

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**
 - **Soluciones de conectividad USB**
 - **Proyectos relacionados**
 - **Arquitectura**
 - **Conclusiones**
 - **Trabajos a futuro**
 - **Prototipos**
-

¿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).

¿Por Que USB?

- Desuso de puertos paralelos, seriales.
 - No hay que abrir la PC.
 - Versátil.
 - Disponibilidad.
 - Varias velocidades (Low, Full y High)
 - Varios tipos de transferencias (Control, Interrupt, Bulk, Isochronous).
 - Tecnología actual.
-

Dificultades de usar USB

- Lógica más compleja en el dispositivo .
 - Mayor nivel de conocimientos y dificultad para el desarrollo de dispositivos.
 - No se puede usar directamente.
 - Mayor precio.
-

Motivación

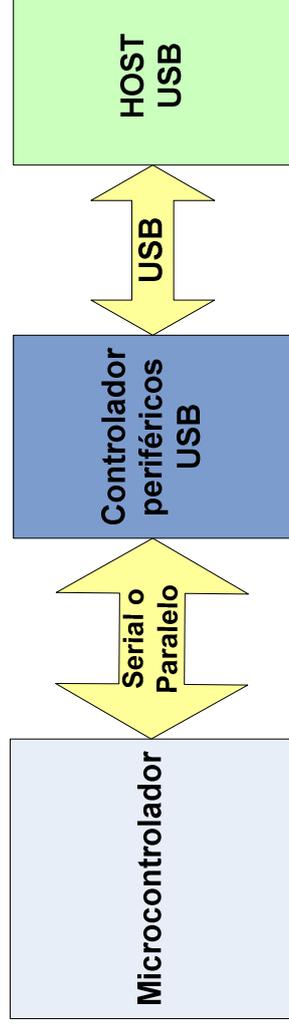
- ❑ Utilizar una PC para comunicarse con dispositivos electrónicos, logrando:
 - Aumentar la potencialidad de los dispositivos.
 - Aprovechar las capacidades de procesamiento, y almacenamiento del PC.
 - Aumentar la Interacción con el mundo físico.
 - Simplificar el manejo de los dispositivos.
 - Disminuir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

Objetivos

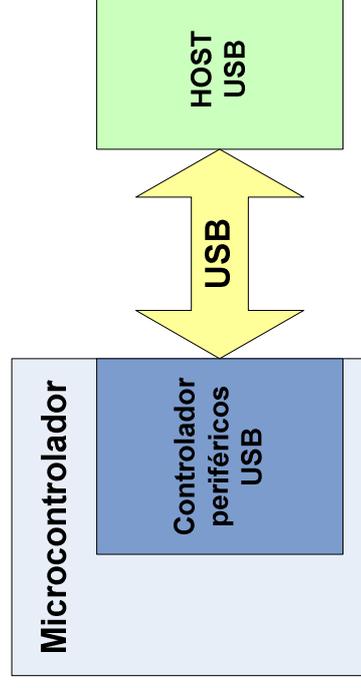
- Construcción de hardware y software necesarios para facilitar la comunicación con dispositivos electrónicos por medio del USB.
 - Ocultar la complejidad de la tecnología USB.
 - Arquitectura modularizada y extensible.
 - Firmware, API, protocolo de comunicación, Drivers.
 - Bibliotecas de alto nivel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

Soluciones de conectividad USB

- Se investigaron en profundidad:
 - Controladores de periféricos USB



■ Externos



■ Embebido en un microcontrolador

Elección del microcontrolador

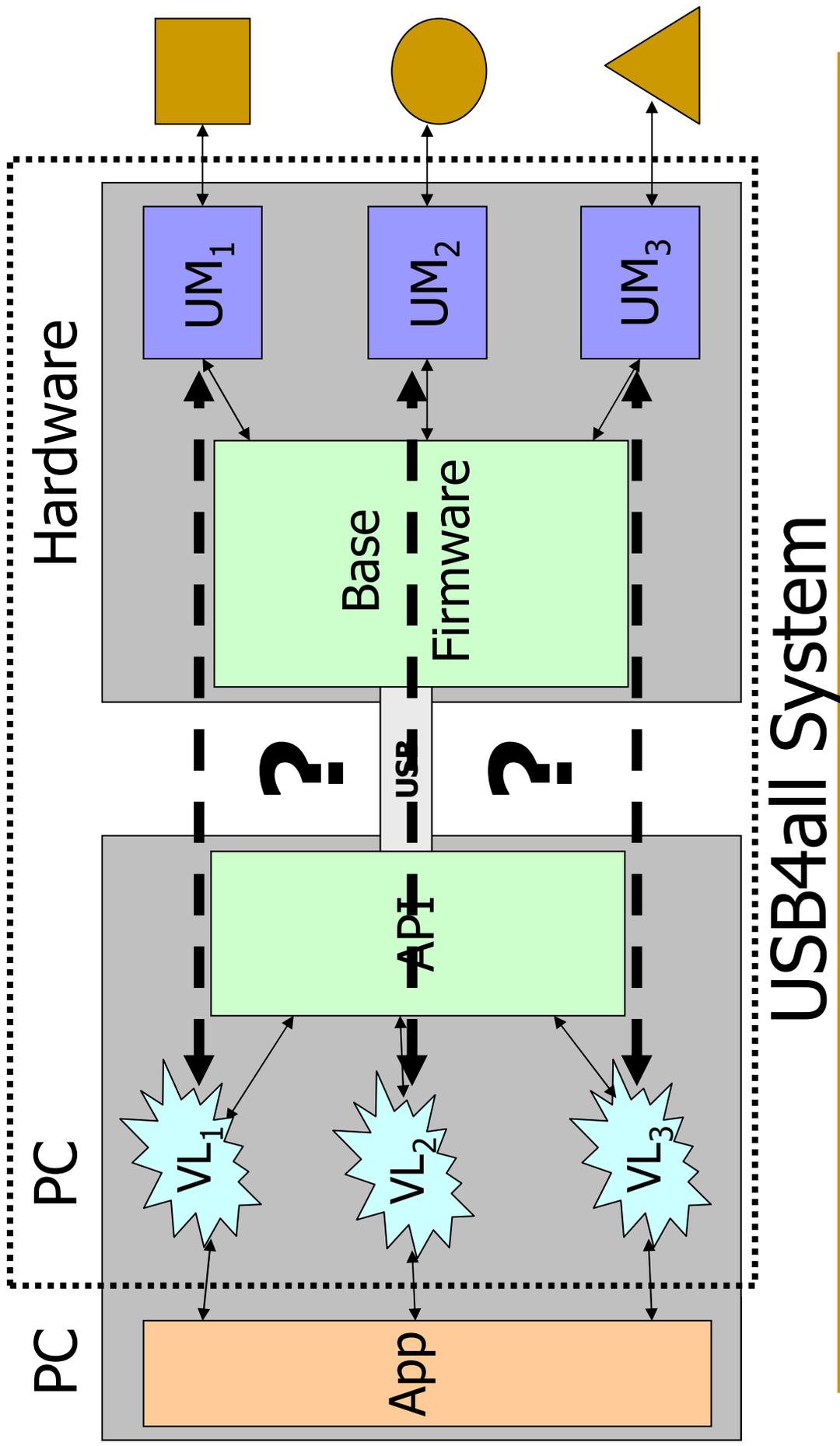
- La elección final se realiza entre el PIC18F4550 y el AT90USB1287
 - Aspectos Técnicos
 - El AT90USB1287 en general es superior al PIC18F4550.
 - Documentación
 - Mayor documentación y notas de aplicación disponible del PIC18F4550.
 - Infraestructura y Conocimientos Previos
 - Experiencia previa (taller 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 y en package DIP40.
 - ***Se decide utilizar el PIC18F4550 para la implementación en el proyecto de grado.***
-

Arquitectura

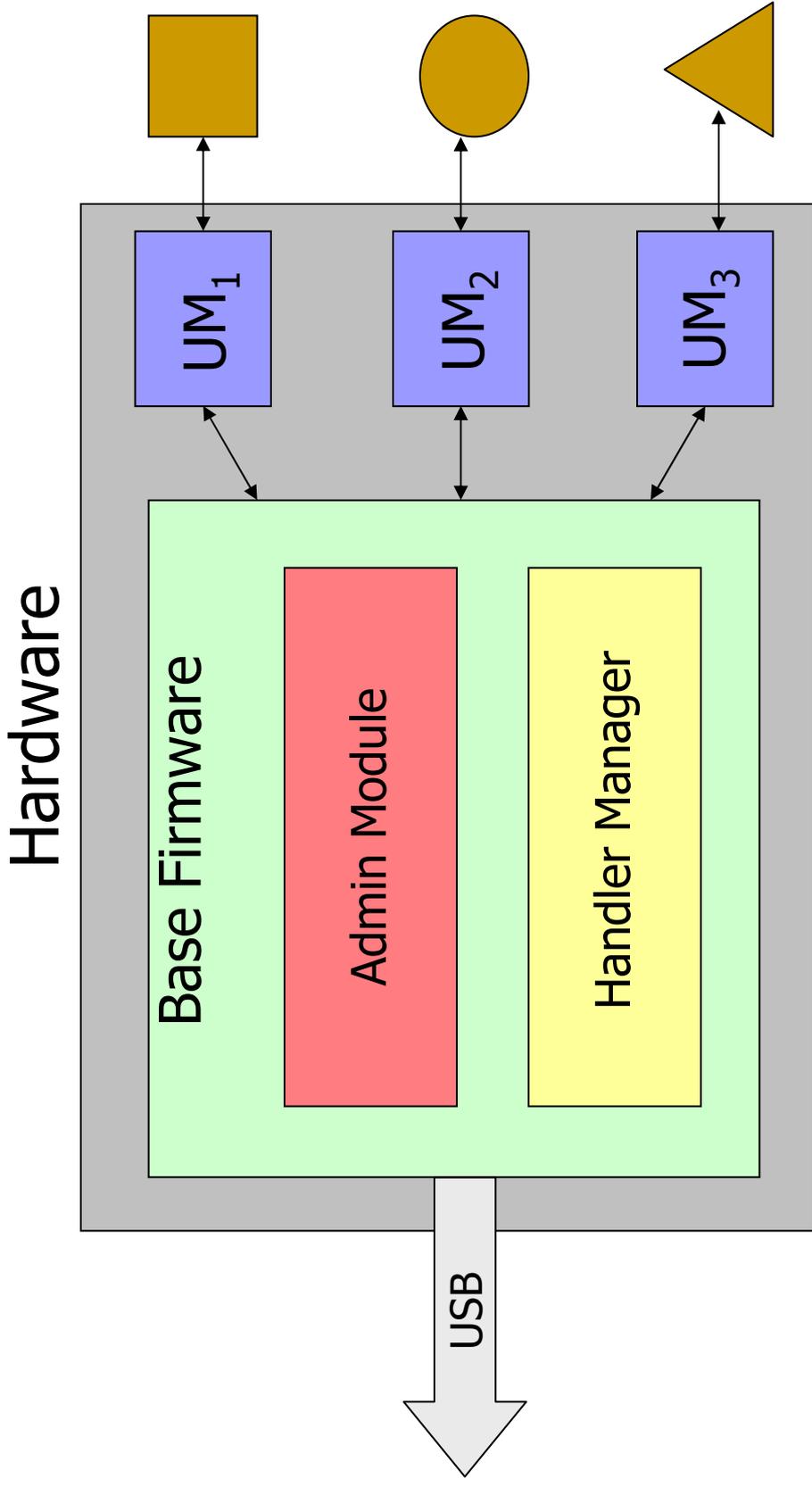
PPT's de la Arquitectura (punteo)

- Primero diagrama simple de funcionamiento con animación. (Diagrama en papel)
- Diagrama con elementos simples del ambiente basboard.
- Diagrama con elementos simples del PC.
- Detalle general del manejo de eventos (patrón observer de los dynamic y proxies, recursos compartidos).
- Diagrama general de acceso al USB de la API.
- Partes físicas.
- Baseboard, elementos de hardware, características generales (serial number, tipos de trans., etc.) y configuración.
- Diagrama que muestre los distintos protocolos, comunicación virtual (las partes).
- Caso de multiplaca o escenario de uso.

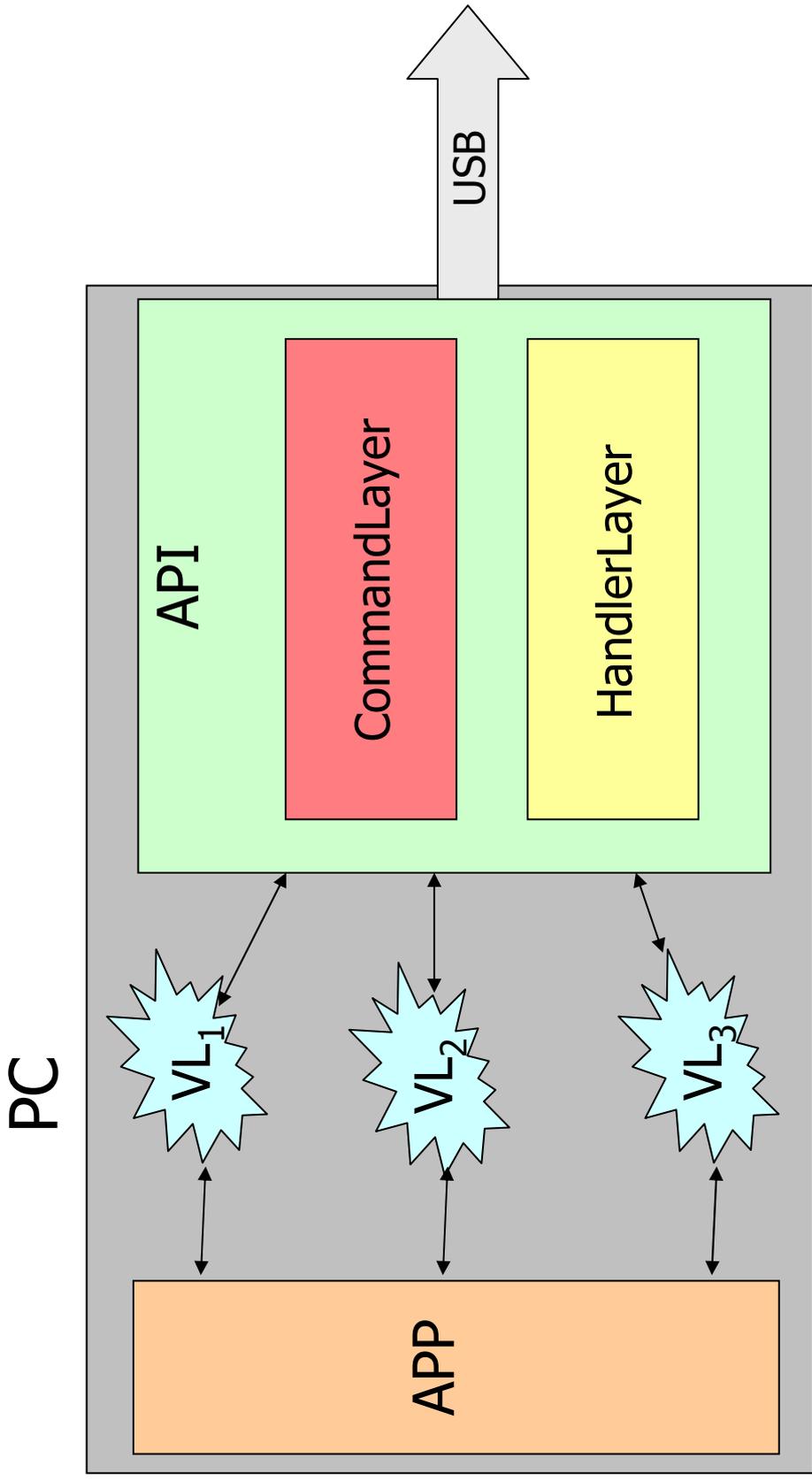
La idea general ...



El hardware bajo la lupa ...

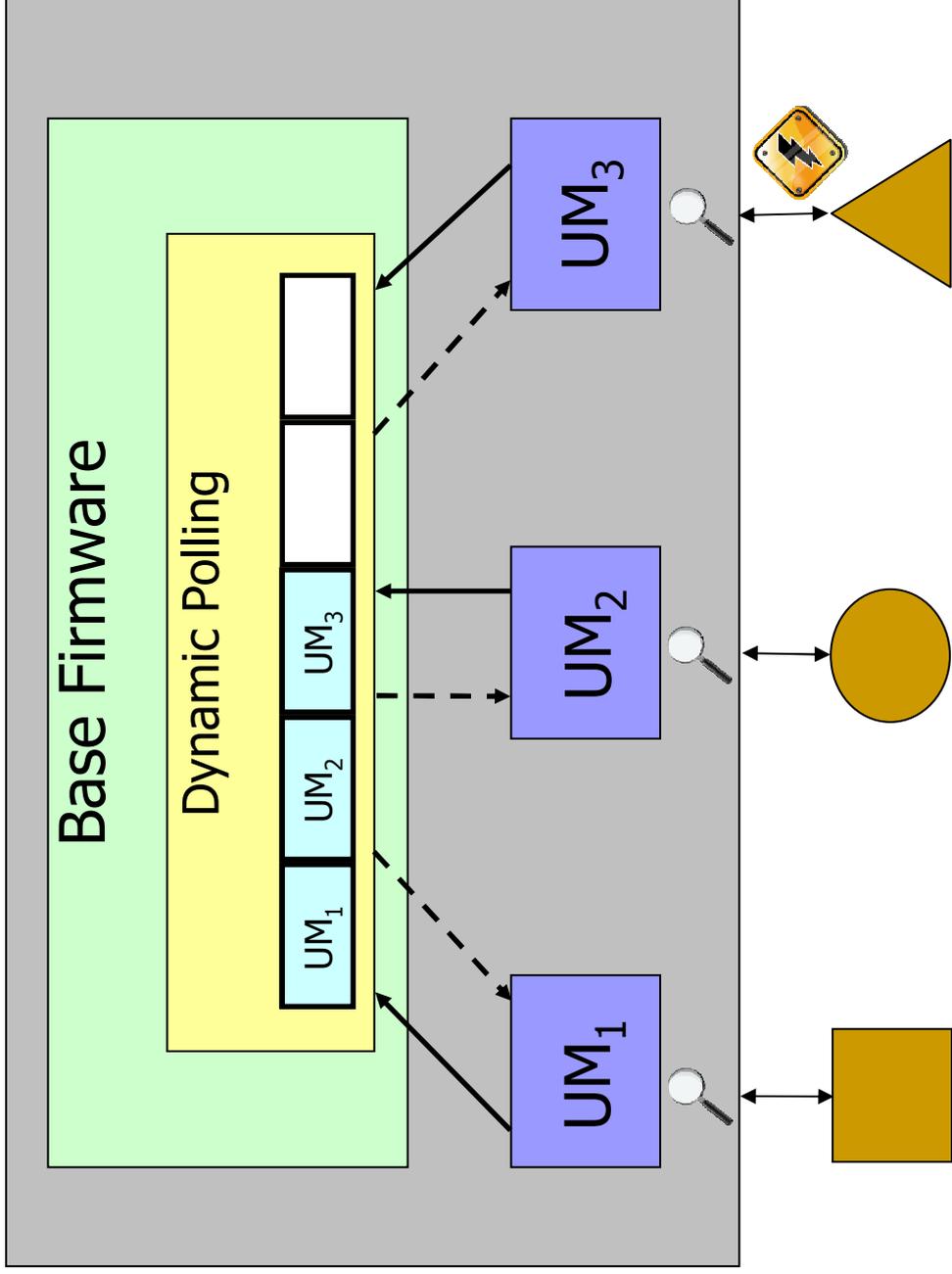


El PC bajo la lupa ...

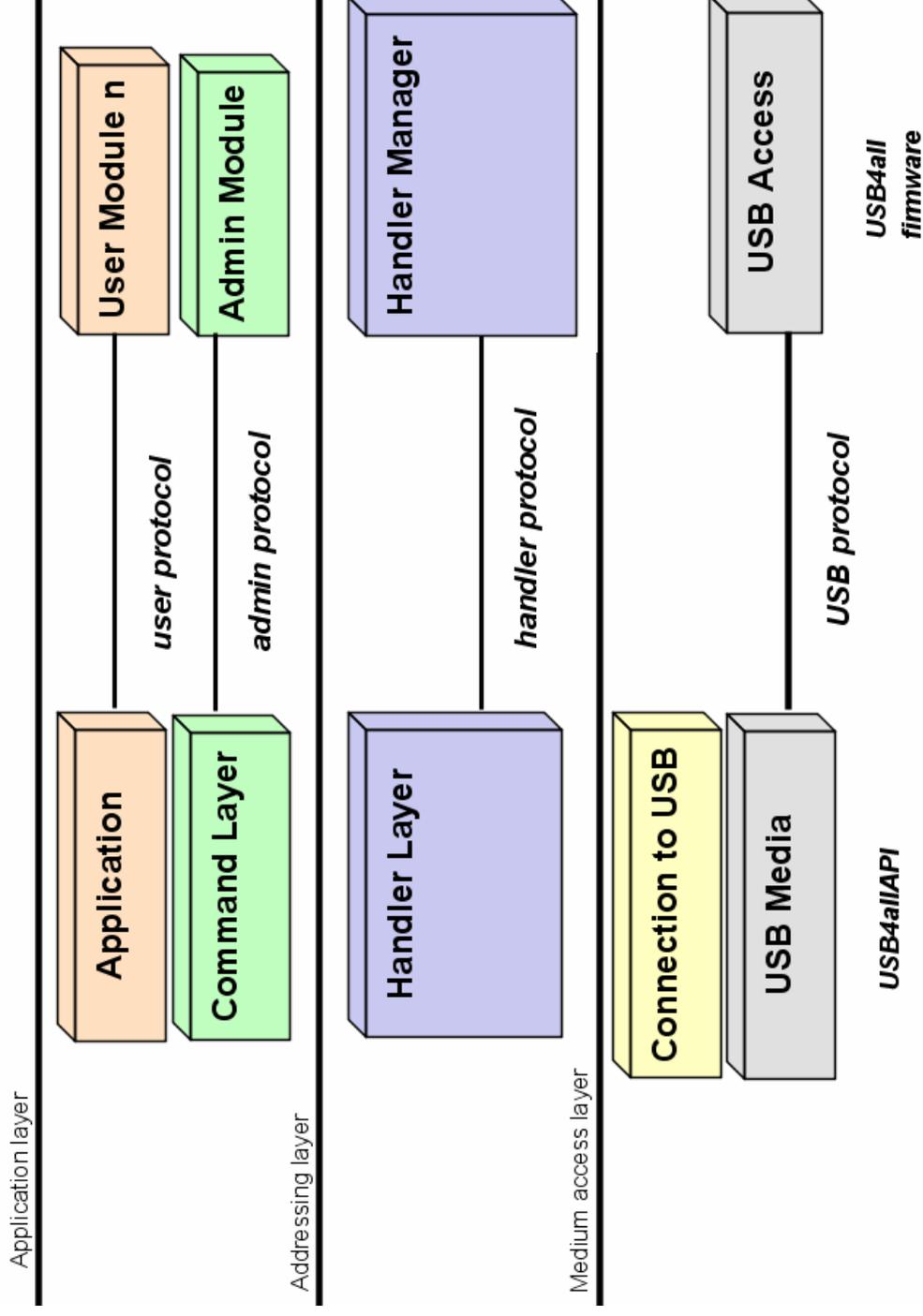


Manejo de los eventos

Hardware



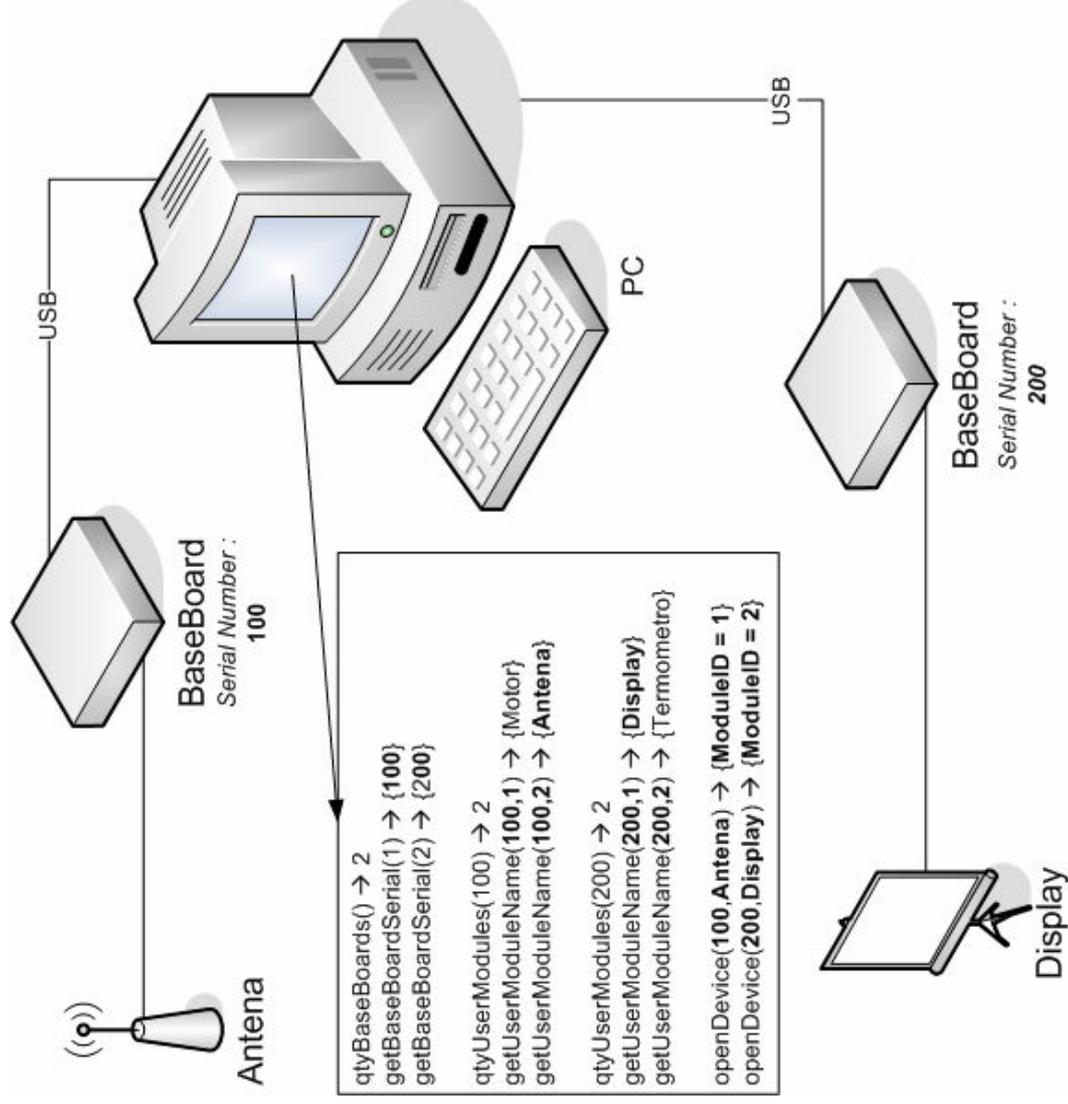
Protocolo y comunicación



BaseBoard



Escenario de uso



Características y aportes de la solución

Artefactos Construidos (punteo)

- Library
- Driver Linux
- Metodología de programación.
- Resumen de artefactos construidos.
 - Baseboard
 - API
 - Firmware
 - Library
 - Driver Linux
 - Prototipos
 - Utilitarios

Características

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

Características Cont.

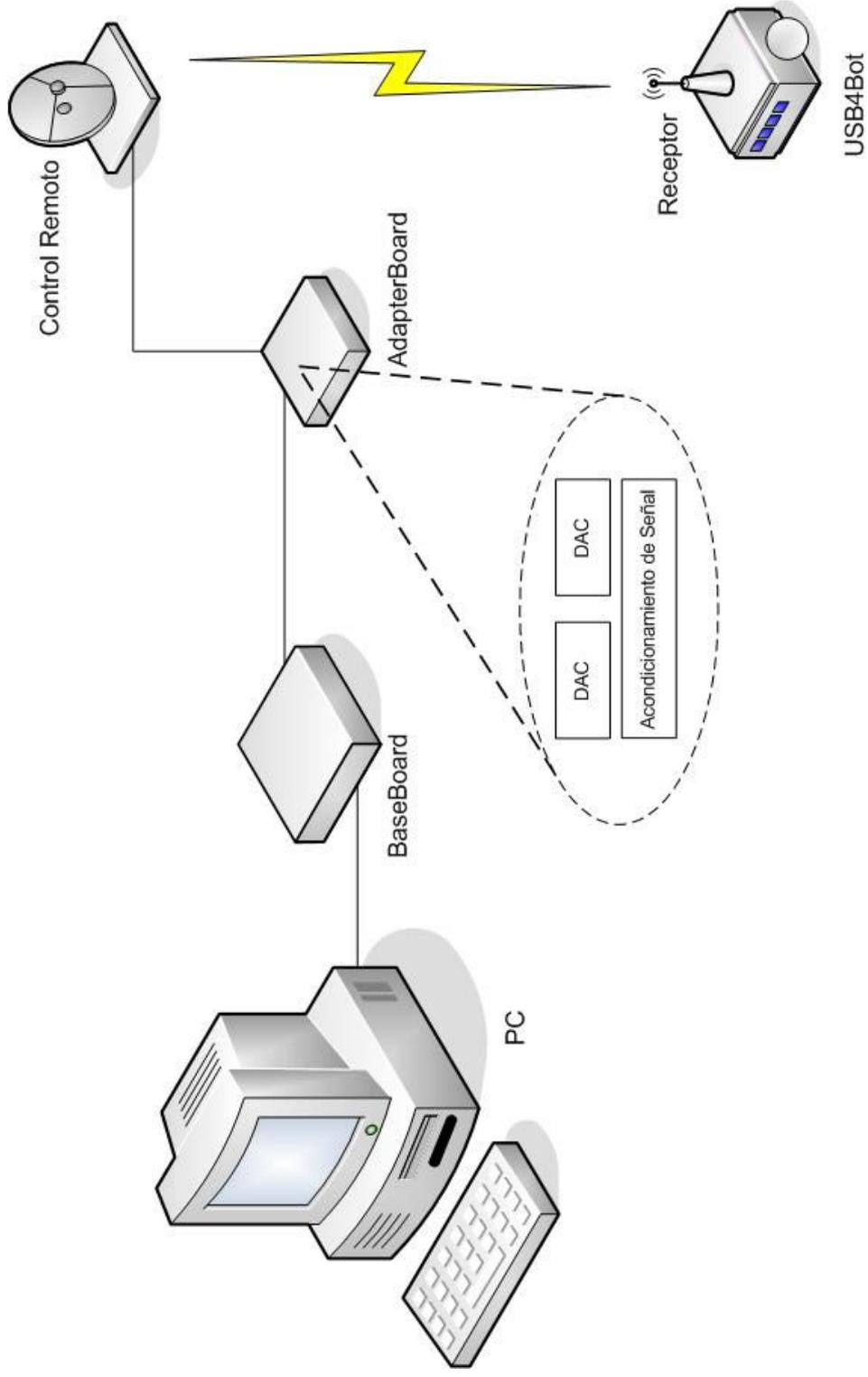
- Multi-plataforma
 - Multi-lenguaje de programación.
 - Orientación a objetos
 - No uso de conversores USB-Serial
 - Costos económicos.
 - Open Source Software y Hardware
-

Aportes de la solución

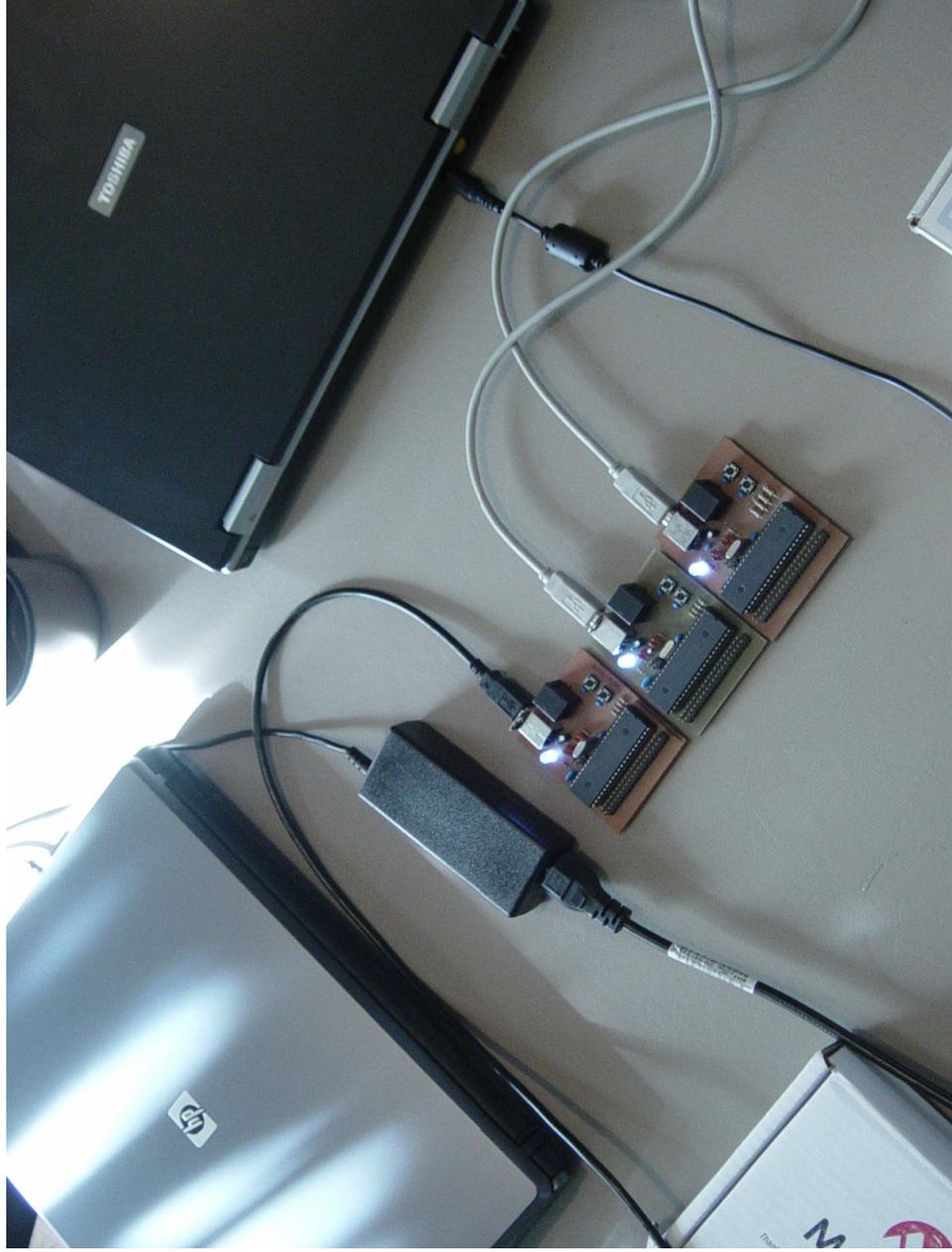
- Extensión del dominio de acción del PC y dispositivos.
 - Desarrollo guiado y amigable.
 - Perfiles de usuarios.
 - Fomenta la colaboración entre usuarios.
 - Apoyo a tiempo real.
 - Driver USB genérico para Linux.
 - Prototipos rápidos.
-

Demostraciones

USB4bot



Preguntas



Biblioteca orientada a objetos

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

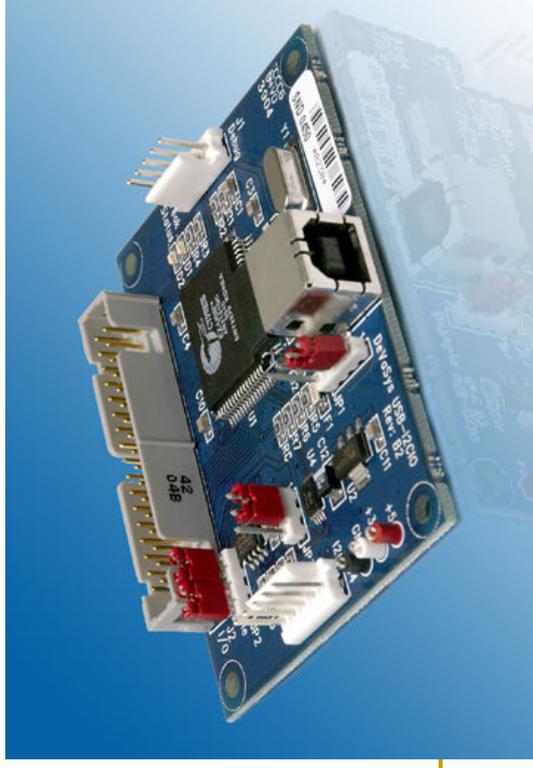
Proyectos Relacionados

Relevamiento

- Algunas de las soluciones relevadas
 - DevaSys
 - Arduino
 - Wiring
 - CUI
 - IO Warrior
 - AID
 - Gainer
 - Elexol
 - ActiveWire
 - DataTranslation
-

Comparación – DevaSys

- Puntos débiles
 - ❑ No utiliza todos los tipos de transferencia USB.
 - ❑ Sólo utiliza la velocidad USB low.
 - ❑ No utiliza todos los recursos de hardware.
 - ❑ Driver sólo para Windows.
 - ❑ API no pensada para comunicación genérica.
 - ❑ Firmware no expandible.



Comparación – DevaSys

- Puntos fuertes
 - Diseño modular del hardware.
 - Permite tener múltiples placas.
 - Documentación aceptable.
 - Firmware actualizable vía USB.

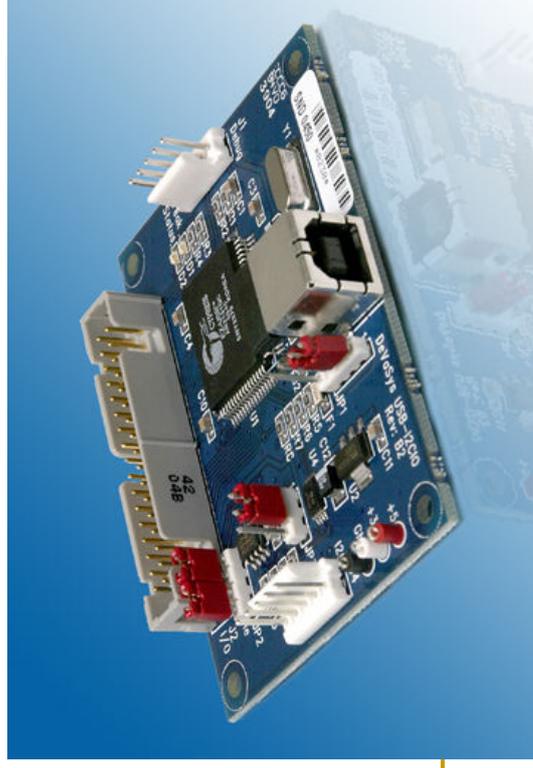


Tabla de comparación entre microcontroladores

	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)	Soporta transferencias interrupt y bulk	Soporta todas las transferencias	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



DIP



TQFP



QFN

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
-

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
-

USB4BOT a futuro...

- Uso de tecnología Wireless USB

