

System library

1.00

Generated on Fri Nov 29 2024 for System library by Doxygen 1.12.0

Fri Nov 29 2024 22:16:29

1 System Library	1
2 Topic Index	3
2.1 Topics	3
3 Namespace Index	5
3.1 Namespace List	5
4 Class Index	7
4.1 Class List	7
5 Topic Documentation	9
5.1 System	9
5.1.1 Detailed Description	9
5.1.2 Version	10
5.1.2.1 Detailed Description	10
5.1.2.2 Variable Documentation	10
5.1.3 Produce Date	11
5.1.3.1 Detailed Description	11
5.1.3.2 Variable Documentation	11
6 Namespace Documentation	13
6.1 Lib Namespace Reference	13
6.2 Lib::System Namespace Reference	13
6.2.1 Detailed Description	13
7 Class Documentation	15
7.1 Lib::System::Configuration Class Reference	15
7.1.1 Detailed Description	15
7.1.2 Constructor & Destructor Documentation	16
7.1.2.1 Configuration()	16
7.1.2.2 ~Configuration()	16
7.1.3 Member Function Documentation	16
7.1.3.1 isEmpty()	16
7.1.3.2 operator uint32_t()	16
7.1.3.3 operator=()	17
7.2 Lib::System::Version::Firmware Class Reference	18
7.2.1 Detailed Description	18
7.2.2 Constructor & Destructor Documentation	21
7.2.2.1 Firmware()	21
7.2.2.2 ~Firmware()	21
7.2.3 Member Function Documentation	21
7.2.3.1 getBuildDate()	21
7.2.3.2 operator std::string()	22

7.2.3.3 operator uint32_t()	22
7.2.3.4 stringToDateTime()	22
7.2.3.5 stringToU32()	23
7.2.3.6 subVersion()	23
7.2.3.7 toString()	23
7.3 Lib::System::Version::Hardware Class Reference	24
7.3.1 Detailed Description	24
7.3.2 Constructor & Destructor Documentation	26
7.3.2.1 Hardware()	26
7.3.3 Member Function Documentation	26
7.3.3.1 operator std::string()	26
7.3.3.2 operator uint32_t()	26
7.3.3.3 stringToDateTime()	26
7.3.3.4 stringToU32()	27
7.3.3.5 subVersion()	27
7.3.3.6 toString()	27
7.4 Lib::System::Parameters< T > Class Template Reference	28
7.4.1 Detailed Description	28
7.4.2 Constructor & Destructor Documentation	29
7.4.2.1 Parameters()	29
7.4.2.2 ~Parameters()	29
7.4.3 Member Function Documentation	29
7.4.3.1 getData()	29
7.4.3.2 restoreByDefault()	29
7.4.3.3 setup()	30
7.4.3.4 write()	30
7.5 Lib::System::ProduceDate Class Reference	31
7.5.1 Detailed Description	31
7.5.2 Member Typedef Documentation	32
7.5.2.1 DateTime	32
7.5.3 Constructor & Destructor Documentation	32
7.5.3.1 ProduceDate()	32
7.5.3.2 ~ProduceDate()	32
7.5.4 Member Function Documentation	32
7.5.4.1 isEmpty()	32
7.5.4.2 read()	33
7.5.4.3 write()	34
7.6 Lib::System::SerialNumber Class Reference	34
7.6.1 Detailed Description	34
7.6.2 Constructor & Destructor Documentation	35
7.6.2.1 SerialNumber()	35
7.6.2.2 ~SerialNumber()	35

7.6.3 Member Function Documentation	35
7.6.3.1 operator std::string()	35
7.6.3.2 operator uint32_t()	36
7.6.3.3 toString()	36
7.7 Lib::System::Parameters< T >::StorageBase Class Reference	37
7.7.1 Detailed Description	37
7.7.2 Constructor & Destructor Documentation	38
7.7.2.1 StorageBase()	38
7.7.2.2 ~StorageBase()	39
7.7.3 Member Function Documentation	39
7.7.3.1 close()	39
7.7.3.2 open()	39
7.7.3.3 read()	40
7.7.3.4 write()	40
7.7.4 Friends And Related Symbol Documentation	40
7.7.4.1 Parameters	40
7.8 Lib::System::System Class Reference	41
7.8.1 Detailed Description	41
7.8.2 Member Typedef Documentation	42
7.8.2.1 LastResetState	42
7.8.3 Constructor & Destructor Documentation	42
7.8.3.1 System()	42
7.8.3.2 ~System()	42
7.8.4 Member Function Documentation	43
7.8.4.1 restart()	43
7.8.5 Member Data Documentation	43
7.8.5.1 configuration	43
7.8.5.2 lastResetState	43
7.8.5.3 serialNumber	43
7.9 Lib::System::Version Class Reference	43
7.9.1 Detailed Description	43
Index	45

Chapter 1

System Library

namespace : `Lib::System`

Chapter 2

Topic Index

2.1 Topics

Here is a list of all topics with brief descriptions:

System	9
Version	10
Produce Date	11

Chapter 3

Namespace Index

3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Lib	13
Lib::System	
System library	13

Chapter 4

Class Index

4.1 Class List

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

Lib::System::Configuration	
System configuration class	15
Lib::System::Version::Firmware	
Firmware Version	18
Lib::System::Version::Hardware	
Hardware Version	24
Lib::System::Parameters< T >	
System parameters class	28
Lib::System::ProduceDate	
Produce Date class	31
Lib::System::SerialNumber	
Serial Nubmer class	34
Lib::System::Parameters< T >::StorageBase	
StorageBase class	37
Lib::System::System	
System class	41
Lib::System::Version	
System Version	43

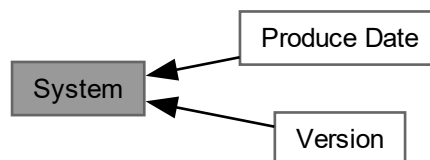
Chapter 5

Topic Documentation

5.1 System

5.1.1 Detailed Description

Collaboration diagram for System:



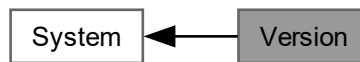
Topics

- [Version](#)
- [Produce Date](#)

5.1.2 Version

5.1.2.1 Detailed Description

Collaboration diagram for Version:



Classes

- class [Lib::System::Version::Hardware](#)
Hardware Version.
- class [Lib::System::Version::Firmware](#)
Firmware Version.

Variables

- [Version::Firmware Lib::System::System::firmwareVersion](#)
Instance of the Firmware version.
- [Version::Hardware Lib::System::System::hardwareVersion](#)
Instance of the Hardware version.

5.1.2.2 Variable Documentation

5.1.2.2.1 firmwareVersion

[Version::Firmware](#) `Lib::System::System::firmwareVersion`

Instance of the Firmware version.

5.1.2.2.2 hardwareVersion

[Version::Hardware](#) `Lib::System::System::hardwareVersion`

Instance of the Hardware version.

5.1.3 Produce Date

5.1.3.1 Detailed Description

Collaboration diagram for Produce Date:



Classes

- class [Lib::System::ProduceDate](#)
Produce Date class.

Variables

- [ProduceDate Lib::System::System::produceDate](#)
Instance of the Produce date.
- [ProduceDate Lib::System::System::saleDate](#)
Instance of the Sale date.

5.1.3.2 Variable Documentation

5.1.3.2.1 produceDate

[ProduceDate](#) Lib::System::System::produceDate

Instance of the Produce date.

5.1.3.2.2 saleDate

[ProduceDate](#) Lib::System::System::saleDate

Instance of the Sale date.

Chapter 6

Namespace Documentation

6.1 Lib Namespace Reference

Namespaces

- namespace [System](#)
System library.

6.2 Lib::System Namespace Reference

6.2.1 Detailed Description

System library.

Classes

- class [Configuration](#)
System configuration class.
- class [Parameters](#)
System parameters class.
- class [ProduceDate](#)
Produce Date class.
- class [SerialNumber](#)
Serial Nubmer class.
- class [System](#)
System class.
- class [Version](#)
System Version.

Chapter 7

Class Documentation

7.1 Lib::System::Configuration Class Reference

7.1.1 Detailed Description

System configuration class.

This class stores a configuration word for the system. It defines some constant parameters that the system maintains.

Collaboration diagram for Lib::System::Configuration:

Lib::System::Configuration	
+	Configuration()
+	~Configuration()
+	operator uint32_t()
+	operator=()
+	isEmpty()

Public Member Functions

- [Configuration](#) (uint32_t offset)
Configuration ctor.
- virtual [~Configuration](#) ()=default
Configuration dtor.
- [operator uint32_t](#) () const
Reads the value from OTM Rom.
- void [operator=](#) (uint32_t value)
Writes value to OTP Rom.
- virtual bool [isEmpty](#) () const
Checks if the field in the OTP Flash memory where the date should be empty.

7.1.2 Constructor & Destructor Documentation

7.1.2.1 Configuration()

```
Lib::System::Configuration::Configuration (
    uint32_t offset) [inline]
```

Configuration ctor.

Parameters

<i>offset</i>	- offset in OTP flash memory
---------------	------------------------------

7.1.2.2 ~Configuration()

```
virtual Lib::System::Configuration::~~Configuration () [virtual], [default]
```

Configuration dtor.

7.1.3 Member Function Documentation

7.1.3.1 isEmpty()

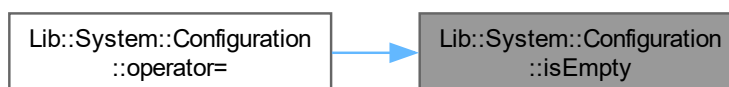
```
virtual bool Lib::System::Configuration::isEmpty () const [inline], [virtual]
```

Checks if the field in the OTP Flash memory where the date should be empty.

Returns

bool - is empty

Here is the caller graph for this function:



7.1.3.2 operator uint32_t()

```
Lib::System::Configuration::operator uint32_t () const [inline]
```

Reads the value from OTM Rom.

Returns

value

7.1.3.3 operator=()

```
void Lib::System::Configuration::operator= (  
    uint32_t value) [inline]
```

Writes value to OTP Rom.

Parameters

<i>value</i>	-value
--------------	--------

Returns

returns `true` on successful save

Here is the call graph for this function:

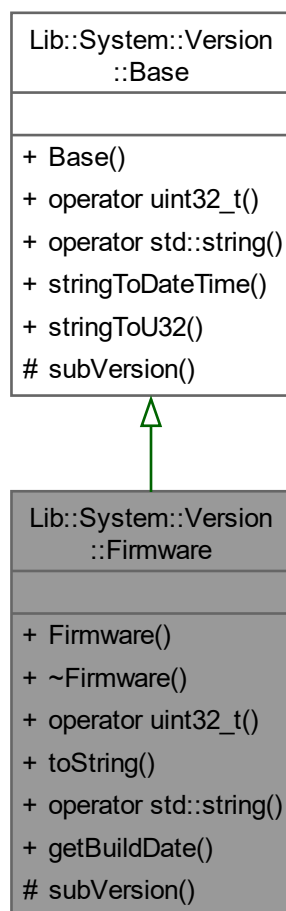


7.2 Lib::System::Version::Firmware Class Reference

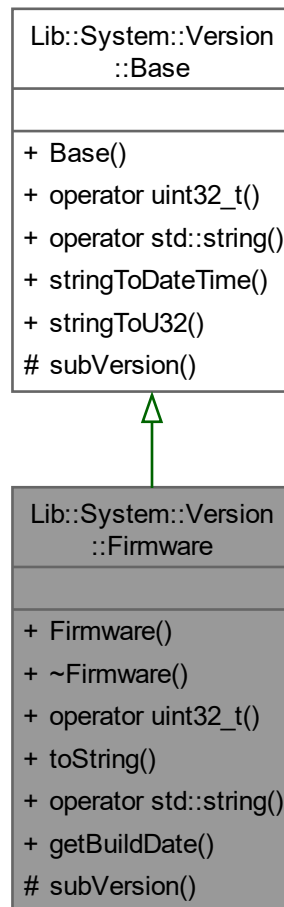
7.2.1 Detailed Description

Firmware Version.

Inheritance diagram for Lib::System::Version::Firmware:



Collaboration diagram for Lib::System::Version::Firmware:



Public Member Functions

- `Firmware` (const std::string &string, uint32_t subVersion)
Firmware ctor.
- `~Firmware` ()=default
Firmware dtor.
- virtual `operator uint32_t` () const
Gets the firmware version.
- virtual std::string `toString` () const
Converts version to string.
- virtual `operator std::string` () const
Type conversion operator.
- virtual const std::string `getBuildDate` () const
Gets the build date.

Protected Member Functions

- virtual uint32_t [subVersion](#) () const override
Gets a sub version.

Static Protected Member Functions

- static const Lib::Helper::DateTime [stringToDateTime](#) (const char *timeStr, const char *dateStr)
Converts the Date/Time Strings To DateTime.
- static uint32_t [stringToU32](#) (const std::string &[string](#))
Converts the version string to uint32_t.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 Firmware()

```
Lib::System::Version::Firmware::Firmware (
    const std::string & string,
    uint32_t subVersion) [inline]
```

Firmware ctor.

Parameters

<i>string</i>	
---------------	--

7.2.2.2 ~Firmware()

```
Lib::System::Version::Firmware::~~Firmware () [default]
```

Firmware dtor.

7.2.3 Member Function Documentation

7.2.3.1 getBuildDate()

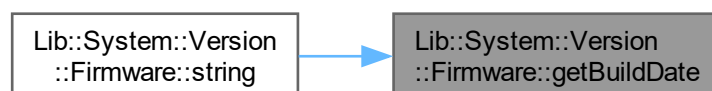
```
virtual const std::string Lib::System::Version::Firmware::getBuildDate () const [inline],
[virtual]
```

Gets the build date.

Returns

returns a string of the compilation date

Here is the caller graph for this function:



7.2.3.2 operator std::string()

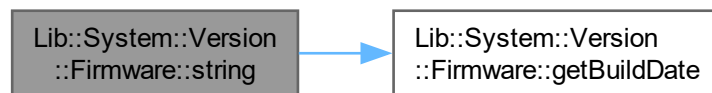
```
virtual Lib::System::Version::Firmware::operator std::string () const [inline], [virtual]
```

Type conversion operator.

Returns

std::string

Here is the call graph for this function:



7.2.3.3 operator uint32_t()

```
virtual Lib::System::Version::Firmware::operator uint32_t () const [inline], [virtual]
```

Gets the firmware version.

Returns

uint32_t

7.2.3.4 stringToDateTime()

```
static const Lib::Helper::DateTime Lib::System::Version::Base::stringToDateTime (
    const char * timeStr,
    const char * dateStr) [inline], [static], [inherited]
```

Converts the Date/Time Strings To DateTime.

Parameters

<i>timeStr</i>	
<i>dateStr</i>	

Returns

DateTime

7.2.3.5 stringToU32()

```
static uint32_t Lib::System::Version::Base::stringToU32 (  
    const std::string & string) [inline], [static], [inherited]
```

Converts the version string to uint32_t.

Returns

uint32_t

Note

format of version string must be x.xx

7.2.3.6 subVersion()

```
virtual uint32_t Lib::System::Version::Firmware::subVersion () const [inline], [override],  
[protected], [virtual]
```

Gets a sub version.

Returns

7.2.3.7 toString()

```
virtual std::string Lib::System::Version::Firmware::toString () const [inline], [virtual]
```

Converts version to string.

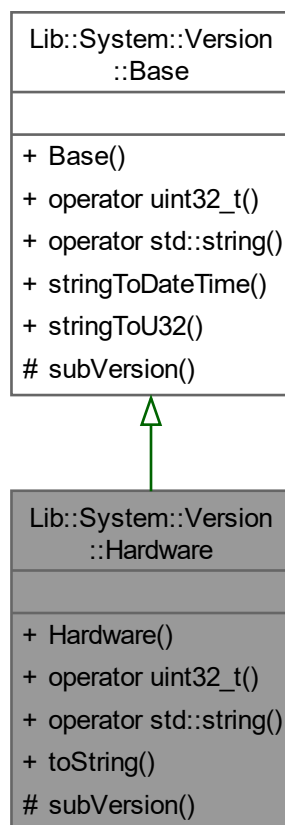
Returns

7.3 Lib::System::Version::Hardware Class Reference

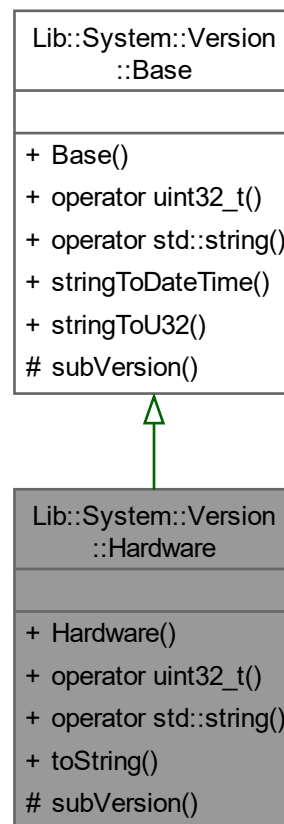
7.3.1 Detailed Description

Hardware Version.

Inheritance diagram for Lib::System::Version::Hardware:



Collaboration diagram for Lib::System::Version::Hardware:



Public Member Functions

- [Hardware](#) (const std::string &[string](#), uint32_t [subVersion](#))
Hardware version ctor.
- virtual [operator uint32_t](#) () const override
Get Hardware Version.
- virtual [operator std::string](#) () const override
Get The Version String.
- virtual std::string [toString](#) () const
Convert version to string.

Protected Member Functions

- virtual uint32_t [subVersion](#) () const override

Static Protected Member Functions

- static const Lib::Helper::DateTime [stringToDateTime](#) (const char *timeStr, const char *dateStr)
Converts the Date/Time Strings To DateTime.
- static uint32_t [stringToU32](#) (const std::string &string)
Converts the version string to uint32_t.

7.3.2 Constructor & Destructor Documentation

7.3.2.1 Hardware()

```
Lib::System::Version::Hardware::Hardware (
    const std::string & string,
    uint32_t subVersion) [inline]
```

Hardware version ctor.

Parameters

string	
------------------------	--

7.3.3 Member Function Documentation

7.3.3.1 operator std::string()

```
virtual Lib::System::Version::Hardware::operator std::string () const [inline], [override],
[virtual]
```

Get The Version String.

Returns

std::string

7.3.3.2 operator uint32_t()

```
virtual Lib::System::Version::Hardware::operator uint32_t () const [inline], [override],
[virtual]
```

Get Hardware Version.

Returns

uint32_t

7.3.3.3 stringToDateTime()

```
static const Lib::Helper::DateTime Lib::System::Version::Base::stringToDateTime (
    const char * timeStr,
    const char * dateStr) [inline], [static], [inherited]
```

Converts the Date/Time Strings To DateTime.

Parameters

<i>timeStr</i>	
<i>dateStr</i>	

Returns

DateTime

7.3.3.4 stringToU32()

```
static uint32_t Lib::System::Version::Base::stringToU32 (  
    const std::string & string)    [inline], [static], [inherited]
```

Converts the version string to uint32_t.

Returns

uint32_t

Note

format of version string must be x.xx

7.3.3.5 subVersion()

```
virtual uint32_t Lib::System::Version::Hardware::subVersion () const    [inline], [override],  
[protected], [virtual]
```

7.3.3.6 toString()

```
virtual std::string Lib::System::Version::Hardware::toString () const    [inline], [virtual]
```

Convert version to string.

Returns

7.4 Lib::System::Parameters< T > Class Template Reference

7.4.1 Detailed Description

```
template<class T>
class Lib::System::Parameters< T >
```

System parameters class.

See also

Example:

```
System::Parametes
System::ParametesData
```

Collaboration diagram for Lib::System::Parameters< T >:

Lib::System::Parameters< T >	
#	Parameters()
#	~Parameters()
#	getData()
#	setup()
#	write()
#	restoreByDefault()

Classes

- class [StorageBase](#)
StorageBase class.

Protected Member Functions

- [Parameters](#) ([StorageBase](#) *storage)
Parameters ctor.
- [~Parameters](#) ()=default
Parameters dtor.
- virtual T & [getData](#) ()
Get the reference of data.
- virtual bool [setup](#) ()
Setup the storage.
- virtual bool [write](#) ()
Write the system parameters data.
- virtual void [restoreByDefault](#) ()=0

7.4.2 Constructor & Destructor Documentation

7.4.2.1 Parameters()

```
template<class T >
Lib::System::Parameters< T >::Parameters (
    StorageBase * storage) [inline], [protected]
```

Parameters ctor.

Parameters

<i>storage</i>	- pointer to storage instance
----------------	-------------------------------

7.4.2.2 ~Parameters()

```
template<class T >
Lib::System::Parameters< T >::~~Parameters () [protected], [default]
```

Parameters dtor.

7.4.3 Member Function Documentation

7.4.3.1 getData()

```
template<class T >
virtual T & Lib::System::Parameters< T >::getData () [inline], [protected], [virtual]
```

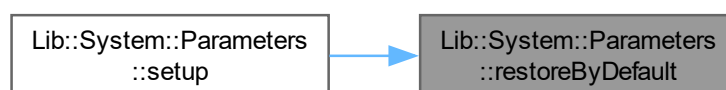
Get the reference of data.

Returns

7.4.3.2 restoreByDefault()

```
template<class T >
virtual void Lib::System::Parameters< T >::restoreByDefault () [protected], [pure virtual]
```

Here is the caller graph for this function:



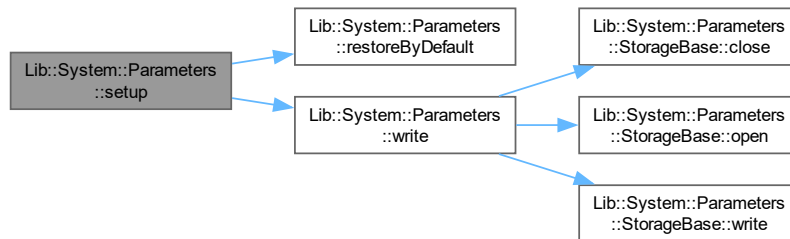
7.4.3.3 setup()

```
template<class T >
virtual bool Lib::System::Parameters< T >::setup () [inline], [protected], [virtual]
```

Setup the storage.

Returns

Here is the call graph for this function:



7.4.3.4 write()

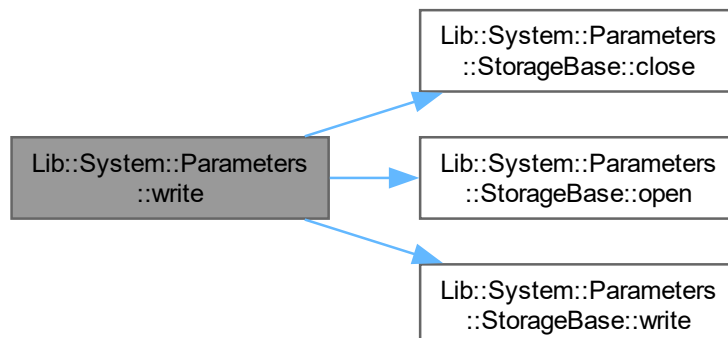
```
template<class T >
virtual bool Lib::System::Parameters< T >::write () [inline], [protected], [virtual]
```

Write the system parameters data.

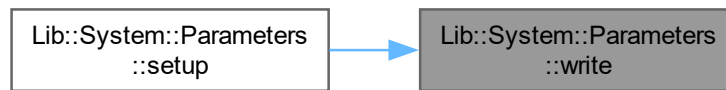
Returns

true if success

Here is the call graph for this function:



Here is the caller graph for this function:

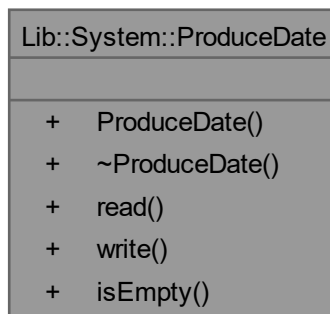


7.5 Lib::System::ProduceDate Class Reference

7.5.1 Detailed Description

Produce Date class.

Collaboration diagram for Lib::System::ProduceDate:



Public Types

- using [DateTime](#) = Lib::Helper::DateTime

Public Member Functions

- [ProduceDate](#) (uint32_t offset)
ProduceDate ctor.
- virtual [~ProduceDate](#) ()=default
Produce Date dtor.
- virtual bool [read](#) ([DateTime](#) &result) const
Read the Produce Date.
- virtual bool [write](#) (const [DateTime](#) &value)
Write Produce Date in OTP ROM.
- virtual bool [isEmpty](#) () const
Checks if the field in the OTP Flash memory where the date should be empty.

7.5.2 Member Typedef Documentation

7.5.2.1 DateTime

```
using Lib::System::ProduceDate::DateTime = Lib::Helper::DateTime
```

7.5.3 Constructor & Destructor Documentation

7.5.3.1 ProduceDate()

```
Lib::System::ProduceDate::ProduceDate (
    uint32_t offset) [inline]
```

ProduceDate ctor.

Parameters

<i>offset</i>	- offset in OTP Flash Memory
---------------	------------------------------

7.5.3.2 ~ProduceDate()

```
virtual Lib::System::ProduceDate::~~ProduceDate () [virtual], [default]
```

Produce Date dtor.

7.5.4 Member Function Documentation

7.5.4.1 isEmpty()

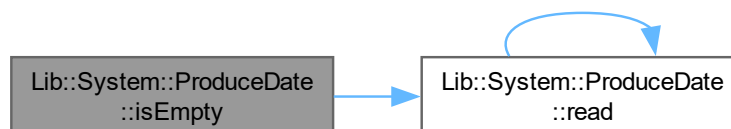
```
virtual bool Lib::System::ProduceDate::isEmpty () const [inline], [virtual]
```

Checks if the field in the OTP Flash memory where the date should be empty.

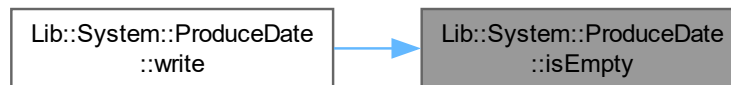
Returns

bool - is empty

Here is the call graph for this function:



Here is the caller graph for this function:



7.5.4.2 read()

```
virtual bool Lib::System::ProduceDate::read (
    DateTime & result) const [inline], [virtual]
```

Read the Produce Date.

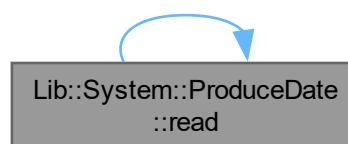
Parameters

<i>result</i>	reference to result
---------------	---------------------

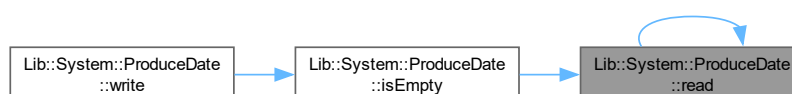
Returns

bool

Here is the call graph for this function:



Here is the caller graph for this function:



7.5.4.3 write()

```
virtual bool Lib::System::ProduceDate::write (
    const DateTime & value) [inline], [virtual]
```

Write Produce Date in OTP ROM.

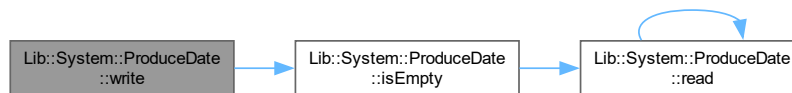
Parameters

<i>value</i>	- reference to DateTime instance
--------------	----------------------------------

Returns

bool - truth in a successful record

Here is the call graph for this function:

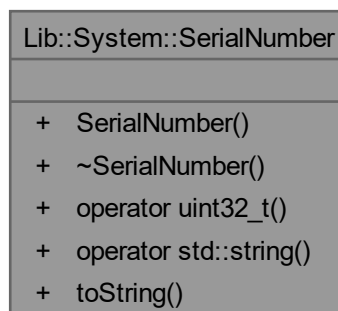


7.6 Lib::System::SerialNumber Class Reference

7.6.1 Detailed Description

Serial Nubmer class.

Collaboration diagram for Lib::System::SerialNumber:



Public Member Functions

- [SerialNumber](#) ()
Serial Number ctor.
- virtual [~SerialNumber](#) ()=default
Serial Number dtor.
- virtual [operator uint32_t](#) () const
Type Conversion operator.
- virtual [operator std::string](#) () const
Type Conversion operator.
- virtual std::string [toString](#) () const
get Serial Number as string

7.6.2 Constructor & Destructor Documentation

7.6.2.1 SerialNumber()

```
Lib::System::SerialNumber::SerialNumber () [inline]
```

Serial Number ctor.

7.6.2.2 ~SerialNumber()

```
virtual Lib::System::SerialNumber::~~SerialNumber () [virtual], [default]
```

Serial Number dtor.

7.6.3 Member Function Documentation

7.6.3.1 operator std::string()

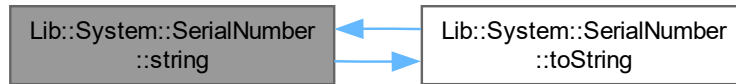
```
virtual Lib::System::SerialNumber::operator std::string () const [inline], [virtual]
```

Type Conversion operator.

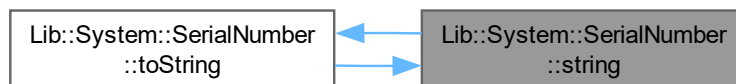
Returns

uint32_t

Here is the call graph for this function:



Here is the caller graph for this function:

**7.6.3.2 operator uint32_t()**

```
virtual Lib::System::SerialNumber::operator uint32_t () const [inline], [virtual]
```

Type Conversion operator.

Returns

uint32_t

7.6.3.3 toString()

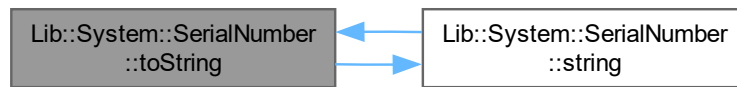
```
virtual std::string Lib::System::SerialNumber::toString () const [inline], [virtual]
```

get Serial Number as string

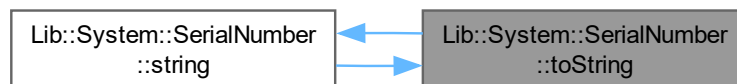
Returns

const std::string

Here is the call graph for this function:



Here is the caller graph for this function:



7.7 Lib::System::Parameters< T >::StorageBase Class Reference

7.7.1 Detailed Description

```
template<class T>
class Lib::System::Parameters< T >::StorageBase
```

StorageBase class.

Collaboration diagram for Lib::System::Parameters< T >::StorageBase:

Lib::System::Parameters < T >::StorageBase
<pre># StorageBase() # ~StorageBase() # open() # close() # read() # write()</pre>

Protected Member Functions

- [StorageBase](#) ()=default
Storage Base ctor.
- virtual [~StorageBase](#) ()=default
- virtual bool [open](#) ()=0
Open the _storage.
- virtual bool [close](#) ()=0
Close the _storage.
- virtual bool [read](#) (void *dst, size_t len)=0
Read the parameters data.
- virtual bool [write](#) (void *src, size_t len)=0
Write the parameters data.

Friends

- class [Parameters](#)

7.7.2 Constructor & Destructor Documentation

7.7.2.1 StorageBase()

```
template<class T >
Lib::System::Parameters< T >::StorageBase::StorageBase () [protected], [default]
```

Storage Base ctor.

7.7.2.2 ~StorageBase()

```
template<class T >
virtual Lib::System::Parameters< T >::StorageBase::~StorageBase () [protected], [virtual],
[default]
```

7.7.3 Member Function Documentation

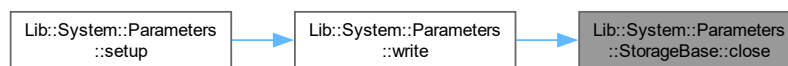
7.7.3.1 close()

```
template<class T >
virtual bool Lib::System::Parameters< T >::StorageBase::close () [protected], [pure virtual]
```

Close the _storage.

Returns

Here is the caller graph for this function:



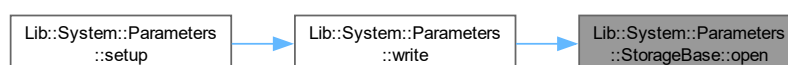
7.7.3.2 open()

```
template<class T >
virtual bool Lib::System::Parameters< T >::StorageBase::open () [protected], [pure virtual]
```

Open the _storage.

Returns

Here is the caller graph for this function:



7.7.3.3 read()

```
template<class T >
virtual bool Lib::System::Parameters< T >::StorageBase::read (
    void * dst,
    size_t len) [protected], [pure virtual]
```

Read the parameters data.

Returns

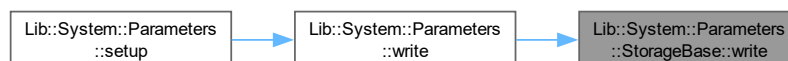
7.7.3.4 write()

```
template<class T >
virtual bool Lib::System::Parameters< T >::StorageBase::write (
    void * src,
    size_t len) [protected], [pure virtual]
```

Write the parameters data.

Returns

Here is the caller graph for this function:



7.7.4 Friends And Related Symbol Documentation

7.7.4.1 Parameters

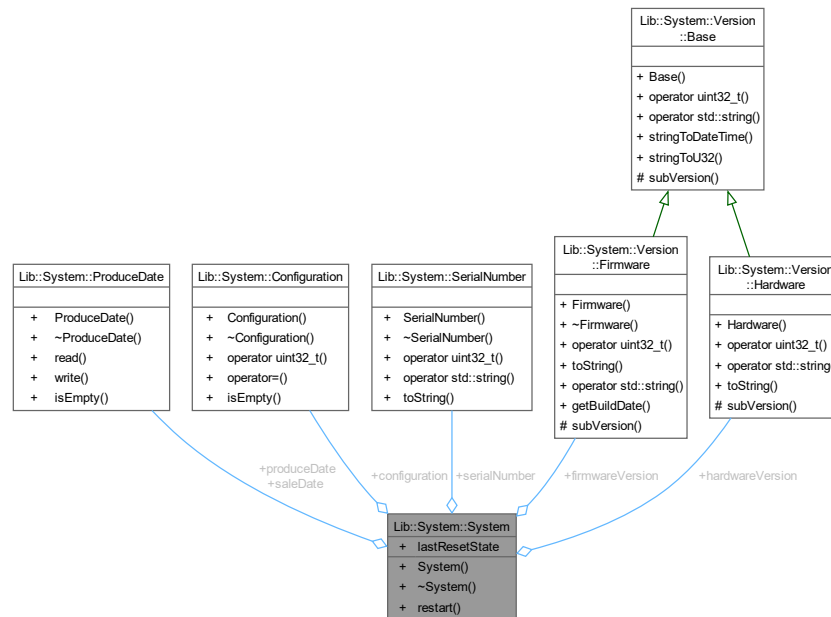
```
template<class T >
friend class Parameters [friend]
```

7.8 Lib::System::System Class Reference

7.8.1 Detailed Description

System class.

Collaboration diagram for Lib::System::System:



Public Types

- using `LastResetState` = `Lib::HAL::LastResetState::State`

Public Member Functions

- `System` (`const std::string &fwVersionString`, `uint32_t fwSubversion`, `const std::string &hwVersionString`, `uint32_t hwSubversion`)
System ctor.
- `~System` ()=default
System dtor.
- void `restart` ()
Restarts the system.

Public Attributes

- [LastResetState lastResetState](#)
Instance of the Last reset state.
- [ProduceDate produceDate](#)
Instance of the Produce date.
- [ProduceDate saleDate](#)
Instance of the Sale date.
- [Configuration configuration](#)
Instance of the System configuration word.
- [SerialNumber serialNumber](#)
Instance of the system serial number.
- [Version::Firmware firmwareVersion](#)
Instance of the Firmware version.
- [Version::Hardware hardwareVersion](#)
Instance of the Hardware version.

7.8.2 Member Typedef Documentation

7.8.2.1 LastResetState

```
using Lib::System::System::LastResetState = Lib::HAL::LastResetState::State
```

7.8.3 Constructor & Destructor Documentation

7.8.3.1 System()

```
Lib::System::System::System (
    const std::string & fwVersionString,
    uint32_t fwSubversion,
    const std::string & hwVersionString,
    uint32_t hwSubversion) [inline]
```

System ctor.

Parameters

<i>fwVersionString</i>	Firmware version string
<i>fwSubversion</i>	Firmware sub-version
<i>hwVersionString</i>	Hardware version string
<i>hwSubversion</i>	Hardware sub-version

7.8.3.2 ~System()

```
Lib::System::System::~~System () [default]
```

System dtor.

7.8.4 Member Function Documentation

7.8.4.1 restart()

```
void Lib::System::System::restart () [inline]
```

Restarts the system.

7.8.5 Member Data Documentation

7.8.5.1 configuration

```
Configuration Lib::System::System::configuration
```

Instance of the System configuration word.

7.8.5.2 lastResetState

```
LastResetState Lib::System::System::lastResetState
```

Instance of the Last reset state.

7.8.5.3 serialNumber

```
SerialNumber Lib::System::System::serialNumber
```

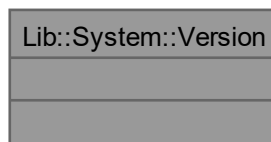
Instance of the system serial number.

7.9 Lib::System::Version Class Reference

7.9.1 Detailed Description

System Version.

Collaboration diagram for Lib::System::Version:



Classes

- class [Firmware](#)
Firmware Version.
- class [Hardware](#)
Hardware Version.

Index

- ~Configuration
 - Lib::System::Configuration, [16](#)
- ~Firmware
 - Lib::System::Version::Firmware, [21](#)
- ~Parameters
 - Lib::System::Parameters< T >, [29](#)
- ~ProduceDate
 - Lib::System::ProduceDate, [32](#)
- ~SerialNumber
 - Lib::System::SerialNumber, [35](#)
- ~StorageBase
 - Lib::System::Parameters< T >::StorageBase, [38](#)
- ~System
 - Lib::System::System, [42](#)
- close
 - Lib::System::Parameters< T >::StorageBase, [39](#)
- Configuration
 - Lib::System::Configuration, [16](#)
- configuration
 - Lib::System::System, [43](#)
- DateTime
 - Lib::System::ProduceDate, [32](#)
- Firmware
 - Lib::System::Version::Firmware, [21](#)
- firmwareVersion
 - Version, [10](#)
- getBuildDate
 - Lib::System::Version::Firmware, [21](#)
- getData
 - Lib::System::Parameters< T >, [29](#)
- Hardware
 - Lib::System::Version::Hardware, [26](#)
- hardwareVersion
 - Version, [10](#)
- isEmpty
 - Lib::System::Configuration, [16](#)
 - Lib::System::ProduceDate, [32](#)
- LastResetState
 - Lib::System::System, [42](#)
- lastResetState
 - Lib::System::System, [43](#)
- Lib, [13](#)
- Lib::System, [13](#)
- Lib::System::Configuration, [15](#)
- ~Configuration, [16](#)
- Configuration, [16](#)
- isEmpty, [16](#)
- operator uint32_t, [16](#)
- operator=, [16](#)
- Lib::System::Parameters< T >, [28](#)
- ~Parameters, [29](#)
- getData, [29](#)
- Parameters, [29](#)
- restoreByDefault, [29](#)
- setup, [29](#)
- write, [30](#)
- Lib::System::Parameters< T >::StorageBase, [37](#)
- ~StorageBase, [38](#)
- close, [39](#)
- open, [39](#)
- Parameters, [40](#)
- read, [39](#)
- StorageBase, [38](#)
- write, [40](#)
- Lib::System::ProduceDate, [31](#)
- ~ProduceDate, [32](#)
- DateTime, [32](#)
- isEmpty, [32](#)
- ProduceDate, [32](#)
- read, [33](#)
- write, [33](#)
- Lib::System::SerialNumber, [34](#)
- ~SerialNumber, [35](#)
- operator std::string, [35](#)
- operator uint32_t, [36](#)
- SerialNumber, [35](#)
- toString, [36](#)
- Lib::System::System, [41](#)
- ~System, [42](#)
- configuration, [43](#)
- LastResetState, [42](#)
- lastResetState, [43](#)
- restart, [43](#)
- serialNumber, [43](#)
- System, [42](#)
- Lib::System::Version, [43](#)
- Lib::System::Version::Firmware, [18](#)
- ~Firmware, [21](#)
- Firmware, [21](#)
- getBuildDate, [21](#)
- operator std::string, [21](#)
- operator uint32_t, [22](#)
- stringToDateTime, [22](#)

- stringToU32, [22](#)
- subVersion, [23](#)
- toString, [23](#)
- Lib::System::Version::Hardware, [24](#)
 - Hardware, [26](#)
 - operator std::string, [26](#)
 - operator uint32_t, [26](#)
 - stringToDateTime, [26](#)
 - stringToU32, [27](#)
 - subVersion, [27](#)
 - toString, [27](#)
- open
 - Lib::System::Parameters< T >::StorageBase, [39](#)
- operator std::string
 - Lib::System::SerialNumber, [35](#)
 - Lib::System::Version::Firmware, [21](#)
 - Lib::System::Version::Hardware, [26](#)
- operator uint32_t
 - Lib::System::Configuration, [16](#)
 - Lib::System::SerialNumber, [36](#)
 - Lib::System::Version::Firmware, [22](#)
 - Lib::System::Version::Hardware, [26](#)
- operator=
 - Lib::System::Configuration, [16](#)
- Parameters
 - Lib::System::Parameters< T >, [29](#)
 - Lib::System::Parameters< T >::StorageBase, [40](#)
- Produce Date, [11](#)
 - produceDate, [11](#)
 - saleDate, [11](#)
- ProduceDate
 - Lib::System::ProduceDate, [32](#)
- produceDate
 - Produce Date, [11](#)
- read
 - Lib::System::Parameters< T >::StorageBase, [39](#)
 - Lib::System::ProduceDate, [33](#)
- restart
 - Lib::System::System, [43](#)
- restoreByDefault
 - Lib::System::Parameters< T >, [29](#)
- saleDate
 - Produce Date, [11](#)
- SerialNumber
 - Lib::System::SerialNumber, [35](#)
- serialNumber
 - Lib::System::System, [43](#)
- setup
 - Lib::System::Parameters< T >, [29](#)
- StorageBase
 - Lib::System::Parameters< T >::StorageBase, [38](#)
- stringToDateTime
 - Lib::System::Version::Firmware, [22](#)
 - Lib::System::Version::Hardware, [26](#)
- stringToU32
 - Lib::System::Version::Firmware, [22](#)
 - Lib::System::Version::Hardware, [27](#)
- subVersion
 - Lib::System::Version::Firmware, [23](#)
 - Lib::System::Version::Hardware, [27](#)
- System, [9](#)
 - Lib::System::System, [42](#)
- System Library, [1](#)
- toString
 - Lib::System::SerialNumber, [36](#)
 - Lib::System::Version::Firmware, [23](#)
 - Lib::System::Version::Hardware, [27](#)
- Version, [10](#)
 - firmwareVersion, [10](#)
 - hardwareVersion, [10](#)
- write
 - Lib::System::Parameters< T >, [30](#)
 - Lib::System::Parameters< T >::StorageBase, [40](#)
 - Lib::System::ProduceDate, [33](#)