43.2.62 template<class T> NSVec2<T> NSVec2< T >::tt ( ) const [inline]

6.43.2.63 template<class T> NSVec2<T> NSVec2< T >::uu ( ) const [inline]

6.43.2.64 template<class T> NSVec2<T> NSVec2< T >::vu ( ) const [inline]

6.43.2.65 template<class T> NSVec2<T> NSVec2< T >::vv ( ) const [inline]

6.43.2.66 template<class T> NSVec2<T> NSVec2< T >::ww ( ) const [inline]

6.43.2.67 template<class T> NSVec2<T> NSVec2< T >::xx ( ) const [inline]

6.43.2.68 template<class T> NSVec2<T> NSVec2< T >::yx ( ) const [inline]

6.43.2.69 template<class T> NSVec2<T> NSVec2< T >::yy ( ) const [inline]

6.43.3 Member Data Documentation

6.43.3.1 union { ... }

6.43.3.2 template<class T> T NSVec2< T >::data[2]

6.43.3.3 template<class T> T NSVec2< T >::h

6.43.3.4 template<class T> T NSVec2< T >::s

6.43.3.5 template<class T> T NSVec2< T >::t

6.43.3.6 template<class T> T NSVec2< T >::u

6.43.3.7 template<class T> T NSVec2< T >::v

6.43.3.8 template<class T> T NSVec2< T >::w

6.43.3.9 template<class T> T NSVec2< T >::x

6.43.3.10 template<class T> T NSVec2< T >::y

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

- /home/dprandle/Documents/code/ctrlmod/include/nsvec2.h

6.44 NSVec3< T > Struct Template Reference

`#include <nsvec2.h>`

Public Types

- enum CoordSys { Cartesian, Cylindrical, Spherical }
- enum EulerOrder {
  XYZ, XZY, YXZ, YZX,
  ZXY, ZYX }

Public Member Functions

- NSVec3 (const NSVec3< T > &copy)
- NSVec3 (const T &val=static_cast< T >(0))
- NSVec3 (const NSVec2< T > &xy, const T &z_)
- NSVec3 (const T &x_, const NSVec2< T > &yz)
- NSVec3 (const T &x_, const T &y_, const T &z_=static_cast< T >(0))
- NSVec3< T > & abs ()
- T angleTo (const NSVec3< T > pVec, bool pRads=false) const
- NSVec3< T > & ceil ()
- NSVec3< T > & clamp (const T &min=static_cast< T >(0), const T &max=static_cast< T >(0))
- NSVec3< T > & cross (const NSVec3< T > &crossWith)
- NSVec3< T > cylindrical (bool pRads=false) const
- T distanceTo (const NSVec3< T > &pVec) const
- NSVec3< T > & eulerFrom (const NSVec4< T > &axisAngle, EulerOrder order=XYZ, bool pRads=false)
- NSVec3< T > & eulerFrom (const nsquat< T > &orientation, EulerOrder order, bool rads)
- NSVec3< T > & eulerFrom (const nsmat3< T > &rotationMat3, EulerOrder order=XYZ, bool pRads=false)
- NSVec3< T > & eulerFrom (const nsmat4< T > &transform, EulerOrder order=XYZ, bool pRads=false)
- NSVec3< T > & eulerFrom (const NSVec3< T > &vec, const NSVec3< T > &toVec, EulerOrder order=XYZ, bool pRads=false)
- NSVec3< T > & floor ()
- NSVec3< T > & fract ()
- NSVec3< T > & from (CoordSys coordSys, const NSVec3< T > &vec, bool pRads=false)
- T length () const
- T lengthSq () const
- template<class T2 >
  NSVec3< T > & lerp (const NSVec3< T > &vec, const T2 &scalingFactor)
- T min ()
- NSVec3< T > & minimize (const NSVec3< T > &rhs)
- T max ()
- NSVec3< T > & maximize (const NSVec3< T > &rhs)
- NSVec3< T > & normalize ()
- NSVec3< T > & projectOn (const NSVec3< T > &vec)
- NSVec3< T > & projectOnPlane (const NSVec3< T > &planeNormal)
- NSVec3< T > & reflect (const NSVec3< T > &normal)
- NSVec3< T > & round ()
- NSVec3< T > & roundToZero ()
- NSVec3< T > & scalingFrom (const nsmat3< T > &transform)
- NSVec3< T > & scalingFrom (const nsmat4< T > &transform)
- NSVec3< T > & translationFrom (const nsmat4< T > &transform)
- NSVec3< T > & set (const T &pVal)
- NSVec3< T > & set (const T &pX, const T &pY, const T &pZ)
- NSVec3< T > & set (const NSVec2< T > &xy, const T &pZ)
- NSVec3< T > & set (const T &pX, const NSVec2< T > &yz)
- NSVec3< T > & setLength (const T &len)
- NSVec3< T > spherical (bool pRads=false) const
- std::string toString (CoordSys disp=Cartesian) const

- NSVec3< T > operator+ (const NSVec3< T > &rhs) const
- NSVec3< T > operator- (const NSVec3< T > &rhs) const
- T operator* (const NSVec3< T > &rhs) const
- nsmat3< T > operator^ (const NSVec3< T > &pRHS) const
- NSVec3< T > operator% (const NSVec3< T > &rhs) const
- NSVec3< T > operator/ (const NSVec3< T > &rhs) const
- NSVec3< T > operator* (const T &rhs) const
- NSVec3< T > operator/ (const T &rhs) const
- NSVec3< T > & operator= (const NSVec3< T > &rhs)
- NSVec3< T > operator++ (int32_t)
- NSVec3< T > operator-- (int32_t)
- NSVec3< T > & operator++ ()
- NSVec3< T > & operator-- ()
- NSVec3< T > & operator+= (const NSVec3< T > &rhs)
- NSVec3< T > & operator-= (const NSVec3< T > &rhs)
- NSVec3< T > & operator%= (const NSVec3< T > &rhs)
- NSVec3< T > & operator/= (const NSVec3< T > &rhs)
- NSVec3< T > & operator*= (const T &rhs)
- NSVec3< T > & operator/= (const T &rhs)
- bool operator== (const NSVec3< T > &rhs) const
- bool operator!= (const NSVec3< T > &rhs) const
- bool operator== (const T &rhs) const
- bool operator!= (const T &rhs) const
- const T & operator[] (const uint32_t &pVal) const
- T & operator[] (const uint32_t &pVal)
- NSVec3< T > xxx () const
- NSVec3< T > xxy () const
- NSVec3< T > xxz () const
- NSVec3< T > xyx () const
- NSVec3< T > xyy () const
- NSVec3< T > xzx () const
- NSVec3< T > xzy () const
- NSVec3< T > xzz () const
- NSVec3< T > yxx () const
- NSVec3< T > yxy () const
- NSVec3< T > yxz () const
- NSVec3< T > yyx () const
- NSVec3< T > yyy () const
- NSVec3< T > yyz () const
- NSVec3< T > yzx () const
- NSVec3< T > yzy () const
- NSVec3< T > yzz () const
- NSVec3< T > zxx () const
- NSVec3< T > zxy () const
- NSVec3< T > zxz () const
- NSVec3< T > zyx () const
- NSVec3< T > zyy () const
- NSVec3< T > zyz () const
- NSVec3< T > zzx () const
- NSVec3< T > zzy () const
- NSVec3< T > zzz () const
- NSVec3< T > rrr () const
- NSVec3< T > rrg () const
- NSVec3< T > rrb () const
- NSVec3< T > rgr () const

- NSVec3< T > rgg () const
- NSVec3< T > rbr () const
- NSVec3< T > rbg () const
- NSVec3< T > rbb () const
- NSVec3< T > grr () const
- NSVec3< T > grg () const
- NSVec3< T > grb () const
- NSVec3< T > ggr () const
- NSVec3< T > ggg () const
- NSVec3< T > ggb () const
- NSVec3< T > gbr () const
- NSVec3< T > gbg () const
- NSVec3< T > gbb () const
- NSVec3< T > brr () const
- NSVec3< T > brg () const
- NSVec3< T > brb () const
- NSVec3< T > bgr () const
- NSVec3< T > bgg () const
- NSVec3< T > bgb () const
- NSVec3< T > bbr () const
- NSVec3< T > bbg () const
- NSVec3< T > bbb () const
- NSVec3< T > sss () const
- NSVec3< T > sst () const
- NSVec3< T > ssp () const
- NSVec3< T > sts () const
- NSVec3< T > stt () const
- NSVec3< T > sps () const
- NSVec3< T > spt () const
- NSVec3< T > spp () const
- NSVec3< T > tss () const
- NSVec3< T > tst () const
- NSVec3< T > tsp () const
- NSVec3< T > tts () const
- NSVec3< T > ttt () const
- NSVec3< T > ttp () const
- NSVec3< T > tps () const
- NSVec3< T > tpt () const
- NSVec3< T > tpp () const
- NSVec3< T > pss () const
- NSVec3< T > pst () const
- NSVec3< T > psp () const
- NSVec3< T > pts () const
- NSVec3< T > ptt () const
- NSVec3< T > ptp () const
- NSVec3< T > pps () const
- NSVec3< T > ppt () const
- NSVec3< T > ppp () const
- NSVec2< T > xx () const
- NSVec2< T > xy () const
- NSVec2< T > xz () const
- NSVec2< T > yx () const
- NSVec2< T > yy () const
- NSVec2< T > yz () const
- NSVec2< T > zx () const

- NSVec2< T > zy () const
- NSVec2< T > zz () const
- NSVec2< T > rr () const
- NSVec2< T > rg () const
- NSVec2< T > rb () const
- NSVec2< T > gr () const
- NSVec2< T > gg () const
- NSVec2< T > gb () const
- NSVec2< T > br () const
- NSVec2< T > bg () const
- NSVec2< T > bb () const
- NSVec2< T > ss () const
- NSVec2< T > st () const
- NSVec2< T > sp () const
- NSVec2< T > ts () const
- NSVec2< T > tt () const
- NSVec2< T > tp () const
- NSVec2< T > ps () const
- NSVec2< T > pt () const
- NSVec2< T > pp () const

Public Attributes

- union {
  T data [3]
  struct {
  T x
  T y
  T z
  }
  struct {
  T r
  T g
  T b
  }
  struct {
  T s
  T t
  T p
  }
  struct {
  T P
  T I
  T D
  }
  };

6.44.1 Member Enumeration Documentation

6.44.1.1 template<class T> enum NSVec3::CoordSys

Enumerator

Cartesian

Cylindrical

Spherical

6.44.1.2 template<class T> enum NSVec3::EulerOrder

Enumerator

XYZ

XZY

YXZ

YZX

ZXY

ZYX

6.44.2 Constructor & Destructor Documentation

6.44.2.1 template<class T> NSVec3< T >::NSVec3 (const NSVec3< T > & *copy*) [inline]

6.44.2.2 template<class T> NSVec3< T >::NSVec3 (const T & *val* = static_cast<T>(0)) [inline]

6.44.2.3 template<class T> NSVec3< T >::NSVec3 (const NSVec2< T > & *xy*, const T & *z_*) [inline]

6.44.2.4 template<class T> NSVec3< T >::NSVec3 (const T & *x_*, const NSVec2< T > & *yz*) [inline]

6.44.2.5 template<class T> NSVec3< T >::NSVec3 (const T & *x_*, const T & *y_*, const T & *z_* = static_cast<T>(0)) [inline]

6.44.3 Member Function Documentation

6.44.3.1 template<class T> NSVec3<T>& NSVec3< T >::abs ( ) [inline]

6.44.3.2 template<class T> T NSVec3< T >::angleTo (const NSVec3< T > *pVec*, bool *pRads* = false) const [inline]

6.44.3.3 template<class T> NSVec2<T> NSVec3< T >::bb ( ) const [inline]

6.44.3.4 template<class T> NSVec3<T> NSVec3< T >::bbb ( ) const [inline]

6.44.3.5 template<class T> NSVec3<T> NSVec3< T >::bbg ( ) const [inline]

6.44.3.6 template<class T> NSVec3<T> NSVec3< T >::bbr ( ) const [inline]

6.44.3.7 template<class T> NSVec2<T> NSVec3< T >::bg ( ) const [inline]

6.44.3.8 template<class T> NSVec3<T> NSVec3< T >::bgb ( ) const [inline]

6.44.3.9 template<class T> NSVec3<T> NSVec3< T >::bgg ( ) const [inline]

6.44.3.10 template<class T> NSVec3<T> NSVec3< T >::bgr ( ) const [inline]

6.44.3.11 template<class T> NSVec2<T> NSVec3< T >::br ( ) const [inline]

6.44.3.12 template<class T> NSVec3<T> NSVec3< T >::brb ( ) const [inline]

6.44.3.13 template<class T> NSVec3<T> NSVec3< T >::brg ( ) const [inline]

6.44.3.14 template<class T> NSVec3<T> NSVec3< T >::brr ( ) const [inline]

6.44.3.15 template<class T> NSVec3<T>& NSVec3< T >::ceil ( ) [inline]

6.44.3.16 template<class T> NSVec3<T>& NSVec3< T >::clamp (const T & *min* = static_cast<T>(0), const T & *max* = static_cast<T>(0)) [inline]

6.44.3.17 template<class T> NSVec3<T>& NSVec3< T >::cross (const NSVec3< T > & *crossWith*) [inline]

6.44.3.18 template<class T> NSVec3<T> NSVec3< T >::cylindrical (bool *pRads* = false) const [inline]

6.44.3.19 template<class T> T NSVec3< T >::distanceTo (const NSVec3< T > & *pVec*) const [inline]

6.44.3.20 template<class T> NSVec3<T>& NSVec3< T >::eulerFrom (const NSVec4< T > & *axisAngle*, EulerOrder *order* = XYZ, bool *pRads* = false) [inline]

6.44.3.21 template<class T> NSVec3<T>& NSVec3< T >::eulerFrom (const nsquat< T > & *orientation*, EulerOrder *order*, bool *rads*) [inline]

6.44.3.22 template<class T> NSVec3<T>& NSVec3< T >::eulerFrom (const nsmat3< T > & *rotationMat3*, EulerOrder *order* = XYZ, bool *pRads* = false) [inline]

6.44.3.23 template<class T> NSVec3<T>& NSVec3< T >::eulerFrom (const nsmat4< T > & *transform*, EulerOrder *order* = XYZ, bool *pRads* = false) [inline]

6.44.3.24 template<class T> NSVec3<T>& NSVec3< T >::eulerFrom (const NSVec3< T > & *vec*, const NSVec3< T > & *toVec*, EulerOrder *order* = XYZ, bool *pRads* = false) [inline]

6.44.3.25 template<class T> NSVec3<T>& NSVec3< T >::floor ( ) [inline]

6.44.3.26 template<class T> NSVec3<T>& NSVec3< T >::fract ( ) [inline]

6.44.3.27 template<class T> NSVec3<T>& NSVec3< T >::from (CoordSys *coordSys*, const NSVec3< T > & *vec*, bool *pRads* = false) [inline]

6.44.3.28 template<class T> NSVec2<T> NSVec3< T >::gb ( ) const [inline]

6.44.3.29 template<class T> NSVec3<T> NSVec3< T >::gbb ( ) const [inline]

6.44.3.30 template<class T> NSVec3<T> NSVec3< T >::gbg ( ) const [inline]

6.44.3.31 template<class T> NSVec3<T> NSVec3< T >::gbr ( ) const [inline]

6.44.3.32 template<class T> NSVec2<T> NSVec3< T >::gg ( ) const [inline]

6.44.3.33 template<class T> NSVec3<T> NSVec3< T >::ggb ( ) const [inline]

6.44.3.34 template<class T> NSVec3<T> NSVec3< T >::ggg ( ) const [inline]

6.44.3.35 template<class T> NSVec3<T> NSVec3< T >::ggr ( ) const [inline]

6.44.3.36 template<class T> NSVec2<T> NSVec3< T >::gr ( ) const [inline]

6.44.3.37 template<class T> NSVec3<T> NSVec3< T >::grb ( ) const [inline]

6.44.3.38 template<class T> NSVec3<T> NSVec3< T >::grg ( ) const [inline]

6.44.3.39 template<class T> NSVec3<T> NSVec3< T >::grr ( ) const [inline]

6.44.3.40 template<class T> T NSVec3< T >::length ( ) const [inline]

6.44.3.41 template<class T> T NSVec3< T >::lengthSq ( ) const [inline]

6.44.3.42 template<class T> template<class T2 > NSVec3<T>& NSVec3< T >::lerp (const NSVec3< T > & *vec,* const T2 & *scalingFactor*) [inline]

6.44.3.43 template<class T> T NSVec3< T >::max ( ) [inline]

6.44.3.44 template<class T> NSVec3<T>& NSVec3< T >::maximize (const NSVec3< T > & *rhs*) [inline]

6.44.3.45 template<class T> T NSVec3< T >::min ( ) [inline]

6.44.3.46 template<class T> NSVec3<T>& NSVec3< T >::minimize (const NSVec3< T > & *rhs*) [inline]

6.44.3.47 template<class T> NSVec3<T>& NSVec3< T >::normalize ( ) [inline]

6.44.3.48 template<class T> bool NSVec3< T >::operator!= (const NSVec3< T > & *rhs*) const [inline]

6.44.3.49 template<class T> bool NSVec3< T >::operator!= (const T & *rhs*) const [inline]

6.44.3.50 template<class T> NSVec3<T> NSVec3< T >::operator% (const NSVec3< T > & *rhs*) const [inline]

6.44.3.51 template<class T> NSVec3<T>& NSVec3< T >::operator%= (const NSVec3< T > & *rhs*) [inline]

6.44.3.52 template<class T> T NSVec3< T >::operator* (const NSVec3< T > & *rhs*) const [inline]

6.44.3.53 template<class T> NSVec3<T> NSVec3< T >::operator* (const T & *rhs*) const [inline]

6.44.3.54 template<class T> NSVec3<T>& NSVec3< T >::operator*= (const T & *rhs*) [inline]

6.44.3.55 template<class T> NSVec3<T> NSVec3< T >::operator+ (const NSVec3< T > & *rhs*) const [inline]

6.44.3.56 template<class T> NSVec3<T> NSVec3< T >::operator++ (int32_t) [inline]

6.44.3.57 template<class T> NSVec3<T>& NSVec3< T >::operator++ ( ) [inline]

6.44.3.58 template<class T> NSVec3<T>& NSVec3< T >::operator+= (const NSVec3< T > & *rhs*) [inline]

6.44.3.59 template<class T> NSVec3<T> NSVec3< T >::operator- (const NSVec3< T > & *rhs*) const [inline]

6.44.3.60 template<class T> NSVec3<T> NSVec3< T >::operator-- (int32_t) [inline]

6.44.3.61 template<class T> NSVec3<T>& NSVec3< T >::operator-- ( ) [inline]

6.44.3.62 template<class T> NSVec3<T>& NSVec3< T >::operator-= (const NSVec3< T > & *rhs*) [inline]

6.44.3.63 template<class T> NSVec3<T> NSVec3< T >::operator/ (const NSVec3< T > & *rhs*) const [inline]

6.44.3.64 template<class T> NSVec3<T> NSVec3< T >::operator/ (const T & *rhs*) const [inline]

6.44.3.65 template<class T> NSVec3<T>& NSVec3< T >::operator/= (const NSVec3< T > & *rhs*) [inline]

6.44.3.66 template<class T> NSVec3<T>& NSVec3< T >::operator/= (const T & *rhs*) [inline]

6.44.3.67 template<class T> NSVec3<T>& NSVec3< T >::operator= (const NSVec3< T > & *rhs*) [inline]

6.44.3.68 template<class T> bool NSVec3< T >::operator== (const NSVec3< T > & *rhs*) const [inline]

6.44.3.69 template<class T> bool NSVec3< T >::operator== (const T & *rhs*) const [inline]

6.44.3.70 template<class T> const T& NSVec3< T >::operator[] (const uint32_t & *pVal*) const [inline]

6.44.3.71 template<class T> T& NSVec3< T >::operator[] (const uint32_t & *pVal*) [inline]

6.44.3.72 template<class T> nsmat3<T> NSVec3< T >::operator^ (const NSVec3< T > & *pRHS*) const [inline]

6.44.3.73 template<class T> NSVec2<T> NSVec3< T >::pp ( ) const [inline]

6.44.3.74 template<class T> NSVec3<T> NSVec3< T >::ppp ( ) const [inline]

6.44.3.75 template<class T> NSVec3<T> NSVec3< T >::pps ( ) const [inline]

6.44.3.76 template<class T> NSVec3<T> NSVec3< T >::ppt ( ) const [inline]

6.44.3.77 template<class T> NSVec3<T>& NSVec3< T >::projectOn (const NSVec3< T > & *vec*) [inline]

6.44.3.78 template<class T> NSVec3<T>& NSVec3< T >::projectOnPlane (const NSVec3< T > & *planeNormal*) [inline]

6.44.3.79 template<class T> NSVec2<T> NSVec3< T >::ps ( ) const [inline]

6.44.3.80 template<class T> NSVec3<T> NSVec3< T >::psp ( ) const [inline]

6.44.3.81 template<class T> NSVec3<T> NSVec3< T >::pss ( ) const [inline]

6.44.3.82 template<class T> NSVec3<T> NSVec3< T >::pst ( ) const [inline]

6.44.3.83 template<class T> NSVec2<T> NSVec3< T >::pt ( ) const [inline]

6.44.3.84 template<class T> NSVec3<T> NSVec3< T >::ptp ( ) const [inline]

6.44.3.85 template<class T> NSVec3<T> NSVec3< T >::pts ( ) const [inline]

6.44.3.86 template<class T> NSVec3<T> NSVec3< T >::ptt ( ) const [inline]

6.44.3.87 template<class T> NSVec2<T> NSVec3< T >::rb ( ) const [inline]

6.44.3.88 template<class T> NSVec3<T> NSVec3< T >::rbb ( ) const [inline]

6.44.3.89 template<class T> NSVec3<T> NSVec3< T >::rbg ( ) const [inline]

6.44.3.90 template<class T> NSVec3<T> NSVec3< T >::rbr ( ) const [inline]

6.44.3.91 template<class T> NSVec3<T>& NSVec3< T >::reflect (const NSVec3< T > & *normal*) [inline]

6.44.3.92 template<class T> NSVec2<T> NSVec3< T >::rg ( ) const [inline]

6.44.3.93 template<class T> NSVec3<T> NSVec3< T >::rgg ( ) const [inline]

6.44.3.94 template<class T> NSVec3<T> NSVec3< T >::rgr ( ) const [inline]

6.44.3.95 template<class T> NSVec3<T>& NSVec3< T >::round ( ) [inline]

6.44.3.96 template<class T> NSVec3<T>& NSVec3< T >::roundToZero ( ) [inline]

6.44.3.97 template<class T> NSVec2<T> NSVec3< T >::rr ( ) const [inline]

6.44.3.98 template<class T> NSVec3<T> NSVec3< T >::rrb ( ) const [inline]

6.44.3.99 template<class T> NSVec3<T> NSVec3< T >::rrg ( ) const [inline]

6.44.3.100 template<class T> NSVec3<T> NSVec3< T >::rrr ( ) const [inline]

6.44.3.101 template<class T> NSVec3<T>& NSVec3< T >::scalingFrom (const nsmat3< T > & *transform*) [inline]

6.44.3.102 template<class T> NSVec3<T>& NSVec3< T >::scalingFrom (const nsmat4< T > & *transform*) [inline]

6.44.3.103 template<class T> NSVec3<T>& NSVec3< T >::set (const T & *pVal*) [inline]

6.44.3.104 template<class T> NSVec3<T>& NSVec3< T >::set (const T & *pX*, const T & *pY*, const T & *pZ*) [inline]

6.44.3.105 template<class T> NSVec3<T>& NSVec3< T >::set (const NSVec2< T > & *xy*, const T & *pZ*) [inline]

6.44.3.106 template<class T> NSVec3<T>& NSVec3< T >::set (const T & *pX*, const NSVec2< T > & *yz*) [inline]

6.44.3.107 template<class T> NSVec3<T>& NSVec3< T >::setLength (const T & *len*) [inline]

6.44.3.108 template<class T> NSVec2<T> NSVec3< T >::sp ( ) const [inline]

6.44.3.109 template<class T> NSVec3<T> NSVec3< T >::spherical (bool *pRads* = false) const [inline]

6.44.3.110 template<class T> NSVec3<T> NSVec3< T >::spp ( ) const [inline]

6.44.3.111 template<class T> NSVec3<T> NSVec3< T >::sps ( ) const [inline]

6.44.3.112 template<class T> NSVec3<T> NSVec3< T >::spt ( ) const [inline]

6.44.3.113 template<class T> NSVec2<T> NSVec3< T >::ss ( ) const [inline]

6.44.3.114 template<class T> NSVec3<T> NSVec3< T >::ssp ( ) const [inline]

6.44.3.115 template<class T> NSVec3<T> NSVec3< T >::sss ( ) const [inline]

6.44.3.116 template<class T> NSVec3<T> NSVec3< T >::sst ( ) const [inline]

6.44.3.117 template<class T> NSVec2<T> NSVec3< T >::st ( ) const [inline]

6.44.3.118 template<class T> NSVec3<T> NSVec3< T >::sts ( ) const [inline]

6.44.3.119 template<class T> NSVec3<T> NSVec3< T >::stt ( ) const [inline]

6.44.3.120 template<class T> std::string NSVec3< T >::toString (CoordSys *disp* = Cartesian) const [inline]

6.44.3.121 template<class T> NSVec2<T> NSVec3< T >::tp ( ) const [inline]

6.44.3.122 template<class T> NSVec3<T> NSVec3< T >::tpp ( ) const [inline]

6.44.3.123 template<class T> NSVec3<T> NSVec3< T >::tps ( ) const [inline]

6.44.3.124 template<class T> NSVec3<T> NSVec3< T >::tpt ( ) const [inline]

6.44.3.125 template<class T> NSVec3<T>& NSVec3< T >::translationFrom (const nsmat4< T > & *transform*) [inline]

6.44.3.126 template<class T> NSVec2<T> NSVec3< T >::ts ( ) const [inline]

6.44.3.127 template<class T> NSVec3<T> NSVec3< T >::tsp ( ) const [inline]

6.44.3.128 template<class T> NSVec3<T> NSVec3< T >::tss ( ) const [inline]

6.44.3.129 template<class T> NSVec3<T> NSVec3< T >::tst ( ) const [inline]

6.44.3.130 template<class T> NSVec2<T> NSVec3< T >::tt ( ) const [inline]

6.44.3.131 template<class T> NSVec3<T> NSVec3< T >::ttp ( ) const [inline]

6.44.3.132 template<class T> NSVec3<T> NSVec3< T >::tts ( ) const [inline]

6.44.3.133 template<class T> NSVec3<T> NSVec3< T >::ttt ( ) const [inline]

6.44.3.134 template<class T> NSVec2<T> NSVec3< T >::xx ( ) const [inline]

6.44.3.135 template<class T> NSVec3<T> NSVec3< T >::xxx ( ) const [inline]

6.44.3.136 template<class T> NSVec3<T> NSVec3< T >::xxy ( ) const [inline]

6.44.3.137 template<class T> NSVec3<T> NSVec3< T >::xxz ( ) const [inline]

6.44.3.138 template<class T> NSVec2<T> NSVec3< T >::xy ( ) const [inline]

6.44.3.139 template<class T> NSVec3<T> NSVec3< T >::xyx ( ) const [inline]

6.44.3.140 template<class T> NSVec3<T> NSVec3< T >::xyy ( ) const [inline]

6.44.3.141 template<class T> NSVec2<T> NSVec3< T >::xz ( ) const [inline]

6.44.3.142 template<class T> NSVec3<T> NSVec3< T >::xzx ( ) const [inline]

6.44.3.143 template<class T> NSVec3<T> NSVec3< T >::xzy ( ) const [inline]

6.44.3.144 template<class T> NSVec3<T> NSVec3< T >::xzz ( ) const [inline]

6.44.3.145 template<class T> NSVec2<T> NSVec3< T >::yx ( ) const [inline]

6.44.3.146 template<class T> NSVec3<T> NSVec3< T >::yxx ( ) const [inline]

6.44.3.147 template<class T> NSVec3<T> NSVec3< T >::yxy ( ) const [inline]

6.44.3.148 template<class T> NSVec3<T> NSVec3< T >::yxz ( ) const [inline]

6.44.3.149 template<class T> NSVec2<T> NSVec3< T >::yy ( ) const [inline]

6.44.3.150 template<class T> NSVec3<T> NSVec3< T >::yyx ( ) const [inline]

6.44.3.151 template<class T> NSVec3<T> NSVec3< T >::yyy ( ) const [inline]

6.44.3.152 template<class T> NSVec3<T> NSVec3< T >::yyz ( ) const [inline]

6.44.3.153 template<class T> NSVec2<T> NSVec3< T >::yz ( ) const [inline]

6.44.3.154 template<class T> NSVec3<T> NSVec3< T >::yzx ( ) const [inline]

6.44.3.155 template<class T> NSVec3<T> NSVec3< T >::yzy ( ) const [inline]

6.44.3.156 template<class T> NSVec3<T> NSVec3< T >::yzz ( ) const [inline]

6.44.3.157 template<class T> NSVec2<T> NSVec3< T >::zx ( ) const [inline]

6.44.3.158 template<class T> NSVec3<T> NSVec3< T >::zxx ( ) const [inline]

6.44.3.159 template<class T> NSVec3<T> NSVec3< T >::zxy ( ) const [inline]

6.44.3.160 template<class T> NSVec3<T> NSVec3< T >::zxz ( ) const [inline]

6.44.3.161 template<class T> NSVec2<T> NSVec3< T >::zy ( ) const [inline]

6.44.3.162 template<class T> NSVec3<T> NSVec3< T >::zyx ( ) const [inline]

6.44.3.163 template<class T> NSVec3<T> NSVec3< T >::zyy ( ) const [inline]

6.44.3.164 template<class T> NSVec3<T> NSVec3< T >::zyz ( ) const [inline]

6.44.3.165 template<class T> NSVec2<T> NSVec3< T >::zz ( ) const [inline]

6.44.3.166 template<class T> NSVec3<T> NSVec3< T >::zzx ( ) const [inline]

6.44.3.167 template<class T> NSVec3<T> NSVec3< T >::zzy ( ) const [inline]

6.44.3.168 template<class T> NSVec3<T> NSVec3< T >::zzz ( ) const [inline]

6.44.4 Member Data Documentation

6.44.4.1 union { ... }

6.44.4.2 template<class T> T NSVec3< T >::b

6.44.4.3 template<class T> T NSVec3< T >::D

6.44.4.4 template<class T> T NSVec3< T >::data[3]

6.44.4.5 template<class T> T NSVec3< T >::g

6.44.4.6 template<class T> T NSVec3< T >::l

6.44.4.7 template<class T> T NSVec3< T >::p

6.44.4.8 template<class T> T NSVec3< T >::P

6.44.4.9 template<class T> T NSVec3< T >::r

6.44.4.10 template<class T> T NSVec3< T >::s

6.44.4.11 template<class T> T NSVec3< T >::t

6.44.4.12 template<class T> T NSVec3< T >::x

6.44.4.13 template<class T> T NSVec3< T >::y

6.44.4.14 template<class T> T NSVec3< T >::z

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

- /home/dprandle/Documents/code/ctrlmod/include/nsvec2.h
- /home/dprandle/Documents/code/ctrlmod/include/nsvec3.h

6.45 NSVec4< T > Struct Template Reference

`#include <nsvec2.h>`

Public Member Functions

- NSVec4 (const NSVec4< T > &copy)
- NSVec4 (const T &val=static_cast< T >(0))
- NSVec4 (const T &pX, const T &pY, const T &pZ=static_cast< T >(0), const T &pW=static_cast< T >(0))
- NSVec4 (const NSVec3< T > &xyz, const T &pW=static_cast< T >(1))
- NSVec4 (const T &pX, const NSVec3< T > &yzw)
- NSVec4 (const NSVec2< T > &xy, const T &pZ=static_cast< T >(0), const T &pW=static_cast< T >(0))
- NSVec4 (const T &pX, const NSVec2< T > &yz, const T &pW=static_cast< T >(0))
- NSVec4 (const T &pX, const T &pY, const NSVec2< T > &zw)
- NSVec3< T > & abs ()
- NSVec4< T > & axisAngleFrom (const NSVec3< T > &euler, typename NSVec3< T >::EulerOrder order, bool rads=false)
- NSVec4< T > & axisAngleFrom (const nsquat< T > &orientation, bool rads=false)
- NSVec4< T > & axisAngleFrom (const nsmat3< T > &rotationMat3, bool rads=false)
- NSVec4< T > & axisAngleFrom (const nsmat4< T > &transform, bool rads=false)
- NSVec4< T > & axisAngleFrom (const NSVec3< T > &vec, const NSVec3< T > &toVec, bool rads=false)
- NSVec4< T > & ceil ()
- NSVec4< T > & clamp (const T &min=static_cast< T >(0), const T &max=static_cast< T >(0))
- T distanceTo (const NSVec4< T > &pVec) const
- NSVec4< T > & floor ()
- NSVec4< T > & fract ()
- T length () const
- T lengthSq () const

- template<class T2 >
  NSVec4< T > & lerp (const NSVec4< T > &vec, const T2 &scalingFactor)
- T min ()
- NSVec4< T > & minimize (const NSVec4< T > &rhs)
- T max ()
- NSVec4< T > & maximize (const NSVec4< T > &rhs)
- NSVec4< T > & normalize ()
- NSVec4< T > & round ()
- NSVec4< T > & roundToZero ()
- NSVec3< T > & scalingFrom (const nsmat3< T > &transform)
- NSVec3< T > & scalingFrom (const nsmat4< T > &transform)
- NSVec4< T > & set (const T &pVal)
- NSVec4< T > & set (const T &pX, const T &pY, const T &pZ, const T &pW)
- NSVec4< T > & set (const NSVec3< T > &xyz, const T &pW)
- NSVec4< T > & set (const T &pX, const NSVec3< T > &yzw)
- NSVec4< T > & set (const NSVec2< T > &xy, const T &pZ, const T &pW)
- NSVec4< T > & set (const T &pX, const NSVec2< T > &yz, const T &pW)
- NSVec4< T > & set (const T &pX, const T &pY, const NSVec2< T > &zw)
- NSVec4< T > & setLength (const T &len)
- std::string toString ()
- NSVec4< T > & translationFrom (const nsmat4< T > &transform)
- NSVec4< T > operator+ (const NSVec4< T > &rhs) const
- NSVec4< T > operator- (const NSVec4< T > &rhs) const
- T operator∗ (const NSVec4< T > &rhs) const
- nsmat4< T > operator^ (const NSVec4< T > &pRHS) const
- NSVec4< T > operator% (const NSVec4< T > &rhs) const
- NSVec4< T > operator/ (const NSVec4< T > &rhs) const
- NSVec4< T > operator∗ (const T &rhs) const
- NSVec4< T > operator/ (const T &rhs) const
- NSVec4< T > & operator= (const NSVec4< T > &rhs)
- NSVec4< T > operator++ (int32_t)
- NSVec4< T > operator-- (int32_t)
- NSVec4< T > & operator++ ()
- NSVec4< T > & operator-- ()
- NSVec4< T > & operator+= (const NSVec4< T > &rhs)
- NSVec4< T > & operator-= (const NSVec4< T > &rhs)
- NSVec4< T > & operator%= (const NSVec4< T > &rhs)
- NSVec4< T > & operator/= (const NSVec4< T > &rhs)
- NSVec4< T > & operator∗= (const T &rhs)
- NSVec4< T > & operator/= (const T &rhs)
- bool operator== (const NSVec4< T > &rhs) const
- bool operator< (const NSVec4< T > &rhs) const
- bool operator> (const NSVec4< T > &rhs) const
- bool operator>= (const NSVec4< T > &rhs) const
- bool operator<= (const NSVec4< T > &rhs) const
- bool operator!= (const NSVec4< T > &rhs) const
- bool operator== (const T &rhs) const
- bool operator!= (const T &rhs) const
- const T & operator[] (const uint32_t &pVal) const
- T & operator[] (const uint32_t &pVal)
- NSVec4< T > xxxx () const
- NSVec4< T > xxxy () const
- NSVec4< T > xxxz () const
- NSVec4< T > xxxw () const
- NSVec4< T > xxyx () const

- NSVec4< T > xxyy () const
- NSVec4< T > xxyz () const
- NSVec4< T > xxyw () const
- NSVec4< T > xxzx () const
- NSVec4< T > xxzy () const
- NSVec4< T > xxzz () const
- NSVec4< T > xxzw () const
- NSVec4< T > xxwx () const
- NSVec4< T > xxwy () const
- NSVec4< T > xxwz () const
- NSVec4< T > xxww () const
- NSVec4< T > xyxx () const
- NSVec4< T > xyxy () const
- NSVec4< T > xyxz () const
- NSVec4< T > xyxw () const
- NSVec4< T > xyyx () const
- NSVec4< T > xyyy () const
- NSVec4< T > xyyz () const
- NSVec4< T > xyyw () const
- NSVec4< T > xywx () const
- NSVec4< T > xywy () const
- NSVec4< T > xywz () const
- NSVec4< T > xyww () const
- NSVec4< T > xzxx () const
- NSVec4< T > xzxy () const
- NSVec4< T > xzxz () const
- NSVec4< T > xzxw () const
- NSVec4< T > xzyx () const
- NSVec4< T > xzyy () const
- NSVec4< T > xzyz () const
- NSVec4< T > xzyw () const
- NSVec4< T > xzzx () const
- NSVec4< T > xzzy () const
- NSVec4< T > xzzz () const
- NSVec4< T > xzzw () const
- NSVec4< T > xzwx () const
- NSVec4< T > xzwy () const
- NSVec4< T > xzwz () const
- NSVec4< T > xzww () const
- NSVec4< T > xwxx () const
- NSVec4< T > xwxy () const
- NSVec4< T > xwxz () const
- NSVec4< T > xwxw () const
- NSVec4< T > xwyx () const
- NSVec4< T > xwyy () const
- NSVec4< T > xwyz () const
- NSVec4< T > xwyw () const
- NSVec4< T > xwzx () const
- NSVec4< T > xwzy () const
- NSVec4< T > xwzz () const
- NSVec4< T > xwzw () const
- NSVec4< T > xwwx () const
- NSVec4< T > xwwy () const
- NSVec4< T > xwwz () const
- NSVec4< T > xwww () const

- NSVec4< T > yxxx () const
- NSVec4< T > yxxy () const
- NSVec4< T > yxxz () const
- NSVec4< T > yxxw () const
- NSVec4< T > yxyx () const
- NSVec4< T > yxyy () const
- NSVec4< T > yxyz () const
- NSVec4< T > yxyw () const
- NSVec4< T > yxzx () const
- NSVec4< T > yxzy () const
- NSVec4< T > yxzz () const
- NSVec4< T > yxzw () const
- NSVec4< T > yxwx () const
- NSVec4< T > yxwy () const
- NSVec4< T > yxwz () const
- NSVec4< T > yxww () const
- NSVec4< T > yyxx () const
- NSVec4< T > yyxy () const
- NSVec4< T > yyxz () const
- NSVec4< T > yyxw () const
- NSVec4< T > yyyx () const
- NSVec4< T > yyyy () const
- NSVec4< T > yyyz () const
- NSVec4< T > yyyw () const
- NSVec4< T > yyzx () const
- NSVec4< T > yyzy () const
- NSVec4< T > yyzz () const
- NSVec4< T > yyzw () const
- NSVec4< T > yywx () const
- NSVec4< T > yywy () const
- NSVec4< T > yywz () const
- NSVec4< T > yyww () const
- NSVec4< T > yzxx () const
- NSVec4< T > yzxy () const
- NSVec4< T > yzxz () const
- NSVec4< T > yzxw () const
- NSVec4< T > yzyx () const
- NSVec4< T > yzyy () const
- NSVec4< T > yzyz () const
- NSVec4< T > yzyw () const
- NSVec4< T > yzzx () const
- NSVec4< T > yzzy () const
- NSVec4< T > yzzz () const
- NSVec4< T > yzzw () const
- NSVec4< T > yzwx () const
- NSVec4< T > yzwy () const
- NSVec4< T > yzwz () const
- NSVec4< T > yzww () const
- NSVec4< T > ywxx () const
- NSVec4< T > ywxy () const
- NSVec4< T > ywxz () const
- NSVec4< T > ywxw () const
- NSVec4< T > ywyx () const
- NSVec4< T > ywyy () const
- NSVec4< T > ywyz () const

- NSVec4< T > ywyw () const
- NSVec4< T > ywzx () const
- NSVec4< T > ywzy () const
- NSVec4< T > ywzz () const
- NSVec4< T > ywzw () const
- NSVec4< T > ywwx () const
- NSVec4< T > ywwy () const
- NSVec4< T > ywwz () const
- NSVec4< T > ywww () const
- NSVec4< T > zxxx () const
- NSVec4< T > zxxy () const
- NSVec4< T > zxxz () const
- NSVec4< T > zxxw () const
- NSVec4< T > zxyx () const
- NSVec4< T > zxyy () const
- NSVec4< T > zxyz () const
- NSVec4< T > zxyw () const
- NSVec4< T > zxzx () const
- NSVec4< T > zxzy () const
- NSVec4< T > zxzz () const
- NSVec4< T > zxzw () const
- NSVec4< T > zxwx () const
- NSVec4< T > zxwy () const
- NSVec4< T > zxwz () const
- NSVec4< T > zxww () const
- NSVec4< T > zyxx () const
- NSVec4< T > zyxy () const
- NSVec4< T > zyxz () const
- NSVec4< T > zyxw () const
- NSVec4< T > zyyx () const
- NSVec4< T > zyyy () const
- NSVec4< T > zyyz () const
- NSVec4< T > zyyw () const
- NSVec4< T > zyzx () const
- NSVec4< T > zyzy () const
- NSVec4< T > zyzz () const
- NSVec4< T > zyzw () const
- NSVec4< T > zywx () const
- NSVec4< T > zywy () const
- NSVec4< T > zywz () const
- NSVec4< T > zyww () const
- NSVec4< T > zzxx () const
- NSVec4< T > zzxy () const
- NSVec4< T > zzxz () const
- NSVec4< T > zzxw () const
- NSVec4< T > zzyx () const
- NSVec4< T > zzyy () const
- NSVec4< T > zzyz () const
- NSVec4< T > zzyw () const
- NSVec4< T > zzzx () const
- NSVec4< T > zzzy () const
- NSVec4< T > zzzz () const
- NSVec4< T > zzzw () const
- NSVec4< T > zzwx () const
- NSVec4< T > zzwy () const

- NSVec4< T > zzwz () const
- NSVec4< T > zzww () const
- NSVec4< T > zwxx () const
- NSVec4< T > zwxy () const
- NSVec4< T > zwxz () const
- NSVec4< T > zwxw () const
- NSVec4< T > zwyx () const
- NSVec4< T > zwyy () const
- NSVec4< T > zwyz () const
- NSVec4< T > zwyw () const
- NSVec4< T > zwzx () const
- NSVec4< T > zwzy () const
- NSVec4< T > zwzz () const
- NSVec4< T > zwzw () const
- NSVec4< T > zwwx () const
- NSVec4< T > zwwy () const
- NSVec4< T > zwwz () const
- NSVec4< T > zwww () const
- NSVec4< T > wxxx () const
- NSVec4< T > wxxy () const
- NSVec4< T > wxxz () const
- NSVec4< T > wxxw () const
- NSVec4< T > wxyx () const
- NSVec4< T > wxyy () const
- NSVec4< T > wxyz () const
- NSVec4< T > wxyw () const
- NSVec4< T > wxzx () const
- NSVec4< T > wxzy () const
- NSVec4< T > wxzz () const
- NSVec4< T > wxzw () const
- NSVec4< T > wxwx () const
- NSVec4< T > wxwy () const
- NSVec4< T > wxwz () const
- NSVec4< T > wxww () const
- NSVec4< T > wyxx () const
- NSVec4< T > wyxy () const
- NSVec4< T > wyxz () const
- NSVec4< T > wyxw () const
- NSVec4< T > wyyx () const
- NSVec4< T > wyyy () const
- NSVec4< T > wyyz () const
- NSVec4< T > wyyw () const
- NSVec4< T > wyzx () const
- NSVec4< T > wyzy () const
- NSVec4< T > wyzz () const
- NSVec4< T > wyzw () const
- NSVec4< T > wywx () const
- NSVec4< T > wywy () const
- NSVec4< T > wywz () const
- NSVec4< T > wyww () const
- NSVec4< T > wzxx () const
- NSVec4< T > wzxy () const
- NSVec4< T > wzxz () const
- NSVec4< T > wzxw () const
- NSVec4< T > wzyx () const

- NSVec4< T > wzyy () const
- NSVec4< T > wzyz () const
- NSVec4< T > wzyw () const
- NSVec4< T > wzzx () const
- NSVec4< T > wzzy () const
- NSVec4< T > wzzz () const
- NSVec4< T > wzzw () const
- NSVec4< T > wzwx () const
- NSVec4< T > wzwy () const
- NSVec4< T > wzwz () const
- NSVec4< T > wzww () const
- NSVec4< T > wwxx () const
- NSVec4< T > wwxy () const
- NSVec4< T > wwxz () const
- NSVec4< T > wwxw () const
- NSVec4< T > wwyx () const
- NSVec4< T > wwyy () const
- NSVec4< T > wwyz () const
- NSVec4< T > wwyw () const
- NSVec4< T > wwzx () const
- NSVec4< T > wwzy () const
- NSVec4< T > wwzz () const
- NSVec4< T > wwzw () const
- NSVec4< T > wwwx () const
- NSVec4< T > wwwy () const
- NSVec4< T > wwwz () const
- NSVec4< T > wwww () const
- NSVec4< T > rrrr () const
- NSVec4< T > rrrg () const
- NSVec4< T > rrrb () const
- NSVec4< T > rrra () const
- NSVec4< T > rrgr () const
- NSVec4< T > rrgg () const
- NSVec4< T > rrgb () const
- NSVec4< T > rrga () const
- NSVec4< T > rrbr () const
- NSVec4< T > rrbg () const
- NSVec4< T > rrbb () const
- NSVec4< T > rrba () const
- NSVec4< T > rrar () const
- NSVec4< T > rrag () const
- NSVec4< T > rrab () const
- NSVec4< T > rraa () const
- NSVec4< T > rgrr () const
- NSVec4< T > rgrg () const
- NSVec4< T > rgrb () const
- NSVec4< T > rgra () const
- NSVec4< T > rggr () const
- NSVec4< T > rggg () const
- NSVec4< T > rggb () const
- NSVec4< T > rgga () const
- NSVec4< T > rgbr () const
- NSVec4< T > rgbg () const
- NSVec4< T > rgbb () const
- NSVec4< T > rgar () const

- NSVec4< T > rgag () const
- NSVec4< T > rgab () const
- NSVec4< T > rgaa () const
- NSVec4< T > rbrr () const
- NSVec4< T > rbrg () const
- NSVec4< T > rbrb () const
- NSVec4< T > rbra () const
- NSVec4< T > rbgr () const
- NSVec4< T > rbgg () const
- NSVec4< T > rbgb () const
- NSVec4< T > rbga () const
- NSVec4< T > rbbr () const
- NSVec4< T > rbbg () const
- NSVec4< T > rbbb () const
- NSVec4< T > rbba () const
- NSVec4< T > rbar () const
- NSVec4< T > rbag () const
- NSVec4< T > rbab () const
- NSVec4< T > rbaa () const
- NSVec4< T > rarr () const
- NSVec4< T > rarg () const
- NSVec4< T > rarb () const
- NSVec4< T > rara () const
- NSVec4< T > ragr () const
- NSVec4< T > ragg () const
- NSVec4< T > ragb () const
- NSVec4< T > raga () const
- NSVec4< T > rabr () const
- NSVec4< T > rabg () const
- NSVec4< T > rabb () const
- NSVec4< T > raba () const
- NSVec4< T > raar () const
- NSVec4< T > raag () const
- NSVec4< T > raab () const
- NSVec4< T > raaa () const
- NSVec4< T > grrr () const
- NSVec4< T > grrg () const
- NSVec4< T > grrb () const
- NSVec4< T > grra () const
- NSVec4< T > grgr () const
- NSVec4< T > grgg () const
- NSVec4< T > grgb () const
- NSVec4< T > grga () const
- NSVec4< T > grbr () const
- NSVec4< T > grbg () const
- NSVec4< T > grbb () const
- NSVec4< T > grba () const
- NSVec4< T > grar () const
- NSVec4< T > grag () const
- NSVec4< T > grab () const
- NSVec4< T > graa () const
- NSVec4< T > ggrr () const
- NSVec4< T > ggrg () const
- NSVec4< T > ggrb () const
- NSVec4< T > ggra () const

- NSVec4< T > gggr () const
- NSVec4< T > gggg () const
- NSVec4< T > gggb () const
- NSVec4< T > ggga () const
- NSVec4< T > ggbr () const
- NSVec4< T > ggbg () const
- NSVec4< T > ggbb () const
- NSVec4< T > ggba () const
- NSVec4< T > ggar () const
- NSVec4< T > ggag () const
- NSVec4< T > ggab () const
- NSVec4< T > ggaa () const
- NSVec4< T > gbrr () const
- NSVec4< T > gbrg () const
- NSVec4< T > gbrb () const
- NSVec4< T > gbra () const
- NSVec4< T > gbgr () const
- NSVec4< T > gbgg () const
- NSVec4< T > gbgb () const
- NSVec4< T > gbga () const
- NSVec4< T > gbbr () const
- NSVec4< T > gbbg () const
- NSVec4< T > gbbb () const
- NSVec4< T > gbba () const
- NSVec4< T > gbar () const
- NSVec4< T > gbag () const
- NSVec4< T > gbab () const
- NSVec4< T > gbaa () const
- NSVec4< T > garr () const
- NSVec4< T > garg () const
- NSVec4< T > garb () const
- NSVec4< T > gara () const
- NSVec4< T > gagr () const
- NSVec4< T > gagg () const
- NSVec4< T > gagb () const
- NSVec4< T > gaga () const
- NSVec4< T > gabr () const
- NSVec4< T > gabg () const
- NSVec4< T > gabb () const
- NSVec4< T > gaba () const
- NSVec4< T > gaar () const
- NSVec4< T > gaag () const
- NSVec4< T > gaab () const
- NSVec4< T > gaaa () const
- NSVec4< T > brrr () const
- NSVec4< T > brrg () const
- NSVec4< T > brrb () const
- NSVec4< T > brra () const
- NSVec4< T > brgr () const
- NSVec4< T > brgg () const
- NSVec4< T > brgb () const
- NSVec4< T > brga () const
- NSVec4< T > brbr () const
- NSVec4< T > brbg () const
- NSVec4< T > brbb () const

- NSVec4< T > brba () const
- NSVec4< T > brar () const
- NSVec4< T > brag () const
- NSVec4< T > brab () const
- NSVec4< T > braa () const
- NSVec4< T > bgrr () const
- NSVec4< T > bgrg () const
- NSVec4< T > bgrb () const
- NSVec4< T > bgra () const
- NSVec4< T > bggr () const
- NSVec4< T > bggg () const
- NSVec4< T > bggb () const
- NSVec4< T > bgga () const
- NSVec4< T > bgbr () const
- NSVec4< T > bgbg () const
- NSVec4< T > bgbb () const
- NSVec4< T > bgba () const
- NSVec4< T > bgar () const
- NSVec4< T > bgag () const
- NSVec4< T > bgab () const
- NSVec4< T > bgaa () const
- NSVec4< T > bbrr () const
- NSVec4< T > bbrg () const
- NSVec4< T > bbrb () const
- NSVec4< T > bbra () const
- NSVec4< T > bbgr () const
- NSVec4< T > bbgg () const
- NSVec4< T > bbgb () const
- NSVec4< T > bbga () const
- NSVec4< T > bbbr () const
- NSVec4< T > bbbg () const
- NSVec4< T > bbbb () const
- NSVec4< T > bbba () const
- NSVec4< T > bbar () const
- NSVec4< T > bbag () const
- NSVec4< T > bbab () const
- NSVec4< T > bbaa () const
- NSVec4< T > barr () const
- NSVec4< T > barg () const
- NSVec4< T > barb () const
- NSVec4< T > bara () const
- NSVec4< T > bagr () const
- NSVec4< T > bagg () const
- NSVec4< T > bagb () const
- NSVec4< T > baga () const
- NSVec4< T > babr () const
- NSVec4< T > babg () const
- NSVec4< T > babb () const
- NSVec4< T > baba () const
- NSVec4< T > baar () const
- NSVec4< T > baag () const
- NSVec4< T > baab () const
- NSVec4< T > baaa () const
- NSVec4< T > arrr () const
- NSVec4< T > arrg () const

- NSVec4< T > arrb () const
- NSVec4< T > arra () const
- NSVec4< T > argr () const
- NSVec4< T > argg () const
- NSVec4< T > argb () const
- NSVec4< T > arga () const
- NSVec4< T > arbr () const
- NSVec4< T > arbg () const
- NSVec4< T > arbb () const
- NSVec4< T > arba () const
- NSVec4< T > arar () const
- NSVec4< T > arag () const
- NSVec4< T > arab () const
- NSVec4< T > araa () const
- NSVec4< T > agrr () const
- NSVec4< T > agrg () const
- NSVec4< T > agrb () const
- NSVec4< T > agra () const
- NSVec4< T > aggr () const
- NSVec4< T > aggg () const
- NSVec4< T > aggb () const
- NSVec4< T > agga () const
- NSVec4< T > agbr () const
- NSVec4< T > agbg () const
- NSVec4< T > agbb () const
- NSVec4< T > agba () const
- NSVec4< T > agar () const
- NSVec4< T > agag () const
- NSVec4< T > agab () const
- NSVec4< T > agaa () const
- NSVec4< T > abrr () const
- NSVec4< T > abrg () const
- NSVec4< T > abrb () const
- NSVec4< T > abra () const
- NSVec4< T > abgr () const
- NSVec4< T > abgg () const
- NSVec4< T > abgb () const
- NSVec4< T > abga () const
- NSVec4< T > abbr () const
- NSVec4< T > abbg () const
- NSVec4< T > abbb () const
- NSVec4< T > abba () const
- NSVec4< T > abar () const
- NSVec4< T > abag () const
- NSVec4< T > abab () const
- NSVec4< T > abaa () const
- NSVec4< T > aarr () const
- NSVec4< T > aarg () const
- NSVec4< T > aarb () const
- NSVec4< T > aara () const
- NSVec4< T > aagr () const
- NSVec4< T > aagg () const
- NSVec4< T > aagb () const
- NSVec4< T > aaga () const
- NSVec4< T > aabr () const

- NSVec4< T > aabg () const
- NSVec4< T > aabb () const
- NSVec4< T > aaba () const
- NSVec4< T > aaar () const
- NSVec4< T > aaag () const
- NSVec4< T > aaab () const
- NSVec4< T > aaaa () const
- NSVec4< T > ssss () const
- NSVec4< T > ssst () const
- NSVec4< T > sssp () const
- NSVec4< T > sssq () const
- NSVec4< T > ssts () const
- NSVec4< T > sstt () const
- NSVec4< T > sstp () const
- NSVec4< T > sstq () const
- NSVec4< T > ssps () const
- NSVec4< T > sspt () const
- NSVec4< T > sspp () const
- NSVec4< T > sspq () const
- NSVec4< T > ssqs () const
- NSVec4< T > ssqt () const
- NSVec4< T > ssqp () const
- NSVec4< T > ssqq () const
- NSVec4< T > stss () const
- NSVec4< T > stst () const
- NSVec4< T > stsp () const
- NSVec4< T > stsq () const
- NSVec4< T > stts () const
- NSVec4< T > sttt () const
- NSVec4< T > sttp () const
- NSVec4< T > sttq () const
- NSVec4< T > stps () const
- NSVec4< T > stpt () const
- NSVec4< T > stpp () const
- NSVec4< T > stqs () const
- NSVec4< T > stqt () const
- NSVec4< T > stqp () const
- NSVec4< T > stqq () const
- NSVec4< T > spss () const
- NSVec4< T > spst () const
- NSVec4< T > spsp () const
- NSVec4< T > spsq () const
- NSVec4< T > spts () const
- NSVec4< T > sptt () const
- NSVec4< T > sptp () const
- NSVec4< T > sptq () const
- NSVec4< T > spps () const
- NSVec4< T > sppt () const
- NSVec4< T > sppp () const
- NSVec4< T > sppq () const
- NSVec4< T > spqs () const
- NSVec4< T > spqt () const
- NSVec4< T > spqp () const
- NSVec4< T > spqq () const
- NSVec4< T > sqss () const

- NSVec4< T > sqst () const
- NSVec4< T > sqsp () const
- NSVec4< T > sqsq () const
- NSVec4< T > sqts () const
- NSVec4< T > sqtt () const
- NSVec4< T > sqtp () const
- NSVec4< T > sqtq () const
- NSVec4< T > sqps () const
- NSVec4< T > sqpt () const
- NSVec4< T > sqpp () const
- NSVec4< T > sqpq () const
- NSVec4< T > sqqs () const
- NSVec4< T > sqqt () const
- NSVec4< T > sqqp () const
- NSVec4< T > sqqq () const
- NSVec4< T > tsss () const
- NSVec4< T > tsst () const
- NSVec4< T > tssp () const
- NSVec4< T > tssq () const
- NSVec4< T > tsts () const
- NSVec4< T > tstt () const
- NSVec4< T > tstp () const
- NSVec4< T > tstq () const
- NSVec4< T > tsps () const
- NSVec4< T > tspt () const
- NSVec4< T > tspp () const
- NSVec4< T > tspq () const
- NSVec4< T > tsqs () const
- NSVec4< T > tsqt () const
- NSVec4< T > tsqp () const
- NSVec4< T > tsqq () const
- NSVec4< T > ttss () const
- NSVec4< T > ttst () const
- NSVec4< T > ttsp () const
- NSVec4< T > ttsq () const
- NSVec4< T > ttts () const
- NSVec4< T > tttt () const
- NSVec4< T > tttp () const
- NSVec4< T > tttq () const
- NSVec4< T > ttps () const
- NSVec4< T > ttpt () const
- NSVec4< T > ttpp () const
- NSVec4< T > ttpq () const
- NSVec4< T > ttqs () const
- NSVec4< T > ttqt () const
- NSVec4< T > ttqp () const
- NSVec4< T > ttqq () const
- NSVec4< T > tpss () const
- NSVec4< T > tpst () const
- NSVec4< T > tpsp () const
- NSVec4< T > tpsq () const
- NSVec4< T > tpts () const
- NSVec4< T > tptt () const
- NSVec4< T > tptp () const
- NSVec4< T > tptq () const

- NSVec4< T > tpps () const
- NSVec4< T > tppt () const
- NSVec4< T > tppp () const
- NSVec4< T > tppq () const
- NSVec4< T > tpqs () const
- NSVec4< T > tpqt () const
- NSVec4< T > tpqp () const
- NSVec4< T > tpqq () const
- NSVec4< T > tqss () const
- NSVec4< T > tqst () const
- NSVec4< T > tqsp () const
- NSVec4< T > tqsq () const
- NSVec4< T > tqts () const
- NSVec4< T > tqtt () const
- NSVec4< T > tqtp () const
- NSVec4< T > tqtq () const
- NSVec4< T > tqps () const
- NSVec4< T > tqpt () const
- NSVec4< T > tqpp () const
- NSVec4< T > tqpq () const
- NSVec4< T > tqqs () const
- NSVec4< T > tqqt () const
- NSVec4< T > tqqp () const
- NSVec4< T > tqqq () const
- NSVec4< T > psss () const
- NSVec4< T > psst () const
- NSVec4< T > pssp () const
- NSVec4< T > pssq () const
- NSVec4< T > psts () const
- NSVec4< T > pstt () const
- NSVec4< T > pstp () const
- NSVec4< T > pstq () const
- NSVec4< T > psps () const
- NSVec4< T > pspt () const
- NSVec4< T > pspp () const
- NSVec4< T > pspq () const
- NSVec4< T > psqs () const
- NSVec4< T > psqt () const
- NSVec4< T > psqp () const
- NSVec4< T > psqq () const
- NSVec4< T > ptss () const
- NSVec4< T > ptst () const
- NSVec4< T > ptsp () const
- NSVec4< T > ptsq () const
- NSVec4< T > ptts () const
- NSVec4< T > pttt () const
- NSVec4< T > pttp () const
- NSVec4< T > pttq () const
- NSVec4< T > ptps () const
- NSVec4< T > ptpt () const
- NSVec4< T > ptpp () const
- NSVec4< T > ptpq () const
- NSVec4< T > ptqs () const
- NSVec4< T > ptqt () const
- NSVec4< T > ptqp () const

- NSVec4< T > ptqq () const
- NSVec4< T > ppss () const
- NSVec4< T > ppst () const
- NSVec4< T > ppsp () const
- NSVec4< T > ppsq () const
- NSVec4< T > ppts () const
- NSVec4< T > pptt () const
- NSVec4< T > pptp () const
- NSVec4< T > pptq () const
- NSVec4< T > ppps () const
- NSVec4< T > pppt () const
- NSVec4< T > pppp () const
- NSVec4< T > pppq () const
- NSVec4< T > ppqs () const
- NSVec4< T > ppqt () const
- NSVec4< T > ppqp () const
- NSVec4< T > ppqq () const
- NSVec4< T > pqss () const
- NSVec4< T > pqst () const
- NSVec4< T > pqsp () const
- NSVec4< T > pqsq () const
- NSVec4< T > pqts () const
- NSVec4< T > pqtt () const
- NSVec4< T > pqtp () const
- NSVec4< T > pqtq () const
- NSVec4< T > pqps () const
- NSVec4< T > pqpt () const
- NSVec4< T > pqpp () const
- NSVec4< T > pqpq () const
- NSVec4< T > pqqs () const
- NSVec4< T > pqqt () const
- NSVec4< T > pqqp () const
- NSVec4< T > pqqq () const
- NSVec4< T > qsss () const
- NSVec4< T > qsst () const
- NSVec4< T > qssp () const
- NSVec4< T > qssq () const
- NSVec4< T > qsts () const
- NSVec4< T > qstt () const
- NSVec4< T > qstp () const
- NSVec4< T > qstq () const
- NSVec4< T > qsps () const
- NSVec4< T > qspt () const
- NSVec4< T > qspp () const
- NSVec4< T > qspq () const
- NSVec4< T > qsqs () const
- NSVec4< T > qsqt () const
- NSVec4< T > qsqp () const
- NSVec4< T > qsqq () const
- NSVec4< T > qtss () const
- NSVec4< T > qtst () const
- NSVec4< T > qtsp () const
- NSVec4< T > qtsq () const
- NSVec4< T > qtts () const
- NSVec4< T > qttt () const

- NSVec4< T > qttp () const
- NSVec4< T > qttq () const
- NSVec4< T > qtps () const
- NSVec4< T > qtpt () const
- NSVec4< T > qtpp () const
- NSVec4< T > qtpq () const
- NSVec4< T > qtqs () const
- NSVec4< T > qtqt () const
- NSVec4< T > qtqp () const
- NSVec4< T > qtqq () const
- NSVec4< T > qpss () const
- NSVec4< T > qpst () const
- NSVec4< T > qpsp () const
- NSVec4< T > qpsq () const
- NSVec4< T > qpts () const
- NSVec4< T > qptt () const
- NSVec4< T > qptp () const
- NSVec4< T > qptq () const
- NSVec4< T > qpps () const
- NSVec4< T > qppt () const
- NSVec4< T > qppp () const
- NSVec4< T > qppq () const
- NSVec4< T > qpqs () const
- NSVec4< T > qpqt () const
- NSVec4< T > qpqp () const
- NSVec4< T > qpqq () const
- NSVec4< T > qqss () const
- NSVec4< T > qqst () const
- NSVec4< T > qqsp () const
- NSVec4< T > qqsq () const
- NSVec4< T > qqts () const
- NSVec4< T > qqtt () const
- NSVec4< T > qqtp () const
- NSVec4< T > qqtq () const
- NSVec4< T > qqps () const
- NSVec4< T > qqpt () const
- NSVec4< T > qqpp () const
- NSVec4< T > qqpq () const
- NSVec4< T > qqqs () const
- NSVec4< T > qqqt () const
- NSVec4< T > qqqp () const
- NSVec4< T > qqqq () const
- NSVec3< T > xxx () const
- NSVec3< T > xxy () const
- NSVec3< T > xxz () const
- NSVec3< T > xxw () const
- NSVec3< T > xyx () const
- NSVec3< T > xyy () const
- NSVec3< T > xyz () const
- NSVec3< T > xyw () const
- NSVec3< T > xzx () const
- NSVec3< T > xzy () const
- NSVec3< T > xzz () const
- NSVec3< T > xzw () const
- NSVec3< T > xwx () const

- NSVec3< T > xwy () const
- NSVec3< T > xwz () const
- NSVec3< T > xww () const
- NSVec3< T > yxx () const
- NSVec3< T > yxy () const
- NSVec3< T > yxz () const
- NSVec3< T > yxw () const
- NSVec3< T > yyx () const
- NSVec3< T > yyy () const
- NSVec3< T > yyz () const
- NSVec3< T > yyw () const
- NSVec3< T > yzx () const
- NSVec3< T > yzy () const
- NSVec3< T > yzz () const
- NSVec3< T > yzw () const
- NSVec3< T > ywx () const
- NSVec3< T > ywy () const
- NSVec3< T > ywz () const
- NSVec3< T > yww () const
- NSVec3< T > zxx () const
- NSVec3< T > zxy () const
- NSVec3< T > zxz () const
- NSVec3< T > zxw () const
- NSVec3< T > zyx () const
- NSVec3< T > zyy () const
- NSVec3< T > zyz () const
- NSVec3< T > zyw () const
- NSVec3< T > zzx () const
- NSVec3< T > zzy () const
- NSVec3< T > zzz () const
- NSVec3< T > zzw () const
- NSVec3< T > zwx () const
- NSVec3< T > zwy () const
- NSVec3< T > zwz () const
- NSVec3< T > zww () const
- NSVec3< T > wxx () const
- NSVec3< T > wxy () const
- NSVec3< T > wxz () const
- NSVec3< T > wxw () const
- NSVec3< T > wyx () const
- NSVec3< T > wyy () const
- NSVec3< T > wyz () const
- NSVec3< T > wyw () const
- NSVec3< T > wzx () const
- NSVec3< T > wzy () const
- NSVec3< T > wzz () const
- NSVec3< T > wzw () const
- NSVec3< T > wwx () const
- NSVec3< T > wwy () const
- NSVec3< T > wwz () const
- NSVec3< T > www () const
- NSVec3< T > rrr () const
- NSVec3< T > rrg () const
- NSVec3< T > rrb () const
- NSVec3< T > rra () const

- NSVec3< T > rgr () const
- NSVec3< T > rgg () const
- NSVec3< T > rgb () const
- NSVec3< T > rga () const
- NSVec3< T > rbr () const
- NSVec3< T > rbg () const
- NSVec3< T > rbb () const
- NSVec3< T > rba () const
- NSVec3< T > rar () const
- NSVec3< T > rag () const
- NSVec3< T > rab () const
- NSVec3< T > raa () const
- NSVec3< T > grr () const
- NSVec3< T > grg () const
- NSVec3< T > grb () const
- NSVec3< T > gra () const
- NSVec3< T > ggr () const
- NSVec3< T > ggg () const
- NSVec3< T > ggb () const
- NSVec3< T > gga () const
- NSVec3< T > gbr () const
- NSVec3< T > gbg () const
- NSVec3< T > gbb () const
- NSVec3< T > gba () const
- NSVec3< T > gar () const
- NSVec3< T > gag () const
- NSVec3< T > gab () const
- NSVec3< T > gaa () const
- NSVec3< T > brr () const
- NSVec3< T > brg () const
- NSVec3< T > brb () const
- NSVec3< T > bra () const
- NSVec3< T > bgr () const
- NSVec3< T > bgg () const
- NSVec3< T > bgb () const
- NSVec3< T > bga () const
- NSVec3< T > bbr () const
- NSVec3< T > bbg () const
- NSVec3< T > bbb () const
- NSVec3< T > bba () const
- NSVec3< T > bar () const
- NSVec3< T > bag () const
- NSVec3< T > bab () const
- NSVec3< T > baa () const
- NSVec3< T > arr () const
- NSVec3< T > arg () const
- NSVec3< T > arb () const
- NSVec3< T > ara () const
- NSVec3< T > agr () const
- NSVec3< T > agg () const
- NSVec3< T > agb () const
- NSVec3< T > aga () const
- NSVec3< T > abr () const
- NSVec3< T > abg () const
- NSVec3< T > abb () const

- NSVec3< T > aba () const
- NSVec3< T > aar () const
- NSVec3< T > aag () const
- NSVec3< T > aab () const
- NSVec3< T > aaa () const
- NSVec3< T > sss () const
- NSVec3< T > sst () const
- NSVec3< T > ssp () const
- NSVec3< T > ssq () const
- NSVec3< T > sts () const
- NSVec3< T > stt () const
- NSVec3< T > stp () const
- NSVec3< T > stq () const
- NSVec3< T > sps () const
- NSVec3< T > spt () const
- NSVec3< T > spp () const
- NSVec3< T > spq () const
- NSVec3< T > sqs () const
- NSVec3< T > sqt () const
- NSVec3< T > sqp () const
- NSVec3< T > sqq () const
- NSVec3< T > tss () const
- NSVec3< T > tst () const
- NSVec3< T > tsp () const
- NSVec3< T > tsq () const
- NSVec3< T > tts () const
- NSVec3< T > ttt () const
- NSVec3< T > ttp () const
- NSVec3< T > ttq () const
- NSVec3< T > tps () const
- NSVec3< T > tpt () const
- NSVec3< T > tpp () const
- NSVec3< T > tpq () const
- NSVec3< T > tqs () const
- NSVec3< T > tqt () const
- NSVec3< T > tqp () const
- NSVec3< T > tqq () const
- NSVec3< T > pss () const
- NSVec3< T > pst () const
- NSVec3< T > psp () const
- NSVec3< T > psq () const
- NSVec3< T > pts () const
- NSVec3< T > ptt () const
- NSVec3< T > ptp () const
- NSVec3< T > ptq () const
- NSVec3< T > pps () const
- NSVec3< T > ppt () const
- NSVec3< T > ppp () const
- NSVec3< T > ppq () const
- NSVec3< T > pqs () const
- NSVec3< T > pqt () const
- NSVec3< T > pqp () const
- NSVec3< T > pqq () const
- NSVec3< T > qss () const
- NSVec3< T > qst () const

- NSVec3< T > qsp () const
- NSVec3< T > qsq () const
- NSVec3< T > qts () const
- NSVec3< T > qtt () const
- NSVec3< T > qtp () const
- NSVec3< T > qtq () const
- NSVec3< T > qps () const
- NSVec3< T > qpt () const
- NSVec3< T > qpp () const
- NSVec3< T > qpq () const
- NSVec3< T > qqs () const
- NSVec3< T > qqt () const
- NSVec3< T > qqp () const
- NSVec3< T > qqq () const
- NSVec2< T > xx () const
- NSVec2< T > xy () const
- NSVec2< T > xz () const
- NSVec2< T > xw () const
- NSVec2< T > yx () const
- NSVec2< T > yy () const
- NSVec2< T > yz () const
- NSVec2< T > yw () const
- NSVec2< T > zx () const
- NSVec2< T > zy () const
- NSVec2< T > zz () const
- NSVec2< T > zw () const
- NSVec2< T > wx () const
- NSVec2< T > wy () const
- NSVec2< T > wz () const
- NSVec2< T > ww () const
- NSVec2< T > rr () const
- NSVec2< T > rg () const
- NSVec2< T > rb () const
- NSVec2< T > ra () const
- NSVec2< T > gr () const
- NSVec2< T > gg () const
- NSVec2< T > gb () const
- NSVec2< T > ga () const
- NSVec2< T > br () const
- NSVec2< T > bg () const
- NSVec2< T > bb () const
- NSVec2< T > ba () const
- NSVec2< T > ar () const
- NSVec2< T > ag () const
- NSVec2< T > ab () const
- NSVec2< T > aa () const
- NSVec2< T > ss () const
- NSVec2< T > st () const
- NSVec2< T > sp () const
- NSVec2< T > sq () const
- NSVec2< T > ts () const
- NSVec2< T > tt () const
- NSVec2< T > tp () const
- NSVec2< T > tq () const
- NSVec2< T > ps () const

- NSVec2< T > pt () const
- NSVec2< T > pp () const
- NSVec2< T > pq () const
- NSVec2< T > qs () const
- NSVec2< T > qt () const
- NSVec2< T > qp () const
- NSVec2< T > qq () const

Public Attributes

- union {
  T data [4]
  struct {
  T x
  T y
  T z
  T w
  }
  struct {
  T r
  T g
  T b
  T a
  }
  struct {
  T s
  T t
  T p
  T q
  }
  };

6.45.1 Constructor & Destructor Documentation

6.45.1.1 template<class T> NSVec4< T >::NSVec4 (const NSVec4< T > & *copy*) [inline]

6.45.1.2 template<class T> NSVec4< T >::NSVec4 (const T & *val* = static_cast<T>(0)) [inline]

6.45.1.3 template<class T> NSVec4< T >::NSVec4 (const T & *pX*, const T & *pY*, const T & *pZ* = static_cast<T>(0), const T & *pW* = static_cast<T>(0)) [inline]

6.45.1.4 template<class T> NSVec4< T >::NSVec4 (const NSVec3< T > & *xyz*, const T & *pW* = static_cast<T>(1)) [inline]

6.45.1.5 template<class T> NSVec4< T >::NSVec4 (const T & *pX*, const NSVec3< T > & *yzw*) [inline]

6.45.1.6 template<class T> NSVec4< T >::NSVec4 (const NSVec2< T > & *xy*, const T & *pZ* = static_cast<T>(0), const T & *pW* = static_cast<T>(0)) [inline]

6.45.1.7 template<class T> NSVec4< T >::NSVec4 (const T & *pX*, const NSVec2< T > & *yz*, const T & *pW* = static_cast<T>(0)) [inline]

6.45.1.8 template<class T> NSVec4< T >::NSVec4 (const T & *pX*, const T & *pY*, const NSVec2< T > & *zw*) [inline]

6.45.2 Member Function Documentation

6.45.2.1 template<class T> NSVec2<T> NSVec4< T >::aa ( ) const [inline]

6.45.2.2 template<class T> NSVec3<T> NSVec4< T >::aaa ( ) const [inline]

6.45.2.3 template<class T> NSVec4<T> NSVec4< T >::aaaa ( ) const [inline]

6.45.2.4 template<class T> NSVec4<T> NSVec4< T >::aaab ( ) const [inline]

6.45.2.5 template<class T> NSVec4<T> NSVec4< T >::aaag ( ) const [inline]

6.45.2.6 template<class T> NSVec4<T> NSVec4< T >::aaar ( ) const [inline]

6.45.2.7 template<class T> NSVec3<T> NSVec4< T >::aab ( ) const [inline]

6.45.2.8 template<class T> NSVec4<T> NSVec4< T >::aaba ( ) const [inline]

6.45.2.9 template<class T> NSVec4<T> NSVec4< T >::aabb ( ) const [inline]

6.45.2.10 template<class T> NSVec4<T> NSVec4< T >::aabg ( ) const [inline]

6.45.2.11 template<class T> NSVec4<T> NSVec4< T >::aabr ( ) const [inline]

6.45.2.12 template<class T> NSVec3<T> NSVec4< T >::aag ( ) const [inline]

6.45.2.13 template<class T> NSVec4<T> NSVec4< T >::aaga ( ) const [inline]

6.45.2.14 template<class T> NSVec4<T> NSVec4< T >::aagb ( ) const [inline]

6.45.2.15 template<class T> NSVec4<T> NSVec4< T >::aagg ( ) const [inline]

6.45.2.16 template<class T> NSVec4<T> NSVec4< T >::aagr ( ) const [inline]

6.45.2.17 template<class T> NSVec3<T> NSVec4< T >::aar ( ) const [inline]

6.45.2.18 template<class T> NSVec4<T> NSVec4< T >::aara ( ) const [inline]

6.45.2.19 template<class T> NSVec4<T> NSVec4< T >::aarb ( ) const [inline]

6.45.2.20 template<class T> NSVec4<T> NSVec4< T >::aarg ( ) const [inline]

6.45.2.21 template<class T> NSVec4<T> NSVec4< T >::aarr ( ) const [inline]

6.45.2.22 template<class T> NSVec2<T> NSVec4< T >::ab ( ) const [inline]

6.45.2.23 template<class T> NSVec3<T> NSVec4< T >::aba ( ) const [inline]

6.45.2.24 template<class T> NSVec4<T> NSVec4< T >::abaa ( ) const [inline]

6.45.2.25 template<class T> NSVec4<T> NSVec4< T >::abab ( ) const [inline]

6.45.2.26 template<class T> NSVec4<T> NSVec4< T >::abag ( ) const [inline]

6.45.2.27 template<class T> NSVec4<T> NSVec4< T >::abar ( ) const [inline]

6.45.2.28 template<class T> NSVec3<T> NSVec4< T >::abb ( ) const [inline]

6.45.2.29 template<class T> NSVec4<T> NSVec4< T >::abba ( ) const [inline]

6.45.2.30 template<class T> NSVec4<T> NSVec4< T >::abbb ( ) const [inline]

6.45.2.31 template<class T> NSVec4<T> NSVec4< T >::abbg ( ) const [inline]

6.45.2.32 template<class T> NSVec4<T> NSVec4< T >::abbr ( ) const [inline]

6.45.2.33 template<class T> NSVec3<T> NSVec4< T >::abg ( ) const [inline]

6.45.2.34 template<class T> NSVec4<T> NSVec4< T >::abga ( ) const [inline]

6.45.2.35 template<class T> NSVec4<T> NSVec4< T >::abgb ( ) const [inline]

6.45.2.36 template<class T> NSVec4<T> NSVec4< T >::abgg ( ) const [inline]

6.45.2.37 template<class T> NSVec4<T> NSVec4< T >::abgr ( ) const [inline]

6.45.2.38 template<class T> NSVec3<T> NSVec4< T >::abr ( ) const [inline]

6.45.2.39 template<class T> NSVec4<T> NSVec4< T >::abra ( ) const [inline]

6.45.2.40 template<class T> NSVec4<T> NSVec4< T >::abrb ( ) const [inline]

6.45.2.41 template<class T> NSVec4<T> NSVec4< T >::abrg ( ) const [inline]

6.45.2.42 template<class T> NSVec4<T> NSVec4< T >::abrr ( ) const [inline]

6.45.2.43 template<class T> NSVec3<T>& NSVec4< T >::abs ( ) [inline]

6.45.2.44 template<class T> NSVec2<T> NSVec4< T >::ag ( ) const [inline]

6.45.2.45 template<class T> NSVec3<T> NSVec4< T >::aga ( ) const [inline]

6.45.2.46 template<class T> NSVec4<T> NSVec4< T >::agaa ( ) const [inline]

6.45.2.47 template<class T> NSVec4<T> NSVec4< T >::agab ( ) const [inline]

6.45.2.48 template<class T> NSVec4<T> NSVec4< T >::agag ( ) const [inline]

6.45.2.49 template<class T> NSVec4<T> NSVec4< T >::agar ( ) const [inline]

6.45.2.50 template<class T> NSVec3<T> NSVec4< T >::agb ( ) const [inline]

6.45.2.51 template<class T> NSVec4<T> NSVec4< T >::agba ( ) const [inline]

6.45.2.52 template<class T> NSVec4<T> NSVec4< T >::agbb ( ) const [inline]

6.45.2.53 template<class T> NSVec4<T> NSVec4< T >::agbg ( ) const [inline]

6.45.2.54 template<class T> NSVec4<T> NSVec4< T >::agbr ( ) const [inline]

6.45.2.55 template<class T> NSVec3<T> NSVec4< T >::agg ( ) const [inline]

6.45.2.56 template<class T> NSVec4<T> NSVec4< T >::agga ( ) const [inline]

6.45.2.57 template<class T> NSVec4<T> NSVec4< T >::aggb ( ) const [inline]

6.45.2.58 template<class T> NSVec4<T> NSVec4< T >::aggg ( ) const [inline]

6.45.2.59 template<class T> NSVec4<T> NSVec4< T >::aggr ( ) const [inline]

6.45.2.60 template<class T> NSVec3<T> NSVec4< T >::agr ( ) const [inline]

6.45.2.61 template<class T> NSVec4<T> NSVec4< T >::agra ( ) const [inline]

6.45.2.62 template<class T> NSVec4<T> NSVec4< T >::agrb ( ) const [inline]

6.45.2.63 template<class T> NSVec4<T> NSVec4< T >::agrg ( ) const [inline]

6.45.2.64 template<class T> NSVec4<T> NSVec4< T >::agrr ( ) const [inline]

6.45.2.65 template<class T> NSVec2<T> NSVec4< T >::ar ( ) const [inline]

6.45.2.66 template<class T> NSVec3<T> NSVec4< T >::ara ( ) const [inline]

6.45.2.67 template<class T> NSVec4<T> NSVec4< T >::araa ( ) const [inline]

6.45.2.68 template<class T> NSVec4<T> NSVec4< T >::arab ( ) const [inline]

6.45.2.69 template<class T> NSVec4<T> NSVec4< T >::arag ( ) const [inline]

6.45.2.70 template<class T> NSVec4<T> NSVec4< T >::arar ( ) const [inline]

6.45.2.71 template<class T> NSVec3<T> NSVec4< T >::arb ( ) const [inline]

6.45.2.72 template<class T> NSVec4<T> NSVec4< T >::arba ( ) const [inline]

6.45.2.73 template<class T> NSVec4<T> NSVec4< T >::arbb ( ) const [inline]

6.45.2.74 template<class T> NSVec4<T> NSVec4< T >::arbg ( ) const [inline]

6.45.2.75 template<class T> NSVec4<T> NSVec4< T >::arbr ( ) const [inline]

6.45.2.76 template<class T> NSVec3<T> NSVec4< T >::arg ( ) const [inline]

6.45.2.77 template<class T> NSVec4<T> NSVec4< T >::arga ( ) const [inline]

6.45.2.78 template<class T> NSVec4<T> NSVec4< T >::argb ( ) const [inline]

6.45.2.79 template<class T> NSVec4<T> NSVec4< T >::argg ( ) const [inline]

6.45.2.80 template<class T> NSVec4<T> NSVec4< T >::argr ( ) const [inline]

6.45.2.81 template<class T> NSVec3<T> NSVec4< T >::arr ( ) const [inline]

6.45.2.82 template<class T> NSVec4<T> NSVec4< T >::arra ( ) const [inline]

6.45.2.83 template<class T> NSVec4<T> NSVec4< T >::arrb ( ) const [inline]

6.45.2.84 template<class T> NSVec4<T> NSVec4< T >::arrg ( ) const [inline]

6.45.2.85 template<class T> NSVec4<T> NSVec4< T >::arrr ( ) const [inline]

6.45.2.86 template<class T> NSVec4<T>& NSVec4< T >::axisAngleFrom (const NSVec3< T > & *euler,* typename NSVec3< T >::EulerOrder *order,* bool *rads* = false) [inline]

6.45.2.87 template<class T> NSVec4<T>& NSVec4< T >::axisAngleFrom (const nsquat< T > & *orientation,* bool *rads* = false) [inline]

6.45.2.88 template<class T> NSVec4<T>& NSVec4< T >::axisAngleFrom (const nsmat3< T > & *rotationMat3,* bool *rads* = false) [inline]

6.45.2.89 template<class T> NSVec4<T>& NSVec4< T >::axisAngleFrom (const nsmat4< T > & *transform,* bool *rads* = false) [inline]

6.45.2.90 template<class T> NSVec4<T>& NSVec4< T >::axisAngleFrom (const NSVec3< T > & *vec,* const NSVec3< T > & *toVec,* bool *rads* = false) [inline]

6.45.2.91 template<class T> NSVec2<T> NSVec4< T >::ba ( ) const [inline]

6.45.2.92 template<class T> NSVec3<T> NSVec4< T >::baa ( ) const [inline]

6.45.2.93 template<class T> NSVec4<T> NSVec4< T >::baaa ( ) const [inline]

6.45.2.94 template<class T> NSVec4<T> NSVec4< T >::baab ( ) const [inline]

6.45.2.95 template<class T> NSVec4<T> NSVec4< T >::baag ( ) const [inline]

6.45.2.96 template<class T> NSVec4<T> NSVec4< T >::baar ( ) const [inline]

6.45.2.97 template<class T> NSVec3<T> NSVec4< T >::bab ( ) const [inline]

6.45.2.98 template<class T> NSVec4<T> NSVec4< T >::baba ( ) const [inline]

6.45.2.99 template<class T> NSVec4<T> NSVec4< T >::babb ( ) const [inline]

6.45.2.100 template<class T> NSVec4<T> NSVec4< T >::babg ( ) const [inline]

6.45.2.101 template<class T> NSVec4<T> NSVec4< T >::babr ( ) const [inline]

6.45.2.102 template<class T> NSVec3<T> NSVec4< T >::bag ( ) const [inline]

6.45.2.103 template<class T> NSVec4<T> NSVec4< T >::baga ( ) const [inline]

6.45.2.104 template<class T> NSVec4<T> NSVec4< T >::bagb ( ) const [inline]

6.45.2.105 template<class T> NSVec4<T> NSVec4< T >::bagg ( ) const [inline]

6.45.2.106 template<class T> NSVec4<T> NSVec4< T >::bagr ( ) const [inline]

6.45.2.107 template<class T> NSVec3<T> NSVec4< T >::bar ( ) const [inline]

6.45.2.108 template<class T> NSVec4<T> NSVec4< T >::bara ( ) const [inline]

6.45.2.109 template<class T> NSVec4<T> NSVec4< T >::barb ( ) const [inline]

6.45.2.110 template<class T> NSVec4<T> NSVec4< T >::barg ( ) const [inline]

6.45.2.111 template<class T> NSVec4<T> NSVec4< T >::barr ( ) const [inline]

6.45.2.112 template<class T> NSVec2<T> NSVec4< T >::bb ( ) const [inline]

6.45.2.113 template<class T> NSVec3<T> NSVec4< T >::bba ( ) const [inline]

6.45.2.114 template<class T> NSVec4<T> NSVec4< T >::bbaa ( ) const [inline]

6.45.2.115 template<class T> NSVec4<T> NSVec4< T >::bbab ( ) const [inline]

6.45.2.116 template<class T> NSVec4<T> NSVec4< T >::bbag ( ) const [inline]

6.45.2.117 template<class T> NSVec4<T> NSVec4< T >::bbar ( ) const [inline]

6.45.2.118 template<class T> NSVec3<T> NSVec4< T >::bbb ( ) const [inline]

6.45.2.119 template<class T> NSVec4<T> NSVec4< T >::bbba ( ) const [inline]

6.45.2.120 template<class T> NSVec4<T> NSVec4< T >::bbbb ( ) const [inline]

6.45.2.121 template<class T> NSVec4<T> NSVec4< T >::bbbg ( ) const [inline]

6.45.2.122 template<class T> NSVec4<T> NSVec4< T >::bbbr ( ) const [inline]

6.45.2.123 template<class T> NSVec3<T> NSVec4< T >::bbg ( ) const [inline]

6.45.2.124 template<class T> NSVec4<T> NSVec4< T >::bbga ( ) const [inline]

6.45.2.125 template<class T> NSVec4<T> NSVec4< T >::bbgb ( ) const [inline]

6.45.2.126 template<class T> NSVec4<T> NSVec4< T >::bbgg ( ) const [inline]

6.45.2.127 template<class T> NSVec4<T> NSVec4< T >::bbgr ( ) const [inline]

6.45.2.128 template<class T> NSVec3<T> NSVec4< T >::bbr ( ) const [inline]

6.45.2.129 template<class T> NSVec4<T> NSVec4< T >::bbra ( ) const [inline]

6.45.2.130 template<class T> NSVec4<T> NSVec4< T >::bbrb ( ) const [inline]

6.45.2.131 template<class T> NSVec4<T> NSVec4< T >::bbrg ( ) const [inline]

6.45.2.132 template<class T> NSVec4<T> NSVec4< T >::bbrr ( ) const [inline]

6.45.2.133 template<class T> NSVec2<T> NSVec4< T >::bg ( ) const [inline]

6.45.2.134 template<class T> NSVec3<T> NSVec4< T >::bga ( ) const [inline]

6.45.2.135 template<class T> NSVec4<T> NSVec4< T >::bgaa ( ) const [inline]

6.45.2.136 template<class T> NSVec4<T> NSVec4< T >::bgab ( ) const [inline]

6.45.2.137 template<class T> NSVec4<T> NSVec4< T >::bgag ( ) const [inline]

6.45.2.138 template<class T> NSVec4<T> NSVec4< T >::bgar ( ) const [inline]

6.45.2.139 template<class T> NSVec3<T> NSVec4< T >::bgb ( ) const [inline]

6.45.2.140 template<class T> NSVec4<T> NSVec4< T >::bgba ( ) const [inline]

6.45.2.141 template<class T> NSVec4<T> NSVec4< T >::bgbb ( ) const [inline]

6.45.2.142 template<class T> NSVec4<T> NSVec4< T >::bgbg ( ) const [inline]

6.45.2.143 template<class T> NSVec4<T> NSVec4< T >::bgbr ( ) const [inline]

6.45.2.144 template<class T> NSVec3<T> NSVec4< T >::bgg ( ) const [inline]

6.45.2.145 template<class T> NSVec4<T> NSVec4< T >::bgga ( ) const [inline]

6.45.2.146 template<class T> NSVec4<T> NSVec4< T >::bggb ( ) const [inline]

6.45.2.147 template<class T> NSVec4<T> NSVec4< T >::bggg ( ) const [inline]

6.45.2.148 template<class T> NSVec4<T> NSVec4< T >::bggr ( ) const [inline]

6.45.2.149 template<class T> NSVec3<T> NSVec4< T >::bgr ( ) const [inline]

6.45.2.150 template<class T> NSVec4<T> NSVec4< T >::bgra ( ) const [inline]

6.45.2.151 template<class T> NSVec4<T> NSVec4< T >::bgrb ( ) const [inline]

6.45.2.152 template<class T> NSVec4<T> NSVec4< T >::bgrg ( ) const [inline]

6.45.2.153 template<class T> NSVec4<T> NSVec4< T >::bgrr ( ) const [inline]

6.45.2.154 template<class T> NSVec2<T> NSVec4< T >::br ( ) const [inline]

6.45.2.155 template<class T> NSVec3<T> NSVec4< T >::bra ( ) const [inline]

6.45.2.156 template<class T> NSVec4<T> NSVec4< T >::braa ( ) const [inline]

6.45.2.157 template<class T> NSVec4<T> NSVec4< T >::brab ( ) const [inline]

6.45.2.158 template<class T> NSVec4<T> NSVec4< T >::brag ( ) const [inline]

6.45.2.159 template<class T> NSVec4<T> NSVec4< T >::brar ( ) const [inline]

6.45.2.160 template<class T> NSVec3<T> NSVec4< T >::brb ( ) const [inline]

6.45.2.161 template<class T> NSVec4<T> NSVec4< T >::brba ( ) const [inline]

6.45.2.162 template<class T> NSVec4<T> NSVec4< T >::brbb ( ) const [inline]

6.45.2.163 template<class T> NSVec4<T> NSVec4< T >::brbg ( ) const [inline]

6.45.2.164 template<class T> NSVec4<T> NSVec4< T >::brbr ( ) const [inline]

6.45.2.165 template<class T> NSVec3<T> NSVec4< T >::brg ( ) const [inline]

6.45.2.166 template<class T> NSVec4<T> NSVec4< T >::brga ( ) const [inline]

6.45.2.167 template<class T> NSVec4<T> NSVec4< T >::brgb ( ) const [inline]

6.45.2.168 template<class T> NSVec4<T> NSVec4< T >::brgg ( ) const [inline]

6.45.2.169 template<class T> NSVec4<T> NSVec4< T >::brgr ( ) const [inline]

6.45.2.170 template<class T> NSVec3<T> NSVec4< T >::brr ( ) const [inline]

6.45.2.171 template<class T> NSVec4<T> NSVec4< T >::brra ( ) const [inline]

6.45.2.172 template<class T> NSVec4<T> NSVec4< T >::brrb ( ) const [inline]

6.45.2.173 template<class T> NSVec4<T> NSVec4< T >::brrg ( ) const [inline]

6.45.2.174 template<class T> NSVec4<T> NSVec4< T >::brrr ( ) const [inline]

6.45.2.175 template<class T> NSVec4<T>& NSVec4< T >::ceil ( ) [inline]

6.45.2.176 template<class T> NSVec4<T>& NSVec4< T >::clamp (const T & *min* = static_cast<T>(0), const T & *max* = static_cast<T>(0)) [inline]

6.45.2.177 template<class T> T NSVec4< T >::distanceTo (const NSVec4< T > & *pVec*) const [inline]

6.45.2.178 template<class T> NSVec4<T>& NSVec4< T >::floor ( ) [inline]

6.45.2.179 template<class T> NSVec4<T>& NSVec4< T >::fract ( ) [inline]

6.45.2.180 template<class T> NSVec2<T> NSVec4< T >::ga ( ) const [inline]

6.45.2.181 template<class T> NSVec3<T> NSVec4< T >::gaa ( ) const [inline]

6.45.2.182 template<class T> NSVec4<T> NSVec4< T >::gaaa ( ) const [inline]

6.45.2.183 template<class T> NSVec4<T> NSVec4< T >::gaab ( ) const [inline]

6.45.2.184 template<class T> NSVec4<T> NSVec4< T >::gaag ( ) const [inline]

6.45.2.185 template<class T> NSVec4<T> NSVec4< T >::gaar ( ) const [inline]

6.45.2.186 template<class T> NSVec3<T> NSVec4< T >::gab ( ) const [inline]

6.45.2.187 template<class T> NSVec4<T> NSVec4< T >::gaba ( ) const [inline]

6.45.2.188 template<class T> NSVec4<T> NSVec4< T >::gabb ( ) const [inline]

6.45.2.189 template<class T> NSVec4<T> NSVec4< T >::gabg ( ) const [inline]

6.45.2.190 template<class T> NSVec4<T> NSVec4< T >::gabr ( ) const [inline]

6.45.2.191 template<class T> NSVec3<T> NSVec4< T >::gag ( ) const [inline]

6.45.2.192 template<class T> NSVec4<T> NSVec4< T >::gaga ( ) const [inline]

6.45.2.193 template<class T> NSVec4<T> NSVec4< T >::gagb ( ) const [inline]

6.45.2.194 template<class T> NSVec4<T> NSVec4< T >::gagg ( ) const [inline]

6.45.2.195 template<class T> NSVec4<T> NSVec4< T >::gagr ( ) const [inline]

6.45.2.196 template<class T> NSVec3<T> NSVec4< T >::gar ( ) const [inline]

6.45.2.197 template<class T> NSVec4<T> NSVec4< T >::gara ( ) const [inline]

6.45.2.198 template<class T> NSVec4<T> NSVec4< T >::garb ( ) const [inline]

6.45.2.199 template<class T> NSVec4<T> NSVec4< T >::garg ( ) const [inline]

6.45.2.200 template<class T> NSVec4<T> NSVec4< T >::garr ( ) const [inline]

6.45.2.201 template<class T> NSVec2<T> NSVec4< T >::gb ( ) const [inline]

6.45.2.202 template<class T> NSVec3<T> NSVec4< T >::gba ( ) const [inline]

6.45.2.203 template<class T> NSVec4<T> NSVec4< T >::gbaa ( ) const [inline]

6.45.2.204 template<class T> NSVec4<T> NSVec4< T >::gbab ( ) const [inline]

6.45.2.205 template<class T> NSVec4<T> NSVec4< T >::gbag ( ) const [inline]

6.45.2.206 template<class T> NSVec4<T> NSVec4< T >::gbar ( ) const [inline]

6.45.2.207 template<class T> NSVec3<T> NSVec4< T >::gbb ( ) const [inline]

6.45.2.208 template<class T> NSVec4<T> NSVec4< T >::gbba ( ) const [inline]

6.45.2.209 template<class T> NSVec4<T> NSVec4< T >::gbbb ( ) const [inline]

6.45.2.210 template<class T> NSVec4<T> NSVec4< T >::gbbg ( ) const [inline]

6.45.2.211 template<class T> NSVec4<T> NSVec4< T >::gbbr ( ) const [inline]

6.45.2.212 template<class T> NSVec3<T> NSVec4< T >::gbg ( ) const [inline]

6.45.2.213 template<class T> NSVec4<T> NSVec4< T >::gbga ( ) const [inline]

6.45.2.214 template<class T> NSVec4<T> NSVec4< T >::gbgb ( ) const [inline]

6.45.2.215 template<class T> NSVec4<T> NSVec4< T >::gbgg ( ) const [inline]

6.45.2.216 template<class T> NSVec4<T> NSVec4< T >::gbgr ( ) const [inline]

6.45.2.217 template<class T> NSVec3<T> NSVec4< T >::gbr ( ) const [inline]

6.45.2.218 template<class T> NSVec4<T> NSVec4< T >::gbra ( ) const [inline]

6.45.2.219 template<class T> NSVec4<T> NSVec4< T >::gbrb ( ) const [inline]

6.45.2.220 template<class T> NSVec4<T> NSVec4< T >::gbrg ( ) const [inline]

6.45.2.221 template<class T> NSVec4<T> NSVec4< T >::gbrr ( ) const [inline]

6.45.2.222 template<class T> NSVec2<T> NSVec4< T >::gg ( ) const [inline]

6.45.2.223 template<class T> NSVec3<T> NSVec4< T >::gga ( ) const [inline]

6.45.2.224 template<class T> NSVec4<T> NSVec4< T >::ggaa ( ) const [inline]

6.45.2.225 template<class T> NSVec4<T> NSVec4< T >::ggab ( ) const [inline]

6.45.2.226 template<class T> NSVec4<T> NSVec4< T >::ggag ( ) const [inline]

6.45.2.227 template<class T> NSVec4<T> NSVec4< T >::ggar ( ) const [inline]

6.45.2.228 template<class T> NSVec3<T> NSVec4< T >::ggb ( ) const [inline]

6.45.2.229 template<class T> NSVec4<T> NSVec4< T >::ggba ( ) const [inline]

6.45.2.230 template<class T> NSVec4<T> NSVec4< T >::ggbb ( ) const [inline]

6.45.2.231 template<class T> NSVec4<T> NSVec4< T >::ggbg ( ) const [inline]

6.45.2.232 template<class T> NSVec4<T> NSVec4< T >::ggbr ( ) const [inline]

6.45.2.233 template<class T> NSVec3<T> NSVec4< T >::ggg ( ) const [inline]

6.45.2.234 template<class T> NSVec4<T> NSVec4< T >::ggga ( ) const [inline]

6.45.2.235 template<class T> NSVec4<T> NSVec4< T >::gggb ( ) const [inline]

6.45.2.236 template<class T> NSVec4<T> NSVec4< T >::gggg ( ) const [inline]

6.45.2.237 template<class T> NSVec4<T> NSVec4< T >::gggr ( ) const [inline]

6.45.2.238 template<class T> NSVec3<T> NSVec4< T >::ggr ( ) const [inline]

6.45.2.239 template<class T> NSVec4<T> NSVec4< T >::ggra ( ) const [inline]

6.45.2.240 template<class T> NSVec4<T> NSVec4< T >::ggrb ( ) const [inline]

6.45.2.241 template<class T> NSVec4<T> NSVec4< T >::ggrg ( ) const [inline]

6.45.2.242 template<class T> NSVec4<T> NSVec4< T >::ggrr ( ) const [inline]

6.45.2.243 template<class T> NSVec2<T> NSVec4< T >::gr ( ) const [inline]

6.45.2.244 template<class T> NSVec3<T> NSVec4< T >::gra ( ) const [inline]

6.45.2.245 template<class T> NSVec4<T> NSVec4< T >::graa ( ) const [inline]

6.45.2.246 template<class T> NSVec4<T> NSVec4< T >::grab ( ) const [inline]

6.45.2.247 template<class T> NSVec4<T> NSVec4< T >::grag ( ) const [inline]

6.45.2.248 template<class T> NSVec4<T> NSVec4< T >::grar ( ) const [inline]

6.45.2.249 template<class T> NSVec3<T> NSVec4< T >::grb ( ) const [inline]

6.45.2.250 template<class T> NSVec4<T> NSVec4< T >::grba ( ) const [inline]

6.45.2.251 template<class T> NSVec4<T> NSVec4< T >::grbb ( ) const [inline]

6.45.2.252 template<class T> NSVec4<T> NSVec4< T >::grbg ( ) const [inline]

6.45.2.253 template<class T> NSVec4<T> NSVec4< T >::grbr ( ) const [inline]

6.45.2.254 template<class T> NSVec3<T> NSVec4< T >::grg ( ) const [inline]

6.45.2.255 template<class T> NSVec4<T> NSVec4< T >::grga ( ) const [inline]

6.45.2.256 template<class T> NSVec4<T> NSVec4< T >::grgb ( ) const [inline]

6.45.2.257 template<class T> NSVec4<T> NSVec4< T >::grgg ( ) const [inline]

6.45.2.258 template<class T> NSVec4<T> NSVec4< T >::grgr ( ) const [inline]

6.45.2.259 template<class T> NSVec3<T> NSVec4< T >::grr ( ) const [inline]

6.45.2.260 template<class T> NSVec4<T> NSVec4< T >::grra ( ) const [inline]

6.45.2.261 template<class T> NSVec4<T> NSVec4< T >::grrb ( ) const [inline]

6.45.2.262 template<class T> NSVec4<T> NSVec4< T >::grrg ( ) const [inline]

6.45.2.263 template<class T> NSVec4<T> NSVec4< T >::grrr ( ) const [inline]

6.45.2.264 template<class T> T NSVec4< T >::length ( ) const [inline]

6.45.2.265 template<class T> T NSVec4< T >::lengthSq ( ) const [inline]

6.45.2.266 template<class T> template<class T2 > NSVec4<T>& NSVec4< T >::lerp (const NSVec4< T > & *vec,* const T2 & *scalingFactor*) [inline]

6.45.2.267 template<class T> T NSVec4< T >::max ( ) [inline]

6.45.2.268 template<class T> NSVec4<T>& NSVec4< T >::maximize (const NSVec4< T > & *rhs*) [inline]

6.45.2.269 template<class T> T NSVec4< T >::min ( ) [inline]

6.45.2.270 template<class T> NSVec4<T>& NSVec4< T >::minimize (const NSVec4< T > & *rhs*) [inline]

6.45.2.271 template<class T> NSVec4<T>& NSVec4< T >::normalize ( ) [inline]

6.45.2.272 template<class T> bool NSVec4< T >::operator!= (const NSVec4< T > & *rhs*) const [inline]

6.45.2.273 template<class T> bool NSVec4< T >::operator!= (const T & *rhs*) const [inline]

6.45.2.274 template<class T> NSVec4<T> NSVec4< T >::operator% (const NSVec4< T > & *rhs*) const [inline]

6.45.2.275 template<class T> NSVec4<T>& NSVec4< T >::operator%= (const NSVec4< T > & *rhs*) [inline]

6.45.2.276 template<class T> T NSVec4< T >::operator* (const NSVec4< T > & *rhs*) const [inline]

6.45.2.277 template<class T> NSVec4<T> NSVec4< T >::operator* (const T & *rhs*) const [inline]

6.45.2.278 template<class T> NSVec4<T>& NSVec4< T >::operator*= (const T & *rhs*) [inline]

6.45.2.279 template<class T> NSVec4<T> NSVec4< T >::operator+ (const NSVec4< T > & *rhs*) const [inline]

6.45.2.280 template<class T> NSVec4<T> NSVec4< T >::operator++ (int32_t) [inline]

6.45.2.281 template<class T> NSVec4<T>& NSVec4< T >::operator++ ( ) [inline]

6.45.2.282 template<class T> NSVec4<T>& NSVec4< T >::operator+= (const NSVec4< T > & *rhs*) [inline]

6.45.2.283 template<class T> NSVec4<T> NSVec4< T >::operator- (const NSVec4< T > & *rhs*) const [inline]

6.45.2.284 template<class T> NSVec4<T> NSVec4< T >::operator-- (int32_t) [inline]

6.45.2.285 template<class T> NSVec4<T>& NSVec4< T >::operator-- ( ) [inline]

6.45.2.286 template<class T> NSVec4<T>& NSVec4< T >::operator-= (const NSVec4< T > & *rhs*) [inline]

6.45.2.287 template<class T> NSVec4<T> NSVec4< T >::operator/ (const NSVec4< T > & *rhs*) const [inline]

6.45.2.288 template<class T> NSVec4<T> NSVec4< T >::operator/ (const T & *rhs*) const [inline]

6.45.2.289 template<class T> NSVec4<T>& NSVec4< T >::operator/= (const NSVec4< T > & *rhs*) [inline]

6.45.2.290 template<class T> NSVec4<T>& NSVec4< T >::operator/= (const T & *rhs*) [inline]

6.45.2.291 template<class T> bool NSVec4< T >::operator< (const NSVec4< T > & *rhs*) const [inline]

6.45.2.292 template<class T> bool NSVec4< T >::operator<= (const NSVec4< T > & *rhs*) const [inline]

6.45.2.293 template<class T> NSVec4<T>& NSVec4< T >::operator= (const NSVec4< T > & *rhs*) [inline]

6.45.2.294 template<class T> bool NSVec4< T >::operator== (const NSVec4< T > & *rhs*) const [inline]

6.45.2.295 template<class T> bool NSVec4< T >::operator== (const T & *rhs*) const [inline]

6.45.2.296 template<class T> bool NSVec4< T >::operator> (const NSVec4< T > & *rhs*) const [inline]

6.45.2.297 template<class T> bool NSVec4< T >::operator>= (const NSVec4< T > & *rhs*) const [inline]

6.45.2.298 template<class T> const T& NSVec4< T >::operator[] (const uint32_t & *pVal*) const [inline]

6.45.2.299 template<class T> T& NSVec4< T >::operator[] (const uint32_t & *pVal*) [inline]

6.45.2.300 template<class T> nsmat4<T> NSVec4< T >::operator^ (const NSVec4< T > & *pRHS*) const [inline]

6.45.2.301 template<class T> NSVec2<T> NSVec4< T >::pp ( ) const [inline]

6.45.2.302 template<class T> NSVec3<T> NSVec4< T >::ppp ( ) const [inline]

6.45.2.303 template<class T> NSVec4<T> NSVec4< T >::pppp ( ) const [inline]

6.45.2.304 template<class T> NSVec4<T> NSVec4< T >::pppq ( ) const [inline]

6.45.2.305 template<class T> NSVec4<T> NSVec4< T >::ppps ( ) const [inline]

6.45.2.306 template<class T> NSVec4<T> NSVec4< T >::pppt ( ) const [inline]

6.45.2.307 template<class T> NSVec3<T> NSVec4< T >::ppq ( ) const [inline]

6.45.2.308 template<class T> NSVec4<T> NSVec4< T >::ppqp ( ) const [inline]

6.45.2.309 template<class T> NSVec4<T> NSVec4< T >::ppqq ( ) const [inline]

6.45.2.310 template<class T> NSVec4<T> NSVec4< T >::ppqs ( ) const [inline]

6.45.2.311 template<class T> NSVec4<T> NSVec4< T >::ppqt ( ) const [inline]

6.45.2.312 template<class T> NSVec3<T> NSVec4< T >::pps ( ) const [inline]

6.45.2.313 template<class T> NSVec4<T> NSVec4< T >::ppsp ( ) const [inline]

6.45.2.314 template<class T> NSVec4<T> NSVec4< T >::ppsq ( ) const [inline]

6.45.2.315 template<class T> NSVec4<T> NSVec4< T >::ppss ( ) const [inline]

6.45.2.316 template<class T> NSVec4<T> NSVec4< T >::ppst ( ) const [inline]

6.45.2.317 template<class T> NSVec3<T> NSVec4< T >::ppt ( ) const [inline]

6.45.2.318 template<class T> NSVec4<T> NSVec4< T >::pptp ( ) const [inline]

6.45.2.319 template<class T> NSVec4<T> NSVec4< T >::pptq ( ) const [inline]

6.45.2.320 template<class T> NSVec4<T> NSVec4< T >::ppts ( ) const [inline]

6.45.2.321 template<class T> NSVec4<T> NSVec4< T >::pptt ( ) const [inline]

6.45.2.322 template<class T> NSVec2<T> NSVec4< T >::pq ( ) const [inline]

6.45.2.323 template<class T> NSVec3<T> NSVec4< T >::pqp ( ) const [inline]

6.45.2.324 template<class T> NSVec4<T> NSVec4< T >::pqpp ( ) const [inline]

6.45.2.325 template<class T> NSVec4<T> NSVec4< T >::pqpq ( ) const [inline]

6.45.2.326 template<class T> NSVec4<T> NSVec4< T >::pqps ( ) const [inline]

6.45.2.327 template<class T> NSVec4<T> NSVec4< T >::pqpt ( ) const [inline]

6.45.2.328 template<class T> NSVec3<T> NSVec4< T >::pqq ( ) const [inline]

6.45.2.329 template<class T> NSVec4<T> NSVec4< T >::pqqp ( ) const [inline]

6.45.2.330 template<class T> NSVec4<T> NSVec4< T >::pqqq ( ) const [inline]

6.45.2.331 template<class T> NSVec4<T> NSVec4< T >::pqqs ( ) const [inline]

6.45.2.332 template<class T> NSVec4<T> NSVec4< T >::pqqt ( ) const [inline]

6.45.2.333 template<class T> NSVec3<T> NSVec4< T >::pqs ( ) const [inline]

6.45.2.334 template<class T> NSVec4<T> NSVec4< T >::pqsp ( ) const [inline]

6.45.2.335 template<class T> NSVec4<T> NSVec4< T >::pqsq ( ) const [inline]

6.45.2.336 template<class T> NSVec4<T> NSVec4< T >::pqss ( ) const [inline]

6.45.2.337 template<class T> NSVec4<T> NSVec4< T >::pqst ( ) const [inline]

6.45.2.338 template<class T> NSVec3<T> NSVec4< T >::pqt ( ) const [inline]

6.45.2.339 template<class T> NSVec4<T> NSVec4< T >::pqtp ( ) const [inline]

6.45.2.340 template<class T> NSVec4<T> NSVec4< T >::pqtq ( ) const [inline]

6.45.2.341 template<class T> NSVec4<T> NSVec4< T >::pqts ( ) const [inline]

6.45.2.342 template<class T> NSVec4<T> NSVec4< T >::pqtt ( ) const [inline]

6.45.2.343 template<class T> NSVec2<T> NSVec4< T >::ps ( ) const [inline]

6.45.2.344 template<class T> NSVec3<T> NSVec4< T >::psp ( ) const [inline]

6.45.2.345 template<class T> NSVec4<T> NSVec4< T >::pspp ( ) const [inline]

6.45.2.346 template<class T> NSVec4<T> NSVec4< T >::pspq ( ) const [inline]

6.45.2.347 template<class T> NSVec4<T> NSVec4< T >::psps ( ) const [inline]

6.45.2.348 template<class T> NSVec4<T> NSVec4< T >::pspt ( ) const [inline]

6.45.2.349 template<class T> NSVec3<T> NSVec4< T >::psq ( ) const [inline]

6.45.2.350 template<class T> NSVec4<T> NSVec4< T >::psqp ( ) const [inline]

6.45.2.351 template<class T> NSVec4<T> NSVec4< T >::psqq ( ) const [inline]

6.45.2.352 template<class T> NSVec4<T> NSVec4< T >::psqs ( ) const [inline]

6.45.2.353 template<class T> NSVec4<T> NSVec4< T >::psqt ( ) const [inline]

6.45.2.354 template<class T> NSVec3<T> NSVec4< T >::pss ( ) const [inline]

6.45.2.355 template<class T> NSVec4<T> NSVec4< T >::pssp ( ) const [inline]

6.45.2.356 template<class T> NSVec4<T> NSVec4< T >::pssq ( ) const [inline]

6.45.2.357 template<class T> NSVec4<T> NSVec4< T >::psss ( ) const [inline]

6.45.2.358 template<class T> NSVec4<T> NSVec4< T >::psst ( ) const [inline]

6.45.2.359 template<class T> NSVec3<T> NSVec4< T >::pst ( ) const [inline]

6.45.2.360 template<class T> NSVec4<T> NSVec4< T >::pstp ( ) const [inline]

6.45.2.361 template<class T> NSVec4<T> NSVec4< T >::pstq ( ) const [inline]

6.45.2.362 template<class T> NSVec4<T> NSVec4< T >::psts ( ) const [inline]

6.45.2.363 template<class T> NSVec4<T> NSVec4< T >::pstt ( ) const [inline]

6.45.2.364 template<class T> NSVec2<T> NSVec4< T >::pt ( ) const [inline]

6.45.2.365 template<class T> NSVec3<T> NSVec4< T >::ptp ( ) const [inline]

6.45.2.366 template<class T> NSVec4<T> NSVec4< T >::ptpp ( ) const [inline]

6.45.2.367 template<class T> NSVec4<T> NSVec4< T >::ptpq ( ) const [inline]

6.45.2.368 template<class T> NSVec4<T> NSVec4< T >::ptps ( ) const [inline]

6.45.2.369 template<class T> NSVec4<T> NSVec4< T >::ptpt ( ) const [inline]

6.45.2.370 template<class T> NSVec3<T> NSVec4< T >::ptq ( ) const [inline]

6.45.2.371 template<class T> NSVec4<T> NSVec4< T >::ptqp ( ) const [inline]

6.45.2.372 template<class T> NSVec4<T> NSVec4< T >::ptqq ( ) const [inline]

6.45.2.373 template<class T> NSVec4<T> NSVec4< T >::ptqs ( ) const [inline]

6.45.2.374 template<class T> NSVec4<T> NSVec4< T >::ptqt ( ) const [inline]

6.45.2.375 template<class T> NSVec3<T> NSVec4< T >::pts ( ) const [inline]

6.45.2.376 template<class T> NSVec4<T> NSVec4< T >::ptsp ( ) const [inline]

6.45.2.377 template<class T> NSVec4<T> NSVec4< T >::ptsq ( ) const [inline]

6.45.2.378 template<class T> NSVec4<T> NSVec4< T >::ptss ( ) const [inline]

6.45.2.379 template<class T> NSVec4<T> NSVec4< T >::ptst ( ) const [inline]

6.45.2.380 template<class T> NSVec3<T> NSVec4< T >::ptt ( ) const [inline]

6.45.2.381 template<class T> NSVec4<T> NSVec4< T >::pttp ( ) const [inline]

6.45.2.382 template<class T> NSVec4<T> NSVec4< T >::pttq ( ) const [inline]

6.45.2.383 template<class T> NSVec4<T> NSVec4< T >::ptts ( ) const [inline]

6.45.2.384 template<class T> NSVec4<T> NSVec4< T >::pttt ( ) const [inline]

6.45.2.385 template<class T> NSVec2<T> NSVec4< T >::qp ( ) const [inline]

6.45.2.386 template<class T> NSVec3<T> NSVec4< T >::qpp ( ) const [inline]

6.45.2.387 template<class T> NSVec4<T> NSVec4< T >::qppp ( ) const [inline]

6.45.2.388 template<class T> NSVec4<T> NSVec4< T >::qppq ( ) const [inline]

6.45.2.389 template<class T> NSVec4<T> NSVec4< T >::qpps ( ) const [inline]

6.45.2.390 template<class T> NSVec4<T> NSVec4< T >::qppt ( ) const [inline]

6.45.2.391 template<class T> NSVec3<T> NSVec4< T >::qpq ( ) const [inline]

6.45.2.392 template<class T> NSVec4<T> NSVec4< T >::qpqp ( ) const [inline]

6.45.2.393 template<class T> NSVec4<T> NSVec4< T >::qpqq ( ) const [inline]

6.45.2.394 template<class T> NSVec4<T> NSVec4< T >::qpqs ( ) const [inline]

6.45.2.395 template<class T> NSVec4<T> NSVec4< T >::qpqt ( ) const [inline]

6.45.2.396 template<class T> NSVec3<T> NSVec4< T >::qps ( ) const [inline]

6.45.2.397 template<class T> NSVec4<T> NSVec4< T >::qpsp ( ) const [inline]

6.45.2.398 template<class T> NSVec4<T> NSVec4< T >::qpsq ( ) const [inline]

6.45.2.399 template<class T> NSVec4<T> NSVec4< T >::qpss ( ) const [inline]

6.45.2.400 template<class T> NSVec4<T> NSVec4< T >::qpst ( ) const [inline]

6.45.2.401 template<class T> NSVec3<T> NSVec4< T >::qpt ( ) const [inline]

6.45.2.402 template<class T> NSVec4<T> NSVec4< T >::qptp ( ) const [inline]

6.45.2.403 template<class T> NSVec4<T> NSVec4< T >::qptq ( ) const [inline]

6.45.2.404 template<class T> NSVec4<T> NSVec4< T >::qpts ( ) const [inline]

6.45.2.405 template<class T> NSVec4<T> NSVec4< T >::qptt ( ) const [inline]

6.45.2.406 template<class T> NSVec2<T> NSVec4< T >::qq ( ) const [inline]

6.45.2.407 template<class T> NSVec3<T> NSVec4< T >::qqp ( ) const [inline]

6.45.2.408 template<class T> NSVec4<T> NSVec4< T >::qqpp ( ) const [inline]

6.45.2.409 template<class T> NSVec4<T> NSVec4< T >::qqpq ( ) const [inline]

6.45.2.410 template<class T> NSVec4<T> NSVec4< T >::qqps ( ) const [inline]

6.45.2.411 template<class T> NSVec4<T> NSVec4< T >::qqpt ( ) const [inline]

6.45.2.412 template<class T> NSVec3<T> NSVec4< T >::qqq ( ) const [inline]

6.45.2.413 template<class T> NSVec4<T> NSVec4< T >::qqqp ( ) const [inline]

6.45.2.414 template<class T> NSVec4<T> NSVec4< T >::qqqq ( ) const [inline]

6.45.2.415 template<class T> NSVec4<T> NSVec4< T >::qqqs ( ) const [inline]

6.45.2.416 template<class T> NSVec4<T> NSVec4< T >::qqqt ( ) const [inline]

6.45.2.417 template<class T> NSVec3<T> NSVec4< T >::qqs ( ) const [inline]

6.45.2.418 template<class T> NSVec4<T> NSVec4< T >::qqsp ( ) const [inline]

6.45.2.419 template<class T> NSVec4<T> NSVec4< T >::qqsq ( ) const [inline]

6.45.2.420 template<class T> NSVec4<T> NSVec4< T >::qqss ( ) const [inline]

6.45.2.421 template<class T> NSVec4<T> NSVec4< T >::qqst ( ) const [inline]

6.45.2.422 template<class T> NSVec3<T> NSVec4< T >::qqt ( ) const [inline]

6.45.2.423 template<class T> NSVec4<T> NSVec4< T >::qqtp ( ) const [inline]

6.45.2.424 template<class T> NSVec4<T> NSVec4< T >::qqtq ( ) const [inline]

6.45.2.425 template<class T> NSVec4<T> NSVec4< T >::qqts ( ) const [inline]

6.45.2.426 template<class T> NSVec4<T> NSVec4< T >::qqtt ( ) const [inline]

6.45.2.427 template<class T> NSVec2<T> NSVec4< T >::qs ( ) const [inline]

6.45.2.428 template<class T> NSVec3<T> NSVec4< T >::qsp ( ) const [inline]

6.45.2.429 template<class T> NSVec4<T> NSVec4< T >::qspp ( ) const [inline]

6.45.2.430 template<class T> NSVec4<T> NSVec4< T >::qspq ( ) const [inline]

6.45.2.431 template<class T> NSVec4<T> NSVec4< T >::qsps ( ) const [inline]

6.45.2.432 template<class T> NSVec4<T> NSVec4< T >::qspt ( ) const [inline]

6.45.2.433 template<class T> NSVec3<T> NSVec4< T >::qsq ( ) const [inline]

6.45.2.434 template<class T> NSVec4<T> NSVec4< T >::qsqp ( ) const [inline]

6.45.2.435 template<class T> NSVec4<T> NSVec4< T >::qsqq ( ) const [inline]

6.45.2.436 template<class T> NSVec4<T> NSVec4< T >::qsqs ( ) const [inline]

6.45.2.437 template<class T> NSVec4<T> NSVec4< T >::qsqt ( ) const [inline]

6.45.2.438 template<class T> NSVec3<T> NSVec4< T >::qss ( ) const [inline]

6.45.2.439 template<class T> NSVec4<T> NSVec4< T >::qssp ( ) const [inline]

6.45.2.440 template<class T> NSVec4<T> NSVec4< T >::qssq ( ) const [inline]

6.45.2.441 template<class T> NSVec4<T> NSVec4< T >::qsss ( ) const [inline]

6.45.2.442 template<class T> NSVec4<T> NSVec4< T >::qsst ( ) const [inline]

6.45.2.443 template<class T> NSVec3<T> NSVec4< T >::qst ( ) const [inline]

6.45.2.444 template<class T> NSVec4<T> NSVec4< T >::qstp ( ) const [inline]

6.45.2.445 template<class T> NSVec4<T> NSVec4< T >::qstq ( ) const [inline]

6.45.2.446 template<class T> NSVec4<T> NSVec4< T >::qsts ( ) const [inline]

6.45.2.447 template<class T> NSVec4<T> NSVec4< T >::qstt ( ) const [inline]

6.45.2.448 template<class T> NSVec2<T> NSVec4< T >::qt ( ) const [inline]

6.45.2.449 template<class T> NSVec3<T> NSVec4< T >::qtp ( ) const [inline]

6.45.2.450 template<class T> NSVec4<T> NSVec4< T >::qtpp ( ) const [inline]

6.45.2.451 template<class T> NSVec4<T> NSVec4< T >::qtpq ( ) const [inline]

6.45.2.452 template<class T> NSVec4<T> NSVec4< T >::qtps ( ) const [inline]

6.45.2.453 template<class T> NSVec4<T> NSVec4< T >::qtpt ( ) const [inline]

6.45.2.454 template<class T> NSVec3<T> NSVec4< T >::qtq ( ) const [inline]

6.45.2.455 template<class T> NSVec4<T> NSVec4< T >::qtqp ( ) const [inline]

6.45.2.456 template<class T> NSVec4<T> NSVec4< T >::qtqq ( ) const [inline]

6.45.2.457 template<class T> NSVec4<T> NSVec4< T >::qtqs ( ) const [inline]

6.45.2.458 template<class T> NSVec4<T> NSVec4< T >::qtqt ( ) const [inline]

6.45.2.459 template<class T> NSVec3<T> NSVec4< T >::qts ( ) const [inline]

6.45.2.460 template<class T> NSVec4<T> NSVec4< T >::qtsp ( ) const [inline]

6.45.2.461 template<class T> NSVec4<T> NSVec4< T >::qtsq ( ) const [inline]

6.45.2.462 template<class T> NSVec4<T> NSVec4< T >::qtss ( ) const [inline]

6.45.2.463 template<class T> NSVec4<T> NSVec4< T >::qtst ( ) const [inline]

6.45.2.464 template<class T> NSVec3<T> NSVec4< T >::qtt ( ) const [inline]

6.45.2.465 template<class T> NSVec4<T> NSVec4< T >::qttp ( ) const [inline]

6.45.2.466 template<class T> NSVec4<T> NSVec4< T >::qttq ( ) const [inline]

6.45.2.467 template<class T> NSVec4<T> NSVec4< T >::qtts ( ) const [inline]

6.45.2.468 template<class T> NSVec4<T> NSVec4< T >::qttt ( ) const [inline]

6.45.2.469 template<class T> NSVec2<T> NSVec4< T >::ra ( ) const [inline]

6.45.2.470 template<class T> NSVec3<T> NSVec4< T >::raa ( ) const [inline]

6.45.2.471 template<class T> NSVec4<T> NSVec4< T >::raaa ( ) const [inline]

6.45.2.472 template<class T> NSVec4<T> NSVec4< T >::raab ( ) const [inline]

6.45.2.473 template<class T> NSVec4<T> NSVec4< T >::raag ( ) const [inline]

6.45.2.474 template<class T> NSVec4<T> NSVec4< T >::raar ( ) const [inline]

6.45.2.475 template<class T> NSVec3<T> NSVec4< T >::rab ( ) const [inline]

6.45.2.476 template<class T> NSVec4<T> NSVec4< T >::raba ( ) const [inline]

6.45.2.477 template<class T> NSVec4<T> NSVec4< T >::rabb ( ) const [inline]

6.45.2.478 template<class T> NSVec4<T> NSVec4< T >::rabg ( ) const [inline]

6.45.2.479 template<class T> NSVec4<T> NSVec4< T >::rabr ( ) const [inline]

6.45.2.480 template<class T> NSVec3<T> NSVec4< T >::rag ( ) const [inline]

6.45.2.481 template<class T> NSVec4<T> NSVec4< T >::raga ( ) const [inline]

6.45.2.482 template<class T> NSVec4<T> NSVec4< T >::ragb ( ) const [inline]

6.45.2.483 template<class T> NSVec4<T> NSVec4< T >::ragg ( ) const [inline]

6.45.2.484 template<class T> NSVec4<T> NSVec4< T >::ragr ( ) const [inline]

6.45.2.485 template<class T> NSVec3<T> NSVec4< T >::rar ( ) const [inline]

6.45.2.486 template<class T> NSVec4<T> NSVec4< T >::rara ( ) const [inline]

6.45.2.487 template<class T> NSVec4<T> NSVec4< T >::rarb ( ) const [inline]

6.45.2.488 template<class T> NSVec4<T> NSVec4< T >::rarg ( ) const [inline]

6.45.2.489 template<class T> NSVec4<T> NSVec4< T >::rarr ( ) const [inline]

6.45.2.490 template<class T> NSVec2<T> NSVec4< T >::rb ( ) const [inline]

6.45.2.491 template<class T> NSVec3<T> NSVec4< T >::rba ( ) const [inline]

6.45.2.492 template<class T> NSVec4<T> NSVec4< T >::rbaa ( ) const [inline]

6.45.2.493 template<class T> NSVec4<T> NSVec4< T >::rbab ( ) const [inline]

6.45.2.494 template<class T> NSVec4<T> NSVec4< T >::rbag ( ) const [inline]

6.45.2.495 template<class T> NSVec4<T> NSVec4< T >::rbar ( ) const [inline]

6.45.2.496 template<class T> NSVec3<T> NSVec4< T >::rbb ( ) const [inline]

6.45.2.497 template<class T> NSVec4<T> NSVec4< T >::rbba ( ) const [inline]

6.45.2.498 template<class T> NSVec4<T> NSVec4< T >::rbbb ( ) const [inline]

6.45.2.499 template<class T> NSVec4<T> NSVec4< T >::rbbg ( ) const [inline]

6.45.2.500 template<class T> NSVec4<T> NSVec4< T >::rbbr ( ) const [inline]

6.45.2.501 template<class T> NSVec3<T> NSVec4< T >::rbg ( ) const [inline]

6.45.2.502 template<class T> NSVec4<T> NSVec4< T >::rbga ( ) const [inline]

6.45.2.503 template<class T> NSVec4<T> NSVec4< T >::rbgb ( ) const [inline]

6.45.2.504 template<class T> NSVec4<T> NSVec4< T >::rbgg ( ) const [inline]

6.45.2.505 template<class T> NSVec4<T> NSVec4< T >::rbgr ( ) const [inline]

6.45.2.506 template<class T> NSVec3<T> NSVec4< T >::rbr ( ) const [inline]

6.45.2.507 template<class T> NSVec4<T> NSVec4< T >::rbra ( ) const [inline]

6.45.2.508 template<class T> NSVec4<T> NSVec4< T >::rbrb ( ) const [inline]

6.45.2.509 template<class T> NSVec4<T> NSVec4< T >::rbrg ( ) const [inline]

6.45.2.510 template<class T> NSVec4<T> NSVec4< T >::rbrr ( ) const [inline]

6.45.2.511 template<class T> NSVec2<T> NSVec4< T >::rg ( ) const [inline]

6.45.2.512 template<class T> NSVec3<T> NSVec4< T >::rga ( ) const [inline]

6.45.2.513 template<class T> NSVec4<T> NSVec4< T >::rgaa ( ) const [inline]

6.45.2.514 template<class T> NSVec4<T> NSVec4< T >::rgab ( ) const [inline]

6.45.2.515 template<class T> NSVec4<T> NSVec4< T >::rgag ( ) const [inline]

6.45.2.516 template<class T> NSVec4<T> NSVec4< T >::rgar ( ) const [inline]

6.45.2.517 template<class T> NSVec3<T> NSVec4< T >::rgb ( ) const [inline]

6.45.2.518 template<class T> NSVec4<T> NSVec4< T >::rgbb ( ) const [inline]

6.45.2.519 template<class T> NSVec4<T> NSVec4< T >::rgbg ( ) const [inline]

6.45.2.520 template<class T> NSVec4<T> NSVec4< T >::rgbr ( ) const [inline]

6.45.2.521 template<class T> NSVec3<T> NSVec4< T >::rgg ( ) const [inline]

6.45.2.522 template<class T> NSVec4<T> NSVec4< T >::rgga ( ) const [inline]

6.45.2.523 template<class T> NSVec4<T> NSVec4< T >::rggb ( ) const [inline]

6.45.2.524 template<class T> NSVec4<T> NSVec4< T >::rggg ( ) const [inline]

6.45.2.525 template<class T> NSVec4<T> NSVec4< T >::rggr ( ) const [inline]

6.45.2.526 template<class T> NSVec3<T> NSVec4< T >::rgr ( ) const [inline]

6.45.2.527 template<class T> NSVec4<T> NSVec4< T >::rgra ( ) const [inline]

6.45.2.528 template<class T> NSVec4<T> NSVec4< T >::rgrb ( ) const [inline]

6.45.2.529 template<class T> NSVec4<T> NSVec4< T >::rgrg ( ) const [inline]

6.45.2.530 template<class T> NSVec4<T> NSVec4< T >::rgrr ( ) const [inline]

6.45.2.531 template<class T> NSVec4<T>& NSVec4< T >::round ( ) [inline]

6.45.2.532 template<class T> NSVec4<T>& NSVec4< T >::roundToZero ( ) [inline]

6.45.2.533 template<class T> NSVec2<T> NSVec4< T >::rr ( ) const [inline]

6.45.2.534 template<class T> NSVec3<T> NSVec4< T >::rra ( ) const [inline]

6.45.2.535 template<class T> NSVec4<T> NSVec4< T >::rraa ( ) const [inline]

6.45.2.536 template<class T> NSVec4<T> NSVec4< T >::rrab ( ) const [inline]

6.45.2.537 template<class T> NSVec4<T> NSVec4< T >::rrag ( ) const [inline]

6.45.2.538 template<class T> NSVec4<T> NSVec4< T >::rrar ( ) const [inline]

6.45.2.539 template<class T> NSVec3<T> NSVec4< T >::rrb ( ) const [inline]

6.45.2.540 template<class T> NSVec4<T> NSVec4< T >::rrba ( ) const [inline]

6.45.2.541 template<class T> NSVec4<T> NSVec4< T >::rrbb ( ) const [inline]

6.45.2.542 template<class T> NSVec4<T> NSVec4< T >::rrbg ( ) const [inline]

6.45.2.543 template<class T> NSVec4<T> NSVec4< T >::rrbr ( ) const [inline]

6.45.2.544 template<class T> NSVec3<T> NSVec4< T >::rrg ( ) const [inline]

6.45.2.545 template<class T> NSVec4<T> NSVec4< T >::rrga ( ) const [inline]

6.45.2.546 template<class T> NSVec4<T> NSVec4< T >::rrgb ( ) const [inline]

6.45.2.547 template<class T> NSVec4<T> NSVec4< T >::rrgg ( ) const [inline]

6.45.2.548 template<class T> NSVec4<T> NSVec4< T >::rrgr ( ) const [inline]

6.45.2.549 template<class T> NSVec3<T> NSVec4< T >::rrr ( ) const [inline]

6.45.2.550 template<class T> NSVec4<T> NSVec4< T >::rrra ( ) const [inline]

6.45.2.551 template<class T> NSVec4<T> NSVec4< T >::rrrb ( ) const [inline]

6.45.2.552 template<class T> NSVec4<T> NSVec4< T >::rrrg ( ) const [inline]

6.45.2.553 template<class T> NSVec4<T> NSVec4< T >::rrrr ( ) const [inline]

6.45.2.554 template<class T> NSVec3<T>& NSVec4< T >::scalingFrom (const nsmat3< T > & *transform*) [inline]

6.45.2.555 template<class T> NSVec3<T>& NSVec4< T >::scalingFrom (const nsmat4< T > & *transform*) [inline]

6.45.2.556 template<class T> NSVec4<T>& NSVec4< T >::set (const T & *pVal*) [inline]

6.45.2.557 template<class T> NSVec4<T>& NSVec4< T >::set (const T & *pX,* const T & *pY,* const T & *pZ,* const T & *pW*) [inline]

6.45.2.558 template<class T> NSVec4<T>& NSVec4< T >::set (const NSVec3< T > & *xyz,* const T & *pW*) [inline]

6.45.2.559 template<class T> NSVec4<T>& NSVec4< T >::set (const T & *pX,* const NSVec3< T > & *yzw*) [inline]

6.45.2.560 template<class T> NSVec4<T>& NSVec4< T >::set (const NSVec2< T > & *xy,* const T & *pZ,* const T & *pW*) [inline]

6.45.2.561 template<class T> NSVec4<T>& NSVec4< T >::set (const T & *pX,* const NSVec2< T > & *yz,* const T & *pW*) [inline]

6.45.2.562 template<class T> NSVec4<T>& NSVec4< T >::set (const T & *pX,* const T & *pY,* const NSVec2< T > & *zw*) [inline]

6.45.2.563 template<class T> NSVec4<T>& NSVec4< T >::setLength (const T & *len*) [inline]

6.45.2.564 template<class T> NSVec2<T> NSVec4< T >::sp ( ) const [inline]

6.45.2.565 template<class T> NSVec3<T> NSVec4< T >::spp ( ) const [inline]

6.45.2.566 template<class T> NSVec4<T> NSVec4< T >::sppp ( ) const [inline]

6.45.2.567 template<class T> NSVec4<T> NSVec4< T >::sppq ( ) const [inline]

6.45.2.568 template<class T> NSVec4<T> NSVec4< T >::spps ( ) const [inline]

6.45.2.569 template<class T> NSVec4<T> NSVec4< T >::sppt ( ) const [inline]

6.45.2.570 template<class T> NSVec3<T> NSVec4< T >::spq ( ) const [inline]

6.45.2.571 template<class T> NSVec4<T> NSVec4< T >::spqp ( ) const [inline]

6.45.2.572 template<class T> NSVec4<T> NSVec4< T >::spqq ( ) const [inline]

6.45.2.573 template<class T> NSVec4<T> NSVec4< T >::spqs ( ) const [inline]

6.45.2.574 template<class T> NSVec4<T> NSVec4< T >::spqt ( ) const [inline]

6.45.2.575 template<class T> NSVec3<T> NSVec4< T >::sps ( ) const [inline]

6.45.2.576 template<class T> NSVec4<T> NSVec4< T >::spsp ( ) const [inline]

6.45.2.577 template<class T> NSVec4<T> NSVec4< T >::spsq ( ) const [inline]

6.45.2.578 template<class T> NSVec4<T> NSVec4< T >::spss ( ) const [inline]

6.45.2.579 template<class T> NSVec4<T> NSVec4< T >::spst ( ) const [inline]

6.45.2.580 template<class T> NSVec3<T> NSVec4< T >::spt ( ) const [inline]

6.45.2.581 template<class T> NSVec4<T> NSVec4< T >::sptp ( ) const [inline]

6.45.2.582 template<class T> NSVec4<T> NSVec4< T >::sptq ( ) const [inline]

6.45.2.583 template<class T> NSVec4<T> NSVec4< T >::spts ( ) const [inline]

6.45.2.584 template<class T> NSVec4<T> NSVec4< T >::sptt ( ) const [inline]

6.45.2.585 template<class T> NSVec2<T> NSVec4< T >::sq ( ) const [inline]

6.45.2.586 template<class T> NSVec3<T> NSVec4< T >::sqp ( ) const [inline]

6.45.2.587 template<class T> NSVec4<T> NSVec4< T >::sqpp ( ) const [inline]

6.45.2.588 template<class T> NSVec4<T> NSVec4< T >::sqpq ( ) const [inline]

6.45.2.589 template<class T> NSVec4<T> NSVec4< T >::sqps ( ) const [inline]

6.45.2.590 template<class T> NSVec4<T> NSVec4< T >::sqpt ( ) const [inline]

6.45.2.591 template<class T> NSVec3<T> NSVec4< T >::sqq ( ) const [inline]

6.45.2.592 template<class T> NSVec4<T> NSVec4< T >::sqqp ( ) const [inline]

6.45.2.593 template<class T> NSVec4<T> NSVec4< T >::sqqq ( ) const [inline]

6.45.2.594 template<class T> NSVec4<T> NSVec4< T >::sqqs ( ) const [inline]

6.45.2.595 template<class T> NSVec4<T> NSVec4< T >::sqqt ( ) const [inline]

6.45.2.596 template<class T> NSVec3<T> NSVec4< T >::sqs ( ) const [inline]

6.45.2.597 template<class T> NSVec4<T> NSVec4< T >::sqsp ( ) const [inline]

6.45.2.598 template<class T> NSVec4<T> NSVec4< T >::sqsq ( ) const [inline]

6.45.2.599 template<class T> NSVec4<T> NSVec4< T >::sqss ( ) const [inline]

6.45.2.600 template<class T> NSVec4<T> NSVec4< T >::sqst ( ) const [inline]

6.45.2.601 template<class T> NSVec3<T> NSVec4< T >::sqt ( ) const [inline]

6.45.2.602 template<class T> NSVec4<T> NSVec4< T >::sqtp ( ) const [inline]

6.45.2.603 template<class T> NSVec4<T> NSVec4< T >::sqtq ( ) const [inline]

6.45.2.604 template<class T> NSVec4<T> NSVec4< T >::sqts ( ) const [inline]

6.45.2.605 template<class T> NSVec4<T> NSVec4< T >::sqtt ( ) const [inline]

6.45.2.606 template<class T> NSVec2<T> NSVec4< T >::ss ( ) const [inline]

6.45.2.607 template<class T> NSVec3<T> NSVec4< T >::ssp ( ) const [inline]

6.45.2.608 template<class T> NSVec4<T> NSVec4< T >::sspp ( ) const [inline]

6.45.2.609 template<class T> NSVec4<T> NSVec4< T >::sspq ( ) const [inline]

6.45.2.610 template<class T> NSVec4<T> NSVec4< T >::ssps ( ) const [inline]

6.45.2.611 template<class T> NSVec4<T> NSVec4< T >::sspt ( ) const [inline]

6.45.2.612 template<class T> NSVec3<T> NSVec4< T >::ssq ( ) const [inline]

6.45.2.613 template<class T> NSVec4<T> NSVec4< T >::ssqp ( ) const [inline]

6.45.2.614 template<class T> NSVec4<T> NSVec4< T >::ssqq ( ) const [inline]

6.45.2.615 template<class T> NSVec4<T> NSVec4< T >::ssqs ( ) const [inline]

6.45.2.616 template<class T> NSVec4<T> NSVec4< T >::ssqt ( ) const [inline]

6.45.2.617 template<class T> NSVec3<T> NSVec4< T >::sss ( ) const [inline]

6.45.2.618 template<class T> NSVec4<T> NSVec4< T >::sssp ( ) const [inline]

6.45.2.619 template<class T> NSVec4<T> NSVec4< T >::sssq ( ) const [inline]

6.45.2.620 template<class T> NSVec4<T> NSVec4< T >::ssss ( ) const [inline]

6.45.2.621 template<class T> NSVec4<T> NSVec4< T >::ssst ( ) const [inline]

6.45.2.622 template<class T> NSVec3<T> NSVec4< T >::sst ( ) const [inline]

6.45.2.623 template<class T> NSVec4<T> NSVec4< T >::sstp ( ) const [inline]

6.45.2.624 template<class T> NSVec4<T> NSVec4< T >::sstq ( ) const [inline]

6.45.2.625 template<class T> NSVec4<T> NSVec4< T >::ssts ( ) const [inline]

6.45.2.626 template<class T> NSVec4<T> NSVec4< T >::sstt ( ) const [inline]

6.45.2.627 template<class T> NSVec2<T> NSVec4< T >::st ( ) const [inline]

6.45.2.628 template<class T> NSVec3<T> NSVec4< T >::stp ( ) const [inline]

6.45.2.629 template<class T> NSVec4<T> NSVec4< T >::stpp ( ) const [inline]

6.45.2.630 template<class T> NSVec4<T> NSVec4< T >::stps ( ) const [inline]

6.45.2.631 template<class T> NSVec4<T> NSVec4< T >::stpt ( ) const [inline]

6.45.2.632 template<class T> NSVec3<T> NSVec4< T >::stq ( ) const [inline]

6.45.2.633 template<class T> NSVec4<T> NSVec4< T >::stqp ( ) const [inline]

6.45.2.634 template<class T> NSVec4<T> NSVec4< T >::stqq ( ) const [inline]

6.45.2.635 template<class T> NSVec4<T> NSVec4< T >::stqs ( ) const [inline]

6.45.2.636 template<class T> NSVec4<T> NSVec4< T >::stqt ( ) const [inline]

6.45.2.637 template<class T> NSVec3<T> NSVec4< T >::sts ( ) const [inline]

6.45.2.638 template<class T> NSVec4<T> NSVec4< T >::stsp ( ) const [inline]

6.45.2.639 template<class T> NSVec4<T> NSVec4< T >::stsq ( ) const [inline]

6.45.2.640 template<class T> NSVec4<T> NSVec4< T >::stss ( ) const [inline]

6.45.2.641 template<class T> NSVec4<T> NSVec4< T >::stst ( ) const [inline]

6.45.2.642 template<class T> NSVec3<T> NSVec4< T >::stt ( ) const [inline]

6.45.2.643 template<class T> NSVec4<T> NSVec4< T >::sttp ( ) const [inline]

6.45.2.644 template<class T> NSVec4<T> NSVec4< T >::sttq ( ) const [inline]

6.45.2.645 template<class T> NSVec4<T> NSVec4< T >::stts ( ) const [inline]

6.45.2.646 template<class T> NSVec4<T> NSVec4< T >::sttt ( ) const [inline]

6.45.2.647 template<class T> std::string NSVec4< T >::toString ( ) [inline]

6.45.2.648 template<class T> NSVec2<T> NSVec4< T >::tp ( ) const [inline]

6.45.2.649 template<class T> NSVec3<T> NSVec4< T >::tpp ( ) const [inline]

6.45.2.650 template<class T> NSVec4<T> NSVec4< T >::tppp ( ) const [inline]

6.45.2.651 template<class T> NSVec4<T> NSVec4< T >::tppq ( ) const [inline]

6.45.2.652 template<class T> NSVec4<T> NSVec4< T >::tpps ( ) const [inline]

6.45.2.653 template<class T> NSVec4<T> NSVec4< T >::tppt ( ) const [inline]

6.45.2.654 template<class T> NSVec3<T> NSVec4< T >::tpq ( ) const [inline]

6.45.2.655 template<class T> NSVec4<T> NSVec4< T >::tpqp ( ) const [inline]

6.45.2.656 template<class T> NSVec4<T> NSVec4< T >::tpqq ( ) const [inline]

6.45.2.657 template<class T> NSVec4<T> NSVec4< T >::tpqs ( ) const [inline]

6.45.2.658 template<class T> NSVec4<T> NSVec4< T >::tpqt ( ) const [inline]

6.45.2.659 template<class T> NSVec3<T> NSVec4< T >::tps ( ) const [inline]

6.45.2.660 template<class T> NSVec4<T> NSVec4< T >::tpsp ( ) const [inline]

6.45.2.661 template<class T> NSVec4<T> NSVec4< T >::tpsq ( ) const [inline]

6.45.2.662 template<class T> NSVec4<T> NSVec4< T >::tpss ( ) const [inline]

6.45.2.663 template<class T> NSVec4<T> NSVec4< T >::tpst ( ) const [inline]

6.45.2.664 template<class T> NSVec3<T> NSVec4< T >::tpt ( ) const [inline]

6.45.2.665 template<class T> NSVec4<T> NSVec4< T >::tptp ( ) const [inline]

6.45.2.666 template<class T> NSVec4<T> NSVec4< T >::tptq ( ) const [inline]

6.45.2.667 template<class T> NSVec4<T> NSVec4< T >::tpts ( ) const [inline]

6.45.2.668 template<class T> NSVec4<T> NSVec4< T >::tptt ( ) const [inline]

6.45.2.669 template<class T> NSVec2<T> NSVec4< T >::tq ( ) const [inline]

6.45.2.670 template<class T> NSVec3<T> NSVec4< T >::tqp ( ) const [inline]

6.45.2.671 template<class T> NSVec4<T> NSVec4< T >::tqpp ( ) const [inline]

6.45.2.672 template<class T> NSVec4<T> NSVec4< T >::tqpq ( ) const [inline]

6.45.2.673 template<class T> NSVec4<T> NSVec4< T >::tqps ( ) const [inline]

6.45.2.674 template<class T> NSVec4<T> NSVec4< T >::tqpt ( ) const [inline]

6.45.2.675 template<class T> NSVec3<T> NSVec4< T >::tqq ( ) const [inline]

6.45.2.676 template<class T> NSVec4<T> NSVec4< T >::tqqp ( ) const [inline]

6.45.2.677 template<class T> NSVec4<T> NSVec4< T >::tqqq ( ) const [inline]

6.45.2.678 template<class T> NSVec4<T> NSVec4< T >::tqqs ( ) const [inline]

6.45.2.679 template<class T> NSVec4<T> NSVec4< T >::tqqt ( ) const [inline]

6.45.2.680 template<class T> NSVec3<T> NSVec4< T >::tqs ( ) const [inline]

6.45.2.681 template<class T> NSVec4<T> NSVec4< T >::tqsp ( ) const [inline]

6.45.2.682 template<class T> NSVec4<T> NSVec4< T >::tqsq ( ) const [inline]

6.45.2.683 template<class T> NSVec4<T> NSVec4< T >::tqss ( ) const [inline]

6.45.2.684 template<class T> NSVec4<T> NSVec4< T >::tqst ( ) const [inline]

6.45.2.685 template<class T> NSVec3<T> NSVec4< T >::tqt ( ) const [inline]

6.45.2.686 template<class T> NSVec4<T> NSVec4< T >::tqtp ( ) const [inline]

6.45.2.687 template<class T> NSVec4<T> NSVec4< T >::tqtq ( ) const [inline]

6.45.2.688 template<class T> NSVec4<T> NSVec4< T >::tqts ( ) const [inline]

6.45.2.689 template<class T> NSVec4<T> NSVec4< T >::tqtt ( ) const [inline]

6.45.2.690 template<class T> NSVec4<T>& NSVec4< T >::translationFrom (const nsmat4< T > & *transform*) [inline]

6.45.2.691 template<class T> NSVec2<T> NSVec4< T >::ts ( ) const [inline]

6.45.2.692 template<class T> NSVec3<T> NSVec4< T >::tsp ( ) const [inline]

6.45.2.693 template<class T> NSVec4<T> NSVec4< T >::tspp ( ) const [inline]

6.45.2.694 template<class T> NSVec4<T> NSVec4< T >::tspq ( ) const [inline]

6.45.2.695 template<class T> NSVec4<T> NSVec4< T >::tsps ( ) const [inline]

6.45.2.696 template<class T> NSVec4<T> NSVec4< T >::tspt ( ) const [inline]

6.45.2.697 template<class T> NSVec3<T> NSVec4< T >::tsq ( ) const [inline]

6.45.2.698 template<class T> NSVec4<T> NSVec4< T >::tsqp ( ) const [inline]

6.45.2.699 template<class T> NSVec4<T> NSVec4< T >::tsqq ( ) const [inline]

6.45.2.700 template<class T> NSVec4<T> NSVec4< T >::tsqs ( ) const [inline]

6.45.2.701 template<class T> NSVec4<T> NSVec4< T >::tsqt ( ) const [inline]

6.45.2.702 template<class T> NSVec3<T> NSVec4< T >::tss ( ) const [inline]

6.45.2.703 template<class T> NSVec4<T> NSVec4< T >::tssp ( ) const [inline]

6.45.2.704 template<class T> NSVec4<T> NSVec4< T >::tssq ( ) const [inline]

6.45.2.705 template<class T> NSVec4<T> NSVec4< T >::tsss ( ) const [inline]

6.45.2.706 template<class T> NSVec4<T> NSVec4< T >::tsst ( ) const [inline]

6.45.2.707 template<class T> NSVec3<T> NSVec4< T >::tst ( ) const [inline]

6.45.2.708 template<class T> NSVec4<T> NSVec4< T >::tstp ( ) const [inline]

6.45.2.709 template<class T> NSVec4<T> NSVec4< T >::tstq ( ) const [inline]

6.45.2.710 template<class T> NSVec4<T> NSVec4< T >::tsts ( ) const [inline]

6.45.2.711 template<class T> NSVec4<T> NSVec4< T >::tstt ( ) const [inline]

6.45.2.712 template<class T> NSVec2<T> NSVec4< T >::tt ( ) const [inline]

6.45.2.713 template<class T> NSVec3<T> NSVec4< T >::ttp ( ) const [inline]

6.45.2.714 template<class T> NSVec4<T> NSVec4< T >::ttpp ( ) const [inline]

6.45.2.715 template<class T> NSVec4<T> NSVec4< T >::ttpq ( ) const [inline]

6.45.2.716 template<class T> NSVec4<T> NSVec4< T >::ttps ( ) const [inline]

6.45.2.717 template<class T> NSVec4<T> NSVec4< T >::ttpt ( ) const [inline]

6.45.2.718 template<class T> NSVec3<T> NSVec4< T >::ttq ( ) const [inline]

6.45.2.719 template<class T> NSVec4<T> NSVec4< T >::ttqp ( ) const [inline]

6.45.2.720 template<class T> NSVec4<T> NSVec4< T >::ttqq ( ) const [inline]

6.45.2.721 template<class T> NSVec4<T> NSVec4< T >::ttqs ( ) const [inline]

6.45.2.722 template<class T> NSVec4<T> NSVec4< T >::ttqt ( ) const [inline]

6.45.2.723 template<class T> NSVec3<T> NSVec4< T >::tts ( ) const [inline]

6.45.2.724 template<class T> NSVec4<T> NSVec4< T >::ttsp ( ) const [inline]

6.45.2.725 template<class T> NSVec4<T> NSVec4< T >::ttsq ( ) const [inline]

6.45.2.726 template<class T> NSVec4<T> NSVec4< T >::ttss ( ) const [inline]

6.45.2.727 template<class T> NSVec4<T> NSVec4< T >::ttst ( ) const [inline]

6.45.2.728 template<class T> NSVec3<T> NSVec4< T >::ttt ( ) const [inline]

6.45.2.729 template<class T> NSVec4<T> NSVec4< T >::tttp ( ) const [inline]

6.45.2.730 template<class T> NSVec4<T> NSVec4< T >::tttq ( ) const [inline]

6.45.2.731 template<class T> NSVec4<T> NSVec4< T >::ttts ( ) const [inline]

6.45.2.732 template<class T> NSVec4<T> NSVec4< T >::tttt ( ) const [inline]

6.45.2.733 template<class T> NSVec2<T> NSVec4< T >::ww ( ) const [inline]

6.45.2.734 template<class T> NSVec3<T> NSVec4< T >::www ( ) const [inline]

6.45.2.735 template<class T> NSVec4<T> NSVec4< T >::wwww ( ) const [inline]

6.45.2.736 template<class T> NSVec4<T> NSVec4< T >::wwwx ( ) const [inline]

6.45.2.737 template<class T> NSVec4<T> NSVec4< T >::wwwy ( ) const [inline]

6.45.2.738 template<class T> NSVec4<T> NSVec4< T >::wwwz ( ) const [inline]

6.45.2.739 template<class T> NSVec3<T> NSVec4< T >::wwx ( ) const [inline]

6.45.2.740 template<class T> NSVec4<T> NSVec4< T >::wwxw ( ) const [inline]

6.45.2.741 template<class T> NSVec4<T> NSVec4< T >::wwxx ( ) const [inline]

6.45.2.742 template<class T> NSVec4<T> NSVec4< T >::wwxy ( ) const [inline]

6.45.2.743 template<class T> NSVec4<T> NSVec4< T >::wwxz ( ) const [inline]

6.45.2.744 template<class T> NSVec3<T> NSVec4< T >::wwy ( ) const [inline]

6.45.2.745 template<class T> NSVec4<T> NSVec4< T >::wwyw ( ) const [inline]

6.45.2.746 template<class T> NSVec4<T> NSVec4< T >::wwyx ( ) const [inline]

6.45.2.747 template<class T> NSVec4<T> NSVec4< T >::wwyy ( ) const [inline]

6.45.2.748 template<class T> NSVec4<T> NSVec4< T >::wwyz ( ) const [inline]

6.45.2.749 template<class T> NSVec3<T> NSVec4< T >::wwz ( ) const [inline]

6.45.2.750 template<class T> NSVec4<T> NSVec4< T >::wwzw ( ) const [inline]

6.45.2.751 template<class T> NSVec4<T> NSVec4< T >::wwzx ( ) const [inline]

6.45.2.752 template<class T> NSVec4<T> NSVec4< T >::wwzy ( ) const [inline]

6.45.2.753 template<class T> NSVec4<T> NSVec4< T >::wwzz ( ) const [inline]

6.45.2.754 template<class T> NSVec2<T> NSVec4< T >::wx ( ) const [inline]

6.45.2.755 template<class T> NSVec3<T> NSVec4< T >::wxw ( ) const [inline]

6.45.2.756 template<class T> NSVec4<T> NSVec4< T >::wxww ( ) const [inline]

6.45.2.757 template<class T> NSVec4<T> NSVec4< T >::wxwx ( ) const [inline]

6.45.2.758 template<class T> NSVec4<T> NSVec4< T >::wxwy ( ) const [inline]

6.45.2.759 template<class T> NSVec4<T> NSVec4< T >::wxwz ( ) const [inline]

6.45.2.760 template<class T> NSVec3<T> NSVec4< T >::wxx ( ) const [inline]

6.45.2.761 template<class T> NSVec4<T> NSVec4< T >::wxxw ( ) const [inline]

6.45.2.762 template<class T> NSVec4<T> NSVec4< T >::wxxx ( ) const [inline]

6.45.2.763 template<class T> NSVec4<T> NSVec4< T >::wxxy ( ) const [inline]

6.45.2.764 template<class T> NSVec4<T> NSVec4< T >::wxxz ( ) const [inline]

6.45.2.765 template<class T> NSVec3<T> NSVec4< T >::wxy ( ) const [inline]

6.45.2.766 template<class T> NSVec4<T> NSVec4< T >::wxyw ( ) const [inline]

6.45.2.767 template<class T> NSVec4<T> NSVec4< T >::wxyx ( ) const [inline]

6.45.2.768 template<class T> NSVec4<T> NSVec4< T >::wxyy ( ) const [inline]

6.45.2.769 template<class T> NSVec4<T> NSVec4< T >::wxyz ( ) const [inline]

6.45.2.770 template<class T> NSVec3<T> NSVec4< T >::wxz ( ) const [inline]

6.45.2.771 template<class T> NSVec4<T> NSVec4< T >::wxzw ( ) const [inline]

6.45.2.772 template<class T> NSVec4<T> NSVec4< T >::wxzx ( ) const [inline]

6.45.2.773 template<class T> NSVec4<T> NSVec4< T >::wxzy ( ) const [inline]

6.45.2.774 template<class T> NSVec4<T> NSVec4< T >::wxzz ( ) const [inline]

6.45.2.775 template<class T> NSVec2<T> NSVec4< T >::wy ( ) const [inline]

6.45.2.776 template<class T> NSVec3<T> NSVec4< T >::wyw ( ) const [inline]

6.45.2.777 template<class T> NSVec4<T> NSVec4< T >::wyww ( ) const [inline]

6.45.2.778 template<class T> NSVec4<T> NSVec4< T >::wywx ( ) const [inline]

6.45.2.779 template<class T> NSVec4<T> NSVec4< T >::wywy ( ) const [inline]

6.45.2.780 template<class T> NSVec4<T> NSVec4< T >::wywz ( ) const [inline]

6.45.2.781 template<class T> NSVec3<T> NSVec4< T >::wyx ( ) const [inline]

6.45.2.782 template<class T> NSVec4<T> NSVec4< T >::wyxw ( ) const [inline]

6.45.2.783 template<class T> NSVec4<T> NSVec4< T >::wyxx ( ) const [inline]

6.45.2.784 template<class T> NSVec4<T> NSVec4< T >::wyxy ( ) const [inline]

6.45.2.785 template<class T> NSVec4<T> NSVec4< T >::wyxz ( ) const [inline]

6.45.2.786 template<class T> NSVec3<T> NSVec4< T >::wyy ( ) const [inline]

6.45.2.787 template<class T> NSVec4<T> NSVec4< T >::wyyw ( ) const [inline]

6.45.2.788 template<class T> NSVec4<T> NSVec4< T >::wyyx ( ) const [inline]

6.45.2.789 template<class T> NSVec4<T> NSVec4< T >::wyyy ( ) const [inline]

6.45.2.790 template<class T> NSVec4<T> NSVec4< T >::wyyz ( ) const [inline]

6.45.2.791 template<class T> NSVec3<T> NSVec4< T >::wyz ( ) const [inline]

6.45.2.792 template<class T> NSVec4<T> NSVec4< T >::wyzw ( ) const [inline]

6.45.2.793 template<class T> NSVec4<T> NSVec4< T >::wyzx ( ) const [inline]

6.45.2.794 template<class T> NSVec4<T> NSVec4< T >::wyzy ( ) const [inline]

6.45.2.795 template<class T> NSVec4<T> NSVec4< T >::wyzz ( ) const [inline]

6.45.2.796 template<class T> NSVec2<T> NSVec4< T >::wz ( ) const [inline]

6.45.2.797 template<class T> NSVec3<T> NSVec4< T >::wzw ( ) const [inline]

6.45.2.798 template<class T> NSVec4<T> NSVec4< T >::wzww ( ) const [inline]

6.45.2.799 template<class T> NSVec4<T> NSVec4< T >::wzwx ( ) const [inline]

6.45.2.800 template<class T> NSVec4<T> NSVec4< T >::wzwy ( ) const [inline]

6.45.2.801 template<class T> NSVec4<T> NSVec4< T >::wzwz ( ) const [inline]

6.45.2.802 template<class T> NSVec3<T> NSVec4< T >::wzx ( ) const [inline]

6.45.2.803 template<class T> NSVec4<T> NSVec4< T >::wzxw ( ) const [inline]

6.45.2.804 template<class T> NSVec4<T> NSVec4< T >::wzxx ( ) const [inline]

6.45.2.805 template<class T> NSVec4<T> NSVec4< T >::wzxy ( ) const [inline]

6.45.2.806 template<class T> NSVec4<T> NSVec4< T >::wzxz ( ) const [inline]

6.45.2.807 template<class T> NSVec3<T> NSVec4< T >::wzy ( ) const [inline]

6.45.2.808 template<class T> NSVec4<T> NSVec4< T >::wzyw ( ) const [inline]

6.45.2.809 template<class T> NSVec4<T> NSVec4< T >::wzyx ( ) const [inline]

6.45.2.810 template<class T> NSVec4<T> NSVec4< T >::wzyy ( ) const [inline]

6.45.2.811 template<class T> NSVec4<T> NSVec4< T >::wzyz ( ) const [inline]

6.45.2.812 template<class T> NSVec3<T> NSVec4< T >::wzz ( ) const [inline]

6.45.2.813 template<class T> NSVec4<T> NSVec4< T >::wzzw ( ) const [inline]

6.45.2.814 template<class T> NSVec4<T> NSVec4< T >::wzzx ( ) const [inline]

6.45.2.815 template<class T> NSVec4<T> NSVec4< T >::wzzy ( ) const [inline]

6.45.2.816 template<class T> NSVec4<T> NSVec4< T >::wzzz ( ) const [inline]

6.45.2.817 template<class T> NSVec2<T> NSVec4< T >::xw ( ) const [inline]

6.45.2.818 template<class T> NSVec3<T> NSVec4< T >::xww ( ) const [inline]

6.45.2.819 template<class T> NSVec4<T> NSVec4< T >::xwww ( ) const [inline]

6.45.2.820 template<class T> NSVec4<T> NSVec4< T >::xwwx ( ) const [inline]

6.45.2.821 template<class T> NSVec4<T> NSVec4< T >::xwwy ( ) const [inline]

6.45.2.822 template<class T> NSVec4<T> NSVec4< T >::xwwz ( ) const [inline]

6.45.2.823 template<class T> NSVec3<T> NSVec4< T >::xwx ( ) const [inline]

6.45.2.824 template<class T> NSVec4<T> NSVec4< T >::xwxw ( ) const [inline]

6.45.2.825 template<class T> NSVec4<T> NSVec4< T >::xwxx ( ) const [inline]

6.45.2.826 template<class T> NSVec4<T> NSVec4< T >::xwxy ( ) const [inline]

6.45.2.827 template<class T> NSVec4<T> NSVec4< T >::xwxz ( ) const [inline]

6.45.2.828 template<class T> NSVec3<T> NSVec4< T >::xwy ( ) const [inline]

6.45.2.829 template<class T> NSVec4<T> NSVec4< T >::xwyw ( ) const [inline]

6.45.2.830 template<class T> NSVec4<T> NSVec4< T >::xwyx ( ) const [inline]

6.45.2.831 template<class T> NSVec4<T> NSVec4< T >::xwyy ( ) const [inline]

6.45.2.832 template<class T> NSVec4<T> NSVec4< T >::xwyz ( ) const [inline]

6.45.2.833 template<class T> NSVec3<T> NSVec4< T >::xwz ( ) const [inline]

6.45.2.834 template<class T> NSVec4<T> NSVec4< T >::xwzw ( ) const [inline]

6.45.2.835 template<class T> NSVec4<T> NSVec4< T >::xwzx ( ) const [inline]

6.45.2.836 template<class T> NSVec4<T> NSVec4< T >::xwzy ( ) const [inline]

6.45.2.837 template<class T> NSVec4<T> NSVec4< T >::xwzz ( ) const [inline]

6.45.2.838 template<class T> NSVec2<T> NSVec4< T >::xx ( ) const [inline]

6.45.2.839 template<class T> NSVec3<T> NSVec4< T >::xxw ( ) const [inline]

6.45.2.840 template<class T> NSVec4<T> NSVec4< T >::xxww ( ) const [inline]

6.45.2.841 template<class T> NSVec4<T> NSVec4< T >::xxwx ( ) const [inline]

6.45.2.842 template<class T> NSVec4<T> NSVec4< T >::xxwy ( ) const [inline]

6.45.2.843 template<class T> NSVec4<T> NSVec4< T >::xxwz ( ) const [inline]

6.45.2.844 template<class T> NSVec3<T> NSVec4< T >::xxx ( ) const [inline]

6.45.2.845 template<class T> NSVec4<T> NSVec4< T >::xxxw ( ) const [inline]

6.45.2.846 template<class T> NSVec4<T> NSVec4< T >::xxxx ( ) const [inline]

6.45.2.847 template<class T> NSVec4<T> NSVec4< T >::xxxy ( ) const [inline]

6.45.2.848 template<class T> NSVec4<T> NSVec4< T >::xxxz ( ) const [inline]

6.45.2.849 template<class T> NSVec3<T> NSVec4< T >::xxy ( ) const [inline]

6.45.2.850 template<class T> NSVec4<T> NSVec4< T >::xxyw ( ) const [inline]

6.45.2.851 template<class T> NSVec4<T> NSVec4< T >::xxyx ( ) const [inline]

6.45.2.852 template<class T> NSVec4<T> NSVec4< T >::xxyy ( ) const [inline]

6.45.2.853 template<class T> NSVec4<T> NSVec4< T >::xxyz ( ) const [inline]

6.45.2.854 template<class T> NSVec3<T> NSVec4< T >::xxz ( ) const [inline]

6.45.2.855 template<class T> NSVec4<T> NSVec4< T >::xxzw ( ) const [inline]

6.45.2.856 template<class T> NSVec4<T> NSVec4< T >::xxzx ( ) const [inline]

6.45.2.857 template<class T> NSVec4<T> NSVec4< T >::xxzy ( ) const [inline]

6.45.2.858 template<class T> NSVec4<T> NSVec4< T >::xxzz ( ) const [inline]

6.45.2.859 template<class T> NSVec2<T> NSVec4< T >::xy ( ) const [inline]

6.45.2.860 template<class T> NSVec3<T> NSVec4< T >::xyw ( ) const [inline]

6.45.2.861 template<class T> NSVec4<T> NSVec4< T >::xyww ( ) const [inline]

6.45.2.862 template<class T> NSVec4<T> NSVec4< T >::xywx ( ) const [inline]

6.45.2.863 template<class T> NSVec4<T> NSVec4< T >::xywy ( ) const [inline]

6.45.2.864 template<class T> NSVec4<T> NSVec4< T >::xywz ( ) const [inline]

6.45.2.865 template<class T> NSVec3<T> NSVec4< T >::xyx ( ) const [inline]

6.45.2.866 template<class T> NSVec4<T> NSVec4< T >::xyxw ( ) const [inline]

6.45.2.867 template<class T> NSVec4<T> NSVec4< T >::xyxx ( ) const [inline]

6.45.2.868 template<class T> NSVec4<T> NSVec4< T >::xyxy ( ) const [inline]

6.45.2.869 template<class T> NSVec4<T> NSVec4< T >::xyxz ( ) const [inline]

6.45.2.870 template<class T> NSVec3<T> NSVec4< T >::xyy ( ) const [inline]

6.45.2.871 template<class T> NSVec4<T> NSVec4< T >::xyyw ( ) const [inline]

6.45.2.872 template<class T> NSVec4<T> NSVec4< T >::xyyx ( ) const [inline]

6.45.2.873 template<class T> NSVec4<T> NSVec4< T >::xyyy ( ) const [inline]

6.45.2.874 template<class T> NSVec4<T> NSVec4< T >::xyyz ( ) const [inline]

6.45.2.875 template<class T> NSVec3<T> NSVec4< T >::xyz ( ) const [inline]

6.45.2.876 template<class T> NSVec2<T> NSVec4< T >::xz ( ) const [inline]

6.45.2.877 template<class T> NSVec3<T> NSVec4< T >::xzw ( ) const [inline]

6.45.2.878 template<class T> NSVec4<T> NSVec4< T >::xzww ( ) const [inline]

6.45.2.879 template<class T> NSVec4<T> NSVec4< T >::xzwx ( ) const [inline]

6.45.2.880 template<class T> NSVec4<T> NSVec4< T >::xzwy ( ) const [inline]

6.45.2.881 template<class T> NSVec4<T> NSVec4< T >::xzwz ( ) const [inline]

6.45.2.882 template<class T> NSVec3<T> NSVec4< T >::xzx ( ) const [inline]

6.45.2.883 template<class T> NSVec4<T> NSVec4< T >::xzxw ( ) const [inline]

6.45.2.884 template<class T> NSVec4<T> NSVec4< T >::xzxx ( ) const [inline]

6.45.2.885 template<class T> NSVec4<T> NSVec4< T >::xzxy ( ) const [inline]

6.45.2.886 template<class T> NSVec4<T> NSVec4< T >::xzxz ( ) const [inline]

6.45.2.887 template<class T> NSVec3<T> NSVec4< T >::xzy ( ) const [inline]

6.45.2.888 template<class T> NSVec4<T> NSVec4< T >::xzyw ( ) const [inline]

6.45.2.889 template<class T> NSVec4<T> NSVec4< T >::xzyx ( ) const [inline]

6.45.2.890 template<class T> NSVec4<T> NSVec4< T >::xzyy ( ) const [inline]

6.45.2.891 template<class T> NSVec4<T> NSVec4< T >::xzyz ( ) const [inline]

6.45.2.892 template<class T> NSVec3<T> NSVec4< T >::xzz ( ) const [inline]

6.45.2.893 template<class T> NSVec4<T> NSVec4< T >::xzzw ( ) const [inline]

6.45.2.894 template<class T> NSVec4<T> NSVec4< T >::xzzx ( ) const [inline]

6.45.2.895 template<class T> NSVec4<T> NSVec4< T >::xzzy ( ) const [inline]

6.45.2.896 template<class T> NSVec4<T> NSVec4< T >::xzzz ( ) const [inline]

6.45.2.897 template<class T> NSVec2<T> NSVec4< T >::yw ( ) const [inline]

6.45.2.898 template<class T> NSVec3<T> NSVec4< T >::yww ( ) const [inline]

6.45.2.899 template<class T> NSVec4<T> NSVec4< T >::ywww ( ) const [inline]

6.45.2.900 template<class T> NSVec4<T> NSVec4< T >::ywwx ( ) const [inline]

6.45.2.901 template<class T> NSVec4<T> NSVec4< T >::ywwy ( ) const [inline]

6.45.2.902 template<class T> NSVec4<T> NSVec4< T >::ywwz ( ) const [inline]

6.45.2.903 template<class T> NSVec3<T> NSVec4< T >::ywx ( ) const [inline]

6.45.2.904 template<class T> NSVec4<T> NSVec4< T >::ywxw ( ) const [inline]

6.45.2.905 template<class T> NSVec4<T> NSVec4< T >::ywxx ( ) const [inline]

6.45.2.906 template<class T> NSVec4<T> NSVec4< T >::ywxy ( ) const [inline]

6.45.2.907 template<class T> NSVec4<T> NSVec4< T >::ywxz ( ) const [inline]

6.45.2.908 template<class T> NSVec3<T> NSVec4< T >::ywy ( ) const [inline]

6.45.2.909 template<class T> NSVec4<T> NSVec4< T >::ywyw ( ) const [inline]

6.45.2.910 template<class T> NSVec4<T> NSVec4< T >::ywyx ( ) const [inline]

6.45.2.911 template<class T> NSVec4<T> NSVec4< T >::ywyy ( ) const [inline]

6.45.2.912 template<class T> NSVec4<T> NSVec4< T >::ywyz ( ) const [inline]

6.45.2.913 template<class T> NSVec3<T> NSVec4< T >::ywz ( ) const [inline]

6.45.2.914 template<class T> NSVec4<T> NSVec4< T >::ywzw ( ) const [inline]

6.45.2.915 template<class T> NSVec4<T> NSVec4< T >::ywzx ( ) const [inline]

6.45.2.916 template<class T> NSVec4<T> NSVec4< T >::ywzy ( ) const [inline]

6.45.2.917 template<class T> NSVec4<T> NSVec4< T >::ywzz ( ) const [inline]

6.45.2.918 template<class T> NSVec2<T> NSVec4< T >::yx ( ) const [inline]

6.45.2.919 template<class T> NSVec3<T> NSVec4< T >::yxw ( ) const [inline]

6.45.2.920 template<class T> NSVec4<T> NSVec4< T >::yxww ( ) const [inline]

6.45.2.921 template<class T> NSVec4<T> NSVec4< T >::yxwx ( ) const [inline]

6.45.2.922 template<class T> NSVec4<T> NSVec4< T >::yxwy ( ) const [inline]

6.45.2.923 template<class T> NSVec4<T> NSVec4< T >::yxwz ( ) const [inline]

6.45.2.924 template<class T> NSVec3<T> NSVec4< T >::yxx ( ) const [inline]

6.45.2.925 template<class T> NSVec4<T> NSVec4< T >::yxxw ( ) const [inline]

6.45.2.926 template<class T> NSVec4<T> NSVec4< T >::yxxx ( ) const [inline]

6.45.2.927 template<class T> NSVec4<T> NSVec4< T >::yxxy ( ) const [inline]

6.45.2.928 template<class T> NSVec4<T> NSVec4< T >::yxxz ( ) const [inline]

6.45.2.929 template<class T> NSVec3<T> NSVec4< T >::yxy ( ) const [inline]

6.45.2.930 template<class T> NSVec4<T> NSVec4< T >::yxyw ( ) const [inline]

6.45.2.931 template<class T> NSVec4<T> NSVec4< T >::yxyx ( ) const [inline]

6.45.2.932 template<class T> NSVec4<T> NSVec4< T >::yxyy ( ) const [inline]

6.45.2.933 template<class T> NSVec4<T> NSVec4< T >::yxyz ( ) const [inline]

6.45.2.934 template<class T> NSVec3<T> NSVec4< T >::yxz ( ) const [inline]

6.45.2.935 template<class T> NSVec4<T> NSVec4< T >::yxzw ( ) const [inline]

6.45.2.936 template<class T> NSVec4<T> NSVec4< T >::yxzx ( ) const [inline]

6.45.2.937 template<class T> NSVec4<T> NSVec4< T >::yxzy ( ) const [inline]

6.45.2.938 template<class T> NSVec4<T> NSVec4< T >::yxzz ( ) const [inline]

6.45.2.939 template<class T> NSVec2<T> NSVec4< T >::yy ( ) const [inline]

6.45.2.940 template<class T> NSVec3<T> NSVec4< T >::yyw ( ) const [inline]

6.45.2.941 template<class T> NSVec4<T> NSVec4< T >::yyww ( ) const [inline]

6.45.2.942 template<class T> NSVec4<T> NSVec4< T >::yywx ( ) const [inline]

6.45.2.943 template<class T> NSVec4<T> NSVec4< T >::yywy ( ) const [inline]

6.45.2.944 template<class T> NSVec4<T> NSVec4< T >::yywz ( ) const [inline]

6.45.2.945 template<class T> NSVec3<T> NSVec4< T >::yyx ( ) const [inline]

6.45.2.946 template<class T> NSVec4<T> NSVec4< T >::yyxw ( ) const [inline]

6.45.2.947 template<class T> NSVec4<T> NSVec4< T >::yyxx ( ) const [inline]

6.45.2.948 template<class T> NSVec4<T> NSVec4< T >::yyxy ( ) const [inline]

6.45.2.949 template<class T> NSVec4<T> NSVec4< T >::yyxz ( ) const [inline]

6.45.2.950 template<class T> NSVec3<T> NSVec4< T >::yyy ( ) const [inline]

6.45.2.951 template<class T> NSVec4<T> NSVec4< T >::yyyw ( ) const [inline]

6.45.2.952 template<class T> NSVec4<T> NSVec4< T >::yyyx ( ) const [inline]

6.45.2.953 template<class T> NSVec4<T> NSVec4< T >::yyyy ( ) const [inline]

6.45.2.954 template<class T> NSVec4<T> NSVec4< T >::yyyz ( ) const [inline]

6.45.2.955 template<class T> NSVec3<T> NSVec4< T >::yyz ( ) const [inline]

6.45.2.956 template<class T> NSVec4<T> NSVec4< T >::yyzw ( ) const [inline]

6.45.2.957 template<class T> NSVec4<T> NSVec4< T >::yyzx ( ) const [inline]

6.45.2.958 template<class T> NSVec4<T> NSVec4< T >::yyzy ( ) const [inline]

6.45.2.959 template<class T> NSVec4<T> NSVec4< T >::yyzz ( ) const [inline]

6.45.2.960 template<class T> NSVec2<T> NSVec4< T >::yz ( ) const [inline]

6.45.2.961 template<class T> NSVec3<T> NSVec4< T >::yzw ( ) const [inline]

6.45.2.962 template<class T> NSVec4<T> NSVec4< T >::yzww ( ) const [inline]

6.45.2.963 template<class T> NSVec4<T> NSVec4< T >::yzwx ( ) const [inline]

6.45.2.964 template<class T> NSVec4<T> NSVec4< T >::yzwy ( ) const [inline]

6.45.2.965 template<class T> NSVec4<T> NSVec4< T >::yzwz ( ) const [inline]

6.45.2.966 template<class T> NSVec3<T> NSVec4< T >::yzx ( ) const [inline]

6.45.2.967 template<class T> NSVec4<T> NSVec4< T >::yzxw ( ) const [inline]

6.45.2.968 template<class T> NSVec4<T> NSVec4< T >::yzxx ( ) const [inline]

6.45.2.969 template<class T> NSVec4<T> NSVec4< T >::yzxy ( ) const [inline]

6.45.2.970 template<class T> NSVec4<T> NSVec4< T >::yzxz ( ) const [inline]

6.45.2.971 template<class T> NSVec3<T> NSVec4< T >::yzy ( ) const [inline]

6.45.2.972 template<class T> NSVec4<T> NSVec4< T >::yzyw ( ) const [inline]

6.45.2.973 template<class T> NSVec4<T> NSVec4< T >::yzyx ( ) const [inline]

6.45.2.974 template<class T> NSVec4<T> NSVec4< T >::yzyy ( ) const [inline]

6.45.2.975 template<class T> NSVec4<T> NSVec4< T >::yzyz ( ) const [inline]

6.45.2.976 template<class T> NSVec3<T> NSVec4< T >::yzz ( ) const [inline]

6.45.2.977 template<class T> NSVec4<T> NSVec4< T >::yzzw ( ) const [inline]

6.45.2.978 template<class T> NSVec4<T> NSVec4< T >::yzzx ( ) const [inline]

6.45.2.979 template<class T> NSVec4<T> NSVec4< T >::yzzy ( ) const [inline]

6.45.2.980 template<class T> NSVec4<T> NSVec4< T >::yzzz ( ) const [inline]

6.45.2.981 template<class T> NSVec2<T> NSVec4< T >::zw ( ) const [inline]

6.45.2.982 template<class T> NSVec3<T> NSVec4< T >::zww ( ) const [inline]

6.45.2.983 template<class T> NSVec4<T> NSVec4< T >::zwww ( ) const [inline]

6.45.2.984 template<class T> NSVec4<T> NSVec4< T >::zwwx ( ) const [inline]

6.45.2.985 template<class T> NSVec4<T> NSVec4< T >::zwwy ( ) const [inline]

6.45.2.986 template<class T> NSVec4<T> NSVec4< T >::zwwz ( ) const [inline]

6.45.2.987 template<class T> NSVec3<T> NSVec4< T >::zwx ( ) const [inline]

6.45.2.988 template<class T> NSVec4<T> NSVec4< T >::zwxw ( ) const [inline]

6.45.2.989 template<class T> NSVec4<T> NSVec4< T >::zwxx ( ) const [inline]

6.45.2.990 template<class T> NSVec4<T> NSVec4< T >::zwxy ( ) const [inline]

6.45.2.991 template<class T> NSVec4<T> NSVec4< T >::zwxz ( ) const [inline]

6.45.2.992 template<class T> NSVec3<T> NSVec4< T >::zwy ( ) const [inline]

6.45.2.993 template<class T> NSVec4<T> NSVec4< T >::zwyw ( ) const [inline]

6.45.2.994 template<class T> NSVec4<T> NSVec4< T >::zwyx ( ) const [inline]

6.45.2.995 template<class T> NSVec4<T> NSVec4< T >::zwyy ( ) const [inline]

6.45.2.996 template<class T> NSVec4<T> NSVec4< T >::zwyz ( ) const [inline]

6.45.2.997 template<class T> NSVec3<T> NSVec4< T >::zwz ( ) const [inline]

6.45.2.998 template<class T> NSVec4<T> NSVec4< T >::zwzw ( ) const [inline]

6.45.2.999 template<class T> NSVec4<T> NSVec4< T >::zwzx ( ) const [inline]

6.45.2.1000 template<class T> NSVec4<T> NSVec4< T >::zwzy ( ) const [inline]

6.45.2.1001 template<class T> NSVec4<T> NSVec4< T >::zwzz ( ) const [inline]

6.45.2.1002 template<class T> NSVec2<T> NSVec4< T >::zx ( ) const [inline]

6.45.2.1003 template<class T> NSVec3<T> NSVec4< T >::zxw ( ) const [inline]

6.45.2.1004 template<class T> NSVec4<T> NSVec4< T >::zxww ( ) const [inline]

6.45.2.1005 template<class T> NSVec4<T> NSVec4< T >::zxwx ( ) const [inline]

6.45.2.1006 template<class T> NSVec4<T> NSVec4< T >::zxwy ( ) const [inline]

6.45.2.1007 template<class T> NSVec4<T> NSVec4< T >::zxwz ( ) const [inline]

6.45.2.1008 template<class T> NSVec3<T> NSVec4< T >::zxx ( ) const [inline]

6.45.2.1009 template<class T> NSVec4<T> NSVec4< T >::zxxw ( ) const [inline]

6.45.2.1010 template<class T> NSVec4<T> NSVec4< T >::zxxx ( ) const [inline]

6.45.2.1011 template<class T> NSVec4<T> NSVec4< T >::zxxy ( ) const [inline]

6.45.2.1012 template<class T> NSVec4<T> NSVec4< T >::zxxz ( ) const [inline]

6.45.2.1013 template<class T> NSVec3<T> NSVec4< T >::zxy ( ) const [inline]

6.45.2.1014 template<class T> NSVec4<T> NSVec4< T >::zxyw ( ) const [inline]

6.45.2.1015 template<class T> NSVec4<T> NSVec4< T >::zxyx ( ) const [inline]

6.45.2.1016 template<class T> NSVec4<T> NSVec4< T >::zxyy ( ) const [inline]

6.45.2.1017 template<class T> NSVec4<T> NSVec4< T >::zxyz ( ) const [inline]

6.45.2.1018 template<class T> NSVec3<T> NSVec4< T >::zxz ( ) const [inline]

6.45.2.1019 template<class T> NSVec4<T> NSVec4< T >::zxzw ( ) const [inline]

6.45.2.1020 template<class T> NSVec4<T> NSVec4< T >::zxzx ( ) const [inline]

6.45.2.1021 template<class T> NSVec4<T> NSVec4< T >::zxzy ( ) const [inline]

6.45.2.1022 template<class T> NSVec4<T> NSVec4< T >::zxzz ( ) const [inline]

6.45.2.1023 template<class T> NSVec2<T> NSVec4< T >::zy ( ) const [inline]

6.45.2.1024 template<class T> NSVec3<T> NSVec4< T >::zyw ( ) const [inline]

6.45.2.1025 template<class T> NSVec4<T> NSVec4< T >::zyww ( ) const [inline]

6.45.2.1026 template<class T> NSVec4<T> NSVec4< T >::zywx ( ) const [inline]

6.45.2.1027 template<class T> NSVec4<T> NSVec4< T >::zywy ( ) const [inline]

6.45.2.1028 template<class T> NSVec4<T> NSVec4< T >::zywz ( ) const [inline]

6.45.2.1029 template<class T> NSVec3<T> NSVec4< T >::zyx ( ) const [inline]

6.45.2.1030 template<class T> NSVec4<T> NSVec4< T >::zyxw ( ) const [inline]

6.45.2.1031 template<class T> NSVec4<T> NSVec4< T >::zyxx ( ) const [inline]

6.45.2.1032 template<class T> NSVec4<T> NSVec4< T >::zyxy ( ) const [inline]

6.45.2.1033 template<class T> NSVec4<T> NSVec4< T >::zyxz ( ) const [inline]

6.45.2.1034 template<class T> NSVec3<T> NSVec4< T >::zyy ( ) const [inline]

6.45.2.1035 template<class T> NSVec4<T> NSVec4< T >::zyyw ( ) const [inline]

6.45.2.1036 template<class T> NSVec4<T> NSVec4< T >::zyyx ( ) const [inline]

6.45.2.1037 template<class T> NSVec4<T> NSVec4< T >::zyyy ( ) const [inline]

6.45.2.1038 template<class T> NSVec4<T> NSVec4< T >::zyyz ( ) const [inline]

6.45.2.1039 template<class T> NSVec3<T> NSVec4< T >::zyz ( ) const [inline]

6.45.2.1040 template<class T> NSVec4<T> NSVec4< T >::zyzw ( ) const [inline]

6.45.2.1041 template<class T> NSVec4<T> NSVec4< T >::zyzx ( ) const [inline]

6.45.2.1042 template<class T> NSVec4<T> NSVec4< T >::zyzy ( ) const [inline]

6.45.2.1043 template<class T> NSVec4<T> NSVec4< T >::zyzz ( ) const [inline]

6.45.2.1044 template<class T> NSVec2<T> NSVec4< T >::zz ( ) const [inline]

6.45.2.1045 template<class T> NSVec3<T> NSVec4< T >::zzw ( ) const [inline]

6.45.2.1046 template<class T> NSVec4<T> NSVec4< T >::zzww ( ) const [inline]

6.45.2.1047 template<class T> NSVec4<T> NSVec4< T >::zzwx ( ) const [inline]

6.45.2.1048 template<class T> NSVec4<T> NSVec4< T >::zzwy ( ) const [inline]

6.45.2.1049 template<class T> NSVec4<T> NSVec4< T >::zzwz ( ) const [inline]

6.45.2.1050 template<class T> NSVec3<T> NSVec4< T >::zzx ( ) const [inline]

6.45.2.1051 template<class T> NSVec4<T> NSVec4< T >::zzxw ( ) const [inline]

6.45.2.1052 template<class T> NSVec4<T> NSVec4< T >::zzxx ( ) const [inline]

6.45.2.1053 template<class T> NSVec4<T> NSVec4< T >::zzxy ( ) const [inline]

6.45.2.1054 template<class T> NSVec4<T> NSVec4< T >::zzxz ( ) const [inline]

6.45.2.1055 template<class T> NSVec3<T> NSVec4< T >::zzy ( ) const [inline]

6.45.2.1056 template<class T> NSVec4<T> NSVec4< T >::zzyw ( ) const [inline]

6.45.2.1057 template<class T> NSVec4<T> NSVec4< T >::zzyx ( ) const [inline]

6.45.2.1058 template<class T> NSVec4<T> NSVec4< T >::zzyy ( ) const [inline]

6.45.2.1059 template<class T> NSVec4<T> NSVec4< T >::zzyz ( ) const [inline]

6.45.2.1060 template<class T> NSVec3<T> NSVec4< T >::zzz ( ) const [inline]

6.45.2.1061 template<class T> NSVec4<T> NSVec4< T >::zzzw ( ) const [inline]

6.45.2.1062 template<class T> NSVec4<T> NSVec4< T >::zzzx ( ) const [inline]

6.45.2.1063 template<class T> NSVec4<T> NSVec4< T >::zzzy ( ) const [inline]

6.45.2.1064 template<class T> NSVec4<T> NSVec4< T >::zzzz ( ) const [inline]

6.45.3 Member Data Documentation

6.45.3.1 union { ... }

6.45.3.2 template<class T> T NSVec4< T >::a

6.45.3.3 template<class T> T NSVec4< T >::b

6.45.3.4 template<class T> T NSVec4< T >::data[4]

6.45.3.5 template<class T> T NSVec4< T >::g

6.45.3.6 template<class T> T NSVec4< T >::p

6.45.3.7 template<class T> T NSVec4< T >::q

6.45.3.8 template<class T> T NSVec4< T >::r

6.45.3.9 template<class T> T NSVec4< T >::s

6.45.3.10 template<class T> T NSVec4< T >::t

6.45.3.11 template<class T> T NSVec4< T >::w

6.45.3.12 template<class T> T NSVec4< T >::x

6.45.3.13 template<class T> T NSVec4< T >::y

6.45.3.14 template<class T> T NSVec4< T >::z

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

- /home/dprandle/Documents/code/ctrlmod/include/nsvec2.h
- /home/dprandle/Documents/code/ctrlmod/include/nsvec4.h

6.46 edpid_controller< T >::output_range Struct Reference

`#include <edpid_controller.h>`

Public Member Functions

- output_range (const T &min_=0, const T &max_=0)

Public Attributes

- T min
- T max

6.46.1 Constructor & Destructor Documentation

6.46.1.1 template<class T> edpid_controller< T >::output_range::output_range (const T & *min_* = 0, const T & *max_* = 0) [inline]

6.46.2 Member Data Documentation

6.46.2.1 template<class T> T edpid_controller< T >::output_range::max

6.46.2.2 template<class T> T edpid_controller< T >::output_range::min

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

- /home/dprandle/Documents/code/ctrlmod/include/edpid_controller.h

6.47 edpl_system::pl_gpio Struct Reference

`#include <edplsystem.h>`

Public Member Functions

- pl_gpio (uint32_t mraa_pin, double calibrate_offset=0.0, const vec3 &poffset=vec3(), const quat &orient_=quat())
- ~pl_gpio ()

Static Public Member Functions

- static void isr (void *)

Public Attributes

- mraa::Gpio * pin
- edtimer * timer
- vec3 pos
- quat orient
- double cal_offset
- bool meas_ready

- double sum_dist
- uint32_t mraa_pin_num
- uint32_t meas_count

6.47.1 Constructor & Destructor Documentation

6.47.1.1 edpl_system::pl_gpio::pl_gpio (uint32_t *mraa_pin*, double *calibrate_offset* = 0.0, const vec3 & *poffset* = vec3(), const quat & *orient_* = quat())

6.47.1.2 edpl_system::pl_gpio::~pl_gpio ( )

6.47.2 Member Function Documentation

6.47.2.1 void edpl_system::pl_gpio::isr (void * *pl*) [static]

6.47.3 Member Data Documentation

6.47.3.1 double edpl_system::pl_gpio::cal_offset

6.47.3.2 uint32_t edpl_system::pl_gpio::meas_count

6.47.3.3 bool edpl_system::pl_gpio::meas_ready

6.47.3.4 uint32_t edpl_system::pl_gpio::mraa_pin_num

6.47.3.5 quat edpl_system::pl_gpio::orient

6.47.3.6 mraa::Gpio* edpl_system::pl_gpio::pin

6.47.3.7 vec3 edpl_system::pl_gpio::pos

6.47.3.8 double edpl_system::pl_gpio::sum_dist

6.47.3.9 edtimer* edpl_system::pl_gpio::timer

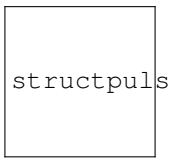
The documentation for this struct was generated from the following files:

- /home/dprandle/Documents/code/ctrlmod/include/edplsystem.h
- /home/dprandle/Documents/code/ctrlmod/src/edplsystem.cpp

6.48 pulsed_light_message Struct Reference

`#include <edmessage.h>`

Inheritance diagram for pulsed_light_message:



Public Member Functions

- uint32_t size ()
- std::string type ()

Static Public Member Functions

- static std::string Type ()

Public Attributes

- union {
  struct {
  double distance1
  double distance2
  uint32_t mraa_pin1
  uint32_t mraa_pin2
  double pos1 [3]
  double pos2 [3]
  double orientation1 [4]
  double orientation2 [4]
  }
  uint8_t data [136]
  };

6.48.1 Member Function Documentation

6.48.1.1 uint32_t pulsed_light_message::size ( ) [inline]

6.48.1.2 std::string pulsed_light_message::type ( ) [inline], [virtual]

Implements edmessage.

6.48.1.3 static std::string pulsed_light_message::Type ( ) [inline], [static]

6.48.2 Member Data Documentation

6.48.2.1 union { ... }

6.48.2.2 uint8_t pulsed_light_message::data[136]

6.48.2.3 double pulsed_light_message::distance1

6.48.2.4 double pulsed_light_message::distance2

6.48.2.5 uint32_t pulsed_light_message::mraa_pin1

6.48.2.6 uint32_t pulsed_light_message::mraa_pin2

6.48.2.7 double pulsed_light_message::orientation1[4]

6.48.2.8 double pulsed_light_message::orientation2[4]

6.48.2.9 double pulsed_light_message::pos1[3]

6.48.2.10 double pulsed_light_message::pos2[3]

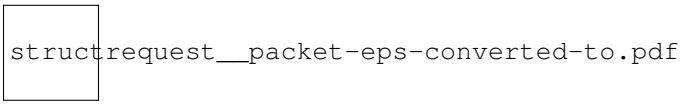
The documentation for this struct was generated from the following file:

- /home/dprandle/Documents/code/ctrlmod/include/edmessage.h

6.49 request_packet Struct Reference

`#include <edrplidar_packets.h>`

Inheritance diagram for request_packet:



Public Member Functions

- request_packet (uint8_t MSB, uint8_t LSB)
- virtual ~request_packet ()

Public Attributes

- union {
  struct {
  uint8_t msB
  uint8_t lsB
  }
  uint8_t data [2]
  };

6.49.1 Constructor & Destructor Documentation

6.49.1.1 request_packet::request_packet (uint8_t *MSB,* uint8_t *LSB*) [inline]

6.49.1.2 virtual request_packet::~request_packet ( ) [inline], [virtual]

6.49.2 Member Data Documentation

6.49.2.1 union { ... }

6.49.2.2 uint8_t request_packet::data[2]

6.49.2.3 uint8_t request_packet::lsB

6.49.2.4 uint8_t request_packet::msB

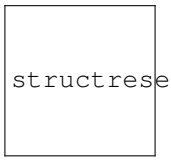
The documentation for this struct was generated from the following file:

- /home/dprandle/Documents/code/ctrlmod/include/edrplidar_packets.h

6.50 reset_request Struct Reference

`#include <edrplidar_packets.h>`

Inheritance diagram for reset_request:



Public Member Functions

- reset_request ()

Additional Inherited Members

6.50.1 Constructor & Destructor Documentation

6.50.1.1 reset_request::reset_request ( ) [inline]

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

- /home/dprandle/Documents/code/ctrlmod/include/edrplidar_packets.h

6.51 rplidar_error_message Struct Reference

`#include <edmessage.h>`

Inheritance diagram for rplidar_error_message:

Public Member Functions

- rplidar_error_message ()
- virtual std::string type ()

Static Public Member Functions

- static std::string Type ()

Public Attributes

- uint8_t message [100]

6.51.1 Constructor & Destructor Documentation

6.51.1.1 rplidar_error_message::rplidar_error_message ( )

6.51.2 Member Function Documentation

6.51.2.1 virtual std::string rplidar_error_message::type ( ) [inline], [virtual]

Implements edmessage.

6.51.2.2 static std::string rplidar_error_message::Type ( ) [inline], [static]

6.51.3 Member Data Documentation

6.51.3.1 uint8_t rplidar_error_message::message[100]

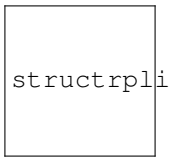
The documentation for this struct was generated from the following files:

- /home/dprandle/Documents/code/ctrlmod/include/edmessage.h
- /home/dprandle/Documents/code/ctrlmod/src/edmessage.cpp

6.52 rplidar_firmware_message Struct Reference

`#include <edmessage.h>`

Inheritance diagram for rplidar_firmware_message:



Public Member Functions

- virtual std::string type ()

Static Public Member Functions

- static std::string Type ()

Public Attributes

- firmware_data_packet device_firmware

6.52.1 Member Function Documentation

6.52.1.1 virtual std::string rplidar_firmware_message::type ( ) [inline], [virtual]

Implements edmessage.

6.52.1.2 static std::string rplidar_firmware_message::Type ( ) [inline], [static]

6.52.2 Member Data Documentation

6.52.2.1 firmware_data_packet rplidar_firmware_message::device_firmware

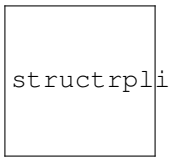
The documentation for this struct was generated from the following file:

- /home/dprandle/Documents/code/ctrlmod/include/edmessage.h

6.53 rplidar_health_message Struct Reference

`#include <edmessage.h>`

Inheritance diagram for rplidar_health_message:



Public Member Functions

- virtual std::string type ()

Static Public Member Functions

- static std::string Type ()

Public Attributes

- health_data_packet device_health

6.53.1 Member Function Documentation

6.53.1.1 virtual std::string rplidar_health_message::type ( ) [inline], [virtual]

Implements edmessage.

6.53.1.2 static std::string rplidar_health_message::Type ( ) [inline], [static]

6.53.2 Member Data Documentation

6.53.2.1 health_data_packet rplidar_health_message::device_health

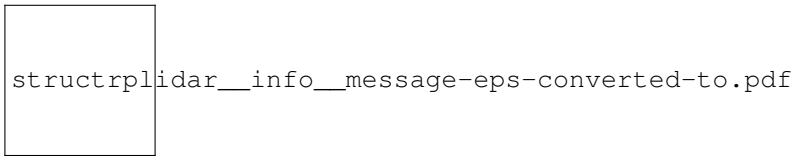
The documentation for this struct was generated from the following file:

- /home/dprandle/Documents/code/ctrlmod/include/edmessage.h

6.54 rplidar_info_message Struct Reference

`#include <edmessage.h>`

Inheritance diagram for rplidar_info_message:



Public Member Functions

- virtual std::string type ()

Static Public Member Functions

- static std::string Type ()

Public Attributes

- info_data_packet device_info

6.54.1 Member Function Documentation

6.54.1.1 virtual std::string rplidar_info_message::type ( ) [inline], [virtual]

Implements edmessage.

6.54.1.2 static std::string rplidar_info_message::Type ( ) [inline], [static]

6.54.2 Member Data Documentation

6.54.2.1 info_data_packet rplidar_info_message::device_info

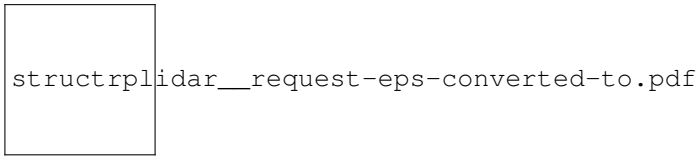
The documentation for this struct was generated from the following file:

- /home/dprandle/Documents/code/ctrlmod/include/edmessage.h

6.55 rplidar_request Struct Reference

`#include <edmessage.h>`

Inheritance diagram for rplidar_request:



Public Types

- enum req_type {
  HealthReq, InfoReq, StartScan, ForceScan,
  StopScan, Reset }

Public Member Functions

- virtual std::string type ()

Static Public Member Functions

- static std::string Type ()

Public Attributes

- req_type r_type

6.55.1 Member Enumeration Documentation

6.55.1.1 enum rplidar_request::req_type

Enumerator

HealthReq

InfoReq

StartScan

ForceScan

StopScan

Reset

6.55.2 Member Function Documentation

6.55.2.1 virtual std::string rplidar_request::type ( ) [inline], [virtual]

Implements edmessage.

6.55.2.2 static std::string rplidar_request::Type ( ) [inline], [static]

6.55.3 Member Data Documentation

6.55.3.1 req_type rplidar_request::r_type

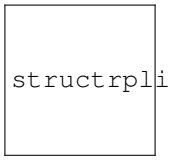
The documentation for this struct was generated from the following file:

- /home/dprandle/Documents/code/ctrlmod/include/edmessage.h

6.56 rplidar_scan_message Struct Reference

`#include <edmessage.h>`

Inheritance diagram for rplidar_scan_message:



Public Member Functions

- virtual std::string type ()

Static Public Member Functions

- static std::string Type ()

Public Attributes

- complete_scan_data_packet scan_data
- uint32_t millis_timestamp

6.56.1 Member Function Documentation

6.56.1.1 virtual std::string rplidar_scan_message::type ( ) [inline], [virtual]

Implements edmessage.

6.56.1.2 static std::string rplidar_scan_message::Type ( ) [inline], [static]

6.56.2 Member Data Documentation

6.56.2.1 uint32_t rplidar_scan_message::millis_timestamp

6.56.2.2 complete_scan_data_packet rplidar_scan_message::scan_data

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

- /home/dprandle/Documents/code/ctrlmod/include/edmessage.h

6.57 scan_data_packet Struct Reference

`#include <edrplidar_packets.h>`

Inheritance diagram for scan_data_packet:

Public Member Functions

- scan_data_packet ()
- virtual std::string toString ()
- std::string type ()
- virtual uint32_t size ()
- virtual uint8_t & operator[] (uint32_t index)
- virtual uint8_t * dataptr ()

Static Public Member Functions

- static std::string Type ()
- static uint32_t Size ()

Public Attributes

- union {
  struct {
  uint8_t qual_s_sn
  uint8_t angle6to0_C
  uint8_t angle14to7
  uint8_t distance7to0
  uint8_t distance15to8
  }
  uint8_t data [5]
  };

6.57.1 Constructor & Destructor Documentation

6.57.1.1 scan_data_packet::scan_data_packet ( )

6.57.2 Member Function Documentation

6.57.2.1 virtual uint8_t* scan_data_packet::dataptr ( ) [inline], [virtual]

Implements data_packet.

6.57.2.2 virtual uint8_t& scan_data_packet::operator[] (uint32_t *index*) [inline], [virtual]

Implements data_packet.

6.57.2.3 virtual uint32_t scan_data_packet::size ( ) [inline], [virtual]

Implements data_packet.

6.57.2.4 static uint32_t scan_data_packet::Size ( ) [inline], [static]

6.57.2.5 std::string scan_data_packet::toString ( ) [virtual]

Implements data_packet.

6.57.2.6 std::string scan_data_packet::type ( ) [inline], [virtual]

Implements data_packet.

6.57.2.7 static std::string scan_data_packet::Type ( ) [inline], [static]

6.57.3 Member Data Documentation

6.57.3.1 union { ... }

6.57.3.2 uint8_t scan_data_packet::angle14to7

6.57.3.3 uint8_t scan_data_packet::angle6to0_C

6.57.3.4 uint8_t scan_data_packet::data[5]

6.57.3.5 uint8_t scan_data_packet::distance15to8

6.57.3.6 uint8_t scan_data_packet::distance7to0

6.57.3.7 uint8_t scan_data_packet::qual_s_sn

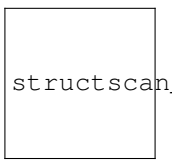
The documentation for this struct was generated from the following files:

- /home/dprandle/Documents/code/ctrlmod/include/edrplidar_packets.h
- /home/dprandle/Documents/code/ctrlmod/src/edrplidar_packets.cpp

6.58 scan_descriptor Struct Reference

`#include <edrplidar_packets.h>`

Inheritance diagram for scan_descriptor:



Public Member Functions

- scan_descriptor ()
- virtual std::string type ()

Additional Inherited Members

6.58.1 Constructor & Destructor Documentation

6.58.1.1 scan_descriptor::scan_descriptor ( ) [inline]

6.58.2 Member Function Documentation

6.58.2.1 virtual std::string scan_descriptor::type ( ) [inline], [virtual]

Implements descriptor_packet.

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

- /home/dprandle/Documents/code/ctrlmod/include/edrplidar_packets.h

6.59 start_scan_request Struct Reference

`#include <edrplidar_packets.h>`

Inheritance diagram for start_scan_request:

Public Member Functions

- start_scan_request ()

Additional Inherited Members

6.59.1 Constructor & Destructor Documentation

6.59.1.1 start_scan_request::start_scan_request ( ) [inline]

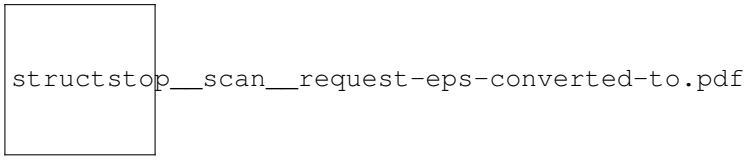
The documentation for this struct was generated from the following file:

- /home/dprandle/Documents/code/ctrlmod/include/edrplidar_packets.h

6.60 stop_scan_request Struct Reference

`#include <edrplidar_packets.h>`

Inheritance diagram for stop_scan_request:



Public Member Functions

- stop_scan_request ()

Additional Inherited Members

6.60.1 Constructor & Destructor Documentation

6.60.1.1 stop_scan_request::stop_scan_request ( ) [inline]

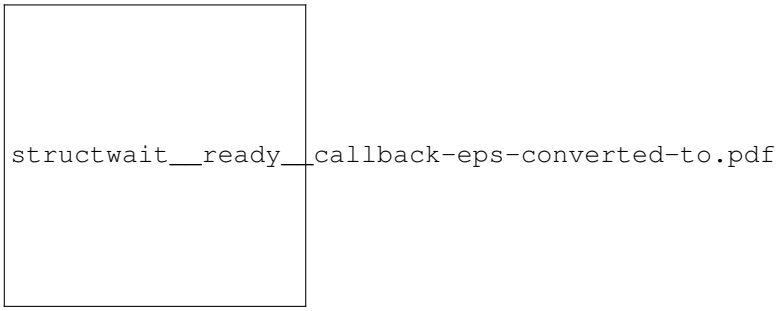
The documentation for this struct was generated from the following file:

- /home/dprandle/Documents/code/ctrlmod/include/edrplidar_packets.h

6.61 wait_ready_callback Struct Reference

`#include <edcallback.h>`

Inheritance diagram for wait_ready_callback:



Public Member Functions

- virtual void exec ()

Additional Inherited Members

6.61.1 Member Function Documentation

6.61.1.1 void wait_ready_callback::exec ( ) [virtual]

Implements edcallback.

Reimplemented in command_wait_callback.

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

- /home/dprandle/Documents/code/ctrlmod/include/edcallback.h
- /home/dprandle/Documents/code/ctrlmod/src/edcallback.cpp

6.62 edthreaded_fd::WriteVal Struct Reference

`#include <edthreaded_fd.h>`

Public Member Functions

- WriteVal (uint8_t byte_=0x00, int32_t response_size_=0)

Public Attributes

- uint8_t byte
- uint32_t response_size

6.62.1 Constructor & Destructor Documentation

6.62.1.1 edthreaded_fd::WriteVal::WriteVal (uint8_t *byte_* = `0x00`, int32_t *response_size_* = `0`) `[inline]`

6.62.2 Member Data Documentation

6.62.2.1 uint8_t edthreaded_fd::WriteVal::byte

6.62.2.2 uint32_t edthreaded_fd::WriteVal::response_size

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

- /home/dprandle/Documents/code/ctrlmod/include/edthreaded_fd.h

Chapter 7

File Documentation

7.1 /home/dprandle/Documents/code/ctrlmod/include/edcallback.h File Reference

Classes

- struct edcallback
- struct edtimer_callback
- struct wait_ready_callback

7.2 /home/dprandle/Documents/code/ctrlmod/include/edcomm_system.h File Reference

```
#include <edglobal.h>
#include <edsystem.h>
#include <vector>
```

Classes

- struct Command
- class edcomm_system

Macros

- #define SOCKET_BUFF_SIZE
- #define COMMAND_BYTE_SIZE 72

7.2.1 Macro Definition Documentation

7.2.1.1 #define COMMAND_BYTE_SIZE 72

7.2.1.2 #define SOCKET_BUFF_SIZE

7.3 /home/dprandle/Documents/code/ctrlmod/include/edglobal.h File Reference

```
#include <stdint.h>
```

Macros

- #define CONSOLE_OUT

7.3.1 Macro Definition Documentation

7.3.1.1 #define CONSOLE_OUT

7.4 /home/dprandle/Documents/code/ctrlmod/include/edi2c.h File Reference

Declaration file for edi2c class.

```
#include <edthreaded_fd.h>
#include <string>
```

Classes

- class edi2c
  edi2c

Macros

- #define DEFAULT_READ_DELAY 20
- #define DEFAULT_WRITE_DELAY 20

7.4.1 Detailed Description

Declaration file for edi2c class.

Author

Daniel <dprandle-CZ-17>

Date

Thu Aug 27 17:39:59 2015

7.4.2 Macro Definition Documentation

7.4.2.1 #define DEFAULT_READ_DELAY 20

7.4.2.2 #define DEFAULT_WRITE_DELAY 20

7.5 /home/dprandle/Documents/code/ctrlmod/include/edimu_system.h File Reference

```
#include <edsystem.h>
#include <nsmath.h>
```

Classes

- class edimu_system

Macros

- #define LSM9DS0_XM_ADDR 0x1D
- #define LSM9DS0_G_ADDR 0x6B
- #define WHO_AM_I_G 0x0F
- #define CTRL_REG1_G 0x20
- #define CTRL_REG2_G 0x21
- #define CTRL_REG3_G 0x22
- #define CTRL_REG4_G 0x23
- #define CTRL_REG5_G 0x24
- #define REFERENCE_G 0x25
- #define STATUS_REG_G 0x27
- #define OUT_X_L_G 0x28
- #define OUT_X_H_G 0x29
- #define OUT_Y_L_G 0x2A
- #define OUT_Y_H_G 0x2B
- #define OUT_Z_L_G 0x2C
- #define OUT_Z_H_G 0x2D
- #define FIFO_CTRL_REG_G 0x2E
- #define FIFO_SRC_REG_G 0x2F
- #define INT1_CFG_G 0x30
- #define INT1_SRC_G 0x31
- #define INT1_THS_XH_G 0x32
- #define INT1_THS_XL_G 0x33
- #define INT1_THS_YH_G 0x34
- #define INT1_THS_YL_G 0x35
- #define INT1_THS_ZH_G 0x36
- #define INT1_THS_ZL_G 0x37
- #define INT1_DURATION_G 0x38
- #define OUT_TEMP_L_XM 0x05
- #define OUT_TEMP_H_XM 0x06
- #define STATUS_REG_M 0x07
- #define OUT_X_L_M 0x08
- #define OUT_X_H_M 0x09
- #define OUT_Y_L_M 0x0A
- #define OUT_Y_H_M 0x0B
- #define OUT_Z_L_M 0x0C
- #define OUT_Z_H_M 0x0D
- #define WHO_AM_I_XM 0x0F
- #define INT_CTRL_REG_M 0x12
- #define INT_SRC_REG_M 0x13
- #define INT_THS_L_M 0x14
- #define INT_THS_H_M 0x15
- #define OFFSET_X_L_M 0x16
- #define OFFSET_X_H_M 0x17
- #define OFFSET_Y_L_M 0x18
- #define OFFSET_Y_H_M 0x19
- #define OFFSET_Z_L_M 0x1A
- #define OFFSET_Z_H_M 0x1B
- #define REFERENCE_X 0x1C
- #define REFERENCE_Y 0x1D
- #define REFERENCE_Z 0x1E
- #define CTRL_REG0_XM 0x1F
- #define CTRL_REG1_XM 0x20
- #define CTRL_REG2_XM 0x21

- #define CTRL_REG3_XM 0x22
- #define CTRL_REG4_XM 0x23
- #define CTRL_REG5_XM 0x24
- #define CTRL_REG6_XM 0x25
- #define CTRL_REG7_XM 0x26
- #define STATUS_REG_A 0x27
- #define OUT_X_L_A 0x28
- #define OUT_X_H_A 0x29
- #define OUT_Y_L_A 0x2A
- #define OUT_Y_H_A 0x2B
- #define OUT_Z_L_A 0x2C
- #define OUT_Z_H_A 0x2D
- #define FIFO_CTRL_REG 0x2E
- #define FIFO_SRC_REG 0x2F
- #define INT_GEN_1_REG 0x30
- #define INT_GEN_1_SRC 0x31
- #define INT_GEN_1_THS 0x32
- #define INT_GEN_1_DURATION 0x33
- #define INT_GEN_2_REG 0x34
- #define INT_GEN_2_SRC 0x35
- #define INT_GEN_2_THS 0x36
- #define INT_GEN_2_DURATION 0x37
- #define CLICK_CFG 0x38
- #define CLICK_SRC 0x39
- #define CLICK_THS 0x3A
- #define TIME_LIMIT 0x3B
- #define TIME_LATENCY 0x3C
- #define TIME_WINDOW 0x3D
- #define ACT_THS 0x3E
- #define ACT_DUR 0x3F

7.5.1 Macro Definition Documentation

7.5.1.1 #define ACT_DUR 0x3F

7.5.1.2 #define ACT_THS 0x3E

7.5.1.3 #define CLICK_CFG 0x38

7.5.1.4 #define CLICK_SRC 0x39

7.5.1.5 #define CLICK_THS 0x3A

7.5.1.6 #define CTRL_REG0_XM 0x1F

7.5.1.7 #define CTRL_REG1_G 0x20

7.5.1.8 #define CTRL_REG1_XM 0x20

7.5.1.9 #define CTRL_REG2_G 0x21

7.5.1.10 #define CTRL_REG2_XM 0x21

7.5.1.11 #define CTRL_REG3_G 0x22

7.5.1.12 #define CTRL_REG3_XM 0x22

7.5.1.13 #define CTRL_REG4_G 0x23

7.5.1.14 #define CTRL_REG4_XM 0x23

7.5.1.15 #define CTRL_REG5_G 0x24

7.5.1.16 #define CTRL_REG5_XM 0x24

7.5.1.17 #define CTRL_REG6_XM 0x25

7.5.1.18 #define CTRL_REG7_XM 0x26

7.5.1.19 #define FIFO_CTRL_REG 0x2E

7.5.1.20 #define FIFO_CTRL_REG_G 0x2E

7.5.1.21 #define FIFO_SRC_REG 0x2F

7.5.1.22 #define FIFO_SRC_REG_G 0x2F

7.5.1.23 #define INT1_CFG_G 0x30

7.5.1.24 #define INT1_DURATION_G 0x38

7.5.1.25 #define INT1_SRC_G 0x31

7.5.1.26 #define INT1_THS_XH_G 0x32

7.5.1.27 #define INT1_THS_XL_G 0x33

7.5.1.28 #define INT1_THS_YH_G 0x34

7.5.1.29 #define INT1_THS_YL_G 0x35

7.5.1.30 #define INT1_THS_ZH_G 0x36

7.5.1.31 #define INT1_THS_ZL_G 0x37

7.5.1.32 #define INT_CTRL_REG_M 0x12

7.5.1.33 #define INT_GEN_1_DURATION 0x33

7.5.1.34 #define INT_GEN_1_REG 0x30

7.5.1.35 #define INT_GEN_1_SRC 0x31

7.5.1.36 #define INT_GEN_1_THS 0x32

7.5.1.37 #define INT_GEN_2_DURATION 0x37

7.5.1.38 #define INT_GEN_2_REG 0x34

7.5.1.39 #define INT_GEN_2_SRC 0x35

7.5.1.40 #define INT_GEN_2_THS 0x36

7.5.1.41 #define INT_SRC_REG_M 0x13

7.5.1.42 #define INT_THS_H_M 0x15

7.5.1.43 #define INT_THS_L_M 0x14

7.5.1.44 #define LSM9DS0_G_ADDR 0x6B

7.5.1.45 #define LSM9DS0_XM_ADDR 0x1D

7.5.1.46 #define OFFSET_X_H_M 0x17

7.5.1.47 #define OFFSET_X_L_M 0x16

7.5.1.48 #define OFFSET_Y_H_M 0x19

7.5.1.49 #define OFFSET_Y_L_M 0x18

7.5.1.50 #define OFFSET_Z_H_M 0x1B

7.5.1.51 #define OFFSET_Z_L_M 0x1A

7.5.1.52 #define OUT_TEMP_H_XM 0x06

7.5.1.53 #define OUT_TEMP_L_XM 0x05

Accel/Magneto Registers

7.5.1.54 #define OUT_X_H_A 0x29

7.5.1.55 #define OUT_X_H_G 0x29

7.5.1.56 #define OUT_X_H_M 0x09

7.5.1.57 #define OUT_X_L_A 0x28

7.5.1.58 #define OUT_X_L_G 0x28

7.5.1.59 #define OUT_X_L_M 0x08

7.5.1.60 #define OUT_Y_H_A 0x2B

7.5.1.61 #define OUT_Y_H_G 0x2B

7.5.1.62 #define OUT_Y_H_M 0x0B

7.5.1.63 #define OUT_Y_L_A 0x2A

7.5.1.64 #define OUT_Y_L_G 0x2A

7.5.1.65 #define OUT_Y_L_M 0x0A

7.5.1.66 #define OUT_Z_H_A 0x2D

7.5.1.67 #define OUT_Z_H_G 0x2D

7.5.1.68 #define OUT_Z_H_M 0x0D

7.5.1.69 #define OUT_Z_L_A 0x2C

7.5.1.70 #define OUT_Z_L_G 0x2C

7.5.1.71 #define OUT_Z_L_M 0x0C

7.5.1.72 #define REFERENCE_G 0x25

7.5.1.73 #define REFERENCE_X 0x1C

7.5.1.74 #define REFERENCE_Y 0x1D

7.5.1.75 #define REFERENCE_Z 0x1E

7.5.1.76 #define STATUS_REG_A 0x27

7.5.1.77 #define STATUS_REG_G 0x27

7.5.1.78 #define STATUS_REG_M 0x07

7.5.1.79 #define TIME_LATENCY 0x3C

7.5.1.80 #define TIME_LIMIT 0x3B

7.5.1.81 #define TIME_WINDOW 0x3D

7.5.1.82 #define WHO_AM_I_G 0x0F

Gyro Registers

7.5.1.83 #define WHO_AM_I_XM 0x0F

7.6 /home/dprandle/Documents/code/ctrlmod/include/edlogging_system.h File Reference

```
#include <edsystem.h>
#include <edglobal.h>
#include <edrplidar_packets.h>
```

Classes

- class edlogging_system

7.7 /home/dprandle/Documents/code/ctrlmod/include/edmctrl.h File Reference

Header file for master controller.

```
#include <string>
#include <map>
```

Classes

- class edmctrl

Macros

- #define edm edmctrl::inst()

Typedefs

- typedef std::map< std::string, edsystem * > sysmap

7.7.1 Detailed Description

Header file for master controller.

Author

Daniel <dprandle-CZ-17>

Date

Fri Jul 10 09:20:01 2015

7.7.2 Macro Definition Documentation

7.7.2.1 #define edm edmctrl::inst()

7.7.3 Typedef Documentation

7.7.3.1 typedef std::map<std::string,edsystem*> sysmap

7.8 /home/dprandle/Documents/code/ctrlmod/include/edmessage.h File Reference

```
#include <edglobal.h>
#include <nsmath.h>
#include <edrplidar_packets.h>
```

Classes

- struct edmessage
- struct pulsed_light_message
- struct nav_message
- struct rplidar_request
- struct nav_system_request
- struct rplidar_scan_message

- struct rplidar_error_message
- struct rplidar_info_message
- struct rplidar_health_message
- struct rplidar_firmware_message

7.9 /home/dprandle/Documents/code/ctrlmod/include/edmessage_dispatch.h File Reference

```
#include <edutility.h>
#include <edglobal.h>
#include <nsmath.h>
#include <map>
#include <set>
#include <deque>
#include <edsystem.h>
```

Classes

- class edmessage_dispatch

  Class edmessage_dispatch.

7.10 /home/dprandle/Documents/code/ctrlmod/include/ednavsystem.h File Reference

Navigation system header file.

```
#include <edsystem.h>
#include <nsmath.h>
#include <edpid_controller.h>
#include <edcallback.h>
```

Classes

- class ednav_system
- struct instruction_callback

Namespaces

- mraa

Macros

- #define ARDUINO_ADDRESS 0x04
- #define ALT_DIF_MIN -100
- #define ALT_DIF_MAX 100
- #define LIDAR_DIST_DIFF_MIN -5000
- #define LIDAR_DIST_DIFF_MAX 5000
- #define YAW_ANGLE_DIFF_MIN -45
- #define YAW_ANGLE_DIFF_MAX 45
- #define THROTTLE_MIN -500

- #define THROTTLE_MAX 500
- #define PITCH_MIN -500
- #define PITCH_MAX 500
- #define ROLL_MIN -500
- #define ROLL_MAX 500
- #define YAW_MIN -500
- #define YAW_MAX 500
- #define G_CONST 5000000

7.10.1 Detailed Description

Navigation system header file.

Author

Daniel <dprandle-CZ-17>

Date

Fri Jul 10 10:34:08 2015

7.10.2 Macro Definition Documentation

7.10.2.1 #define ALT_DIF_MAX 100

cm

7.10.2.2 #define ALT_DIF_MIN -100

cm

7.10.2.3 #define ARDUINO_ADDRESS 0x04

7.10.2.4 #define G_CONST 5000000

7.10.2.5 #define LIDAR_DIST_DIFF_MAX 5000

cm

7.10.2.6 #define LIDAR_DIST_DIFF_MIN -5000

cm

7.10.2.7 #define PITCH_MAX 500

7.10.2.8 #define PITCH_MIN -500

7.10.2.9 #define ROLL_MAX 500

7.10.2.10 #define ROLL_MIN -500

7.10.2.11 #define THROTTLE_MAX 500

7.10.2.12 #define THROTTLE_MIN -500

scaled min

7.10.2.13 #define YAW_ANGLE_DIFF_MAX 45

7.10.2.14 #define YAW_ANGLE_DIFF_MIN -45

7.10.2.15 #define YAW_MAX 500

7.10.2.16 #define YAW_MIN -500

7.11 /home/dprandle/Documents/code/ctrlmod/include/edpid_controller.h File Reference

```
#include <nsmath.h>
```

Classes

- class edpid_controller< T >
- struct edpid_controller< T >::output_range

7.12 /home/dprandle/Documents/code/ctrlmod/include/edplsystem.h File Reference

System responsible for creating messages with laser distances.

```
#include <edglobal.h>
#include <edsystem.h>
#include <mraa/gpio.hpp>
#include <edcallback.h>
#include <nsmath.h>
#include <map>
```

Classes

- class edpl_system
- struct edpl_system::pl_gpio
- struct edpl_callback

Macros

- #define GPIO_14 36
- #define GPIO_15 48
- #define GPIO_48 33
- #define GPIO_49 47

7.12.1 Detailed Description

System responsible for creating messages with laser distances.

Author

Daniel <dprandle-CZ-17>

Date

Tue Jul 7 09:19:49 2015

7.12.2 Macro Definition Documentation

7.12.2.1 #define GPIO_14 36

7.12.2.2 #define GPIO_15 48

7.12.2.3 #define GPIO_48 33

7.12.2.4 #define GPIO_49 47

7.13 /home/dprandle/Documents/code/ctrlmod/include/edrplidar_packets.h File Reference

```
#include <string>
```

Classes

- struct data_packet
- struct scan_data_packet
- struct complete_scan_data_packet
- struct health_data_packet
- struct info_data_packet
- struct firmware_data_packet
- struct request_packet
- struct stop_scan_request
- struct start_scan_request
- struct force_scan_request
- struct reset_request
- struct device_info_request
- struct device_health_request
- struct descriptor_packet
- struct scan_descriptor
- struct device_info_descriptor
- struct device_health_descriptor

7.14 /home/dprandle/Documents/code/ctrlmod/include/edrplidar_system.h File Reference

```
#include <edsystem.h>
#include <vector>
#include <edutility.h>
#include <edrplidar_packets.h>
#include <edcallback.h>
#include <mraa/uart.h>
#include <edmessage.h>
```

Classes

- class edrplidar_system

Macros

- #define XV_BAUD 115200

7.14.1 Macro Definition Documentation

7.14.1.1 #define XV_BAUD 115200

7.15 /home/dprandle/Documents/code/ctrlmod/include/edsocket.h File Reference

```
#include <edthreaded_fd.h>
```

Classes

- class edsocket

7.16 /home/dprandle/Documents/code/ctrlmod/include/edsystem.h File Reference

```
#include <string>
```

Classes

- class edsystem

7.17 /home/dprandle/Documents/code/ctrlmod/include/edthreaded_fd.h File Reference

```
#include <edglobal.h>
#include <pthread.h>
#include <vector>
#include <edcallback.h>
```

Classes

- class edthreaded_fd
- struct edthreaded_fd::Error
- struct edthreaded_fd::WriteVal
- struct command_wait_callback

Macros

- #define DEFAULT_FD_WRITE_BUFFER_SIZE 5120
- #define DEFAULT_FD_READ_BUFFER_SIZE 5120
- #define FD_TMP_BUFFER_SIZE 1024

7.17.1 Macro Definition Documentation

7.17.1.1 #define DEFAULT_FD_READ_BUFFER_SIZE 5120

7.17.1.2 #define DEFAULT_FD_WRITE_BUFFER_SIZE 5120

7.17.1.3 #define FD_TMP_BUFFER_SIZE 1024

7.18 /home/dprandle/Documents/code/ctrlmod/include/edtimer.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>
#include <chrono>
```

Classes

- class edtimer

  class edtimer

7.19 /home/dprandle/Documents/code/ctrlmod/include/eduart.h File Reference

```
#include <termios.h>
#include <edthreaded_fd.h>
#include <string>
```

Classes

- class eduart
- struct eduart::DataFormat

7.20 /home/dprandle/Documents/code/ctrlmod/include/edutility.h File Reference

```
#include <edglobal.h>
#include <iostream>
#include <string>
#include <sstream>
#include <pthread.h>
```

Functions

- uint32_t hash_id (const std::string &to_hash)
- bool log_message (const std::string &msg, const std::string &fname="status.log", bool timestamp=true)
- void cprint (const std::string &str)
- std::string timestamp ()
- void delay (double ms)
- std::string to_hex (uint8_t byte)
- std::string to_hex (int16_t two_bytes)
- std::string to_hex (uint16_t two_bytes)
- std::string to_hex (int32_t four_bytes)
- std::string to_hex (uint32_t four_bytes)
- void zero_buf (uint8_t *buf, uint32_t size)
- void copy_buf (const uint8_t *src, uint8_t *dest, uint32_t size, uint32_t src_offset=0, uint32_t dest_offset=0)

  Copy buffer.

7.20.1 Function Documentation

7.20.1.1 void copy_buf (const uint8_t * *src*, uint8_t * *dest*, uint32_t *size*, uint32_t *src_offset* = `0`, uint32_t *dest_offset* = `0`)

Copy buffer.

Copy the source buffer to destination buffer with possible offsets in each buffer

Parameters

src	The source buffer
dest	The destination buffer
size	Amount of items to copy
src_offset	Offset in to the source buffer (defaults to 0)
dest_offset	Offset in to the destination buffer (defaults to 0)

7.20.1.2 void cprint (const std::string & *str*)

7.20.1.3 void delay (double *ms*)

7.20.1.4 uint32_t hash_id (const std::string & *to_hash*)

7.20.1.5 bool log_message (const std::string & *msg*, const std::string & *fname* = `"status.log"`, bool *timestamp* = `true`)

7.20.1.6 std::string timestamp ( )

7.20.1.7 std::string to_hex (uint8_t *byte*)

7.20.1.8 std::string to_hex (int16_t *two_bytes*)

7.20.1.9 std::string to_hex (uint16_t *two_bytes*)

7.20.1.10 std::string to_hex (int32_t *four_bytes*)

7.20.1.11 std::string to_hex (uint32_t *four_bytes*)

7.20.1.12 void zero_buf (uint8_t * *buf*, uint32_t *size*)

7.21 /home/dprandle/Documents/code/ctrlmod/include/nsmat2.h File Reference

```
#include "nsquat.h"
```

Classes

- struct nsmat2< T >

Functions

- template<class T >
  nsmat2< T > operator* (const int32_t &pLHS, const nsmat2< T > &pRHS)
- template<class T >
  nsmat2< T > operator* (const float &pLHS, const nsmat2< T > &pRHS)
- template<class T >
  nsmat2< T > operator* (const double &pLHS, const nsmat2< T > &pRHS)
- template<class T >
  nsmat2< T > operator/ (const int32_t &pLHS, const nsmat2< T > &pRHS)
- template<class T >
  nsmat2< T > operator/ (const float &pLHS, const nsmat2< T > &pRHS)
- template<class T >
  nsmat2< T > operator/ (const double &pLHS, const nsmat2< T > &pRHS)
- template<class T >
  NSVec2< T > operator* (const NSVec2< T > &lhs, const nsmat2< T > &rhs)
- template<class T >
  NSVec2< T > operator/ (const NSVec2< T > &lhs, const nsmat2< T > &rhs)
- template<class T >
  nsmat2< T > operator% (const NSVec2< T > &lhs, const nsmat2< T > &rhs)
- template<class T >
  T determinant (const nsmat2< T > &mat)
- template<class T >
  nsmat2< T > rotation2d_mat2 (const T &angle, bool rads=false)
- template<class T >
  nsmat2< T > rotation2d_mat2 (const nsmat3< T > &transform2d)
- template<class T >
  nsmat2< T > rotation2d_mat2 (const nsmat2< T > &transform2d)
- template<class T >
  nsmat2< T > scaling2d_mat2 (const NSVec2< T > &scale)
- template<class T >
  nsmat2< T > scaling2d_mat2 (const nsmat2< T > &transform2d)
- template<class T >
  nsmat2< T > scaling2d_mat2 (const nsmat3< T > &transform2d)
- template<class T >
  nsmat2< T > transpose (const nsmat2< T > mat)
- template<class T >
  nsmat2< T > inverse (const nsmat2< T > mat)
- template<class PUPer , class T >
  void pup (PUPer &p, nsmat2< T > &m2)
- template<class PUPer , class T >
  void pup (PUPer &p, nsmat2< T > &m2, const std::string &varName)

7.21.1 Function Documentation

7.21.1.1 template<class T > T determinant (const nsmat2< T > & *mat*)

7.21.1.2 template<class T > nsmat2< T > inverse (const nsmat2< T > *mat*)

7.21.1.3 template<class T > nsmat2< T > operator% (const NSVec2< T > & *lhs*, const nsmat2< T > & *rhs*)

7.21.1.4 template<class T > nsmat2< T > operator* (const int32_t & *pLHS*, const nsmat2< T > & *pRHS*)

7.21.1.5 template<class T > nsmat2< T > operator* (const float & *pLHS*, const nsmat2< T > & *pRHS*)

7.21.1.6 template<class T > nsmat2< T > operator* (const double & *pLHS*, const nsmat2< T > & *pRHS*)

7.21.1.7 template<class T > NSVec2< T > operator* (const NSVec2< T > & *lhs*, const nsmat2< T > & *rhs*)

7.21.1.8 template<class T > nsmat2< T > operator/ (const int32_t & *pLHS*, const nsmat2< T > & *pRHS*)

7.21.1.9 template<class T > nsmat2< T > operator/ (const float & *pLHS*, const nsmat2< T > & *pRHS*)

7.21.1.10 template<class T > nsmat2< T > operator/ (const double & *pLHS*, const nsmat2< T > & *pRHS*)

7.21.1.11 template<class T > NSVec2< T > operator/ (const NSVec2< T > & *lhs*, const nsmat2< T > & *rhs*)

7.21.1.12 template<class PUPer , class T > void pup (PUPer & *p*, nsmat2< T > & *m2*)

7.21.1.13 template<class PUPer , class T > void pup (PUPer & *p*, nsmat2< T > & *m2*, const std::string & *varName*)

7.21.1.14 template<class T > nsmat2< T > rotation2d_mat2 (const T & *angle*, bool *rads* = `false`)

7.21.1.15 template<class T > nsmat2< T > rotation2d_mat2 (const nsmat3< T > & *transform2d*)

7.21.1.16 template<class T > nsmat2< T > rotation2d_mat2 (const nsmat2< T > & *transform2d*)

7.21.1.17 template<class T > nsmat2< T > scaling2d_mat2 (const NSVec2< T > & *scale*)

7.21.1.18 template<class T > nsmat2< T > scaling2d_mat2 (const nsmat2< T > & *transform2d*)

7.21.1.19 template<class T > nsmat2< T > scaling2d_mat2 (const nsmat3< T > & *transform2d*)

7.21.1.20 template<class T > nsmat2< T > transpose (const nsmat2< T > *mat*)

7.22 /home/dprandle/Documents/code/ctrlmod/include/nsmat3.h File Reference

```
#include "nsmat2.h"
```

Classes

- struct nsmat3< T >

Functions

- template<class T >
  NSVec3< T > operator* (const NSVec3< T > &lhs, const nsmat3< T > &rhs)

- template<class T >
  NSVec3< T > operator/ (const NSVec3< T > &lhs, const nsmat3< T > &rhs)
- template<class T >
  nsmat3< T > operator% (const NSVec3< T > &lhs, const nsmat3< T > &rhs)
- template<class T >
  nsmat3< T > operator* (const int32_t &pLHS, const nsmat3< T > &pRHS)
- template<class T >
  nsmat3< T > operator* (const float &pLHS, const nsmat3< T > &pRHS)
- template<class T >
  nsmat3< T > operator* (const double &pLHS, const nsmat3< T > &pRHS)
- template<class T >
  nsmat3< T > operator/ (const int32_t &pLHS, const nsmat3< T > &pRHS)
- template<class T >
  nsmat3< T > operator/ (const float &pLHS, const nsmat3< T > &pRHS)
- template<class T >
  nsmat3< T > operator/ (const double &pLHS, const nsmat3< T > &pRHS)
- template<class T >
  T determinant (const nsmat3< T > &mat)
- template<class T >
  nsmat3< T > rotation2d_mat3 (const T angle, bool rads=false)
- template<class T >
  nsmat3< T > rotation2d_mat3 (const nsmat3< T > &transform2d)
- template<class T >
  nsmat3< T > rotation2d_mat3 (const nsmat2< T > &transform2d)
- template<class T >
  nsmat3< T > rotation_mat3 (const NSVec4< T > &axisAngle, bool rads=false)
- template<class T >
  nsmat3< T > rotation_mat3 (const NSVec3< T > &euler, typename NSVec3< T >::EulerOrder order, bool rads=false)
- template<class T >
  nsmat3< T > rotation_mat3 (const nsquat< T > &orientation)
- template<class T >
  nsmat3< T > rotation_mat3 (const NSVec3< T > &vec, const NSVec3< T > &toVec)
- template<class T >
  nsmat3< T > rotation_mat3 (const nsmat4< T > &transform)
- template<class T >
  nsmat3< T > scaling2d_mat3 (const NSVec2< T > &scale)
- template<class T >
  nsmat3< T > scaling2d_mat3 (const nsmat2< T > &transform2d)
- template<class T >
  nsmat3< T > scaling2d_mat3 (const nsmat3< T > &transform2d)
- template<class T >
  nsmat3< T > scaling_mat3 (const NSVec3< T > &scale)
- template<class T >
  nsmat3< T > scaling_mat3 (const nsmat3< T > &transform)
- template<class T >
  nsmat3< T > scaling_mat3 (const nsmat4< T > &transform)
- template<class T >
  nsmat3< T > translation2d_mat3 (const NSVec3< T > &v3)
- template<class T >
  nsmat3< T > translation2d_mat3 (const NSVec2< T > &v2)
- template<class T >
  nsmat3< T > transpose (const nsmat3< T > mat)
- template<class T >
  nsmat3< T > inverse (const nsmat3< T > mat)
- template<class PUPer , class T >
  void pup (PUPer &p, nsmat3< T > &m3)

- template<class T >
  nsmat3< T > rotationMat3 (const nsmat3< T > &transform)
- template<class PUPer , class T >
  void pup (PUPer &p, nsmat3< T > &m3, const std::string &varName)

7.22.1 Function Documentation

7.22.1.1 template<class T > T determinant (const nsmat3< T > & *mat*)

7.22.1.2 template<class T > nsmat3< T > inverse (const nsmat3< T > *mat*)

7.22.1.3 template<class T > nsmat3< T > operator% (const NSVec3< T > & *lhs*, const nsmat3< T > & *rhs*)

7.22.1.4 template<class T > NSVec3< T > operator* (const NSVec3< T > & *lhs*, const nsmat3< T > & *rhs*)

7.22.1.5 template<class T > nsmat3< T > operator* (const int32_t & *pLHS*, const nsmat3< T > & *pRHS*)

7.22.1.6 template<class T > nsmat3< T > operator* (const float & *pLHS*, const nsmat3< T > & *pRHS*)

7.22.1.7 template<class T > nsmat3< T > operator* (const double & *pLHS*, const nsmat3< T > & *pRHS*)

7.22.1.8 template<class T > NSVec3< T > operator/ (const NSVec3< T > & *lhs*, const nsmat3< T > & *rhs*)

7.22.1.9 template<class T > nsmat3< T > operator/ (const int32_t & *pLHS*, const nsmat3< T > & *pRHS*)

7.22.1.10 template<class T > nsmat3< T > operator/ (const float & *pLHS*, const nsmat3< T > & *pRHS*)

7.22.1.11 template<class T > nsmat3< T > operator/ (const double & *pLHS*, const nsmat3< T > & *pRHS*)

7.22.1.12 template<class PUPer , class T > void pup (PUPer & *p*, nsmat3< T > & *m3*)

7.22.1.13 template<class PUPer , class T > void pup (PUPer & *p*, nsmat3< T > & *m3*, const std::string & *varName*)

7.22.1.14 template<class T > nsmat3< T > rotation2d_mat3 (const T *angle*, bool *rads* = `false`)

7.22.1.15 template<class T > nsmat3< T > rotation2d_mat3 (const nsmat3< T > & *transform2d*)

7.22.1.16 template<class T > nsmat3< T > rotation2d_mat3 (const nsmat2< T > & *transform2d*)

7.22.1.17 template<class T > nsmat3< T > rotation_mat3 (const NSVec4< T > & *axisAngle*, bool *rads* = `false`)

7.22.1.18 template<class T > nsmat3< T > rotation_mat3 (const NSVec3< T > & *euler*, typename NSVec3< T >::EulerOrder *order*, bool *rads* = `false`)

7.22.1.19 template<class T > nsmat3< T > rotation_mat3 (const nsquat< T > & *orientation*)

7.22.1.20 template<class T > nsmat3< T > rotation_mat3 (const NSVec3< T > & *vec*, const NSVec3< T > & *toVec*)

7.22.1.21 template<class T > nsmat3< T > rotation_mat3 (const nsmat4< T > & *transform*)

7.22.1.22 template<class T > nsmat3<T> rotationMat3 (const nsmat3< T > & *transform*)

7.22.1.23 template<class T > nsmat3< T > scaling2d_mat3 (const NSVec2< T > & *scale*)

7.22.1.24 template<class T > nsmat3< T > scaling2d_mat3 (const nsmat2< T > & *transform2d*)

7.22.1.25 template<class T > nsmat3< T > scaling2d_mat3 (const nsmat3< T > & *transform2d*)

7.22.1.26 template<class T > nsmat3< T > scaling_mat3 (const NSVec3< T > & *scale*)

7.22.1.27 template<class T > nsmat3< T > scaling_mat3 (const nsmat3< T > & *transform*)

7.22.1.28 template<class T > nsmat3< T > scaling_mat3 (const nsmat4< T > & *transform*)

7.22.1.29 template<class T > nsmat3< T > translation2d_mat3 (const NSVec3< T > & *v3*)

7.22.1.30 template<class T > nsmat3< T > translation2d_mat3 (const NSVec2< T > & *v2*)

7.22.1.31 template<class T > nsmat3< T > transpose (const nsmat3< T > *mat*)

7.23 /home/dprandle/Documents/code/ctrlmod/include/nsmat4.h File Reference

```
#include "nsmat3.h"
```

Classes

- struct nsmat4< T >

Functions

- template<class T >
  nsmat4< T > operator* (const int32_t &pLHS, const nsmat4< T > &pRHS)
- template<class T >
  nsmat4< T > operator* (const float &pLHS, const nsmat4< T > &pRHS)
- template<class T >
  nsmat4< T > operator* (const double &pLHS, const nsmat4< T > &pRHS)
- template<class T >
  nsmat4< T > operator/ (const int32_t &pLHS, const nsmat4< T > &pRHS)
- template<class T >
  nsmat4< T > operator/ (const float &pLHS, const nsmat4< T > &pRHS)
- template<class T >
  nsmat4< T > operator/ (const double &pLHS, const nsmat4< T > &pRHS)
- template<class T >
  NSVec3< T > operator* (const NSVec3< T > &lhs, const nsmat4< T > &rhs)
- template<class T >
  NSVec3< T > operator/ (const NSVec3< T > &lhs, const nsmat4< T > &rhs)
- template<class T >
  nsmat4< T > operator% (const NSVec3< T > &lhs, const nsmat4< T > &rhs)
- template<class T >
  T determinant (const nsmat4< T > &mat)
- template<class T >
  nsmat4< T > inverse (const nsmat4< T > mat)
- template<class T >
  nsmat4< T > ortho (const T &left, const T &right, const T &top, const T &bottom, const T &near, const T &far)
- template<class T >
  nsmat4< T > perspective (const T &fovAngle, const T &aspectRatio, const T &zNear, const T &zFar)
- template<class T >
  nsmat4< T > rotation_mat4 (const NSVec4< T > &axisAngle, bool rads=false)

- template<class T >
  nsmat4< T > rotation_mat4 (const NSVec3< T > &euler, typename NSVec3< T >::EulerOrder order, bool rads=false)
- template<class T >
  nsmat4< T > rotation_mat4 (const nsquat< T > &orientation)
- template<class T >
  nsmat4< T > rotation_mat4 (const NSVec3< T > &vec, const NSVec3< T > &toVec)
- template<class T >
  nsmat4< T > rotation_mat4 (const nsmat4< T > &transform)
- template<class T >
  nsmat4< T > scaling_mat4 (const NSVec3< T > &scale)
- template<class T >
  nsmat4< T > scaling_mat4 (const nsmat3< T > &transform)
- template<class T >
  nsmat4< T > scaling_mat4 (const nsmat4< T > &transform)
- template<class T >
  nsmat4< T > transpose (const nsmat4< T > mat)
- template<class T >
  nsmat4< T > translation_mat4 (const NSVec3< T > &pos)
- template<class T >
  nsmat4< T > translation_mat4 (const NSVec4< T > &posw)
- template<class T >
  nsmat4< T > translation_mat4 (const nsmat4< T > &transform)
- template<class PUPer , class T >
  void pup (PUPer &p, nsmat4< T > &m4)
- template<class T >
  nsmat4< T > rotationMat4 (const nsmat3< T > &transform)
- template<class PUPer , class T >
  void pup (PUPer &p, nsmat4< T > &m4, const std::string &varName)

7.23.1 Function Documentation

7.23.1.1 template<class T > T determinant (const nsmat4< T > & *mat*)

7.23.1.2 template<class T > nsmat4< T > inverse (const nsmat4< T > *mat*)

7.23.1.3 template<class T > nsmat4< T > operator% (const NSVec3< T > & *lhs*, const nsmat4< T > & *rhs*)

7.23.1.4 template<class T > nsmat4< T > operator* (const int32_t & *pLHS*, const nsmat4< T > & *pRHS*)

7.23.1.5 template<class T > nsmat4< T > operator* (const float & *pLHS*, const nsmat4< T > & *pRHS*)

7.23.1.6 template<class T > nsmat4< T > operator* (const double & *pLHS*, const nsmat4< T > & *pRHS*)

7.23.1.7 template<class T > NSVec3< T > operator* (const NSVec3< T > & *lhs*, const nsmat4< T > & *rhs*)

7.23.1.8 template<class T > nsmat4< T > operator/ (const int32_t & *pLHS*, const nsmat4< T > & *pRHS*)

7.23.1.9 template<class T > nsmat4< T > operator/ (const float & *pLHS*, const nsmat4< T > & *pRHS*)

7.23.1.10 template<class T > nsmat4< T > operator/ (const double & *pLHS*, const nsmat4< T > & *pRHS*)

7.23.1.11 template<class T > NSVec3< T > operator/ (const NSVec3< T > & *lhs*, const nsmat4< T > & *rhs*)

7.23.1.12 template<class T > nsmat4< T > ortho (const T & *left*, const T & *right*, const T & *top*, const T & *bottom*, const T & *near*, const T & *far*)

7.23.1.13 template<class T > nsmat4< T > perspective (const T & *fovAngle,* const T & *aspectRatio,* const T & *zNear,* const T & *zFar*)

7.23.1.14 template<class PUPer , class T > void pup (PUPer & *p,* nsmat4< T > & *m4*)

7.23.1.15 template<class PUPer , class T > void pup (PUPer & *p,* nsmat4< T > & *m4,* const std::string & *varName*)

7.23.1.16 template<class T > nsmat4< T > rotation_mat4 (const NSVec4< T > & *axisAngle,* bool *rads* = `false`)

7.23.1.17 template<class T > nsmat4< T > rotation_mat4 (const NSVec3< T > & *euler,* typename NSVec3< T >::EulerOrder *order,* bool *rads* = `false`)

7.23.1.18 template<class T > nsmat4< T > rotation_mat4 (const nsquat< T > & *orientation*)

7.23.1.19 template<class T > nsmat4< T > rotation_mat4 (const NSVec3< T > & *vec,* const NSVec3< T > & *toVec*)

7.23.1.20 template<class T > nsmat4< T > rotation_mat4 (const nsmat4< T > & *transform*)

7.23.1.21 template<class T > nsmat4<T> rotationMat4 (const nsmat3< T > & *transform*)

7.23.1.22 template<class T > nsmat4< T > scaling_mat4 (const NSVec3< T > & *scale*)

7.23.1.23 template<class T > nsmat4< T > scaling_mat4 (const nsmat3< T > & *transform*)

7.23.1.24 template<class T > nsmat4< T > scaling_mat4 (const nsmat4< T > & *transform*)

7.23.1.25 template<class T > nsmat4< T > translation_mat4 (const NSVec3< T > & *pos*)

7.23.1.26 template<class T > nsmat4< T > translation_mat4 (const NSVec4< T > & *posw*)

7.23.1.27 template<class T > nsmat4< T > translation_mat4 (const nsmat4< T > & *transform*)

7.23.1.28 template<class T > nsmat4< T > transpose (const nsmat4< T > *mat*)

7.24 /home/dprandle/Documents/code/ctrlmod/include/nsmath.h File Reference

```
#include <exception>
#include <stdexcept>
#include <cstdlib>
#include <cmath>
#include <string>
#include <sstream>
#include <edglobal.h>
#include <vector>
#include "nsmat4.h"
```

Classes

- struct NSBoundingBox

Namespaces

- std

Macros

- #define PI 3.14159265359f
- #define EPS 0.0001f

Typedefs

- typedef NSVec2< uint8_t > cvec2
- typedef NSVec2< int32_t > ivec2
- typedef NSVec2< uint8_t > ucvec2
- typedef NSVec2< uint32_t > uivec2
- typedef NSVec2< float > fvec2
- typedef NSVec2< double > vec2
- typedef NSVec3< uint8_t > cvec3
- typedef NSVec3< int32_t > ivec3
- typedef NSVec3< uint8_t > ucvec3
- typedef NSVec3< uint32_t > uivec3
- typedef NSVec3< float > fvec3
- typedef NSVec3< double > vec3
- typedef NSVec4< uint8_t > cvec4
- typedef NSVec4< int32_t > ivec4
- typedef NSVec4< uint8_t > ucvec4
- typedef NSVec4< uint32_t > uivec4
- typedef NSVec4< float > fvec4
- typedef NSVec4< double > vec4
- typedef nsquat< uint8_t > cquat
- typedef nsquat< int32_t > iquat
- typedef nsquat< uint8_t > ucquat
- typedef nsquat< uint32_t > uiquat
- typedef nsquat< float > fquat
- typedef nsquat< double > quat
- typedef nsmat2< uint8_t > cmat2
- typedef nsmat2< int32_t > imat2
- typedef nsmat2< uint8_t > ucmat2
- typedef nsmat2< uint32_t > uimat2
- typedef nsmat2< float > fmat2
- typedef nsmat2< double > mat2
- typedef nsmat3< uint8_t > cmat3
- typedef nsmat3< int32_t > imat3
- typedef nsmat3< uint8_t > ucmat3
- typedef nsmat3< uint32_t > uimat3
- typedef nsmat3< float > fmat3
- typedef nsmat3< double > mat3
- typedef nsmat4< uint8_t > cmat4
- typedef nsmat4< uint8_t > ucmat4
- typedef nsmat4< int32_t > imat4
- typedef nsmat4< uint32_t > uimat4
- typedef nsmat4< float > fmat4
- typedef nsmat4< double > mat4

Functions

- template<class T >
  T std::round (const T &n)
- float clampf (float pVal, const float &pMin, const float &pMax)
- double clamp (double pVal, const double &pMin, const double &pMax)
- float fractf (const float &num)
- double fract (const double &num)
- float lerp (int32_t low, int32_t high, int32_t middle)
- float lerp (uint32_t low, uint32_t high, uint32_t middle)
- float lerp (float low, float high, float middle)
- double lerp (double low, double high, double middle)
- template<class T >
  T degrees (const T &val)
- template<class T >
  T radians (const T &val)
- float random_float (float pHigh=1.0f, float pLow=0.0f)

7.24.1 Macro Definition Documentation

7.24.1.1 #define EPS 0.0001f

7.24.1.2 #define PI 3.14159265359f

7.24.2 Typedef Documentation

7.24.2.1 typedef nsmat2<uint8_t> cmat2

7.24.2.2 typedef nsmat3<uint8_t> cmat3

7.24.2.3 typedef nsmat4<uint8_t> cmat4

7.24.2.4 typedef nsquat<uint8_t> cquat

7.24.2.5 typedef NSVec2<uint8_t> cvec2

7.24.2.6 typedef NSVec3<uint8_t> cvec3

7.24.2.7 typedef NSVec4<uint8_t> cvec4

7.24.2.8 typedef nsmat2<float> fmat2

7.24.2.9 typedef nsmat3<float> fmat3

7.24.2.10 typedef nsmat4<float> fmat4

7.24.2.11 typedef nsquat<float> fquat

7.24.2.12 typedef NSVec2<float> fvec2

7.24.2.13 typedef NSVec3<float> fvec3

7.24.2.14 typedef NSVec4<float> fvec4

7.24.2.15 typedef nsmat2<int32_t> imat2

7.24.2.16 typedef nsmat3<int32_t> imat3

7.24.2.17 typedef nsmat4<int32_t> imat4

7.24.2.18 typedef nsquat<int32_t> iquat

7.24.2.19 typedef NSVec2<int32_t> ivec2

7.24.2.20 typedef NSVec3<int32_t> ivec3

7.24.2.21 typedef NSVec4<int32_t> ivec4

7.24.2.22 typedef nsmat2<double> mat2

7.24.2.23 typedef nsmat3<double> mat3

7.24.2.24 typedef nsmat4<double> mat4

7.24.2.25 typedef nsquat<double> quat

7.24.2.26 typedef nsmat2<uint8_t> ucmat2

7.24.2.27 typedef nsmat3<uint8_t> ucmat3

7.24.2.28 typedef nsmat4<uint8_t> ucmat4

7.24.2.29 typedef nsquat<uint8_t> ucquat

7.24.2.30 typedef NSVec2<uint8_t> ucvec2

7.24.2.31 typedef NSVec3<uint8_t> ucvec3

7.24.2.32 typedef NSVec4<uint8_t> ucvec4

7.24.2.33 typedef nsmat2<uint32_t> uimat2

7.24.2.34 typedef nsmat3<uint32_t> uimat3

7.24.2.35 typedef nsmat4<uint32_t> uimat4

7.24.2.36 typedef nsquat<uint32_t> uiquat

7.24.2.37 typedef NSVec2<uint32_t> uivec2

7.24.2.38 typedef NSVec3<uint32_t> uivec3

7.24.2.39 typedef NSVec4<uint32_t> uivec4

7.24.2.40 typedef NSVec2<double> vec2

7.24.2.41 typedef NSVec3<double> vec3

7.24.2.42 typedef NSVec4<double> vec4

7.24.3 Function Documentation

7.24.3.1 double clamp (double *pVal,* const double & *pMin,* const double & *pMax*)

7.24.3.2 float clampf (float *pVal,* const float & *pMin,* const float & *pMax*)

7.24.3.3 template<class T > T degrees (const T & *val*)

7.24.3.4 double fract (const double & *num*)

7.24.3.5 float fractf (const float & *num*)

7.24.3.6 float lerp (int32_t *low,* int32_t *high,* int32_t *middle*)

7.24.3.7 float lerp (uint32_t *low,* uint32_t *high,* uint32_t *middle*)

7.24.3.8 float lerp (float *low,* float *high,* float *middle*)

7.24.3.9 double lerp (double *low,* double *high,* double *middle*)

7.24.3.10 template<class T > T radians (const T & *val*)

7.24.3.11 float random_float (float *pHigh* = `1.0f`, float *pLow* = `0.0f`)

7.25 /home/dprandle/Documents/code/ctrlmod/include/nsquat.h File Reference

```
#include "nsvec4.h"
```

Classes

- struct nsquat< T >

Functions

- template<class T >
  nsquat< T > operator* (const int32_t &pLHS, const nsquat< T > &pRHS)
- template<class T >
  nsquat< T > operator* (const float &pLHS, const nsquat< T > &pRHS)
- template<class T >
  nsquat< T > operator* (const double &pLHS, const nsquat< T > &pRHS)
- template<class T >
  nsquat< T > operator/ (const int32_t &pLHS, const nsquat< T > &pRHS)
- template<class T >
  nsquat< T > operator/ (const float &pLHS, const nsquat< T > &pRHS)
- template<class T >
  nsquat< T > operator/ (const double &pLHS, const nsquat< T > &pRHS)
- template<class T >
  nsquat< T > orientation (const nsmat3< T > &rotationMat3)
- template<class T >
  nsquat< T > orientation (const NSVec4< T > &axisAngle, bool pRads=false)
- template<class T >
  nsquat< T > orientation (const NSVec3< T > &euler, typename NSVec3< T >::EulerOrder order, bool pRads=false)
- template<class T >
  nsquat< T > orientation (const NSVec3< T > &vec, const NSVec3< T > &toVec)

- template<class T >
  nsquat< T > orientation (const nsmat4< T > &transform)
- template<class T >
  nsquat< T > conjugate (const nsquat< T > &quat)
- template<class T >
  T dot (const nsquat< T > &pLeft, const nsquat< T > &pRight)
- template<class T >
  nsquat< T > inverse (const nsquat< T > &quat)
- template<class T >
  nsquat< T > normalize (const nsquat< T > &quat)
- template<class T >
  nsquat< T > slerp (const nsquat< T > &lhs, const nsquat< T > &rhs, const T &scalingFactor)
- template<class PUPer , class T >
  void pup (PUPer &p, nsquat< T > &q4)
- template<class PUPer , class T >
  void pup (PUPer &p, nsquat< T > &q4, const std::string &varName)

7.25.1 Function Documentation

7.25.1.1 template<class T > nsquat< T > conjugate (const nsquat< T > & *quat*)

7.25.1.2 template<class T > T dot (const nsquat< T > & *pLeft,* const nsquat< T > & *pRight*)

7.25.1.3 template<class T > nsquat< T > inverse (const nsquat< T > & *quat*)

7.25.1.4 template<class T > nsquat< T > normalize (const nsquat< T > & *quat*)

7.25.1.5 template<class T > nsquat< T > operator* (const int32_t & *pLHS,* const nsquat< T > & *pRHS*)

7.25.1.6 template<class T > nsquat< T > operator* (const float & *pLHS,* const nsquat< T > & *pRHS*)

7.25.1.7 template<class T > nsquat< T > operator* (const double & *pLHS,* const nsquat< T > & *pRHS*)

7.25.1.8 template<class T > nsquat< T > operator/ (const int32_t & *pLHS,* const nsquat< T > & *pRHS*)

7.25.1.9 template<class T > nsquat< T > operator/ (const float & *pLHS,* const nsquat< T > & *pRHS*)

7.25.1.10 template<class T > nsquat< T > operator/ (const double & *pLHS,* const nsquat< T > & *pRHS*)

7.25.1.11 template<class T > nsquat< T > orientation (const nsmat3< T > & *rotationMat3*)

7.25.1.12 template<class T > nsquat< T > orientation (const NSVec4< T > & *axisAngle,* bool *pRads* = `false`)

7.25.1.13 template<class T > nsquat< T > orientation (const NSVec3< T > & *euler,* typename NSVec3< T >::EulerOrder *order,* bool *pRads* = `false`)

7.25.1.14 template<class T > nsquat< T > orientation (const NSVec3< T > & *vec,* const NSVec3< T > & *toVec*)

7.25.1.15 template<class T > nsquat< T > orientation (const nsmat4< T > & *transform*)

7.25.1.16 template<class PUPer , class T > void pup (PUPer & *p,* nsquat< T > & *q4*)

7.25.1.17 template<class PUPer , class T > void pup (PUPer & *p,* nsquat< T > & *q4,* const std::string & *varName*)

7.25.1.18 template<class T > nsquat< T > slerp (const nsquat< T > & *lhs,* const nsquat< T > & *rhs,* const T & *scalingFactor*)

7.26 /home/dprandle/Documents/code/ctrlmod/include/nsvec2.h File Reference

```
#include <cmath>
```

Classes

- struct NSVec3< T >
- struct NSVec4< T >
- struct nsquat< T >
- struct nsmat2< T >
- struct nsmat3< T >
- struct nsmat4< T >
- struct NSVec2< T >
- struct NSVec2< T >

Functions

- template<class T >
  NSVec2< T > operator* (const int32_t &pLHS, const NSVec2< T > &pRHS)
- template<class T >
  NSVec2< T > operator* (const float &pLHS, const NSVec2< T > &pRHS)
- template<class T >
  NSVec2< T > operator* (const double &pLHS, const NSVec2< T > &pRHS)
- template<class T >
  NSVec2< T > operator/ (const int32_t &pLHS, const NSVec2< T > &pRHS)
- template<class T >
  NSVec2< T > operator/ (const float &pLHS, const NSVec2< T > &pRHS)
- template<class T >
  NSVec2< T > operator/ (const double &pLHS, const NSVec2< T > &pRHS)
- template<class T >
  NSVec2< T > abs (const NSVec2< T > &pVec)
- template<class T >
  NSVec2< T > ceil (const NSVec2< T > &pVec)
- template<class T >
  NSVec2< T > clamp (const NSVec2< T > &pVec, const T &pMin, const T &pMax)
- template<class T >
  T distance (const NSVec2< T > &lvec, const NSVec2< T > &rvec)
- template<class T >
  T dot (const NSVec2< T > &pLeft, const NSVec2< T > &pRight)
- template<class T >
  NSVec2< T > floor (const NSVec2< T > &pVec)
- template<class T >
  NSVec2< T > fract (const NSVec2< T > &vec)
- template<class T >
  T length (const NSVec2< T > &pVec)
- template<class T , class T2 >
  NSVec2< T > lerp (const NSVec2< T > &lhs, const NSVec2< T > &rhs, T2 scalingFactor)
- template<class T >
  NSVec2< T > min (const NSVec2< T > &pLeft, const NSVec2< T > &pRight)

- template<class T >
  NSVec2< T > max (const NSVec2< T > &pLeft, const NSVec2< T > &pRight)
- template<class T >
  NSVec2< T > normalize (const NSVec2< T > &pRHS)
- template<class T >
  NSVec2< T > project (const NSVec2< T > &a, const NSVec2< T > &b)
- template<class T >
  NSVec2< T > project_plane (const NSVec2< T > &a, const NSVec2< T > &normal)
- template<class T >
  NSVec2< T > reflect (const NSVec2< T > &incident, const NSVec2< T > &normal)
- template<class T >
  NSVec2< T > round (const NSVec2< T > &pVec)
- template<class T >
  NSVec2< T > scaling2d_vec (const nsmat2< T > &transform2d)
- template<class T >
  NSVec2< T > scaling2d_vec (const nsmat3< T > &transform2d)
- template<class T >
  NSVec2< T > translation2d_vec (const nsmat3< T > &transform2d)
- template<class PUPer , class T >
  void pup (PUPer &p, NSVec2< T > &v2)
- template<class PUPer , class T >
  void pup (PUPer &p, NSVec2< T > &v2, const std::string &varName)

7.26.1 Function Documentation

7.26.1.1 template<class T > NSVec2< T > abs (const NSVec2< T > & *pVec*)

7.26.1.2 template<class T > NSVec2< T > ceil (const NSVec2< T > & *pVec*)

7.26.1.3 template<class T > NSVec2< T > clamp (const NSVec2< T > & *pVec,* const T & *pMin,* const T & *pMax*)

7.26.1.4 template<class T > T distance (const NSVec2< T > & *lvec,* const NSVec2< T > & *rvec*)

7.26.1.5 template<class T > T dot (const NSVec2< T > & *pLeft,* const NSVec2< T > & *pRight*)

7.26.1.6 template<class T > NSVec2< T > floor (const NSVec2< T > & *pVec*)

7.26.1.7 template<class T > NSVec2< T > fract (const NSVec2< T > & *vec*)

7.26.1.8 template<class T > T length (const NSVec2< T > & *pVec*)

7.26.1.9 template<class T , class T2 > NSVec2< T > lerp (const NSVec2< T > & *lhs,* const NSVec2< T > & *rhs,* T2 *scalingFactor*)

7.26.1.10 template<class T > NSVec2< T > max (const NSVec2< T > & *pLeft,* const NSVec2< T > & *pRight*)

7.26.1.11 template<class T > NSVec2< T > min (const NSVec2< T > & *pLeft,* const NSVec2< T > & *pRight*)

7.26.1.12 template<class T > NSVec2< T > normalize (const NSVec2< T > & *pRHS*)

7.26.1.13 template<class T > NSVec2< T > operator* (const int32_t & *pLHS,* const NSVec2< T > & *pRHS*)

7.26.1.14 template<class T > NSVec2< T > operator* (const float & *pLHS,* const NSVec2< T > & *pRHS*)

7.26.1.15 template<class T > NSVec2< T > operator* (const double & *pLHS,* const NSVec2< T > & *pRHS*)

7.26.1.16 template<class T > NSVec2< T > operator/ (const int32_t & *pLHS,* const NSVec2< T > & *pRHS*)

7.26.1.17 template<class T > NSVec2< T > operator/ (const float & *pLHS,* const NSVec2< T > & *pRHS*)

7.26.1.18 template<class T > NSVec2< T > operator/ (const double & *pLHS,* const NSVec2< T > & *pRHS*)

7.26.1.19 template<class T > NSVec2< T > project (const NSVec2< T > & *a,* const NSVec2< T > & *b*)

7.26.1.20 template<class T > NSVec2< T > project_plane (const NSVec2< T > & *a,* const NSVec2< T > & *normal*)

7.26.1.21 template<class PUPer , class T > void pup (PUPer & *p,* NSVec2< T > & *v2*)

7.26.1.22 template<class PUPer , class T > void pup (PUPer & *p,* NSVec2< T > & *v2,* const std::string & *varName*)

7.26.1.23 template<class T > NSVec2< T > reflect (const NSVec2< T > & *incident,* const NSVec2< T > & *normal*)

7.26.1.24 template<class T > NSVec2< T > round (const NSVec2< T > & *pVec*)

7.26.1.25 template<class T > NSVec2< T > scaling2d_vec (const nsmat2< T > & *transform2d*)

7.26.1.26 template<class T > NSVec2< T > scaling2d_vec (const nsmat3< T > & *transform2d*)

7.26.1.27 template<class T > NSVec2< T > translation2d_vec (const nsmat3< T > & *transform2d*)

7.27 /home/dprandle/Documents/code/ctrlmod/include/nsvec3.h File Reference

```
#include "nsvec2.h"
```

Classes

- struct NSVec3< T >

Functions

- template<class T >
  NSVec3< T > operator* (const int32_t &pLHS, const NSVec3< T > &pRHS)
- template<class T >
  NSVec3< T > operator* (const float &pLHS, const NSVec3< T > &pRHS)
- template<class T >
  NSVec3< T > operator* (const double &pLHS, const NSVec3< T > &pRHS)
- template<class T >
  NSVec3< T > operator/ (const int32_t &pLHS, const NSVec3< T > &pRHS)
- template<class T >
  NSVec3< T > operator/ (const float &pLHS, const NSVec3< T > &pRHS)
- template<class T >
  NSVec3< T > operator/ (const double &pLHS, const NSVec3< T > &pRHS)
- template<class T >
  NSVec3< T > abs (const NSVec3< T > &pVec)
- template<class T >
  NSVec3< T > ceil (const NSVec3< T > &pVec)
- template<class T >
  NSVec3< T > clamp (const NSVec3< T > &pVec, const T &pMin, const T &pMax)
- template<class T >
  NSVec3< T > cross (const NSVec3< T > &pLeft, const NSVec3< T > &pRight)

- template<class T >
  T distance (const NSVec3< T > &lvec, const NSVec3< T > &rvec)
- template<class T >
  T dot (const NSVec3< T > &pLeft, const NSVec3< T > &pRight)
- template<class T >
  NSVec3< T > euler (const NSVec4< T > &axisAngle, typename NSVec3< T >::EulerOrder order, bool pRads=false)
- template<class T >
  NSVec3< T > euler (const nsquat< T > &orientation, typename NSVec3< T >::EulerOrder order, bool rads=false)
- template<class T >
  NSVec3< T > euler (const nsmat3< T > &rotationMat3, typename NSVec3< T >::EulerOrder order, bool pRads=false)
- template<class T >
  NSVec3< T > euler (const nsmat4< T > &transform, typename NSVec3< T >::EulerOrder order, bool pRads=false)
- template<class T >
  NSVec3< T > euler (const NSVec3< T > &vec, const NSVec3< T > &toVec, typename NSVec3< T >-::EulerOrder order, bool pRads=false)
- template<class T >
  NSVec3< T > floor (const NSVec3< T > &pVec)
- template<class T >
  NSVec3< T > fract (const NSVec3< T > &vec)
- template<class T >
  T length (const NSVec3< T > &pVec)
- template<class T , class T2 >
  NSVec3< T > lerp (const NSVec3< T > &lhs, const NSVec3< T > &rhs, T2 scalingFactor)
- template<class T >
  NSVec3< T > min (const NSVec3< T > &pLeft, const NSVec3< T > &pRight)
- template<class T >
  NSVec3< T > max (const NSVec3< T > &pLeft, const NSVec3< T > &pRight)
- template<class T >
  NSVec3< T > normalize (const NSVec3< T > &pRHS)
- template<class T >
  NSVec3< T > project (const NSVec3< T > &a, const NSVec3< T > &b)
- template<class T >
  NSVec3< T > projectPlane (const NSVec3< T > &a, const NSVec3< T > &normal)
- template<class T >
  NSVec3< T > reflect (const NSVec3< T > &incident, const NSVec3< T > &normal)
- template<class T >
  NSVec3< T > round (const NSVec3< T > &pVec)
- template<class T >
  NSVec3< T > scaling_vec (const nsmat3< T > &transform)
- template<class T >
  NSVec3< T > scaling_vec (const nsmat4< T > &transform)
- template<class T >
  NSVec3< T > translation_vec (const nsmat4< T > &transform)
- template<class PUPer , class T >
  void pup (PUPer &p, NSVec3< T > &v3)
- template<class PUPer , class T >
  void pup (PUPer &p, NSVec3< T > &v3, const std::string &varName)

7.27.1 Function Documentation

7.27.1.1 template<class T > NSVec3< T > abs (const NSVec3< T > & *pVec*)

7.27.1.2 template<class T > NSVec3< T > ceil (const NSVec3< T > & *pVec*)

7.27.1.3 template<class T > NSVec3< T > clamp (const NSVec3< T > & *pVec,* const T & *pMin,* const T & *pMax*)

7.27.1.4 template<class T > NSVec3< T > cross (const NSVec3< T > & *pLeft,* const NSVec3< T > & *pRight*)

7.27.1.5 template<class T > T distance (const NSVec3< T > & *lvec,* const NSVec3< T > & *rvec*)

7.27.1.6 template<class T > T dot (const NSVec3< T > & *pLeft,* const NSVec3< T > & *pRight*)

7.27.1.7 template<class T > NSVec3< T > euler (const NSVec4< T > & *axisAngle,* typename NSVec3< T >::EulerOrder *order,* bool *pRads* = `false`)

7.27.1.8 template<class T > NSVec3< T > euler (const nsquat< T > & *orientation,* typename NSVec3< T >::EulerOrder *order,* bool *rads* = `false`)

7.27.1.9 template<class T > NSVec3< T > euler (const nsmat3< T > & *rotationMat3,* typename NSVec3< T >::EulerOrder *order,* bool *pRads* = `false`)

7.27.1.10 template<class T > NSVec3< T > euler (const nsmat4< T > & *transform,* typename NSVec3< T >::EulerOrder *order,* bool *pRads* = `false`)

7.27.1.11 template<class T > NSVec3< T > euler (const NSVec3< T > & *vec,* const NSVec3< T > & *toVec,* typename NSVec3< T >::EulerOrder *order,* bool *pRads* = `false`)

7.27.1.12 template<class T > NSVec3< T > floor (const NSVec3< T > & *pVec*)

7.27.1.13 template<class T > NSVec3< T > fract (const NSVec3< T > & *vec*)

7.27.1.14 template<class T > T length (const NSVec3< T > & *pVec*)

7.27.1.15 template<class T , class T2 > NSVec3< T > lerp (const NSVec3< T > & *lhs,* const NSVec3< T > & *rhs,* T2 *scalingFactor*)

7.27.1.16 template<class T > NSVec3< T > max (const NSVec3< T > & *pLeft,* const NSVec3< T > & *pRight*)

7.27.1.17 template<class T > NSVec3< T > min (const NSVec3< T > & *pLeft,* const NSVec3< T > & *pRight*)

7.27.1.18 template<class T > NSVec3< T > normalize (const NSVec3< T > & *pRHS*)

7.27.1.19 template<class T > NSVec3< T > operator* (const int32_t & *pLHS,* const NSVec3< T > & *pRHS*)

7.27.1.20 template<class T > NSVec3< T > operator* (const float & *pLHS,* const NSVec3< T > & *pRHS*)

7.27.1.21 template<class T > NSVec3< T > operator* (const double & *pLHS,* const NSVec3< T > & *pRHS*)

7.27.1.22 template<class T > NSVec3< T > operator/ (const int32_t & *pLHS,* const NSVec3< T > & *pRHS*)

7.27.1.23 template<class T > NSVec3< T > operator/ (const float & *pLHS,* const NSVec3< T > & *pRHS*)

7.27.1.24 template<class T > NSVec3< T > operator/ (const double & *pLHS,* const NSVec3< T > & *pRHS*)

7.27.1.25 template<class T > NSVec3< T > project (const NSVec3< T > & *a,* const NSVec3< T > & *b*)

7.27.1.26 template<class T > NSVec3< T > projectPlane (const NSVec3< T > & *a,* const NSVec3< T > & *normal*)

7.27.1.27 template<class PUPer , class T > void pup (PUPer & *p,* NSVec3< T > & *v3*)

7.27.1.28 template<class PUPer , class T > void pup (PUPer & *p,* NSVec3< T > & *v3,* const std::string & *varName*)

7.27.1.29 template<class T > NSVec3< T > reflect (const NSVec3< T > & *incident,* const NSVec3< T > & *normal*)

7.27.1.30 template<class T > NSVec3< T > round (const NSVec3< T > & *pVec*)

7.27.1.31 template<class T > NSVec3< T > scaling_vec (const nsmat3< T > & *transform*)

7.27.1.32 template<class T > NSVec3< T > scaling_vec (const nsmat4< T > & *transform*)

7.27.1.33 template<class T > NSVec3< T > translation_vec (const nsmat4< T > & *transform*)

7.28 /home/dprandle/Documents/code/ctrlmod/include/nsvec4.h File Reference

```
#include "nsvec3.h"
```

Classes

- struct NSVec4< T >

Functions

- template<class T >
  NSVec4< T > operator* (const int32_t &pLHS, const NSVec4< T > &pRHS)
- template<class T >
  NSVec4< T > operator* (const float &pLHS, const NSVec4< T > &pRHS)
- template<class T >
  NSVec4< T > operator* (const double &pLHS, const NSVec4< T > &pRHS)
- template<class T >
  NSVec4< T > operator/ (const int32_t &pLHS, const NSVec4< T > &pRHS)
- template<class T >
  NSVec4< T > operator/ (const float &pLHS, const NSVec4< T > &pRHS)
- template<class T >
  NSVec4< T > operator/ (const double &pLHS, const NSVec4< T > &pRHS)
- template<class T >
  NSVec4< T > abs (const NSVec4< T > &pVec)
- template<class T >
  NSVec4< T > axis_angle (const NSVec3< T > &euler, typename NSVec3< T >::EulerOrder order, bool rads=false)
- template<class T >
  NSVec4< T > axis_angle (const nsquat< T > &orientation, bool rads=false)
- template<class T >
  NSVec4< T > axis_angle (const nsmat3< T > &rotationMat3, bool rads=false)
- template<class T >
  NSVec4< T > axis_angle (const nsmat4< T > &transform, bool rads=false)
- template<class T >
  NSVec4< T > axis_angle (const NSVec3< T > &vec, const NSVec3< T > &toVec, bool rads=false)
- template<class T >
  NSVec4< T > ceil (const NSVec4< T > &pVec)
- template<class T >
  NSVec4< T > clamp (const NSVec4< T > &pVec, const T &pMin, const T &pMax)

- template<class T >
  T distance (const NSVec4< T > &lvec, const NSVec4< T > &rvec)
- template<class T >
  T dot (const NSVec4< T > &pLeft, const NSVec4< T > &pRight)
- template<class T >
  NSVec4< T > floor (const NSVec4< T > &pVec)
- template<class T >
  NSVec4< T > fract (const NSVec4< T > &vec)
- template<class T >
  T length (const NSVec4< T > &pVec)
- template<class T , class T2 >
  NSVec4< T > lerp (const NSVec4< T > &lhs, const NSVec4< T > &rhs, T2 scalingFactor)
- template<class T >
  NSVec4< T > min (const NSVec4< T > &pLeft, const NSVec4< T > &pRight)
- template<class T >
  NSVec4< T > max (const NSVec4< T > &pLeft, const NSVec4< T > &pRight)
- template<class T >
  NSVec4< T > normalize (const NSVec4< T > &pRHS)
- template<class T >
  NSVec4< T > round (const NSVec4< T > &pVec)
- template<class PUPer , class T >
  void pup (PUPer &p, NSVec4< T > &v4)
- template<class PUPer , class T >
  void pup (PUPer &p, NSVec4< T > &v4, const std::string &varName)

7.28.1 Function Documentation

7.28.1.1 template<class T > NSVec4< T > abs (const NSVec4< T > & *pVec*)

7.28.1.2 template<class T > NSVec4< T > axis_angle (const NSVec3< T > & *euler,* typename NSVec3< T >::EulerOrder *order,* bool *rads* = `false`)

7.28.1.3 template<class T > NSVec4< T > axis_angle (const nsquat< T > & *orientation,* bool *rads* = `false`)

7.28.1.4 template<class T > NSVec4< T > axis_angle (const nsmat3< T > & *rotationMat3,* bool *rads* = `false`)

7.28.1.5 template<class T > NSVec4< T > axis_angle (const nsmat4< T > & *transform,* bool *rads* = `false`)

7.28.1.6 template<class T > NSVec4< T > axis_angle (const NSVec3< T > & *vec,* const NSVec3< T > & *toVec,* bool *rads* = `false`)

7.28.1.7 template<class T > NSVec4< T > ceil (const NSVec4< T > & *pVec*)

7.28.1.8 template<class T > NSVec4< T > clamp (const NSVec4< T > & *pVec,* const T & *pMin,* const T & *pMax*)

7.28.1.9 template<class T > T distance (const NSVec4< T > & *lvec,* const NSVec4< T > & *rvec*)

7.28.1.10 template<class T > T dot (const NSVec4< T > & *pLeft,* const NSVec4< T > & *pRight*)

7.28.1.11 template<class T > NSVec4< T > floor (const NSVec4< T > & *pVec*)

7.28.1.12 template<class T > NSVec4< T > fract (const NSVec4< T > & *vec*)

7.28.1.13 template<class T > T length (const NSVec4< T > & *pVec*)

7.28.1.14 template<class T , class T2 > NSVec4< T > lerp (const NSVec4< T > & *lhs,* const NSVec4< T > & *rhs,* T2 *scalingFactor*)

7.28.1.15 template<class T > NSVec4< T > max (const NSVec4< T > & *pLeft,* const NSVec4< T > & *pRight*)

7.28.1.16 template<class T > NSVec4< T > min (const NSVec4< T > & *pLeft,* const NSVec4< T > & *pRight*)

7.28.1.17 template<class T > NSVec4< T > normalize (const NSVec4< T > & *pRHS*)

7.28.1.18 template<class T > NSVec4< T > operator* (const int32_t & *pLHS,* const NSVec4< T > & *pRHS*)

7.28.1.19 template<class T > NSVec4< T > operator* (const float & *pLHS,* const NSVec4< T > & *pRHS*)

7.28.1.20 template<class T > NSVec4< T > operator* (const double & *pLHS,* const NSVec4< T > & *pRHS*)

7.28.1.21 template<class T > NSVec4< T > operator/ (const int32_t & *pLHS,* const NSVec4< T > & *pRHS*)

7.28.1.22 template<class T > NSVec4< T > operator/ (const float & *pLHS,* const NSVec4< T > & *pRHS*)

7.28.1.23 template<class T > NSVec4< T > operator/ (const double & *pLHS,* const NSVec4< T > & *pRHS*)

7.28.1.24 template<class PUPer , class T > void pup (PUPer & *p,* NSVec4< T > & *v4*)

7.28.1.25 template<class PUPer , class T > void pup (PUPer & *p,* NSVec4< T > & *v4,* const std::string & *varName*)

7.28.1.26 template<class T > NSVec4< T > round (const NSVec4< T > & *pVec*)

7.29 /home/dprandle/Documents/code/ctrlmod/src/edcallback.cpp File Reference

```
#include <edcallback.h>
#include <edtimer.h>
```

7.30 /home/dprandle/Documents/code/ctrlmod/src/edcomm_system.cpp File Reference

```
#include <edtimer.h>
#include <unistd.h>
#include <edmessage.h>
#include <edmessage_dispatch.h>
#include <edmctrl.h>
#include <edcomm_system.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <edutility.h>
#include <edglobal.h>
#include <string.h>
#include <errno.h>
#include <edsocket.h>
#include <sstream>
```

7.31 /home/dprandle/Documents/code/ctrlmod/src/edi2c.cpp File Reference

```
#include <edi2c.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#include <linux/i2c-dev.h>
#include <sys/ioctl.h>
#include <edutility.h>
```

7.32 /home/dprandle/Documents/code/ctrlmod/src/edimu_system.cpp File Reference

```
#include <edimu_system.h>
#include <edi2c.h>
#include <edutility.h>
```

7.33 /home/dprandle/Documents/code/ctrlmod/src/edlogging_system.cpp File Reference

```
#include <edlogging_system.h>
#include <edmctrl.h>
#include <edmessage_dispatch.h>
#include <edmessage.h>
#include <bitset>
```

7.34 /home/dprandle/Documents/code/ctrlmod/src/edmctrl.cpp File Reference

Master control file for the edison.

```
#include <sstream>
#include <edutility.h>
#include <iostream>
#include <edmctrl.h>
#include <edsystem.h>
#include <string>
#include <vector>
#include <edtimer.h>
#include <edmessage_dispatch.h>
```

7.34.1 Detailed Description

Master control file for the edison.

Author

Daniel <dprandle-CZ-17>

Date

Fri Jul 10 09:19:32 2015

7.35 /home/dprandle/Documents/code/ctrlmod/src/edmessage.cpp File Reference

```
#include <edmessage.h>
#include <edutility.h>
```

7.36 /home/dprandle/Documents/code/ctrlmod/src/edmessage_dispatch.cpp File Reference

```
#include <edmessage_dispatch.h>
#include <edmessage.h>
```

7.37 /home/dprandle/Documents/code/ctrlmod/src/ednavsystem.cpp File Reference

Definition file for navigation system.

```
#include <edutility.h>
#include <ednavsystem.h>
#include <edmctrl.h>
#include <edmessage_dispatch.h>
#include <edmessage.h>
#include <edtimer.h>
#include <edi2c.h>
```

7.37.1 Detailed Description

Definition file for navigation system.

Author

Daniel <dprandle-CZ-17>

Date

Fri Jul 10 10:44:40 2015

7.38 /home/dprandle/Documents/code/ctrlmod/src/edplsystem.cpp File Reference

Definitions for system.

```
#include <edplsystem.h>
#include <edutility.h>
#include <mraa.hpp>
#include <iostream>
#include <edtimer.h>
#include <vector>
#include <edmessage_dispatch.h>
#include <edmessage.h>
#include <edmctrl.h>
```

7.38.1 Detailed Description

Definitions for system.

Author

Daniel <dprandle-CZ-17>

Date

Tue Jul 7 09:32:32 2015

7.39 /home/dprandle/Documents/code/ctrlmod/src/edrplidar_packets.cpp File Reference

```
#include <edrplidar_packets.h>
#include <sstream>
#include <iomanip>
```

7.40 /home/dprandle/Documents/code/ctrlmod/src/edrplidar_system.cpp File Reference

```
#include <eduart.h>
#include <iostream>
#include <edmessage_dispatch.h>
#include <edrplidar_system.h>
#include <edutility.h>
#include <edmctrl.h>
#include <mraa/uart.hpp>
#include <edtimer.h>
#include <unistd.h>
#include <termios.h>
#include <mraa/mraa_internal_types.h>
```

7.41 /home/dprandle/Documents/code/ctrlmod/src/edsocket.cpp File Reference

```
#include <edsocket.h>
#include <unistd.h>
#include <sys/socket.h>
#include <errno.h>
```

7.42 /home/dprandle/Documents/code/ctrlmod/src/edthreaded_fd.cpp File Reference

```
#include <edthreaded_fd.h>
#include <unistd.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <edutility.h>
#include <errno.h>
#include <edtimer.h>
#include <edcallback.h>
```

7.43 /home/dprandle/Documents/code/ctrlmod/src/edtimer.cpp File Reference

```
#include <edutility.h>
#include <edtimer.h>
#include <iostream>
#include <string>
#include <edcallback.h>
```

7.44 /home/dprandle/Documents/code/ctrlmod/src/eduart.cpp File Reference

```
#include <string.h>
#include <edutility.h>
#include <eduart.h>
#include <fcntl.h>
#include <unistd.h>
#include <termios.h>
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
```

7.45 /home/dprandle/Documents/code/ctrlmod/src/edutility.cpp File Reference

```
#include <fstream>
#include <unistd.h>
#include <exception>
#include <stdexcept>
#include <edutility.h>
#include <ctime>
#include <iostream>
#include <sstream>
#include <iomanip>
#include <edglobal.h>
#include <edtimer.h>
```

Functions

- void delay (double ms)
- void cprint (const std::string &str)
- uint32_t hash_id (const std::string &strng)
- bool log_message (const std::string &msg, const std::string &fname, bool tmstmp)
- std::string timestamp ()
- std::string to_hex (uint8_t byte)
- std::string to_hex (int16_t two_bytes)
- std::string to_hex (uint16_t two_bytes)
- std::string to_hex (int32_t four_bytes)
- std::string to_hex (uint32_t four_bytes)
- void zero_buf (uint8_t *buf, uint32_t size)
- void copy_buf (const uint8_t *src, uint8_t *dest, uint32_t size, uint32_t src_offset, uint32_t dest_offset)

  Copy buffer.

7.45.1 Function Documentation

7.45.1.1 void copy_buf (const uint8_t * *src*, uint8_t * *dest*, uint32_t *size*, uint32_t *src_offset* = 0, uint32_t *dest_offset* = 0)

Copy buffer.

Copy the source buffer to destination buffer with possible offsets in each buffer

Parameters

src	The source buffer
dest	The destination buffer
size	Amount of items to copy
src_offset	Offset in to the source buffer (defaults to 0)
dest_offset	Offset in to the destination buffer (defaults to 0)

7.45.1.2 void cprint (const std::string & *str*)

7.45.1.3 void delay (double *ms*)

7.45.1.4 uint32_t hash_id (const std::string & *strng*)

7.45.1.5 bool log_message (const std::string & *msg*, const std::string & *fname*, bool *tmstmp*)

7.45.1.6 std::string timestamp ( )

7.45.1.7 std::string to_hex (uint8_t *byte*)

7.45.1.8 std::string to_hex (int16_t *two_bytes*)

7.45.1.9 std::string to_hex (uint16_t *two_bytes*)

7.45.1.10 std::string to_hex (int32_t *four_bytes*)

7.45.1.11 std::string to_hex (uint32_t *four_bytes*)

7.45.1.12 void zero_buf (uint8_t * *buf*, uint32_t *size*)

7.46 /home/dprandle/Documents/code/ctrlmod/src/main.cpp File Reference

```
#include <edrplidar_system.h>
#include <edmctrl.h>
#include <edplsystem.h>
#include <stdlib.h>
#include <signal.h>
#include <ednavsystem.h>
#include <edmessage_dispatch.h>
#include <edlogging_system.h>
#include <edmessage.h>
#include <edtimer.h>
#include <edcallback.h>
#include <edcomm_system.h>
#include <edimu_system.h>
```

Functions

- void handle_ctrlc (int32_t sig)
- int32_t main (int32_t argc, char ∗argv[])

7.46.1 Function Documentation

7.46.1.1 void handle_ctrlc (int32_t *sig*)

7.46.1.2 int32_t main (int32_t *argc,* char ∗ *argv[]*)

7.47 /home/dprandle/Documents/code/ctrlmod/src/nsmath.cpp File Reference

```
#include <nsmath.h>
```

Macros

- #define FLOAT_EPS 0.00001

Functions

- float clampf (float pVal, const float &pMin, const float &pMax)
- double clamp (double pVal, const double &pMin, const double &pMax)
- float fractf (const float &num)
- double fract (const double &num)
- float lerp (float low, float high, float middle)
- double lerp (double low, double high, double middle)
- float lerp (int32_t low, int32_t high, int32_t middle)
- float lerp (uint32_t low, uint32_t high, uint32_t middle)
- float random_float (float pHigh, float pLow)

7.47.1 Macro Definition Documentation

7.47.1.1 #define FLOAT_EPS 0.00001

7.47.2 Function Documentation

7.47.2.1 double clamp (double *pVal,* const double & *pMin,* const double & *pMax*)

7.47.2.2 float clampf (float *pVal,* const float & *pMin,* const float & *pMax*)

7.47.2.3 double fract (const double & *num*)

7.47.2.4 float fractf (const float & *num*)

7.47.2.5 float lerp (float *low,* float *high,* float *middle*)

7.47.2.6 double lerp (double *low,* double *high,* double *middle*)

7.47.2.7 float lerp (int32_t *low,* int32_t *high,* int32_t *middle*)

7.47.2.8 float lerp (uint32_t *low,* uint32_t *high,* uint32_t *middle*)

7.47.2.9 float random_float (float *pHigh,* float *pLow*)