

USB C++ Library

pre-release

Generated by Doxygen 1.9.1

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 Class Documentation	5
3.1 hwEpDesc Union Reference	5
3.1.1 Member Data Documentation	5
3.1.1.1	6
3.1.1.2 bank	6
3.1.1.3 in	6
3.1.1.4 out	6
3.2 samd21usbDevice Class Reference	6
3.2.1 Constructor & Destructor Documentation	8
3.2.1.1 samd21usbDevice()	8
3.2.2 Member Function Documentation	8
3.2.2.1 addEndpointToHardware()	8
3.2.2.2 attach()	8
3.2.2.3 configureClock()	8
3.2.2.4 configureEndpoint()	9
3.2.2.5 configurePins()	9
3.2.2.6 epSizeCode()	9
3.2.2.7 hardwareExec()	9
3.2.2.8 initialiseHardware()	9
3.2.2.9 interruptHandler()	9
3.2.2.10 readComplete()	9
3.2.2.11 sendRequestError()	10
3.2.2.12 setAddress()	10
3.2.2.13 setMode()	10
3.2.2.14 writeln()	10
3.2.2.15 writelnStall()	10
3.2.2.16 writelnZLP()	10
3.2.3 Friends And Related Function Documentation	11
3.2.3.1 USB_Handler	11
3.3 usbCDC Class Reference	11
3.3.1 Constructor & Destructor Documentation	13
3.3.1.1 usbCDC()	13
3.3.2 Member Function Documentation	13
3.3.2.1 bufferFunctionalDescriptor()	13
3.3.3 Friends And Related Function Documentation	13
3.3.3.1 cdcAcmEndpoint	13
3.3.4 Member Data Documentation	13

3.3.4.1 cdc_fncDesc	13
3.3.4.2 cdc_hdrDesc	14
3.3.4.3 ctrlEndpoint	14
3.4 usbCDC::capabilityDescriptor Struct Reference	14
3.4.1 Member Data Documentation	15
3.4.1.1 bDataInterface	15
3.4.1.2 bDescriptorSubtype	15
3.4.1.3 bDescriptorType	15
3.4.1.4 bFunctionLength	15
3.4.1.5 bmCapabilities	15
3.5 usbCDC::cdcEndpoint Class Reference	16
3.5.1 Constructor & Destructor Documentation	17
3.5.1.1 cdcEndpoint()	17
3.5.2 Member Function Documentation	17
3.5.2.1 dataRecieved()	17
3.5.3 Member Data Documentation	17
3.5.3.1 parent	17
3.6 usbCDC::functionalDescriptor Struct Reference	18
3.6.1 Member Data Documentation	18
3.6.1.1 bControllInterface	18
3.6.1.2 bDescriptorSubtype	19
3.6.1.3 bDescriptorType	19
3.6.1.4 bFunctionLength	19
3.6.1.5 bInterfaceClass	19
3.6.1.6 bInterfaceProtocol	19
3.6.1.7 bInterfaceSubClass	19
3.6.1.8 bSubordinateInterface0	19
3.7 usbCDC::headerDescriptor Struct Reference	20
3.7.1 Member Data Documentation	20
3.7.1.1 bcdCDC	20
3.7.1.2 bDescriptorSubtype	20
3.7.1.3 bDescriptorType	21
3.7.1.4 bFunctionLength	21
3.8 usbCDC_ACM Class Reference	21
3.8.1 Constructor & Destructor Documentation	22
3.8.1.1 usbCDC_ACM()	22
3.8.2 Member Function Documentation	22
3.8.2.1 bufferFunctionalDescriptor()	22
3.8.2.2 subclassCode()	23
3.8.2.3 updateInterfaceDescriptor()	23
3.8.3 Friends And Related Function Documentation	23
3.8.3.1 cdcAcmEndpoint	23

3.8.3.2 usbCDC	23
3.8.4 Member Data Documentation	23
3.8.4.1 acm_mgtDesc	23
3.8.4.2 dInEp	23
3.8.4.3 dOutEp	24
3.9 usbCDC_ACM::cdcAcmEndpoint Class Reference	24
3.9.1 Constructor & Destructor Documentation	25
3.9.1.1 cdcAcmEndpoint()	25
3.9.2 Member Function Documentation	25
3.9.2.1 dataRecieved()	25
3.9.3 Member Data Documentation	25
3.9.3.1 parent	25
3.10 usbCDC_ACM::managementDescriptor Struct Reference	26
3.10.1 Member Data Documentation	26
3.10.1.1 bDescriptorSubtype	26
3.10.1.2 bDescriptorType	26
3.10.1.3 bFunctionLength	27
3.10.1.4 bmCapabilities	27
3.11 usbComponent Class Reference	27
3.11.1 Constructor & Destructor Documentation	28
3.11.1.1 usbComponent()	28
3.11.2 Member Function Documentation	29
3.11.2.1 assignEndpoint()	29
3.11.2.2 assignInterfaceNumbers()	29
3.11.2.3 bufferFunctionalDescriptor()	29
3.11.2.4 bufferInterfaceDescriptor()	29
3.11.2.5 endpoints()	29
3.11.2.6 exec()	29
3.11.2.7 protocolCode()	30
3.11.2.8 subclassCode()	30
3.11.2.9 updateInterfaceDescriptor()	30
3.11.3 Friends And Related Function Documentation	30
3.11.3.1 usbDev	30
3.11.4 Member Data Documentation	30
3.11.4.1 m_cmpNameldx	30
3.11.4.2 m_componentName	30
3.11.4.3 m_ep	31
3.11.4.4 m_nEp	31
3.11.4.5 m_subclass	31
3.11.4.6 usb_ifaceDesc	31
3.12 usbComponent::interfaceDescriptor Struct Reference	31
3.12.1 Member Data Documentation	32

3.12.1.1 bAlternateSetting	32
3.12.1.2 bDescriptorType	32
3.12.1.3 bInterfaceClass	32
3.12.1.4 bInterfaceNumber	32
3.12.1.5 bInterfaceProtocol	32
3.12.1.6 bInterfaceSubClass	33
3.12.1.7 bLength	33
3.12.1.8 bNumEndpoints	33
3.12.1.9 iInterface	33
3.13 usbDev Class Reference	33
3.13.1 Constructor & Destructor Documentation	35
3.13.1.1 usbDev()	36
3.13.2 Member Function Documentation	36
3.13.2.1 addComponent()	36
3.13.2.2 addComponentEndpoints()	36
3.13.2.3 addEndpoint()	36
3.13.2.4 addEndpointToHardware()	36
3.13.2.5 addStringDescriptor()	36
3.13.2.6 attach()	37
3.13.2.7 componentEP()	37
3.13.2.8 componentEPs()	37
3.13.2.9 configDescriptor()	37
3.13.2.10 exec()	37
3.13.2.11 hardwareExec()	37
3.13.2.12 initialise()	38
3.13.2.13 initialiseHardware()	38
3.13.2.14 populateDescriptor()	38
3.13.2.15 setAddress()	38
3.13.2.16 setConfigurationString()	38
3.13.2.17 setMaxPower()	38
3.13.2.18 setMode()	38
3.13.2.19 setProductID()	39
3.13.2.20 setProductString()	39
3.13.2.21 setProductVersion()	39
3.13.2.22 setSerialNumberString()	39
3.13.2.23 setVendorID()	39
3.13.2.24 setVendorString()	39
3.13.2.25 writeln()	40
3.13.2.26 writelnStall()	40
3.13.2.27 writelnZLP()	40
3.13.3 Friends And Related Function Documentation	40
3.13.3.1 deviceEndpoint	40

3.13.3.2 usbEndpoint	40
3.13.4 Member Data Documentation	40
3.13.4.1 epList	41
3.13.4.2 maxHardwareEndpoints	41
3.13.4.3 nComponents	41
3.13.4.4 nEndpoints	41
3.13.4.5 nInterfaces	41
3.13.4.6 nStringDesc	41
3.13.4.7 ud_component	41
3.13.4.8 ud_control	41
3.13.4.9 ud_defaultLang	42
3.13.4.10 ud_descriptor	42
3.13.4.11 ud_languageList	42
3.13.4.12 ud_maxPower	42
3.13.4.13 ud_pid	42
3.13.4.14 ud_powerCfg	42
3.13.4.15 ud_productDesc	42
3.13.4.16 ud_qualifier	42
3.13.4.17 ud_rel	43
3.13.4.18 ud_stringDescriptors	43
3.13.4.19 ud_vendorDesc	43
3.13.4.20 ud_vid	43
3.14 usbDev::configurationDescriptor Struct Reference	43
3.14.1 Member Data Documentation	44
3.14.1.1 bConfigurationValue	44
3.14.1.2 bDescriptorType	44
3.14.1.3 bLength	44
3.14.1.4 bmAttributes	44
3.14.1.5 bMaxPower	44
3.14.1.6 bNumInterfaces	45
3.14.1.7 iConfiguration	45
3.14.1.8 wTotalLength	45
3.15 usbDev::deviceDescriptor Struct Reference	45
3.15.1 Member Data Documentation	46
3.15.1.1 bcdDevice	46
3.15.1.2 bcdUSB	46
3.15.1.3 bDescriptorType	46
3.15.1.4 bDeviceClass	46
3.15.1.5 bDeviceProtocol	47
3.15.1.6 bDeviceSubClass	47
3.15.1.7 bLength	47
3.15.1.8 bMaxPacketSize0	47

3.15.1.9 bNumConfigurations	47
3.15.1.10 idProduct	47
3.15.1.11 idVendor	47
3.15.1.12 iManufacturer	47
3.15.1.13 iProduct	48
3.15.1.14 iSerialNumber	48
3.16 usbDev::deviceEndpoint Class Reference	48
3.16.1 Member Enumeration Documentation	49
3.16.1.1 dataTransferDirection	49
3.16.1.2 recipientType	50
3.16.1.3 requestType	50
3.16.1.4 standardFeatureSelector	50
3.16.2 Constructor & Destructor Documentation	51
3.16.2.1 deviceEndpoint()	51
3.16.3 Member Function Documentation	51
3.16.3.1 dataRecieved()	51
3.16.3.2 setupRecieved()	51
3.16.4 Member Data Documentation	51
3.16.4.1 parent	51
3.17 usbDev::deviceQualifier Struct Reference	52
3.17.1 Member Data Documentation	52
3.17.1.1 bcdUSB	52
3.17.1.2 bDescriptorType	53
3.17.1.3 bDeviceClass	53
3.17.1.4 bDeviceProtocol	53
3.17.1.5 bDeviceSubClass	53
3.17.1.6 bLength	53
3.17.1.7 bMaxPacketSize0	53
3.17.1.8 bNumConfigurations	53
3.17.1.9 bReserved	54
3.18 usbDev::languageDescriptor Struct Reference	54
3.18.1 Member Data Documentation	54
3.18.1.1 bDescriptorType	54
3.18.1.2 bLength	54
3.18.1.3 langId	55
3.19 usbDev::stringDescriptor Class Reference	55
3.19.1 Constructor & Destructor Documentation	55
3.19.1.1 stringDescriptor()	56
3.19.2 Member Function Documentation	56
3.19.2.1 hasString()	56
3.19.2.2 setString()	56
3.19.2.3 writeDescriptor()	56

3.19.3 Member Data Documentation	56
3.19.3.1 bDescriptorType	56
3.19.3.2 bLength	56
3.19.3.3 bString	57
3.19.3.4 nChars	57
3.20 usbEndpoint Class Reference	57
3.20.1 Member Enumeration Documentation	58
3.20.1.1 endpointDirection	58
3.20.1.2 endpointSize	59
3.20.1.3 endpointType	59
3.20.2 Constructor & Destructor Documentation	59
3.20.2.1 usbEndpoint()	59
3.20.3 Member Function Documentation	60
3.20.3.1 dataRecieved()	60
3.20.3.2 hardwareEndpoint()	60
3.20.3.3 setHardware()	60
3.20.3.4 setHardwareEndpoint()	60
3.20.3.5 setupRecieved()	60
3.20.3.6 writeln()	60
3.20.3.7 writelnStall()	61
3.20.3.8 writelnZLP()	61
3.20.4 Member Data Documentation	61
3.20.4.1 bufferSize	61
3.20.4.2 descriptor	61
3.20.4.3 dir	61
3.20.4.4 inBuffer	61
3.20.4.5 outBuffer	61
3.20.4.6 size	62
3.20.4.7 type	62
3.20.4.8 usbHardware	62
3.21 usbEndpoint::endpointDescriptor Struct Reference	62
3.21.1 Member Data Documentation	63
3.21.1.1 bDescriptorType	63
3.21.1.2 bEndpointAddress	63
3.21.1.3 bInterval	63
3.21.1.4 bLength	63
3.21.1.5 bmAttributes	63
3.21.1.6 wMaxPacketSize	64
3.22 usbSetupPacket Class Reference	64
3.22.1 Constructor & Destructor Documentation	64
3.22.1.1 usbSetupPacket()	65
3.22.2 Member Data Documentation	65

3.22.2.1	65
3.22.2.2 bmRequestType	65
3.22.2.3 bRequest	65
3.22.2.4 descriptor	65
3.22.2.5 descriptorIndex	65
3.22.2.6 request	65
3.22.2.7 wIndex	66
3.22.2.8 wLength	66
3.22.2.9 wValue	66
Index	67

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

hwEpDesc	5
usbCDC::capabilityDescriptor	14
usbCDC::functionalDescriptor	18
usbCDC::headerDescriptor	20
usbCDC_ACM::managementDescriptor	26
usbComponent	27
usbCDC	11
usbCDC_ACM	21
usbComponent::interfaceDescriptor	31
usbDev	33
samd21usbDevice	6
usbDev::configurationDescriptor	43
usbDev::deviceDescriptor	45
usbDev::deviceQualifier	52
usbDev::languageDescriptor	54
usbDev::stringDescriptor	55
usbEndpoint	57
usbCDC::cdcEndpoint	16
usbCDC_ACM::cdcAcmEndpoint	24
usbDev::deviceEndpoint	48
usbEndpoint::endpointDescriptor	62
usbSetupPacket	64

Chapter 2

Class Index

2.1 Class List

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

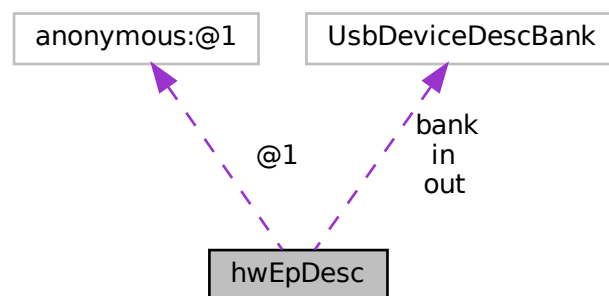
hwEpDesc	5
samd21usbDevice	6
usbCDC	11
usbCDC::capabilityDescriptor	14
usbCDC::cdcEndpoint	16
usbCDC::functionalDescriptor	18
usbCDC::headerDescriptor	20
usbCDC_ACM	21
usbCDC_ACM::cdcAcmEndpoint	24
usbCDC_ACM::managementDescriptor	26
usbComponent	27
usbComponent::interfaceDescriptor	31
usbDev	33
usbDev::configurationDescriptor	43
usbDev::deviceDescriptor	45
usbDev::deviceEndpoint	48
usbDev::deviceQualifier	52
usbDev::languageDescriptor	54
usbDev::stringDescriptor	55
usbEndpoint	57
usbEndpoint::endpointDescriptor	62
usbSetupPacket	64

Chapter 3

Class Documentation

3.1 hwEpDesc Union Reference

Collaboration diagram for hwEpDesc:



Public Attributes

- struct {
 UsbDeviceDescBank `in`
 UsbDeviceDescBank `out`
};
- UsbDeviceDescBank `bank` [2]

3.1.1 Member Data Documentation

3.1.1.1

```
struct { ... }
```

3.1.1.2 bank

```
UsbDeviceDescBank hwEpDesc::bank[2]
```

3.1.1.3 in

```
UsbDeviceDescBank hwEpDesc::in
```

3.1.1.4 out

```
UsbDeviceDescBank hwEpDesc::out
```

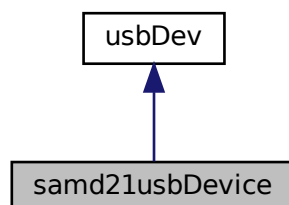
The documentation for this union was generated from the following file:

- `src/hardware/samD21/samD21usbDevice.cpp`

3.2 samd21usbDevice Class Reference

```
#include <samD21usbDevice.h>
```

Inheritance diagram for `samd21usbDevice`:




```

ud_defaultLang
ud_pid
ud_rel

```



- ## Protected Member Functions

- ## Private Member Functions

- Generated by Doxygen

Friends

- void [USB_Handler](#) ()

Additional Inherited Members

3.2.1 Constructor & Destructor Documentation

3.2.1.1 [samd21usbDevice\(\)](#)

```
samd21usbDevice::samd21usbDevice ( )
```

3.2.2 Member Function Documentation

3.2.2.1 [addEndpointToHardware\(\)](#)

```
bool samd21usbDevice::addEndpointToHardware (
    uint8_t idx,
    usbEndpoint * ep ) [private], [virtual]
```

Implements [usbDev](#).

3.2.2.2 [attach\(\)](#)

```
void samd21usbDevice::attach ( ) [virtual]
```

Implements [usbDev](#).

3.2.2.3 [configureClock\(\)](#)

```
void samd21usbDevice::configureClock ( ) [private]
```

This function configures the clocks required for USB. It expects GCLK0 to output a 48MHz clock.

3.2.2.4 configureEndpoint()

```
void samd21usbDevice::configureEndpoint (
    uint8_t idx ) [private]
```

3.2.2.5 configurePins()

```
void samd21usbDevice::configurePins ( ) [private]
```

3.2.2.6 epSizeCode()

```
uint8_t samd21usbDevice::epSizeCode (
    usbEndpoint::endpointSize sz ) [private]
```

3.2.2.7 hardwareExec()

```
void samd21usbDevice::hardwareExec ( ) [protected], [virtual]
```

Implements [usbDev](#).

3.2.2.8 initialiseHardware()

```
void samd21usbDevice::initialiseHardware ( ) [protected], [virtual]
```

Implements [usbDev](#).

3.2.2.9 interruptHandler()

```
void samd21usbDevice::interruptHandler ( ) [private]
```

3.2.2.10 readComplete()

```
void samd21usbDevice::readComplete (
    uint8_t dEp,
    uint8_t bnk ) [private]
```

3.2.2.11 sendRequestError()

```
void samd21usbDevice::sendRequestError (
    usbEndpoint * ep,
    uint8_t bnk = 0 ) [protected]
```

3.2.2.12 setAddress()

```
void samd21usbDevice::setAddress (
    uint8_t addr ) [protected], [virtual]
```

Implements [usbDev](#).

3.2.2.13 setMode()

```
void samd21usbDevice::setMode (
    usbModes mode ) [protected], [virtual]
```

Implements [usbDev](#).

3.2.2.14 writeIn()

```
uint16_t samd21usbDevice::writeIn (
    uint8_t ep,
    uint16_t nBytes ) [protected], [virtual]
```

Implements [usbDev](#).

3.2.2.15 writeInStall()

```
void samd21usbDevice::writeInStall (
    uint8_t ep ) [protected], [virtual]
```

Implements [usbDev](#).

3.2.2.16 writeInZLP()

```
void samd21usbDevice::writeInZLP (
    uint8_t ep ) [protected], [virtual]
```

Implements [usbDev](#).

3.2.3 Friends And Related Function Documentation

3.2.3.1 USB_Handler

```
void USB_Handler ( ) [friend]
```

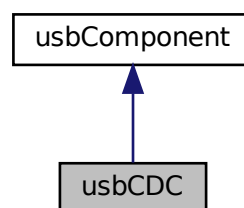
The documentation for this class was generated from the following files:

- src/hardware/samD21/samD21usbDevice.h
- src/hardware/samD21/samD21usbDevice.cpp

3.3 usbCDC Class Reference

```
#include <usbCDC.h>
```

Inheritance diagram for usbCDC:



Additional Inherited Members

3.3.1 Constructor & Destructor Documentation

3.3.1.1 usbCDC()

```
usbCDC::usbCDC (
    usbComponent * subclass,
    const char * name = "CDC" )
```

3.3.2 Member Function Documentation

3.3.2.1 bufferFunctionalDescriptor()

```
void usbCDC::bufferFunctionalDescriptor (
    uint8_t * buffer,
    uint16_t * len ) [protected], [virtual]
```

Reimplemented from [usbComponent](#).

3.3.3 Friends And Related Function Documentation

3.3.3.1 cdcAcmEndpoint

```
friend class cdcAcmEndpoint [friend]
```

3.3.4 Member Data Documentation

3.3.4.1 cdc_fncDesc

```
functionalDescriptor usbCDC::cdc_fncDesc [protected]
```

3.3.4.2 cdc_hdrDesc

```
headerDescriptor usbCDC::cdc_hdrDesc [protected]
```

3.3.4.3 ctrlEndpoint

```
usbEndpoint* usbCDC::ctrlEndpoint [protected]
```

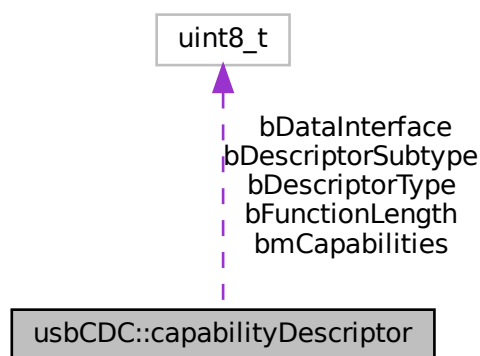
The documentation for this class was generated from the following files:

- src/CDC/usbCDC.h
- src/CDC/USBCDC.cpp

3.4 usbCDC::capabilityDescriptor Struct Reference

```
#include <usbCDC.h>
```

Collaboration diagram for usbCDC::capabilityDescriptor:



Public Attributes

- `uint8_t bDataInterface`
- `uint8_t bDescriptorSubtype`
- `uint8_t bDescriptorType`
- `uint8_t bFunctionLength`
- `uint8_t bmCapabilities`

3.4.1 Member Data Documentation

3.4.1.1 bDataInterface

`uint8_t usbCDC::capabilityDescriptor::bDataInterface`

3.4.1.2 bDescriptorSubtype

`uint8_t usbCDC::capabilityDescriptor::bDescriptorSubtype`

3.4.1.3 bDescriptorType

`uint8_t usbCDC::capabilityDescriptor::bDescriptorType`

3.4.1.4 bFunctionLength

`uint8_t usbCDC::capabilityDescriptor::bFunctionLength`

3.4.1.5 bmCapabilities

`uint8_t usbCDC::capabilityDescriptor::bmCapabilities`

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

- `src/CDC/usbCDC.h`

Private Attributes

- [usbCDC](#) * [parent](#)

Additional Inherited Members

3.5.1 Constructor & Destructor Documentation

3.5.1.1 cdcEndpoint()

```
usbCDC::cdcEndpoint::cdcEndpoint (
    usbCDC * p,
    uint16_t bSize,
    usbEndpoint::endpointSize sz,
    usbEndpoint::endpointDirection uDir,
    usbEndpoint::endpointType uType )
```

3.5.2 Member Function Documentation

3.5.2.1 dataRecieved()

```
void usbCDC::cdcEndpoint::dataRecieved (
    uint16_t nBytes ) [virtual]
```

Implements [usbEndpoint](#).

3.5.3 Member Data Documentation

3.5.3.1 parent

```
usbCDC* usbCDC::cdcEndpoint::parent [private]
```

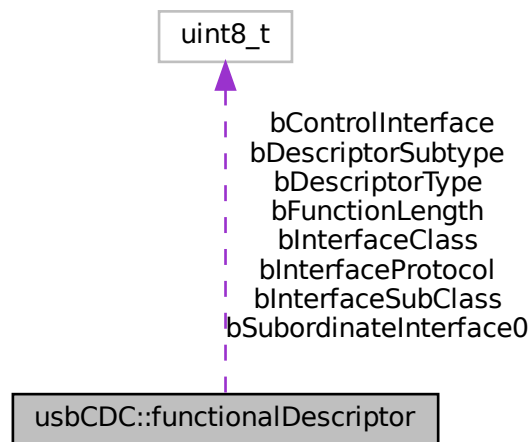
The documentation for this class was generated from the following files:

- `src/CDC/usbCDC.h`
- `src/CDC/USBCDC.cpp`

3.6 usbCDC::functionalDescriptor Struct Reference

```
#include <usbCDC.h>
```

Collaboration diagram for usbCDC::functionalDescriptor:



Public Attributes

- [uint8_t bControllInterface](#)
- [uint8_t bDescriptorSubtype](#)
- [uint8_t bDescriptorType](#)
- [uint8_t bFunctionLength](#)
- [uint8_t bInterfaceClass](#)
- [uint8_t bInterfaceProtocol](#)
- [uint8_t bInterfaceSubClass](#)
- [uint8_t bSubordinateInterface0](#)

3.6.1 Member Data Documentation

3.6.1.1 bControllInterface

```
uint8_t usbCDC::functionalDescriptor::bControlInterface
```

3.6.1.2 bDescriptorSubtype

```
uint8_t usbCDC::functionalDescriptor::bDescriptorSubtype
```

3.6.1.3 bDescriptorType

```
uint8_t usbCDC::functionalDescriptor::bDescriptorType
```

3.6.1.4 bFunctionLength

```
uint8_t usbCDC::functionalDescriptor::bFunctionLength
```

3.6.1.5 bInterfaceClass

```
uint8_t usbCDC::functionalDescriptor::bInterfaceClass
```

3.6.1.6 bInterfaceProtocol

```
uint8_t usbCDC::functionalDescriptor::bInterfaceProtocol
```

3.6.1.7 bInterfaceSubClass

```
uint8_t usbCDC::functionalDescriptor::bInterfaceSubClass
```

3.6.1.8 bSubordinateInterface0

```
uint8_t usbCDC::functionalDescriptor::bSubordinateInterface0
```

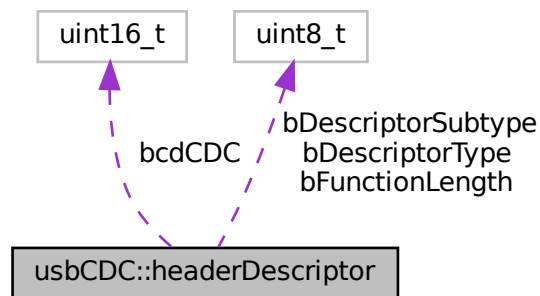
The documentation for this struct was generated from the following file:

- src/CDC/usbCDC.h

3.7 usbCDC::headerDescriptor Struct Reference

```
#include <usbCDC.h>
```

Collaboration diagram for usbCDC::headerDescriptor:



Public Attributes

- `uint16_t` [bcdCDC](#)
- `uint8_t` [bDescriptorSubtype](#)
- `uint8_t` [bDescriptorType](#)
- `uint8_t` [bFunctionLength](#)

3.7.1 Member Data Documentation

3.7.1.1 bcdCDC

```
uint16_t usbCDC::headerDescriptor::bcdCDC
```

3.7.1.2 bDescriptorSubtype

```
uint8_t usbCDC::headerDescriptor::bDescriptorSubtype
```


Classes

- class [cdcAcmEndpoint](#)
- struct [managementDescriptor](#)

Public Member Functions

- [usbCDC_ACM](#) ()
- void [bufferFunctionalDescriptor](#) (uint8_t *buffer, uint16_t *len)
- uint8_t [subclassCode](#) ()
- void [updateInterfaceDescriptor](#) ()

Protected Attributes

- [managementDescriptor](#) [acm_mgtDesc](#)
- [usbCDC_ACM::cdcAcmEndpoint](#) * [dInEp](#)
- [usbCDC_ACM::cdcAcmEndpoint](#) * [dOutEp](#)

Friends

- class [cdcAcmEndpoint](#)
- class [usbCDC](#)

Additional Inherited Members

3.8.1 Constructor & Destructor Documentation

3.8.1.1 [usbCDC_ACM](#)()

```
usbCDC_ACM::usbCDC_ACM ( )
```

3.8.2 Member Function Documentation

3.8.2.1 [bufferFunctionalDescriptor](#)()

```
void usbCDC_ACM::bufferFunctionalDescriptor (
    uint8_t * buffer,
    uint16_t * len ) [virtual]
```

Reimplemented from [usbComponent](#).

3.8.2.2 subclassCode()

```
uint8_t usbCDC_ACM::subclassCode ( ) [virtual]
```

Reimplemented from [usbComponent](#).

3.8.2.3 updateInterfaceDescriptor()

```
void usbCDC_ACM::updateInterfaceDescriptor ( ) [virtual]
```

Reimplemented from [usbComponent](#).

3.8.3 Friends And Related Function Documentation

3.8.3.1 cdcAcmEndpoint

```
friend class cdcAcmEndpoint [friend]
```

3.8.3.2 usbCDC

```
friend class usbCDC [friend]
```

3.8.4 Member Data Documentation

3.8.4.1 acm_mgtDesc

```
managementDescriptor usbCDC_ACM::acm_mgtDesc [protected]
```

3.8.4.2 dInEp

```
usbCDC_ACM::cdcAcmEndpoint* usbCDC_ACM::dInEp [protected]
```


Private Attributes

- [usbCDC_ACM](#) * [parent](#)

Additional Inherited Members

3.9.1 Constructor & Destructor Documentation

3.9.1.1 cdcAcmEndpoint()

```
usbCDC_ACM::cdcAcmEndpoint::cdcAcmEndpoint (
    usbCDC_ACM * p,
    uint16_t bSize,
    usbEndpoint::endpointSize sz,
    usbEndpoint::endpointDirection uDir,
    usbEndpoint::endpointType uType )
```

3.9.2 Member Function Documentation

3.9.2.1 dataRecieved()

```
void usbCDC_ACM::cdcAcmEndpoint::dataRecieved (
    uint16_t nBytes ) [virtual]
```

Implements [usbEndpoint](#).

3.9.3 Member Data Documentation

3.9.3.1 parent

```
usbCDC_ACM* usbCDC_ACM::cdcAcmEndpoint::parent [private]
```

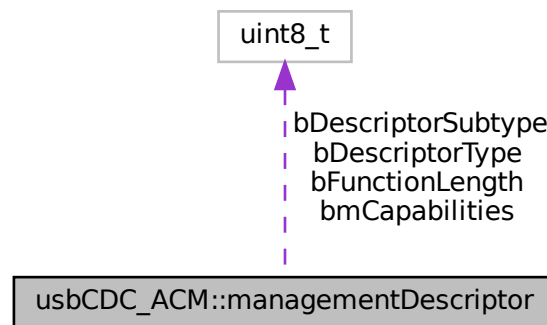
The documentation for this class was generated from the following files:

- `src/CDC/usbCDC_ACM.h`
- `src/CDC/usbCDC_ACM.cpp`

3.10 usbCDC_ACM::managementDescriptor Struct Reference

```
#include <usbCDC_ACM.h>
```

Collaboration diagram for usbCDC_ACM::managementDescriptor:



Public Attributes

- `uint8_t` [bDescriptorSubtype](#)
- `uint8_t` [bDescriptorType](#)
- `uint8_t` [bFunctionLength](#)
- `uint8_t` [bmCapabilities](#)

3.10.1 Member Data Documentation

3.10.1.1 bDescriptorSubtype

```
uint8_t usbCDC_ACM::managementDescriptor::bDescriptorSubtype
```

3.10.1.2 bDescriptorType

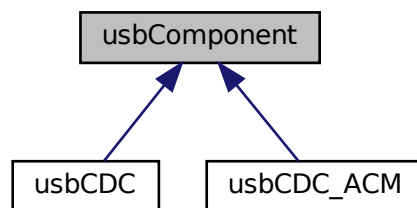
```
uint8_t usbCDC_ACM::managementDescriptor::bDescriptorType
```

```
uint8_t usbCDC_ACM::managementDescriptor::bFunctionLength
```

```
uint8_t usbCDC_ACM::managementDescriptor::bmCapabilities
```

- src/CDC/usbCDC_ACM.h

Inheritance diagram for usbComponent:

[illegible]

Classes

- struct [interfaceDescriptor](#)

Public Member Functions

- [usbComponent](#) (uint8_t classCode=0, uint8_t classControlProtocol=0, [usbComponent](#) *subclass=nullptr, const char *name=nullptr)
- virtual void [bufferFunctionalDescriptor](#) (uint8_t *buffer, uint16_t *len)
- uint8_t [endpoints](#) ()
- virtual void [exec](#) ()
- virtual uint8_t [protocolCode](#) ()
- virtual uint8_t [subclassCode](#) ()

Public Attributes

- [interfaceDescriptor](#) [usb_ifaceDesc](#)

Protected Member Functions

- bool [assignEndpoint](#) ([usbEndpoint](#) *ep)
- uint8_t [assignInterfaceNumbers](#) (uint8_t start)
- void [bufferInterfaceDescriptor](#) (uint8_t *buffer, uint16_t *len)
- virtual void [updateInterfaceDescriptor](#) ()

Protected Attributes

- uint8_t [m_cmpNameIdx](#)
- const char * [m_componentName](#)
- [usbEndpoint](#) * [m_ep](#) [USB_MAX_COMPONENT_ENDPOINTS]
- uint8_t [m_nEp](#) =0
- [usbComponent](#) * [m_subclass](#)

Friends

- class [usbDev](#)

3.11.1 Constructor & Destructor Documentation

3.11.1.1 [usbComponent](#)()

```
usbComponent::usbComponent (
    uint8_t classCode = 0,
    uint8_t classControlProtocol = 0,
    usbComponent * subclass = nullptr,
    const char * name = nullptr )
```

3.11.2 Member Function Documentation

3.11.2.1 assignEndpoint()

```
bool usbComponent::assignEndpoint (
    usbEndpoint * ep ) [protected]
```

3.11.2.2 assignInterfaceNumbers()

```
uint8_t usbComponent::assignInterfaceNumbers (
    uint8_t start ) [protected]
```

3.11.2.3 bufferFunctionalDescriptor()

```
void usbComponent::bufferFunctionalDescriptor (
    uint8_t * buffer,
    uint16_t * len ) [virtual]
```

Reimplemented in [usbCDC_ACM](#), and [usbCDC](#).

3.11.2.4 bufferInterfaceDescriptor()

```
void usbComponent::bufferInterfaceDescriptor (
    uint8_t * buffer,
    uint16_t * len ) [protected]
```

3.11.2.5 endpoints()

```
uint8_t usbComponent::endpoints ( )
```

3.11.2.6 exec()

```
void usbComponent::exec ( ) [virtual]
```

3.11.2.7 protocolCode()

```
uint8_t usbComponent::protocolCode ( ) [virtual]
```

3.11.2.8 subclassCode()

```
uint8_t usbComponent::subclassCode ( ) [virtual]
```

Reimplemented in [usbCDC_ACM](#).

3.11.2.9 updateInterfaceDescriptor()

```
void usbComponent::updateInterfaceDescriptor ( ) [protected], [virtual]
```

Reimplemented in [usbCDC_ACM](#).

3.11.3 Friends And Related Function Documentation

3.11.3.1 usbDev

```
friend class usbDev [friend]
```

3.11.4 Member Data Documentation

3.11.4.1 m_cmpNameIdx

```
uint8_t usbComponent::m_cmpNameIdx [protected]
```

3.11.4.2 m_componentName

```
const char* usbComponent::m_componentName [protected]
```


3.11.4.3 m_ep

```
usbEndpoint* usbComponent::m_ep[USB_MAX_COMPONENT_ENDPOINTS] [protected]
```

3.11.4.4 m_nEp

```
uint8_t usbComponent::m_nEp = 0 [protected]
```

3.11.4.5 m_subclass

```
usbComponent* usbComponent::m_subclass [protected]
```

3.11.4.6 usb_ifaceDesc

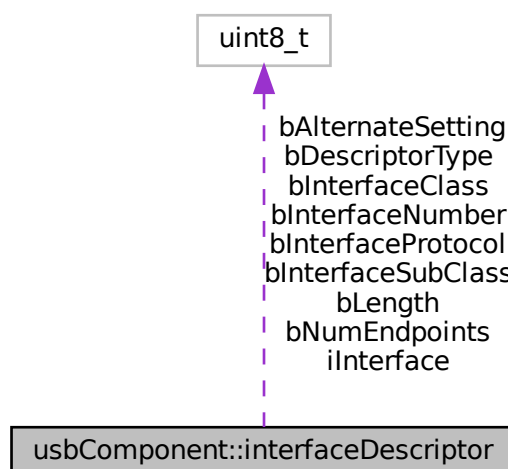
```
interfaceDescriptor usbComponent::usb_ifaceDesc
```

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

- src/usbComponent.h
- src/usbComponent.cpp

3.12 usbComponent::interfaceDescriptor Struct Reference

Collaboration diagram for usbComponent::interfaceDescriptor:



Public Attributes

- [uint8_t bAlternateSetting](#)
- [uint8_t bDescriptorType](#)
- [uint8_t bInterfaceClass](#)
- [uint8_t bInterfaceNumber](#)
- [uint8_t bInterfaceProtocol](#)
- [uint8_t bInterfaceSubClass](#)
- [uint8_t bLength](#)
- [uint8_t bNumEndpoints](#)
- [uint8_t iInterface](#)

3.12.1 Member Data Documentation

3.12.1.1 bAlternateSetting

```
uint8_t usbComponent::interfaceDescriptor::bAlternateSetting
```

3.12.1.2 bDescriptorType

```
uint8_t usbComponent::interfaceDescriptor::bDescriptorType
```

3.12.1.3 bInterfaceClass

```
uint8_t usbComponent::interfaceDescriptor::bInterfaceClass
```

3.12.1.4 bInterfaceNumber

```
uint8_t usbComponent::interfaceDescriptor::bInterfaceNumber
```

3.12.1.5 bInterfaceProtocol

```
uint8_t usbComponent::interfaceDescriptor::bInterfaceProtocol
```

3.12.1.6 bInterfaceSubClass

```
uint8_t usbComponent::interfaceDescriptor::bInterfaceSubClass
```

3.12.1.7 bLength

```
uint8_t usbComponent::interfaceDescriptor::bLength
```

3.12.1.8 bNumEndpoints

```
uint8_t usbComponent::interfaceDescriptor::bNumEndpoints
```

3.12.1.9 iInterface

```
uint8_t usbComponent::interfaceDescriptor::iInterface
```

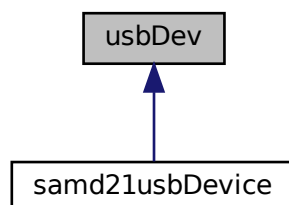
The documentation for this struct was generated from the following file:

- src/usbComponent.h

3.13 usbDev Class Reference

```
#include <usbDev.h>
```

Inheritance diagram for usbDev:





- ## Public Member Functions

- Generated by Doxygen

Protected Member Functions

- bool [addComponentEndpoints](#) ([usbComponent](#) *component)
- bool [addEndpoint](#) ([usbEndpoint](#) *ep)
- virtual bool [addEndpointToHardware](#) (uint8_t idx, [usbEndpoint](#) *ep)=0
- uint8_t [addStringDescriptor](#) (const char *desc)
- [usbEndpoint](#) * [componentEP](#) ([usbComponent](#) *component, uint8_t idx)
- uint8_t [componentEPs](#) ([usbComponent](#) *component)
- void [configDescriptor](#) (uint8_t *buffer, uint16_t *len)
- virtual void [hardwareExec](#) ()=0
- virtual void [initialiseHardware](#) ()=0
- void [populateDescriptor](#) ()
- virtual void [setAddress](#) (uint8_t addr)=0
- virtual void [setMode](#) (usbModeTypes mode)=0
- virtual uint16_t [writeIn](#) (uint8_t ep, uint16_t nBytes)=0
- virtual void [writeInStall](#) (uint8_t ep)=0
- virtual void [writeInZLP](#) (uint8_t ep)=0

Protected Attributes

- [usbEndpoint](#) * [epList](#) [USB_MAX_ENDPOINTS]
- uint8_t [maxHardwareEndpoints](#)
- uint8_t [nComponents](#)
- uint8_t [nEndpoints](#)
- uint8_t [nInterfaces](#)
- uint8_t [nStringDesc](#)
- [usbComponent](#) * [ud_component](#) [USB_MAX_COMPONENTS]
- [usbDev::deviceEndpoint](#) [ud_control](#)
- uint16_t [ud_defaultLang](#)
- [usbDev::deviceDescriptor](#) [ud_descriptor](#)
- [usbDev::languageDescriptor](#) [ud_languageList](#)
- uint8_t [ud_maxPower](#)
- uint16_t [ud_pid](#)
- [usbPowerConfig](#) [ud_powerCfg](#)
- const char * [ud_productDesc](#)
- [usbDev::deviceQualifier](#) [ud_qualifier](#)
- uint16_t [ud_rel](#)
- [usbDev::stringDescriptor](#) [ud_stringDescriptors](#) [USB_MAX_STRINGDESCRIPTORS]
- const char * [ud_vendorDesc](#)
- uint16_t [ud_vid](#)

Friends

- class [deviceEndpoint](#)
- class [usbEndpoint](#)

3.13.1 Constructor & Destructor Documentation

3.13.1.1 usbDev()

```
usbDev::usbDev ( )
```

3.13.2 Member Function Documentation

3.13.2.1 addComponent()

```
bool usbDev::addComponent (
    usbComponent * component )
```

3.13.2.2 addComponentEndpoints()

```
bool usbDev::addComponentEndpoints (
    usbComponent * component ) [protected]
```

3.13.2.3 addEndpoint()

```
bool usbDev::addEndpoint (
    usbEndpoint * ep ) [protected]
```

3.13.2.4 addEndpointToHardware()

```
virtual bool usbDev::addEndpointToHardware (
    uint8_t idx,
    usbEndpoint * ep ) [protected], [pure virtual]
```

Implemented in [samd21usbDevice](#).

3.13.2.5 addStringDescriptor()

```
uint8_t usbDev::addStringDescriptor (
    const char * desc ) [protected]
```

3.13.2.6 attach()

```
virtual void usbDev::attach ( ) [pure virtual]
```

Implemented in [samd21usbDevice](#).

3.13.2.7 componentEP()

```
usbEndpoint * usbDev::componentEP (
    usbComponent * component,
    uint8_t idx ) [protected]
```

3.13.2.8 componentEPs()

```
uint8_t usbDev::componentEPs (
    usbComponent * component ) [protected]
```

3.13.2.9 configDescriptor()

```
void usbDev::configDescriptor (
    uint8_t * buffer,
    uint16_t * len ) [protected]
```

3.13.2.10 exec()

```
void usbDev::exec ( )
```

3.13.2.11 hardwareExec()

```
virtual void usbDev::hardwareExec ( ) [protected], [pure virtual]
```

Implemented in [samd21usbDevice](#).

3.13.2.12 initialise()

```
void usbDev::initialise ( )
```

3.13.2.13 initialiseHardware()

```
virtual void usbDev::initialiseHardware ( ) [protected], [pure virtual]
```

Implemented in [samd21usbDevice](#).

3.13.2.14 populateDescriptor()

```
void usbDev::populateDescriptor ( ) [protected]
```

3.13.2.15 setAddress()

```
virtual void usbDev::setAddress (
    uint8_t addr ) [protected], [pure virtual]
```

Implemented in [samd21usbDevice](#).

3.13.2.16 setConfigurationString()

```
void usbDev::setConfigurationString (
    const char * desc )
```

3.13.2.17 setMaxPower()

```
void usbDev::setMaxPower (
    float current )
```

3.13.2.18 setMode()

```
virtual void usbDev::setMode (
    usbModeTypes mode ) [protected], [pure virtual]
```

Implemented in [samd21usbDevice](#).

3.13.2.19 setProductID()

```
void usbDev::setProductID (
    uint16_t id )
```

3.13.2.20 setProductString()

```
void usbDev::setProductString (
    const char * desc )
```

3.13.2.21 setProductVersion()

```
void usbDev::setProductVersion (
    uint8_t major,
    uint8_t minor,
    uint8_t fix )
```

3.13.2.22 setSerialNumberString()

```
void usbDev::setSerialNumberString (
    const char * desc )
```

3.13.2.23 setVendorID()

```
void usbDev::setVendorID (
    uint16_t id )
```

3.13.2.24 setVendorString()

```
void usbDev::setVendorString (
    const char * desc )
```

3.13.2.25 writeIn()

```
virtual uint16_t usbDev::writeIn (
    uint8_t ep,
    uint16_t nBytes ) [protected], [pure virtual]
```

Implemented in [samd21usbDevice](#).

3.13.2.26 writeInStall()

```
virtual void usbDev::writeInStall (
    uint8_t ep ) [protected], [pure virtual]
```

Implemented in [samd21usbDevice](#).

3.13.2.27 writeInZLP()

```
virtual void usbDev::writeInZLP (
    uint8_t ep ) [protected], [pure virtual]
```

Implemented in [samd21usbDevice](#).

3.13.3 Friends And Related Function Documentation

3.13.3.1 deviceEndpoint

```
friend class deviceEndpoint [friend]
```

3.13.3.2 usbEndpoint

```
friend class usbEndpoint [friend]
```

3.13.4 Member Data Documentation

3.13.4.1 epList

```
usbEndpoint* usbDev::epList [USB_MAX_ENDPOINTS] [protected]
```

3.13.4.2 maxHardwareEndpoints

```
uint8_t usbDev::maxHardwareEndpoints [protected]
```

3.13.4.3 nComponents

```
uint8_t usbDev::nComponents [protected]
```

3.13.4.4 nEndpoints

```
uint8_t usbDev::nEndpoints [protected]
```

3.13.4.5 nInterfaces

```
uint8_t usbDev::nInterfaces [protected]
```

3.13.4.6 nStringDesc

```
uint8_t usbDev::nStringDesc [protected]
```

3.13.4.7 ud_component

```
usbComponent* usbDev::ud_component [USB_MAX_COMPONENTS] [protected]
```

3.13.4.8 ud_control

```
usbDev::deviceEndpoint usbDev::ud_control [protected]
```

3.13.4.9 ud_defaultLang

uint16_t usbDev::ud_defaultLang [protected]

3.13.4.10 ud_descriptor

usbDev::deviceDescriptor usbDev::ud_descriptor [protected]

3.13.4.11 ud_languageList

usbDev::languageDescriptor usbDev::ud_languageList [protected]

3.13.4.12 ud_maxPower

uint8_t usbDev::ud_maxPower [protected]

3.13.4.13 ud_pid

uint16_t usbDev::ud_pid [protected]

3.13.4.14 ud_powerCfg

usbPowerConfig usbDev::ud_powerCfg [protected]

3.13.4.15 ud_productDesc

const char* usbDev::ud_productDesc [protected]

3.13.4.16 ud_qualifier

usbDev::deviceQualifier usbDev::ud_qualifier [protected]

3.13.4.17 ud_rel

```
uint16_t usbDev::ud_rel [protected]
```

3.13.4.18 ud_stringDescriptors

```
usbDev::stringDescriptor usbDev::ud_stringDescriptors[USB_MAX_STRINGDESCRIPTORS] [protected]
```

3.13.4.19 ud_vendorDesc

```
const char* usbDev::ud_vendorDesc [protected]
```

3.13.4.20 ud_vid

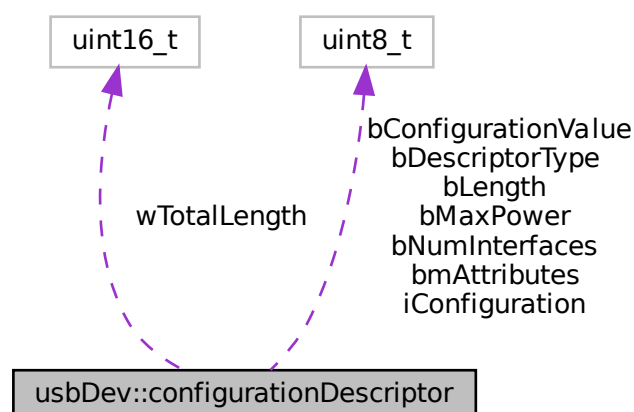
```
uint16_t usbDev::ud_vid [protected]
```

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

- src/usbDev.h
- src/usbDev.cpp

3.14 usbDev::configurationDescriptor Struct Reference

Collaboration diagram for usbDev::configurationDescriptor:



Public Attributes

- [uint8_t bConfigurationValue](#)
- [uint8_t bDescriptorType](#)
- [uint8_t bLength](#)
- [uint8_t bmAttributes](#)
- [uint8_t bMaxPower](#)
- [uint8_t bNumInterfaces](#)
- [uint8_t iConfiguration](#)
- [uint16_t wTotalLength](#)

3.14.1 Member Data Documentation

3.14.1.1 bConfigurationValue

```
uint8_t usbDev::configurationDescriptor::bConfigurationValue
```

3.14.1.2 bDescriptorType

```
uint8_t usbDev::configurationDescriptor::bDescriptorType
```

3.14.1.3 bLength

```
uint8_t usbDev::configurationDescriptor::bLength
```

3.14.1.4 bmAttributes

```
uint8_t usbDev::configurationDescriptor::bmAttributes
```

3.14.1.5 bMaxPower

```
uint8_t usbDev::configurationDescriptor::bMaxPower
```

3.14.1.6 bNumInterfaces

```
uint8_t usbDev::configurationDescriptor::bNumInterfaces
```

3.14.1.7 iConfiguration

```
uint8_t usbDev::configurationDescriptor::iConfiguration
```

3.14.1.8 wTotalLength

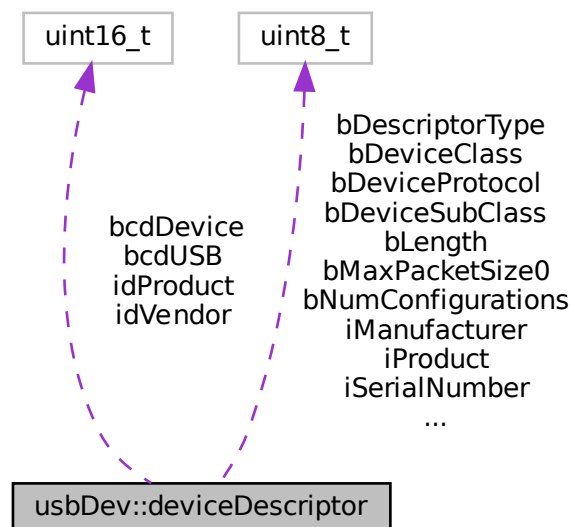
```
uint16_t usbDev::configurationDescriptor::wTotalLength
```

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

- src/usbDev.h

3.15 usbDev::deviceDescriptor Struct Reference

Collaboration diagram for usbDev::deviceDescriptor:



Public Attributes

- uint16_t [bcdDevice](#)
- uint16_t [bcdUSB](#)
- uint8_t [bDescriptorType](#)
- uint8_t [bDeviceClass](#)
- uint8_t [bDeviceProtocol](#)
- uint8_t [bDeviceSubClass](#)
- uint8_t [bLength](#)
- uint8_t [bMaxPacketSize0](#)
- uint8_t [bNumConfigurations](#)
- uint16_t [idProduct](#)
- uint16_t [idVendor](#)
- uint8_t [iManufacturer](#)
- uint8_t [iProduct](#)
- uint8_t [iSerialNumber](#)

3.15.1 Member Data Documentation

3.15.1.1 bcdDevice

uint16_t usbDev::deviceDescriptor::bcdDevice

3.15.1.2 bcdUSB

uint16_t usbDev::deviceDescriptor::bcdUSB

3.15.1.3 bDescriptorType

uint8_t usbDev::deviceDescriptor::bDescriptorType

3.15.1.4 bDeviceClass

uint8_t usbDev::deviceDescriptor::bDeviceClass

3.15.1.5 bDeviceProtocol

```
uint8_t usbDev::deviceDescriptor::bDeviceProtocol
```

3.15.1.6 bDeviceSubClass

```
uint8_t usbDev::deviceDescriptor::bDeviceSubClass
```

3.15.1.7 bLength

```
uint8_t usbDev::deviceDescriptor::bLength
```

3.15.1.8 bMaxPacketSize0

```
uint8_t usbDev::deviceDescriptor::bMaxPacketSize0
```

3.15.1.9 bNumConfigurations

```
uint8_t usbDev::deviceDescriptor::bNumConfigurations
```

3.15.1.10 idProduct

```
uint16_t usbDev::deviceDescriptor::idProduct
```

3.15.1.11 idVendor

```
uint16_t usbDev::deviceDescriptor::idVendor
```

3.15.1.12 iManufacturer

```
uint8_t usbDev::deviceDescriptor::iManufacturer
```

3.15.1.13 iProduct

```
uint8_t usbDev::deviceDescriptor::iProduct
```

3.15.1.14 iSerialNumber

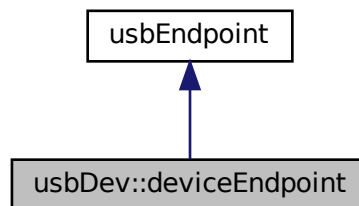
```
uint8_t usbDev::deviceDescriptor::iSerialNumber
```

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

- src/usbDev.h

3.16 usbDev::deviceEndpoint Class Reference

Inheritance diagram for usbDev::deviceEndpoint:



The diagram illustrates the structure of the libusb library, showing the relationships between various USB-related classes and their attributes. The classes are represented by boxes, and their attributes and methods are listed within or next to them. The relationships are indicated by solid lines with open arrows, while dashed lines represent associations or references.

Classes and their attributes:

- usbComponent:**
 - Attributes: `m_cngNameIdx`, `m_nEp`
 - Methods: `usb_ifcDesc`
- usbEndpoint:**
 - Attributes: `m_ep`, `m_subclass`
 - Methods: `epList`, `usbHardware`, `usbComponent`
- usbDev:**
 - Attributes: `parent`, `usbDev`
 - Methods: `usbDev`
- usbDeviceDescriptor:**
 - Attributes: `usbDeviceDescriptor`
 - Methods: `usbDeviceDescriptor`
- usbDeviceQualifier:**
 - Attributes: `usbDeviceQualifier`
 - Methods: `usbDeviceQualifier`
- usbDeviceEndpoint:**
 - Attributes: `usbDeviceEndpoint`
 - Methods: `usbDeviceEndpoint`
- usbDeviceLanguageDescriptor:**
 - Attributes: `usbDeviceLanguageDescriptor`
 - Methods: `usbDeviceLanguageDescriptor`
- usbDeviceStringDescriptor:**
 - Attributes: `usbDeviceStringDescriptor`
 - Methods: `usbDeviceStringDescriptor`
- usbEndpointDescriptor:**
 - Attributes: `usbEndpointDescriptor`
 - Methods: `usbEndpointDescriptor`
- usbInterfaceDescriptor:**
 - Attributes: `usbInterfaceDescriptor`
 - Methods: `usbInterfaceDescriptor`
- usbDeviceClass:**
 - Attributes: `usbDeviceClass`
 - Methods: `usbDeviceClass`
- usbDeviceProtocol:**
 - Attributes: `usbDeviceProtocol`
 - Methods: `usbDeviceProtocol`
- usbDeviceSubClass:**
 - Attributes: `usbDeviceSubClass`
 - Methods: `usbDeviceSubClass`
- usbDeviceConfiguration:**
 - Attributes: `usbDeviceConfiguration`
 - Methods: `usbDeviceConfiguration`
- usbDeviceManufacturer:**
 - Attributes: `usbDeviceManufacturer`
 - Methods: `usbDeviceManufacturer`
- usbDeviceProduct:**
 - Attributes: `usbDeviceProduct`
 - Methods: `usbDeviceProduct`
- usbDeviceSerialNumber:**
 - Attributes: `usbDeviceSerialNumber`
 - Methods: `usbDeviceSerialNumber`
- usbDeviceVendor:**
 - Attributes: `usbDeviceVendor`
 - Methods: `usbDeviceVendor`
- usbDeviceInterface:**
 - Attributes: `usbDeviceInterface`
 - Methods: `usbDeviceInterface`
- usbDeviceInterfaceClass:**
 - Attributes: `usbDeviceInterfaceClass`
 - Methods: `usbDeviceInterfaceClass`
- usbDeviceInterfaceProtocol:**
 - Attributes: `usbDeviceInterfaceProtocol`
 - Methods: `usbDeviceInterfaceProtocol`
- usbDeviceInterfaceSubClass:**
 - Attributes: `usbDeviceInterfaceSubClass`
 - Methods: `usbDeviceInterfaceSubClass`
- usbDeviceInterfaceConfiguration:**
 - Attributes: `usbDeviceInterfaceConfiguration`
 - Methods: `usbDeviceInterfaceConfiguration`
- usbDeviceInterfaceManufacturer:**
 - Attributes: `usbDeviceInterfaceManufacturer`
 - Methods: `usbDeviceInterfaceManufacturer`
- usbDeviceInterfaceProduct:**
 - Attributes: `usbDeviceInterfaceProduct`
 - Methods: `usbDeviceInterfaceProduct`
- usbDeviceInterfaceSerialNumber:**
 - Attributes: `usbDeviceInterfaceSerialNumber`
 - Methods: `usbDeviceInterfaceSerialNumber`
- usbDeviceInterfaceVendor:**
 - Attributes: `usbDeviceInterfaceVendor`
 - Methods: `usbDeviceInterfaceVendor`
- usbDeviceInterfaceInterface:**
 - Attributes: `usbDeviceInterfaceInterface`
 - Methods: `usbDeviceInterfaceInterface`
- usbDeviceInterfaceClass:**
 - Attributes: `usbDeviceInterfaceClass`
 - Methods: `usbDeviceInterfaceClass`
- usbDeviceInterfaceProtocol:**
 - Attributes: `usbDeviceInterfaceProtocol`
 - Methods: `usbDeviceInterfaceProtocol`
- usbDeviceInterfaceSubClass:**
 - Attributes: `usbDeviceInterfaceSubClass`
 - Methods: `usbDeviceInterfaceSubClass`
- usbDeviceInterfaceConfiguration:**
 - Attributes: `usbDeviceInterfaceConfiguration`
 - Methods: `usbDeviceInterfaceConfiguration`
- usbDeviceInterfaceManufacturer:**
 - Attributes: `usbDeviceInterfaceManufacturer`
 - Methods: `usbDeviceInterfaceManufacturer`
- usbDeviceInterfaceProduct:**
 - Attributes: `usbDeviceInterfaceProduct`
 - Methods: `usbDeviceInterfaceProduct`
- usbDeviceInterfaceSerialNumber:**
 - Attributes: `usbDeviceInterfaceSerialNumber`
 - Methods: `usbDeviceInterfaceSerialNumber`
- usbDeviceInterfaceVendor:**
 - Attributes: `usbDeviceInterfaceVendor`
 - Methods: `usbDeviceInterfaceVendor`
- usbDeviceInterfaceInterface:**
 - Attributes: `usbDeviceInterfaceInterface`
 - Methods: `usbDeviceInterfaceInterface`

Other elements:

- usbComponent::interfaceDescriptor**: A dashed line points to the `usbInterfaceDescriptor` class.
- usbEndpoint::descriptor**: A dashed line points to the `usbEndpointDescriptor` class.
- usbDev::parent**: A dashed line points to the `usbDev` class.
- usbDev::usbDev**: A dashed line points to the `usbDev` class.
- usbDeviceDescriptor::usbDeviceDescriptor**: A dashed line points to the `usbDeviceDescriptor` class.
- usbDeviceQualifier::usbDeviceQualifier**: A dashed line points to the `usbDeviceQualifier` class.
- usbDeviceEndpoint::usbDeviceEndpoint**: A dashed line points to the `usbDeviceEndpoint` class.
- usbDeviceLanguageDescriptor::usbDeviceLanguageDescriptor**: A dashed line points to the `usbDeviceLanguageDescriptor` class.
- usbDeviceStringDescriptor::usbDeviceStringDescriptor**: A dashed line points to the `usbDeviceStringDescriptor` class.
- usbEndpointDescriptor::usbEndpointDescriptor**: A dashed line points to the `usbEndpointDescriptor` class.
- usbInterfaceDescriptor::usbInterfaceDescriptor**: A dashed line points to the `usbInterfaceDescriptor` class.
- usbDeviceClass::usbDeviceClass**: A dashed line points to the `usbDeviceClass` class.
- usbDeviceProtocol::usbDeviceProtocol**: A dashed line points to the `usbDeviceProtocol` class.
- usbDeviceSubClass::usbDeviceSubClass**: A dashed line points to the `usbDeviceSubClass` class.
- usbDeviceConfiguration::usbDeviceConfiguration**: A dashed line points to the `usbDeviceConfiguration` class.
- usbDeviceManufacturer::usbDeviceManufacturer**: A dashed line points to the `usbDeviceManufacturer` class.
- usbDeviceProduct::usbDeviceProduct**: A dashed line points to the `usbDeviceProduct` class.
- usbDeviceSerialNumber::usbDeviceSerialNumber**: A dashed line points to the `usbDeviceSerialNumber` class.
- usbDeviceVendor::usbDeviceVendor**: A dashed line points to the `usbDeviceVendor` class.
- usbDeviceInterface::usbDeviceInterface**: A dashed line points to the `usbDeviceInterface` class.
- usbDeviceInterfaceClass::usbDeviceInterfaceClass**: A dashed line points to the `usbDeviceInterfaceClass` class.
- usbDeviceInterfaceProtocol::usbDeviceInterfaceProtocol**: A dashed line points to the `usbDeviceInterfaceProtocol` class.
- usbDeviceInterfaceSubClass::usbDeviceInterfaceSubClass**: A dashed line points to the `usbDeviceInterfaceSubClass` class.
- usbDeviceInterfaceConfiguration::usbDeviceInterfaceConfiguration**: A dashed line points to the `usbDeviceInterfaceConfiguration` class.
- usbDeviceInterfaceManufacturer::usbDeviceInterfaceManufacturer**: A dashed line points to the `usbDeviceInterfaceManufacturer` class.
- usbDeviceInterfaceProduct::usbDeviceInterfaceProduct**: A dashed line points to the `usbDeviceInterfaceProduct` class.
- usbDeviceInterfaceSerialNumber::usbDeviceInterfaceSerialNumber**: A dashed line points to the `usbDeviceInterfaceSerialNumber` class.
- usbDeviceInterfaceVendor::usbDeviceInterfaceVendor**: A dashed line points to the `usbDeviceInterfaceVendor` class.
- usbDeviceInterfaceInterface::usbDeviceInterfaceInterface**: A dashed line points to the `usbDeviceInterfaceInterface` class.

Other elements:

- usbComponent::m_cngNameIdx**: A dashed line points to the `usbComponent` class.
- usbComponent::m_nEp**: A dashed line points to the `usbComponent` class.
- usbEndpoint::m_ep**: A dashed line points to the `usbEndpoint` class.
- usbEndpoint::m_subclass**: A dashed line points to the `usbEndpoint` class.
- usbDev::parent**: A dashed line points to the `usbDev` class.
- usbDev::usbDev**: A dashed line points to the `usbDev` class.
- usbDeviceDescriptor::usbDeviceDescriptor**: A dashed line points to the `usbDeviceDescriptor` class.
- usbDeviceQualifier::usbDeviceQualifier**: A dashed line points to the `usbDeviceQualifier` class.
- usbDeviceEndpoint::usbDeviceEndpoint**: A dashed line points to the `usbDeviceEndpoint` class.
- usbDeviceLanguageDescriptor::usbDeviceLanguageDescriptor**: A dashed line points to the `usbDeviceLanguageDescriptor` class.
- usbDeviceStringDescriptor::usbDeviceStringDescriptor**: A dashed line points to the `usbDeviceStringDescriptor` class.
- usbEndpointDescriptor::usbEndpointDescriptor**: A dashed line points to the `usbEndpointDescriptor` class.
- usbInterfaceDescriptor::usbInterfaceDescriptor**: A dashed line points to the `usbInterfaceDescriptor` class.
- usbDeviceClass::usbDeviceClass**: A dashed line points to the `usbDeviceClass` class.
- usbDeviceProtocol::usbDeviceProtocol**: A dashed line points to the `usbDeviceProtocol` class.
- usbDeviceSubClass::usbDeviceSubClass**: A dashed line points to the `usbDeviceSubClass` class.
- usbDeviceConfiguration::usbDeviceConfiguration**: A dashed line points to the `usbDeviceConfiguration` class.
- usbDeviceManufacturer::usbDeviceManufacturer**: A dashed line points to the `usbDeviceManufacturer` class.
- usbDeviceProduct::usbDeviceProduct**: A dashed line points to the `usbDeviceProduct` class.
- usbDeviceSerialNumber::usbDeviceSerialNumber**: A dashed line points to the `usbDeviceSerialNumber` class.
- usbDeviceVendor::usbDeviceVendor**: A dashed line points to the `usbDeviceVendor` class.
- usbDeviceInterface::usbDeviceInterface**: A dashed line points to the `usbDeviceInterface` class.
- usbDeviceInterfaceClass::usbDeviceInterfaceClass**: A dashed line points to the `usbDeviceInterfaceClass`

- enum class `dataTransferDirection` : `uint8_t` { `toDevice` = 0 , `toHost` = 1 }
- enum class `recipientType` : `uint8_t` { `device` = 0 , `interface` = 1 , `endpoint` = 2 , `other` = 3 }
- enum class `requestType` : `uint8_t` { `standard` = 0 , `classObj` = 1 , `vendor` = 2 , `reserved` = 3 }
- enum class `standardFeatureSelector` : `uint8_t` { `deviceRemoteWakeup` = 1 , `endpointHalt` = 0 , `testMode` = 2 }

- `deviceEndpoint ()`
- `void dataRecieved (uint16_t nBytes)`
- `void setupRecieved (uint16_t nBytes)`

- `usbDev * parent`

3.16.1.1 dataTransferDirection

Generated by Doxygen

Enumerator

toDevice	
toHost	

3.16.1.2 recipientType

```
enum usbDev::deviceEndpoint::recipientType : uint8_t [strong]
```

Enumerator

device	
interface	
endpoint	
other	

3.16.1.3 requestType

```
enum usbDev::deviceEndpoint::requestType : uint8_t [strong]
```

Enumerator

standard	
classObj	
vendor	
reserved	

3.16.1.4 standardFeatureSelector

```
enum usbDev::deviceEndpoint::standardFeatureSelector : uint8_t [strong]
```

Enumerator

deviceRemoteWakeup	
endpointHalt	
testMode	

3.16.2 Constructor & Destructor Documentation

3.16.2.1 deviceEndpoint()

```
usbDev::deviceEndpoint::deviceEndpoint ( )
```

3.16.3 Member Function Documentation

3.16.3.1 dataRecieved()

```
void usbDev::deviceEndpoint::dataRecieved (
    uint16_t nBytes ) [virtual]
```

Implements [usbEndpoint](#).

3.16.3.2 setupRecieved()

```
void usbDev::deviceEndpoint::setupRecieved (
    uint16_t nBytes ) [virtual]
```

Reimplemented from [usbEndpoint](#).

3.16.4 Member Data Documentation

3.16.4.1 parent

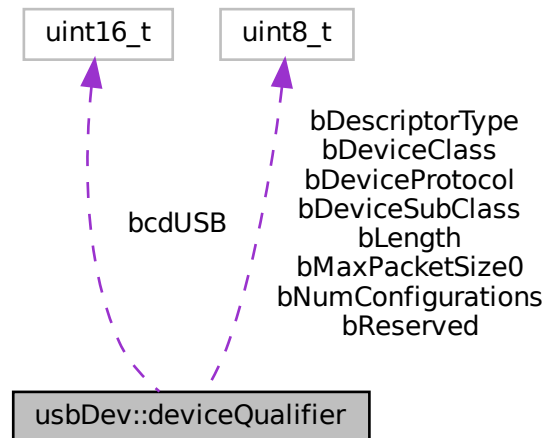
```
usbDev* usbDev::deviceEndpoint::parent
```

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

- src/usbDev.h
- src/usbDev.cpp

3.17 usbDev::deviceQualifier Struct Reference

Collaboration diagram for usbDev::deviceQualifier:



Public Attributes

- uint16_t [bcdUSB](#)
- uint8_t [bDescriptorType](#)
- uint8_t [bDeviceClass](#)
- uint8_t [bDeviceProtocol](#)
- uint8_t [bDeviceSubClass](#)
- uint8_t [bLength](#)
- uint8_t [bMaxPacketSize0](#)
- uint8_t [bNumConfigurations](#)
- uint8_t [bReserved](#)

3.17.1 Member Data Documentation

3.17.1.1 bcdUSB

```
uint16_t usbDev::deviceQualifier::bcdUSB
```

3.17.1.2 bDescriptorType

```
uint8_t usbDev::deviceQualifier::bDescriptorType
```

3.17.1.3 bDeviceClass

```
uint8_t usbDev::deviceQualifier::bDeviceClass
```

3.17.1.4 bDeviceProtocol

```
uint8_t usbDev::deviceQualifier::bDeviceProtocol
```

3.17.1.5 bDeviceSubClass

```
uint8_t usbDev::deviceQualifier::bDeviceSubClass
```

3.17.1.6 bLength

```
uint8_t usbDev::deviceQualifier::bLength
```

3.17.1.7 bMaxPacketSize0

```
uint8_t usbDev::deviceQualifier::bMaxPacketSize0
```

3.17.1.8 bNumConfigurations

```
uint8_t usbDev::deviceQualifier::bNumConfigurations
```

3.17.1.9 bReserved

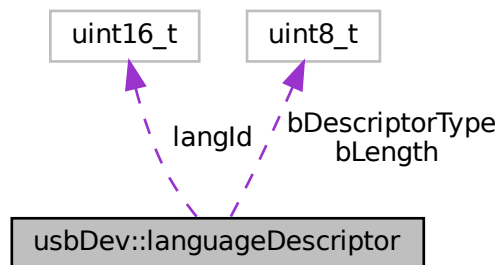
```
uint8_t usbDev::deviceQualifier::bReserved
```

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

- src/usbDev.h

3.18 usbDev::languageDescriptor Struct Reference

Collaboration diagram for usbDev::languageDescriptor:



Public Attributes

- [uint8_t bDescriptorType](#)
- [uint8_t bLength](#)
- [uint16_t langId](#) [1]

3.18.1 Member Data Documentation

3.18.1.1 bDescriptorType

```
uint8_t usbDev::languageDescriptor::bDescriptorType
```

3.18.1.2 bLength

```
uint8_t usbDev::languageDescriptor::bLength
```


3.18.1.3 langId

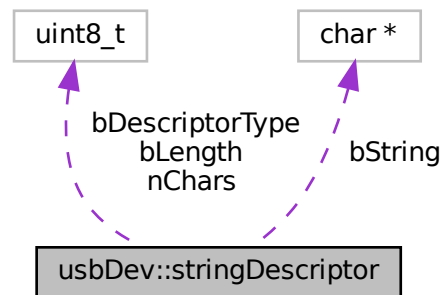
```
uint16_t usbDev::languageDescriptor::langId[1]
```

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

- src/usbDev.h

3.19 usbDev::stringDescriptor Class Reference

Collaboration diagram for usbDev::stringDescriptor:



Public Member Functions

- [stringDescriptor](#) ()
- bool [hasString](#) ()
- void [setString](#) (const char *string)
- void [writeDescriptor](#) (uint8_t *buffer, uint16_t *len)

Private Attributes

- uint8_t [bDescriptorType](#)
- uint8_t [bLength](#)
- char * [bString](#)
- uint8_t [nChars](#)

3.19.1 Constructor & Destructor Documentation

3.19.1.1 stringDescriptor()

```
usbDev::stringDescriptor::stringDescriptor ( )
```

3.19.2 Member Function Documentation

3.19.2.1 hasString()

```
bool usbDev::stringDescriptor::hasString ( )
```

3.19.2.2 setString()

```
void usbDev::stringDescriptor::setString (
    const char * string )
```

3.19.2.3 writeDescriptor()

```
void usbDev::stringDescriptor::writeDescriptor (
    uint8_t * buffer,
    uint16_t * len )
```

3.19.3 Member Data Documentation

3.19.3.1 bDescriptorType

```
uint8_t usbDev::stringDescriptor::bDescriptorType [private]
```

3.19.3.2 bLength

```
uint8_t usbDev::stringDescriptor::bLength [private]
```

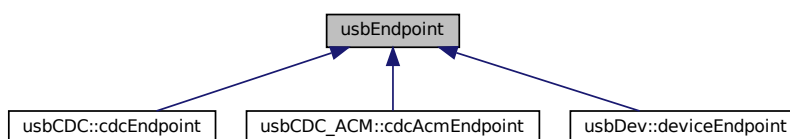
```
char* usbDev::stringDescriptor::bString [private]
```

```
uint8_t usbDev::stringDescriptor::nChars [private]
```

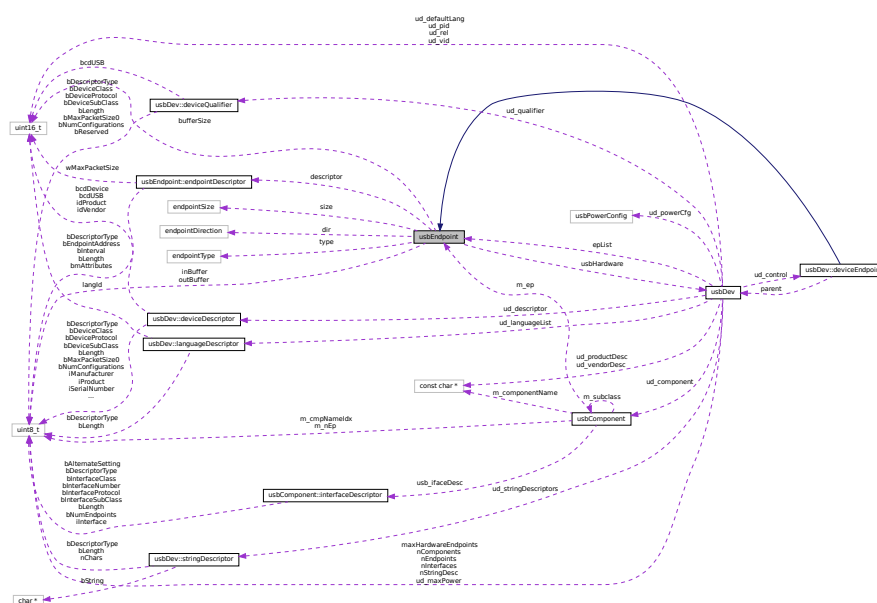
- src/usbDev.h
- src/usbDev.cpp

```
#include <usbEndpoint.h>
```

Inheritance diagram for usbEndpoint:



Collaboration diagram for usbEndpoint:



Classes

- struct [endpointDescriptor](#)

Public Types

- enum class [endpointDirection](#) : uint8_t { [in](#) =0x00 , [out](#) =0x01 }
- enum class [endpointSize](#) : uint16_t {
[b8](#) =8 , [b16](#) =16 , [b32](#) =32 , [b64](#) =64 ,
[b128](#) =128 , [b256](#) =256 , [b512](#) =512 , [b1023](#) =1023 }
- enum class [endpointType](#) : uint8_t {
[control](#) =0x00 , [isochronous](#) =0x01 , [bulk](#) =0x02 , [interrupt](#) =0x03 ,
[dual](#) =0x04 }

Public Member Functions

- [usbEndpoint](#) (uint16_t bufferSize, [usbEndpoint::endpointSize](#) sz, [usbEndpoint::endpointDirection](#) uDir, [usbEndpoint::endpointType](#) uType)
- virtual void [dataRecieved](#) (uint16_t nBytes)=0
- uint8_t [hardwareEndpoint](#) ()
- void [setHardware](#) (usbDev *hw)
- void [setHardwareEndpoint](#) (uint8_t [hardwareEndpoint](#))
- virtual void [setupRecieved](#) (uint16_t nBytes)
- uint16_t [writeIn](#) (uint8_t *data, uint16_t nBytes, uint16_t maxLength=0xFFFF)
- void [writeInStall](#) ()
- void [writeInZLP](#) ()

Public Attributes

- uint16_t [bufferSize](#)
- [endpointDescriptor](#) descriptor
- [usbEndpoint::endpointDirection](#) dir
- uint8_t * [inBuffer](#)
- uint8_t * [outBuffer](#)
- [usbEndpoint::endpointSize](#) size
- [usbEndpoint::endpointType](#) type
- usbDev * [usbHardware](#)

3.20.1 Member Enumeration Documentation

3.20.1.1 endpointDirection

```
enum usbEndpoint::endpointDirection : uint8_t [strong]
```

Enumerator

in	
out	

3.20.1.2 endpointSize

```
enum usbEndpoint::endpointSize : uint16_t [strong]
```

Enumerator

b8	
b16	
b32	
b64	
b128	
b256	
b512	
b1023	

3.20.1.3 endpointType

```
enum usbEndpoint::endpointType : uint8_t [strong]
```

Enumerator

control	
isochronos	
bulk	
interrupt	
dual	

3.20.2 Constructor & Destructor Documentation

3.20.2.1 usbEndpoint()

```
usbEndpoint::usbEndpoint (
    uint16_t bufferSize,
    usbEndpoint::endpointSize sz,
    usbEndpoint::endpointDirection uDir,
    usbEndpoint::endpointType uType )
```

ToDo: Handle Isochronos configuration of bmAttributes bits 2 to 5.

3.20.3 Member Function Documentation

3.20.3.1 dataRecieved()

```
virtual void usbEndpoint::dataRecieved (
    uint16_t nBytes ) [pure virtual]
```

Implemented in [usbDev::deviceEndpoint](#), [usbCDC_ACM::cdcAcmEndpoint](#), and [usbCDC::cdcEndpoint](#).

3.20.3.2 hardwareEndpoint()

```
uint8_t usbEndpoint::hardwareEndpoint ( )
```

3.20.3.3 setHardware()

```
void usbEndpoint::setHardware (
    usbDev * hw )
```

3.20.3.4 setHardwareEndpoint()

```
void usbEndpoint::setHardwareEndpoint (
    uint8_t hardwareEndpoint )
```

3.20.3.5 setupRecieved()

```
void usbEndpoint::setupRecieved (
    uint16_t nBytes ) [virtual]
```

Reimplemented in [usbDev::deviceEndpoint](#).

3.20.3.6 writeIn()

```
uint16_t usbEndpoint::writeIn (
    uint8_t * data,
    uint16_t nBytes,
    uint16_t maxLength = 0xFFFF )
```

3.20.3.7 writeInStall()

```
void usbEndpoint::writeInStall ( )
```

3.20.3.8 writeInZLP()

```
void usbEndpoint::writeInZLP ( )
```

3.20.4 Member Data Documentation

3.20.4.1 bufferSize

```
uint16_t usbEndpoint::bufferSize
```

3.20.4.2 descriptor

```
endpointDescriptor usbEndpoint::descriptor
```

3.20.4.3 dir

```
usbEndpoint::endpointDirection usbEndpoint::dir
```

3.20.4.4 inBuffer

```
uint8_t* usbEndpoint::inBuffer
```

3.20.4.5 outBuffer

```
uint8_t* usbEndpoint::outBuffer
```

3.20.4.6 size

```
usbEndpoint::endpointSize usbEndpoint::size
```

3.20.4.7 type

```
usbEndpoint::endpointType usbEndpoint::type
```

3.20.4.8 usbHardware

```
usbDev* usbEndpoint::usbHardware
```

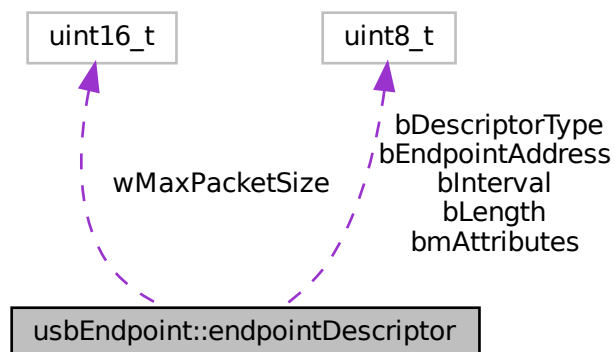
The documentation for this class was generated from the following files:

- src/usbEndpoint.h
- src/usbEndpoint.cpp

3.21 usbEndpoint::endpointDescriptor Struct Reference

```
#include <usbEndpoint.h>
```

Collaboration diagram for usbEndpoint::endpointDescriptor:



Public Attributes

- uint8_t [bDescriptorType](#)
- uint8_t [bEndpointAddress](#)
- uint8_t [bInterval](#)
- uint8_t [bLength](#)
- uint8_t [bmAttributes](#)
- uint16_t [wMaxPacketSize](#)

3.21.1 Member Data Documentation

3.21.1.1 bDescriptorType

uint8_t usbEndpoint::endpointDescriptor::bDescriptorType

3.21.1.2 bEndpointAddress

uint8_t usbEndpoint::endpointDescriptor::bEndpointAddress

3.21.1.3 bInterval

uint8_t usbEndpoint::endpointDescriptor::bInterval

3.21.1.4 bLength

uint8_t usbEndpoint::endpointDescriptor::bLength

3.21.1.5 bmAttributes

uint8_t usbEndpoint::endpointDescriptor::bmAttributes

3.21.1.6 wMaxPacketSize

```
uint16_t usbEndpoint::endpointDescriptor::wMaxPacketSize
```

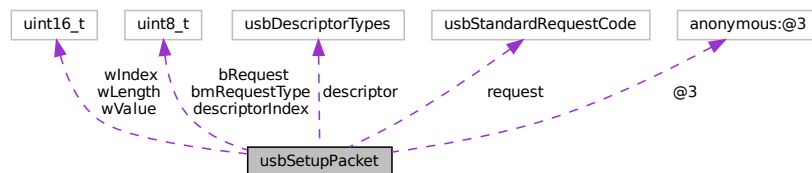
The documentation for this struct was generated from the following file:

- src/usbEndpoint.h

3.22 usbSetupPacket Class Reference

```
#include <usbSetupPacket.h>
```

Collaboration diagram for usbSetupPacket:



Public Member Functions

- [usbSetupPacket](#) (uint8_t *data)

Public Attributes

- union {
 - uint8_t [bRequest](#)
 - usbStandardRequestCode [request](#)
- uint8_t [bmRequestType](#)
- usbDescriptorTypes [descriptor](#)
- uint8_t [descriptorIndex](#)
- uint16_t [wIndex](#)
- uint16_t [wLength](#)
- uint16_t [wValue](#)

3.22.1 Constructor & Destructor Documentation

3.22.1.1 usbSetupPacket()

```
usbSetupPacket::usbSetupPacket (
    uint8_t * data )
```

3.22.2 Member Data Documentation

3.22.2.1

```
union { ... }
```

3.22.2.2 bmRequestType

```
uint8_t usbSetupPacket::bmRequestType
```

3.22.2.3 bRequest

```
uint8_t usbSetupPacket::bRequest
```

3.22.2.4 descriptor

```
usbDescriptorTypes usbSetupPacket::descriptor
```

3.22.2.5 descriptorIndex

```
uint8_t usbSetupPacket::descriptorIndex
```

3.22.2.6 request

```
usbStandardRequestCode usbSetupPacket::request
```

3.22.2.7 wIndex

`uint16_t usbSetupPacket::wIndex`

3.22.2.8 wLength

`uint16_t usbSetupPacket::wLength`

3.22.2.9 wValue

`uint16_t usbSetupPacket::wValue`

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

- `src/packets/usbSetupPacket.h`
- `src/packets/usbSetupPacket.cpp`

Index

- acm_mgtDesc
 - usbCDC_ACM, [23](#)
- addComponent
 - usbDev, [36](#)
- addComponentEndpoints
 - usbDev, [36](#)
- addEndpoint
 - usbDev, [36](#)
- addEndpointToHardware
 - samd21usbDevice, [8](#)
 - usbDev, [36](#)
- addStringDescriptor
 - usbDev, [36](#)
- assignEndpoint
 - usbComponent, [29](#)
- assignInterfaceNumbers
 - usbComponent, [29](#)
- attach
 - samd21usbDevice, [8](#)
 - usbDev, [36](#)
- b1023
 - usbEndpoint, [59](#)
- b128
 - usbEndpoint, [59](#)
- b16
 - usbEndpoint, [59](#)
- b256
 - usbEndpoint, [59](#)
- b32
 - usbEndpoint, [59](#)
- b512
 - usbEndpoint, [59](#)
- b64
 - usbEndpoint, [59](#)
- b8
 - usbEndpoint, [59](#)
- bAlternateSetting
 - usbComponent::interfaceDescriptor, [32](#)
- bank
 - hwEpDesc, [6](#)
- bcdCDC
 - usbCDC::headerDescriptor, [20](#)
- bcdDevice
 - usbDev::deviceDescriptor, [46](#)
- bcdUSB
 - usbDev::deviceDescriptor, [46](#)
 - usbDev::deviceQualifier, [52](#)
- bConfigurationValue
 - usbDev::configurationDescriptor, [44](#)
- bControlInterface
 - usbCDC::functionalDescriptor, [18](#)
- bDataInterface
 - usbCDC::capabilityDescriptor, [15](#)
- bDescriptorSubtype
 - usbCDC::capabilityDescriptor, [15](#)
 - usbCDC::functionalDescriptor, [18](#)
 - usbCDC::headerDescriptor, [20](#)
 - usbCDC_ACM::managementDescriptor, [26](#)
- bDescriptorType
 - usbCDC::capabilityDescriptor, [15](#)
 - usbCDC::functionalDescriptor, [19](#)
 - usbCDC::headerDescriptor, [20](#)
 - usbCDC_ACM::managementDescriptor, [26](#)
 - usbComponent::interfaceDescriptor, [32](#)
 - usbDev::configurationDescriptor, [44](#)
 - usbDev::deviceDescriptor, [46](#)
 - usbDev::deviceQualifier, [52](#)
 - usbDev::languageDescriptor, [54](#)
 - usbDev::stringDescriptor, [56](#)
 - usbEndpoint::endpointDescriptor, [63](#)
- bDeviceClass
 - usbDev::deviceDescriptor, [46](#)
 - usbDev::deviceQualifier, [53](#)
- bDeviceProtocol
 - usbDev::deviceDescriptor, [46](#)
 - usbDev::deviceQualifier, [53](#)
- bDeviceSubClass
 - usbDev::deviceDescriptor, [47](#)
 - usbDev::deviceQualifier, [53](#)
- bEndpointAddress
 - usbEndpoint::endpointDescriptor, [63](#)
- bFunctionLength
 - usbCDC::capabilityDescriptor, [15](#)
 - usbCDC::functionalDescriptor, [19](#)
 - usbCDC::headerDescriptor, [21](#)
 - usbCDC_ACM::managementDescriptor, [26](#)
- bInterfaceClass
 - usbCDC::functionalDescriptor, [19](#)
 - usbComponent::interfaceDescriptor, [32](#)
- bInterfaceNumber
 - usbComponent::interfaceDescriptor, [32](#)
- bInterfaceProtocol
 - usbCDC::functionalDescriptor, [19](#)
 - usbComponent::interfaceDescriptor, [32](#)
- bInterfaceSubClass
 - usbCDC::functionalDescriptor, [19](#)
 - usbComponent::interfaceDescriptor, [32](#)
- bInterval

- usbEndpoint::endpointDescriptor, 63
- bLength
 - usbComponent::interfaceDescriptor, 33
 - usbDev::configurationDescriptor, 44
 - usbDev::deviceDescriptor, 47
 - usbDev::deviceQualifier, 53
 - usbDev::languageDescriptor, 54
 - usbDev::stringDescriptor, 56
 - usbEndpoint::endpointDescriptor, 63
- bmAttributes
 - usbDev::configurationDescriptor, 44
 - usbEndpoint::endpointDescriptor, 63
- bMaxPacketSize0
 - usbDev::deviceDescriptor, 47
 - usbDev::deviceQualifier, 53
- bMaxPower
 - usbDev::configurationDescriptor, 44
- bmCapabilities
 - usbCDC::capabilityDescriptor, 15
 - usbCDC_ACM::managementDescriptor, 27
- bmRequestType
 - usbSetupPacket, 65
- bNumConfigurations
 - usbDev::deviceDescriptor, 47
 - usbDev::deviceQualifier, 53
- bNumEndpoints
 - usbComponent::interfaceDescriptor, 33
- bNumInterfaces
 - usbDev::configurationDescriptor, 44
- bRequest
 - usbSetupPacket, 65
- bReserved
 - usbDev::deviceQualifier, 53
- bString
 - usbDev::stringDescriptor, 56
- bSubordinateInterface0
 - usbCDC::functionalDescriptor, 19
- bufferFunctionalDescriptor
 - usbCDC, 13
 - usbCDC_ACM, 22
 - usbComponent, 29
- bufferInterfaceDescriptor
 - usbComponent, 29
- bufferSize
 - usbEndpoint, 61
- bulk
 - usbEndpoint, 59
- cdc_fncDesc
 - usbCDC, 13
- cdc_hdrDesc
 - usbCDC, 13
- cdcAcmEndpoint
 - usbCDC, 13
 - usbCDC_ACM, 23
 - usbCDC_ACM::cdcAcmEndpoint, 25
- cdcEndpoint
 - usbCDC::cdcEndpoint, 17
- classObj
 - usbDev::deviceEndpoint, 50
- componentEP
 - usbDev, 37
- componentEPs
 - usbDev, 37
- configDescriptor
 - usbDev, 37
- configureClock
 - samd21 usbDevice, 8
- configureEndpoint
 - samd21 usbDevice, 8
- configurePins
 - samd21 usbDevice, 9
- control
 - usbEndpoint, 59
- ctrlEndpoint
 - usbCDC, 14
- dataRecieved
 - usbCDC::cdcEndpoint, 17
 - usbCDC_ACM::cdcAcmEndpoint, 25
 - usbDev::deviceEndpoint, 51
 - usbEndpoint, 60
- dataTransferDirection
 - usbDev::deviceEndpoint, 49
- descriptor
 - usbEndpoint, 61
 - usbSetupPacket, 65
- descriptorIndex
 - usbSetupPacket, 65
- device
 - usbDev::deviceEndpoint, 50
- deviceEndpoint
 - usbDev, 40
 - usbDev::deviceEndpoint, 51
- deviceRemoteWakeup
 - usbDev::deviceEndpoint, 50
- dInEp
 - usbCDC_ACM, 23
- dir
 - usbEndpoint, 61
- dOutEp
 - usbCDC_ACM, 23
- dual
 - usbEndpoint, 59
- endpoint
 - usbDev::deviceEndpoint, 50
- endpointDirection
 - usbEndpoint, 58
- endpointHalt
 - usbDev::deviceEndpoint, 50
- endpoints
 - usbComponent, 29
- endpointSize
 - usbEndpoint, 59
- endpointType
 - usbEndpoint, 59
- epList

- usbDev, 40
- epSizeCode
 - samd21usbDevice, 9
- exec
 - usbComponent, 29
 - usbDev, 37
- hardwareEndpoint
 - usbEndpoint, 60
- hardwareExec
 - samd21usbDevice, 9
 - usbDev, 37
- hasString
 - usbDev::stringDescriptor, 56
- hwEpDesc, 5
 - bank, 6
 - in, 6
 - out, 6
- iConfiguration
 - usbDev::configurationDescriptor, 45
- idProduct
 - usbDev::deviceDescriptor, 47
- idVendor
 - usbDev::deviceDescriptor, 47
- iInterface
 - usbComponent::interfaceDescriptor, 33
- iManufacturer
 - usbDev::deviceDescriptor, 47
- in
 - hwEpDesc, 6
 - usbEndpoint, 58
- inBuffer
 - usbEndpoint, 61
- initialise
 - usbDev, 37
- initialiseHardware
 - samd21usbDevice, 9
 - usbDev, 38
- interface
 - usbDev::deviceEndpoint, 50
- interrupt
 - usbEndpoint, 59
- interruptHandler
 - samd21usbDevice, 9
- iProduct
 - usbDev::deviceDescriptor, 47
- iSerialNumber
 - usbDev::deviceDescriptor, 48
- isochronous
 - usbEndpoint, 59
- langId
 - usbDev::languageDescriptor, 54
- m_cmpNameldx
 - usbComponent, 30
- m_componentName
 - usbComponent, 30
- m_ep
 - usbComponent, 30
- m_nEp
 - usbComponent, 31
- m_subclass
 - usbComponent, 31
- maxHardwareEndpoints
 - usbDev, 41
- nChars
 - usbDev::stringDescriptor, 57
- nComponents
 - usbDev, 41
- nEndpoints
 - usbDev, 41
- nInterfaces
 - usbDev, 41
- nStringDesc
 - usbDev, 41
- other
 - usbDev::deviceEndpoint, 50
- out
 - hwEpDesc, 6
 - usbEndpoint, 58
- outBuffer
 - usbEndpoint, 61
- parent
 - usbCDC::cdcEndpoint, 17
 - usbCDC_ACM::cdcAcmEndpoint, 25
 - usbDev::deviceEndpoint, 51
- populateDescriptor
 - usbDev, 38
- protocolCode
 - usbComponent, 29
- readComplete
 - samd21usbDevice, 9
- recipientType
 - usbDev::deviceEndpoint, 50
- request
 - usbSetupPacket, 65
- requestType
 - usbDev::deviceEndpoint, 50
- reserved
 - usbDev::deviceEndpoint, 50
- samd21usbDevice, 6
 - addEndpointToHardware, 8
 - attach, 8
 - configureClock, 8
 - configureEndpoint, 8
 - configurePins, 9
 - epSizeCode, 9
 - hardwareExec, 9
 - initialiseHardware, 9
 - interruptHandler, 9
 - readComplete, 9

- samd21usbDevice, [8](#)
- sendRequestError, [9](#)
- setAddress, [10](#)
- setMode, [10](#)
- USB_Handler, [11](#)
- writeIn, [10](#)
- writeInStall, [10](#)
- writeInZLP, [10](#)
- sendRequestError
 - samd21usbDevice, [9](#)
- setAddress
 - samd21usbDevice, [10](#)
 - usbDev, [38](#)
- setConfigurationString
 - usbDev, [38](#)
- setHardware
 - usbEndpoint, [60](#)
- setHardwareEndpoint
 - usbEndpoint, [60](#)
- setMaxPower
 - usbDev, [38](#)
- setMode
 - samd21usbDevice, [10](#)
 - usbDev, [38](#)
- setProductID
 - usbDev, [38](#)
- setProductString
 - usbDev, [39](#)
- setProductVersion
 - usbDev, [39](#)
- setSerialNumberString
 - usbDev, [39](#)
- setString
 - usbDev::stringDescriptor, [56](#)
- setupRecieved
 - usbDev::deviceEndpoint, [51](#)
 - usbEndpoint, [60](#)
- setVendorID
 - usbDev, [39](#)
- setVendorString
 - usbDev, [39](#)
- size
 - usbEndpoint, [61](#)
- standard
 - usbDev::deviceEndpoint, [50](#)
- standardFeatureSelector
 - usbDev::deviceEndpoint, [50](#)
- stringDescriptor
 - usbDev::stringDescriptor, [55](#)
- subclassCode
 - usbCDC_ACM, [22](#)
 - usbComponent, [30](#)
- testMode
 - usbDev::deviceEndpoint, [50](#)
- toDevice
 - usbDev::deviceEndpoint, [50](#)
- toHost
 - usbDev::deviceEndpoint, [50](#)
- type
 - usbEndpoint, [62](#)
- ud_component
 - usbDev, [41](#)
- ud_control
 - usbDev, [41](#)
- ud_defaultLang
 - usbDev, [41](#)
- ud_descriptor
 - usbDev, [42](#)
- ud_languageList
 - usbDev, [42](#)
- ud_maxPower
 - usbDev, [42](#)
- ud_pid
 - usbDev, [42](#)
- ud_powerCfg
 - usbDev, [42](#)
- ud_productDesc
 - usbDev, [42](#)
- ud_qualifier
 - usbDev, [42](#)
- ud_rel
 - usbDev, [42](#)
- ud_stringDescriptors
 - usbDev, [43](#)
- ud_vendorDesc
 - usbDev, [43](#)
- ud_vid
 - usbDev, [43](#)
- updateInterfaceDescriptor
 - usbCDC_ACM, [23](#)
 - usbComponent, [30](#)
- USB_Handler
 - samd21usbDevice, [11](#)
- usb_ifaceDesc
 - usbComponent, [31](#)
- usbCDC, [11](#)
 - bufferFunctionalDescriptor, [13](#)
 - cdc_fncDesc, [13](#)
 - cdc_hdrDesc, [13](#)
 - cdcAcmEndpoint, [13](#)
 - ctrlEndpoint, [14](#)
 - usbCDC, [13](#)
 - usbCDC_ACM, [23](#)
- usbCDC::capabilityDescriptor, [14](#)
 - bDataInterface, [15](#)
 - bDescriptorSubtype, [15](#)
 - bDescriptorType, [15](#)
 - bFunctionLength, [15](#)
 - bmCapabilities, [15](#)
- usbCDC::cdcEndpoint, [16](#)
 - cdcEndpoint, [17](#)
 - dataRecieved, [17](#)
 - parent, [17](#)
- usbCDC::functionalDescriptor, [18](#)
 - bControllInterface, [18](#)
 - bDescriptorSubtype, [18](#)

- bDescriptorType, 19
- bFunctionLength, 19
- bInterfaceClass, 19
- bInterfaceProtocol, 19
- bInterfaceSubClass, 19
- bSubordinateInterface0, 19
- usbCDC::headerDescriptor, 20
 - bcdCDC, 20
 - bDescriptorSubtype, 20
 - bDescriptorType, 20
 - bFunctionLength, 21
- usbCDC_ACM, 21
 - acm_mgtDesc, 23
 - bufferFunctionalDescriptor, 22
 - cdcAcmEndpoint, 23
 - dInEp, 23
 - dOutEp, 23
 - subclassCode, 22
 - updateInterfaceDescriptor, 23
 - usbCDC, 23
 - usbCDC_ACM, 22
- usbCDC_ACM::cdcAcmEndpoint, 24
 - cdcAcmEndpoint, 25
 - dataRecieved, 25
 - parent, 25
- usbCDC_ACM::managementDescriptor, 26
 - bDescriptorSubtype, 26
 - bDescriptorType, 26
 - bFunctionLength, 26
 - bmCapabilities, 27
- usbComponent, 27
 - assignEndpoint, 29
 - assignInterfaceNumbers, 29
 - bufferFunctionalDescriptor, 29
 - bufferInterfaceDescriptor, 29
 - endpoints, 29
 - exec, 29
 - m_cmpNameldx, 30
 - m_componentName, 30
 - m_ep, 30
 - m_nEp, 31
 - m_subclass, 31
 - protocolCode, 29
 - subclassCode, 30
 - updateInterfaceDescriptor, 30
 - usb_ifaceDesc, 31
 - usbComponent, 28
 - usbDev, 30
- usbComponent::interfaceDescriptor, 31
 - bAlternateSetting, 32
 - bDescriptorType, 32
 - bInterfaceClass, 32
 - bInterfaceNumber, 32
 - bInterfaceProtocol, 32
 - bInterfaceSubClass, 32
 - bLength, 33
 - bNumEndpoints, 33
 - iInterface, 33
- usbDev, 33
 - addComponent, 36
 - addComponentEndpoints, 36
 - addEndpoint, 36
 - addEndpointToHardware, 36
 - addStringDescriptor, 36
 - attach, 36
 - componentEP, 37
 - componentEPs, 37
 - configDescriptor, 37
 - deviceEndpoint, 40
 - epList, 40
 - exec, 37
 - hardwareExec, 37
 - initialise, 37
 - initialiseHardware, 38
 - maxHardwareEndpoints, 41
 - nComponents, 41
 - nEndpoints, 41
 - nInterfaces, 41
 - nStringDesc, 41
 - populateDescriptor, 38
 - setAddress, 38
 - setConfigurationString, 38
 - setMaxPower, 38
 - setMode, 38
 - setProductID, 38
 - setProductString, 39
 - setProductVersion, 39
 - setSerialNumberString, 39
 - setVendorID, 39
 - setVendorString, 39
 - ud_component, 41
 - ud_control, 41
 - ud_defaultLang, 41
 - ud_descriptor, 42
 - ud_languageList, 42
 - ud_maxPower, 42
 - ud_pid, 42
 - ud_powerCfg, 42
 - ud_productDesc, 42
 - ud_qualifier, 42
 - ud_rel, 42
 - ud_stringDescriptors, 43
 - ud_vendorDesc, 43
 - ud_vid, 43
 - usbComponent, 30
 - usbDev, 35
 - usbEndpoint, 40
 - writeln, 39
 - writelnStall, 40
 - writelnZLP, 40
- usbDev::configurationDescriptor, 43
 - bConfigurationValue, 44
 - bDescriptorType, 44
 - bLength, 44
 - bmAttributes, 44
 - bMaxPower, 44

- bNumInterfaces, [44](#)
 - iConfiguration, [45](#)
 - wTotalLength, [45](#)
- usbDev::deviceDescriptor, [45](#)
 - bcdDevice, [46](#)
 - bcdUSB, [46](#)
 - bDescriptorType, [46](#)
 - bDeviceClass, [46](#)
 - bDeviceProtocol, [46](#)
 - bDeviceSubClass, [47](#)
 - bLength, [47](#)
 - bMaxPacketSize0, [47](#)
 - bNumConfigurations, [47](#)
 - idProduct, [47](#)
 - idVendor, [47](#)
 - iManufacturer, [47](#)
 - iProduct, [47](#)
 - iSerialNumber, [48](#)
- usbDev::deviceEndpoint, [48](#)
 - classObj, [50](#)
 - dataRecieved, [51](#)
 - dataTransferDirection, [49](#)
 - device, [50](#)
 - deviceEndpoint, [51](#)
 - deviceRemoteWakeUp, [50](#)
 - endpoint, [50](#)
 - endpointHalt, [50](#)
 - interface, [50](#)
 - other, [50](#)
 - parent, [51](#)
 - recipientType, [50](#)
 - requestType, [50](#)
 - reserved, [50](#)
 - setupRecieved, [51](#)
 - standard, [50](#)
 - standardFeatureSelector, [50](#)
 - testMode, [50](#)
 - toDevice, [50](#)
 - toHost, [50](#)
 - vendor, [50](#)
- usbDev::deviceQualifier, [52](#)
 - bcdUSB, [52](#)
 - bDescriptorType, [52](#)
 - bDeviceClass, [53](#)
 - bDeviceProtocol, [53](#)
 - bDeviceSubClass, [53](#)
 - bLength, [53](#)
 - bMaxPacketSize0, [53](#)
 - bNumConfigurations, [53](#)
 - bReserved, [53](#)
- usbDev::languageDescriptor, [54](#)
 - bDescriptorType, [54](#)
 - bLength, [54](#)
 - langId, [54](#)
- usbDev::stringDescriptor, [55](#)
 - bDescriptorType, [56](#)
 - bLength, [56](#)
 - bString, [56](#)
 - hasString, [56](#)
 - nChars, [57](#)
 - setString, [56](#)
 - stringDescriptor, [55](#)
 - writeDescriptor, [56](#)
- usbEndpoint, [57](#)
 - b1023, [59](#)
 - b128, [59](#)
 - b16, [59](#)
 - b256, [59](#)
 - b32, [59](#)
 - b512, [59](#)
 - b64, [59](#)
 - b8, [59](#)
 - bufferSize, [61](#)
 - bulk, [59](#)
 - control, [59](#)
 - dataRecieved, [60](#)
 - descriptor, [61](#)
 - dir, [61](#)
 - dual, [59](#)
 - endpointDirection, [58](#)
 - endpointSize, [59](#)
 - endpointType, [59](#)
 - hardwareEndpoint, [60](#)
 - in, [58](#)
 - inBuffer, [61](#)
 - interrupt, [59](#)
 - isochronous, [59](#)
 - out, [58](#)
 - outBuffer, [61](#)
 - setHardware, [60](#)
 - setHardwareEndpoint, [60](#)
 - setupRecieved, [60](#)
 - size, [61](#)
 - type, [62](#)
 - usbDev, [40](#)
 - usbEndpoint, [59](#)
 - usbHardware, [62](#)
 - writeln, [60](#)
 - writelnInstall, [60](#)
 - writelnZLP, [61](#)
- usbEndpoint::endpointDescriptor, [62](#)
 - bDescriptorType, [63](#)
 - bEndpointAddress, [63](#)
 - bInterval, [63](#)
 - bLength, [63](#)
 - bmAttributes, [63](#)
 - wMaxPacketSize, [63](#)
- usbHardware
 - usbEndpoint, [62](#)
- usbSetupPacket, [64](#)
 - bmRequestType, [65](#)
 - bRequest, [65](#)
 - descriptor, [65](#)
 - descriptorIndex, [65](#)
 - request, [65](#)
 - usbSetupPacket, [64](#)

- wIndex, [65](#)
- wLength, [66](#)
- wValue, [66](#)
- vendor
 - usbDev::deviceEndpoint, [50](#)
- wIndex
 - usbSetupPacket, [65](#)
- wLength
 - usbSetupPacket, [66](#)
- wMaxPacketSize
 - usbEndpoint::endpointDescriptor, [63](#)
- writeDescriptor
 - usbDev::stringDescriptor, [56](#)
- writeIn
 - samd21usbDevice, [10](#)
 - usbDev, [39](#)
 - usbEndpoint, [60](#)
- writeInStall
 - samd21usbDevice, [10](#)
 - usbDev, [40](#)
 - usbEndpoint, [60](#)
- writeInZLP
 - samd21usbDevice, [10](#)
 - usbDev, [40](#)
 - usbEndpoint, [61](#)
- wTotalLength
 - usbDev::configurationDescriptor, [45](#)
- wValue
 - usbSetupPacket, [66](#)