تشریح پورت USB

ناصرقاسمی 7811212316
کارشناسی پیوسته کامپیوتر نرم افزار
naser_ghasemi@hotmail.com

چکیده

ما در این مقاله به بررسی خصوصیات پورت USB می پردازیم.دز قسمت اول مقدمه ای راجع به تعریف پورت و انواع آن و نیز محل قرارگیری پورت در یک کامپیوتر شخصی[1] مطرح می کنیم. سپس انگیزه تولید پورت USB و اهداف طراحی وهمچنین طبقه بندی کاربردهای آن و نیز ویژگیهای آن را بیان می کنیم. بعد از آن مقداری از خصوصیات فیزیکی و الکتریکی آن را می گوییم. سپس ارتباط بین قسمت های نرم افزاری و سخت افزاری آن را بیان می کنیم. در ابتدا نحوه ارتباط یک ابزار USB و سیستم میزبان ، مکانیزم های کنترل و نیز جریان داده ها را مطرح می کنیم. سپس وظایف کامپیوتر میزبان دز قبال این پورت و ابزار های متصل به آن را بیان می کنیم. در پیاده سازی ، این پوزت به چند لایه تقسیم می شود که ما به شرح هر لایه و پیکربندی های آن می پردازیم. سپس چگونگی انتقال داده و خطاهای انتقال مطرح می کنیم. سپس مدیریت منابع و راه انداز[2] های هر لایه را بررسی می کنیم.

[1] PC (Personal Computer)
[2] Driver

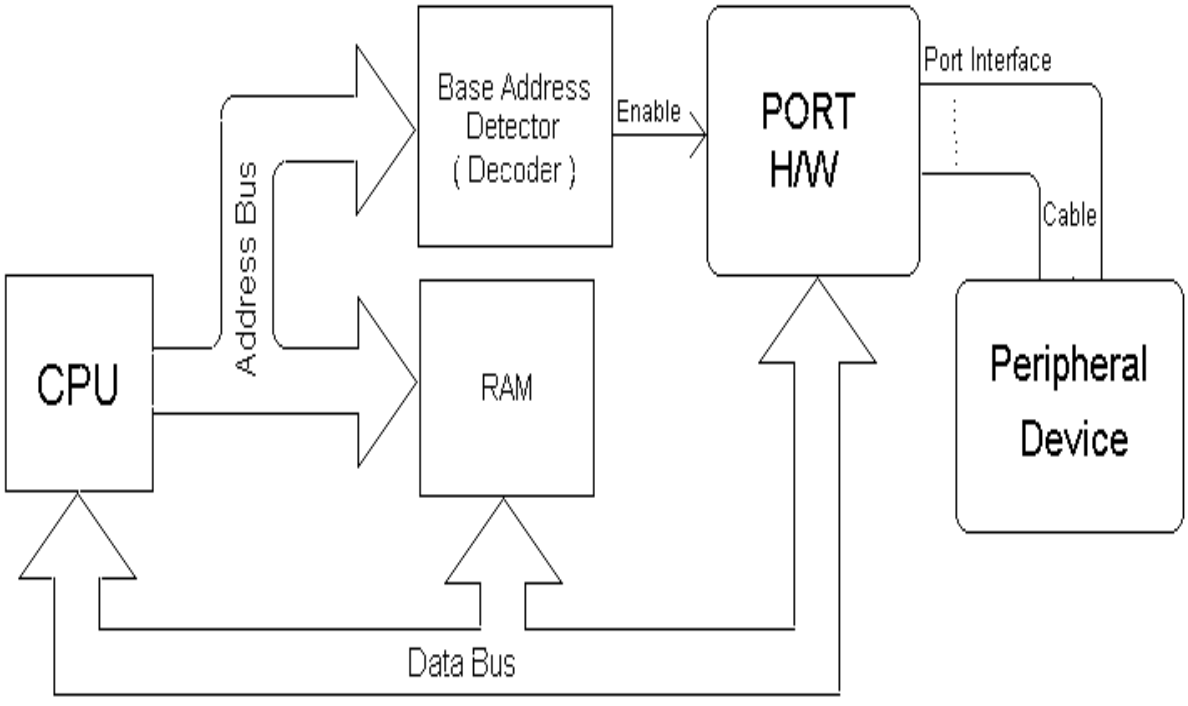
کلمات کلیدی

پورت، USB ، راه انداز USB ، میزبان USB، USB Root Hub ، پیکربندی[3] USB

1- مقدمه

تعریف و انواع پورت[4]

پورت (یا گذرگاه) یک واسط است بین پردازنده مرکزی[5] و سخت افزارهای دیگر. پورت شبیه یک کانال ورودی یا خروجی دیتا است که پردازنده اصلی را به یک سخت افزار جانبی متصل می کند. هر پورت دارای یک آدرس می باشد که با استفاده از آن آدرس می توان به آن پورت دسترسی پیدا کرد. نمودار زیر محل قرار گرفتن و نیز ارتباط یک پورت را با پردازنده مرکزی و حافظه اصلی نشان می دهد.



نمودار 1 : محل قرارگیری پورت در یک کامپیوتر

[3] Configuration
[4] Port
[5] CPU

دسترسی به یک پورت به این صورت است که ریز پردازنده دستوری دارد که با استفاده از آن می توان داده ای را در یک پورت با آدرس مشخص کپی کرد. به عنوان مثال ریزپردازنده 8085 دستورالعملی به نام OUT دارد که عملوند آن یک آدرس است و به عنوان مثال با نوشتن OUT 1000h مقدار موجود در رجیستر A را در پورتی با آدرس h1000 کپی می کند. در جدول زیر آدرس چند پورت در یک کامپیوتر شخصی آمده است.

پورت	آدرس در PC
Hard Drive Controller	320 – 32F
First Serial Interface	3F8 – 3FF
Second Serial Interface	2F8 – 2FF
Disk Controller	3F0 – 3F8
Interface for Parallel Printer	378 – 37F

یکی از انواع پورت **پورت های سریال**[6] می باشند. ین نوع پورت یکی از اجزاء اصلی یک PC می باشد. پورت پارالل 8 بیت داده را در واحد زمان انتقال می دهد اما پورت سریال فقط یک بیت را در واحد زمان انتقال می دهد. بنابراین 8 برابر کندتر از **پورت های پارالل**[7] می باشد. در سال 1969 مؤ سسه EIA[8] استاندارد RS-232 را تاسیس کرد. در این استاندارد پروتکل انتقال اطلاعات ، کابل ها ، سیگنالهای الکتریکی و رابط ها برای یک ارتباط RS-232 تعریف شده است.

انگیزه تولید این پورت

انگیزه تولید پورت USB سه مورد (وابسته به هم) زیر می باشد :

- **اتصال PC به تلفن با سرعت بالا**
- **استفاده آسان**

ترکیب واسط های گرافیکی کاربر و مکانیزم های سخت افزاری و نرم افزاری همردیف با تولید معماری های جدید گذرگاه ، کامپیوترهایی را ساخته اند که کمتر نیاز به پیکربندی دارد و نیز پیکربندی دوباره آن آسان است. از دیدگاه کاربر ، واسط های ورودی و خروجی PC ، مانند پورت های سریال و پارالل ، واسط های صفحه کلید /ماوس/دسته بازی و غیره خصوصیت PNP[9] را ندارند.

---

[6] Serial Ports

[7] Parallel Port

[8] Electronic Industries Association

[9] Plug-and-Play

- **توسعه پورت**

اضافه شدن وسایل جانبی خارجی ادامه یافت تا جاییکه دسترس پذیری پورت ها را تحت فشار قرار گرفت. فقدان یک گذرگاه جانبی دارای دوجهت، کم هزینه، دارای سرعت کم تا متوسط مانع ازدیاد وسایل جانبی مانند فاکس/مودم، ماشینهای جوابگو[10] ، اسکنر ها و ... شد. برای هر عملکرد یا توانایی که به PC اضافه می شود، واسط های جدیدی برای پاسخ گویی به این نیاز تعریف می شود.USB پاسخی برای نیازهای فوق می باشد. چرا که یک واسط سریال سریع ، هم زمان ، کم هزینه و اتصال پویا می باشد.

اهداف طراحی

USB برای توسعه استاندارد شده در معماری PC بیشتر به منظور بهره گیری از تلفن کامپیوتری (CTI)[11] ، مصرف کننده و کاربرد های کارا و مفید تعیین شده است. معیار های زیر در معماری USB در نظر گرفته شده است :

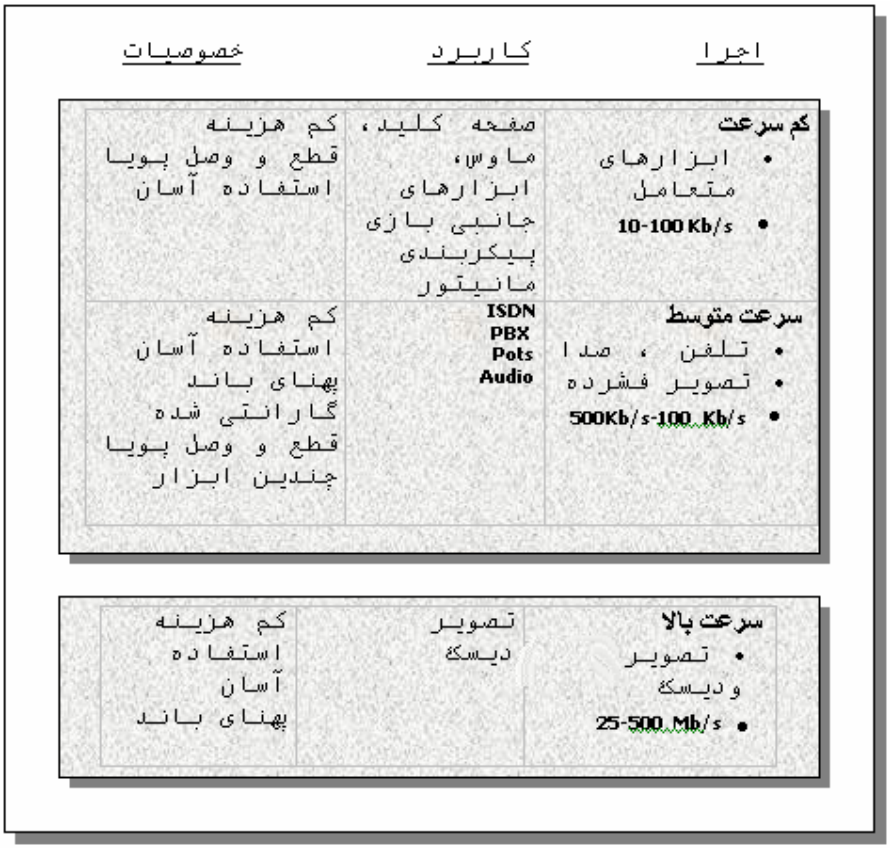
- استفاده آسان برای توسعه جانبی PC .
- راه کم هزینه ای که سرعت انتقال اطلاعات حداکثر تا 12 Mb/s پشتیبانی می کند.
- پشتیبانی کامل برای داده های برای صدا و تصویر فشرده
- پروتکل انعطاف پذیر برای انتقال ترکیب داده های هم زمان و غیر هم زمان
- اجتماع در تکنولوژی ابزار مناسب
- سازگار با پیکربندی های مختلف و فاکتور های تشکیل کامپیوترشخصی
- ارایه یک واسط استاندارد که توانایی انتشار سریع را دارد
- پشتیبانی کلاس های جدیدی از ابزارها که افزون بر توانایی های PC می باشد

طبقه بندی کاربردها

نمودار 2 یک طبقه بندی بر اساس مقدار داده ای که می تواند توسط USB سرویس دهی شود. همانگونه که می بینید ، یک گذرگاه 12 Mb/s در محدوده داده های کم سرعت و با سرعت متوسط قرار می گیرد. نوعا" انواع داده با سرعت متوسط همزمان هستند در حالیکه داده های با سرعت کم از ابزار های متعامل[12] می آیند. این USB که پیشنهاد شد یک گذرگاه رومیزی[13] می باشد

[10] Answering Machine
[11] Computer Telephony Integration
[12] Interactive
[13] Desktop

البته براحتی می تواند در محیط های سیار[14] نیز بکار رود. معماری نرم افزار USB از طریق فراهم کردن امکان پشتیبانی چند USB ، اجازه می دهد در آینده ,USB توسعه هایی پیدا کند.



نمودار 2 : طبقه بندی کاربردها

خصوصیات این پورت

مشخصات USB امکان انتخاب خصوصیاتی از USB را داده و می تواند عملکردهایی را که اجازه مشتق نمودن در سطح سیستم واجزاء را می دهند، فعال سازد. این خصوصیات در قالب مزایای زیر دسته بندی می شوند :

[14] Mobile

❖ آسان بودن استفاده آن توسط کاربرها

- وجود تنها یک مدل برای کابل کشی و اتصالات
- جزییات الکتریکی از کاربر جدا و پنهان نگاه داشته می شود
- ابزارهای جانبی که هر یک هود را معرفی می کنند و نیز نگاشت اتوماتیک عملکردها به راه انداز و نیز پیکربندی
- ابزارهای جانبی که به صورت پویا قابل اتصال به سیستم بوده و قابلیت پیکربندی مجدد را دارند

❖ محدوده وسیع کابردها

- مناسب جهت پهنای باندهای مختلف از چند کیلو بیت در ثانیه تا چندین مگا بیت در ثانیه
- پشتیبانی انتقالات همزمان و نا همزمان بر روی یک سیسم
- پشتیبانی عملیات همزمان چندین ابزار (چندین ارتباط)
- پشتیبانی حداکثر تا 127 ابزار فیزیکی
- پشتیبانی انتقال چندین جریان داده و پیغام بین میزبان و ابزارها
- قابلیت کار با ابزارهای مرکب (به معنای ابزارهایی که از چندین عملکرد تشکیل شده اند)
- سربار[15] جزیی پروتکل که منجر به بهره بری بیشتر از گذرگاه می شود.

❖ پهنای باند همزمان

- پهنای باند گارانتی شده و با تأخیر کم که برای تلفن و صدا مناسب می باشد
- یک کار همزمان ممکن است از تمام پهنای باند گذرگاه استفاده کند

❖ قابلیت انعطاف

- محدوده های وسیعی از اندازه های پاکت را پشتیبانی می کند که این منجر به وجود گزینه های زیادی برای بافرینگ ابزار می شود
- کنترل جریان برای کارکردن با بافر در پروتکل گنجانده شده است

[15] Overhead

❖ استحکام

- مکانیزم بازیافت خطا و کار با خطا در پروتکل گنجانده شده است
- امکان افزودن یا حذف ابزارها به صورت پویا
- قابلیت تشخیص ابزارهای خطادار

❖ هماهنگی با صنعت PC

- پیاده سازی و الحاق پروتکل ساده است
- هماهنگ با معماری Plug And Play در PC
- استفاده از واسط های سیستم عامل های موجود

❖ پیاده سازی کم هزینه

- کانال فرعی کم هزینه با سرعتی معادل 1.5 مگا بیت در ثانیه
- بهینه شده برای مجتمع کردن آن در ابزار ها و سخت افزار میزبان
- مناسب جهت توسعه ابزارهای کم هزینه
- اتصالات و کابل های کم هزینه
- بهره گِیری از تکنولوژی های موجود و مناسب

❖ مسیر ارتقاء

- معماری قابل ارتقاء جهت پشتیبانی چند USB HC در یک سیستم

2- سخت افزار و نرم افزار

اتصال داخلی USB تبادل داده ها[16] بین میزبان[17] (یک کامپیوتر شخصی) و یک وسیله جانبی[18] را پشتیبانی می کند. این فصل درباره واسط های[19] لازم برای تسهیل ارتباط بین یک نرم افزار

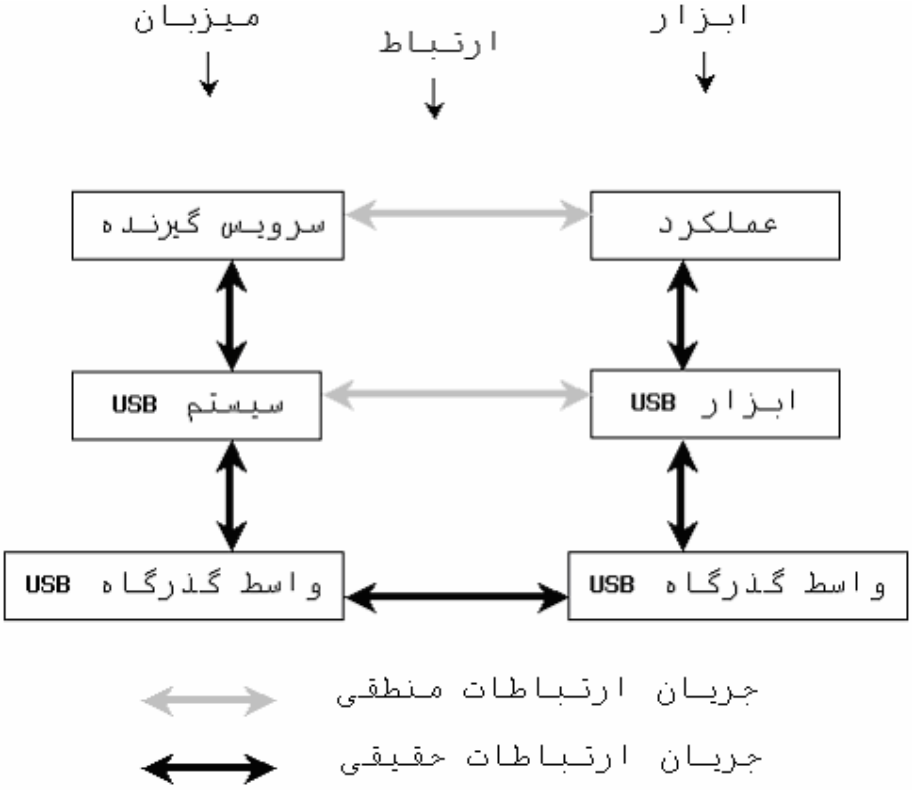
[16] Data
[17] Host
[18] Peripheral Device
[19] Interface

سرویس گیرنده[20] ،که بر روی یک کامپیوتر شخصی[21] قراردارد ، و یک عملکرد[22] که بر روی سخت افزار پیاده سازی می شود، توضیح می دهد.

1-2 مروری بر میزبان USB

ارتباط پورت USB با دیگر قطعات در شکل 3 نشان داده شده است.

میزبان و سخت افزار جانبی به لایه های مجزایی تقسیم شده اند که در شکل زیر نشان داده شده است. فلش های عمودی ارتباط حقیقی در میزبان را مشخص می کند. واسط های مشابه در سخت افزار جانبی یک سری پیاده سازی های[23] ویژه می باشند. تمام ارتباطات بین میزبان و سخت افزار جانبی نهایتا" از طریق کابل USB برقرار می شود. اگرچه واسط های منطقی بین دو لایه افقی هم سطح وجود دارد، اما این ارتباطات بین نرم افزار سرویس گیرنده و عملکردهای مهیا شده بوسیله سخت افزار، به احتیاجات نرم افزاری که در حال حاضر از سخت افزار استفاده می کند و نیز توانایی های سخت افزار بستگی دارد.



شکل 3 : جریان ارتباطات میان ابزار جانبی و میزبان

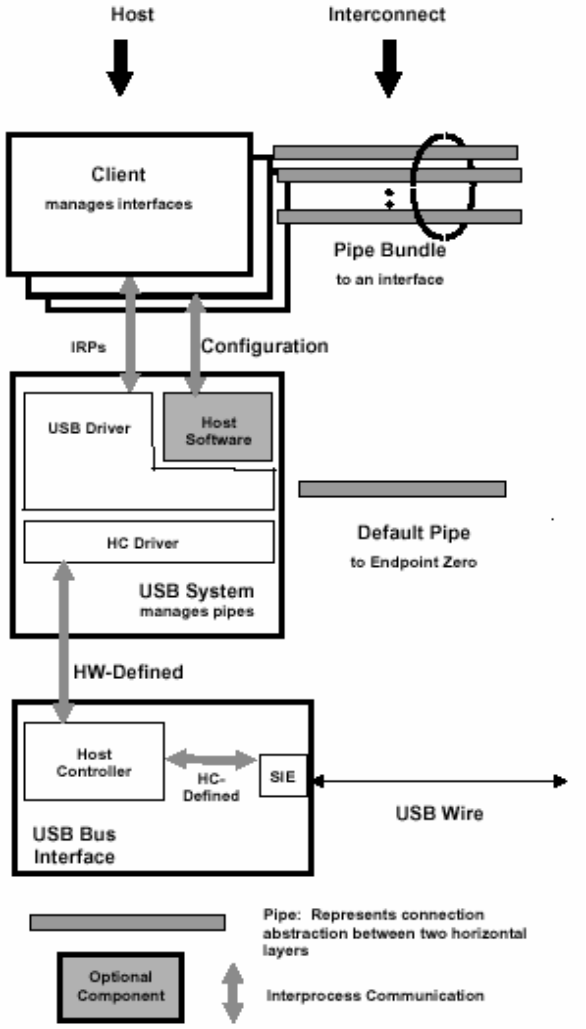
[20] Client
[21] Personal Computer
[22] Function
[23] Implementation

تاثیر متقابل سرویس گیرنده و عملکردهای سخت افزار نیازهایی را برای لایه های زیرین و واسط هایشان ایجاد می کند.
این فصل این مدل را از دیدگاه میزبان و لایه هایش توضیح می دهد. شکل4 ارتباط میزبان با سخت افزار جانبی را نشان می دهد.

تنها یک میزبان (کامپیوتر شخصی) برای یک پورت USB وجود دارد. مهمترین لایه های یک میزبان عبارتند از :

- USB bus interface
- USB System
- سرویس گیرنده

USB bus interface با تاثیرات متقابل لایه الکتریکی و لایه پروتکل سر و کار دارد. از نقطه نظر اتصالات داخلی، USB bus interface توسط سخت افزار جانبی و میزبان مهیا شده است. به عنوان مثال مانند Serial Interface Engine (SIE) . بر روی Host اگر چه ، USB bus interface مسؤلیت های اضافی ناشی از وظیفه یکتای Host در برابر USB دارد و مانند Host Controller پیاده سازی می شود.



نمودار 4 : ارتباطات میزبان

USB System از Host Controller برای مدیریت انتقال داده ها بین میزبان و سخت افزار جانبی استفاده می کند. واسط بین این دو قسمت بستگی به سخت افزار Host Controller دارد. USB System هماهنگ با Host Controller تبدیل بین داده های انتقال یافته از طرف میزبان و داده های موجود در اتصالات USB را انجام می دهد. سیستم USB همچنین مسول مدیریت منابع USB ، مانند عرض باند، و تغذیه BUS[24] را بر عهده دارد، بنابراین دسترسی سرویس گیرنده امکان پذیر می شود.

[24] Bus Power

USB System سه قسمت اصلی دارد :

- راه انداز کنترل کننده میزبان[25] (HCD)
- راه انداز USBD) USB[26])
- نرم افزار میزبان[27]

HCD به منظور طراحی ساده تر پیاده سازی های گوناگون کنترل کننده میزبان که در داخل سیستم USB قرار دارد ، به کار می رود ، به این صورت که یک سرویس گیرنده می تواند با سخت افزار جانبی اش ارتباط دوطرفه داشته باشد بدون اینکه بداند که سخت افزار به چه کنترل کننده ای متصل است.

راه انداز (USB (USBD واسط اصلی میزبان [28] را برای ارتباط سرویس گیرنده و سخت افزار مهیا می کند. واسط بین HCD و USBD با نام واسط راه انداز کنترل کننده میزبان[29] (HCDI) شناخته می شود. سرویس گیرنده ها مستقیما" نمی توانند به این واسط دسترسی پیدا کنند . یک HCDI بخصوص ، توسط هر سیستم عاملی که پیاده سازی های HC را پشتیبانی کند ، تعریف می شود.

USBD مکانیزم انتقال داده ها را به صورت پاکت های درخواست ورودی و خروجی[30] (IRPs) پیاده سازی می کند که شامل درخواست انتقال داده از میان یک کانال داده ای[31] می باشد. علاوه بر این USBD مسؤل آماده کردن تجریدی[32] از سخت افزار برای سرویس گیرنده هایش می باشد. [در واقع سرویس گیرنده فقط از طریق راه انداز می تواند به سخت افزار دسترسی پیدا کند.] که این تجرید می تواند برای پیکربندی[33] و مدیریت حالات[34] دستکاری شود. USBD به عنوان قسمتی از این تجرید دارای یک کانال داده ای می باشد که تمام ابزار های USB به آن دسترسی دارند. این کانال داده ای نمایانگر یک ارتباط منطقی بین USBD و تجرید می باشد.که در شکل 5-2 نشان داده شده است.

---

[25] Host Controller Driver
[26] USB Driver
[27] Host Software
[28] USB Host Interface
[29] Host Controller Driver Interface
[30] I/O Request Packets
[31] Pipe
[32] Abstraction
[33] Configuration
[34] States

در بعضی از سیستم عامل ها یک نرم افزار سیستم غیر USB [35] در دسترس است که مکانیزم های لود و پیکربندی را برای راه انداز سخت افزار فراهم می کند. در چنین سیستم عامل هایی ، راه انداز سخت افزار باید از این واسط ها به جای دسترسی مستقیم به مکانیزم های USBDI استفاده کند.

لایه سرویس گیرنده شامل تمام ماجول های نرم افزاری[36] می شود که مسؤل ارتباط مستقیم با ابزار USB هستند. هنگامی که هر وسیله به سیستم متصل می شود ، این سرویس گیرنده ها ممکن است با سخت افزار جانبی ارتباط مستقیم برقرار کنند.

به طور کلی لایه های Host توانایی های زیر را دارند :

- شناسایی متصل شدن و قطع شدن ابزارهای USB
- مدیریت جریان اطلاعات کنترلی بین میزبان و ابزار USB
- مدیریت جریان داده ها بین میزبان و ابزار USB
- جمع آوری آمار حالات و فعالیت ها
- کنترل واسط الکتریکی بین HC و ابزار USB

در قسمت های زیر این مسؤلیت ها و همچنین احتیاجات USBDI با جزییات بیشتر توضیح داده می شود. واسط های حقیقی که برای ترکیب ویژه از سخت افزار[37] میزبان و سیستم عامل استفاده می شود که در محیط راهنمای[38] سیستم عامل مناسب شرح داده می شود.

تمام مراکز[39] تغییر حالات داخلی و تغییر حالات پورت هایشان را توسط کانال داده ای تغییر حالت[40] گزارش می کنند. مثلا" اطلاع دادن اینکه یک ابزار USB به پورتش وصل یا از آن قطع شده است. یک سرویس گیرنده راه انداز USB عموما" می داند که راه انداز Hub این اطلاعات را دریافت کرده یا نه و همچنین مالک تغییر حالت Hub را می شناسد. در بعضی از سیستم ها راه

[35] Non-USB System Software
[36] Software Entities
[37] Platform
[38] Environment Guide
[39] Hub
[40] Status Change Pipe

انداز Hub قسمتی از نرم افزار میزبان می باشد که توسط سیستم عامل برای مدیریت ابزارها مهیا شده است.

مکانیزم های کنترل

اطلاعات کنترلی ممکن است میزبان و یک ابزار USB از طریق سیگنالهای روی باند[41] و یا سیگنالهای خارج از باند[42] عبور کنند. در روش اول اطلاعات کنترلی با داده ها در یک کانال داده ای ترکیب می شوند، در روش دوم اطلاعات کنترلی در یک کانال داده ای مجزا قرار داده می شوند.

یک کانال داده ای پیغام وجود دارد که کانال پیش فرض[43] برای هر ابزار USB متصل شده به سیستم، نام دارد. این ارتباط منطقی بین میزبان و یک ابزار USB برای جریان اطلاعات کنترلی استفاده می شود.

یک ابزار USB ویژه ممکن است اجازه استفاده از کانالهای اضافی برای انتقال اطلاعات کنترلی ویژه سخت افزار را بدهد. این کانال ها مانند کانال پیش فرض از پروتکل های ارتباطی مشابه استفاده می کنند اما اطلاعات انتقال یافته ویژه ابزار USB هستند و جزء استاندارد مشخصات استاندارد شده[44] USB نیستند.

راه انداز USB ، کانال پیش فرض مشترک را پشتیبانی می کند که این کانال متعلق به سرویس گیرنده هاست و آنها از این کانال استفاده می کنند.

[41] In-Band Signaling
[42] Out-Of-Band Signaling
[43] Default Pipe
[44] Specification

جریان داده ها

HC مسؤل جریان انتقال داده ها بین میزبان و ابزار USB می باشد. این انتقال داده ها مانند یک جریان پیوسته از بایت هاست. USB چهار نوع انتقال داده را پشتیبانی می کند :

- انتقال اطلاعات کنترلی[45]
- انتقال غیر همزمان[46]
- انتقال وقفه ای[47]
- انتقال توده ای[48]

هر وسیله یک یا چند واسط را فراهم می کند که یک سرویس گیرنده ممکن است از آنها برای ارتباط با ابزار استفاده کند. هر واسط از صفر یا چند کانال داده ای ساخته شده است که هر یک به طور جداگانه داده ها را بین یک سرویس گیرنده و یک نقطه انتهایی ویژه[49] انتقال می دهند. راه انداز USB (USBD) واسط ها و کانالها را در یک درخواست صریح نرم افزار میزبان برقرار می کند. هنگامی که درخواست پیکربندی داده شد HC سرویسی را بر اساس پارامترهایی که توسط نرم افزار میزبان مهیا شده است ، ارایه می کند.

یک کانال داده ای چندین مشخصه اصلی دارد. چند نمونه از این مشخصه ها عبارتند از :

- سرعت انتقال اطلاعات
- داده ها با سرعت ثابت و پیوسته انتقال می یابند یا تک تک و پراکنده
- داده ها چه مدت قبل از تحویل ممکن است به تعویق بیافتند
- آیا از دست رفتن داده های انتقال یافته فاجعه انگیز می شوند

یک نقطه انتهایی در ابزار USB مشخصه هایی را دارد که برای یک کانال ویژه ، مورد نیاز است. نقاط انتهایی به عنوان قسمتی از اطلاعات توصیفی[50] ابزار USB می باشد.

---

[45] Control Transfer
[46] Isochronous Transfer
[47] Interrpt Transfer
[48] Bulk Transfer
[49] Particular Endpoint
[50] Characterization Information

جمع آوری آمار حالات و فعالیت ها

سیستم USB و HC به عنوان ارتباط دهنده مشترک برای تمام انتقالات داده ای و کنترلی بین میزبان و ابزار USB ، از جایگاه خوبی جهت پیگیری اطلاعات حالات و فعالیت ها برخوردار هستند. چنین اطلاعاتی به محض اینکه درخواستی از طرف نرم افزار میزبان می آید، تحویل داده می شوند و نرم افزار اجازه می یابد که اطلاعات حالات و فعالیت ها را مدیریت کند.

ملاحظات واسط الکتریکی

میزبان تغذیه ابزار USB که به سسیستم متصل است را بر عهده دارد. این تغذیه توسط یک پورت مهیا می شود.

2-2 نیازهای HC

در تمام پیاده سازی ها ، HC وظایف مشابهی در قبال USB و ابزارهای متصل به آن دارد. این وظایف در زیر توضیح داده شده است.
HC نیازهایی هم از میزبان و هم از USB دارد. در زیر مروری مختصر بر عملکردهای مهیا شده ، آمده است. هر یک از این توانایی ها با جزییات شرح داده شده اند.

سریال کننده / غیر سریال کننده	برای عبور دادن داده ها از میزبان ، HC اطلاعات داده ای و پروتکل را از فرمت اصلی اشان به صورت رشته ای از بیت ها که به USB انتقال داده شده اند، تبدیل می کند. برای داده هایی که به میزبان رسیده یک عمل از پیش تعین شده انجام می شود.
تولید Frame	HC علامت آغاز فریم ها SOF را هر 1 میلی ثانیه یکبار تولید می کند.
پردازش داده	HC درخواست های عبور داده از میزبان و انتقال داده به میزبان را پردازش می کند.
موتور پروتکل	HC پروتکلی که توسط USB شناخته شده است را پشتیبانی می

کند.

کار کردن با خطای انتقال	تمام HC ها در هنگام کشف کردن و عکس العمل[51] در برابر دسته خطاهای تعریف شده ، رفتار مشابهی را از خود نشان می دهند.
مطلع ساختن از راه دور[52]	تمام HC ها باید توانایی بردن Bus به حالت معلق[53] و نیز پاسخ دادن به وقایع مطلع کننده Bus را داشته باشند.
مرکز اصلی[54]	مرکز اصلی عملکردهای استانداردی را برای وصل کردن HC به یک یا چند پورت USB مهیا می کند.
واسط سیستم میزبان	یک مسیر داده با سرعت بالا را بین HC و سیستم میزبان آماده می کند.

کار کردن با حالات

HC یک سری حالت دارد که سیستم USB آنها را مدیریت می کنند. علاوه بر این HC واسطی را برای دو ناحیه از حالات مربوط به USB ارایه می کند. این دو ناحیه عبارتند از :

- انتشار تغییر حالت[55]
- مرکز اصلی

مرکز اصلی مانند سایر ابزارهای USB حالات استاندارد مشابهی را به راه انداز Hub (مرکز) ارایه می کند. HC این حالات و گذار[56] این حالات برای Hub را پشتیبانی می کند.

---

[51] Reacting
[52] Remote Wakeup
[53] Suspended
[54] Root Hub
[55] State Change Propagation
[56] Transition

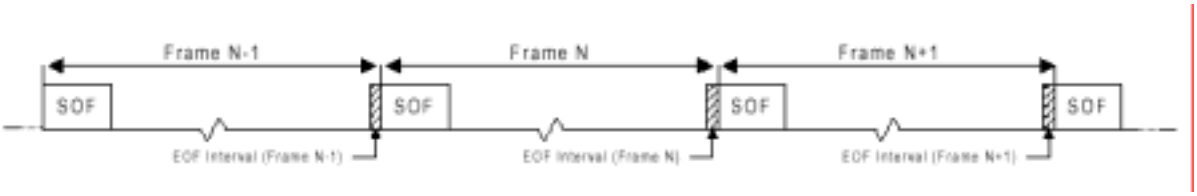
حالت کلی HC مرتبط با حالت Hub اصلی می باشد. هر تغییر حالتی در HC که مرتبط با ابزارهای متصل شده باشد می باید به ابزارهای مربوطه منعکس شده بطوریکه حالت HC و ابزار USB سازگار باشند.

سریال کننده / غیر سریال کننده

انتقال داده ها از میان USB مانند یک رشته بیت فضا اشغال می کند. یک موتور واسط سریال[57] (SIE) به عنوان قسمتی از میزبان یا به عنوان قسمتی از ابزار USB پیاده سازی شده است که انتقال سریالی و غیر سریالی USB را پشتیبانی می کند. این SIE قسمتی از HC می باشد.

تولید فریم ها

تقسیم کردن زمان USB به قسمت های 1 میلی ثانیه ای که "فریم" نامیده می شود، از مسؤلیت های HC می باشد. فریم ها بوسیله HC ایجاد می شوند که در فواصل 1 میلی ثانیه ای علامت های SOF[58] را ایجاد می کند. شکل 5 ایجاد فریم ها را نشان می دهد. علامت SOF اولین داده در هر فریم می باشد. پس از ایجاد SOF، HC در باقیمانده هر فریم برای عبور دادن بقیه اطلاعات آزاد می باشد. هنگامی که HC در حالت عملکرد نرمال خود قرار دارد علامت های SOF باید هر 1 میلی ثانیه یک بار صرفنظر از بقیه فعالیتهای Bus ، تولید شود. اگر HC وارد حالتی شود که در آن تغذیه Bus آماده نیست نباید SOF تولید شود. هنگامی که HC علامت های SOF را تولید نمی کند ممکن است وارد حالت کم کردن تغذیه[59] شود.



شکل 5 : ایجاد فریم ها

[57] Serial Interface Engine
[58] Start Of Frame
[59] Power-Reduced State

علامت های SOF بالاترین حق تقدم دسترسی به Bus را دارند. مدارات الکتریکی در Hub ها ، تمام ارسال کننده های فعال را در طول بازه زمانی EOF[60] از گذرگاه قطع می سازد تا گذرگاه برای ارسال SOF جدید آماده باشد.

HC باید اجازه دهد طول فریم ها با زمان ± 1 بیت تنظیم شود. HC از شماره فریم های جاری که ممکن است توسط سیستم USB خوانده شود، نگهداری می کند.

شماره فریم جاری در موارد زیر بکار می رود :

- برای اینکه فریم هایی با شماره منحصر به فرد داشته باشیم.
- بعد از هر دوره تناوب یک واحد افزایش می یابد.
- تا رسیدن فریم بعدی مجاز است.

میزبان $\underline{11}$ بیت کم ارزش شماره فریم جاری را در هر انتقال علامت SOF ، ارسال می کند. هنگامی که درخواستی از HC می آید، شماره فریم جاری، شماره فریم درخواستی است که انجام شده است. شماره فریمی که توسط میزبان (HC یا HCD) بازگردانده می شود کمتر از 32 بیت است، اگر چه HC احتیاجی به نگهداری بیش از 11 بیت را ندارد.

HC در طول بازه های زمانی EOF باید ارسال اطلاعات را متوقف کند. هنگامی که EOF شروع می شود هر کار زمانبندی شده مشخص برای فریم که باید ارسال شود، قطع می شود. اگر HC در حال اجرای یک تراکنش[61] باشد و در همین لحظه با علامت EOF برخورد کند، آن تراکنش را پایان می دهد.

پردازش داده ها

HC مسؤل دریافت داده ها از سیستم USB و ارسال آن به USB و همچنین مسؤل دریافت داده ها از USB و ارسال آن به سیستم USB می باشد. فرمت خاصی که برای انتقال داده ها بین سیستم USB و HC استفاده می شود، به طور ویژه پیاده سازی شده است.

[60] End OF Frame
[61] Transaction

موتور پروتکل

HC واسط لایه پروتکل USB را مدیریت می کند. HC اطلاعات مربوط به پروتکل را در فریم های ارسالی قرار می دهد. همچنین اطلاعات پروتکل را که دریافت می کند تفسیر می نماید.

کار کردن با خطاهای انتقال

HC می باید توانایی شناسایی و کشف خطاهای رخ داده در فریم های انتقالی را داشته باشد. که از نقطه نظر میزبان عبارتند از :

- حالت timeout که در آن حد اکثر زمانی را که میزبان پس از انتقال یک پاکت یا نشانه صبر خواهد کرد تا خطا گزارش شود ، می باشد. این خطاها زمانی رخ می دهد که مقصد قادر به جوابگویی نبوده یا ساختار اطلاعات انتقالی به قدری مخدوش شده که مقصد قادر به تشخیص اطلاعات دریافتی نمی باشد.
- خطاهای داده که ناشی از گم شدن اطلاعات ارسالی یا غیر مجاز بودن آنها به ترتیب زیرمی باشد :
  - HC یک پاکت (بسته اطلاعاتی در لایه پروتکل) را در زمان کوتاهتری نسبت به زمان انتقال مورد نیاز ارسال و یا در یافت می کند به عنوان مثال می توان از انتقال بعد از EOF نام برد.
  - فیلد CRC غیر مجاز در پاکت دریافتی

- خطای پروتکل
  - PID غیر مجاز که ناشی از عدم توافق اولیه[62] یا توافق اولیه نامناسب می باشد.
  - EOP[63] اشتباه
  - خطای Bit Stuffing

برای هر انتقال داده (چه از نوع یکجا یا توده ای[64] و چه از نوع وقفه ای) ، میزبان می باید تعداد خطاهای رخ داده را نگهداری کند. خطا ناشی از شرایط فوق الذکر می باشد. و هیچ خطایی

[62] Handshake
[63] End Of Protocol
[64] Bulk

بوسیله Nack[65] شناخته نمی شود. اگر تعداد خطاهای مربوط به یک تراکنش به 3 برسد میزبان از ارسال صرفنظر می کند. در اینصورت آخرین خطا به کاربر منعکس می شود تراکنش های هم زمان صرفنظر از نتیجه آن ، فقط یک بار جهت اجرا در نظر گرفته می شوند و لذا هیچ شمارنده ای جهت شمارش خطاهای این نوع تراکنش نگهداری نمی شود.

مطلع ساختن از راه دور

اگر سیستم USB نخواهد که Bus را در حالت معلق (بیکاری موقت) قرار دهد به HC دستوری ارسال می کند تا ارسال ترافیک خود را روی Bus متوقف کند. از قبیل SOF ها . بدین صورت تمام ابزارهای USB در حالت معلق قرار می گیرند در این حالت سیستم USB، HC را قادر می سازد تا پاسخگوی رخدادهای مطلع کننده باشد. بدین ترتیب HC قادر خواهد بود به سیگنالهای مطلع کننده پاسخ داده و سیستم میزبان را مجددا" راه اندازی[66] کند.

مرکز اصلی (Root Hub)

Hub اصلی ارتباط میان HC را با یک یا چند پورت USB برقرار می کند. Hub اصلی از لحاظ عملکرد دقیقا" مانند سایر Hub ها می باشد. جز اینکه واسط سخت افزاری و نرم افزاری بین Hub و HC در غالب پیاده سازی های سخت افزاری خاصی تعریف می شود.

راه اندازی های مجدد پورت

Hub های اصلی مدت زمان راه اندازی مجدد را حد اقل 50 میلی ثانیه در نظر گرفته اند. اگر مدت راه اندازی مجدد توسط سخت افزار کنترل شود و تایمر سخت افزار کمتر از 50 میلی ثانیه باشد سیستم USB می تواند به هر تعداد راه اندازی مجدد پشت سر هم صادر کند تا به میزان کافی این زمان طولانی شود.

[65] Negative Acknowledge
[66] Reset

واسط سیستم میزبان

HC یک گذرگاه واسط پر سرعت به حافظه اصلی (RAM) برقرار می کند. انتقال فیزیکی بین حافظه و کابل USB به طور اتوماتیک توسط HC انجام می شود. زمانیکه بافرهای داده نیاز به پر شدن یا خالی شدن داشته باشند ، HC سیستم USB را مطلع می سازد.

2-3 مروری بر مکانیزم های نرم افزاری

راه انداز HC و راه انداز USB در واقع واسط های نرم افزاری را در سطوح مربوط به خود ارایه می کنند. گرچه این دو راه انداز باید بگونه ای هماهنگ عمل کنندکه نیازهای سیستم USB را برآورده سازد.(نمودار 5-2 را نگاه کنید) نیازهای سیستم USB در مرحله اول به عنوان نیازهای طراحی واسط راه انداز USB بیان می شود. تقسیم وظایف بین راه انداز HC و USB تعریف نشده است گرچه یکی از نیازمندیهای HCDI که می باید حتما" در نظر گرفته شود آن است که در یک سیستم عامل مشخص ، چندین پیاده سازی HC را پشتیبانی می کند.

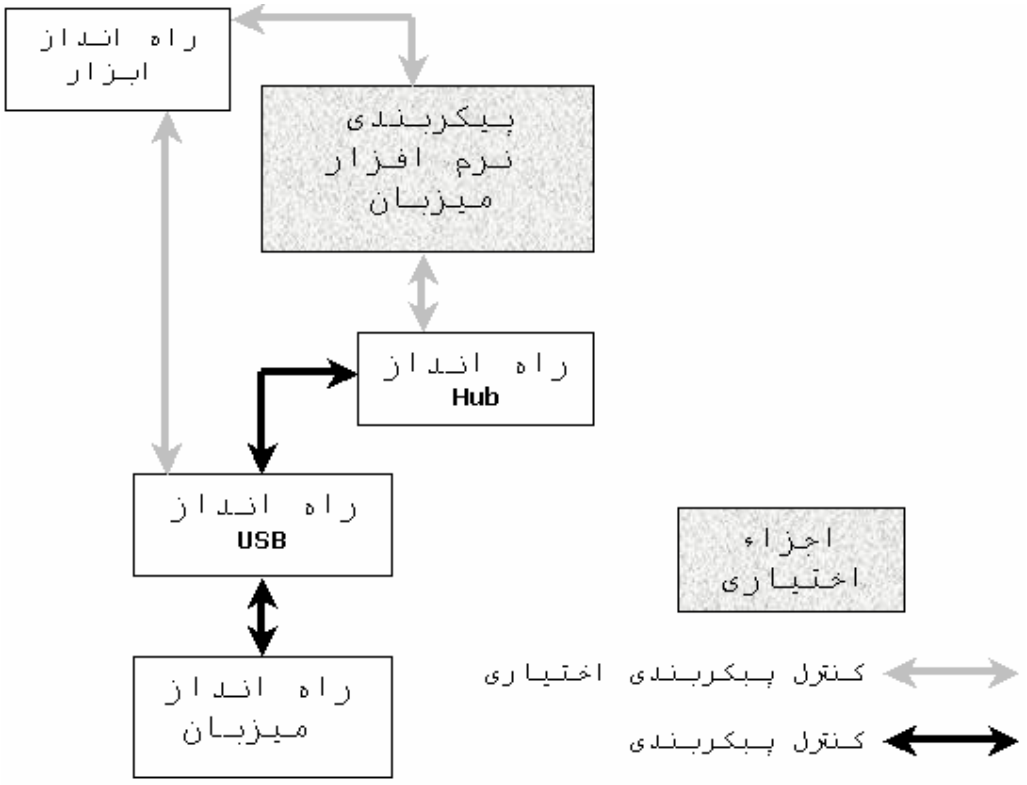
راه انداز HC در واقع تجریدی از HC و نیز انتقال داده از دیدگاه HC روی USB، را فراهم می کند. راه انداز USB نیز تجریدی از ابزار USB و انتقال داده بین راه انداز USB و ابزار USB را فراهم می کند. روی هم رفته سیستم USB به عنوان سیستمی جهت تسهیل انتقال اطلاعات بین سرویس گیرنده و عملکرد ها ، فراهم می کند. همچنین به عنوان نقطه کنترل واسط های خاص USB ابزار USB می باشد. سیستم USB به عنوان سیستمی جهت تسهیل انتقال اطلاعات ، قابلیت های مدیریت بافر را فراهم می کند. همچنین همزمان کردن انتقال داده را مطابق با نیازهای سرویس گیرنده فراهم می سازد.

در قسمت بعد روال های لازم جهت ارسال داده از طریق واسط راه انداز USB تشریح می شود.
پیکربندی ابزار

بسته به نوع سیستم عامل پیکربندی ابزار با استفاده از اجزاء نرم افزاری و سلسله رخدادهای مختلفی انجام می شود. سیستم USB هیچ سیستم عامل خاصی را در نظر نگرفته است اگرچه یک سری نیازمندیهای پایه وجود دارد که می باید توسط هر نوع پیاده سازی سیستم USB لحاظ شود. در برخی از سیستم عامل ها نرم افزار های میزبان موجود ، این نیازمندیها را فراهم

می سازد اما در برخی دیگر سیستم USB خود باید این توانایی ها را فراهم کند. سیستم USB یک سرویس گیرنده خاص برای راه انداز USB به نام راه انداز Hub [67] در نظر می گیرد که جهت افزودن یا حذف ابزار ها از یک Hub بخصوص بکار می رود. زمانیکه این راه انداز Hub در این رابطه (افزودن یا حذف ابزارها) دستوری دریافت کند ، آنگاه نرم افزارهای دیگری از میزبان و سایر سرویس گیرنده های راه انداز USB را در یک سیستم عامل خاص جهت تشخیص و پیکربندی ابزار بکار خواهد گرفت. این مدل در نمودار 6 نشان داده شده است.

زمانیکه یک ابزار USB اضافه می شود راه انداز Hub یک پیغام از Hubی که این تغییر را شناسایی کرده دریافت می کند. راه انداز Hub با استفاده از این اطلاعات که توسط Hub فراهم شده یک شناسه ابزار[68] از راه انداز USB درخواست می کند. راه انداز USB در پاسخ یک کانال داده ای برای آن ابزار ایجاد کرده و پس از آن شناسه ابزار را به راه انداز Hub ارایه می دهد.



نمودار 6 : ارتباطات پیکربندی

در این مرحله ابزار آماده پیکربندی می باشد. برای هر ابزار سه پیکربندی لازم است قبل از آنکه آن ابزار جهت استفاده آماده شود :

[67] Hub Driver
[68] Device Identifier

1. پیکربندی ابزار : این پیکربندی شامل تنظیم تمام پارامترهای USB در آن ابزار و نیز اختصاص تمام منابع میزبان که مرتبط با ابزار است می باشد. این مرحله از طریق تنظیم مقادیر پیکربندی در ابزار انجام می شود. علاوه بر پیکربندی کلی می توان تعداد محدودی از مقادیر پیکربندی را نیز تغییر داد. زمانیکه ابزار پیکربندی شده از دیدگاه ابزار می توان آن را آماده استفاده فرض کرد.
2. پیکربندی USB : به منظور ایجاد یک کانال داده ای راه انداز USB که برای سرویس گیرنده قابل استفاده باشد می باید اطلاعات بیشتری خارج از آنچه برای ابزار لازم است توسط سرویس گیرنده مشخص شود. این اطلاعات که به عنوان خط مشی کانال داده ای[69] شناخته می شود، در واقع نحوه استفاده سرویس گیرنده از کانال داده ای را مشخص می نماید. این اطلاعات شامل مواردی چون حداکثر میزان داده که سرویس گیرنده می تواند با یک درخواست از نوع وقفه منتقل کند، می باشد.
3. پیکربندی عملکردها : پس از آنکه پیکربندی های مرحله 1 و 2 انجام شد، از لحاظ USB ، کانال داده ای کاملا" آماده استفاده می باشد گرچه ممکن است موارد پیکربندی دیگری مانند پیکربندی وابسته به *فروشنده*[70] مورد نیاز باشد قبل از آنکه سرویس گیرنده واقعا" بتواند از کانال داده ای استفاده کند. اما این پیکربندی یک موضوع اختصاصی بین ابزار و سرویس گیرنده است و توسط راه انداز USB استاندارد نمی شود.

مدیریت منابع

هر زمان که یک کانال داده ای توسط راه انداز USB برای یک مقصد برقرار می شود سیستم USB می باید تعیین کند که آیا می تواند کانال داده ای را پشتیبانی کند یا خیر. سیستم USB این تصمیم را بر اساس نیازهایی که در مورد مقصد تشریح شده است ، انجام می دهد. یکی از این نیازها که باید به منظور ایجاد یک کانال داده ای برای یک مقصد، پشتیبانی شود، عبارت از پهنای باند ارسال مورد نیاز جهت انتقال اطلاعات مقصد، می باشد. جهت بررسی پهنای باند در دسترس، دو مرحله وجود دارد : نخست حداکثر زمان اجزای یک تراکنش محاسبه می شود

[69] Pipe Policy
[70] Vendor

سپس با استفاده از جدول زمانبندی فریم تعیین می شود که آیا تراکنش مذکور قابل انجام است یا خیر.

اختصاص پهنای باند برای کانال های داده ای هم زمان و وقفه ای و نیز تعیین اینکه آیا یک تراکنش خاص داده یا کنترل در یک فریم ، قابل ارسال است یا خیر، می تواند توسط یک نرم افزار در سیستم USB انجام شود. اگر زمان اجرای واقعی یک تراکنش در HC از مقدار تعیین شده توسط نرم افزار بیشتر باشد، HC مسؤل تأمین تمامیت[71] فریم می باشد. در ادامه نیازهای این نرم افزار در سیستم USB مورد بحث قرار می گیرد. به منظور تعیین اینکه می توان پهنای باند تخصیص داد و یا اینکه تراکنش در یک فریم انجام شود حداکثر زمان اجرای تراکنش باید محاسبه شود. محاسبه این زمان نیاز به اطلاعات زیر دارد (توجه کنید که برخی از این اطلاعات مکن است لزوما" توسط سرویس گیرنده تعیین نشود) :

- تعداد بایت های داده که باید انتقال پیدا کند.
- نوع انتقال
- عمق توپولوژی : اگر دقت کمتری مورد نیاز باشدحداکثر عمق توپولوژی می باید در نظر گرفته شود.

در این محاسبه باید زمان انتقال بیت (سرعت ارسال)، تاخیر در انتشار[72] سیگنال در توپولوژی و هر گونه تاخیر دیگر که در پیاده سازی خاص وجود دارد از قبیل زمان پردازش در HC در نظر گرفته شود.

انتقال داده ها

پایه تمام ارتباطات سرویس گیرنده و عملکردها، واسط است که عبارت از تعدادی کانال داده ای مرتبط به هم که همگی به یک ابزار USB خاص مربوط می باشد.

در میزبان یک واسط خاص دقیقا" توسط یک سرویس گیرنده مدیریت می شود. این سرویس گیرنده هر کانال آن واسط را از طریق تنظیم خط مشی آن کانال مقداردهی می نماید. که این شامل حداکثر میزان داده که در هر IRP[73] انتقال می یابد و نیز حداکثر زمان سرویس دهی آن کانال می باشد. زمان سرویس دهی کانال داده ای بر حسب میلی ثانیه بیان می شود. و عبارت از

[71] Integrity
[72] Propagation Delay
[73] I/O Request Packet

مدت زمانی است که طول خواهد کشید تا داده IRP برای یک کانال داده ای هم زمان منتقل شود. این زمان ، همچنین بازه زمانی سرکشی به کانال داده ای را برای یک کانال داده ای وقفه ای مشخص می کند. زمانیکه درخواست خاصی کاملا" انجام شود، سرویس گیرنده مطلع خواهد شد. سرویس گیرنده اندازه هر IRP را مدیریت می کند بگونه ای محدودیتهای سیکل وظیفه[74] و تاخیر آن رعایت شود.سایر اطلاعات خط مشی در مورد اطلاعات لازم جهت مطلع کردن سرویس گیرنده می باشد.

سرویس گیرنده فضای بافر مورد نیاز جهت نگهداری داده های انتقال یافته را فراهم می کند. سیستم USB از خط مشی به منظور تعیین فضای کاری اضافی که مورد نیاز خواهد بود، استفاده می کند.

سرویس گیرنده داده های خود را مانند یک جریان اطلاعات پیوسته و سریال می بیند و آنها را به همان طریق که جریان اطلاعاتی روی سایر انواع تکنولوژی های گذرگاه مدیریت می شوند، مدیریت خواهد کرد. سیستم USB بسته به خط مشی خود و محدودیتهای HC درخواست سرویس گیرنده را به درخواست های کوچکتری جهت ارسال روی USB تقسیم می کند. گرچه هنگامی که سیستم USB می خواهد چنین تقسیمی را انجام دهد، دو نیازمندی زیر باید برآورده شوند :

- تقسیم جریان داده به قسمتهای کوچکتر باید از دید سرویس گیرنده مخفی باشد.
- نمونه های USB در تراکنش های ارسالی روی گذرگاه تفکیک نمی شود.

زمانیکه یک سرویس گیرنده قصد انتقال داده را دارد یک IRP به راه انداز USB ارسال می کند. بسته به جهت انتقال داده یک بافر[75] داده خالی یا پر آماده می شود. زمانیکه درخواست کامل شود (به طور موفقیت آمیز یا در اثر رخداد خطا) ، IRP و وضعیت آن به سرویس گیرنده بازگردانده می شود. در مواردی که لازم باشد این وضعیت به ازاء هر تراکنش ارایه می شود.

[74] Duty Cycle
[75] Buffer

4-2 راه انداز HC

HCD در واقع تجریدی از سخت افزار HC و انتقال داده ای روی USB از دیدگاه HC می باشد. واسط راه انداز HC باید نیازهای زیر را برآورده سازد :

- تجریدی از سخت افزار HC فراهم کند.
- تجریدی از انتقال داده روی کابل USB توسط HC فراهم کند.
- تجریدی از تخصیص[76] و آزاد سازی[77] منابع[78] HC جهت ارایه سرویس گارانتی شده به ابزارهای USB فراهم کند.
- بر اساس تعاریف کلاس های Hub، Hub ریشه و رفتارش را ارایه کند. این شامل پشتیبانی Hub ریشه به گونه ای که راه انداز Hub بتواند با آن همانند سایر Hub ها ارتباط برقرار کند، می باشد. در حالت خاص گرچه Hub ریشه می تواند به صورت ترکیبی از سخت افزار و نرم افزار پیاده سازی شود. اما در ابتدا Hub ریشه به آدرس پیش فرض ابزار پاسخ می دهد (از دیدگاه سرویس گیرنده) و در پاسخ اطلاعات توصیفگر[79] را بازمی گرداند که در آن آدرس ابزار مشخص شده و سایر درخواست های کلاس Hub را پشتیبانی می کند. با این وجود تولید تراکنش های گذرگاه به منظور انجام چنین رفتاری ممکن است لازم باشد یا نباشد. و بدین طریق تعامل نزدیکی بین HC و Hub ریشه ایجاد می شود.

راه انداز HC یک واسط نرم افزاری فراهم می کند که تجرید های مورد نیاز را پیاده سازی می کند. وظیفه HCD فراهم نمودن تجریدی است که جزییات سخت افزار HC را مخفی کند. در واقع لایه را به راه انداز HC مربوطه ارجاء می دهیم. یک راه انداز Hub ممکن است چندین HC را مدیریت کند. واسط راه انداز HC مستقیما" توسط سرویس گیرنده قابل دسترسی نیست. بنابراین نیازمندیهای یکدیگر تفاوت می کنند. مکانیزم هایی که توسط راه انداز USB فراهم می شود با استفاده از مکانیزم های فراهم شده توسط سیستم عاملی که USB در آن اجرا می شود، پیاده سازی می شود. در ادمه قابلیت های پایه ای مورد نیاز برای تمامی پیاده سازی های مختلف راه انداز USB ، مورد بحث قرار می گیرد. جهت اطلاع از عملکرد های خاص یک سیستم عامل برای راه انداز USB به راهنمای مربوط به آن سیستم عامل مراجعه کنید. یک نسخه از راه انداز USB دسترسی به یک یا چند راه انداز HC که به نوبت به یک HC یا بیشتر متصل می شوند، را هدایت

[76] Allocation
[77] Deal location
[78] Resource
[79] Descriptor

می کند. چگونگی مدیریت نسخ راه انداز USB در صورت وجود بستگی به سیستم عامل دارد گرچه از دیدگاه سرویس گیرنده، راه انداز USB که سرویس گیرنده با آن در ارتباط است، تمام انواع ابزارهای USB متصل شده را مدیریت می کند.مربوط به واسط راه انداز HC در اینجا مورد بحث قرار نمی گیرد.پایینتر سخت افزار HC همان کابل فیزیکی USB و تمام ابزار های USB متصل شده می باشد. HCD پایینترین لایه نرم افزاری در نرم افزار USB می باشد. . HCD تنها یک سرویس گیرنده دارد که عبارت از راه انداز USBD) USB) می باشد.

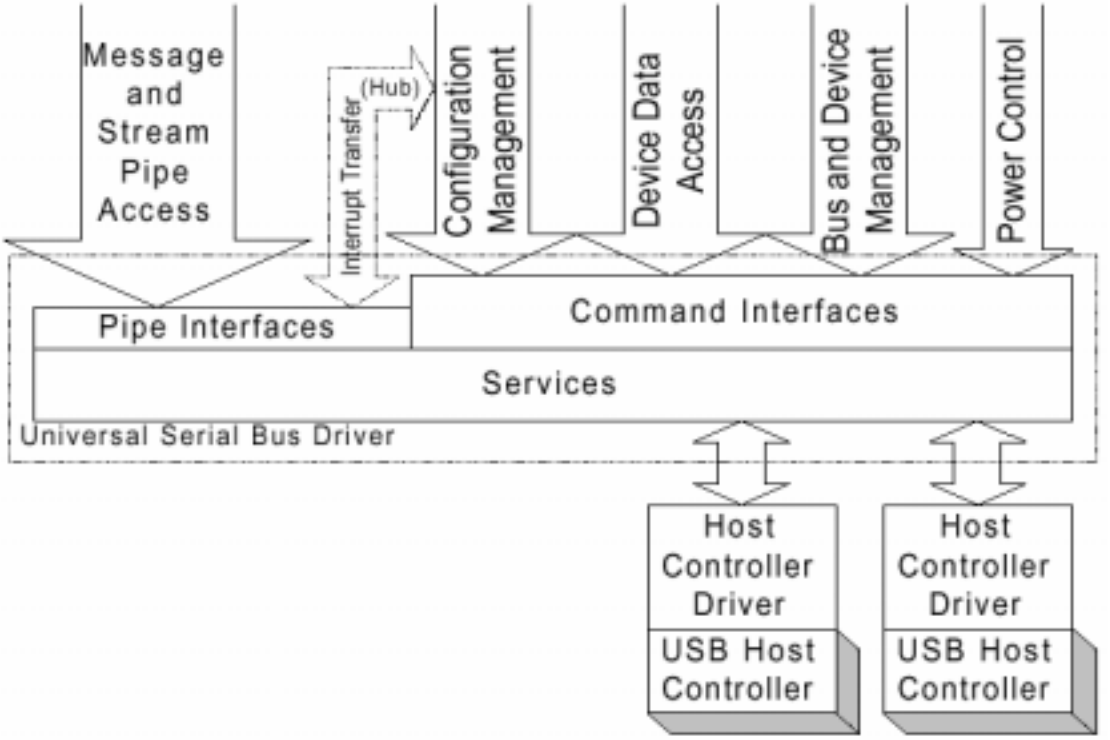
5-2 راه انداز USB

راه انداز USB مجموعه مکانیزم هایی را فراهم می کند که اجزاء سیستم عامل از قبیل راه انداز ابزارها به منظور دسترسی به ابزارهای USB از آنها استفاده می کنیم. بدین صورد دسترسی به ابزار USB تنها از طریق راه انداز USB میسر می باشد. پیاده سازی های راه انداز USB مختص هر سیستم عامل با بررسی راه انداز USB سرویس گیرنده های راه انداز USB دستورات را به ابزار ها هدایت کرده و جریان داده ها را به یا از سمت کانال داده ای حرکت می دهد. راه انداز USB دو دسته مکانیزم نرم افزاری به سرویس گیرنده ها ارایه می کند : *مکانیزم های دستورات و مکانیزم های کانال داده ای.*

مکانیزم های دستورات به سرویس گیرنده ها اجازه می دهد که اعمال[80] راه انداز USB را کنترل و یا پیکربندی نموده و همچنین یک ابزار USB را پیکربندی و به طور عام کنترل کند. در حالت خاص مکانیزم های دستوری کلیه دسترسی ها به کانال داده ای پیش فرض ابزار را فراهم می کند.

مکانیزم های کانال داده ای به سرویس گیرنده راه انداز USB اجازه می دهند تا انتقالات داده و کنترل مخصوص به یک ابزار را مدیریت کنند. مکانیزم های کانال داده ای به سرویس گیرنده اجازه نمی دهند که مستقیما" به کانال داده ای پیش فرض ابزار دسترسی پیدا کند. نمودار 7 ساختار کلی راه انداز USB را نمایش می دهد.

[80] Operation



نمودار 7 : ساختار راه انداز USB

مقداردهی اولیه راه انداز USB

مقدار دهی اولیه راه انداز USB وابسته به نوع سیستم عامل می باشد زمانیکه یک USB خاص که توسط یک راه انداز USB مدیریت می شود، مقداردهی اولیه می گردد. اطلاعات مدیریتی مربوط به آن USB نیز ایجاد می شود. برخی از این اطلاعات مدیریتی عبارتند از آدرس پیش فرض و کانال داده ای پیش فرض ابزار می باشد.

هنگامی که یک ابزار به USB متصل می شود به یک آدرس خاص که آدرس پیش فرض نامیده می شود پاسخ می دهد تا زمانیکه آدرس منحصر بفرد مر بوط به آن ابزار توسط شمارنده گذرگاه تخصیص داده شود. به منظور برقراری ارتباط بین سیستم USB و ابزار جدید ، باید زمانیکه یک ابزار متصل می شود، آدرس پیش فرض ابزار و کانال داده ای پیش فرض آن باید به راه انداز Hub ارایه شود. در طول فرایند مقداردهی اولیه ابزار آدرس پیش فرض به آدرس منحصر بفرد ابزار تغییر می یابد.

استفاده از کانال داده ای راه انداز USB

کانال های داده ای در واقع روش هایی هستند که از طریق آنها یک ابزار با یک نرم افزار میزبان مرتبط می شود. هر کانال داده ای دقیقا" متعلق به یک المان (موجودیت) نرم افزاری در میزبان می باشد. گرچه مفاهیم پایه کانال ها یکسان است یعنی مستقل از مالک کانال، اما برخی تفاوت ها در قابلیت های دو گروه از کانال ها به شرح زیر، برای سرویس گیرنده راه انداز USB وجود دارد.

- کانال های پیش فرض که صاحب آنها راه انداز USB بوده و توسط آن نیز مدیریت می شوند
- تمام انواع دیگر کانال ها که توسط سرویس گیرنده های راه انداز USB مدیریت می شوند و مالک آنها نیز همین سرویس گیرنده ها می باشند.

کانال های پیش فرض هیچ گاه مستقیما" توسط سرویس گیرنده ها در دسترس قرار نمی گیرند. گرچه از این کانال ها اغلب جهت اجرای بخشی از درخواست های سرویس گیرنده در قالب مکانیزم های دستوری استفاده می شود.

کانالهای پیش فرض

راه انداز USB مسؤل تخصیص و مدیریت بافر های لازم به منظور پشتیبانی انتقالات روی کانال پیش فرض که از دید سرویس گیرنده مخفی است، می باشد از قبیل تنظیم آدرس ابزار. برای برخی انتقالات روی کانالی که از دید سرویس گیرنده مخفی نمی باشد از قبیل ارسال دستورات کلاس و فروشنده و یا خواندن توصیف گر ابزار ، سرویس گیرنده می باید خود، بافرهای مورد نیاز را فراهم کند.

کانالهای سرویس گیرنده

هر کانالی که توسط راه انداز USB مالکیت و مدیریت نشود ، توسط سرویس گیرنده راه انداز USB مالکیت و مدیریت می شود. از دیدگاه راه انداز USB تنها یک سرویس گیرنده مالک کانال می باشد. در واقع گروه هماهنگی از سرویس گیرنده ها ، می توانند یک کانال را مدیریت کنند البته در صورتیکه همگی به گونه ای هماهنگ به عنوان یک مالک کانال عمل کنند.

سرویس گیرنده مسؤل فراهم نمودن مقدار بافر مورد نیاز جهت ارایه سرویس با نرخ انتقال انبوه داده کانال داده ای ، در یک بازه زمانی سرویس دهی ، قابل حصول توسط سرویس گیرنده، می باشد. سایر نیازمندیهای بافرینگ توسط سیستم USB مشخص می شود.

قابلیت سرویس دهی راه انداز USB

راه انداز USB سرویس هایی در موارد زیر فراهم می کند :

- پیکربندی از طریق مکانیزم های دستوری
- سرویس های انتقال از طریق مکانیزم های دستوری و کانال داده ای
- آگاه سازی از وقوع رخداد ها
- گزارش دهی از وضعیت و نیز جبران خطا

نتیجه گیری

همانگونه که می دانیم با گذشت زمان نیازهای بشر افزایش می یابد که این افزایش ناشی از آرزو برای رسیدن به آسایش و رفاه بیشتر و از طرف دیگر پیشرفت روز افزون علم و تکنولوژی می باشد. در صنعت کامپیوتر با **بوجود آمدن ابزارهای سریعتر، هوشمندتر** و از سوی دیگر **افزایش ابزارهای جانبی** ، نیاز برای یک پورت که با **هزینه کم** بتوان این ابزارها را با **سرعت بالا (انتقال اطلاعات)** به کامپیوتر وصل و از آنها بطور بهینه استفاده کرد، روز به روز افزایش یافت. پورت USB پاسخی برای نیازهای فوق می باشد. از ویژگیهای این پورت می توان به مواردی مثل آسان بودن استفاده از آن ، محدوده وسیع کاربردها ، پهنای باند بالا ، قابلیت انعطاف ، پیاده سازی کم هزینه و مهمتر از همه معماری قابل ارتقاء می توان اشاره کرد. چرا که معماری آن به گونه ای طراحی شده است که قابلیت پشتیبانی چند USB HC در یک سیستم را دارد.

مراجع

1. "USB Specification" [PDF], chapter 1,3,4,10 , Sep. 23 1998, Revision 1.1 .
2. "PC Intern", Michael Tischer, 5th Edition, chapter 7, 8, 27.
3. "Upgrading and Repairing PCs", Edward Griffin, Eight Edition, 1998, chapter 11.