



UNIVERSIDAD
POLITECNICA
DE VALENCIA

Escuela Técnica Superior de Ingeniería Informática



UNIVERSIDAD POLITÉCNICA DE VALENCIA
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

PROYECTO FINAL DE CARRERA
2009 - 2010

Sistema de Control Embebido conectable a PC: Diseño y Aplicaciones Prácticas

ALUMNO:

ION LADARESCU

DIRECTORES:

ÁNGEL RODAS

ÁNGEL VALERA

EMPRESA:

INDENOVA

TUTOR EMPRESA:

JORDI GISBERT

ÍNDICE

1.	Introducción.....	6
1.1.	Introducción.....	7
1.2.	Objetivos.....	10
2.	Desarrollo Teórico.....	12
2.1.	Microcontroladores.....	13
2.2.	Protocolos de comunicación.....	21
2.2.1.	PWM.....	21
2.2.2.	DMX.....	24
2.2.3.	USB.....	27
2.3.	Sistemas de control embebido.....	31
2.3.1.	Aplicaciones lúdicas.....	31
2.3.2.	Sistema de iluminación profesional.....	33
2.3.3.	Sistema de navegación móvil.....	35
3.	Desarrollo práctico.....	37
3.1.	Desarrollo del dispositivo genérico. Pic18f4550.....	38
3.2.	Desarrollo de aplicaciones de control y monitorización desde PC.....	46
3.3.	Desarrollo de aplicaciones de control embebido.....	52
3.3.1.	Control de autómatas mecánicos.....	53
3.3.2.	Control de proyectores de iluminación.....	58
3.3.3.	Control de un vehículo tipo quad.....	62
4.	Conclusiones.....	71
4.1.	Conclusiones.....	72
4.2.	Trabajos futuros.....	72
5.	Bibliografía.....	73

ÍNDICE FIGURAS

Figura 1 Sistema	7
Figura 2 Interfaces	8
Figura 3 Encapsulado DIP o DIL.	14
Figura 4 Encapsulado FLAT-PACK.	14
Figura 5 Encapsulado SOIC.	14
Figura 6 Encapsulado LPCC.	15
Figura 7 Encapsulado LCCC.....	15
Figura 8 Evolución del mercado de los microcontroladores y previsiones.....	18
Figura 9 Estructura general de un microcontrolador.....	19
Figura 10 Arquitectura Von Neuman	19
Figura 11 Arquitectura Harvard	19
Figura 12 Prestaciones del Pic18f4550 para el dispositivo genérico	20
Figura 13 PWM	21
Figura 14 Esquema empleo PWM.....	22
Figura 15 Simulación del control de un motor AC con PWM	22
Figura 16 El balanceador de señal DMX.....	24
Figura 17 Estructura del protocolo DMX.....	24
Figura 18 SuperSpeedUsb3.0	27
Figura 19 Esquema conectores usb 3.0	27
Figura 20 La evolución del Usb	28
Figura 21 Esquema conectores Usb.....	30
Figura 22 Juguetes que incorporan microcontrolador	31
Figura 23 Proyectores Dmx de ultima generación	33
Figura 24 MACBOR ATV 49 mini.....	35
Figura 25 Familias de microcontroladores Pic	38
Figura 26 Bloques estructurales del Pic 18f4550	39
Figura 27 Bloques funcionales del dispositivo genérico	41
Figura 28 Componentes del dispositivo genérico.....	42
Figura 29 Programador PICKit2	45
Figura 30 Módulos del firmware y la intercomunicación.	47
Figura 31 Principales funcionalidades de la aplicación software.....	48
Figura 32 Clases de la aplicación	50
Figura 33 El violinista – autómatas electromecánicas	54
Figura 34 Figura circuito servo	54

Figura 35 Figura servo desmontado	55
Figura 36 Diagrama de clases control autómatas	56
Figura 37 Esquema balanceador de señal para Dmx	58
Figura 39 Los dimmers: antes y ahora.	59
Figura 40 Dimmer: exterior, interior y un solo canal.	60
Figura 40 Software de control y monitorización	61
Figura 41 El vehículo quad con los actuadores montados y en fase de experimentación.	62
Figura 42 El servo que actúa sobre el freno del quad.....	63
Figura 44 Motor DC para el control de la dirección Quad.....	65
Figura 44 Disposición de los datos en el vector de bytes	67
Figura 45 Diagrama lógica del firmware.....	68
Figura 46 Clases de la aplicación de control y monitorización.	69
Figura 47 Esquema componentes del quad.	69

ÍNDICE TABLAS

Tabla 1 Microcontroladores de la familia M68HC05 de Motorola.....	16
Tabla 2 Microcontroladores de la familia M68HC08 de Motorola.....	16
Tabla 3 Microcontroladores ATMEL de 8 bits.	16
Tabla 4 Microcontroladores de STMicroelectronics de 8 bits.	16
Tabla 5 Microcontroladores de Microchip de 8 bits.	17
Tabla 6 Tabla de tiempos para el protocolo DMX.	25
Tabla 7 Tabla conexionado cable.USB	30
Tabla 8 Comparativa de prestaciones de Pic con conectividad Usb	41
Tabla 9 Equivalencias digital - analógico.....	66

1. Introducción.

1.1. **Introducción.**

La aparición de los ordenadores ha marcado un cambio importante en todos los sectores de la vida, desde aplicaciones lúdicas hasta procesos industriales. Los ordenadores controlan y monitoriza las actividades, almacenando y procesando datos además de reducir sustancialmente el tiempo de cálculo. Pueden realizar tareas programadas e intercambiar datos a larga distancia lo que abre el abanico de posibilidades de uso.

Las necesidades de automatizar pequeñas tareas, reducir el tamaño y el coste ha desencadenado la aparición de una nueva familia de circuitos integrados: los microcontroladores. En realidad ellos son ordenadores en miniatura. Los encontramos en los juguetes, en los electrodomésticos y hasta en las tarjetas de control industrial.

Las diferencias y similitudes entre los dos logros de la tecnología hacen que los ordenadores y los microcontroladores se complementen mutuamente. Con una comunicación a alta velocidad entre ellos se puede conseguir control en tiempo real de amplia variedad de accionadores y actuadores.

La realización de un soporte físico para el microcontrolador, de fácil conexasión a sus puertos, respaldado por una plantilla de programación, hace que el microcontrolador sea mucho más fácil de acceder, programar y utilizar.

En el presente proyecto abordaremos temas diseño e implementación de rutinas para control por ancho de pulso, adquisición de datos y la implementación del protocolo de comunicación Dmx además del diseño y realización del dispositivo genérico basado en microcontrolador.

El sistema

Para una vista general sobre el sistema explicamos el siguiente esquema estructural:

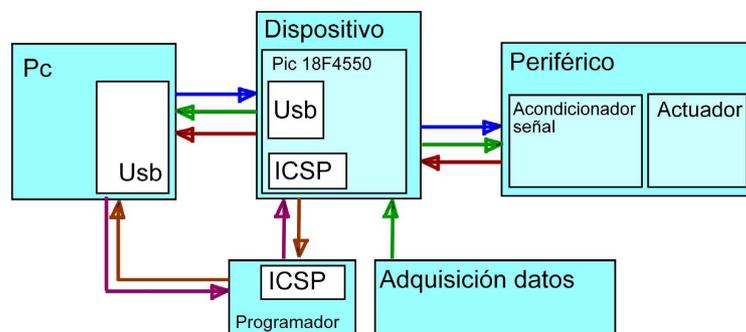


Figura 1 Sistema

La parte que controla el sistema es el Pc que transmite los datos al dispositivo a través del Usb. El dispositivo procesa los datos y en función de las tareas

maneja los actuadores del periférico directamente o encapsula los datos y los reenvía al circuito acondicionador de señal del periférico. El dispositivo también adquiere datos y los envía a PC a través del Usb o los procesa y maneja el periférico en función de ellos, según el modo de funcionamiento programado. El PC visualiza los datos, tanto los del periférico como los capturados por la adquisición de datos. El dispositivo puede ser programado a través del puerto ICSP (In Circuit Serial Programming) por intermedio de un programador para incluir nuevas funcionalidades o actualizaciones del firmware. Podemos describir el sistema como compuesto por varias interfaces que interactúan en cadena.

Esquema de interfaces

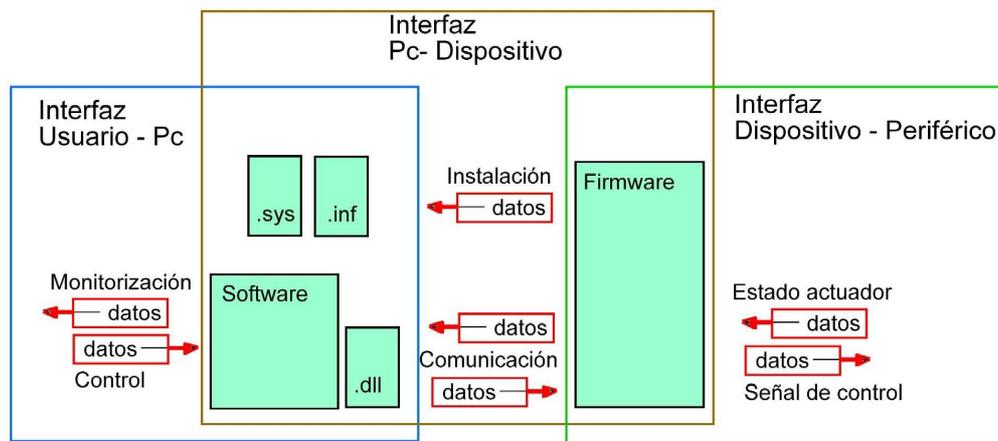


Figura 2 Interfaces

A continuación explicaremos el sistema a través de sus interfaces.

En la primera interfaz, el usuario introduce los datos de control para el periférico. Estos son procesados por el software.

El puente entre las dos primeras interfaces es el software junto a tres archivos: .sys con información para el sistema, .inf, con información del dispositivo y .dll con funciones para la comunicación entre el Pc y el dispositivo.

En la segunda interfaz, los datos son encapsulados y transmitidos al dispositivo a través de las funciones de la librería dinámica para acceso a hardware. El firmware del dispositivo procesa estos datos preparándolos para ser entendidos por el periférico.

El puente entre las interfaces siguientes es el firmware en si que se encarga de recibir y procesar datos, además de manejar los puertos de entrada, salida y procesar datos adquiridos.

En la tercera interfaz, los datos son transmitidos según el protocolo requerido. El periférico puede facilitar datos de su estado y enviarlos al dispositivo. Estos datos son adquiridos por el dispositivo y procesados.

El camino inverso, para la monitorización u otras funcionalidades, es recorrido por los datos de la siguiente manera:

En la segunda interfaz son encapsulados y reenviados al Pc, donde serán preparados para su visualización.

En la primera interfaz, los datos de estado del periférico están visualizados, tratados y/o archivados.

Cabe destacar que al principio, para poder establecer comunicación el dispositivo debe ser reconocido por el sistema operativo (los archivos .sys y .inf). En la segunda interfaz, como primera operación, se realiza la instalación del dispositivo. La instalación, normalmente, se ejecuta una sola vez por puerto Usb.

Hemos de mencionar que en la primera interfaz, los datos facilitados por el usuario pueden ser introducidos a través de controles, especificando un fichero de lectura, o a través de otros dispositivos cómo teclado, ratón, joystick, escáner, etc.

1.2. **Objetivos.**

El objetivo del presente proyecto se centra en el diseño e implementación de un sistema genérico de control embebido, configurable y monitorizado a través de un ordenador. El sistema se adaptará fácilmente a distintos tipos de sensores y actuadores, mediante la programación y configuración en función de las especificaciones de los mismos. A fin de demostrar la utilidad práctica del mismo, se mostrarán ejemplos prácticos de uso sobre distintos ámbitos de aplicación.

El sistema está compuesto por una estructura de componentes físicos, es decir hardware y actuadores, y una parte lógica constituida por el software y el firmware.

Diseñaremos un hardware genérico, programable al que conectaremos un circuito acondicionador de señal en función del tipo de actuadores

La parte lógica debe manejar las señales combinando los datos adquiridos por sensores u otros medios, datos facilitados directamente por el usuario y datos calculados como los resultados de reguladores o convertidores de datos.

La parte hardware será constituida por un ordenador personal al que conectamos un dispositivo Usb diseñado por nosotros, basado en microcontrolador. Diseñaremos la parte acondicionadora de señal en caso de no estar incorporada en los periféricos contenedores de los actuadores. La parte lógica ha de tener una interfaz de control y monitorización del estado de los actuadores y ha de disponer de otras partes, como cálculo y transmisión de señales.

El dispositivo Usb genérico dispondrá de:

- Conectividad usb
- Entrada digital y analógica
- Salida digital.
- Plantilla para el desarrollo de aplicaciones específicas, software y firmware
- Fácil programación y configuración
- El dispositivo genérico se comunica con el ordenador a través del puerto Usb y para ello hemos de desarrollar un software específico.

El proyecto se divide en las siguientes áreas de investigación:

 Análisis del diseño del sistema genérico en función del tipo de microcontrolador.

 Análisis del diseño del sistema genérico en función de la transmisión de datos

 Análisis del diseño del sistema genérico en función del tipo de actuadores

Se plantean los siguientes subobjetivos:

- Diseño del sistema de control, basado en microcontrolador, que gobernará el resto de subsistemas conectados al mismo.
- Diseño del subsistema de entrada/salida que incluirá la interfaz con los sensores y actuadores así como la comunicación con el computador a través del bus Usb.
- Diseño del software y firmware de manejo de señales y control que permitirá la combinación de distintas fuentes de datos y establecerá las estrategias de control/actuación oportunas.
- Creación de un conjunto de aplicaciones prácticas del sistema, en diversos ámbitos, para comprobar la versatilidad del mismo

2. Desarrollo Teórico.

2.1. Microcontroladores.

Un microcontrolador es un circuito integrado que incorpora procesador, memoria ROM y RAM, puertos de entrada/salida y otros dispositivos de propósito especial como conversores A/D, contadores, temporizadores y puertos de comunicación. En otras palabras, un microcontrolador es un computador en miniatura.

Estos elementos le convierte en una completa solución para el diseño de sistemas empotrados. Los podemos encontrar en todos los aparatos, instrumentos o maquinaria, desde juguetes, electrodomésticos hasta unidades de disco, impresoras o coches. Hay fabricantes que ofrecen una amplia variedad de microcontroladores, siendo el diseñador del dispositivo el que hace la elección.

Los inicios de los microcontroladores

A principios de los años 70, la división de microelectrónica de la empresa Intel, diseña el microprocesador de 16 bits CP1600. Esta aparición trajo como consecuencia la fabricación de un conjunto de circuitos integrados que resolvían las necesidades de memoria, de entradas/salidas, temporizadores, conversores, etc. Los fabricantes de microprocesadores, conscientes de la importancia de este mercado, pusieron a disposición de los técnicos una gama variada de circuitos integrados que facilitaban la construcción de sistemas.

En 1975 la misma empresa diseña un controlador rápido pero limitado y con pocas instrucciones para funcionar en combinación con el CP1600. Era un chip destinado a controlar E/S: el PIC (Peripheral Interface Controller).

En el año 1976, se integran junto con el microprocesador los subsistemas que anteriormente formaban unidades especializadas e independientes, pero unidas por las pistas de circuito impreso con el microprocesador formando lo que se conoce como sistema. Este nuevo circuito integrado se llamó microcomputador monopastilla, es decir el microcontrolador.

El primer microcontrolador de 8 bits se considera al 8048, fabricado por Intel en el 1979.

Otra de las principales empresas del mundo de los dispositivos microprogramables es Motorola, que dispone del potente microcontrolador 68HC11 de 8 bits para datos y 16 para direcciones.

Los microcontroladores PIC de la empresa Microchip han sido conocidos durante los 20 años últimos pero se adjudicó el título de los más populares gracias al reducido coste, herramientas de desarrollo y fácil integración.

También Zilog ha irrumpido con fuerza en el mercado de los microcontroladores. Recientemente ha lanzado algunos modelos con memoria OTP de la familia Z86XX, muy completos y fáciles de manejar.

Otras empresas como SGS-Thomson, Hitachi, Texas, Toshiba, National Semi-Conductor y muchos otros, abarcan pequeñas partes del mercado.

Tipos de encapsulados de microcontroladores:

Encapsulado DIP o DIL. Este es el encapsulado más empleado en montaje por taladro pasante en placa. Este puede ser cerámico (marrón) o de plástico (negro). Un dato importante en todos los componentes es la distancia entre patillas que poseen, en los circuitos integrados es de vital importancia este dato, así en este tipo el estándar se establece en 0,1 pulgadas (2,54 mm.). Se suelen fabricar a partir de 4, 6, 8, 14, 16, 22, 24, 28, 32, 40, 48, 64 patillas, estos son los que más se utilizan.



Figura 3 Encapsulado DIP o DIL.

Encapsulado FLAT-PACK. Se diseñan para ser soldados en máquinas automáticas o semiautomáticas, ya que por la disposición de sus patillas de pueden soldar por puntos. El material con el que se fabrican es cerámico. La distancia entre patillas es de 1,27 mm., la mitad que en los DIP.



Figura 4 Encapsulado FLAT-PACK.

Encapsulado SOIC. Circuito integrado de pequeño contorno. Son los más populares en los circuitos de lógica combinatorial, tanto en TTL como en CMOS. Se sueldan directamente sobre las pistas de la placa de circuito impreso, en un área denominada footprint. La distancia entre patillas es de 1,27 mm. (0,05").



Figura 5 Encapsulado SOIC.

Encapsulado LPPC. Se emplea en técnicas de montaje superficial pero, generalmente, montados en zócalos, esto es debido a que por la forma en J que tienen sus terminales la soldadura es difícil de verificar con garantías. Esto permite su uso en técnicas de montaje convencional. Se fabrican en material plástico. La distancia entre terminales es de 1,27 mm.



Figura 6 Encapsulado LPPC.

Encapsulado LCCC. Al igual que el anterior se monta en zócalo y puede utilizarse tanto en montaje superficial como en montaje de taladro pasante. Se fabrica en material cerámico.

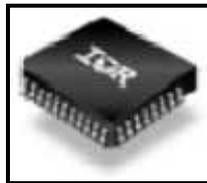


Figura 7 Encapsulado LCCC.

Los encapsulados anteriormente señalados son los más importantes y los más utilizados, aunque no los únicos que existen.

Principales fabricantes

Algunos de los fabricantes más importantes y especializados en microcontroladores son los siguientes:

- Motorola Semiconductors Corp.
- Atmel Corp.
- Microchip Technology Corp.
- STMicroelectronics

Familia de microcontroladores de 8 bits M68HC05 de Motorola:

DISPOSITIVO	RAM	EEPROM	TIMER	A/D	WMM	I/O	MHZ	TENSIÓN
68HC05B8	176	256	4	8-8b	2	32	1,2.1	3.3-5V
68HC05B16	352	256	4	8-8b	2	32	1,2.1	3.3-5V
68HC05B32	352	256	4	8-8b	2	32	1,2.1	3.3-5V
68HC705B16	528	256	4	8-8b	2	32	2.1,4	3.3-5V
68HC705B32	528	256	4	8-8b	2	32	1,2.1	3.3-5V
68HC05X16	352	256	4	8-8b	2	32	2.2	5V
68HC05X32	528	256	4	8-8b	2	32	2.2	5V
68HC705X32	528	256	4	8-8b	2	32	2.2	5V

Tabla 1 Microcontroladores de la familia M68HC05 de Motorola.

Familia de microcontroladores de 8 bits M68HC08 de Motorola:

DISPOSITIVO	RAM	TIMER	A/D	PWM	I/O	MHZ	TENSIÓN
68HC08JK8	256	2	10-8b	2-16B	15	8	3V
68HC908JL8	256	2	13-8b	2-16b	26	8	3V
68HC08JL8	256	2	13-8b	2-16b	26	8	3V
68HC908KX2	192	2	4-8b	2-16b	13	8	3-5V
68HC908KX8	192	2	4-8b	2-16b	13	8	5V

Tabla 2 Microcontroladores de la familia M68HC08 de Motorola.

Los microcontroladores de ATMEL de 8 bits incorporan de tres a ocho canales PWM, trabajan a 16 MHz y poseen bastantes canales de convertidor analógico digital:

DISPOSITIVO	RAM	TIMER	A/D	PWM	I/O	MHZ	TENSIÓN
Atmega169		3	8-10b	4	53	16	4.5-5.5V
Atmega16		3	8-10b	4	32	16	4.5-5.5V
Atmega8		3	6-8,10b	3	23	16	4.5-5.5V
Atmega8535		3	8-10b	4	32	16	4.5-5.5V
Atmega32		3	8-10b	4	32	16	4.5-5.5V

Tabla 3 Microcontroladores ATMEL de 8 bits.

STMicroelectronics presenta también tres familias de microcontroladores de 8 bits: ST5, ST6, ST7 y ST9. Algunos de los que presentan funciones A/D y PWM son los siguientes:

DISPOSITIVO	RAM	TIMER	A/D	PWM	I/O	MHZ	TENSIÓN
ST52430K2	256	3	8-8b	3	23		3-5.5V
ST52510K2	512	2	8-10b	2	22		2.7-5.5V
ST72262G1	256	2	6-10b	2	22		2.4-5.5V
ST72621J2	384	2	8-10b	2	31		4-5.5V
ST92141K4	512	2	8-8b	2	15		4-5.5V

Tabla 4 Microcontroladores de STMicroelectronics de 8 bits.

La familia de microcontroladores de Microchip es la más extensa contando cerca de 176 microcontroladores distintos. A continuación se muestra alguno de ellos:

DISPOSITIVO	RAM	TIMER	A/D	PWM	I/O	MHZ	TENSIÓN
PIC16F873	192	3	5-10b	2	21	20	2-5.5V
PIC16F874	192	3	8-10b	2	32	20	2-5.5V
PIC16F876	368	3	5-10b	2	21	20	2-5.5V
PIC16F877	368	3	8-10b	2	32	20	2-5.5V
PIC16F73	192	3	5-8b	2	22	20	
PIC16F74	192	3	8-8b	2	33	20	
PIC16F76	368	3	5-8b	2	22	20	
PIC16F77	368	3	8-8b	2	33	20	

Tabla 5 Microcontroladores de Microchip de 8 bits.

Se han mostrado los microcontroladores de 8 bits por ser una de las especificaciones requeridas, aunque también existen microcontroladores de 4, 16 y 32 bits.

Los microcontroladores de 16 y 32 bits a pesar de no ser tan cotidianos como los de 8 bits son utilizados cada vez más en aplicaciones en que los datos requieren más precisión (instrumentación, operaciones matemáticas complejas, etc.).

Estos microcontroladores de 16 y 32 bits suelen ser muy rápidos si se comparan con los de 8 bits además de ser capaces de direccionar y gestionar grandes volúmenes de memoria.

Evolución en el mercado

Aunque en el mercado de la Microinformática la mayor atención la acaparan los desarrollos de los microprocesadores, lo cierto es que se venden cientos de microcontroladores por cada uno de aquéllos.

La evolución del mercado mundial de microcontroladores de 8 bits ha experimentado un espectacular incremento a partir del año 1990, en el que se vendieron 590 millones de unidades. En 1992 se vendieron 950 millones y en 1996 se superaron los 2.000 millones de unidades.

La distribución de las ventas según su aplicación es la siguiente:

- Una tercera parte se absorbe en las aplicaciones relacionadas con los computadores y sus periféricos.
- La cuarta parte se utiliza en las aplicaciones de gran consumo (electrodomésticos, juegos, TV, vídeo, etc).
- El 16 % de las ventas mundiales se destinó al área de las comunicaciones.
- Otro 16% fue empleado en aplicaciones industriales.
- El resto de los microcontroladores vendidos en el mundo, aproximadamente un 10% fueron adquiridos por las industrias de automoción.

Respecto al mercado de los potentes microcontroladores de 16 bits, el crecimiento también ha sido su norma. De los 23 millones de unidades vendidas en 1988, se ha pasado a diez veces más en 1996, siendo la ofimática y las comunicaciones las áreas que absorbieron la mayor parte.

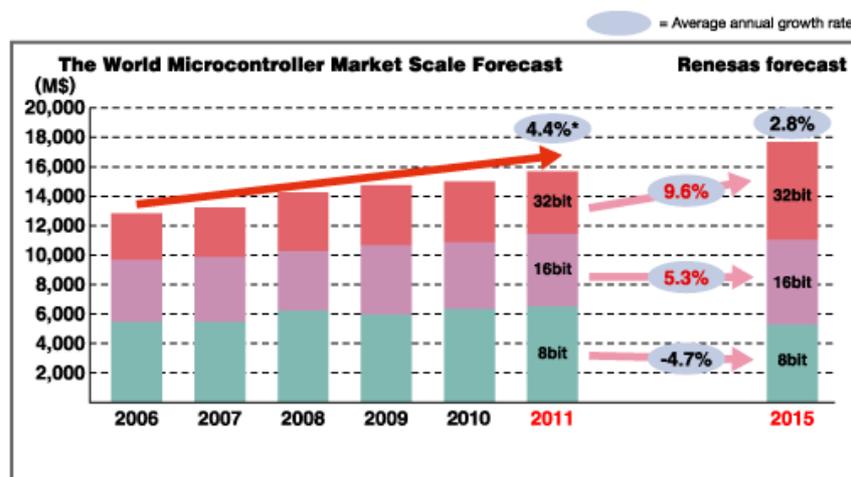


Figura 8 Evolución del mercado de los microcontroladores y previsiones

Estructura general de un microcontrolador:

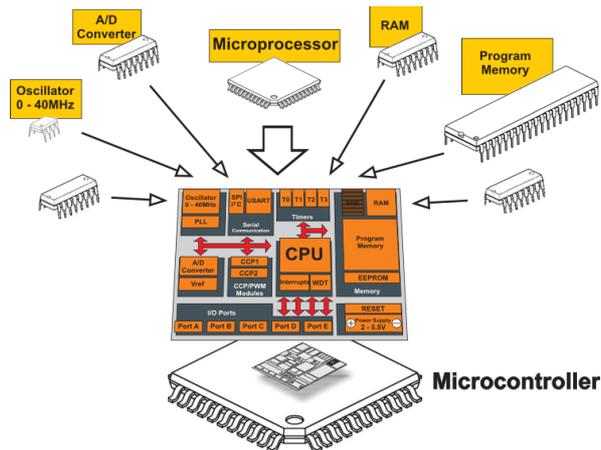


Figura 9 Estructura general de un microcontrolador

Arquitecturas:

Arquitectura **Von Newman** dispone de una sola memoria principal donde se almacenan datos e instrucciones de forma indistinta. A dicha memoria se accede a través de un sistema de buses único (direcciones, datos y control).

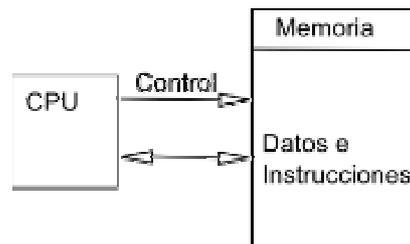


Figura 10 Arquitectura Von Neuman

Arquitectura **Harvard** dispone de dos memorias independientes, una que contiene sólo instrucciones, y otra que contiene sólo datos. Ambas disponen de sus respectivos sistemas de buses de acceso y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas memorias.

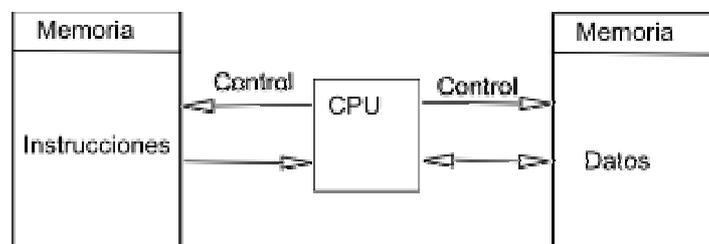


Figura 11 Arquitectura Harvard

Podemos clasificar los microcontroladores según el **juego de instrucciones**

CISC (complex instruction set computer)

Es un juego de instrucciones muy amplio que permite operaciones complejas entre operandos situados en la memoria o en los registros internos. Son instrucciones potentes.

RISC (reduced instruction set computer)

Es un juego de pocas instrucciones que realizan funciones sencillas. Las instrucciones RISC tienen las siguientes características fundamentales:

- Son instrucciones de tamaño fijo en un reducido número de formatos.
- Sólo las instrucciones de carga y almacenamiento acceden a memoria.

SISC (specific instruction set computer)

Es un juego de instrucciones específicos para tareas concretas.

En definitiva, los microcontroladores suelen incluir variedad de funciones especiales que se pueden utilizar gracias a los dispositivos internos incluidos dentro de ellos, entre otras:

La memoria de programa generalmente es una Flash EEPROM.

Tiene puertos de Entrada y Salida (Configurables por software).

Poseen contadores de propósito especial.

Tiene incluido un reloj del sistema que permite contabilizar tiempo.

Algunos modelos incluyen convertores A/D.

Tiene Memoria EEPROM para almacenar datos.

Tiene puerto de comunicaciones.

Manejan velocidades de operación hasta 20 MHz.

Algunos de estos dispositivos tienen puerto de comunicaciones serial.

Tienen entradas para interrupción.

La programación es rápida.

Las herramientas de desarrollo son económicas y se encuentran disponibles en a red, las cuales incluyen el ensamblador y simulador.

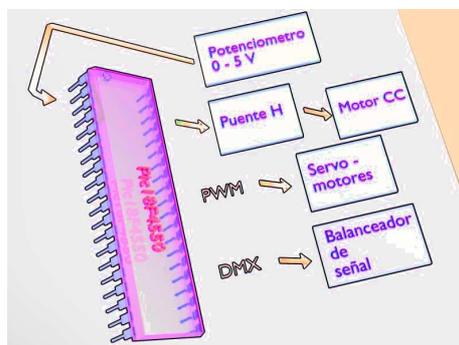


Figura 12 Prestaciones del Pic18f4550 para el dispositivo genérico

Hoy en día, los microcontroladores son un buen recurso al alcance de cualquier desarrollador de hardware.

2.2. *Protocolos de comunicación.*

2.2.1. PWM.

El control por ancho de pulso (Pulse Wide Modulate) es la forma de control que puede ser empleada también cómo protocolo de comunicación. La principal ventaja es su simplicidad respaldada por la generalidad de sus parámetros. PWM es una señal periódica en la que los datos son representados por el ancho del pulso. Esto quiere decir que se aplica un pulso cada intervalo de tiempo. La duración del pulso ha de ser inferior o igual al intervalo tiempo.

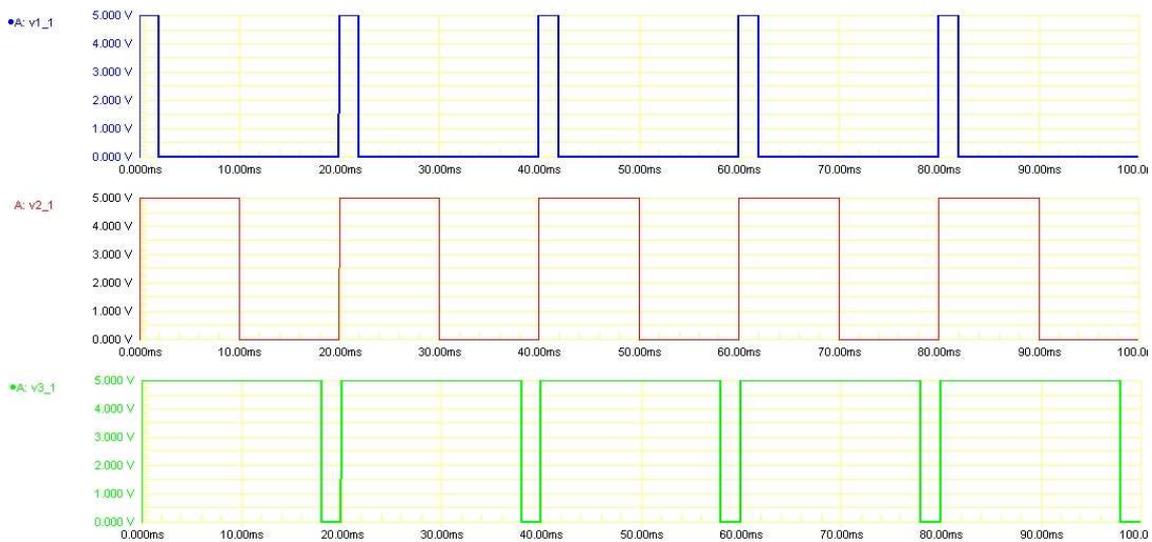


Figura 13 PWM

Las variables a tener en cuenta son el ancho de pulso y la frecuencia de envío de pulso. El ciclo de trabajo se calcula dividiendo el ancho de pulso por la frecuencia.

Las aplicaciones son en el campo del control de motores y parte de un convertidor analógico digital.

En caso del control de motores, el ancho de pulso se interpreta cómo velocidad de giro para los motores de corriente continua y cómo posición para los servos. Para las conversiones, se hace la traducción del voltaje a pulsos a través de un filtro.

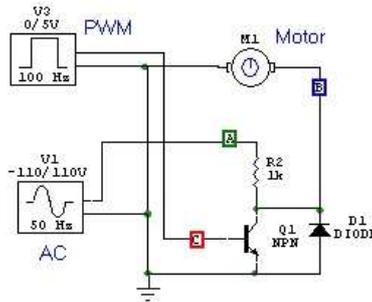


Figura 14 Esquema empleo PWM

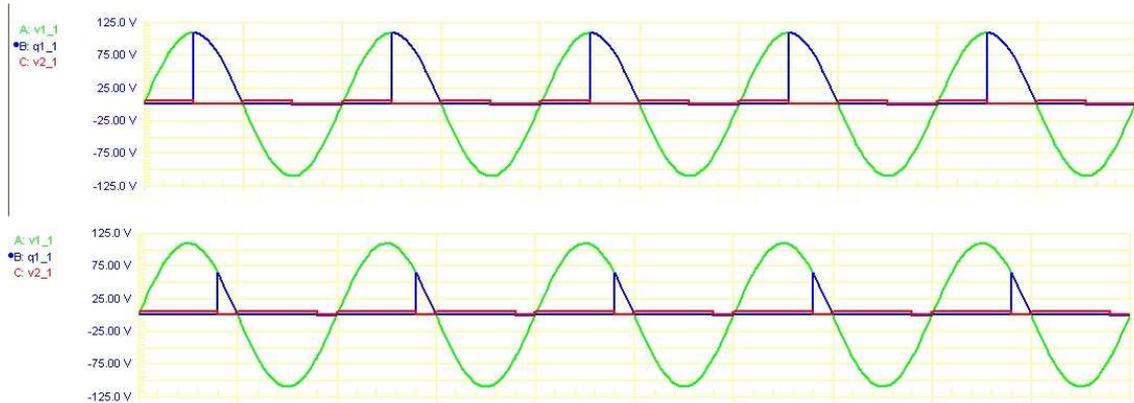


Figura 15 Simulación del control de un motor AC con PWM

Parte de la electrónica de control la velocidad de rotación de un motor y la simulación mediante PWM invertida al 50% y 80% respectivamente. A – corriente alterna, B – señal PWM, C – salida alimentación controlada para Motor. Actúa solamente en el ciclo positivo de la fase. Se debe sincronizar la fase con el Pwm ajustándolo a la misma frecuencia y emplear un detector de paso por cero.

Ejemplos del empleo del control por ancho de pulso del valor de la corriente continua:

El control del valor de la corriente continua a partir de una onda cuadrada;

El valor de la corriente continua en función del ancho del pulso.

La corriente continua se puede crear empleando un filtro pasa-bajos.

El control del valor variando el periodo del pulso (frecuencia)

La flexibilidad de la solución PWM permite a cualquier diseñador de sistemas emplearlo con las más variadas configuraciones: ancho de pulso, forma de onda, frecuencia, todo ello constituyendo una sencilla y eficaz forma de comunicación exenta de encabezados o redundancias.

En cuanto al sistema físico idóneo para el PWM, no hay especificaciones sobre el sistema en el que se puede implementar, dejando al diseñador decidir y especificar las características del bus de transmisión en función de la frecuencia y corriente.

Actualmente el control por ancho de pulso está empleado desde el control de la velocidad de rotación de los ventiladores existentes en los ordenadores, pasando por el posicionamiento gracias a los servomotores y hasta los variadores de intensidad para el control de velocidad de motores de corriente alterna.

2.2.2. DMX.

DMX es un protocolo utilizado para el control de la iluminación. Surgió en el 1986 para solucionar el problema de incompatibilidad que existía entre los distintos fabricantes de aparatos de control de iluminación que usaban protocolos propietarios.

El protocolo DMX es un protocolo muy directo. Consta en un bloque de un mínimo de 512 octetos sobre una línea serie RS485. Estos 512 octetos, también llamados un universo, representan 512 diversos valores para regular la iluminación o son parámetros (por ejemplo, color o número del gobo para los proyectores móviles). La simplicidad del protocolo permite utilizar microcontroladores para enviar/recibe la señal de DMX. La señal debe de ser balanceada, normalmente se usa el componente SN75176 que es bidireccional.

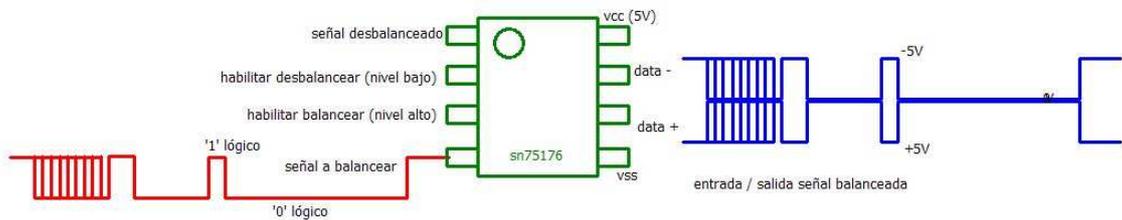


Figura 16 El balanceador de señal DMX

Se utiliza una velocidad 250kbit/sec, por lo tanto no es posible generar DMX con un puerto serial estándar del PC (velocidad máxima puerto serie 100kbit/sec).

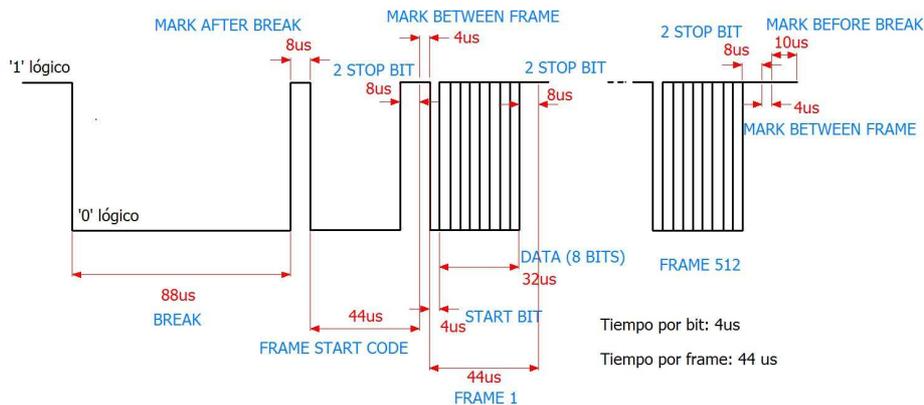


Figura 17 Estructura del protocolo DMX

DMX utiliza una codificación estándar del octeto 8N2, que significa 1 startbit, 8 databits y 2 stopbits. Para saber cuándo un nuevo bloque de 512 comienzos del octeto, tiene una secuencia especial del reajuste que se transmite.

No hay señal	1	tiempo: continuo			
Break	0	tiempo min 88us	tiempo max 176us		
Mark after break	1	tiempo 8us			
Start Code:	1 frame				
	start bit	0	tiempo 4us		
	codigo inicio	8 bits a 0	tiempo 8bitsx4us=32us		
	stop bits	2 bits a 1	tiempo 2bitsx4us=8us		
Mark Between Frame	1	tiempo min 0us	tiempo max 1 sec		
Universo	min 1 frames				
		frame	start bit	0	tiempo 4us
			codigo inicio	8 bits valor	tiempo 8bitsx4us=32us
			stop bits	2 bits a 1	tiempo 2bitsx4us=8us
		mark between frame	1	tiempo min 0us	tiempo max 1sec
	max 512 frames				
Mark Before Break	1	tiempo min 0us	tiempo max 1 sec		

Tabla 6 Tabla de tiempos para el protocolo DMX.

Especificaciones físicas del Bus DMX:

es un bus serie: adecuado para la transmisión de datos según la norma EIA 485 para 250 Kbaudios. El medio de transmisión es cable de impedancia nominal de 120 ohmios (80 - 150) y baja capacitancia., apantallado y de par trenzado. Para algunas instalaciones, se requiere doble línea y en ese caso se emplea cable con sendos pares trenzados y pantalla.

- El grosor mínimo de los conductores depende de la distancia: 24 AWG hasta 300 metros, 22 AWG hasta 500 metros. Aunque teóricamente se podrían duplicar estas distancias, por las condiciones habituales de uso, no es recomendable.
- Conectores XLR de 5 pines (1: masa, 2: data-, 3: data+, 4: data'-, 5: data'+). Los pines 4 y 5 sólo se emplean en caso de utilizar cable con doble par. Lo normal es utilizar únicamente los pines 1, 2 y 3.

Limitaciones del protocolo

El máximo de 512 canales sólo se pueden aumentar multiplexando los dimmers de tal manera que el modulo multiplexor envía el “universo” al grupo de dimmers correspondiente.



Mesa de control de iluminación profesional y dimmers respectivamente que emplean el protocolo DMX

Las tendencias son optimistas: debido a demanda masiva de aparatos dmx utilizados en la iluminación profesional, el dmx será un protocolo duradero y en continuo desarrollo. Actualmente se han ampliado el numero de universos que puede manejar un aparato dmx y se usa encapsulado sobre ethernet para el control a largas distancias o para la gestión de redes complejas.

2.2.3. USB.

El puerto Usb, actualmente se encuentra en la versión 2.0 y lleva unos ocho años sin cambios. Es mucho tiempo teniendo en cuenta que la informática avanza con pasos gigantescos. Intel ha estado trabajando en la nueva versión Usb 3.0 y a finales del 2008 ha publicado las especificaciones del nuevo puerto.

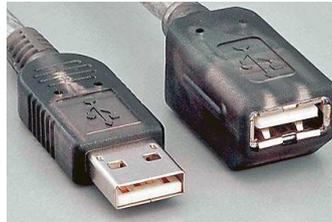


Figura 18 SuperSpeedUsb3.0

Con más líneas en el bus, dispondrá de comunicación bidireccional simultánea y soportará más carga por parte de dispositivos que necesitan alimentación, nueve veces la corriente que suministra el Usb 2.0. La velocidad de transmisión será sustancialmente mejorada llegando a los 4.8Gb/s, es decir una velocidad de vértigo.

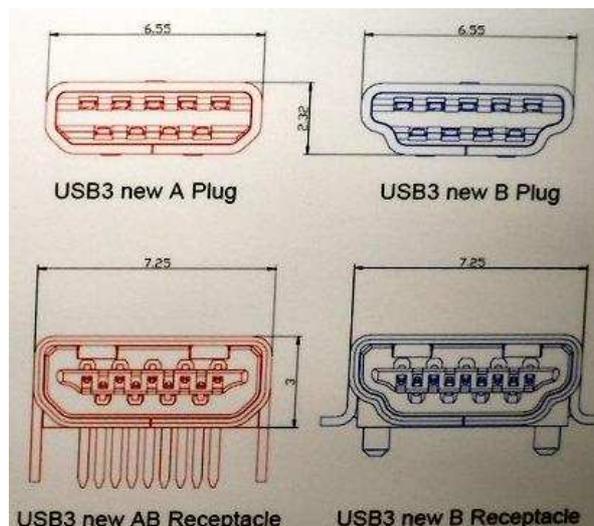


Figura 19 Esquema conectores usb 3.0

Otra novedad respecto a su antecesor será la comunicación basada en interrupciones, Usb 2.0 utilizaba la técnica de pooling. Como cualquier trabajo bien hecho, Usb 3.0 dispondrá de total compatibilidad con su anterior versión y su aparición, en dispositivos completamente compatibles, está prevista para principios del 2010.

Pero las raíces de los puertos Universal Serial Bus radican en su creación en el 1996 por las empresas IBM, Intel, Northern Telecom, Compaq, Microsoft, Digital Equipment Corporation y NEC para poder conectar periféricos a un ordenador. Con ello se conseguía a la vez mejorar la conectividad plug and

play de los periféricos, permitiendo conexiones “en caliente” y suministrando alimentación al dispositivo.

2.2.3.1. Estándares.

USB 0.9: Primer borrador, publicado en Noviembre de 1995.

USB 1.0: Publicada en 1996 (“Low speed”), ofrece 1.5 Mbps, (“Full speed”), es de 12 Mbps,

USB 1.1: Publicada en 1998, añade detalles y precisiones a la norma inicial; es el estándar mínimo que debe cumplir un dispositivo USB.

USB 2.0: Su versión final fue publicada en Abril del 2000; High speed” 480Mbps



USB Evolution

Spec	Data Rate and Performance	Applications	Notes
USB 1.1 Low-speed USB 2.0 Low-speed 	1.5Mbps	Keyboard, mouse, joystick 	Low cost but limited performance; type and number of endpoints limited
USB 1.1 Full-speed USB 2.0 Full-speed 	12Mbps	Printers, audio devices, webcams 	Moderate performance; guaranteed latency; guaranteed bandwidth
USB 2.0 High-speed 	480Mbps	Video, storage, imaging & more... 	Vast bandwidth improvements. Adoption in high-end consumer segment
USB 2.0 On-The-Go (OTG) 	480Mbps or 12Mbps	PDAs, MP3 players, cameras 	Peripheral device can communicate directly without Host PC Backward compatible
USB Wireless (WUSB) 	480Mbps @ 3 meters 110Mbps @ 10 meters		Recently specified. Still going through adoption process

© 2008 Microchip Technology Incorporated. All Rights Reserved. USB Solutions from Microchip Slide 6

Figura 20 La evolución del Usb

2.2.3.2. Historia.

El primer ordenador que incluyó un puerto USB de forma estándar fue el iMac de Apple, presentado en Marzo de 1998, que utilizaba esta conexión para el teclado y el ratón. Por su parte el mundo del PC solo comenzó a utilizarlo cuando Microsoft introdujo los controladores correspondientes en la versión OSR 2.1 de Windows 95.

2.2.3.3. Topología.

Los dispositivos USB adoptan una topología de estrella y se organiza por niveles a partir de un controlador host instalado en la placa base, que actúa de interfaz entre el bus de ésta y el primer dispositivo USB, el denominado concentrador raíz (“Root hub”), instalado también en la placa.

El bus USB soporta intercambio simultáneo de datos entre un ordenador anfitrión y un amplio conjunto de periféricos.
En un bus USB existen dos tipos de elementos: Anfitrión (“host”) y dispositivos;

2.2.3.4. Funcionamiento.

El bus serie USB es síncrono, y utiliza el algoritmo de codificación NRZI (“Non Return to Zero Inverted”)

Existen cuatro tipos de paquetes distintos:

Token, Datos, Handshake y Especial.

El máximo de datos por paquete es de 8; 16; 32 y 64 Bytes. Se utiliza un sistema de detección y corrección de errores bastante robusto tipo CRC (“Cyclical Redundancy Check”).

El funcionamiento está centrado en el host, todas las transacciones se originan en él. Es el controlador host el que decide todas las acciones, incluyendo el número asignado a cada dispositivo (esta asignación es realizada automáticamente por el controlador “host” cada vez que se inicia el sistema o se añade, o elimina, un nuevo dispositivo en el bus), su ancho de banda, etc. Cuando se detecta un nuevo dispositivo es el host el encargado de cargar los drivers oportunos sin necesidad de intervención por el usuario.

El sistema utiliza cuatro tipo de transacciones que resuelven todas las posibles situaciones de comunicación. Cada transacción utiliza un mínimo de tres paquetes, el primero es siempre un Token que avisa al dispositivo que puede iniciar la transmisión.

- Transferencia de control (“Control transfer”): Ocurre cuando un dispositivo se conecta por primera vez.
- Transferencia de pila de datos (“Bulk data transfer”): Este proceso se utiliza para enviar gran cantida de datos de una sola vez.
- Transferencia por interrupción (“Interrupt data transfer”): Este proceso se utiliza cuando se solicita enviar información por el bus en una sola dirección (de la función al host).
- Transferencia de datos isócrona (“Isochronous data transfer”): Este proceso se utiliza cuando es necesario enviar datos en tiempo real. Los datos son enviados con una cadencia precisa ajustada a un reloj, de modo que la transmisión es a velocidad constante.

2.2.3.5. Cables y conectores.

El cable de bus USB es de 4 hilos, y comprende líneas de señal (datos) y alimentación, con lo que las funciones pueden utilizar un único cable.

Pin	Nombre	Descripción	Color
1	VBUS	+ 5 V. CC	rojo
2	D-	Data -	azul
3	D+	Data +	amarillo
4	GND	Tierra	verde

Tabla 7 Tabla conexasión cable.USB

Se usan dos tipos de conectores, A y B. Los de tipo A utilizan la hembra en el sistema anfitrión, y suelen usarse en dispositivos en los que la conexión es permanente (por ejemplo, ratones y teclados). En general podemos afirmar que la hembra de los conectores A están en el lado del host (PC) o de los concentradores (hubs), mientras las de tipo B están del lado de los periféricos.

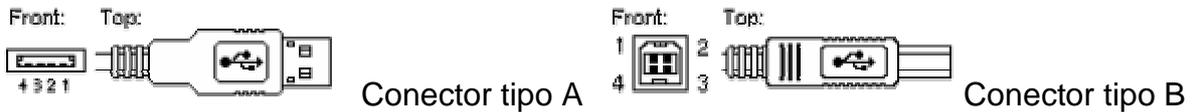


Figura 21 Esquema conectores Usb

A la espera del nuevo superspeed 3.0 sabemos que podemos contar con las ventajas del 2.0 en las versiones miniUsb para conectar el móvil, cámara fotográfica o video, y que algunos fabricantes han usado para crear versiones todo-en-uno, como ejemplo siendo los móviles HTC con puerto miniUsb de cinco pines con conexión de auriculares.

La introducción del Usb ha sido exitosa desde sus principios y es considerada una solución amplia y veloz ofreciendo a la vez versatilidad y comodidad.

2.3. **Sistemas de control embebido.**

2.3.1. **Aplicaciones lúdicas.**

La industria de los juegos y los juguetes incorporando cada vez más en sus productos logros tecnológico. Juegos informáticos, juguetes robotizados y libros que cuentan los cuentos son unos ejemplos. También los sistemas de control embebido están presentes en las salas de juego apoyando una rama de la diversión de la más generadora de beneficios.



Figura 22 Juguetes que incorporan microcontrolador

Juguetes actuales que incorporan microcontrolador. Son capaces de gestionar protocolos de comunicación con el mando a distancia por radiofrecuencia y controlar servos y luces, es programable y tiene varios modos de funcionamiento.

El ordenador es una herramienta potente y el dispositivo genérico permite la conexión de cualquier tipo de juguete para ser interactuado en tiempo real. Las aplicaciones lúdicas son un buen punto de partida para experimentar con el control del sistema genérico objeto de este proyecto. Además se puede utilizar la parte artística para una buena publicidad.

Las marionetas son un medio de expresión que reúne la escultura, la pintura y la cinemática. Desde la antigüedad se han empleado para divertir, contar o criticar y un objetivo ha sido automatizar.

Los autómatas mecánicos se han hecho visibles desde tiempos muy lejanos. La construcción de unos autómatas robotizados controlados por ordenador se convierte en nuestra primera aplicación. El tema es la música, las marionetas son los músicos y el director es el ordenador. El director dirige a los músicos en una frenética comunicación a través del puerto Usb.

Las ventajas de unos autómatas mecánicos controlados por ordenador son entre otras las precisiones de movimiento, el sincronismo de los movimientos con el sonido y el grande número de elementos que se pueden mover a la vez.

Esta aplicación necesita una adaptación específica del software de control y un tedioso estudio para el diseño de los mecanismos.

La programación de los movimientos se realizará en pocos pasos gracias a la programación abreviada: se trata de la interpretación de los datos existentes en archivos de sonido y su posterior conversión a los movimientos de los músicos. En futuras versiones ampliaremos el campo con la interpretación de partituras.

En esta primera aplicación práctica emplearemos Pwm para el control de los servos. Quizás lo más interesante de esta aplicación es el control de un número elevado de servos y la optimización del algoritmo de envío de señal para conseguir el máximo rendimiento de los recursos del microcontrolador.

En gran parte, el resultado final está condicionado por los mecanismos. Por ello intentaremos sintetizar y factorizar para obtener un máximo rendimiento de un motor maestro, de tal manera que con valores parecidos se consiga un amplio abanico de gestos.

2.3.2. Sistema de iluminación profesional.

Controlar la iluminación en el mundo del teatro, cine y televisión se ha convertido en una necesidad que enriquece las escenas aportando calidad y sello de un trabajo profesional. El empleo de escenas con efectos de iluminación insólitas, transiciones y cambio de colores proyectados sobre el vestuario de los actores y las escenografías enriquece el espectáculo y llena de emociones difíciles de describir. Lo que hace unos años era una difícil tarea a la que se necesitaba varios días de montaje, enfoque y aprendizaje o programación de las escenas con limitaciones, se ha convertido en un trabajo lleno de emoción para los diseñadores que pueden contemplar el resultado en directo a golpe de ratón. Ahora deben saber que el límite es su propia imaginación.



Figura 23 Projectores Dmx de última generación

En la figura 23 observamos periféricos de iluminación: proyectores móviles DMX dotados con circuito para la recepción del protocolo y actuación según 24 y 11 canales respectivamente para posicionamiento, intensidad, color y forma de la área de iluminación entre otras. Incorporan microcontrolador.

Configuraremos nuestro sistema para reproducir efectos de iluminación de encendido/apagado de proyectores con velocidad variable en función de las escenas, disparadas de manera manual o automática, fundidos o relámpagos. Todo esto se consigue gracias a una mesa de control, el pc y dispositivo genérico en nuestro caso, conectada a un dimmer que maneja los proyectores - actuadores.

Cómo a hemos comentado, la distancia máxima entre nuestro dispositivo y el pc es de 5m necesitamos poder llevar la señal a más distancia. Los dimmers usan un protocolo que permite una distancia máxima de hasta 500m. Se trata de implementar el protocolo DMX en nuestro dispositivo genérico para que este envíe los datos hasta los dimmers. Hemos de acondicionar la salida de señal del pin configurado como DMX para cumplir con los requerimientos del bus DMX.

La incorporación de un sistema de control embebido a las aplicaciones de control de la iluminación aporta una considerable evolución en la programación, almacenamiento y manejo de escenas y efectos, pasando de la tediosa tarea de la atención constante de varios técnicos durante el montaje y la

representación a la programación de la misma una vez y cargar y ejecutar en las futuras actuaciones.

Actualmente, en la iluminación profesional, se encapsula el universo DMX sobre Ethernet, para una transmisión más fiable y a larga distancia, sobre todo si se trata de una red más compleja de proyectores. A la vez que la informática ha evolucionado, sus aplicaciones en el campo de los proyectores se ha dejado ver. Hay proyectores motorizados, llamados móviles, que utilizan hasta 24 canales DMX para el control de posición en dos ejes, cambia el color del filtro, maneja formas recortadas llamadas "gobos" o la posición de la lente para enfocar o desenfocar los bordes del radio iluminado, además de la intensidad de la luz. Otra ventaja inmediata del encapsulamiento del DMX sobre Ethernet son la posibilidad de la gestión de los proyectores de manera inalámbrica. Adaptar nuestro sistema a una comunicación sobre Ethernet constituye uno de los futuros trabajos que nos planteamos.

La aplicación de nuestro sistema en la iluminación profesional supone emplear el mismo en una aplicación profesional por la implementación de un protocolo estándar y la realización de un circuito acondicionador de señal específico.

2.3.3. Sistema de navegación móvil.

La navegación móvil automatizada es un objetivo perseguido continuamente y no tan fácil de realizar. Hoy en día se ha conseguido con técnicas nada económicas, sobretodo por el problema de posicionamiento, lectura de sensores, interpretación de obstáculos, reconocimiento de formas, en otras palabras, la combinación de inteligencia artificial y visión por ordenador con un buen respaldo hardware.

Lo que planteamos cómo aplicación practica es la adaptación del sistema aquí presentado para dar soporte físico entre los sensores de posición y los actuadores para posicionar el vehículo en el lugar deseado.

La incorporación del sistema de control embebido facilita la gestión de los actuadores incorporados en un sistema de navegación móvil, siendo éste capaz de transmitir a las partes electromecánicas las maniobras necesarias para cumplir el objetivo. Con ello se consigue autonomía de actuación una vez recibido el objetivo e informado continuamente el estado hasta el cumplimiento. En el sistema de navegación móvil planteado, nosotros intervendremos en la parte de control del acelerador, freno y dirección al que llamaremos el sistema de movilidad. Estos tres elementos son manejados automáticamente por el ordenador al través del sistema genérico configurado adecuadamente para actuar como puente.



Figura 24 MACBOR ATV 49 mini

El vehículo quad, concretamente el modelo MACBOR ATV 49 mini, es el medio de desplazamiento. Dispone de un motor térmico de 50cc de consumo reducido. Necesita alimentación eléctrica para el motor de arranque. Los frenos han de ser programados para actuar en modo exclusivo frente al acelerador y viceversa. El control de acelerador está acoplado a un limitador de velocidad en el puño de gas y hay un mando a distancia que permite detener el vehículo desde lejos en caso de necesidad para un paro de emergencia durante las pruebas.

La dirección de desplazamiento del vehículo es controlada por la posición de las ruedas delanteras en el eje Z, es decir perpendicular a la superficie. El giro del vehículo se consigue gracias a un pistón controlado por un motor DC y acoplados con un mecanismo de tornillo sinfín. Hemos montado un

potenciómetro en el eje de la dirección y su lectura nos indicará la posición angular de las ruedas delanteras.

En esta aplicación, usaremos adquisición de datos por la lectura del potenciómetro, control de servomotores para freno y aceleración y por último, control de motor DC en función de la lectura analógica del potenciómetro. Con estas modificaciones junto a la creación de una librería para facilitar el uso de las funciones sobre los actuadores, daremos apoyo al control del vehículo realizado con control manual, semi-automatizado o totalmente automatizado y controlado por detección de obstáculos gracias a sensores y a un dispositivo de rastreo láser.

Nos planteamos cómo trabajos futuros la implementación de un regulador Pid para el control de giro del vehículo y la mejora del frenado por la aplicación de algoritmos para reducir la distancia de frenado.

3. Desarrollo práctico.

3.1. **Desarrollo del dispositivo genérico. Pic18f4550.**

Todos los microcontroladores que se fabrican en el presente tienen excelentes prestaciones pero el más adecuado no es el más servicios ofrecen. Según la aplicación y los requerimientos se ha de buscar en los modelos que ofrecen los varios fabricantes. Un diseñador ha de conocer todo el abanico que se oferta en el mercado de microcontroladores para elegir el adecuado para cada situación.

La elección suele hacerse en primer lugar por las funciones que ha de cumplir, descartando una parte que no interesa y por último en función del coste y facilidades de uso. En nuestro caso, para buscar en función de la aplicación se ha de tener en cuenta:

- La rapidez y precisión del procesamiento de datos
- El tipo y número de entradas y salidas
- El tipo y cantidad de memoria
- La precisión del ancho de palabra
- Entradas analógicas
- Comunicación serie
- Interrupciones.
- El tamaño del microcontrolador para el diseño de la placa.

Microcontroladores PIC

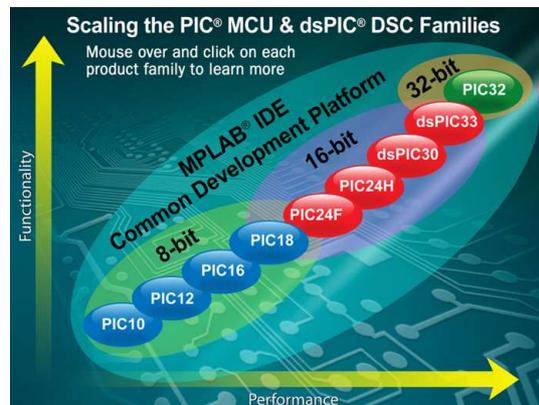


Figura 25 Familias de microcontroladores Pic

La popularidad de los PIC hace que sean incluidos en la mayoría de los proyectos debido a su bajo coste, reducido consumo, pequeño tamaño, de fácil programación y abundancia de herramientas económicas de soporte.

El Pic18F4550 se puede encontrar entre los microcontroladores de 8bits. Tiene una arquitectura Harvard y se caracteriza por 32kB de memoria para el programa y 2,2KB para los datos.

Dispone de un juego de instrucciones RISC. Tiene 40 pines entre cuales se encuentran los cinco puertos configurables como entrada o salida.

Destacamos la conectividad Usb v2.0, los 4 temporizadores y los 13 canales configurables como entrada analógica de 10 bits.

Estructura del PIC18f4550.

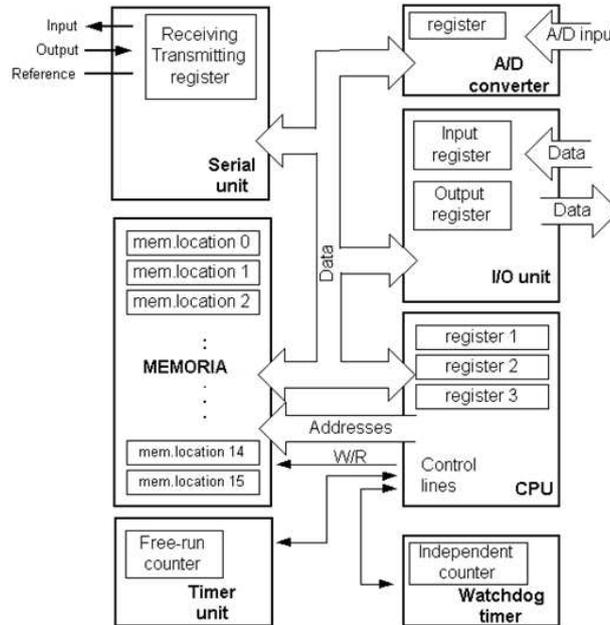


Figura 26 Bloques estructurales del Pic 18f4550

A continuación describiremos las funcionalidades que incorporaremos en nuestro proyecto.

Entrada / Salida

Los puertos Entrada/Salida los podemos configurar en función de cada aplicación teniendo en cuenta que los pines RC4 y RC5 serán de uso exclusivo para la comunicación Usb. En los ejemplos de aplicaciones prácticas usaremos, como norma general, el puerto B para configurar salidas generales y el puerto A para entradas analógicas o digitales.

Convertor Analógico Digital

Dispone de un convertor Analógico Digital de 10 bits. Hay 13 pines que se pueden configurar como entrada analógica, siempre en cascada, es decir si quieres seleccionar el pin AN2 cómo entrada analógica, obligatoriamente estarán seleccionados AN0 y AN1. Es un punto a tener en cuenta a la hora del diseño del dispositivo. La configuración se realiza a través de los registros ADCON.

Interrupciones y temporizador

El pic dispone de varios tipos de interrupciones y prioridades de las interrupciones. Las funciones definidas con prioridad baja interrumpirán cualquier ejecución menos las definidas con prioridad alta. Destacamos la interrupción Usb y la interrupción del Temporizador al vencer un periodo de tiempo que configuraremos cómo variable dentro de nuestras aplicaciones prácticas.

Las interrupciones se pueden habilitar, deshabilitar o configurar desde el registro INTCON.

Reloj

El microcontrolador necesita un reloj externo pero tenemos que configurar sus pulsos para obtener 48MHz necesarios para la comunicación Usb. A un oscilador de cuarzo de 12MHz se le debe multiplicar por 3 su frecuencia.

Desarrollo del dispositivo genérico.

El elemento principal del dispositivo es el microcontrolador.

Requerimientos para al microcontrolador

- **Comunicación Usb**

La necesidad de una comunicación rápida y fiable nos lleva a la utilización del puerto Usb.

- **Interrupciones**

Son necesarias para temporizar eventos. Imprescindible en el envío de Pwm.

- **Adquisición de datos:**

El microcontrolador tiene que disponer de entradas analógicas y conversión al digital. Esta parte es imprescindible para la lectura del potenciómetro situado en el eje de la dirección. Hay una familia de microcontroladores de alta resolución pero nos condiciona el factor económico, por lo tanto una conversión a un dato de 10 bits estables es más que suficiente.

- **Frecuencia de funcionamiento**

Debemos emplear un microcontrolador que funcione a alta frecuencia de reloj para asegurar una buena transmisión Usb combinada con la señal de salida para los actuadores y la lectura de los elementos analógicos.

- **Fácil programación**

Valoraremos un microcontrolador que tiene una fácil y rápida programación para hacer posible el desarrollo del programa interno (firmware) y la posterior grabación con la ayuda de herramientas económicas o gratuitas, ejemplos de uso y dispositivos de programación sencillos y económicos.

- **Memoria interna**

Es importante disponer de suficiente memoria para los datos y para el programa. Una restricción en este campo sería negativa ya que actuaría directamente sobre la parte genérica del dispositivo.

Elección del microcontrolador

De todos los microcontroladores nos hemos centrado en los que llevan preparada le comunicación Usb. Hemos encontrado los microcontroladores Pic 18fx550 de la casa Microchip. Más concretamente pic 18f2550 con 28 pines y 18f4550 de 40 pines.

La tabla comparativa de especificaciones nos aclara la elección y la diferencia de precio es desestimable.

Device	Program Memory		Data Memory		I/O	10-Bit A/D (ch)	CCP/ECCP (PWM)	SPP	MSSP		EAUSART	Comparators	Timers 8/16-Bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)					SPI	Master I ² C™			
PIC18F2455	24K	12288	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F2550	32K	16384	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F4455	24K	12288	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3
PIC18F4550	32K	16384	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3

Tabla 8 Comparativa de prestaciones de Pic con conectividad Usb

Hemos elegido pic18f4550 por todas las extras que el pic18f2550 no ofrece, mirando a realizar un dispositivo genérico de lo más versátil.(más canales A/D y más puertos de E/S)

Requerimientos del dispositivo

Diagrama funcional por bloques del dispositivo.

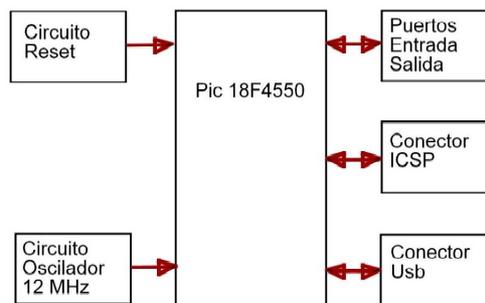


Figura 27 Bloques funcionales del dispositivo genérico

A continuación describiremos los circuitos adicionales para el dispositivo genérico

Circuito del reset

El reset del microcontrolador, el pin MCLR, se activa a nivel bajo, por lo tanto hemos de conectarlo a 5V fijos. Para poder resetear de forma manual el microcontrolador añadiremos un circuito externo que al pulsar un pulsador se conecte a 0V el pin del reset. Cabe destacar que para la grabación del microcontrolador, este pin se debe conectar a 12V para que el microcontrolador pase a modo grabación. El pin está conectado al puerto ICSP y de poner 12V en el pin se encargará el programador.

Circuito del reloj

El circuito externo del reloj se compone de un oscilador de cuarzo de 12MHz puesto en paralelo con los pines OSC1 y OSC2. cada uno de estos dos pines está en serie con un condensador de 22pF a la masa del circuito.

Comunicación Usb

Los pines RC5 y RC4 están conectados a data+ y data- respectivamente del microcontrolador. En serie con el Pin18 VUsb del microcontrolador conectamos en serie un condensador de 470uF a la masa del circuito.

Circuito de programación:

ICSP, In Circuit Serial Programming, es el circuito diseñado en esta dirección y la mayoría de los microcontroladores lo incorpora. Al aplicar 12V en el pin Reset del microcontrolador, este pasa a modo programación conmutando el pin RB6 como reloj y RB7 para datos.

Puertos E/S

Para los puertos hemos de tener en cuenta no superar las especificaciones de cada pin. Pueden ser configurados como entrada digital, entrada analógica o salida digital.

En especial para este microcontrolador no se pueden superar los 25mA por pin. La estructura electrónica del dispositivo es simple: el conector ICSP es necesario para la programación del microcontrolador (escribir el firmware) y el circuito del reset para reiniciar el microcontrolador. Los puertos de entrada/salida son configurables según la aplicación.

El conector USB será de tipo B, hembra con los pines conectados según las indicaciones del fabricante del microcontrolador. El controlador será montado sobre un zócalo, para evitar daños a la hora de soldar, teniendo la posibilidad de su extracción para programación y depuración en un programador externo, además de poderlo fácilmente reemplazar si es necesario. Dispondrá de un oscilador de cuarzo que se configurará a la frecuencia necesaria para la comunicación usb. En nuestro caso hemos usado de oscilador de cuarzo de 12000Hz, configurando el microcontrolador para que funcione a 48000Hz . Es aconsejable dotar el dispositivo de un botón de reset para su reinicio. Hemos configurado los puertos como entrada/salida, analógicos o digitales en función de las necesidades de actuación. Por ultimo, hemos dotado el dispositivo de un puerto ICSP para programación del microcontrolador.

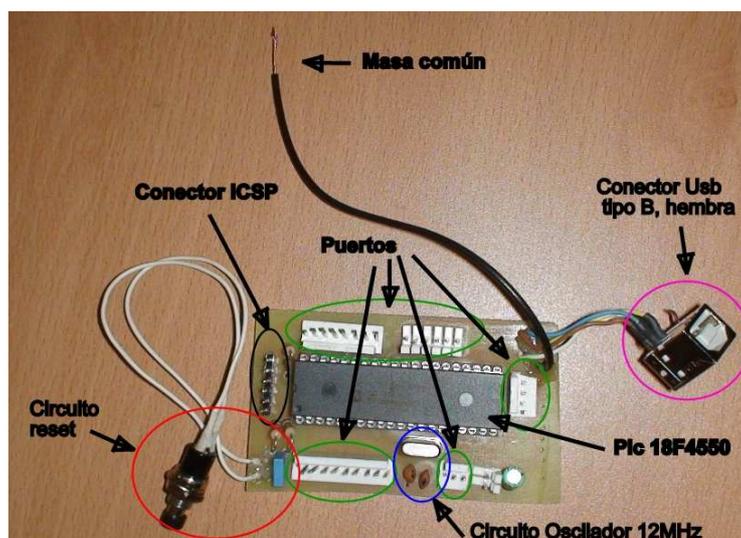


Figura 28 Componentes del dispositivo genérico

Diseño Hardware

El esquema funcional lo hemos diseñado para conseguir un dispositivo genérico. En esta dirección hemos reducido el abanico de posibilidades que ofrece el microcontrolador, implementando las imprescindibles para su programación, conectividad al Usb, reinicio, oscilador externo y los puertos de entrada salida, dejando la configuración de estos últimos condicionados por el periférico a controlar.

Hemos añadido un puerto extra, ICSP, para grabar el microcontrolador sin extraerlo de la placa.

El circuito de Reset va conectado al pin MCLR del microcontrolador, activo a nivel bajo. Por lo tanto, el interruptor es normal abierto que, al cerrar, pone un "0" lógico en el Mclr. El condensador de este circuito sirve para absorber los parásitos causados por el efecto "rebote" del interruptor.

El circuito ICSP va conectado según las indicaciones In Circuit Serial Programming. El Microcontrolador pasa al estado de grabación al recibir 12V en el pin Mclr, correspondiente al pin 1 del Icspp. Los pines del Puerto B (RB7 y RB6, correspondientes a pin 6 y 5 del Icspp) son utilizados para la grabación, siendo conmutados como entrada para la escritura y salida para la lectura. El Pin RB5 del Pic correspondiente al pin4 del Icspp se utiliza solamente para la depuración. Los pines 2 y 3 del Icspp son la alimentación del Pic, Vdd y Vss, es decir 5V y masa. El programador PicKit2 da la opción de elegir la fuente de alimentación si detecta el microcontrolador alimentado. Si no, lo alimenta el.

El circuito oscilador está compuesto por un cristal de cuarzo de 12Mhz conectado a los pines osc1 y osc2 del microcontrolador, cada pin en serie con un condensador de 22pf a la masa del circuito.

Los puertos de entrada/salida están son conectores directos con los pines de los puertos del microcontrolador.

Hemos conectado un condensador de 470uF al Pin Vusb del Pic con la masa del circuito según las indicaciones del fabricante de microcontrolador para absorber los posibles picos de tensión.

Hemos dotado con un conector para extender la alimentación de partes auxiliares como leds, potenciómetros o acondicionar como voltaje de referencia.

Teniendo en cuenta que el circuito está alimentado del puerto usb del Pc, todo lo que se conecta a este pin no debe consumir más de la diferencia de la corriente suministrada por el puerto usb y el consumo del Pic. El consumo del Pic variará en función de los elementos conectados a sus puertos, siendo la suma de todos ellos más los de mantenimiento en marcha del Pic.

Conectividad con actuadores

Descripción.

Los periféricos son dispositivos remotos que realizan una tarea específica cómo encender una luz, mover un objeto, etc. Están compuestos por una parte electrónica de adaptación de la señal y otra parte actuadora.

Entendemos por actuador los dispositivos encargados de realizar un trabajo en función de los datos recibidos de un controlador. Pueden ser desde simples indicadores (leds, bcd) hasta otros dispositivos con los que se puede establecer una comunicación. Algunos de ellos están preparados para la conexión directa al microcontrolador, es decir tecnología TTL (transistor – transistor Logic) como son los servomotores de modelismo, pero la mayoría carecen de estas características y se deben acondicionar. Mejor dicho, la conexión directa con los actuadores no es aconsejable excepto en los montajes en los que las especificaciones de los actuadores son compatibles con las de los puertos de salida del microcontrolador (Corriente máxima, diferencia de potencial, etc). En caso de no cumplir estos requisitos, se ha de diseñar y realizar un circuito electrónico para realizar esta tarea. Estos circuitos son específicos para cada tipo de actuador.

Circuito electrónico acondicionador de señal

Este circuito es un circuito acondicionador de señal parte integrante del periférico. Debe disponer de componentes para proteger el microcontrolador y los actuadores, siendo recomendable el uso de los opto acopladores.

Se ha de tener en cuenta que la masa del microcontrolador debe ser común con la del actuador.

Realizar la Placa

Hemos optado por la realización artesanal de la placa, Hemos marcado con rotulador permanente el pcb calcado sobre la placa virgen de fibra-cobre, hechos los agujeros para los componentes y repasados con rotulador. Hemos preparado el líquido para la corrosión, constituido por 50% sulfamant, 20% agua oxigenada 80 volúmenes y 30% agua, a una temperatura de 25 – 30 gradosC. Luego hemos introducido la placa en ácido para la corrosión. La corrosión dura unos aproximativos 20 minutos. Al finalizar la corrosión, lavamos con agua la placa, la limpiamos con alcohol de quemar para eliminar los residuos del rotulador, la secamos y hemos repasado las rutas con estaño. Por ultimo hemos comprobando la continuidad en las rutas.

Hemos montado los componentes soldando desde los que están más pegados a la placa hasta los más altos, revisado el conexionado y las rutas.

Comprobación.

La comprobación es necesaria para descartar fallos de construcción. Se repararán punto por punto para una comprobación exhaustiva de las conexiones.

Grabación

Una vez comprobado el buen conexionado de los componentes se procede a la grabación del microcontrolador.

Para la grabación hemos usado el conjunto PicKit2:



Figura 29 Programador PICkit2

PicKit2 graba a través del puerto ICSP implementado en el dispositivo. PicKit2 Es un programador que se conecta al puerto Usb del Pc. Es rápido y ofrece opciones de configuración de la grabación, el tipo de accesos futuros al Firmware (enable protected code). También gestiona la alimentación del microcontrolador si este no estuviese conectado a la fuente.

El programa de grabación importa el archivo del firmware “.hex” lo envía a través del Usb al programador y este es el encargado de escribir el microcontrolador. Dispone de opciones como la lectura del firmware en el microcontrolador, borrar el firmware existente, verificación de la escritura, configurar el tipo de restricciones de acceso al firmware o a la memoria, entre otras. Es gratuito pero no compatible con otros tipos de programadores. Otra alternativa para la grabación del Pic es el entorno de programación mplab que dispone de una función que reconoce el programador (reconoce varios programadores), graba y verifica si se ha realizado con éxito. Además dispone de varias funciones como depuración en tiempo real y visualización del estado de los registros del pic.

Hemos diseñado el firmware para transmitir los datos de control al periférico implementando funciones que actúan directamente sobre los puertos. Hemos elegido el protocolo necesario para la comunicación con el periférico en función del tipo de actuador. Este protocolo será el encargado de encapsular los datos recibidos a través del USB y enviarlos a través de sus puertos, o bien, manejar los puertos en función de los datos recibidos, siendo el otro extremo de la parte lógica del sistema objeto de este proyecto.

Conectividad al PC

Como cualquier dispositivo, este ha de ser reconocido por el sistema operativo. Para ellos hemos editado el archivo de información “.inf” incluyendo información específica del dispositivo.

Los datos más relevantes de este archivo son los datos identificativos del dispositivo PID (Product ID)y VID (Vendor ID), la ruta de los archivos para la instalación, el driver “mchpusb.sys” en nuestro caso y los nombres de fabricante, software y del dispositivo, “Strings”.

3.2. *Desarrollo de aplicaciones de control y monitorización desde PC.*

En este apartado describiremos una aplicación genérica compuesta por dos módulos, Un módulo estará instalado en el ordenador personal para el control general del sistema y el segundo modulo se instalará en el microcontrolador. Los dos módulos han de estar comunicados continuamente.

Esta aplicación es genérica y constituye una plantilla para la implementación del control del sistema, pudiendo ser modificada en función del tipo de sistema en el que se va a aplicar.

Para el desarrollo de la aplicación hemos optado por avanzar con pasos pequeños pero seguros. Antes del diseño del firmware hemos implementado la comunicación usb. Una vez reconocido el dispositivo Usb e instalado, el programa residente en el ordenador hace una petición de la versión del firmware del microcontrolador y este la devuelve. Conseguida la comprobación, es decir la comunicación Usb, se ha procede al desarrollo de las demás funciones de cada módulo, primero haciendo una transferencia de datos como valores para actuadores y el manejo de ellos. Por último se diseña e implementa las demás funciones, Estas tareas se realizan en paralelo, es decir para el firmware y para el software.

Diseño del firmware.

El Firmware (entendiendo software del microcontrolador) lo hemos diseñado basándonos en las necesidades del sistema:

- funciones para la manipulación de los datos recibidos a través del Usb y el correspondiente tratamiento según la aplicación.
- funciones para el manejo de los actuadores: envío de señales o protocolo .
- funciones para la adquisición de datos.
- Funciones de cálculo adicionales

Hemos usado las siguientes funciones propias del microcontrolador:

- Apertura y uso de la comunicación Usb
- Configuración e inicialización de las interrupciones para el manejo de los actuadores.
- Configuración e inicialización de los puertos del microcontrolador

Mostramos a continuación una vista general sobre el planteamiento del firmware.

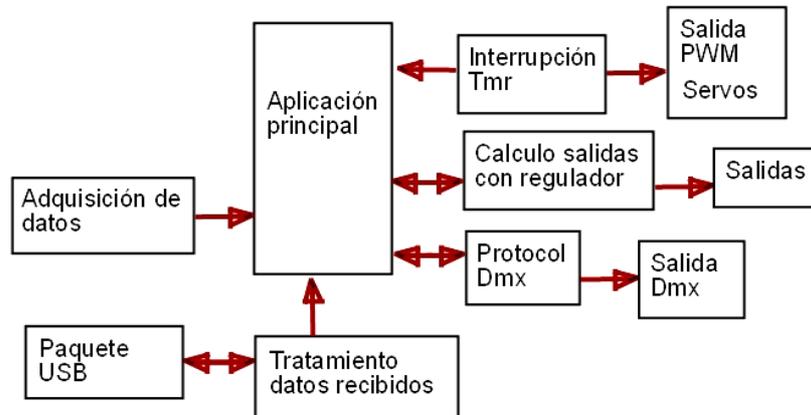


Figura 30 Módulos del firmware y la intercomunicación.

Los datos procesados o recibidos por Usb son enviados directamente a los puertos. Hemos elegido el protocolo necesario para la comunicación con el periférico en función del tipo de actuador. Este protocolo será el encargado de encapsular los datos recibidos a través del USB y enviarlos a través de los puertos del microcontrolador configurados para la tarea, o bien, manejar los puertos en función de los datos recibidos.

El microcontrolador utilizado, Pic18f4550, dispone de un puerto de salida pwm pero para aplicaciones que necesitan controlar por pwm más de un actuador queda insuficiente. Por lo tanto hemos de implementar una función para generar pwm en cualquier pin se necesite. Lo más práctico es el uso de interrupción para temporizar los ciclos Pwm combinado con el uso de máscaras para manejar un puerto por operación.

Elegir el lenguaje y el entorno de programación

Hemos optado por el lenguaje C dentro del entorno de programación MpLab de la casa Microchip, gratuito. Hemos compilado con C18 Student Edition, también gratuito, de la misma casa. Constituyen unas excelentes herramientas que cuentan además con un buen soporte de aprendizaje y ayuda. Ofrece la posibilidad de visualizar los datos de los registros o las variables. Dispone de archivos predefinidos con la definición registros y datos frecuentemente usados para cada uno de los microcontroladores Pic que fabrica o ha fabricado Microchip.

Edición del Firmware

Como punto de partida hemos configurado el microcontrolador con los valores que hemos especificado en el archivo de información .inf. para poder ser reconocido por el sistema operativo. Luego hemos especificado el tamaño del paquete a enviar, recibir.

Según los datos que se quieren manejar a través del Usb, se define el tamaño del paquete y se configura byte a byte la destinación de cada uno. El máximo

de datos que un paquete usb puede manejar es de 64Bytes. Este paquete lo configuramos de la siguiente manera:

Para las aplicaciones prácticas variaremos el tamaño del paquete y los valores de encabezamiento en función de las necesidades.

A continuación hemos creado la función de recibir datos y el tratamiento de las mismas. Para poder ser enviados a los periféricos, hemos configurado los puertos de entrada/salida. Las interrupciones las hemos inicializado para dar paso al control sincronizado del periférico o del envío de datos, enviando según protocolo. Llegados a este punto hemos de implementar mecanismo de acceso a datos a través de semáforos para evitar interbloqueos o lectura de datos erróneas. Suele ocurrir cuando se dispara una interrupción vencimiento del reloj y se interrumpe un proceso de recepción de datos. En el peor de los casos se produciría una excepción al no poder acceder a un dato y derivaría en el reset del microcontrolador. Por último, hemos creado el bucle principal para atender de forma ordenada las tareas.

Archivos de instalación

Los archivos de instalación constituyen el enlace entre los dos medios lógicos, firmware y software. Estos archivos son utilizados por el sistema operativo para reconocer y configurar el dispositivo. También son utilizados por el software a través de llamadas a las funciones de la librería específica para Usb facilitada por el fabricante.

El archivo de información sobre el dispositivo tiene la extensión “.inf”. En el especificaremos:

Vendor ID y Provider ID que deben ser los mismos especificados tanto en la clase UsbDll del software como en usbdsc.c del firmware.

Las rutas del driver mchpusb.sys (si no se especifica la hemos de indicar manualmente)

Los Strings de los nombres que muestra el dispositivo.

Los archivos necesarios tienen las extensiones: inf, .sys

Diseño del software

Diagrama de funcionalidades de la aplicación

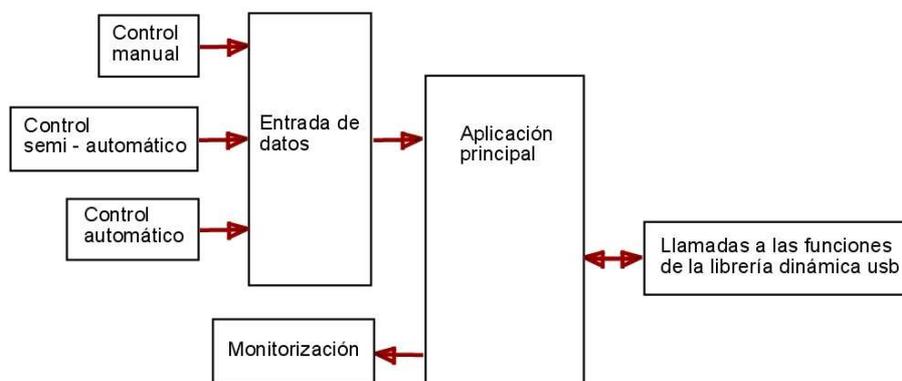


Figura 31 Principales funcionalidades de la aplicación software

Hemos dotado el programa de las siguientes funcionalidades:

Parte de control: Distinguimos aquí tres tipos:

1. Control manual
 - 1.1. Los propios controles de la ventana de la aplicación. El usuario introduce datos en los controles de la ventana del programa. Estos pueden ser cuadros de texto, listas editables, botones, deslizadores, etc.
 - 1.2. Medios físicos de control. El usuario usa el joystick, el teclado, el ratón u otros periféricos para facilitar los datos a enviar, la aplicación siendo la encargada de capturar estos datos
2. Control semi-automático
 - 2.1. Los datos son obtenidos de otros dispositivos o del mismo periférico a controlar. Se trata de realimentación, la intervención del usuario es mínima pero decisiva, definiendo las preferencias del sistema, que tipo de algoritmo se aplicará, cuando se interviene, etc.
3. Control automático
 - 3.1. Datos obtenidos de ficheros. El usuario especifica uno o varios archivos de dónde se leerán los datos para el control del periférico

Comunicación Usb: Hemos utilizado la librería "mpusbapi.dll" facilitada por el fabricante del microcontrolador. Para el envío / recepción de datos llamaremos a las funciones específicas para acceder al hardware, es decir al dispositivo. Monitorización: Para monitorizar, estados, posiciones, etc. hemos dotado el software de funciones que muestran en pantalla los datos de los actuadores.

Elegir el lenguaje de programación y la herramienta de edición:

El lenguaje C# es un lenguaje que nos ofrece todas las facilidades para una rápida programación, siendo un lenguaje intuitivo con una buena base de documentación. Los programas creados con este lenguaje corren sobre una máquina virtual. Para ejecutar nuestro programa es suficiente tener instalada NetFrame versión 2.0. Para aplicaciones más complejas se recomienda instalar también la versión 3.5.

La herramienta de edición es el Visual C# 2008 Express Edition, disponible en la página de Microsoft, junto a la documentación MSDN. Ambas son gratuitas. Dispone de buenos ejemplos cómo apoyo al programador. El entorno es muy amigable, intuitivo, el editor dispone de muchas facilidades y herramientas de programación y depuración.

Organización del trabajo de edición.

En primer lugar hemos de crear una clase que importa la librería mpusbapi.dll. Ahora se podrá acceder al hardware con una sola llamada. Necesitamos en segundo lugar una clase para mostrar los datos a enviar y los datos actuales del actuador. Además hemos de crear varias clase específicas para cada tipo de adquisición, procesamiento o almacenamiento de datos. Punto seguido crearemos varias clases para encapsular la estructura de datos y por último crearemos la clase principal que dirige el flujo de datos.

Software

Estructura lógica para el software

Siguiendo los pasos descritos en la primera parte del proyecto:

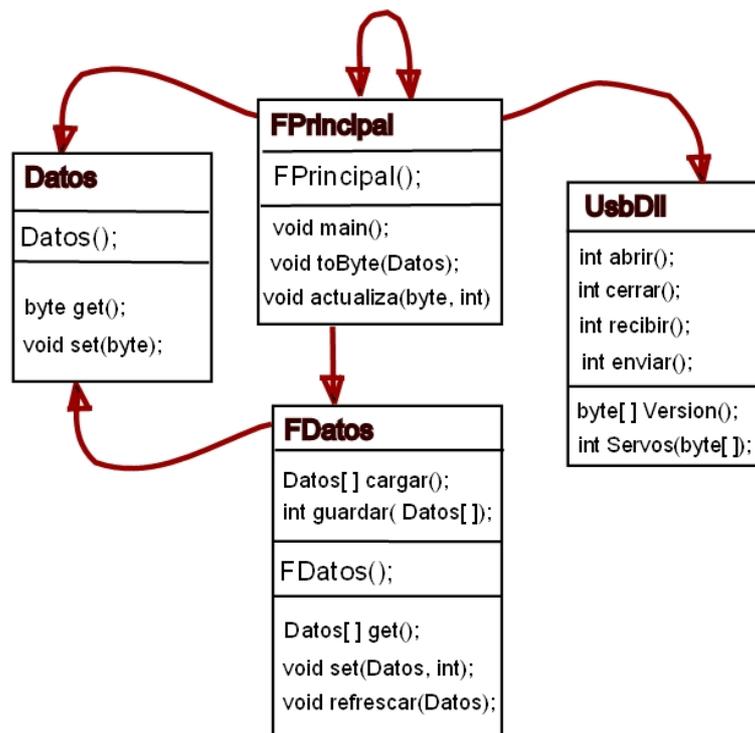


Figura 32 Clases de la aplicación

UsbDll es la clase que importa la librería "mpusbapi.dll".

Se importan las funciones de la librería "mpusbapi.dll".

En el constructor se Inicializa los String Vid, Pid y las tuberías para la comunicación Usb.

Métodos privados:

int abrir(): abre la tubería de comunicación, devuelve "0" si la operación de ha ejecutado con éxito, "-1" en caso contrario.

int cerrar(): cierra la tubería de comunicación, devuelve "0" si la operación de ha ejecutado con éxito, "-1" en caso contrario.

int enviar(): envía los datos llamando a la función _MPUSBWrite de la dll, devuelve "0" si la operación de ha ejecutado con éxito, "-1" en caso contrario.

int recibir(): recibe los datos llamando a la función _MPUSBRead de la dll, devuelve "0" si la operación de ha ejecutado con éxito, "-1" en caso contrario.

Métodos públicos:

byte[] Versión(): recibe la versión de la firmware

int Servos(byte[]) prepara el paquete de datos para los actuadores tipo servomotores y llama al método enviar(). devuelve "0" si la operación de ha ejecutado con éxito, "-1" en caso contrario.

FDatos es la clase encargada de facilitar los datos, actualizarlos y mostrarlos.

El constructor inicializa la lista de Datos y las variables globales.

Tiene métodos privados para abrir ficheros en modo lectura o escritura: cargar() y guardar().

Métodos públicos:

get() devuelve la lista de datos a enviar.

set(Datos, int): actualiza los Datos de la posición especificada en int.

refrescar(Datos) muestra los Datos en pantalla.

Datos es la clase que almacena la información disponiendo de constructor que inicializa los datos, y métodos públicos de get que devuelve el dato en byte y set que actualiza el dato.

FPrincipal es la clase que arranca la aplicación. En el método main llama a su constructor. El constructor inicializa la estructura de datos, lanza la ventana de adquisición y visualización de datos e inicializa la comunicación obteniendo la versión del firmware tras la llamada `usbDll.Version()` y mostrando en pantalla el resultado. Esta clase dispone de botones que implementan métodos para controlar el flujo de datos.

3.3. *Desarrollo de aplicaciones de control embebido.*

A lo largo de la historia hemos podido ver una búsqueda continua para mejorar la vida cotidiana en todos los aspectos. Lo que en un principio era una tarea de un largo periodo de tiempo, hoy en día se limita a un simple parpadeo. Es el ejemplo de los cálculos, la escritura y envío de documentos. También hemos visto cómo las distancias se han reducido y 3000 km se pueden recorrer en unas tres horas. La tecnología se ha impuesto hasta en los campos más remotos de nuestra civilización y seguirá haciéndolo por mucho tiempo.

Las aplicaciones de control embebido son una pequeña parte de el mecanismo de control que ayuda al desarrollo de proyectos que requieren automatismo, cálculo, comunicación y rápida respuesta, convirtiéndose en partes integrantes de estos proyectos,

Los campos de aplicación son de los más variados, desde de máquinas de juego hasta maquinaria pesada, desde una máquina expendedora de refrescos hasta los aparatos médicos más precisos.

Las aplicaciones prácticas que plantearemos tienen el fin de comprobar si realmente se ha conseguido realizar un dispositivo genérico y versátil.

Abordaremos unas aplicaciones prácticas para emplear varios tipos de los actuadores y sus respectivos protocolos de control. Para cada una de la aplicación seguiremos los siguientes pasos:

Estudio sobre los periféricos y los actuadores.

Se ha de tener en cuenta que tipo de protocolo se ha de emplear, las especificaciones físicas y el acondicionador de señal.

Estimación de recursos hardware necesarios.

Diseño del paquete para intercambio de datos por Usb

Implementación de la aplicación de control con los cambios necesarios en la plantilla.

En algún ejemplo cubriremos más áreas de las establecidas, cómo el diseño de mecanismos para los autómatas mecánicos. Generalmente nos vamos a centrar en la configuración del sistema genérico para el control de los actuadores incorporados. Por otra parte, a pesar de la posibilidad del control de un número elevado de actuadores, nos centraremos en el tipo de actuador y en su control. El límite máximo de cantidad y las aplicaciones de uso es objeto de un posible trabajo futuro.

3.3.1. Control de autómatas mecánicos.

La avance tecnológico y los medios de comunicación modernos han quitado protagonismo a los eventos sociales como el teatro y otras disciplinas artesanales antiguas. Aún se encuentran entidades que organizan festivales, muestras y encuentros teatrales en los cual maestros artesanos y artistas exponen sus trabajos, algunos a la vieja usanza y otros más contemporáneos. Pero al ser una actividad con pocos beneficios y difícil de mantener, cada vez se organiza menos y algunos trabajos que pueden funcionar con poco mantenimiento como los teatros de autómatas han quedado en el olvido. Pero, por suerte, de vez en cuando aparece una persona interesada en este arte y al encontrarse con antigüedades de este mundillo, desea rescatarlas. Se han recuperado auténticas joyas de maquinaria bajo forma de autómatas mecánicos que representan escenas de obras teatrales. Otras han sido remodeladas, cambiando los mecanismos por los actores y los manipuladores como es el caso de “El belen de Tirisiti” que se expone en fechas de Navidad en Alcoi, Comunidad Valenciana. Esta primera aplicación práctica rinde un homenaje a los constructores anónimos de estas obras que combina la ingeniería con el arte.

Autómatas mecánicas controladas por ordenador.

Hemos diseñado y construido un juego de cuatro marionetas interpretando personajes de músicos de una banda.

Hemos reunido un pianista, un violinista, un guitarrista y un batería. Hemos configurado el sistema para que el ordenador interprete el papel de director y manipulador a la vez.

Los autómatas mecánicos los hemos previsto de un sistema complejo de mecanismos para reducir a dos el número de servomotores para el movimiento de cada uno de los músicos y conseguir al mismo tiempo mayor atracción por el cuadro que ofrece el conjunto de muchas piezas mecánicas en movimiento a la vista, sugiriendo la reproducción de los gestos de los músicos.

3.3.1.1. Sistema Mecánico.

Diseño y construcción de los autómatas mecánicos.

Hemos diseñado los mecanismos utilizando el potente software de animación 3d Blender. Este programa gratuito dispone de un amplio abanico de posibilidades de edición pero nosotros hemos utilizado la parte de animación de “Armature”.

Se trata de objetos unidos entre si que se configuran para realizar animaciones complejas. Disponen de varias formas de generar las animaciones, entre ellas la cinemática inversa. La animación generada nos sirve a cuenta de simulación.

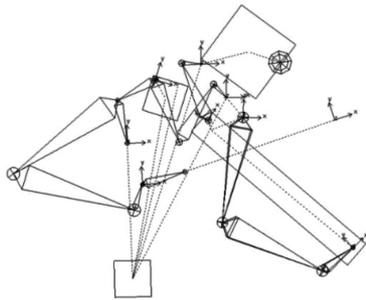


Figura 33 El violinista – autómeta electromecánico

Una vez diseñados los mecanismos y comprobada su funcionalidad, hemos impreso en papel el modelo y lo hemos calcado sobre planchas de madera para recortar las piezas. Hemos ejecutado los agujeros para el ensamblaje, ensamblado, comprobar el movimiento y por ultimo montar los servos en el lugar adecuado.

3.3.1.2. Sistemas de Actuación.

Servos

Un servomotor es un sistema compuesto por un motor eléctrico con capacidad de ser controlado, tanto en velocidad como en posición, generalmente controlado por ancho de pulso (pwm).

Un servomotor dispone de un circuito electrónico. Puede ser analógico o digital. El circuito electrónico analógico gobierna el motor DC incorporado en función del resultado de la comparación del voltaje resultado de la conversión de la señal Pwm recibida y el voltaje del potenciómetro incorporado, conectado físicamente al motor a través de un sistema reductor de engranajes..

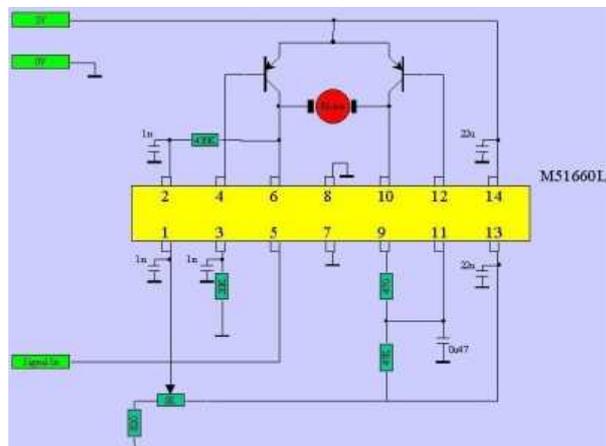
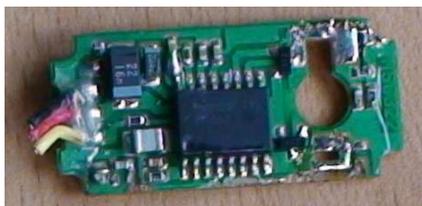


Figura 34 Figura circuito servo

Aplicando nuestro sistema genérico, hemos definido cómo periférico a cada músico, y los dos servos de cada músico que realiza el movimiento siendo los actuadores.

Los servos disponen de circuito propio de acondicionamiento de señal funcionando con PWM y tecnología TTL, por lo tanto compatible con nuestro microcontrolador.

El microcontrolador dispone de un pin por el que, configurado para la tarea, envía señal Pwm. En nuestro caso necesitamos más de un servo a controlar por lo que hemos de implementar salida de señal Pwm.

Por lo tanto, hemos hecho las siguientes modificaciones:

Firmware:

La interrupción provocada por el temporizador, preparada en la parte general de la implementación de la firmware, nos sirve para llamar a la nuestra función de manejo de puertos. Hemos hecho un estudio para optimizar los recursos del microcontrolador. Como el microcontrolador dispone de suficiente memoria para guardar 180 valores por puerto en los que guardaremos el estado de los puertos. Estas Máscaras sirven para tener acceso inmediato a un valor a través de una tabla y no perder tiempo con desplazamiento de bit para actualizar pin por pin. La interrupción funcionará según el siguiente algoritmo

Por último desactivamos la entrada analógica dejando todo el tiempo del microprocesador a las tareas de la comunicación y el manejo de puertos para transmisión del pwm al recibir la interrupción.

Software

El software también sufre modificaciones en la parte de estructura de datos ya que necesitaremos almacenar la posición de cada servo en cada momento.

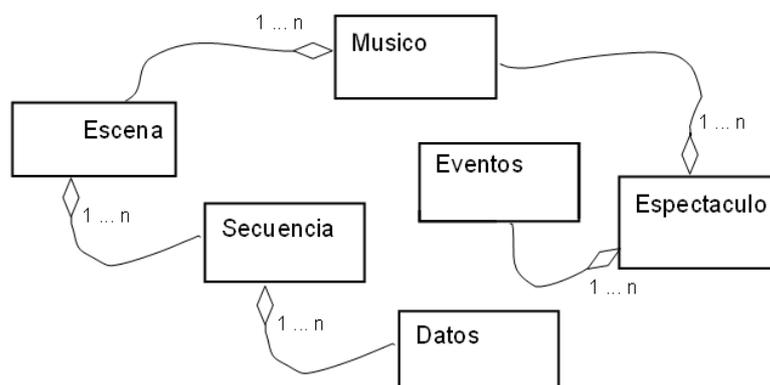


Figura 36 Diagrama de clases control autómatas

Definiremos varios conceptos para organizar los datos.

El espectáculo tiene uno o más músicos. Un músico tiene sus escenas que pueden tener una o más repeticiones. Las escenas tienen secuencias. Una secuencia guarda los datos, es decir las posiciones de los servos de cada músico. También cada secuencia tiene un tiempo de espera antes de enviar el

dato y una velocidad de reproducción. Si el tiempo de espera es cero, la secuencia no llega a reproducirse, de lo contrario, se enlazaría todas las secuencias de todas las escenas hasta el final de la programación o hasta que se encuentra el tiempo de espera cero en alguna de las secuencias del músico. Por otra parte, el Espectáculo tiene uno o más eventos. Los eventos son necesarios para sincronizar los movimientos en el tiempo. Un evento almacena el momento del disparo de una secuencia de una escena de un músico. El último evento de un espectáculo debe ser un evento especial llamado "Parada" para finalizar todos los movimientos.

Para conseguir suavidad en el movimiento, la diferencia entre la posición actual y la posición deseada se divide en varios pasos que se ejecutarán en intervalos equidistantes. Tenemos previstos para un trabajo futuro el diseño de un algoritmo para el manejo de curvas de aceleración y de frenado del movimiento de los servos al empezar y finalizar el movimiento.

Calibrado de los servos:

Es un importante paso. No todos los mecanismos pueden cumplir con un recorrido completo de 180 grados. Hemos de posicionar cada servo en sus posiciones límite y configurar las conversiones oportunas por software. Para ello hemos dispuesto un control específico para cada músico que posiciona el servo en el límite y almacena este valor. Para una mayor seguridad de no sobrepasar este límite, hemos creado una función de comprobación de estos límites en el firmware.

Programación de los autómatas mecánicos

Hemos hecho una programación de prueba para ver el conjunto en acción. Nos hemos dado cuenta que es una programación muy precisa a cuenta de una tediosa programación. A pesar de unos resultados estupendos, el tiempo de programación sigue siendo elevado. Para mejorar el tiempo de programación hemos pensado en recopilar datos de un archivo audio, convirtiéndolos mediante algoritmo adecuado en las posiciones idóneas para cada movimiento de cada músico. Hemos creado una clase que abre un archivo de audio ".wav", lee todos los valores, extraído el más significativo teniendo en cuenta que el servo mueve 60 grados en 0,8 segundos a 5V de alimentación. A partir de estos datos hemos creado un algoritmo de conversión de datos con el objetivo que el músico se mueva con suavidad y sin interrupciones, salvo las de efectos especiales deseados. (pausas de música o momentos de tensión) consiguiendo una programación compleja en pocos segundos. Hemos ampliado a la lectura y extracción de datos de archivos midi. Planteamos como trabajo futuro la interpretación de partituras musicales.

3.3.2. Control de proyectores de iluminación.

Una gran importancia en la evolución del teatro lo ha tenido la iluminación. En las antiguas obras el fuego iluminaba los rostros de los actores, creando dinamismo y expresiones que potenciaban las escenas.

Ha sido un paso grande cuando se ha comenzado posicionar mejor estos puntos de luz y mucho más cuando se ha conseguido controlar la intensidad de la luz. La introducción de los filtros de colores ambientaba las escenas y la combinación de todos estos elementos hizo que los espectáculos de teatro atraigan cada vez más. Esta técnica fue empleada en la televisión y mejorada con el tiempo.

Actualmente, el trabajo de iluminador cubre tanto el diseño de la iluminación como el montaje y la programación de las escenas.

Si antiguamente el nivel de la llama ambientaba una escena, hoy lo hacen mesas de iluminación sofisticadas, con ordenador personal incluido. El protocolo específico se ha desarrollado para unificar el trabajo de los constructores de dimmers, es decir los controladores físicos de los proyectores. Siguiendo cerca del espectáculo, nos hemos planteado cómo segunda aplicación práctica la configuración del sistema para gestionar la iluminación con varios proyectores conectados a un dimmer estándar DMX. El sistema implementará el protocolo con limitación de 6 canales. Planteamos la implementación de varios universos de 512 canales como trabajo futuro.

3.3.2.1. Sistemas de Actuación.

Actuador Dmx

Para el envío de señal Dmx es necesaria un circuito acondicionador de señal. Lo hemos basado en el componente electrónico balanceador de señal SN75176

Para una mejora de la lectura del nivel alto o bajo en la señal se usa un balanceador que prácticamente dobla el valor de la diferencia de potencial, es decir para 5V, en Data+ habrá +5V y en Data- -5V, mientras que un nivel cero pondrá tanto en Data+ como en Data- a 0V.

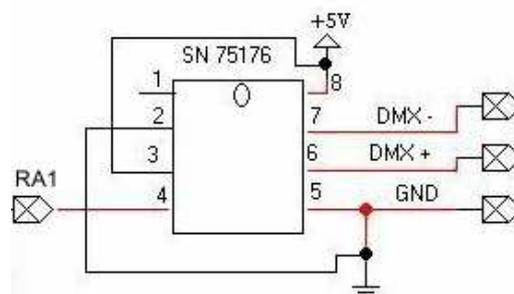


Figura 37 Esquema balanceador de señal para Dmx

El actuador Dmx es un dimmer de potencia, cerca de 2200w por canal, para el control de la iluminación. Principalmente está basado en un triac y un detector de paso por cero para sincronizar el cierre de la puerta del triac. La señal Dmx recibida está procesada y los valores seccionados excita la puerta del triac correspondiente a cada canal. El triac está cortando la onda sinusoidal variando así el tiempo de on del corriente. La lámpara estará encendida solamente el tiempo de apertura del triac, La corriente eléctrica en España es de 50 Hz, 25 ciclos positivos y 25 negativos – es triac funciona igual para los dos tipos de ciclo – un ciclo positivo o negativo es de 10ms, es decir que un valor de 50% de encendido de una lámpara es cuando el triac permanece abierto 5ms por ciclo contados a partir de la detección del cruce por cero.

Dispone de optoacopladores para aislar el circuito de potencia del de control. Hay dimmers que aún conservan el puerto analógico, y, en el momento de conectar una mesa de luces analógica, pasan a la función analógica y las puertas de los triac están gestionadas por el circuito RC de la mesa. (el modo más sencillo de regular la iluminación)

Los dimmers son aparatos que:

- regulan la intensidad de los proyectores
- tienen memoria para almacenar los niveles de intensidad
- reciben valores de intensidad de la controladora y los memoriza

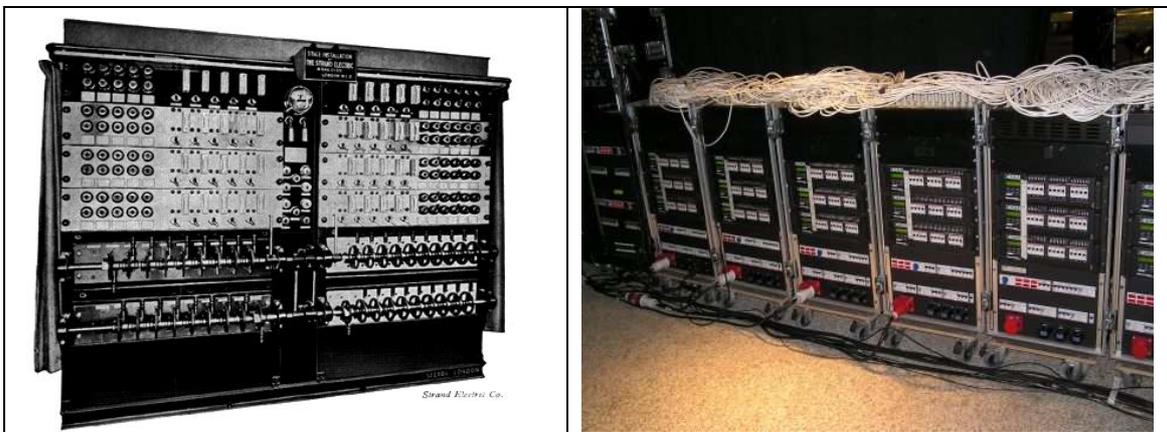


Figura 38 Los dimmers: antes y ahora.

El dimmer



Figura 39 Dimmer: exterior, interior y un solo canal.

Componentes principales: puerto analógico, puerto DMX in puerto DMX out fusibles de protección de fase y por canal (hay dimmers que tienen magneto-térmicos que sustituyen los fusibles), filtros antiparasitarios, modulo de control, modulo test.

3.3.2.2. Sistema de Alimentación.

Incorporado en el mismo dimmer: alimentación del control (5V, 250mA) y alimentación de potencia (230V – 380V y 2200W/canal. Al usar alimentación trifásica, los dimmers actuales incorporan repartidores de carga.

3.3.2.3. Sistema de Control.

Firmware

Hemos de implementar el protocolo Dmx y enviar los datos actualizados periódicamente. Aunque el universo está compuesto por información para 512 canales, en esta versión implementaremos solamente 6 de ellos, dejando la ampliación para un trabajo futuro.

El esquema de funcionamiento se basa en la recepción de los valores por Usb, actualizar los datos, encapsular los datos actualizados bajo especificaciones Dmx e inyectar los datos al pin correspondiente del microcontrolador.

Software de control y monitorización

Es una tarea temporizada que envía continuamente los valores deseados de los canales de iluminación. Para ello dotaremos la interfaz de control con un deslizador y un numero de canal deseado (patch) para cada canal. Al arrancar el envío temporizado, los valores son capturados , encapsulados para el envío por puerto Usb.

Para el desarrollo de esta aplicación hemos usado el lenguaje c++ y el entorno de programación VisualC++. Es una aplicación sencilla compuesta de dos clases, una dedicada a la interfaz usb que importa la librería de comunicación con el microcontrolador y la segunda clase dotada con la interfaz de captura de datos a través de controles numéricos.

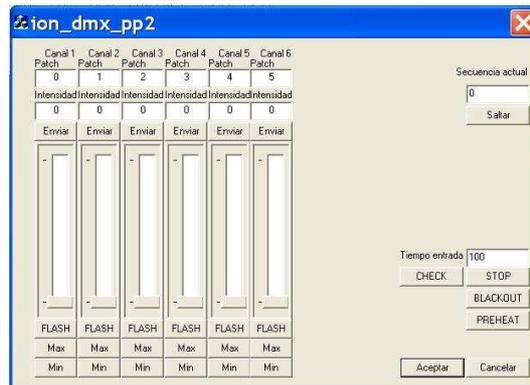


Figura 40 Software de control y monitorización

Una vez montado el sistema, se procede a la comprobación, se activan los seis primeros canales en la casilla patch y se pulsa el botón preheat, que tiene la función de enviar el valor de 20 a todos los canales para el precalentamiento del filamento de las lámparas incandescentes.

Seguido, pulsamos blackout, lo que enviará el valor cero a todos los canales y por último empezaremos subir los deslizadores de cada canal suavemente y de manera individual para comprobar tanto la correcta transmisión y recepción de datos como el correcto funcionamiento del actuador.

3.3.3. Control de un vehículo tipo quad.

La Robótica es una combinación de varias disciplinas, como la Mecánica, la Electronica y la Informática, que junto al conocimiento de la aplicación a la que se enfoca brinda a investigadores un amplio y variado campo de trabajo que a pesar de estar en fases iniciales, avanza rápidamente.

Cualquier objetivo definido en el campo de la Robótica llena de pasión y sobretodo de satisfacción en el momento del logro. Actualmente Hay muchos proyectos funcionando, sobretodo en la industria que ha tenido un importante crecimiento y mayores beneficios desde que estos proyectos se han hecho realidad.

La aplicación práctica que propondremos contribuirá en un futuro próximo al desarrollo de investigaciones mucho más ambiciosas en el campo de la Robótica.

Ello forma parte de un estudio en el cuál participan otros grupos de trabajo que ayudarán a la automatización completo del vehículo. Las principales tareas que se desarrollarán en el vehículo son:

- Control del acelerador del vehículo.
- Control del freno del vehículo.
- Control de la dirección del vehículo.
- Control de sensores para la detención de obstáculos más próximos.
- Control de láser para la detención de obstáculos más lejanos.
- Control del sistema de visión para el reconocimiento de formas.

El siguiente proyecto desarrolla los primeros tres puntos. El objetivo consiste en desarrollar un mecanismo de control, que transfiera los datos desde el ordenador a un microcontrolador, capaz de emitir señales que nos permitan interactuar con el servomotor del acelerador, freno y el posicionamiento de la dirección del vehículo.

El control del acelerador, freno y dirección de un quad.



Figura 41 El vehículo quad con los actuadores montados y en fase de experimentación.

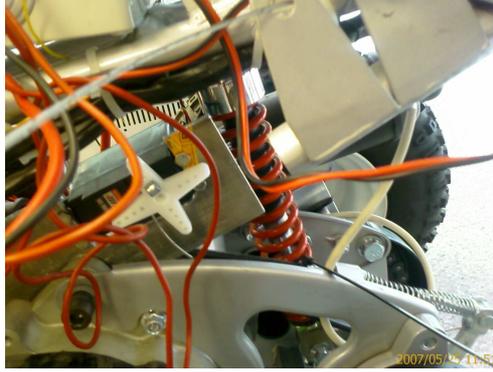


Figura 42 El servo que actúa sobre el freno del quad

En el sistema de navegación móvil planteado, nosotros intervendremos en la parte de control del acelerador, freno y dirección al que llamamos el sistema de movilidad. Estos tres elementos son dirigidos automáticamente por otro sistema de adquisición de datos basado en rastreo de objetos.

El quad es un vehículo de tracción mecánica sobre cuatro ruedas que dispone de un motor térmico de dos tiempos. Está dotado con un motor de arranque eléctrico, un sistema de acelerador que controla la válvula de gasolina, el sistema de frenado que actúa sobre las ruedas traseras y el sistema de dirección que maneja las ruedas delanteras.

El acelerador está gobernado por un servo mientras que otro servo maneja el sistema de frenado.

La dirección del vehículo se maneja gracias a un sistema electromecánico constituido por un motor de corriente continua que, mediante un engranaje tipo sinfín, posiciona un pistón que actúa sobre el ángulo de dirección de las ruedas delanteras.

El ángulo de giro se obtiene del cálculo del valor de un potenciómetro situado en el eje de giro de la dirección.

Por lo tanto, hay dos tipos de actuadores:
servos para el acelerador y el freno controlados por pwm
sistema de posicionamiento lineal en bucle cerrado con realimentación.

3.3.3.1. Sistema Mecánico.

Acoplamiento de los servos

Los servos están sujetos al chasis del vehículo. En la cruceta conectada al eje principal del servo se monta el cable del freno o la válvula de entrada para la inyección. Se comprueba que los servos tienen suficiente movilidad para ofrecer el giro necesario para acelerador suelto y llegar hasta aceleración máxima. En el caso del freno, se ha de comprobar que el freno puede estar perfectamente suelto o bloqueado para impedir el movimiento de las ruedas.

Hemos de asegurar que el servo está bien centrado y equilibrado con el sistema al que se conecta para que el trabajo mecánico sea óptimo.

Acoplamiento del motor DC al sistema de dirección

Este motor está previsto de doble pistón, los dos han de ser conectados a las ruedas delanteras, se ha de montar en su posición central y las ruedas han de estar paralelas. Una vez acoplados los pistones a las ruedas, estas se ponen rectas y el motor se sujeta al chasis del vehículo.

Acoplamiento del potenciómetro en el eje de la dirección

El potenciómetro tiene la utilidad de encoder, es decir, ha de facilitar la posición de las ruedas delanteras. Para ello se acoplará directamente en el eje de la dirección. Puesto que el máximo ángulo de giro permitido es de 45 grados y el potenciómetro tiene un giro máximo de aproximativo 340 grados efectivos, una mejora sustancial en la resolución de la posición de la dirección sería un acoplamiento reductor mediante elementos mecánicos reductores y la utilización de lecturas comparativas de voltaje del microcontrolador para aumentar la resolución de los datos adquiridos. Las tareas de mejora las dejamos pendiente para trabajos futuros.

3.3.3.2. Sistemas de Actuación .

Motores DC

Los motores D.C se clasifican de acuerdo al tipo de bobinado del campo como motores Serie, Shunt, Shunt estabilizado, o Compuesto (Compound). Sin embargo algunos de ellos pueden ser auto excitados o de excitación separada o pueden tener campos de imán permanente.

Ellos muestran curvas muy diferentes de torque-velocidad y se conectan en diferentes configuraciones para diferentes aplicaciones.

Algunos motores D.C utilizan imán permanente como campo principal, especialmente los de potencia (HP) fraccionada ($1/4, 1/2, 3/4$) y baja potencia. Los motores de imán permanente tienen la ventaja de no requerir una fuente de potencia para el campo, pero tienen la desventaja de ser susceptibles a la desmagnetización por cargas de choque eléctricas o mecánicas. Los campos de imán permanente no se pueden ajustar para entonar el motor para ajustarse a la aplicación, como pueden los de campo bobinado.

Las características velocidad - torque dan al motor DC una versátil aplicación. El torque de régimen de un motor D.C es dado a una velocidad específica llamada Velocidad Base.

El motor para el movimiento de la dirección del quad y las especificaciones en la chapa del mismo.

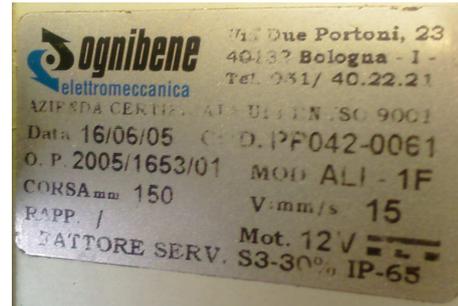


Figura 43 Motor DC para el control de la dirección Quad

En la figura se pueden observar las especificaciones del motor para el control de la dirección. En el eje principal se encuentra montado el potenciómetro que indica la posición del ángulo de dirección.

En la hoja de datos del fabricante se especifica la utilización de este actuador en sistemas en vehículos que no superan 50km/h.

Se necesita un circuito electrónico adicional para el control del motor DC. Hemos optado por realizar este circuito basando en el integrado 18200, Puente-H de potencia con control pwm al que emplearemos para gobernar el motor de la dirección. La señal Pwm puede controlar la velocidad de giro pero este la velocidad máxima de movimiento del pistón no es muy elevada por lo que optaremos por el control todo-nada.

Por lo tanto, en esta aplicación práctica, configuraremos el sistema genérico para el envío de dos señales pwm a los servos del acelerador y freno, una señal todo-nada para el control del motor DC y para la adquisición de datos analógicas para la lectura del valor del potenciómetro.

Montaremos un circuito adicional al sistema de dirección para el control del motor DC. Normalmente se emplea un puente-H pero por cuestiones físicas y trabajos futuros hemos optado por el circuito integrado 18200 que puede mover izquierda, derecha, desembragar, dirección o mantener fija en una posición además del control de la velocidad de giro por pwm.

El control de la dirección que planteamos en esta aplicación práctica es sencilla: mover la dirección hasta que el potenciómetro nos devuelve un valor igual a la posición deseada. En realidad se trata de la implementación de un servo digital dónde hemos sustituido el control electrónico normalmente usado por los impulsos del microcontrolador.

La optimización del posicionamiento de la dirección implica la implementación de un regulador para mejorar la respuesta del sistema y lo hemos previsto para trabajos futuros.

3.3.3.3. Sistema de Adquisición de datos.

Usaremos la adquisición de datos analógica y conversión a digital a través de una entrada analógica del microcontrolador, definida a este fin. Esta dispone de una resolución de 10bits, es decir el rango de valores de 0 y 5 voltios serán convertidas en un rango de 0 a 1023 valores digitales.

valor digital	valor analógico
1023	5V
0	0V
511	2.5V
205	1V

Tabla 9 Equivalencias digital - analógico

Una mejora sustancial en la resolución de las lecturas se consigue empleando una característica que ofrece el microcontrolador Pic, Vref. Dos de los pines se configuran cómo voltaje referencia positivo y negativo y la lectura se traduce a este rango especificado.

Los valores devueltos por le potenciómetro están en el rango 619 para giro máximo derecha y 719 para giro máximo izquierda. Esto nos da un resolución de solamente 100 valores. Configurando Vref- en 1.77V y Vref+ a 4.77V, teniendo en cuenta que el valor centro del potenciómetro está en 3.27V, obtendremos una resolución tres veces mayor.

Configuración del microcontrolador para la adquisición de datos

Lectura potenciómetro: configuración ADCON

El microcontrolador dispone de 13 entradas analógicas de los cuales activaremos una para la adquisición de datos y dos para los valores de referencia. Los registros encargados de la adquisición y conversión de analógico a digital son ADCONx.

3.3.3.4. Sistema de Alimentación.

Alimentación servos, motor DC, potenciómetro

Circuito batería

La batería del quad debe suministrar suficiente corriente para motor de arranque del quad (12V). y para la alimentación de los servos y del sistema de control del quad. Suministraremos 5V para los servos, regulados con el componente LM7805 y 12V para la dirección, regulados con el componente LM7812,

El potenciómetro se ha de alimentar a 5V, tomando corriente de la misma fuente que los motores.

3.3.3.5. Sistema de Control.

Firmware:

El dispositivo debe atender ordenes del pc a través del usb y gobernar los dos servos del acelerador y freno además del sistema de posicionamiento de la dirección constituido por lectura del potenciómetro, calculo del error y aplicación del regulador y actuar sobre el potenciómetro.

Los datos que recibe el microcontrolador son valores de aceleración / desaceleración por el posicionamiento del servo del acelerador en un rango de 0 a 180 grados. De forma análoga se recibe los datos para el freno con el valor cero representando el freno libre y 180 freno activado a máximo. Otro dato relevante es la dirección de giro, 0 a izquierda y 1 a derecha. Por ultimo, el valor de la posición de la dirección, comparable con el valor del potenciómetro.

Los valores del acelerador, freno, posición giro y sentido giro de la dirección del vehículo son determinados en el momento de la calibración y recibidos por el firmware



Figura 44 Disposición de los datos en el vector de bytes

Los datos a transmitir al PC son las posiciones de los dos servos y la lectura del potenciómetro junto al error a corregir.

Estructura funcional del firmware

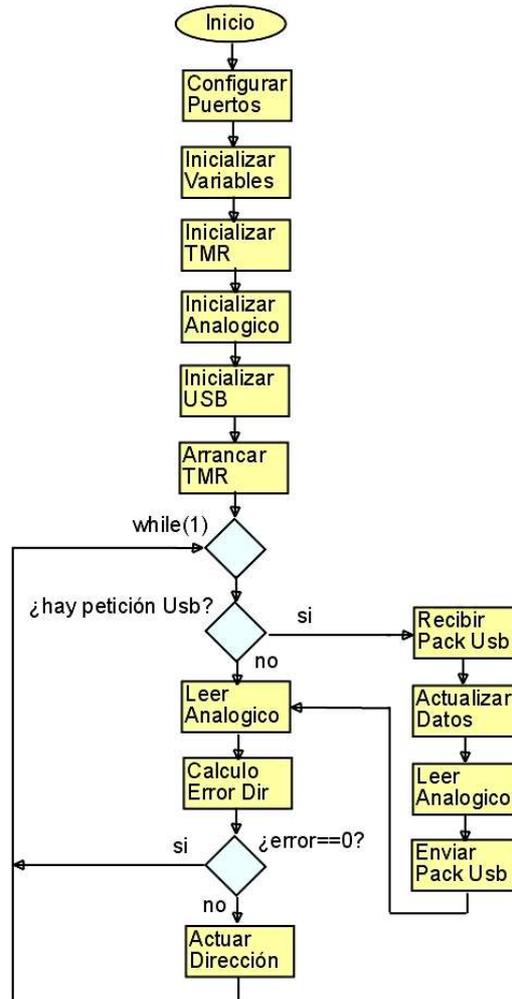


Figura 45 Diagrama lógico del firmware

Para nuestra aplicación práctica, la interpretación de los datos analógicos es la correspondencia entre estos y los valores angulares de giro de la dirección.

Aplicación software de control y monitorización de acelerador, freno y dirección con control remoto.

Este es un programa para probar las funcionalidades de los sistemas implementados en el quad y garantizar un perfecto control manual. Está organizado por módulos de tal manera que para desarrollar futuras aplicaciones basta con añadir los módulos que interesa al nuevo proyecto.

Definimos tres módulos principales:

la comunicación usb con el dispositivo genérico,

la comunicación mediante sockets (wifi) con el control remoto

la monitorización de los datos a enviar y de los datos recibidos.

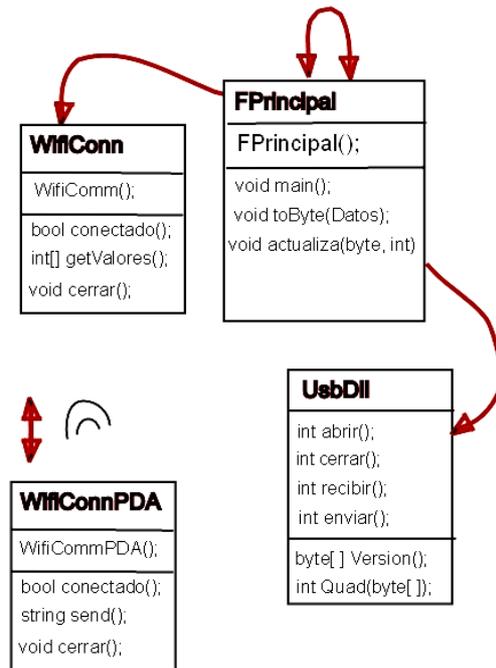


Figura 46 Clases de la aplicación de control y monitorización.

Para la movilidad se plantea una aplicación con conectividad wifi para el control remoto.

El último módulo es la aplicación principal.

La interfaz de comunicación con el dispositivo usb será le encargada de la conversión de los datos mostrados / requeridos a datos que el microcontrolador entienda. Esta comunicación tiene la segunda prioridad, la más alta prioridad la tiene el control remoto.

La interfaz remota es la encargada, de recibir los datos del control remoto y actualizar los datos a enviar al dispositivo usb.

El control remoto ha de recibir periódicamente los estados actualizados de los actuadores del vehículo por parte de la interfaz de monitorización.

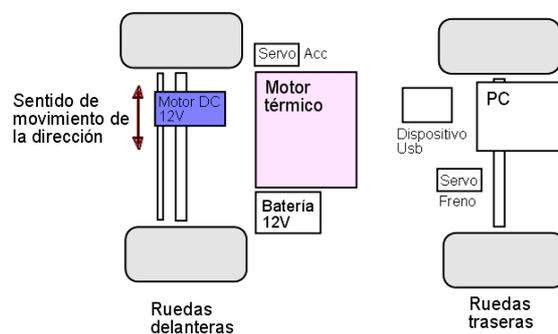


Figura 47 Esquema componentes del quad.

Necesidad de control remoto

El Pc que controla el dispositivo Usb no dispondrá de pantalla o teclado durante el movimiento del vehículo. Para acceder a estos datos implementaremos dos vías mediante configuración y conexión a una red Wifi:

- Control remoto del Pc quad con escritorio remoto la que usaremos para arrancar el programa de control, comprobar el estado del sistema operativo o en caso de emergencia para acceder a la información adquirida. No vamos a usar esta opción para el control del quad por las inconveniencias de tiempo real requerido.
- Control remoto a través de una Pda. El programa de control del pc actúa como servidor, atendiendo las ordenes del programa cliente del Pda.

Control manual de la aplicación

El programa de control de la Pda puede acelerar, frenar o posicionar la dirección. Para más seguridad, el quad dispone de un mando a distancia RF que corta la alimentación con gasolina al motor térmico para casos extremos.

El software del sistema genérico está configurado con controles y mecanismos de sincronización y de exclusión mutua para el control de los actuadores.

Además de los trabajos futuros mencionados queremos plantear la gestión de la alimentación para conseguir una alimentación inteligente además de un sistema alternador para la recarga de las baterías cuando el vehículo se encuentra en marcha.

Y quizás una utilidad que puede resultar a la vez apasionante es la integración de gps para selección de objetivo y posicionamiento.

4. Conclusiones.

4.1. Conclusiones.

El trabajo presentado facilita la conectividad de periféricos de diseño con un ordenador personal para ser controlados y monitorizados de una manera sencilla, rápida y económica. Su aplicación es muy amplia pudiendo ser utilizado también en el ámbito profesional. Mediante aplicaciones prácticas de control embebido hemos conseguido, con la configuración correspondiente del sistema, el control en tiempo real de varios tipos de actuadores. Hace una década este proyecto hubiera sido muy costoso mientras que hoy en día, con el avance de la tecnología, la versatilidad de los componentes y el acceso a la información lo hemos conseguido.

4.2. Trabajos futuros.

Todos los campos de investigación y desarrollo que hemos podido abarcar en este proyecto nos han absorbido y cautivado de tal manera que nos es difícil poner punto final. No queremos decir que “lo mejor queda en el tintero” porque lo que hemos conseguido es un primer gran paso que demuestra que los objetivos planteados son factibles. Deseamos, en primer lugar, mejorar el sistema genérico para el control remoto a larga distancia a través de internet y vencer los problemas para conseguir una transmisión de datos en tiempo real. Los trabajos futuros que planteamos cubre tanto el sistema genérico cómo las aplicaciones prácticas. Las aplicaciones prácticas nos han enseñado las posibilidades actuales y las que podemos conseguir con nuestro sistema y cómo podemos mejorarlo.

Como trabajos futuros especificamos:

Adaptar nuestro sistema para la comunicación sobre Ethernet por conexión inalámbrica.

La implementación de Visión artificial para el reconocimiento de figuras simples.

La implementación de un regulador Pid para el control del motor DC.

Implementación y aplicación de algoritmos para reducir la distancia de frenado en el sistema de navegación móvil.

Mejorar la resolución de la lectura de un valor analógica mediante elementos mecánicos reductores y la utilización de lecturas comparativas con un voltaje de referencia.

Aprovechar a fondo las características del circuito integrado 18200 para el control de la velocidad de giro por pwm.

Gestión de la alimentación para conseguir una alimentación inteligente además de un sistema alternador para la recarga de las baterías cuando el vehículo se encuentra en marcha.

Integración del sistema gps al nuestro sistema genérico para selección por objetivo y posicionamiento.

5. Bibliografía.

www.microchip.com

www.mikroe.com

www.usb.org

www.opendmx.net

“Usb Complete: Everything You Need To Develop Custom Usb Peripherals”
Jan Axelson

“Microcontroladores PIC – La solución en un chip”
J.Mª. Angulo Usategui – E. Martín Cuenca – I. Angulo Martinez

“Introducción a los microcontroladores – Hardware, Software, Aplicaciones”
José Adolfo González Vázquez