# USB CDC通信装置としてのAVR® DU

## AN5792

www.microchip.com製品頁: AVR16DU14, AVR16DU20, AVR16DU28, AVR16DU32, AVR32DU14, AVR32DU20, AVR32DU28, AVR32DU32, AVR64DU28, AVR64DU32

## 序説

著者: Henrik Moe Arnesen and Phillip Olk, Microchip Technology Inc.

通信装置ｸﾗｽ(CDC:Communication Device Class)は万能直列ﾊﾞｽ(USB:Universal Serial Bus)上の全ての通信形式を許す汎用の方法です。このｸﾗｽはﾃﾞｼﾞﾀﾙ電話やｱﾅﾛｸﾞ変復調器(ﾓﾃﾞﾑ)のような電気通信装置とADSLやｹｰﾌﾞﾙ変復調器(ﾓﾃﾞﾑ)のような網装置の接続を可能にします。

CDC装置はかなり複雑な装置の実装を許し、USB上の通信の容易な方法として使うこともできます。この例はﾎｽﾄ側の応用ﾌﾟﾛｸﾞﾗﾑを簡単化する仮想OCMﾎﾟｰﾄとして現れ得るCDC装置です。

この文書はUSB CDCの基本を説明し、MPLAB®ｺｰﾄﾞ構成部(MCC)で提供されるUSB装置階層とでAVR® DUを使う利点とUSB通信を許すために外部変換器をどう置き換えできるかを記述します。

この文書はUSB CDC通信を利用する方法を示す2つの例を記述します。最初はCDCの基本とﾎｽﾄとどう情報交換するかを示す簡単な応用です。他の例は外部直列通信をUSBに変換することを必要とするどの応用でも基礎にすることができるUSB⇔USART橋渡し応用を示します。両例は手動端末情報交換とﾎｽﾄ側応用にそれを統合する方法を示す自動化したPythonｽｸﾘﾌﾟﾄを持ちます。外部ｺｰﾄﾞへのﾘﾝｸとそれを構成設定するための段階的な指示は「簡単なCDC通信」と「USART橋渡しのためのCDC」の項で提供されます。

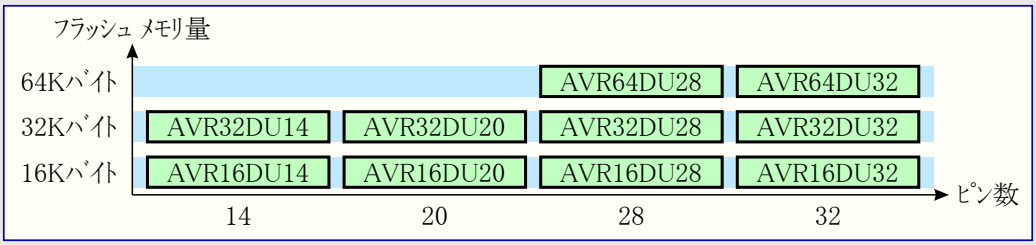
目次

# 1. 関連ﾃﾞﾊﾞｲｽ

本章はこの文書に関連するﾃﾞﾊﾞｲｽを一覧にします。下図はﾋﾟﾝ数の変種とﾒﾓﾘ量を展開して各種系統ﾃﾞﾊﾞｲｽを示します。

- これらのﾃﾞﾊﾞｲｽがﾋﾟﾝ互換で同じまたはより多くの機能を提供するため、垂直方向移植はｺｰﾄﾞ変更なしで可能です。
- 左への水平方向移植はﾋﾟﾝ数と利用可能な機能を減らします。
- 異なるﾌﾗｯｼｭ ﾒﾓﾘ量を持つﾃﾞﾊﾞｲｽは一般的に異なるSRAMとEEPROMも持ちます。

図1. AVR® DU系概要



# 2. USB CDC基礎

## 2.1. CDC概要

本項はCDC USBｸﾗｽの概要を与えます。

USB通信装置ｸﾗｽ(CDC)は変復調器(ﾓﾃﾞﾑ)、電話機のような様々な種類の通信装置と直列ﾃﾞｰﾀの流れを使う網(ﾈｯﾄﾜｰｸ)装置と通信するためにｺﾝﾋﾟｭｰﾀ ｼｽﾃﾑ用に標準化した方法を提供するUSBｸﾗｽです。USBｲﾝﾌﾟﾘﾒﾝﾀｰｽﾞ ﾌｫｰﾗﾑ(USB-IF)がCDCｸﾗｽを定義します。CDCｸﾗｽはｺﾝﾋﾟｭｰﾀへ接続して標準ｼﾘｱﾙ ﾎﾟｰﾄの様に動くことを装置に許すため有用で、独占的なﾄﾞﾗｲﾊﾞの必要なしにUSB上のﾃﾞｰﾀ送受信を容易にし、装置がﾎｽﾄとどう通信するかを定義する装置記述子と標準規約を通して達成します。

ｺﾝﾋﾟｭｰﾀとの通信はUSART⇔USB通信のような1つ以上の直列(ｼﾘｱﾙ)接続を変換することができる独立した装置を通して度々行われます。組み込みUSB支援を持つAVR DU系は追加の変換の必要をなくすことができ、より少ない部品とより多くの構成可能な設定で設計手順を簡単にします。この目的のため、AVRﾃﾞﾊﾞｲｽは通信装置ｸﾗｽ(CDC)ｼﾘｱﾙ ﾄﾞﾗｲﾊﾞを使うことができます。このCDCﾄﾞﾗｲﾊﾞは'UARTのような'ｲﾝﾀｰﾌｪｰｽを提供し、列挙、装置発見、ﾃﾞｰﾀ転送枠組みのような低位USB詳細を管理する必要なしに、ｼﾘｱﾙ通信にUSBを使うことを応用設計者に許します。

USBが装置とｺﾝﾋﾟｭｰﾀ間の標準的な通信接続の1つになる一方で、他のいくつかの通信規約が未だ使われています。RS-485規格はいくつかの工業設定での頼れる通信方法の1つで、より古いRS-232とRS422はより古い構成で未だ普及しています。これらの構成へのｺﾝﾋﾟｭｰﾀ追加はもはや一般的でない直列ﾎﾟｰﾄ ﾊｰﾄﾞｳｪｱ ｲﾝﾀｰﾌｪｰｽを通して起きました。けれども、それら全ての規格は直列通信を使い、従って、AVR DUﾏｲｸﾛ ｺﾝﾄﾛｰﾗとUSART周辺機能を使ってｲﾝﾀｰﾌｪｰｽすることができます。

## 2.2. USB CDCとでのAVR DUに対する使用事例

本項はAVR DUのUSBを使うように移行する背後の動機を検討します。

簡単な通信から高度な応用へ、CDCｸﾗｽの簡単さは多くの多様な実装に対して最初の選択解決策にします。下は最も簡単な形式でCDCｸﾗｽを使ういくつかの例です。

### 仮想ｼﾘｱﾙ ﾎﾟｰﾄ

端末作業を通す通信はUSB CDCを通して通信する最も簡単な方法の1つです。仮想ｼﾘｱﾙ ﾎﾟｰﾄとして発見されたAVR DU構成はｼﾘｱﾙ作業を開始してﾏｲｸﾛ ｺﾝﾄﾛｰﾗとのﾃﾞｰﾀ転送を許します。ｼﾘｱﾙ作業は端末ﾌﾟﾛｸﾞﾗﾑまたはｽｸﾘﾌﾟﾄや応用を通して開始することができ、これはﾏｲｸﾛ ｺﾝﾄﾛｰﾗと通信する高度なﾎｽﾄ側応用を持つことを可能にします。

### ﾃﾞｰﾀ記録

装置からのﾃﾞｰﾀと事象の記録は一般的に外部通信に使われます。仮想ｼﾘｱﾙ ﾎﾟｰﾄに対して上で記述した方法はこれを達成することができます。

### 橋渡し応用

他の通信周辺機能への応用橋渡しは接続したｺﾝﾋﾟｭｰﾀの接続性を広げます。橋渡しすることができるいくつかの周辺機能は次のとおりです。

- USART
- $I^2C$
- SPI

内部周辺機能に加えて、USB上で通信するために基板上の周辺機能に接続した外部送受信部を使って可能性を広げます。

- RS-485, RS-422, RS-232
- CAN

### ﾌﾞｰﾄﾛｰﾀﾞ

CDCｸﾗｽが簡素な通信一括を利用しているため、ﾎｽﾄ側ﾄﾞﾗｲﾊﾞの必要なしにﾌﾞｰﾄﾛｰﾀﾞ内で動くことにも適用可能です。それは全てのﾊﾞｲﾄを送信する簡単なｽｸﾘﾌﾟﾄが必要とされる全てのものです。

## 2.3. 直列通信にCDCを使う利点

かなり何処にもあるUSBｸﾗｽの1つのため、CDCは全てのｵﾍﾟﾚｰﾃｨﾝｸﾞ ｼｽﾃﾑで広く支援され、装置とﾎｽﾄ ｺﾝﾋﾟｭｰﾀまたは他の組み込みｼｽﾃﾑ間の全ての通信に対して良い選択になります。

### ﾎｽﾄ側CDCﾄﾞﾗｲﾊﾞ

CDCｸﾗｽは(Windows 98からの)Windows、macOS、殆どのLinux配布で実装されています。Windows 10からはｵﾍﾟﾚｰﾃｨﾝｸﾞ ｼｽﾃﾑが今やCDCに対して使われる標準ﾄﾞﾗｲﾊﾞを持つため、始動手順で追加の情報が与えられる必要がありません。Windowsのより古い版(Windows 7とそれ以前)を支援するため、正しいﾄﾞﾗｲﾊﾞと関連付けする.infﾌｧｲﾙを提供します。

新しいUSB装置接続時、ｺﾝﾋﾟｭｰﾀは装置ｸﾗｽIDを調べて正しいﾄﾞﾗｲﾊﾞを用意します。命名のような独自情報や独自ﾄﾞﾗｲﾊﾞが必要とされる場合、より多くの構成設定が必要とされます。この応用記述は簡単なCDC通信でだけ扱うので、標準ﾄﾞﾗｲﾊﾞで充分です。

LinuxとmacOSでは装置が標準に適合する識別子を提供する限り、全てのUSB装置がｶｰﾈﾙによって認証されます。ｶｰﾈﾙはﾄﾞﾊﾞｲﾊﾞに対する調査なしにこれを行います。CDC用のﾄﾞﾗｲﾊﾞも長い間OSに含まれています。装置は変復調器(ﾓﾃﾞﾑ)名が後続する/dev/*での変復調器として現れます。この実体での*は通信の方向に応じてtty(ﾃﾚﾀｲﾌﾟ)かcu(電話)のどちらかです。

# 3. USB規約の理解

本章はUSBとCDCｸﾗｽでのいくつかのもっと詳しい情報を網羅します。

## 3.1. USB基本構造の簡単な概要

万能直列ﾊﾞｽ(USB)は現代のｺﾝﾋﾟｭｰﾀ環境のなくてはならない部分になり、それぞれの装置を接続するための何処にでもあるｲﾝﾀｰﾌｪｰｽとして扱います。本項はUSB基本構造とUSB装置階層の上位概要を記述します。

### USB技術と構成要素

高速(High-Speed)USBとしても知られるUSB 2.0は各々が潜在的にﾊﾌﾞと周辺機能装置の複雑な樹状形態に至る、多数のUSBﾎﾟｰﾄを持つ単一ﾎｽﾄ制御器ｲﾝﾀｰﾌｪｰｽである階層化した星形接続形態を提供します。この接続形態の中心はﾎｽﾄ制御器に統合されたﾙｰﾄﾊﾌﾞで、これは追加のﾊﾌﾞまたは直接的に装置へ分岐し、単一ﾎｽﾄへ接続される最大127個の装置を許します。USB 2.0の鍵となる構成要素はﾎｽﾄ制御器、中継器とﾃﾞｰﾀ通行管理部として扱うﾊﾌﾞ、装置それら自身、例えば、記憶ﾄﾞﾗｲﾌﾞや入力周辺機能を含みます。この接続形態の各接続は雑音を減らして信号の完全性を強化するために差動信号を使うﾃﾞｰﾀ転送用の2本と、ﾊﾞｽ給電装置を支援するために電力を配給するための2本で、4線を使います。USB 2.0仕様は複雑化した誤り処理機構と電力管理も導入し、USB接続した装置の多様な全体環境に渡って強化した性能とｴﾈﾙｷﾞｰ効率を保証します。

### USBﾃﾞｰﾀの流れ

ﾃﾞｰﾀの流れはﾎｽﾄ中心型通信様式を通して管理され、これはﾎｽﾄ制御器が全てのﾃﾞｰﾀ転送を開始して装置間の転送を調整します。USB 2.0は4つのﾃﾞｰﾀ転送型、指令と状態操作用の制御転送、ﾌｧｲﾙ転送のように時間に敏感でないﾃﾞｰﾀ本体の大量用の大量(ﾊﾞﾙｸ)転送、音響や映像の流れのように時間に敏感なﾃﾞｰﾀ用の等時(ｱｲｿｸﾛﾅｽ)転送、ｷｰﾎﾞｰﾄﾞやﾏｳｽの入力のように断続的なﾃﾞｰﾀの少量用の割り込み転送を支援します。各転送型はﾃﾞｰﾀの性質に適した信頼性、速度、帯域割り当てを最適化するように設計されています。通信はﾊﾟｹｯﾄで行い、ﾎｽﾄが装置の様々なｴﾝﾄﾞﾎﾟｲﾝﾄに対するｻｰﾋﾞｽを計画し、ﾃﾞｰﾀ ﾊﾟｹｯﾄが正しく配給されて異常の場合に再試行されるのを保証するために指示票(ﾄｰｸﾝ)に基づく規約を使います。USB 2.0のﾃﾞｰﾀの流れは装置からﾎｽﾄへの通信用の'IN'またはﾎｽﾄから装置への通信用の'OUT'のどちらかとして指定されたｴﾝﾄﾞﾎﾟｲﾝﾄで、双方向ﾃﾞｰﾀ転送を処理する能力によっても特性付けされ、従って、柔軟で効率的なﾃﾞｰﾀ交換枠組みを許します。

### USB接続手順

接続手順は装置列挙で始まり、これは新しい装置がUSBﾎﾟｰﾄに接続される時に起きます。ﾎｽﾄ制御器は装置を検出して通信のための一意のｱﾄﾞﾚｽを割り当てます。検出に続き、ﾎｽﾄは装置の能力、ｸﾗｽ、提供者特有情報のように重要な情報を含むそれの記述子を装置に問います。この段階はﾎｽﾄに関して適切なﾄﾞﾗｲﾊﾞを用意して正しい通信規約を確率するために重要です。

ﾎｽﾄは(装置と構成設定の記述子のような)記述子を読むことによって接続した装置を一旦成功裏に識別すると、装置によって提供された構成設定の1つを選ぶために構成設定の設定要求を送ることによって装置を構成設定します。この段階は装置の記述子で予め定義された装置の通信ｴﾝﾄﾞﾎﾟｲﾝﾄを有効にします。ｴﾝﾄﾞﾎﾟｲﾝﾄはﾃﾞｰﾀ転送の型(制御、大量、割り込み、等時)とそれらのﾃﾞｰﾀの流れ方向(INまたはOUT)を指定します。ﾎｽﾄは装置とのﾃﾞｰﾀ交換に必要な通信ﾁｬﾈﾙを確立するためにこれらのｴﾝﾄﾞﾎﾟｲﾝﾄを使います。

構成設定後、装置は使う準備が整い、確立した要素に従ってﾎｽﾄと装置間でﾃﾞｰﾀを転送することができます。装置が外された場合、ﾎｽﾄ制御器は切断を検出して装置のｱﾄﾞﾚｽを取り除き、将来の使用のために資源とｱﾄﾞﾚｽを解放します。

### USB装置階層

USB装置階層は組み込み応用開発者にUSB装置のいくつかの型を設計して開発するための枠組みを提供します。標準USB装置ｸﾗｽ仕様を実行する機能ﾄﾞﾗｲﾊﾞを通して標準USB装置の開発を容易にします。この階層は層で構成され、ﾊｰﾄﾞｳｪｱ抽象化層(HAL:Hardware Abstraction Layer)はUSB周辺機能とｴﾝﾄﾞﾎﾟｲﾝﾄで相互通信し、ｺｱ層はHALとｸﾗｽ層間の処理を扱い、ｸﾗｽ層はﾎｽﾄとのｸﾗｽ特有相互通信を扱います。

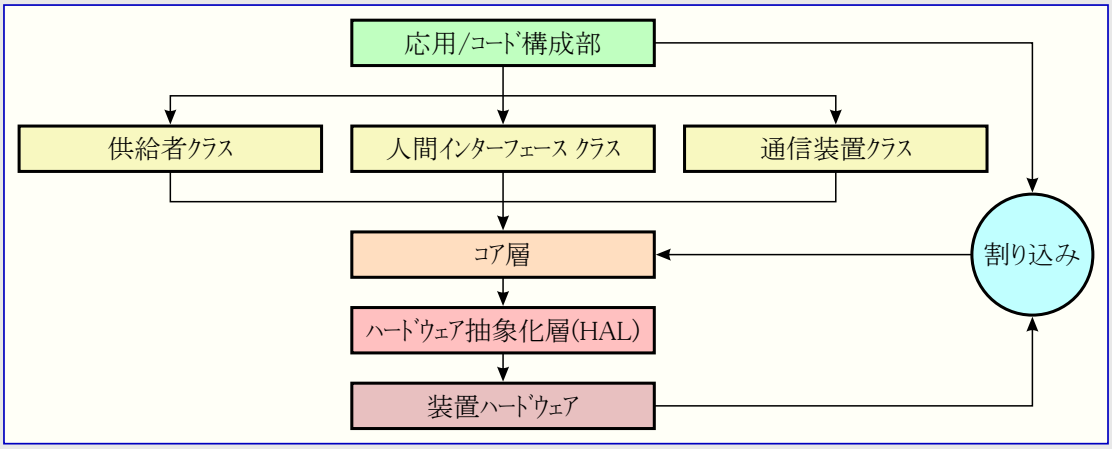
USB装置階層は階層の使い方を強調する例応用を伴い、MPLAB®ｺｰﾄﾞ構成部(MCC)Melody取り付けで利用可能です。これらの例は独自応用を構築するために変更や更新もされます。

USB装置階層は現在以下が特徴です。

- 各種USB装置ｸﾗｽ(CDC、HID、提供者(Vendor))を支援
- 複数構成設定を支援
- 全速(Full-Speed)動作を支援
- 遅延制御転送応答を支援
- 完全な非閉塞(ﾉﾝﾌﾞﾛｯｷﾝｸﾞ)基本構造
- ﾎﾟｰﾘﾝｸﾞと割り込みの両動作を支援
- MCCを通して画像使用者ｲﾝﾀｰﾌｪｰｽ(GUI)構成設定を支援

USB装置階層は次図で図解されるように、規格化されて層化された基本構造が特徴です。

図3-1. USB装置階層基本構造




USB装置階層のより多くの情報は「USB装置階層使用者の手引き」で見つけることができます。

## 3.2. CDCに関連するUSB記述子

USB記述子は何のUSB装置がどう通信するかをﾎｽﾄ ｺﾝﾋﾟｭｰﾀに告げることができます。

### 装置記述子

最上位で、装置記述子は以下を含むけれども、それに限定されないUSB装置の基本情報を含みます。

- USB版
- 装置段階でのｸﾗｽ、副ｸﾗｽ、規約
- 供給者と製品のID (VIDとPID)
- 製造業者、製品、通番
- 装置が持つ異なる構成設定の数

### 構成設定記述子

構成設定記述子は以下を含む構成設定を指定するための設定を含みます。

- 構成設定の大きさ
- ｲﾝﾀｰﾌｪｰｽ数
- 電力属性

USB装置は複数の構成設定を含み、それらを切り替えることができ、例えば、ﾌﾞｰﾄﾛｰﾀﾞ用構成設定と実際の応用のための別の構成設定などです。

### ｲﾝﾀｰﾌｪｰｽ記述子

USBｲﾝﾀｰﾌｪｰｽ記述子は以下を含み、USB構成設定内の特定ｲﾝﾀｰﾌｪｰｽを定義し、それの目的と機能を詳述します。

- 代替設定
- ｴﾝﾄﾞﾎﾟｲﾝﾄ数
- ｲﾝﾀｰﾌｪｰｽのｸﾗｽ、副ｸﾗｽ、規約

### ｴﾝﾄﾞﾎﾟｲﾝﾄ記述子

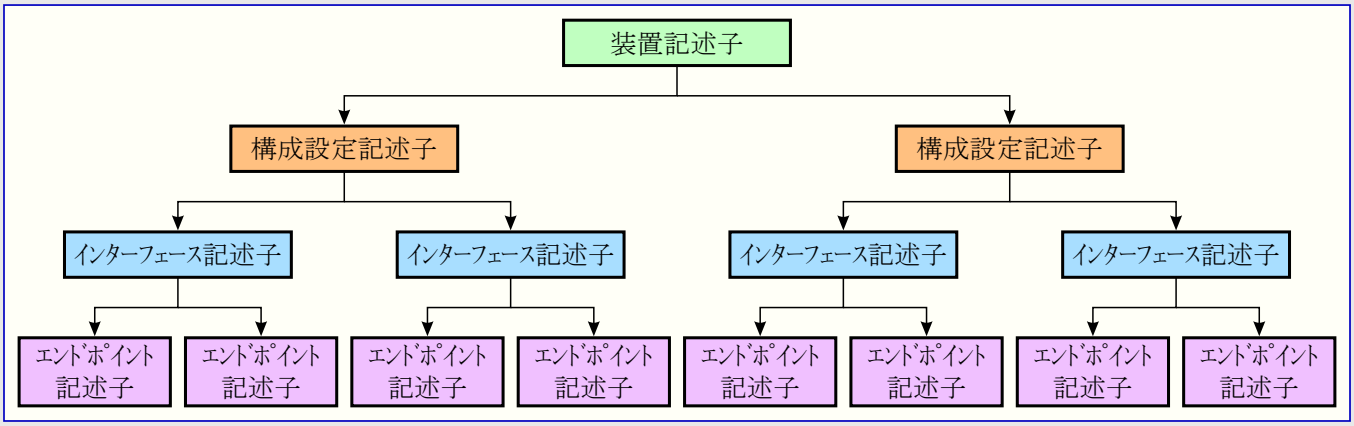
USBｴﾝﾄﾞﾎﾟｲﾝﾄ記述子は以下のような、ｲﾝﾀｰﾌｪｰｽ内の特定のｴﾝﾄﾞﾎﾟｲﾝﾄの属性を指定します。

- ｴﾝﾄﾞﾎﾟｲﾝﾄ ｱﾄﾞﾚｽ
- 方向 (IN/OUT)
- 型 (制御、大量、等時、割り込み)
- ﾊﾟｹｯﾄの大きさ
- ﾎﾟｰﾘﾝｸﾞ間隔

### USB記述子樹形

次図は動いている装置で(上で記述された)各種記述子がUSB記述子樹形でどう可視化され得るかを示します。

図3-2. USB記述子樹形例装置階層基本構造




### 機能的記述子

基本的な装置情報を伝える標準記述子と違い、機能的記述子は特定の装置クラスや機能性に関連する追加の属性を定義することができます。CDCについて、これは通信インターフェースに対して行われ、インターフェースの群化や副クラスで定義された模式のどの部分が装置に実装されるかの情報を含みます。

## 3.3. エンドポイント型とCDCでのそれらの役割

USB通信の文脈に於いて、エンドポイントは装置とホスト間でデータがどう転送されるかを定義する重要な役割を演じます。

### 制御エンドポイント

全てのUSB装置はエンドポイント0として知られる制御エンドポイントを持たなければなりません。制御エンドポイントは装置の管理とCDC装置での準備に使われます。ホストによって関連装置記述子へ送られた要求を処理し、装置構成設定を設定し、装置状態を制御します。例えば、ボーレートを変更するための指令を送ったり、他の通信の要素を構成設定したりすることができます。

### 大量エンドポイント

大量(バルク)エンドポイントは異常検出付きで大量のデータを転送するのに使われ、失敗の場合に再送信しますが、タイミングの保証なしです。CDCではデータの送受信に対して代表的に大量エンドポイントが使われます。これらは実時間転送を必要とせず、確実な配送を必要とする網パケットや直列データのようなデータ パケットの転送に理想的です。

### 割り込みエンドポイント

割り込みエンドポイントは周期的または特定の事象発生時に装置がホストへ送ることが必要な少量のデータを転送します。CDC装置では接続状態の変更や読むためのデータが利用可能なことの合図のような装置状態の変更をホストに通知するかもしれません。

### 等時エンドポイント

等時(アイソクロナス)エンドポイントはこれらが音響や映像の流れのような一定のタイミングを必要とするデータの流れ用に設計されているため、CDC装置では一般的に使われません。これらのエンドポイントは保証されたタイミングを提供しますが、誤り修正を待たず、これは通常CDC装置によって扱われる通信の形式に対して理想的ではありません。

# 4. CDCに関連するUSBクラスと規約

USBのクラス、副クラス、規約はホストに装置の能力を知らせます。

### クラス

USBクラスはUSB装置またはインターフェースによって提供される機能を表す予め定義された部門で、装置とホスト システム間の標準化した通信を容易にします。これらのクラスはドライバに対して再使用とオペレーティング システムに渡る容易な装置認証を許します。いくつかの一般的なクラスは次のとおりです。

- **通信装置クラス(CDC:Communication Device Class)**は変復調器(モデム)、電話機、網(ネットワーク)カードを含む様々な通信と電気通信の形式で通信するためにコンピュータ システムに標準化した方法を提供するUSBクラスです。
- **人間インターフェース装置(HID:Human Interface Device)**はキーボード、マウス、ゲーム コントローラのような装置と他の簡単なインターフェース装置の実装を許すコンピュータ周辺機能に対する規格です。
- **音響装置クラス(ADC:Audio Device Class)**はマイクロフォン、スピーカ、ヘッドセット、音響カードのようにホスト システムと相互通信する音響入出力装置に対する枠組みを提供するクラスです。
- **大容量記憶クラス(MSC:Mass Storage Class)**はフラッシュ ドライブ、外部ハードディスク、メモリ カード読み取り器のような記憶媒体としてホスト システムによって認証されて使われることをUSB装置に許すクラスです。
- **独自または提供者クラス**は標準USBクラスに適合しないUSB装置に使われ、従って独自ドライバを必要とします。これらの装置はそれぞれの目的と機能を扱うことができ、標準USBクラス仕様によって網羅されない専有独自インターフェースと機能を持つ独自の製品の作成を供給者に許します。

いくつかのクラスは装置記述子の装置段階で定義されるのを許す一方で、各種インターフェース記述子は他で定義します。

### 副クラス

USB副(サブ)クラスはUSB装置の動きと機能を更に定義し、装置とホスト間のもっと細かな制御と通信を許します。これらの副クラスは同じ全体的な目的を持つけれども変わった動作形態や応用を持つ装置の違いを許します。この例は次のとおりです。

- 抽象化制御模式に従うCDC装置
- イーサネットを模倣するCDC装置
- MIDIの流れを利用する音響装置
- 映像の流れにだけ関係する映像装置

副クラスは殆どインターフェース記述子に適用されます。

### 規約

USB規約はUSB装置とホスト制御器間の通信を管理する規則と合図規格を関連付け、データが正しく転送されることを保証します。これの例は次のとおりです。

- ITU-T V.250指定規約を持つ抽象化制御模式に適合するCDC装置
- USB EEM規約を使ってイーサネットを模倣するCDC装置
- 規約15を使って映像制御だけに関係する映像装置

### その他

ここで言及されないその他のクラス、副クラス、規約も存在します。より多くの情報についてはUSBでのMicrochip開発者手助け頁をご覧ください。

# 5. CDC例

本章はこの応用記述に付随して作成した2つの例を記述します。

## 5.1. 使ったﾊｰﾄﾞｳｪｱとｿﾌﾄｳｪｱ

### 使ったﾊｰﾄﾞｳｪｱ

- AVR DU
- AVR DU Curiopsity Nano評価ｷｯﾄ

### 使ったｿﾌﾄｳｪｱ

- MPLAB X IDE 6.20またはそれ以降版
- MPLABｺｰﾄﾞ構成部(MCC)ﾌﾟﾗｸﾞｲﾝ 5.1.1またはそれ以降版
- XC8ｺﾝﾊﾟｲﾗ 2.50またはそれ以降版
- MPLABﾃﾞｰﾀ可視器(Data Visualizer) 1.3.1665またはそれ以降版
- AVR Dx DFP 2.6.303またはそれ以降版
- Python 3.11.0またはそれ以降版
- Putty 0.81またはそれ以降版

## 5.2. 簡単なCDC通信

### 5.2.1. 序説

本項はUSB CDC(通信装置ｸﾗｽ)仮想ｼﾘｱﾙ ﾎﾟｰﾄ装置を作成するためにAVR DUﾏｲｸﾛ ｺﾝﾋﾟｭｰﾀを設定して活用する詳細な手引きを提供します。提示するｺｰﾄﾞはMPLABｺｰﾄﾞ構成部(MCC)での準備と構成設定を網羅し、端末とｽｸﾘﾌﾟﾄを通して装置との通信の例を含みます。

- Microchip Discoverでの例ｺｰﾄﾞ
- GitHubでの例ｺｰﾄﾞ

Curiosity Nano評価基盤を利用して、この例は単にCuriosity NanoとUSB Cｹｰﾌﾞﾙとでの評価を許します。

### 5.2.2. 動機

この例の主な目標はAVR DU基盤でUSB CDC通信を実装するのを狙う開発者に包括的な手引きを提供することです。USB CDCはUSB上のｼﾘｱﾙ通信に広く使われる規約で、ﾏｲｸﾛ ｺﾝﾄﾛｰﾗとﾎｽﾄ ｺﾝﾋﾟｭｰﾀ間でのﾃﾞｰﾀ交換を必要とする応用に対して理想的にします。この例に従うことにより、開発者は仮想ｼﾘｱﾙ ﾎﾟｰﾄを素早く設定することができ、USB通信の複雑さよりもむしろそれら応用の核となる機能に集中することを許します。

### 使用事例

- ﾃﾞﾊﾞｯｸﾞと診断:
  - **実時間監視**: ﾏｲｸﾛ ｺﾝﾄﾛｰﾗに接続された感知器や他の周辺機能からの実時間ﾃﾞｰﾀを監視するのに仮想ｼﾘｱﾙ ﾎﾟｰﾄを使うことは特に開発中のﾃﾞﾊﾞｯｸﾞと診断に有用です。
  - **異常記録**: 異常ﾒｯｾｰｼﾞとｼｽﾃﾑ状態を記録するのに仮想ｼﾘｱﾙ ﾎﾟｰﾄを利用することで問題を識別して解決するためにﾎｽﾄ ｺﾝﾋﾟｭｰﾀで分析することができます。
- ﾌｧｰﾑｳｪｱ更新:
  - **ﾌﾞｰﾄﾛｰﾀﾞ通信**: ﾌﾞｰﾄﾛｰﾀﾞは新しいﾌｧｰﾑｳｪｱの写しﾃﾞｰﾀを受け取るのに仮想ｼﾘｱﾙ ﾎﾟｰﾄを使うことができ、信頼に足る素直な更新処理を保証します。
- ﾃﾞｰﾀ採取と制御:
  - **感知器ﾃﾞｰﾀ収集**: ﾏｲｸﾛ ｺﾝﾄﾛｰﾗは様々な感知器からのﾃﾞｰﾀを収集して更なる分析と可視化ためにそれをﾎｽﾄ ｺﾝﾋﾟｭｰﾀに送ることができ、これは環境的な監視、工業的な自動化、科学的な研究のような応用で有用です。
  - **遠隔制御**: ﾎｽﾄ ｺﾝﾋﾟｭｰﾀは作動装置、電動機や他の周辺機能を制御するのに仮想ｼﾘｱﾙ ﾎﾟｰﾄ経由でﾏｲｸﾛ ｺﾝﾄﾛｰﾗに命令を送ることができ、家庭内自動化やﾛﾎﾞｯﾄ工学のような遠隔制御応用を許します。
  - **IoT装置通信**: IoT装置やもっと強力なﾎｽﾄを使う単位部と情報交換することは局所での複雑なﾃﾞｰﾀと算法の処理や制御を許します。
- 人-機械ｲﾝﾀｰﾌｪｽ:
  - **使用者入力**: 仮想ｼﾘｱﾙ ﾎﾟｰﾄはﾎｽﾄ ｺﾝﾋﾟｭｰﾀから使用者入力を受け取ることができ、命令や構成設定変更に応答することをﾏｲｸﾛ ｺﾝﾄﾛｰﾗに許し、これは独自の使用者ｲﾝﾀｰﾌｪｰｽと制御盤のような応用で有用です。
  - **状態帰還**: ﾏｲｸﾛ ｺﾝﾄﾛｰﾗは状態変更と帰還をﾎｽﾄ ｺﾝﾋﾟｭｰﾀに送ることができ、ｼｽﾃﾑ状態についての実時間情報を提供します。

- **教育と試作**:
  - **学習道具**: この例はUSB通信とﾏｲｸﾛ ｺﾝﾄﾛｰﾗのﾌﾟﾛｸﾞﾗﾐﾝｸﾞについて学ぶ学生と趣味人に対する教育的な道具です。仮想ｼﾘｱﾙ ﾎﾟｰﾄを設定して使う実地体験を提供します。
  - **迅速試作**: 開発者はﾏｲｸﾛ ｺﾝﾄﾛｰﾗとﾎｽﾄ ｺﾝﾋﾟｭｰﾀ間の通信に仮想ｼﾘｱﾙ ﾎﾟｰﾄを活用することによって新しい発想を素早く試作して試験することができ、開発処理を加速して商品化までの時間を減らします。

### 5.2.3. 操作

含まれたREADME文書はこの例を設定して使うための包括的な手引きを含みます。

この段階的な手引きは端末応用を使う例を実演します。

1. READMEで記述されるようにﾊｰﾄﾞｳｪｱを接続してください。
2. 例の序説項でﾘﾝｸされたﾌﾟﾛｼﾞｪｸﾄをﾀﾞｳﾝﾛｰﾄﾞして開くか、または準備の手引きを使ってMPLAB Xでそれを作成してください。
3. ﾃﾞﾊﾞｲｽに書いてください。
4. MPLABﾃﾞｰﾀ可視器(Data Visualizer)を開いて端末通信作業を開始してください。
5. 端末ｳｨﾝﾄﾞｳに何かを入力してｷｰが押された時にそれが送り返されるのを監視してください。

含まれたPython®ｽｸﾘﾌﾟﾄは手動で端末作業を初期化することなく、仮想ｼﾘｱﾙ ﾎﾟｰﾄ例を他の応用に統合することを実演します。上の3.までの段階に従い、その後に以下を続けてください。

1. 例のﾌｫﾙﾀﾞにPythonｽｸﾘﾌﾟﾄを配置してください。
2. 設定に何か変更が行われた場合はｽｸﾘﾌﾟﾄを更新してください。
3. ｽｸﾘﾌﾟﾄを走らせて送信した16進値がASCII文字として送り返されるのを観察してください。

この例はCDCｸﾗｽを使ってAVR DUに対してﾃﾞｰﾀをどう送るかの基本的な実演として機能します。小さな変更で、この例は命令の受信や感知器ﾃﾞｰﾀの送信に適応することができ、様々な応用に対する多用途な開始点にします。

### 5.2.4. 結果

この例に従うことにより、開発者は以下を達成するでしょう。

1. **USB CDC通信の成功裏の設定**: AVR DUはCDC装置として識別するように構成設定され、仮想ｼﾘｱﾙ ﾎﾟｰﾄ上でﾎｽﾄ ｺﾝﾋﾟｭｰﾀと通信することを許します。
2. **ﾃﾞｰﾀの送受信**: 例応用はCDCｲﾝﾀｰﾌｪｰｽ経由で受信したどのﾃﾞｰﾀも送り返し、基本的なﾃﾞｰﾀ送受信を実演します。
3. **MPLABﾃﾞｰﾀ可視器との統合**: この例は仮想ｼﾘｱﾙ ﾎﾟｰﾄの監視と相互通信のためにMPLABﾃﾞｰﾀ可視器(Data Visualozer)をどう使うかを示します。
4. **自動化用Pythonｽｸﾘﾌﾟﾄ**: Pythonｽｸﾘﾌﾟﾄは仮想ｼﾘｱﾙ ﾎﾟｰﾄ上でのﾃﾞｰﾀ送受信の自動化を提供し、AVR DUを他のｿﾌﾄｳｪｱ ﾂｰﾙとどう統合するかを示します。

この例の終了で、開発者はﾏｲｸﾛ ｺﾝﾄﾛｰﾗとﾎｽﾄ ｺﾝﾋﾟｭｰﾀ間で信頼できるﾃﾞｰﾀ交換を必要とするより複雑な応用の作成を許すAVR DUでのUSB CDC通信実装をしっかりと理解するでしょう。

## 5.3. USB橋渡しのためのCDC

### 5.3.1. 序説

この文書はUSB CDC(通信装置ｸﾗｽ)でUSART橋渡し装置を作成するのにAVR DUﾏｲｸﾛ ｺﾝﾄﾛｰﾗを設定して活用するための詳細な手引きを提供します。付随するｺｰﾄﾞはMPLABｺｰﾄﾞ構成部(MCC)での準備と構成設定を網羅し、端末とｽｸﾘﾌﾟﾄを通す装置との通信の例を含みます。

- Microchip Discoverでの例ｺｰﾄﾞ
- GitHubでの例ｺｰﾄﾞ

Curiosity Nano評価基盤を利用して、この例は単にCuriosity Nanoと2本のUSB CｹｰﾌﾞﾙとでAVR DUの評価を許します。

注: AVR DUのUSARTﾋﾟﾝをｱｸｾｽするようにCuriosity Nanoの基板上ﾃﾞﾊﾞｯｶﾞを使うことにより、CDC-USART-CDC橋渡しを効率的に作成します。基板上ﾃﾞﾊﾞｯｶﾞは概ね500kbpsまでの通信を支援します。2つのAVR DU制御器を接続することでこの限度を増やすことができます。

### 5.3.2. 動機

この例の主な目標はAVR DU基盤で他の通信周辺機能に橋渡しするためにUSB CDC通信の実装を狙う開発者に包括的な手引きを提供することです。USB CDCはUSB上のｼﾘｱﾙ通信に広く使われる規約で、外部ﾊｰﾄﾞｳｪｱとﾎｽﾄ ｺﾝﾋﾟｭｰﾀ間で直列ﾃﾞｰﾀの交換を必要とする応用に対して理想的にします。この例に従うことにより、開発者はUSARTへの橋渡し応用を素早く準備することができ、USB通信の複雑さよりもむしろそれら応用の核となる機能に集中することを許します。

### 使用事例

- **ﾃﾞﾊﾞｯｸﾞと診断:**
  - **実時間監視:** ﾏｲｸﾛ ｺﾝﾄﾛｰﾗに接続された感知器や他の周辺機能からの実時間ﾃﾞｰﾀを制御部で処理することなく監視するのに橋渡し応用を使うことは特に開発中のﾃﾞﾊﾞｯｸﾞと診断に有用です。
  - **異常記録:** 外部ﾊｰﾄﾞｳｪｱからの異常ﾒｯｾｰｼﾞとｼｽﾃﾑ状態を記録するのに仮想ｼﾘｱﾙ ﾎﾟｰﾄを利用することで問題を識別して解決するのにﾎｽﾄ ｺﾝﾋﾟｭｰﾀで分析することができます。
- **ﾌｧｰﾑｳｪｱ更新:**
  - **ﾌﾞｰﾄﾛｰﾀﾞ通信:** 多くのﾌﾞｰﾄﾛｰﾀﾞ応用は新しいﾌｧｰﾑｳｪｱ ﾃﾞｰﾀを受け取るのにUSART通信を使います。AVR DUの使用はﾌﾟﾛｸﾞﾗﾐﾝｸﾞのために非USB装置をﾎｽﾄ ｺﾝﾋﾟｭｰﾀに接続することを許します。
- **ﾃﾞｰﾀ採取と制御:**
  - **感知器ﾃﾞｰﾀ収集:** ﾏｲｸﾛ ｺﾝﾄﾛｰﾗは様々な感知器からのﾃﾞｰﾀを収集して更なる分析と可視化ためにそれをﾎｽﾄ ｺﾝﾋﾟｭｰﾀに送ることができ、これは環境的な監視、工業的な自動化、科学的な研究のような応用で有用です。
  - **実時間制御ｼｽﾃﾑ:** ﾎｽﾄ ｺﾝﾋﾟｭｰﾀは作動装置、電動機や他の周辺機能を制御するのに仮想ｼﾘｱﾙ ﾎﾟｰﾄ経由でﾏｲｸﾛ ｺﾝﾄﾛｰﾗに命令を送ることができ、家庭内自動化やﾛﾎﾞｯﾄ工学のような遠隔制御応用を許します。
  - **旧来の周辺機能の支援:** より古い周辺機能に対して過去互換性を提供します。多くのより古いｼｽﾃﾑは通信に直列または並列のﾎﾟｰﾄを使います。殆どのより新しいｺﾝﾋﾟｭｰﾀはそれらのﾎﾟｰﾄをなくし、代わりに、それらはAVR DUでの橋渡し応用を使って模倣することができます。
- **教育と試作:**
  - **学習道具:** この例はUSB通信とﾏｲｸﾛ ｺﾝﾄﾛｰﾗのﾌﾟﾛｸﾞﾗﾐﾝｸﾞについて学ぶ学生と趣味人に対する教育的な道具です。橋渡し応用を設定して使う実地体験を提供します。
  - **迅速試作:** 開発者は外部ﾊｰﾄﾞｳｪｱとﾎｽﾄ ｺﾝﾋﾟｭｰﾀ間の通信に橋渡し応用でCDCを活用することによって新しい発想を素早く試作して試験することができ、開発処理を加速して商品化までの時間を減らします。

### 5.3.3. 操作

含まれたREADME文書はこの例を設定して使うための包括的な手引きを含みます。

この段階的な手引きは端末応用を使う例を実演します。

1. READMEで記述されるようにﾊｰﾄﾞｳｪｱを接続してください。
2. 例の序説項でﾘﾝｸされたﾌﾟﾛｼﾞｪｸﾄをﾀﾞｳﾝﾛｰﾄﾞして開くか、または準備の手引きを使ってMPLAB Xでそれを作成してください。
3. ﾃﾞﾊﾞｲｽに書いてください。
4. puTTYまたは他の端末応用を開いてAVR DUとCuriosity Nanoﾃﾞﾊﾞｯｶﾞに対する端末作業を開始してください。
5. 端末ｳｨﾝﾄﾞｳに何かを入力して他の端末ｳｨﾝﾄﾞｳにそれを現れるのを観察してください。

含まれたPython®ｽｸﾘﾌﾟﾄは手動で端末作業を初期化することなく、仮想ｼﾘｱﾙ ﾎﾟｰﾄ例を他の応用に統合することを実演します。上の3.までの段階に従い、その後に以下を続けてください。

1. 例のﾌｫﾙﾀﾞにPythonｽｸﾘﾌﾟﾄを配置してください。
2. 設定に何か変更が行われた場合はｽｸﾘﾌﾟﾄを更新してください。
3. ｽｸﾘﾌﾟﾄを走らせて両方向で送信した16進値がASCII文字として送り返されるのを観察してください。

この例はどのｼﾘｱﾙ ﾃﾞｰﾀもUSARTとで変換するのを実装することができるUSART橋渡し応用を実演します。小さな変更で、CDCを使ってどの直列ｲﾝﾀｰﾌｪｰｽもﾎｽﾄ ｺﾝﾋﾟｭｰﾀに接続するように適応することができます。

### 5.3.4. 結果

この例に従うことにより、開発者は以下を達成するでしょう。

1. **USART橋渡しへのUSB CDCの成功裏の設定:** AVR DUはCDC装置として識別するように構成設定され、仮想ｼﾘｱﾙ ﾎﾟｰﾄ上でﾎｽﾄ ｺﾝﾋﾟｭｰﾀと通信することを許します。
2. **ﾃﾞｰﾀの送受信:** 例応用は2つの端末応用間でｼﾘｱﾙ ﾃﾞｰﾀを送信します。
3. **端末応用との統合:** この例は外部ﾊｰﾄﾞｳｪｱでの監視と相互通信のための端末応用の使用を実演します。
4. **自動化用Pythonｽｸﾘﾌﾟﾄ:** Pythonｽｸﾘﾌﾟﾄは仮想ｼﾘｱﾙ ﾎﾟｰﾄ上でのﾃﾞｰﾀ送受信の自動化を提供し、AVR DUを他のｿﾌﾄｳｪｱ ﾂｰﾙとどう統合するかを示します。

この例の終了で、開発者はﾏｲｸﾛ ｺﾝﾄﾛｰﾗまたは外部ﾊｰﾄﾞｳｪｱとﾎｽﾄ ｺﾝﾋﾟｭｰﾀ間で信頼できるﾃﾞｰﾀ交換を必要とするより複雑な応用の作成を許すAVR DUでのUSB CDC通信実装をしっかりと理解するでしょう。

# 6. 関連資料

- GitHubでのAVR® DUマイクロ コントローラを使うUSB CDC仮想シリアル ポート
- GitHubでのAVR® DUマイクロ コントローラを使うUSART橋渡しのためのUSB CDC
- USB装置階層使用者の手引き
- AVR DU系統の頁
- AVR DU CuriosityNano強化キット

## 7. 改訂履歴

| 文書改訂 | 日付 | 注釈 |
|---|---|---|
| A | 2025年1月 | 初版文書公開 |

# Microchip情報

## 商標

“Microchip”の名称とﾛｺﾞ、“M”のﾛｺﾞ、それと他の名称、ﾛｺﾞ、商標は米国や他の国に於けるMicrochip Technology Incorporatedまたはその系列会社や子会社の登録または未登録の商標です(“Microchip商標”)。Microchip商標に関する情報はhttps://www.microchip.com/en-us/about/legalinformation/microchip-trademarksで見つけることができます。

## 法的通知

この刊行物と契約での情報は設計、試験、応用とのMicrochip製品の統合を含め、Microchip製品でだけ使えます。他の何れの方法でのこの情報の使用はこれらの条件に違反します。ﾃﾞﾊﾞｲｽ応用などに関する情報は皆さまの便宜のためにだけ提供され、更新によって取り換えられるかもしれません。皆さまの応用が皆さまの仕様に合致するのを保証するのは皆さまの責任です。追加支援については最寄りのMicrochip営業所にお問い合わせ頂くか、www.microchip.com/en-us/support/design-help/client-support-servicesで追加支援を得てください。

**この情報はMicrochipによって「現状そのまま」で提供されます。Microchipは非侵害、商品性、特定目的に対する適合性の何れの黙示的保証やその条件、品質、性能に関する保証を含め、明示的にも黙示的にもその情報に関連して書面または表記された書面または黙示の如何なる表明や保証もしません。**

**如何なる場合においても、Microchipは情報またはその使用に関連するあらゆる種類の間接的、特別的、懲罰的、偶発的または結果的な損失、損害、費用または経費に対して責任を負わないものとします。法律で認められている最大限の範囲で、情報またはその使用に関連する全ての請求に対するMicrochipの全責任は、もしあれば、情報のためにMicrochipへ直接支払った料金を超えないものとします。**

生命維持や安全応用でのMicrochipﾃﾞﾊﾞｲｽの使用は完全に購入者の危険性で、購入者はそのような使用に起因する全ての損害、請求、訴訟、費用からMicrochipを擁護し、補償し、免責にすることに同意します。他に言及されない限り、Microchipのどの知的財産権下でも暗黙的または違う方法で許認可は譲渡されません。

## Microchipﾃﾞﾊﾞｲｽ ｺｰﾄﾞ保護機能

Microchip製品での以下のｺｰﾄﾞ保護機能の詳細に注意してください。

- Microchip製品はそれら特定のMicrochipﾃﾞｰﾀｼｰﾄに含まれる仕様に合致します。
- Microchipは動作仕様内で意図した方法と通常条件下で使われる時に、その製品系統が安全であると考えます。
- Microchipはその知的所有権を尊重し、積極的に保護します。Microchip製品のｺｰﾄﾞ保護機能を侵害する試みは固く禁じられ、ﾃﾞｼﾞﾀﾙ ﾐﾚﾆｱﾑ著作権法に違反するかもしれません。
- Microchipや他のどの半導体製造業者もそれのｺｰﾄﾞの安全を保証することはできません。ｺｰﾄﾞ保護は製品が”破ることができない”ことを当社が保証すると言うことを意味しません。ｺｰﾄﾞ保護は常に進化しています。Microchipは当社製品のｺｰﾄﾞ保護機能を継続的に改善することを約束します。

---

日本語© *HERO* 2025.

本応用記述はMicrochipのAN5792応用記述(DS00005792A-2025年1月)の翻訳日本語版です。日本語では不自然となる重複する形容表現は省略されている場合があります。日本語では難解となる表現は大幅に意訳されている部分もあります。必要に応じて一部加筆されています。頁割の変更により、原本より頁数が少なくなっています。

必要と思われる部分には( )内に英語表記や略称などを残す形で表記しています。

青字の部分はﾘﾝｸとなっています。一般的に赤字の0,1は論理0,1を表します。その他の赤字は重要な部分を表します。