

Interfaz USB genérica para comunicación con dispositivos electrónicos

USB4all

Tutores:

- Gonzalo Tejera
- Alexander Sklar

Integrantes:

- Andrés Aguirre
- Carlos Grossy
- Rafael Fernández

Agenda

- Presentación del proyecto
 - Tecnología USB
 - Soluciones de conectividad USB
 - Arquitectura
 - Características y aportes de la solución
 - Caso de uso: USB4bot
-

¿Que es el proyecto?

- Una respuesta a la necesidad de comunicar de forma **sencilla y genérica** dispositivos electrónicos no necesariamente pensados para interactuar con un PC.
- La solución se basa en tres puntos:
 - Un componente de hardware.
 - Un medio de comunicación (USB).
 - Una arquitectura (software y firmware).

Motivación

- ❑ Utilizar una PC para comunicarse con dispositivos electrónicos, logrando:
 - Aumentar la potencialidad de los dispositivos.
 - Aprovechar las capacidades de procesamiento, y almacenamiento del PC.
 - Aumentar la Interacción con el mundo físico.
 - Simplificar el manejo de los dispositivos.
- ❑ Uso de microcontroladores como parte de la solución.

Por Que USB?

- Desuso de puertos paralelos, seriales.
 - No hay que abrir la PC.
 - Versátil.
 - Disponibilidad.
 - Tecnología actual.
-

Descripción del Proyecto

■ Objetivo

- ❑ Construcción de hardware y software necesarios para facilitar la comunicación con dispositivos electrónicos por medio del USB.
 - ❑ Ocultar la complejidad de la tecnología USB.
 - ❑ Arquitectura modularizada y extensible.
 - Firmware, API, protocolo de comunicación, Drivers.
 - Bibliotecas de alto nivel para distintos dispositivos
 - ❑ Soporte para Linux y Windows.
-

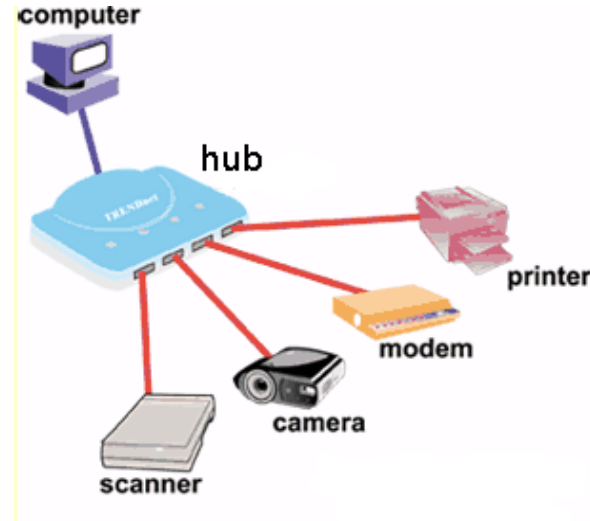
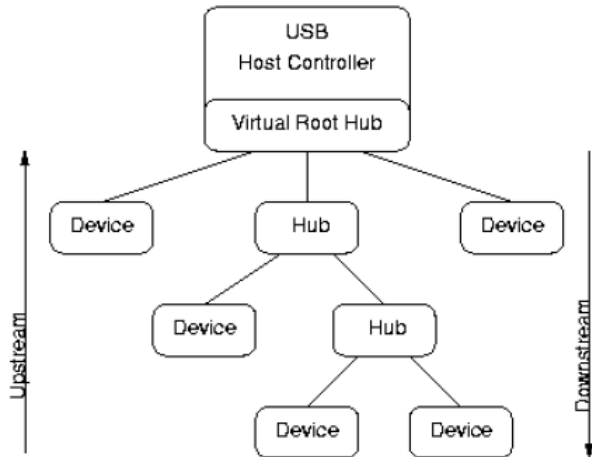
Tecnología USB

Características

- Varias velocidades
 - Low speed – 1.5 Mbit/sec
 - Full speed – 12 Mbit/sec
 - High speed – 480 Mbit/sec
- Detección y configuración automática de los periféricos.
- El bus es arbitrado por el Host (PC)

Como se comunican los dispositivos?

- Frames
- Bus centrado en el host
 - El maestro (host) inicia todas las transferencias
 - Los esclavos (dispositivos) responden a los pedidos



Como se comunican los dispositivos?

- Configuraciones
- Interfaces
 - Representan diferentes funcionalidades



interface 1



interface 3

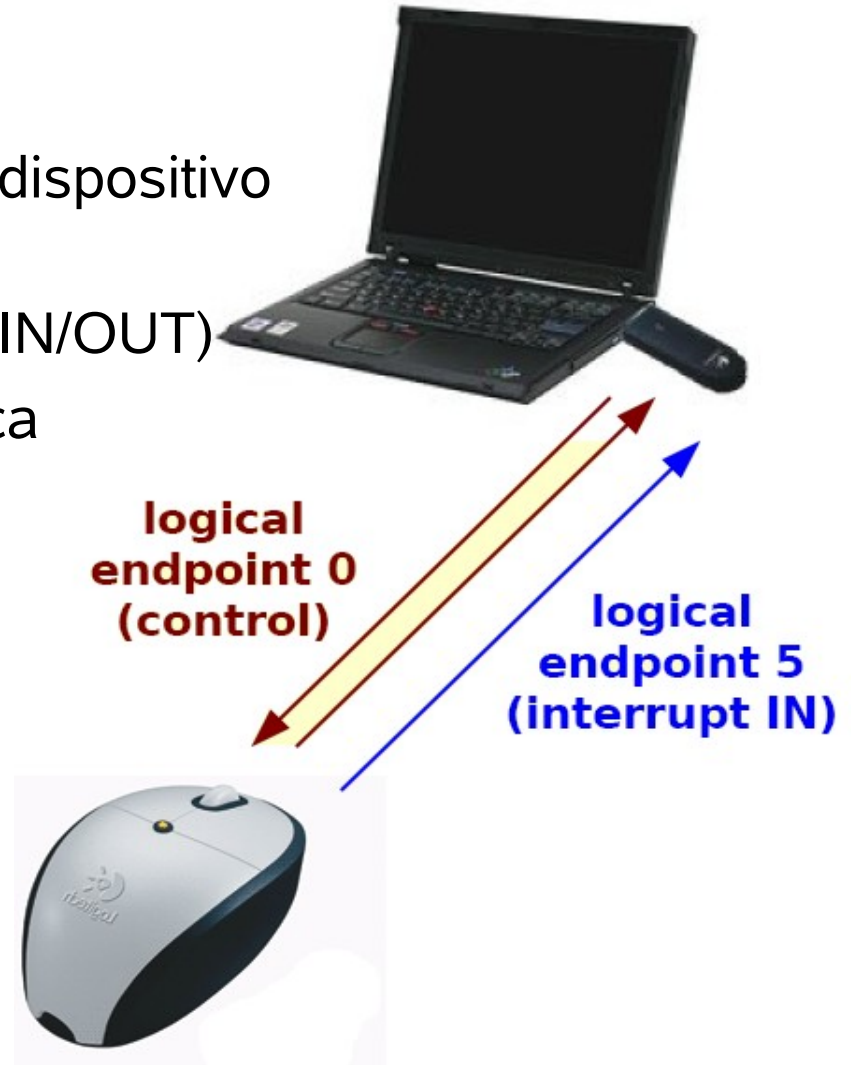


interface 2



Endpoints

- Parte única e identificable de un dispositivo USB
- Poseen una sentido específico (IN/OUT)
- Tienen una dirección (física) única
- Tipos:
 - Control
 - Bulk
 - Interrupt
 - Isochronous



Control Endpoints

- Endpoint 0
 - Usado para tareas de configuración
 - Setear dirección de un dispositivo
 - Obtener los descriptores de un dispositivo (Enumeración)
 - Desconexión
 - Siempre existe y es bidireccional (sentido IN y OUT)
 - 10% de los frames reservado
-

Bulk Endpoints

- Transferencias esporádicas
- No tiene ninguna prioridad garantizada en el scheduling
- Reintentos automaticos



Interrupt Endpoints

- Transferencias periódicas
- Tienen prioridad garantizada en el scheduling
 - Transfieren cada N frames
- Reintentos automáticos
- El host interroga (poll) al c
 - No



Isochronous Endpoints

- Transferencias periodicas
- Tienen prioridad garantizada en el scheduling
- No tiene Reintentos automaticos



Soluciones de conectividad USB

Soluciones de conectividad USB

- Opciones
 - Transceivers USB
 - Conversores USB a serial o paralelo
 - Controladores de periféricos
 - Externos
 - Embebido en un microcontrolador
-

Controladores de periféricos embebido en un microcontrolador

- Se incorpora dentro del mismo microcontrolador el hardware necesario para conectarse directamente al USB.
 - Las comunicaciones se manejan a nivel de endpoints.
 - “Dependencia” de una arquitectura
- Ejemplos:
 - TSUB3210 (Texas Instruments)
 - PIC18F4550 (Microchip)
 - AT90USB1287 (Atmel)

Tabla de comparación entre microcontroladores

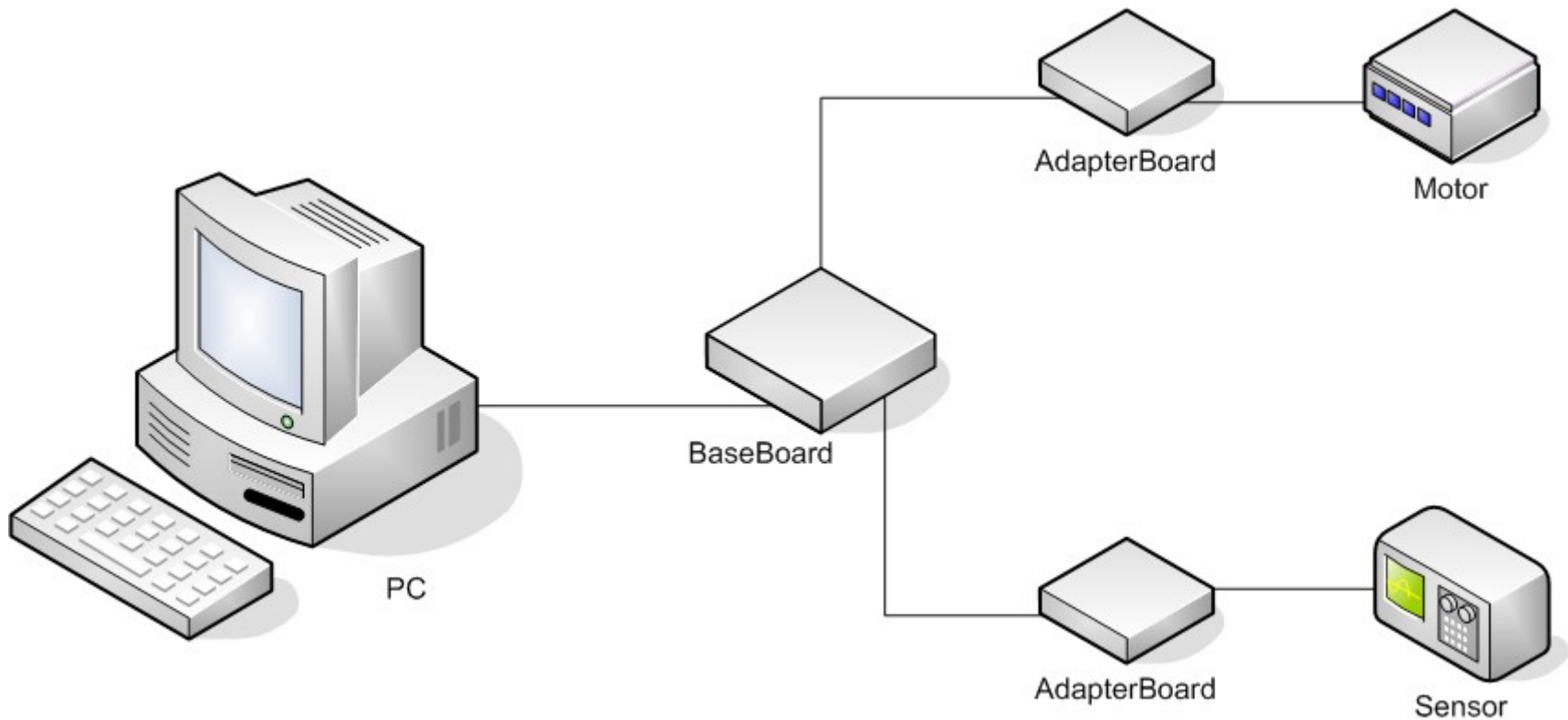
	TUSB3210	PIC18F4550	AT90USB1287
Package	TQFP 64	TQFP 44, QFN 44, DIP 40	TQFP 64, QFN 64
Memoria de programa	*6K ROM, 8K RAM	32Kb Flash autoprogramable	128Kb Flash
Memoria datos	768 bytes	2 Kb	8 Kb (hasta 64 KB externos)
USB 2.0 (full y low speed) Endpoints	3 IN, 3 OUT. Transferencias interrupt y	Hasta 32 soporta todas las transferencias	6 endpoints, soporta todas las transferencias
Eval. de periféricos	1	2	3
Documentación	Poca, algunas notas de aplicación.	Mucha, recursos en la web, muchas notas de aplicación, framework USB	Poca, Framework USB, algunas notas de aplicación.
Entornos de desarrollo y compiladores	En general los de 8052, de 3eras partes, algunos gratuitos.	MPLAB, 3ras partes, varios compiladores	AVR Studio 4, 3ras partes

Elección del microcontrolador

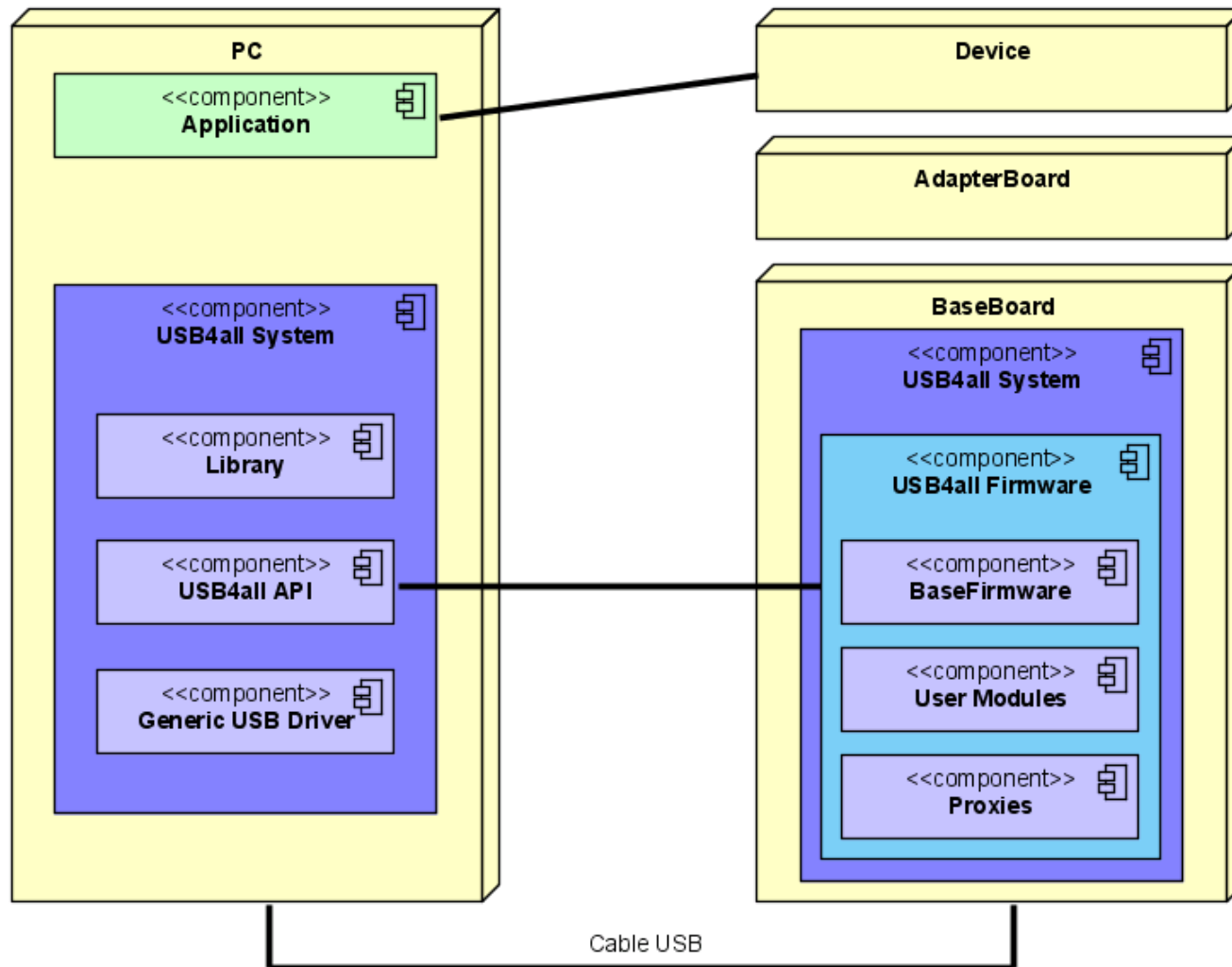
- La elección queda entre el PIC18F4550 y el AT90USB1287 y se tienen en cuenta los siguientes criterios:
 - Aspectos Técnicos
 - El AT90USB1287 en general es superior al PIC18F4550.
 - Documentación
 - Mayor documentación y notas de aplicación disponible del PIC18F4550.
 - Infraestructura y Conocimientos Previos
 - Experiencia previa (taller de firmware)
 - Conocimiento de arquitectura y herramientas de desarrollo.
 - Hardware de programación/debugging disponible.
 - Kit de desarrollo PICDEM FS USB.
 - Disponibilidad
 - PIC18F4550 disponible en plaza.
 - PIC18F4550 disponible en package DIP40.
- ***Se tomó la decisión de usar el PIC18F4550 para la implementación en el proyecto de grado.***

Arquitectura

Panorama General



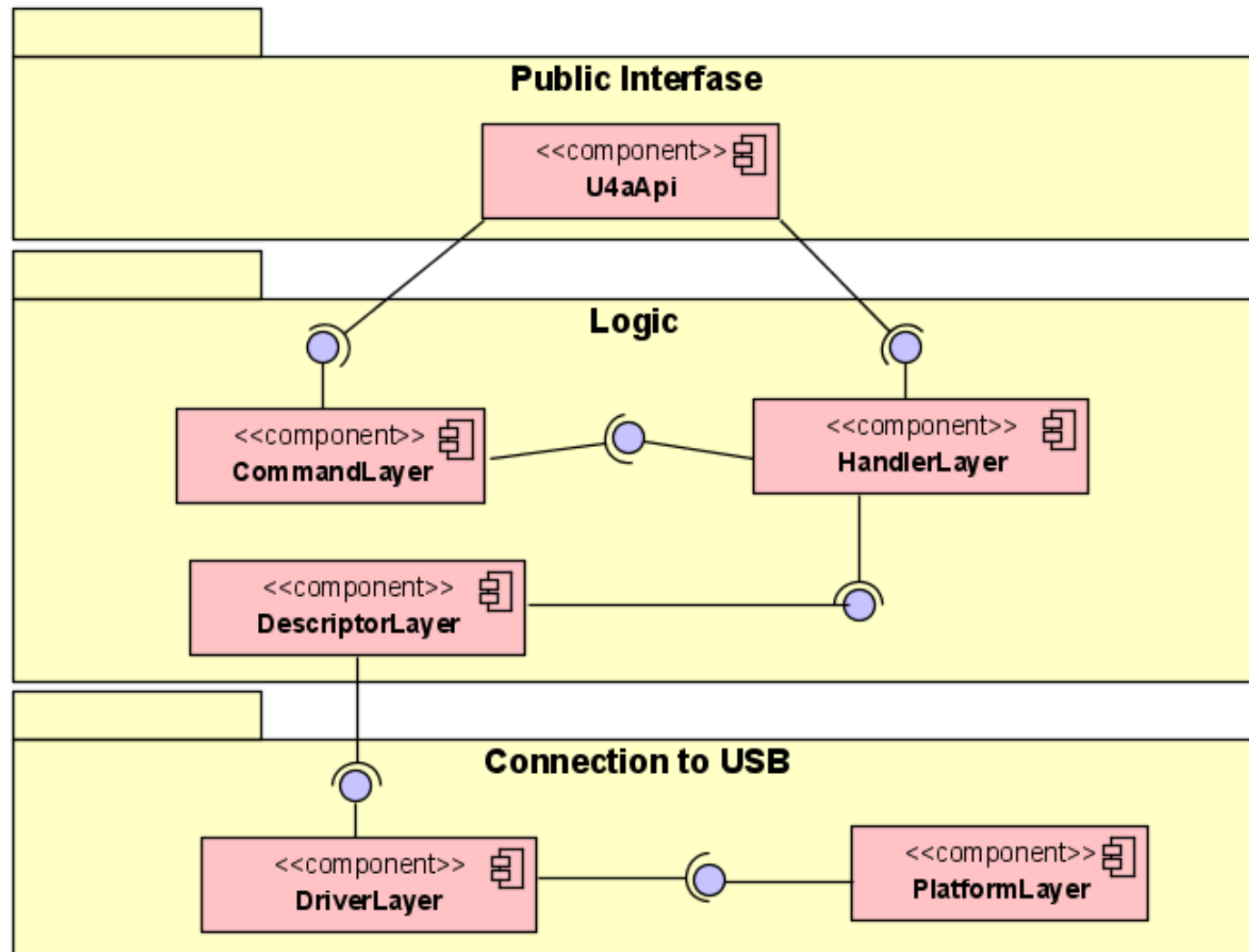
Arquitectura de Software



Biblioteca orientada a objetos

- Encapsula los principales elementos del sistema
 - Implementada en JAVA
 - Fácil de usar
 - Extensible
 - Cuenta con ejemplos
-

USB4all API



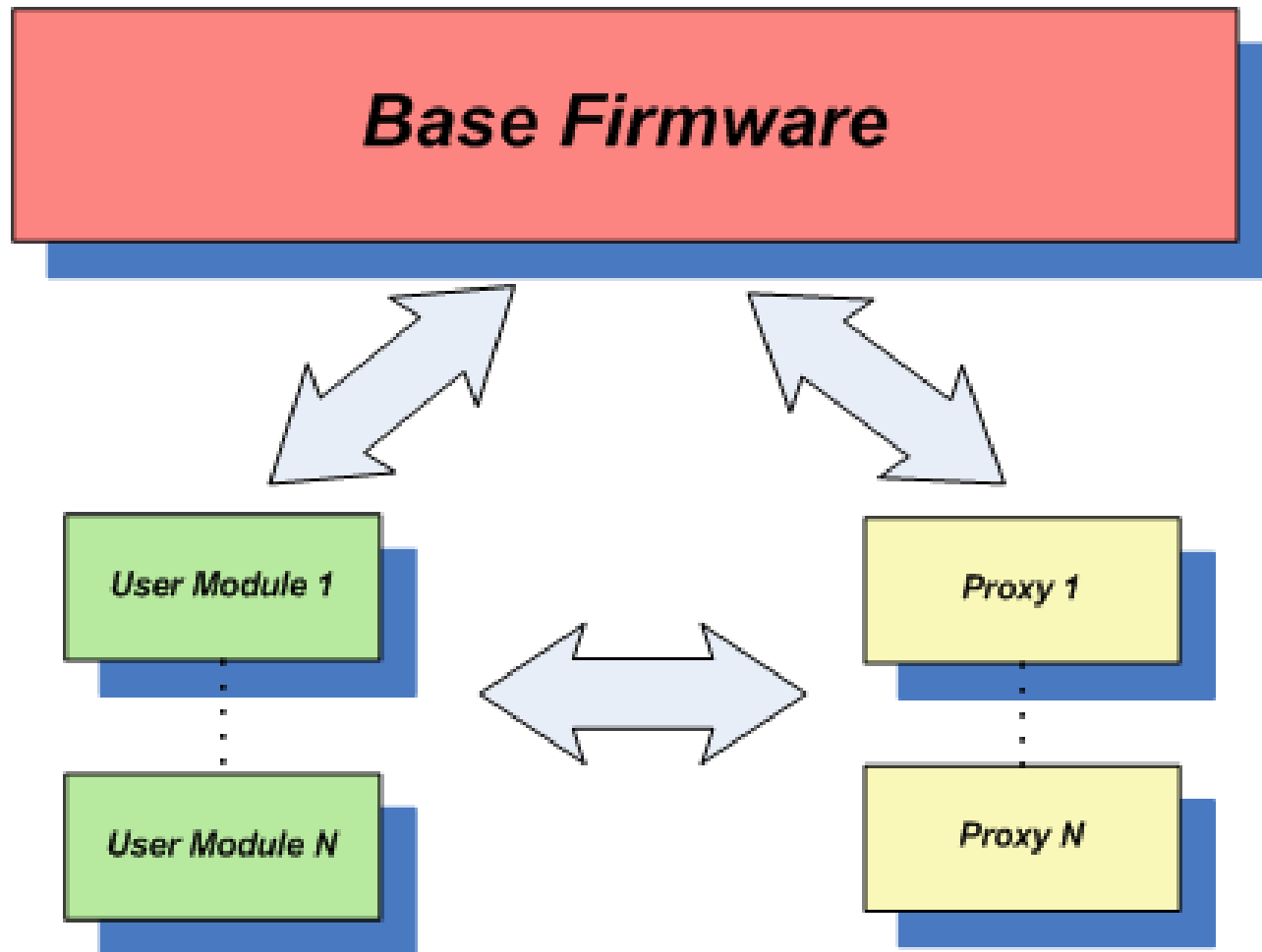
Drivers utilizados

- Microchip USB Generic Driver
 - ❑ Distribución gratuita.
 - ❑ Solo plataforma Windows.
 - LibUSB y LibUSBWin32
 - ❑ Proyecto Open Source.
 - ❑ Plataformas Windows y Linux entre otras.
 - ❑ Solo tipo de transferencias Bulk y Control.
 - Driver Propio (modo Kernel).
 - ❑ Plataforma Linux
-

BaseBoard



USB4all Firmware



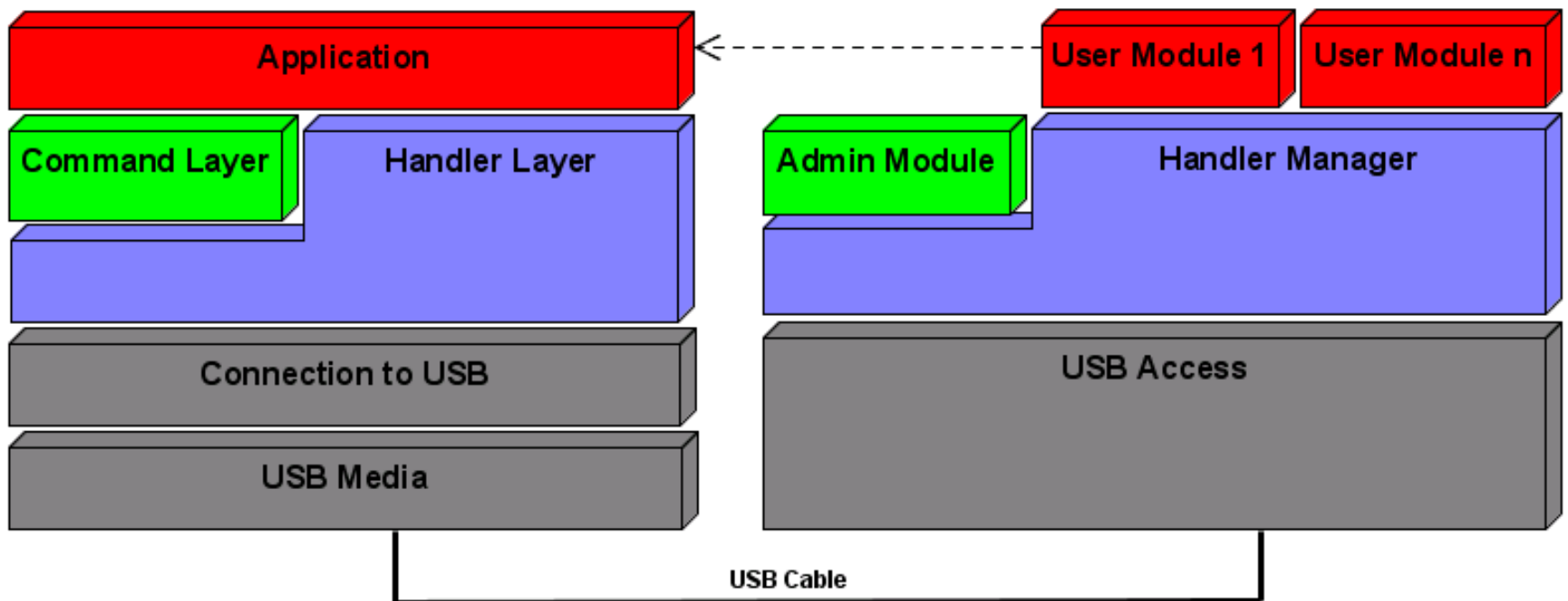
Base Firmware

- Responsable de toda la interacción con el PC mediante USB
- Brinda servicios para que puedan correr varios user modules de forma simultanea.

User Modules

- Encapsula la lógica específica para el manejo de un determinado dispositivo o conjunto de dispositivos.
 - Interactúan con el hardware
 - Motores
 - Sensores
 - Etc
 - Extienden las funcionalidades del firmware
 - Comportamiento similar a un plugin
-

Comunicación virtual



Características y aportes de la solución

Características

- Integral
 - Dispositivo Genérico
 - Protocolo abierto y user modules inteligentes
 - Constructivo
 - Multi-Instancia de baseboards.
-

Características cont

- Multi-plataforma
 - Multi-lenguaje de programación.
 - Orientación a objetos
 - No uso de conversores USB-Serial
 - Costos económicos.
 - Open Source Software y Hardware
-

Aportes de la solución

- Extensión del dominio de acción del PC y dispositivos.
 - Desarrollo guiado y amigable.
 - Perfiles de usuarios.
 - Fomenta la colaboración entre usuarios.
 - Apoyo a tiempo real.
 - Driver USB genérico para Linux.
 - Prototipos rápidos.
-

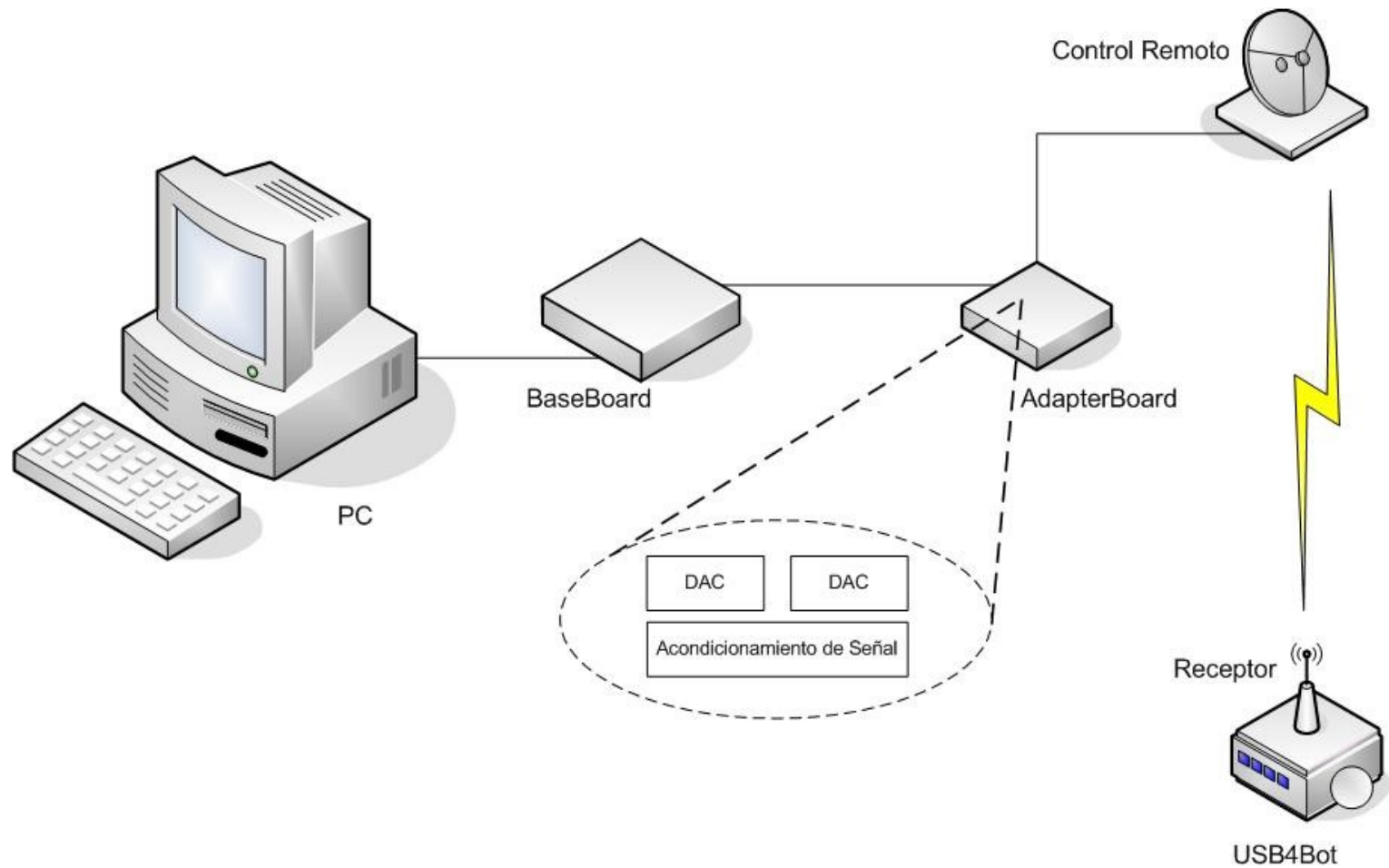
Caso de Uso

USB4bot – Robot de Sumo

USB4bot

- Reutilización de algoritmo de manejo de categoría SUMBOT
- Conexión PC <-> radio control
- Ejemplo de prototipo rápido
 - Construcción de adapterboard (2 DACs y acondicionamiento)
 - Leve modificación del radio control
 - Creación de user module
 - Cambio mínimo en algoritmo para SUMBOT

Funcionamiento



USB4BOT a futuro...

- Uso de tecnología Wireless USB



Preguntas

