USB C++ Library

pre-release

Generated by Doxygen 1.9.1

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

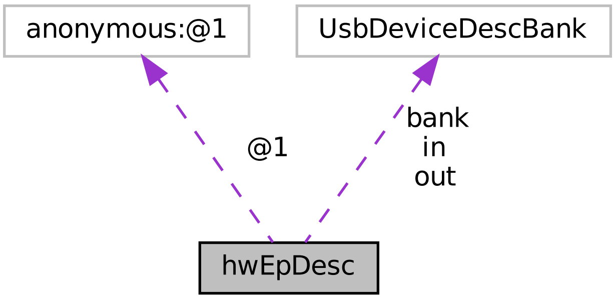
Here are the classes, structs, unions and interfaces with brief descriptions:

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
```

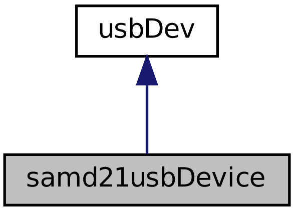
The documentation for this union was generated from the following file:

- src/hardware/samD21/samD21usbDevice.cpp

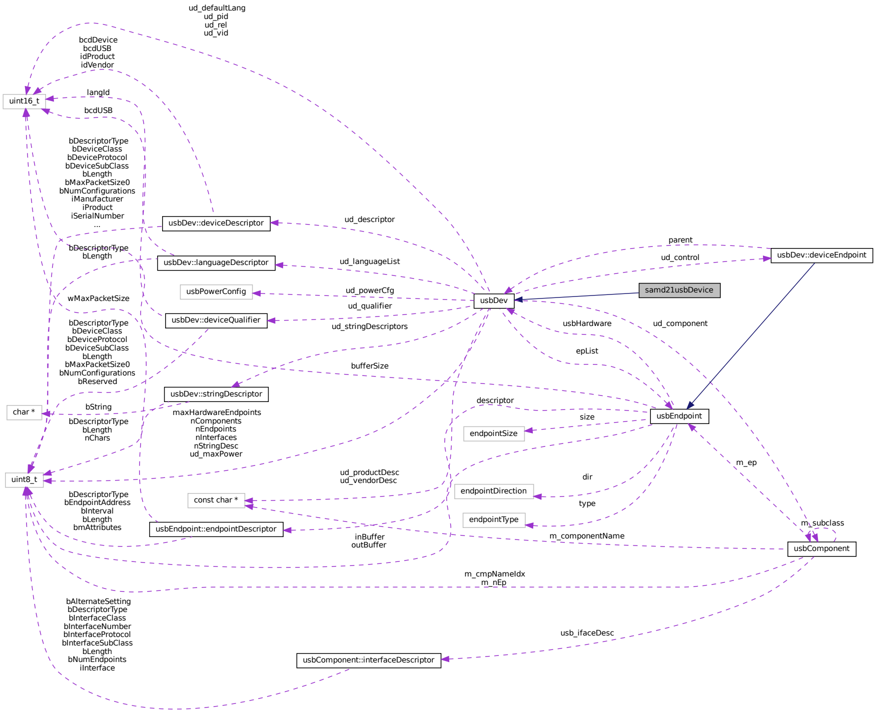
3.2 samd21usbDevice Class Reference

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

Inheritance diagram for samd21usbDevice:



Collaboration diagram for samd21usbDevice:



Public Member Functions

- samd21usbDevice ()
- void attach ()

Protected Member Functions

- void hardwareExec ()
- void initialiseHardware ()
- void sendRequestError (usbEndpoint ∗ep, uint8_t bnk=0)
- void setAddress (uint8_t addr)
- void setMode (usbModeTypes mode)
- uint16_t writeIn (uint8_t ep, uint16_t nBytes)
- void writeInStall (uint8_t ep)
- void writeInZLP (uint8_t ep)

Private Member Functions

- bool addEndpointToHardware (uint8_t idx, usbEndpoint ∗ep)
- void configureClock ()
- void configureEndpoint (uint8_t idx)
- void configurePins ()
- uint8_t epSizeCode (usbEndpoint::endpointSize sz)
- void interruptHandler ()
- void readComplete (uint8_t dEp, uint8_t bnk)

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 (
            usbModeTypes 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]
```

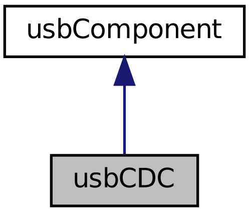
The documentation for this class was generated from the following files:

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

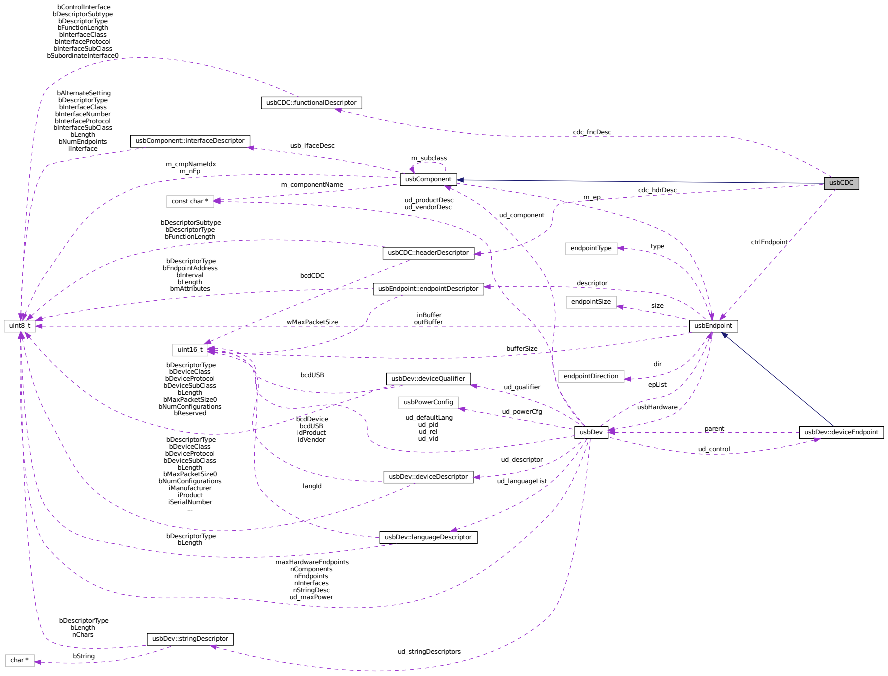
3.3 usbCDC Class Reference

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

Inheritance diagram for usbCDC:



Collaboration diagram for usbCDC:



Classes

- struct capabilityDescriptor
- class cdcEndpoint
- struct functionalDescriptor
- struct headerDescriptor

Public Member Functions

- usbCDC (usbComponent ∗subclass, const char ∗name="CDC")

Protected Member Functions

- void bufferFunctionalDescriptor (uint8_t ∗buffer, uint16_t ∗len)

Protected Attributes

- functionalDescriptor cdc_fncDesc
- headerDescriptor cdc_hdrDesc
- usbEndpoint ∗ ctrlEndpoint

Friends

- class cdcAcmEndpoint

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]
```

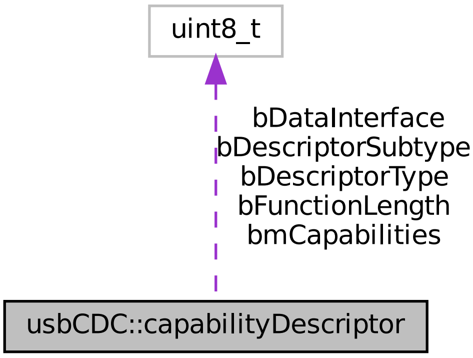
The documentation for this class was generated from the following files:

- 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

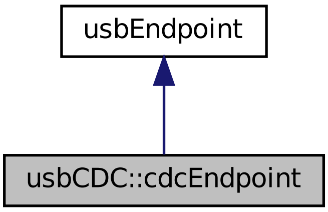
`uint8_t usbCDC::capabilityDescriptor::bmCapabilities`

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

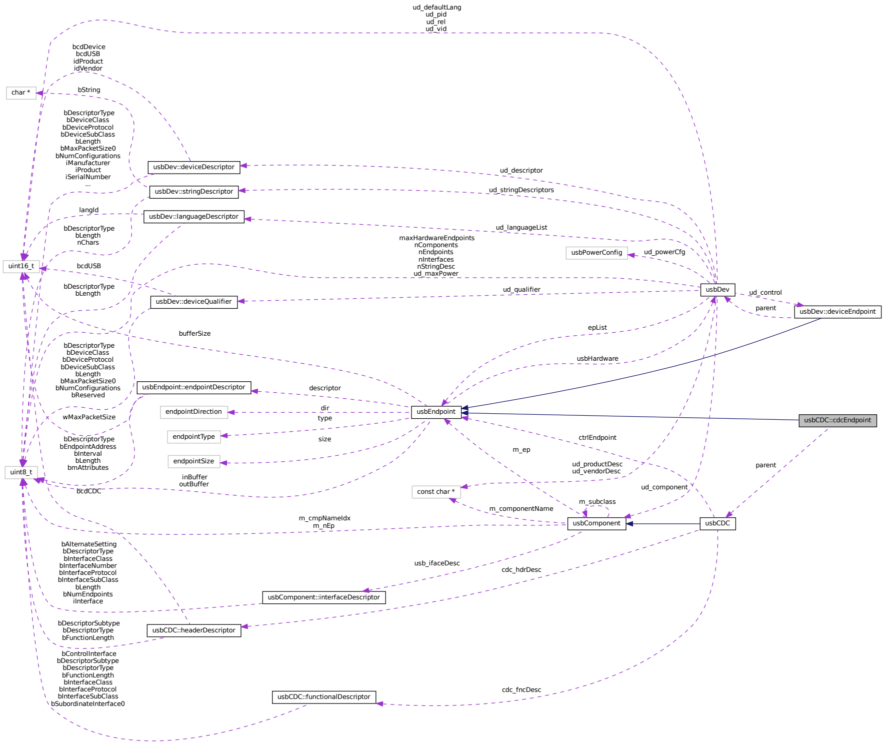
- src/CDC/usbCDC.h

3.5 usbCDC::cdcEndpoint Class Reference

Inheritance diagram for usbCDC::cdcEndpoint:



Collaboration diagram for usbCDC::cdcEndpoint:



Public Member Functions

- cdcEndpoint (usbCDC ∗p, uint16_t bSize, usbEndpoint::endpointSize sz, usbEndpoint::endpointDirection u↩
Dir, usbEndpoint::endpointType uType)
- void dataRecieved (uint16_t nBytes)

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]
```

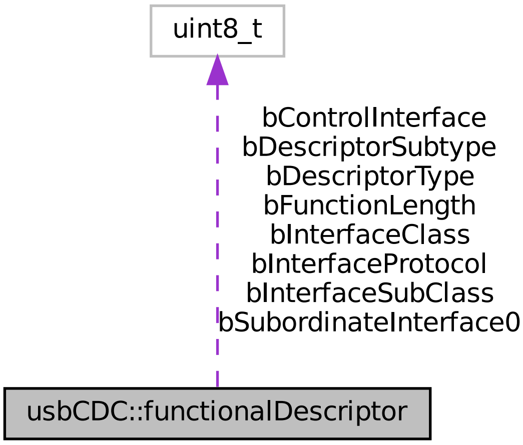
The documentation for this class was generated from the following files:

- 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 bControlInterface
- 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 bControlInterface

```
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
```

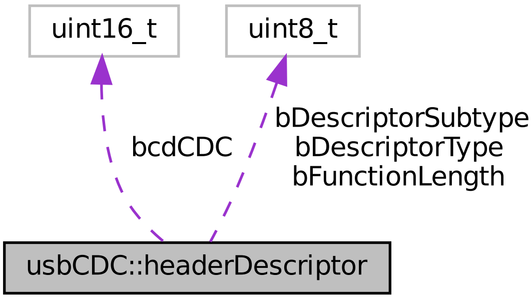
The documentation for this struct was generated from the following file:

- 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
```

3.7.1.3 bDescriptorType

```
uint8_t usbCDC::headerDescriptor::bDescriptorType
```

3.7.1.4 bFunctionLength

```
uint8_t usbCDC::headerDescriptor::bFunctionLength
```

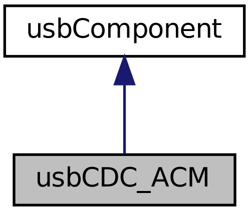
The documentation for this struct was generated from the following file:

- src/CDC/usbCDC.h

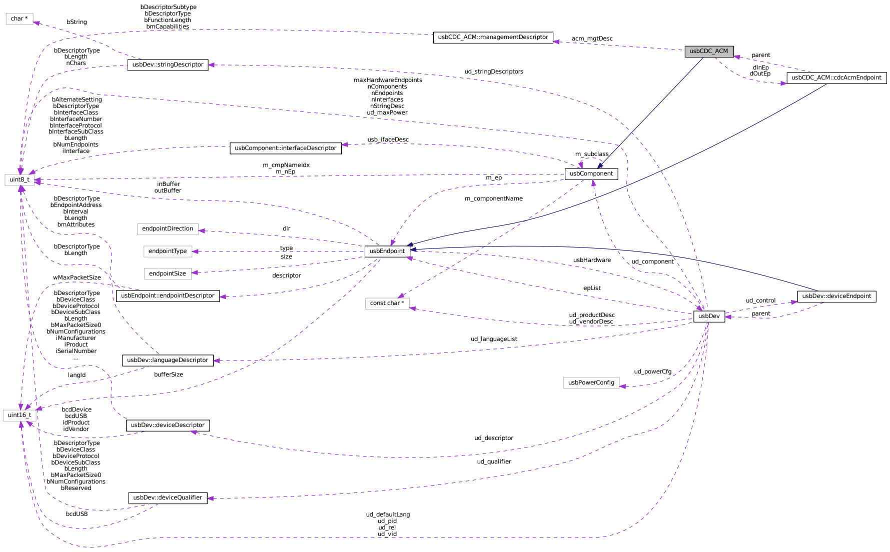
3.8 usbCDC_ACM Class Reference

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

Inheritance diagram for usbCDC_ACM:



Collaboration diagram for usbCDC_ACM:



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]
```

3.8.4.3 dOutEp

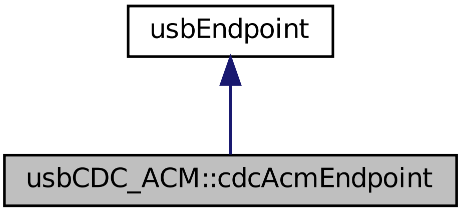
`usbCDC_ACM::cdcAcmEndpoint* usbCDC_ACM::dOutEp [protected]`

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

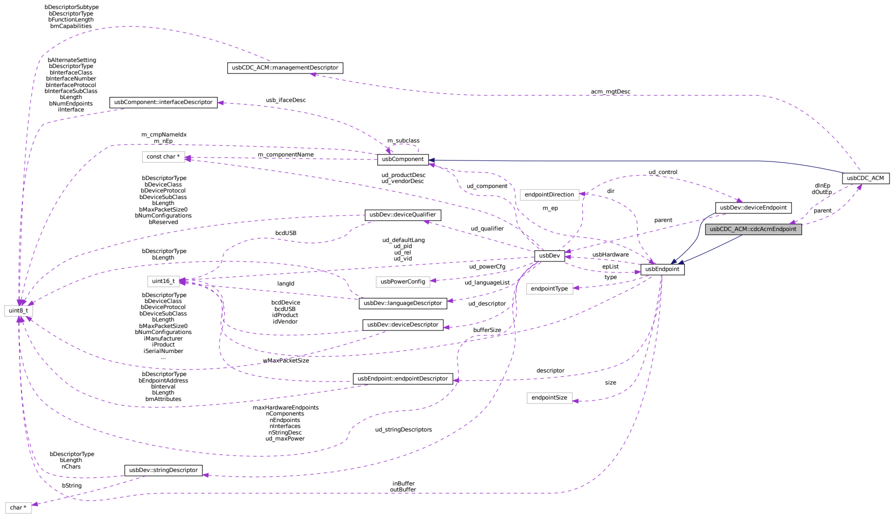
- src/CDC/usbCDC_ACM.h
- src/CDC/usbCDC_ACM.cpp

3.9 usbCDC_ACM::cdcAcmEndpoint Class Reference

Inheritance diagram for usbCDC_ACM::cdcAcmEndpoint:



Collaboration diagram for usbCDC_ACM::cdcAcmEndpoint:



Public Member Functions

- cdcAcmEndpoint (usbCDC_ACM ∗p, uint16_t bSize, usbEndpoint::endpointSize sz, usbEndpoint::endpointDirection uDir, usbEndpoint::endpointType uType)
- void dataRecieved (uint16_t nBytes)

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]
```

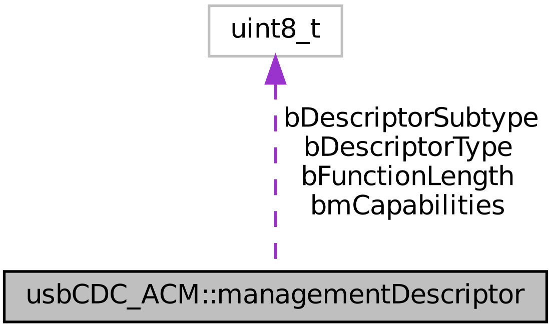
The documentation for this class was generated from the following files:

- 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
```

3.10.1.3 bFunctionLength

`uint8_t usbCDC_ACM::managementDescriptor::bFunctionLength`

3.10.1.4 bmCapabilities

`uint8_t usbCDC_ACM::managementDescriptor::bmCapabilities`

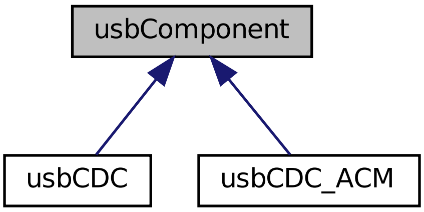
The documentation for this struct was generated from the following file:

- src/CDC/usbCDC_ACM.h

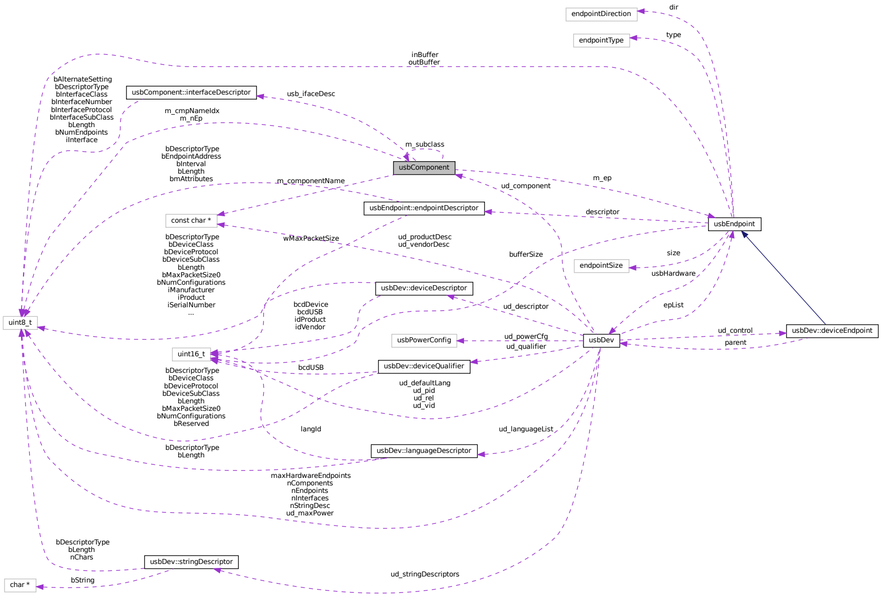
3.11 usbComponent Class Reference

`#include <usbComponent.h>`

Inheritance diagram for usbComponent:



Collaboration diagram for usbComponent:



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

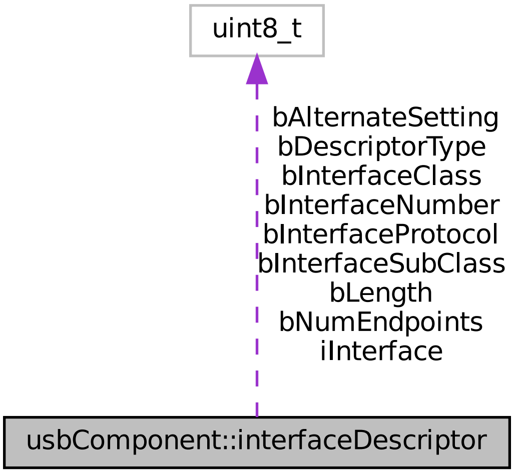
```
interfaceDescriptor usbComponent::usb_ifaceDesc
```

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
```

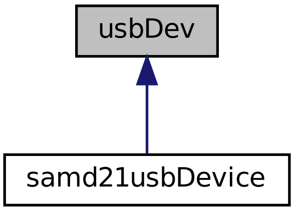
The documentation for this struct was generated from the following file:

- src/usbComponent.h

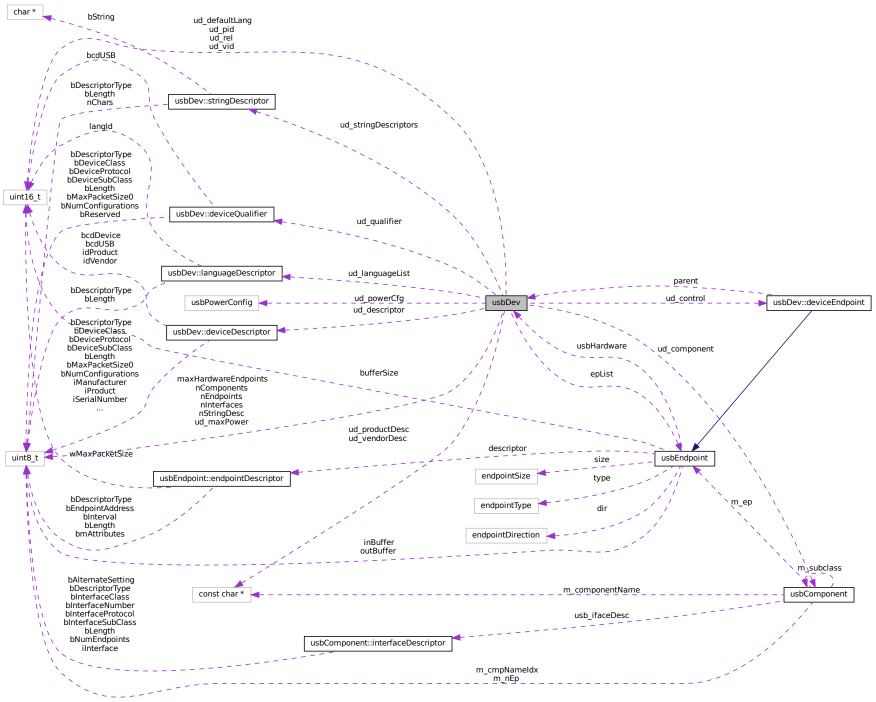
3.13 usbDev Class Reference

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

Inheritance diagram for usbDev:



Collaboration diagram for usbDev:



Classes

- struct configurationDescriptor
- struct deviceDescriptor
- class deviceEndpoint
- struct deviceQualifier
- struct languageDescriptor
- class stringDescriptor

Public Member Functions

- usbDev ()
- bool addComponent (usbComponent ∗component)
- virtual void attach ()=0
- void exec ()
- void initialise ()
- void setConfigurationString (const char ∗desc)
- void setMaxPower (float current)
- void setProductID (uint16_t id)
- void setProductString (const char ∗desc)
- void setProductVersion (uint8_t major, uint8_t minor, uint8_t fix)
- void setSerialNumberString (const char ∗desc)
- void setVendorID (uint16_t id)
- void setVendorString (const char ∗desc)

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

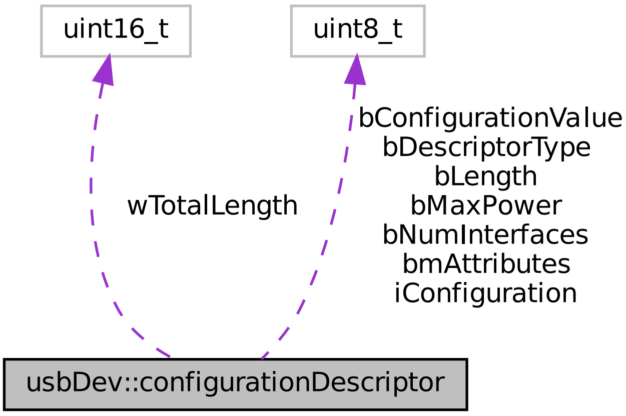
```
uint16_t usbDev::ud_vid [protected]
```

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

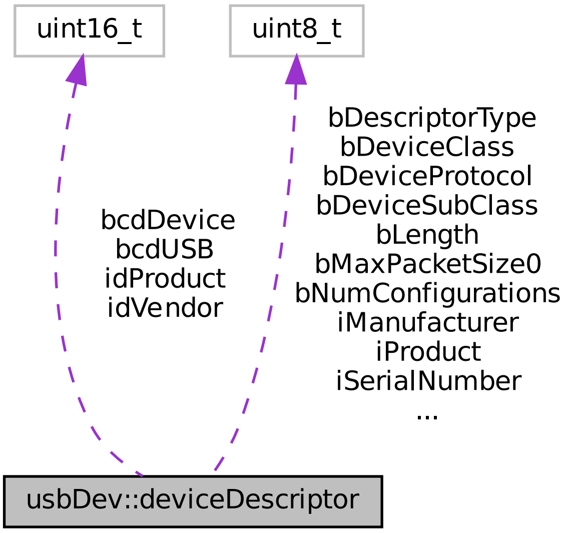
```
uint16_t usbDev::configurationDescriptor::wTotalLength
```

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

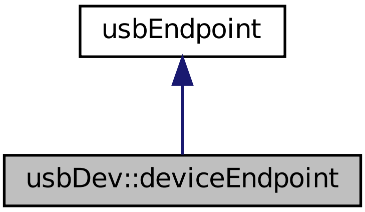
```
uint8_t usbDev::deviceDescriptor::iSerialNumber
```

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

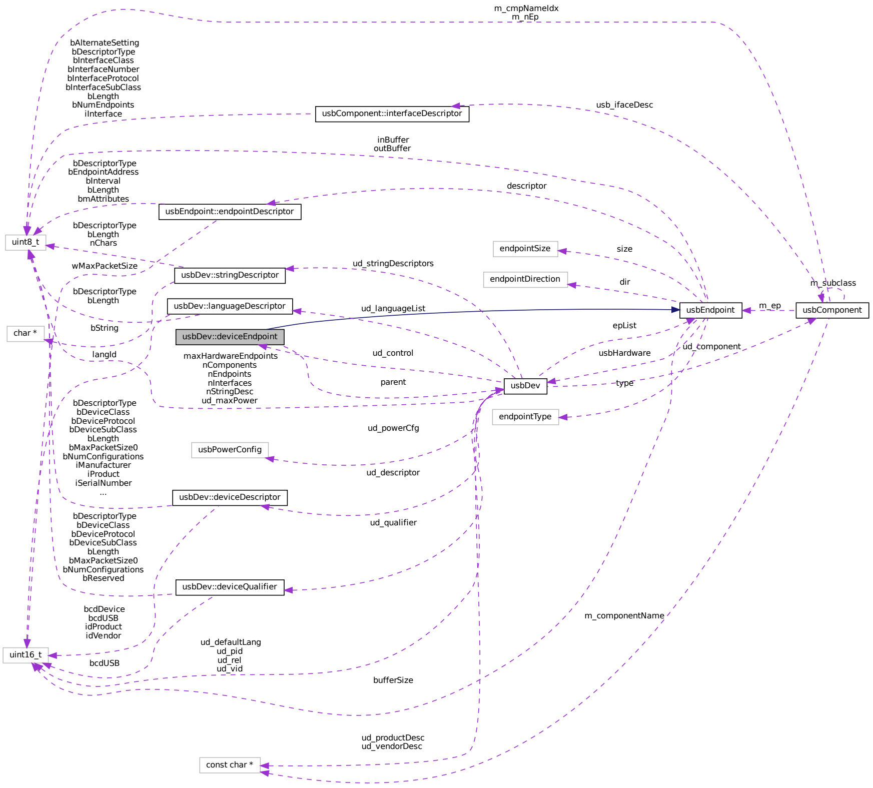
- src/usbDev.h

3.16 usbDev::deviceEndpoint Class Reference

Inheritance diagram for usbDev::deviceEndpoint:



Collaboration diagram for usbDev::deviceEndpoint:



Public Types

- 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 }

Public Member Functions

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

Public Attributes

- usbDev ∗ parent

3.16.1 Member Enumeration Documentation

3.16.1.1 dataTransferDirection

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

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

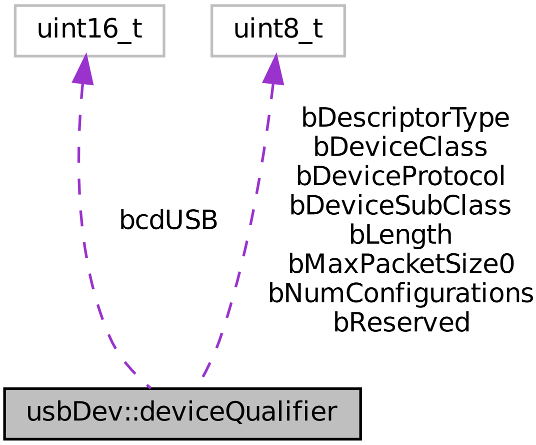
```
usbDev* usbDev::deviceEndpoint::parent
```

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

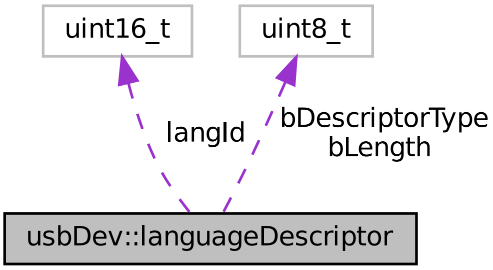
```
uint8_t usbDev::deviceQualifier::bReserved
```

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

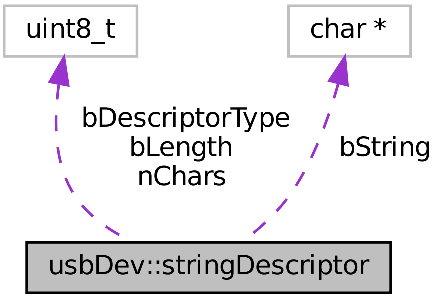
```
uint16_t usbDev::languageDescriptor::langId[1]
```

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]
```

3.19.3.3 bString

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

3.19.3.4 nChars

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

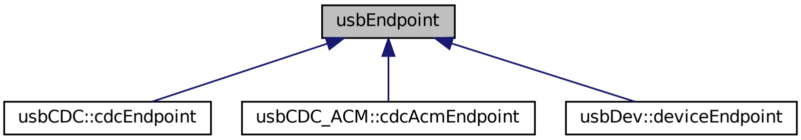
The documentation for this class was generated from the following files:

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

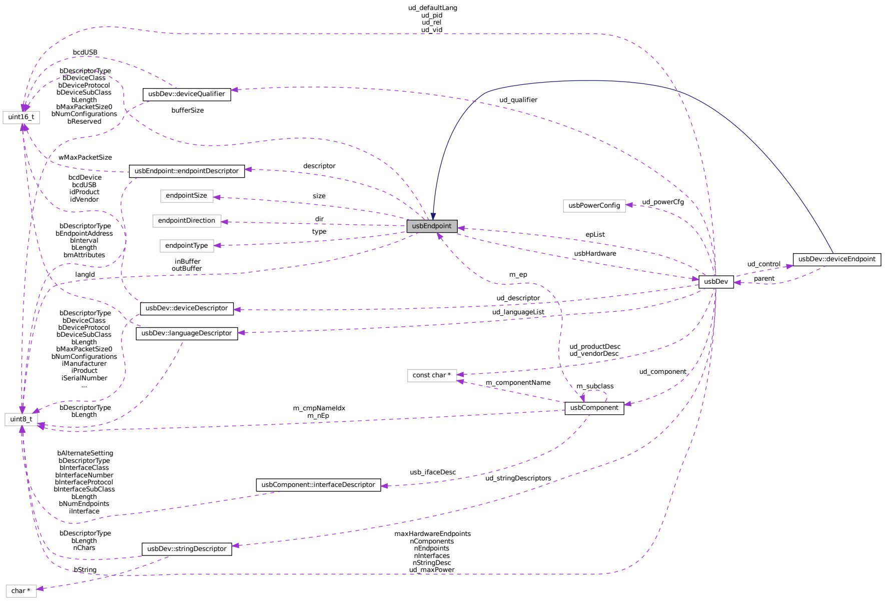
3.20 usbEndpoint Class Reference

```
#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 , isochronos =0x01 , bulk =0x02 , interrupt =0x03 ,
  dual =0x04 }

Public Member Functions

- usbEndpoint (uint16_t bufferSz, 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 bufferSz,
            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
```

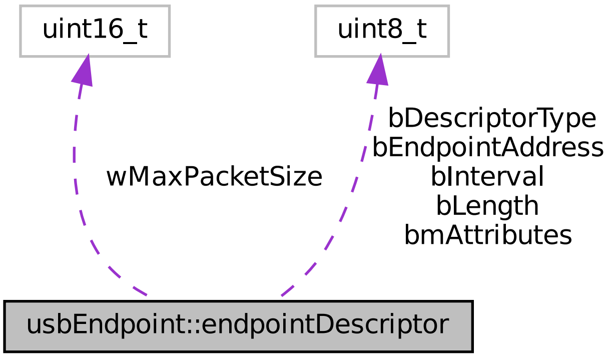
The documentation for this class was generated from the following files:

- 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
```

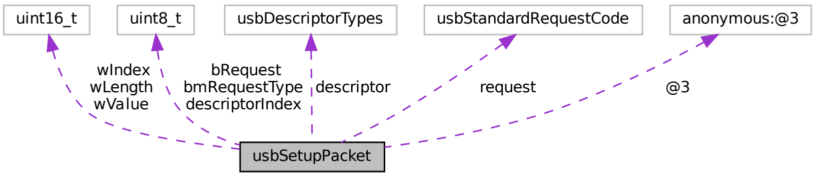
The documentation for this struct was generated from the following file:

- 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