

Trabajo final de Grado

Diseño e Implementación de un Autómata Programable con núcleo CodeSys para Sistemas Animatrónicos

Grado en Ingeniería Electrónica Industrial y Automática

Francisco Javier Huerta Albiar

1 de Marzo de 2019

Memoria

Tutor:

Sergio García-Nieto Rodríguez



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Agradecimientos

En primer lugar, me gustaría agradecer a mi tutor Sergio su apoyo y persistencia, a pesar de que pasara el tiempo nunca me cerró la puerta y me recondujo y ayudo a encontrar el camino para cerrar esta etapa.

También me gustaría dar las gracias a mis dos familias. A mi madre, padre, tía y primos, porque sin ellos no sería quien soy ni habría llegado donde he llegado. Todo lo que soy os lo debo a vosotros.

En segundo lugar a mi familia valenciana, Conchin, Puri, David, Susi, Cris y compañeros de clase, vosotros habéis hecho que sienta valencia como mi casa y habéis contribuido a que mi etapa en la universidad sea sin duda una de las mejores de mi vida.

Quiero agradecer de manera especial a Nuria, mi compañera de clase y ahora de vida, su apoyo y compañía en este largo camino. Ella ha hecho la universidad y estos años descubriendo el mundo, inolvidables.

Para concluir, me gustaría dedicar este trabajo a mi hermano Mario. No solo por ser un hermano, para mi inmejorable, sino también por los consejos y ejemplo que me ha dado siempre. ¡Muchas gracias hermano!

Índice

1	OBJETIVOS Y ALCANCE DEL PROYECTO	9
1.1	Alcance del proyecto y tareas a realizar	10
1.2	Objetivos	10
2	INTRODUCCIÓN Y ATECEDENTES DEL PROYECTO	11
2.1	Introducción	11
2.2	Definición	11
2.3	Antecedentes de los sistemas animatrónicos	11
3	COMPOSICIÓN DE UN SISTEMA ANIMATRÓNICO	15
3.1	Estructura mecánica	16
3.2	Actuadores	16
3.3	Piel o acabado estético	17
3.4	Sistema de control	18
4	ÁMBITO DE APLICACIÓN Y ALTERNATIVAS DE DISEÑO PARA EL SISTEMA DE CONTROL	21
4.1	Ordenador embebido	21
4.1.1	Condiciones de diseño	21
4.1.2	Alternativas de diseño	22
4.2	Interfaz Hardware	23
4.2.1	Actuadores	23
4.2.2	Sensores	29
4.2.3	Comunicaciones	33

5	SOLUCIÓN EN TÉRMINOS DE HARDWARE	35
5.1	Introducción.....	35
5.2	Ordenador embebido	35
5.3	Interfaz hardware	37
5.3.1	Alimentación.....	40
5.3.2	Entradas digitales	43
5.3.3	Salidas digitales.....	46
5.3.4	Relés.....	48
5.3.5	Salidas PWM	50
5.3.6	Salida de audio.....	52
5.3.7	Interfaz DMX.....	55
5.4	Distribución entradas/salidas del sistema.....	55
6	ÁMBITO DE APLICACIÓN Y ALTERNATIVAS DE DISEÑO PARA EL SOFTWARE	59
6.1	Plataformas de programación	59
6.1.1	Condiciones de diseño.....	59
6.1.2	Alternativas de diseño	60
7	SOLUCIÓN EN TÉRMINOS DE SOFTWARE	65
7.1	Programa de validación	66
7.1.1	Funciones o módulos POU.....	68
7.1.2	Ejecución de tareas.....	73
7.1.3	Interfaz gráfica o GUI.....	74
7.1.4	Configuración dispositivo	75

8	RESULTADOS Y TEST	77
8.1	Validación interfaz hardware	77
8.2	Validación ordenador embebido	78
8.3	Validación sistema completo	79
9	CONCLUSIONES	81
	Índice de ilustraciones	83
	Índice de tablas.....	87
	Bibliografía.....	89

1 OBJETIVOS Y ALCANCE DEL PROYECTO

El presente proyecto tiene como objetivo la implementación de una plataforma electrónica flexible que permita el control de sistemas animatrónicos.

La plataforma debe estar diseñada para poder controlar sistemas animatrónicos basados en componentes industriales, lo que permitirá que el tiempo y el coste de la implementación y construcción del sistema se reduzcan considerablemente.

Para ello se hará un estudio con el fin de maximizar el número de entradas y salidas utilizadas en la plataforma de procesamiento elegida, así como de la interfaz que se le asigna a cada una de ellas.

Dado que la construcción de un sistema animatrónico completo es objetivo de este proyecto sólo se centrará en el desarrollo del HW de la plataforma electrónica que permitirá, junto con un montaje electromecánico, la construcción de un sistema completo.

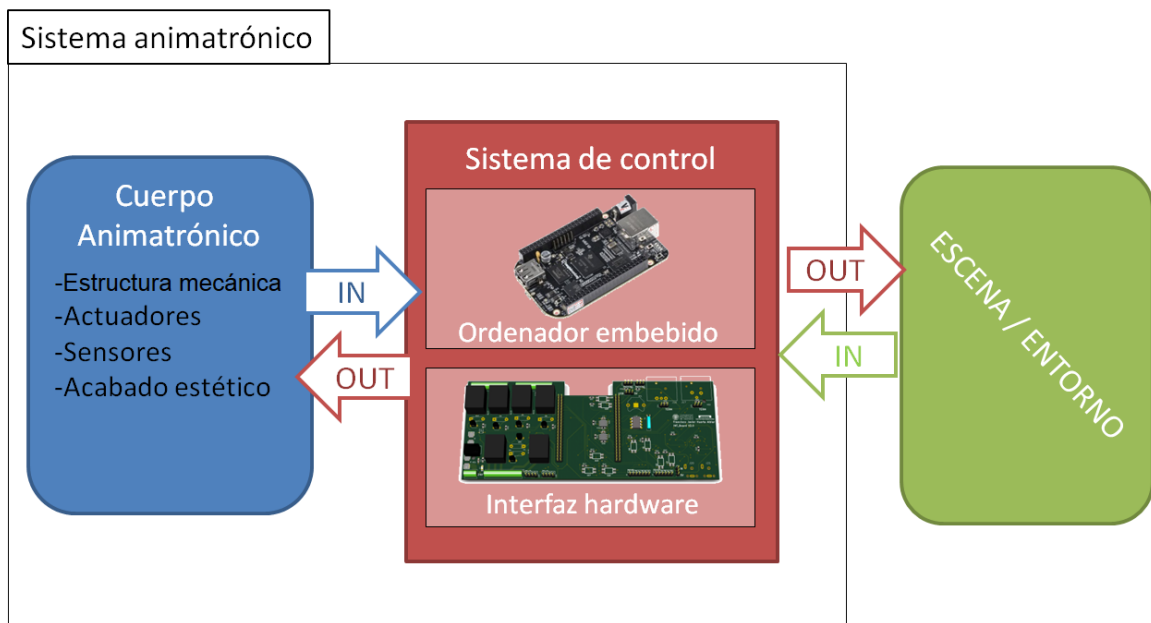


Ilustración 1 - Esquema sistema animatrónico

Para alcanzar todos estos objetivos, será necesario aplicar los conocimientos adquiridos durante el periodo formativo del Grado de Ingeniería Electrónica Industrial y Automática además del aprendizaje de nuevos conceptos que aparecen cuando se enfrentan proyectos reales.

1.1 Alcance del proyecto y tareas a realizar

Para poder acometer los objetivos principales establecidos, a continuación se presenta una enumeración de las tareas más importantes a realizar.

- Estudio de las necesidades de los sistemas animatrónicos.
- Estudio de las interfaces más usuales para elementos electromecánicos industriales y de consumo.
- Selección de un ordenador embebido para la plataforma.
- Definición de las entradas y salidas de la plataforma electrónica.
- Diseño del hardware necesario adicional para la plataforma mecánica.
- Fabricación de la plataforma.
- Desarrollo software básico para controlar y verificar la plataforma electrónica.

1.2 Objetivos

- Diseñar una plataforma electrónica que permita el control de sistemas animatrónicos.
- La plataforma electrónica debe ser flexible y enfocada al control de elementos electromecánicos industriales y comerciales.
- Desarrollar un software básico que permita verificar la mayoría de entradas/salidas de la plataforma.

2 INTRODUCCIÓN Y ATECEDENTES DEL PROYECTO

2.1 *Introducción*

Antes de entrar al apartado descriptivo de este proyecto es fundamental entender qué es un sistema animatrónico, de modo que se pueda tener una orientación a la hora de comprender el diseño y las elecciones que se han tomado para su realización.

También se realizará una pequeña revisión de la historia de estos sistemas desde sus comienzos hasta la actualidad, con el fin de tener una visión general de la utilización de estos sistemas y del potencial que tienen en la actualidad.

2.2 *Definición*

Los sistemas animatrónicos son aquellos mecanismos robóticos o electrónicos que tienen como objetivo simular el aspecto y el comportamiento de un ser vivo. Los más habituales tienen un aspecto antropomórfico. [1] Estos sistemas suelen ser marionetas que gracias a la mecatrónica que lo componen pueden ser programados y controlados remotamente.

2.3 *Antecedentes de los sistemas animatrónicos*

Los fabricantes de relojes ya trabajaban en sistemas de gran complejidad mecánica donde había figuras que aparecían y desaparecían e incluso se movían sincronizados con el sonido del reloj. Estos serían los antecesores de lo que hoy conocemos como sistemas animatrónicos.

Un buen ejemplo de hasta donde podían llegar en esa época es el reloj astronómico de Praga, que fue construido en el año 1400. Este reloj no sólo da la hora, si no que es capaz de indicar con sorprendente exactitud las fases lunares y estaciones de una forma gráfica, además de ofrecer un espectáculo de figuras en movimiento.



Ilustración 2 - Reloj astronómico de Praga (www.disfrutapraga.com).

Puede parecer un mecanismo sencillo pero para la época sin duda fue una revolución y seguramente la semilla de la introducción de estos automatismos para llamar la atención del espectador, que posteriormente alimentará el interés por la evolución de estos sistemas [2].

Esto sólo fue el comienzo de este concepto, pero para conocer los antecedentes de la animatrónica moderna, se debe estudiar a su precursor Walt Disney.

Disney entendía la animatrónica como una novedad que atraería público a ver sus películas y parques temáticos, es más, creía que en un futuro estos sistemas podrían sustituir a actores y actrices en shows repetitivos cumpliendo siempre los horarios.

Es por esto que Walt Disney llevó la tecnología hasta el extremo de lo conocido en la época para conseguir su objetivo.

Su primera creación fue "9 tall dancing man", basado en un sistema rotativo de cámara. Pese a que el sistema fuera muy básico fue el pistoletazo de salida para la era de los sistemas animatrónicos que posteriormente darían lugar a sistemas más complejos y perfeccionados como los pájaros "Tiki encantados" para Disneyland (1963) y la primera figura humanoide de Abraham Lincoln para la feria mundial (1964).



Ilustración 3 - Sistemas animatrónicos Walt Disney; Abraham Lincoln, pájaro Tiki, "9 tall dancing man" (www.thewaltdisneycompany.com).

A día de hoy alguno de los sistemas animatrónicos más avanzados siguen estando en los parques Disney, pero ahora su uso se ha popularizado y extendido [1][3][4].

Se puede encontrar estos sistemas también en grandes producciones de cine simulando todo tipo de seres vivos. Un gran ejemplo de la utilización de éstos son los sistemas animatrónicos utilizados en las películas de Jurassic Park.

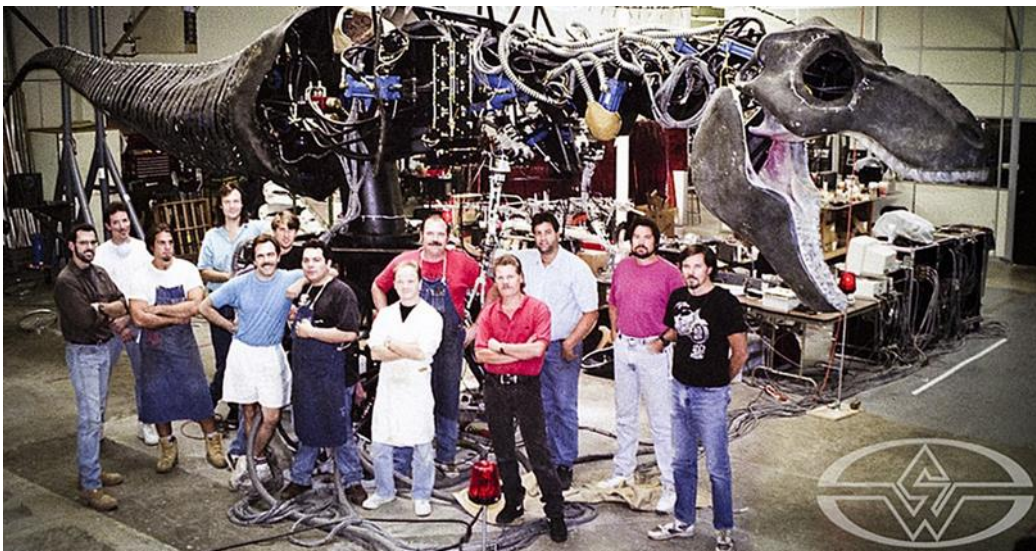


Ilustración 4 - Equipo Stan Winstons School con sistema animatrónico de la película Jurassic Park (www.stanwinstonschool.com).

En la actualidad se está dando un paso más allá con la animatrónica humanoide en la que se están integrando las técnicas más avanzadas en el campo de la inteligencia artificial y del modelado artístico. Con esto se pretende conseguir una interacción casi

humana con los sistemas, haciendo muy complicado diferenciar si lo que el usuario tiene delante es un robot o un humano.



**Ilustración 5 - Sistema animatrónico humanoide FRED de Engineered Arts
(www.engineeredarts.co.uk).**

3 COMPOSICIÓN DE UN SISTEMA ANIMATRÓNICO

Los sistemas animatrónicos están compuestos por cuatro elementos principales:

- Estructura mecánica.
- Actuadores y sensores.
- Piel o acabado estético.
- Sistema de control.

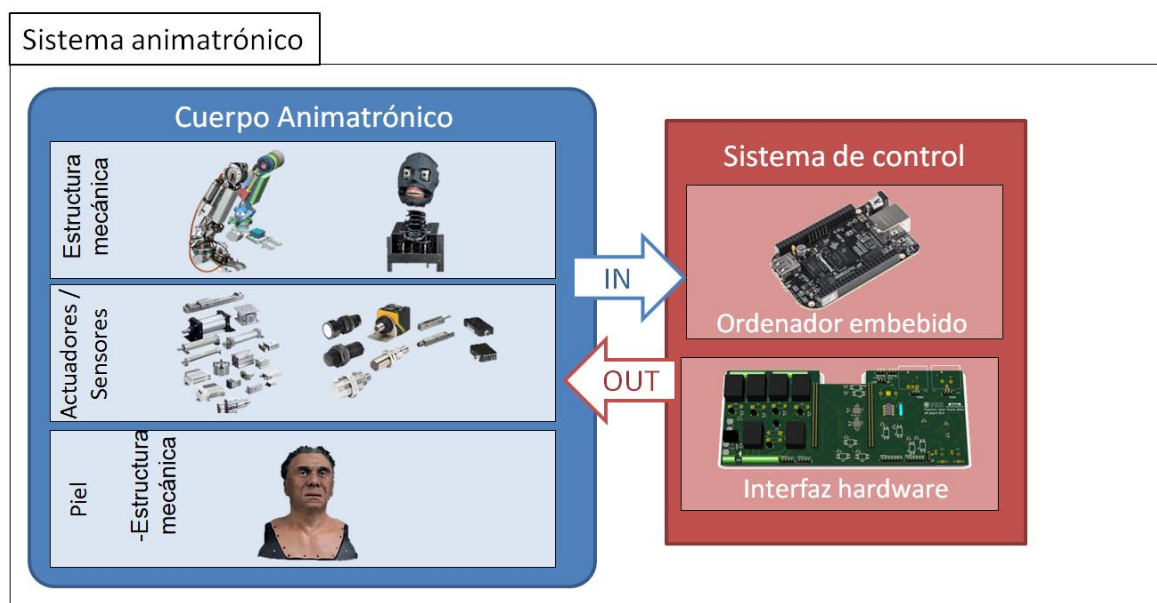


Ilustración 6 - Esquema sistema animatrónico enfocado al TFG.

La conjugación de todos estos elementos hacen que un robot se convierta en un sistema animatrónico capaz de pasar por un ser vivo [5].

En este apartado se describirá en detalle cada una de las partes fundamentales de un sistema animatrónico. A pesar de que este proyecto sólo se centre en el diseño del sistema de control, es muy importante comprender las características y limitaciones de cada una de sus partes para que el diseño este optimizado y sea flexible.

3.1 Estructura mecánica

El esqueleto, como viene implícito en el nombre, representa los huesos del sistema animatrónico. La función principal de este componente es dar sustentación y forma de ser vivo al sistema completo.

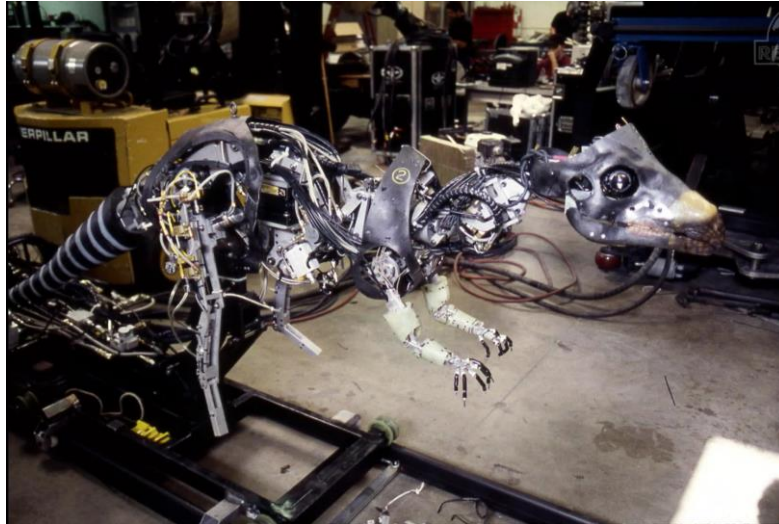


Ilustración 7 - Esqueleto sistema animatrónico Jurassic Park (www.stanwinstonschool.com).

Esta parte suele estar compuesta por estructuras metálicas y rótulas mecánicas que permiten al sistema animatrónico, junto con los actuadores, simular la forma y el movimiento del ser vivo objetivo.

Este componente tendrá una influencia muy reducida sobre el presente proyecto. Dado que se busca un sistema flexible no se predefinirá el manejo de ningún esqueleto como puede ser el de un humano o un pájaro, si no que se trabajará con conceptos generales que permitan el manejo de ambas estructuras.

3.2 Actuadores

Como se ha explicado en apartados anteriores, lo que diferencia a un sistema animatrónico de otros robots es que los primeros intentan imitar/representar a seres vivos.

El punto crítico a la hora de simular un ser vivo es conseguir una cinemática natural, por lo que será fundamental que el componente que se desarrolle en este proyecto ofrezca una gran flexibilidad a la hora de elegir y controlar sistemas cinemáticos. Todo

esto junto a un buen diseño mecánico y control hará que el sistema animatrónico sea de alta calidad.



Ilustración 8 - Interpretación de una extremidad desde modelo biológico a implementación mecatrónica (Daniel Kuehn, ACTIVE SPINE AND FEET WITH INCREASED SENSING CAPABILITIES FOR WALKING ROBOTS).

Más tarde se hará un análisis de los sistemas cinemáticos disponibles en el mercado para tener una idea del hardware necesario para que el sistema de control pueda manejar el mayor número posible de actuadores.

3.3 Piel o acabado estético

Una vez se dispone del sistema completo, a pesar de que los movimientos y el comportamiento del sistema sean casi idénticos al de un ser vivo, éste no cumple su objetivo principal de simular lo máximo posible un ser vivo, dado que estéticamente se siguen viendo todos los sistemas electrónicos y mecánicos.

Por ello es muy importante este último componente, ya que el acabado estético es la piel del sistema y enmascara todos los otros componentes para que el parecido al ser vivo sea lo mayor posible.

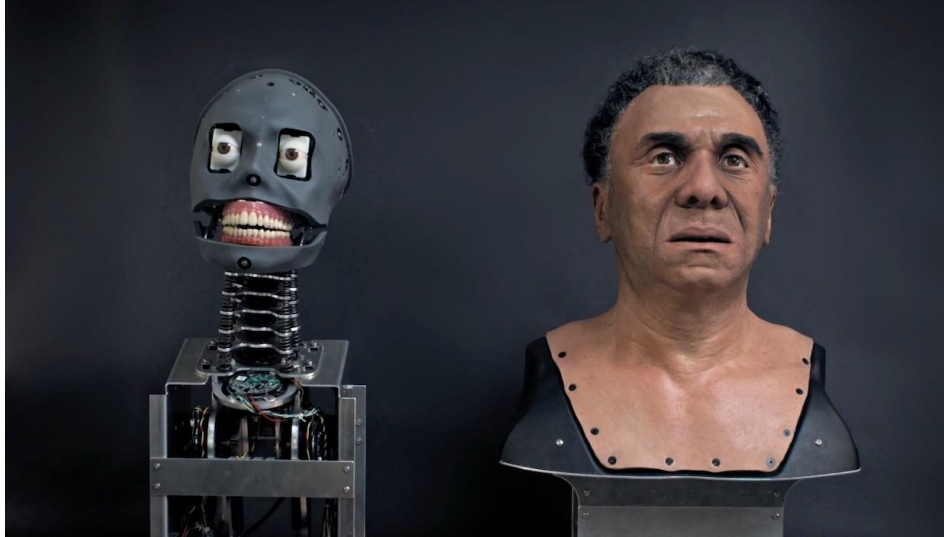


Ilustración 9 - Esqueleto y acabado estético de FRED (www.engineeredarts.co.uk).

Aunque esta parte parezca sencilla, tiene una gama muy amplia de aplicación que va desde sencillos acabados superficiales de tela hasta complejos acabados estéticos de látex y silicona.

Como se ha explicado en el apartado del esqueleto, este proyecto no se centrará en ninguna forma antropomórfica definida por lo que no se contemplará este componente en el diseño.

3.4 Sistema de control

El sistema de control es, usando la analogía de un ser vivo, el cerebro del sistema animatrónico. Este cerebro se debe encargar de coordinar y procesar todas las interacciones del sistema como son: movimientos, sonidos, respuesta a estímulos por medios de sensores, etc.

Los sistemas de control de los sistemas animatrónicos suelen estar compuestos de dos partes principales:

- Ordenador embebido.
- Interfaz de hardware.

El ordenador embebido se encarga de albergar la programación y la lógica de control del sistema. Por otro lado, la interfaz de hardware le permite al ordenador embebido controlar los actuadores y leer los sensores necesarios para controlar el sistema animatrónico.



Ilustración 10 - Esquema sistema de control TFG.

Normalmente esta parte es una compleja caja negra que se compra a un distribuidor comercial y que se debe controlar con un software cerrado. Por esto uno de los objetivos principales del proyecto será diseñar un sistema de bajo coste que sea flexible y abierto para poder modificarlo y utilizarlo según conveniencia del desarrollador.

4 ÁMBITO DE APLICACIÓN Y ALTERNATIVAS DE DISEÑO PARA EL SISTEMA DE CONTROL

Una vez se conocen los componentes del sistema se estudiarán las diferentes alternativas para las variables que afectarán al diseño del sistema de control.

Se empezará por el ordenador embebido, evaluando las alternativas disponibles en el mercado y sus características. A continuación los diferentes periféricos (motores, sensores e interfaces multimedia) que permitirán definir todo los componentes de la interfaz de hardware.

4.1 Ordenador embebido

Este componente, como se ha comentado anteriormente, es el cerebro fundamental para el funcionamiento autónomo de todos los mecanismos del sistema. Por ello, es muy importante la elección correcta de éste a la hora de diseñar el sistema.

En los siguientes apartados se tratarán las condiciones de diseño que se deben tener en cuenta para la elección del sistema de procesado y también las diferentes alternativas comerciales que hay en el mercado para identificar cuál cumple mejor los requerimientos del proyecto.

4.1.1 Condiciones de diseño

A la hora de elegir el sistema de procesado se debe tener en cuenta principalmente el número de entradas y salidas que proporciona el mismo, cuantas más entradas y salidas disponga el sistema, más flexible será para controlar sistemas con diferentes estructuras.

No sólo es importante la cantidad de entras y salidas sino también su topología:

- Salidas de uso general.
- Entradas de uso general.
- PWM.
- UARTs, SPI, I2C, etc.
- Salidas de audio.

En la elección del sistema de procesado no es sólo importante el hardware, es decir, el procesador en el que se basa el sistema, sino también en el software necesario para el manejo del mismo y la posterior aplicación en el sistema electrónico.

A continuación, se describen las alternativas disponibles en el mercado comparando las características expuestas en este apartado.

4.1.2 Alternativas de diseño

Dado que uno de los objetivos es que el sistema no tenga un coste elevado, compararemos las dos plataformas más económicas en términos de computación embebida basada en Linux.

Estas plataformas son Raspberry Pi y Beaglebone Black, cuyas características se comparan en la siguiente tabla:

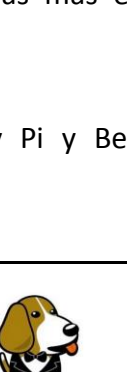
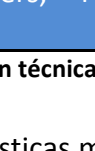
	 beaglebone	 RaspberryPi
Procesador	1 GHz TI Sitara AM3359	1,2 GHz quadcore
Arquitectura	32-Bit	64-Bit
RAM	512MB DDR3L	1 GB LPDDR2
Almacenamiento	SD+ 4GB on-board	SD
Salidas accesible	92	40
GPIO	65	24
Interfaces digitales	2 I2C, 2 SPI, 7 Analog Inputs, 8 PWM, 4 Timers, 4 UARTS, HDMI	I2C, 2 SPI, HDMI, UART

Tabla 1 – Comparación técnica Beaglebone vs Raspberry Pi.

Ambas placas ofrecen unas características muy parecidas en términos de capacidad de procesamiento. Se debe destacar también que ambas plataformas disponen de una gran comunidad que las respalde, lo que asegura un desarrollo más eficaz y también más flexible para posibles modificaciones futuras.

Por otro lado, se diferencian en gran medida en las entradas y salidas disponibles, lo que hará decantar la decisión a la hora de elegir una de las dos plataformas.

4.2 Interfaz Hardware

Dado que se debe diseñar un sistema flexible no se definirán unos periféricos determinados, si no que se diseñará el hardware para ocupar el máximo de su capacidad y que pueda así manejar un mayor número de sistemas distintos.

Por ello, para decidir que componentes integran la interfaz hardware del sistema de control, se hará un análisis de los principales y más frecuentes periféricos que deben ser controlados por el ordenador embebido y su topología, teniendo como resultado los tipos de entradas y salidas que se deben implementar en la interfaz.

Como se ha comentado en los objetivos del proyecto, todos los periféricos que se analizarán serán de uso industrial, lo que permitirá un diseño rápido del sistema animatrónico con un coste más reducido que si se diseñara con componentes específicos [6].

4.2.1 Actuadores

Los actuadores son los encargados en convertir los impulsos electrónicos de control en movimiento. La interacción entre estos movimientos y el esqueleto producen la cinemática del sistema animatrónico.

En función de la naturaleza del movimiento generado podemos clasificar los actuadores en dos grupos:

- Actuadores rotativos: producen movimientos giratorios.
- Actuadores lineales: como bien indica estos actuadores producen movimientos lineales.

A la hora de diseñar un sistema animatrónico, raramente se podrá alcanzar la naturalidad del movimiento sin combinar ambos tipos de actuadores. Por lo que, el sistema que se va a diseñar en este proyecto deberá contemplar ambos.

La figura que se muestra a continuación es un buen ejemplo de la conjugación de motor y actuadores lineales para simular la cinemática de una pierna humana:

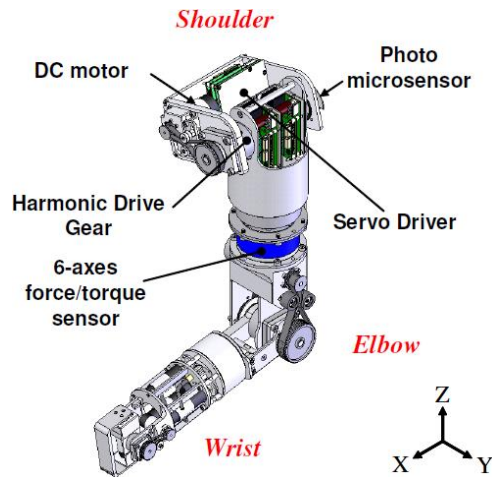


Ilustración 11 - Actuadores presentes en un modelo de extremidad (www.hindawi.com).

Dentro de estas categorías se pueden encontrar dispositivos que ofrecen características técnicas diferentes como peso, par, precisión y velocidad entre otras y que condicionarán el diseño según las necesidades y requerimientos del sistema.

A continuación se analizarán brevemente los diferentes actuadores que el sistema de control debería de poder controlar para cumplir los objetivos del proyecto.

4.2.1.1 Actuadores giratorios o motores

Servomotores [7]

Los servomotores o servos tienen como característica principal, dentro de la categoría de los motores, su precisión a la hora de situar el eje en una posición.



Ilustración 12 - Tipos variados de servos (www.superrobotica.com).

Los servomotores analógicos, por ejemplo, están compuestos por:

- Motor DC: Genera el movimiento circular del servo.
- Potenciómetro: Detecta la posición del eje.
- Circuito de control: Controla la posición del eje con el input del potenciómetro.
- Transmisión: Controla la velocidad de giro del eje y lo bloquea cuando está en la posición requerida.

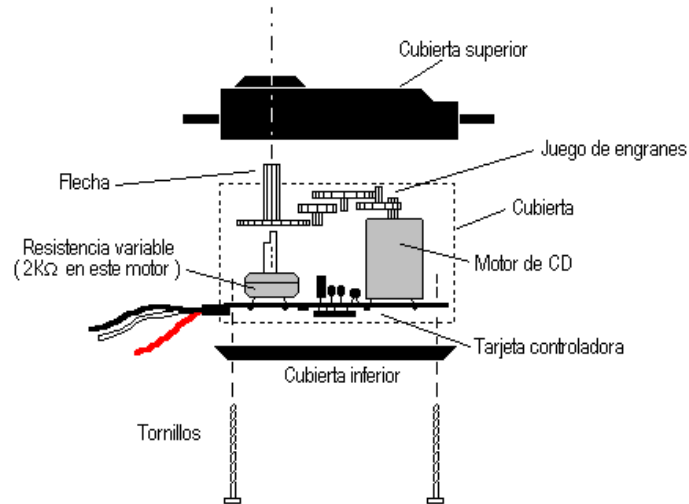


Ilustración 13 - Esquema componentes de un servo (www.comofunciona.co.com).

Para que el servo funcione se le deben proporcionar tres señales: alimentación, masa y una señal de control.

El servo es controlado mediante un pulso eléctrico de ancho modulable o PWM (“pulse width modulation”). Como se muestra en la siguiente figura, el ancho de pulso de esta señal definirá el ángulo de giro del eje del motor.

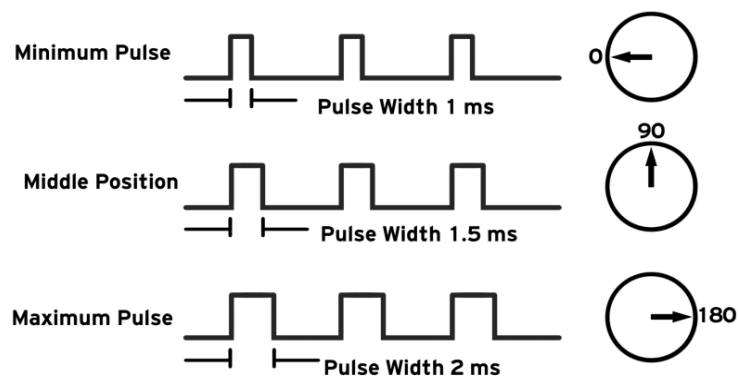


Ilustración 14 - Relación ancho de pulso vs ángulo de giro de un servo 180° (www.jameco.com).

La aplicación de los servos en sistemas animatrónicos, suele ser para simular movimientos precisos pero que requieren poco esfuerzo o carga para el motor, como por ejemplo, la simulación del parpadeo o gesticulaciones de la cara.

Motores AC y DC [8]

Los motores AC y DC son utilizados para movimientos circulares con mayores requerimientos de velocidad o par.

La diferencia fundamental entre estos dos motores es su alimentación; mientras el motor DC se debe alimentar con una corriente continua, el motor AC debe ser alimentado con una corriente alterna.

Dentro de los motores DC también podemos diferenciar entre dos tipos principales:

- **Motores DC con escobillas:** estos motores disponen de unas escobillas que están conectadas al colector del mismo, de manera que están encargadas de conmutar mecánicamente la corriente de las bobinas del motor. Es importante tener en cuenta que estas escobillas sufren desgaste mecánico por el rozamiento.
- **Motores DC sin escobillas o “brushless”:** en este caso el motor no incorpora colector ni escobillas para cambiar la polaridad del rotor, si no que la conmutación de las bobinas se hacen mediante un circuito electrónico. El ciclo de vida de estos es mayor gracias a que no necesita el rozamiento de las escobillas.

Por otro lado, dentro de la familia de los motores AC de una fase, está el motor de “jaula de ardilla”. El núcleo del rotor de estos motores está construido de chapas estampadas de acero al silicio en el interior de las cuales se disponen unas barras, generalmente de aluminio moldeado a presión. Las barras del devanado van conectadas a unos anillos conductores denominados anillos extremos. El bobinado así dispuesto tiene forma de jaula de ardilla [21].

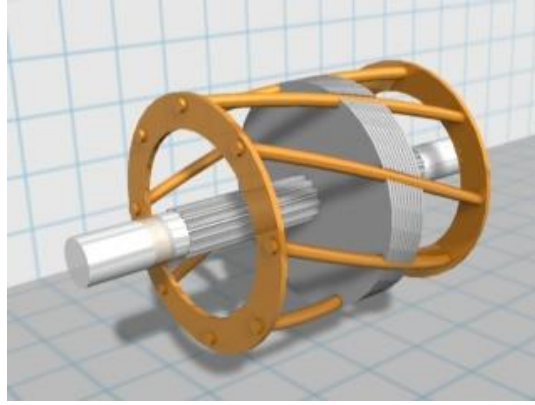


Ilustración 15 - rotor de jaula de ardilla. (Google Imágenes - CC BY ND)

Respecto a la utilización de estos dos tipos de motores, los motores DC suelen ser usados para trabajos que requieran un mayor par.

En cambio los motores AC destacan por su complejo control por lo que normalmente en el terreno industrial los motores AC se controlan por medio de variadores. Estos dispositivos permiten simplificar el control de los motores industriales, estos variadores se suele controlar de manera digital mediante buses de comunicación o con una señal PWM más una salida digital que permiten controlar así respectivamente velocidad de giro y sentido.

En los sistemas animatrónicos se usan estos motores en los giros con mayor carga, como son las articulaciones de extremidades o simular el giro de la cadera entre otros.

4.2.1.2 Actuadores lineales [8]

En el campo de los actuadores lineales se pueden encontrar tres tipos principales en la industria:

- Pistones hidráulicos
- Pistones neumáticos
- Pistones eléctricos

Dada su sencillez y coste el presente proyecto solo va profundizar en los pistones neumáticos ya que estos cuadran más con los objetivos finales del mismo.

Pistones neumáticos

Los pistones neumáticos se utilizan en movimientos lineales puesto que ofrecen una buena relación fuerza/peso y soportan todo tipo de ambientes. También es importante su característica de poder mantener cargas sin necesidad de un freno.

Por otro lado y como desventaja de estos sistemas, requieren de un gran sistema tras ellos para su funcionamiento. A continuación, se muestra un esquema de su funcionamiento:

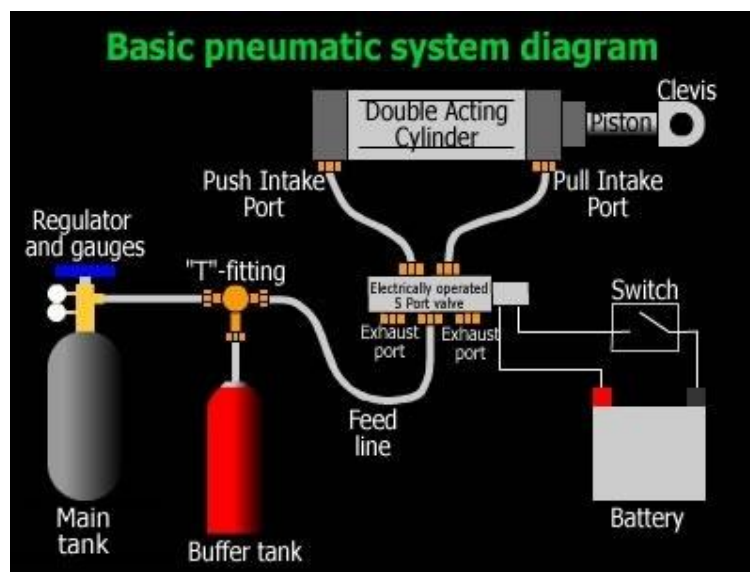


Ilustración 16 - Esquema sistema neumático (www.quora.com).

La necesidad de un sistema neumáticos que alimente los pistones hace que estos sistemas sólo se utilicen en casos en los que la carga sea muy alta y no pueda ser gestionada por actuadores eléctricos.

El sistema de control deberá actuar sobre la válvula que acciona el pistón neumático. Aplicado al sistema de control del proyecto, y más específicamente a la parte de interfaz hardware, se debe asegurar tener salidas digitales que sean capaces de excitar la válvula.

El responsable de accionar la válvula es un solenoide que debe ser activado aplicándole una tensión. En el campo industrial los niveles de alimentación más utilizados para los solenoides son 24V DC o 230V AC.

4.2.2 Sensores

Los sensores son la realimentación del sistema de control, son los encargados de dar al ordenador embebido el estado de las magnitudes físicas del sistema, de modo que pueda accionar los actuadores en consecuencia.

En los sistemas animatrónicos se pueden encontrar sensores muy diversos (inerciales, proximidad, cámaras, etc.) pero la mayoría están orientados a dar un feedback al sistema de la posición de las extremidades y partes del cuerpo del sistema animatrónico o verificar que una acción se ha completado correctamente.

A continuación, se describen alguno de los tipos de sensores utilizados para este fin.

4.2.2.1 Sensores capacitivos e inductivos

Los sensores más utilizados para detectar posición o proximidad son los capacitivos o inductivos. Ambos tienen una misma funcionalidad, detectar objetos sin la necesidad de contacto, pero su principio de funcionamiento es muy distinto.

Los sensores de proximidad capacitivos son capaces de detectar objetos independientemente de su material, basándose en la característica eléctrica de la capacitancia. La capacitancia es la propiedad eléctrica que aparece al aplicar una carga eléctrica a dos objetos conductivos con un espacio entre ellos.

En el esquema siguiente se muestra visualmente como el sensor detecta la presencia del objeto sin contacto.

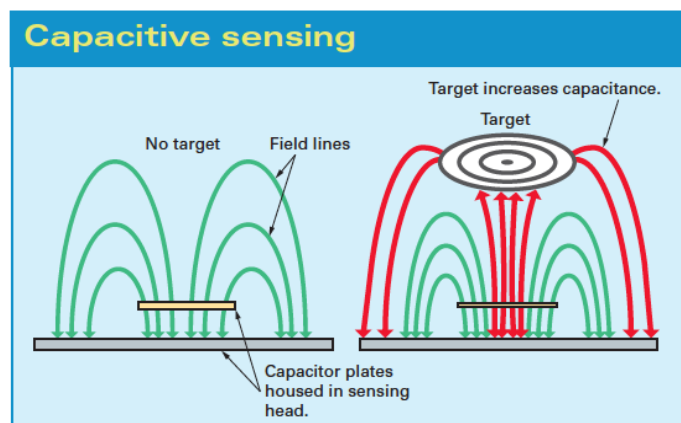


Ilustración 17 - Funcionamiento sensor capacitivo (www.termoregulatory.pl).

Por otro lado, los sensores inductivos se basan en la teoría de las corrientes inducidas, consiguiendo así detectar materiales ferrosos cuando entran en el campo de inducción del sensor.

En el siguiente esquema se muestra visualmente como el campo se ve modificado por el material ferroso y detecta así el sensor su presencia.

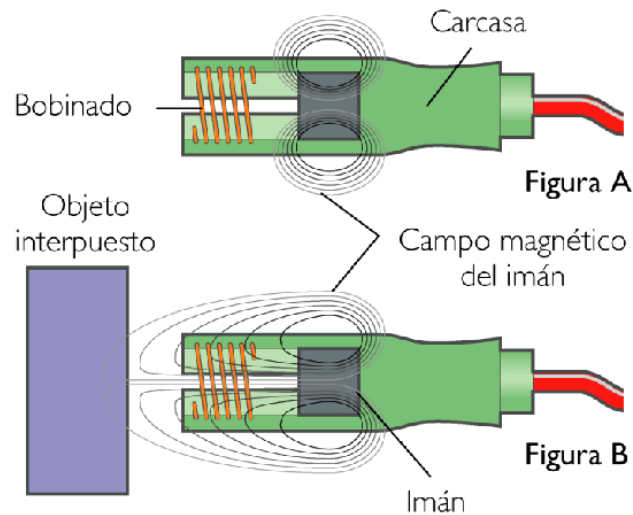


Ilustración 18 - Funcionamiento sensor inductivo (www.temporizadores-sensores.blogspot.com).

Estos sensores aplicados al mundo de la animatrónica se pueden encontrar en diferentes usos, como pueden ser la detección de la posición de partes del esqueleto o la detección de cercanía a un objetivo para comenzar una acción.

La respuesta electrónica de estos sensores es muy sencilla dado que responden con una tensión a nivel alto cuando se detecta el objeto y nivel bajo cuando no haya presencia de ningún objeto. Los sensores se pueden calibrar de modo que detecten el objeto a partir de la distancia deseada por el diseñador.

Estos sensores tienen una salida de 0V para nivel bajo y una salida de entre 5 y 24V, dependiendo del sensor elegido, para nivel alto.

4.2.2.2 Finales de carrera

Estos sensores son muy básicos, simplemente son un interruptor que se cierra cuando un objeto entra en contacto con él por medio de un mecanismo. En función de la

aplicación hay diferentes sensores donde el accionamiento mecánico se adapta al contacto a detectar.



**Ilustración 19 - Conjunto variado de finales de carrera industriales
(www.library.automationdirect.com).**

Este tipo de sensores son muy utilizados en el mundo de la electrónica como medida de seguridad, por ejemplo, para marcar los extremos físicos del movimiento de una extremidad sin que el esqueleto se dañe.

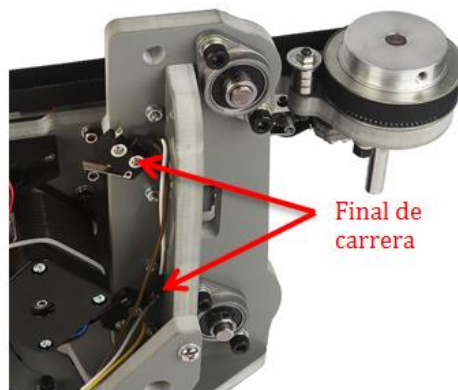


Ilustración 20 - Utilización de finales de carrear como protección de una extremidad mecánica.

La salida de este sensor depende de la alimentación que le proporcione el diseñador, al ser un interruptor abrirá o cerrará el circuito dando un nivel alto o bajo para poder detectar el contacto.

4.2.2.3 Encoders[9]

Los encoders no son un sensor como tal, sino que son un dispositivo que gracias a sus mecanismos electrónicos y mecánicos son capaces de traducir un movimiento giratorio (posición y velocidad) en señales electrónicas.

El mecanismo más usual para estos dispositivos es óptico, que gracias a un disco codificado, emisores y receptores ópticos, son capaces de detectar la posición de giro de un eje.

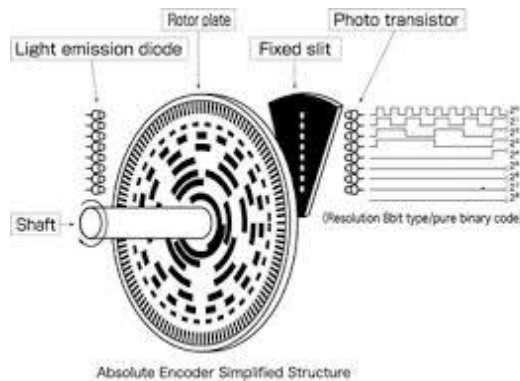


Ilustración 21 - Esquema funcionamiento de un Encoder (www.logicbus.com.mx).

En animatrónica se suele acoplar estos dispositivos en las articulaciones que giran mediante los diferentes actuadores, de modo que, el ordenador de control sabe en todo momento la posición exacta del eje de la articulación y puede controlar el sistema en consecuencia.



Ilustración 22 - Conjunto variando de encoders para uso industrial (www.anaheimautomation.com).

La salida de estos dispositivos suele ser digital en paralelo como se muestra en la ilustración 22. En función de la precisión tiene más o menos bits de resolución en la posición.

4.2.3 Comunicaciones

Dado que uno de los usos más frecuentes de los sistemas animatrónicos es formar parte de rodajes cinematográficos y representación en escenas artísticas, es muy importante que el sistema sea capaz de sincronizarse con el resto de “actores” de la escena, como pueden ser: luces, sonidos, efectos especiales o incluso otros sistemas animatrónicos.

Para ello, es necesario que el sistema disponga en su interfaz de hardware de lo necesario para poder implementar una interfaz multimedia con otros dispositivos.

Sin duda uno de los protocolos más extendidos para este propósito es el DMX512, por ello ahora se va a analizar un poco más a fondo para conocer sus características y uso.

4.2.3.1 Protocolo DMX512[10]

Este protocolo está basado en una configuración de bus, de modo que se puedan añadir dispositivos en paralelo fácilmente, como se muestra en el siguiente esquema:

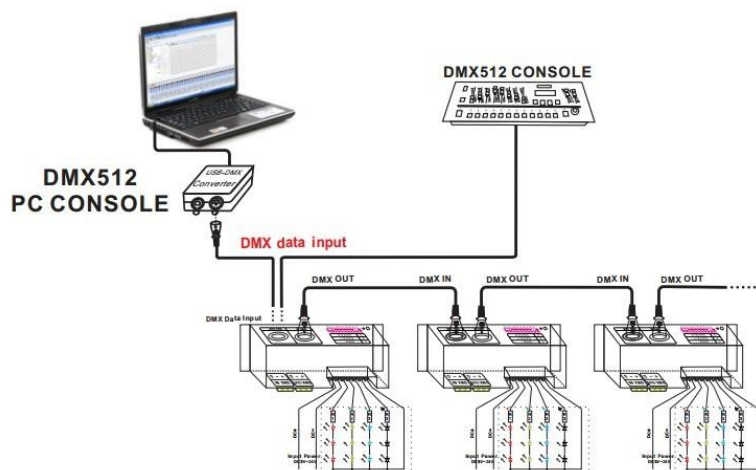


Ilustración 23 - Arquitecturas DMX512 (PHILIPS Introduction to DMX 2008)..

Éste es capaz de controlar hasta un máximo de 512 variables o canales. El valor contenido en cada canal irá desde 0 hasta 255. La “string” que se transmite 40 veces por segundo (+/- 250 kbaud) tiene la siguiente forma:

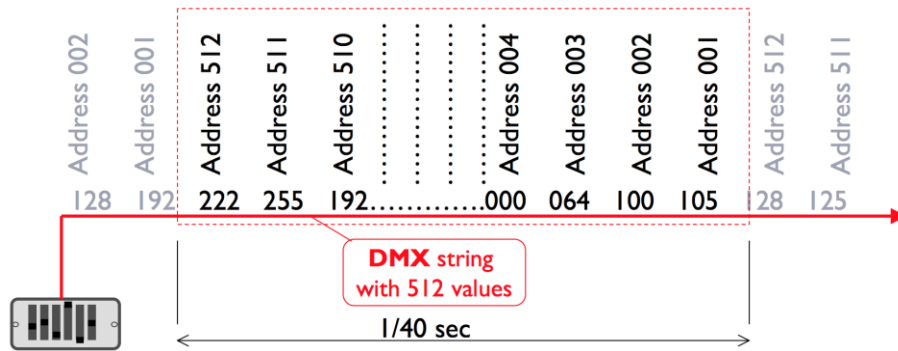


Ilustración 24 - "String" DMX512(PHILIPS Introduction to DMX 2008).

Un ejemplo de utilización de este protocolo para el control de luminarias RGB, se basa en alocar tres direcciones o canales por cada luminaria, de modo que cada canal representa la cantidad de cada color principal (rojo, verde y azul) en la luminaria. A continuación, la ilustración 26 muestra el esquema descriptivo:

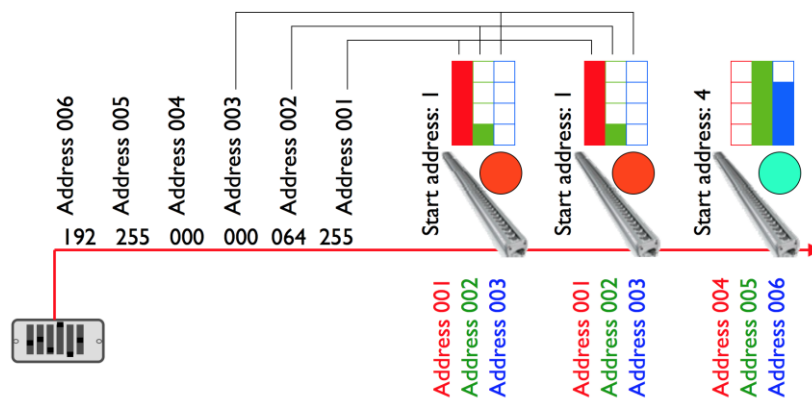


Ilustración 25 - Aplicación DMX512 para control RGB iluminación (PHILIPS Introduction to DMX 2008)..

Respecto al interfaz hardware del protocolo, éste se basa en la capa física RS485 (según el sistema OSI), por lo que el cable que se suele utilizar es un par trenzado de tres hilos apantallado. Por último, el conector establecido para la conexión de estos sistemas es el XLR 5 [11].

5 SOLUCIÓN EN TÉRMINOS DE HARDWARE

5.1 Introducción

Una vez se conocen las opciones y características para el sistema embebido y los actuadores y sensores que deben ser manejados por el sistema de control, se puede entrar en el grueso del diseño del proyecto definiendo la solución que se ha aplicado para cada uno de los puntos a resolver.

En primer lugar, se presentará la solución para el ordenador embebido que más se ajusta a los objetivos del presente proyecto. A continuación, se mostrará cada una de las soluciones aplicadas en la interfaz de hardware, que permitirá el control de todos los periféricos. Por último, se distribuirán todas las entradas y salidas utilizadas en el mapeado del ordenador embebido.

5.2 Ordenador embebido

Tras evaluar todas las características de los ordenadores embebidos presentados en el apartado 4 de este proyecto, se ha decidido elegir el ordenador Embebido “Beaglebone Black” (BBB) como sistema de procesamiento principal de nuestra plataforma animatrónica.

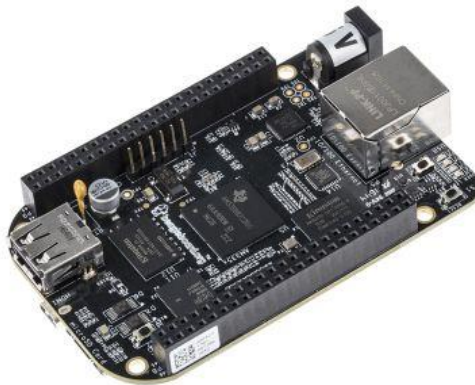


Ilustración 26 - Beaglebone Black (www.beagleboard.org).

Las razones por las que se ha seleccionado la BBB sobre la Raspberry Pi son las siguientes:

- BBB proporciona acceso a un mayor de entradas/salidas, lo que permitirá integrar más periféricos y que el sistema sea flexible a la hora del diseño del sistema animatrónico final.
- El ordenador seleccionado soporta la carga del sistema operativo tanto desde un micro-SD como reprogramando su memoria flash. Esto da robustez al sistema a la hora de resistir vibraciones en el caso de que se embarque el sistema de control dentro del sistema animatrónico.
- Las cabeceras del ordenador ponen a disposición del usuario los pines necesarios para la transmisión de audio por I2S que se explicará posteriormente en el apartado de salida de audio.
- BBB está también preparada para la fácil integración de drivers para protocolos de comunicación RS485.

A pesar de que las características entre ambas opciones son bastante parecidas, los argumentos anteriormente nombrados han hecho decantar la selección.

La plataforma Beaglebone Black es un sistema embebido basado en un procesador ARM que permitirá gestionar sus recursos mediante un sistema operativo Linux en su interior y que de este modo facilitará mucho el desarrollo software y la integración con sistemas multimedia de la escena.

En términos de hardware, la BBB proporciona dos grandes cabeceras donde están accesibles 96 conexiones configurables para el sistema de control. Estas cabeceras están diseñadas para un concepto de pila, lo que facilitará la creación de la interfaz hardware y una conexión robusta entre ellas.

Las cabeceras se distribuyen de la siguiente manera en sus diferentes configuraciones:

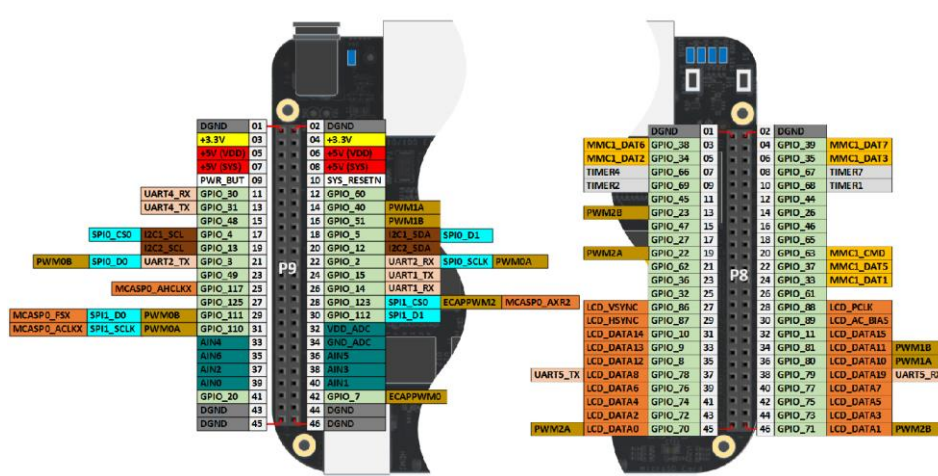


Ilustración 27 - Funcionalidad cabeceras BBB (es.mathworks.com).

5.3 Interfaz hardware

La solución para la interfaz de hardware mostrará como se ha diseñado la PCB con los componentes que permitirán al ordenador embebido controlar del sistema.

Como se ha comentado en el apartado anterior, la BBB está pensada para ser integrada en un concepto de pila, por lo que ésta condicionará en gran medida el concepto de distribución de componentes de la PCB. La PCB dispondrá de dos grandes cabeceras macho de 2x23 pines que encajaran en las cabeceras hembra de la BBB.

Las cabeceras se han situado en la parte central, lo que permitirá integrar el resto de componentes y conectores a ambos lados de la BBB de modo que queden distribuidos de una manera uniforme.

Dado que el sistema que se está diseñando está basado en estándares industriales, a la hora de diseñar la interfaz hardware y la integración del ordenador embebido tenemos que asegurar que todos estos componentes se pueden encapsular en una caja que cumpla estos estándares. [12]

El estándar más extendido en la industria para el montaje de sistemas de control es el armario con carriles DIN TS35 [5], estos carriles permiten el montaje modular de diferentes dispositivos de control. La figura muestra un montaje en armario DIN y las dimensiones de un carril DIN TS35.

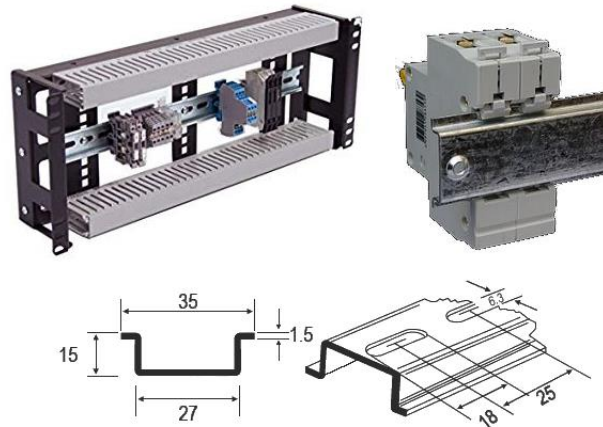


Ilustración 28 - Collage rail DIN.

Las medidas de los módulos que se instalan en armarios DIN tienen también restricciones de tamaño, un máximo de 90 mm en altura y un máximo de 212 mm de anchura [12][13]. Dado que la altura de la BBB es de 86,36 mm deberemos ir al máximo del estándar en lo que se refiere a altura y tras hacer la colocación de todos los componentes también se necesitara la máxima anchura para que haya espacio suficiente para todos los conectores alrededor de la BBB.

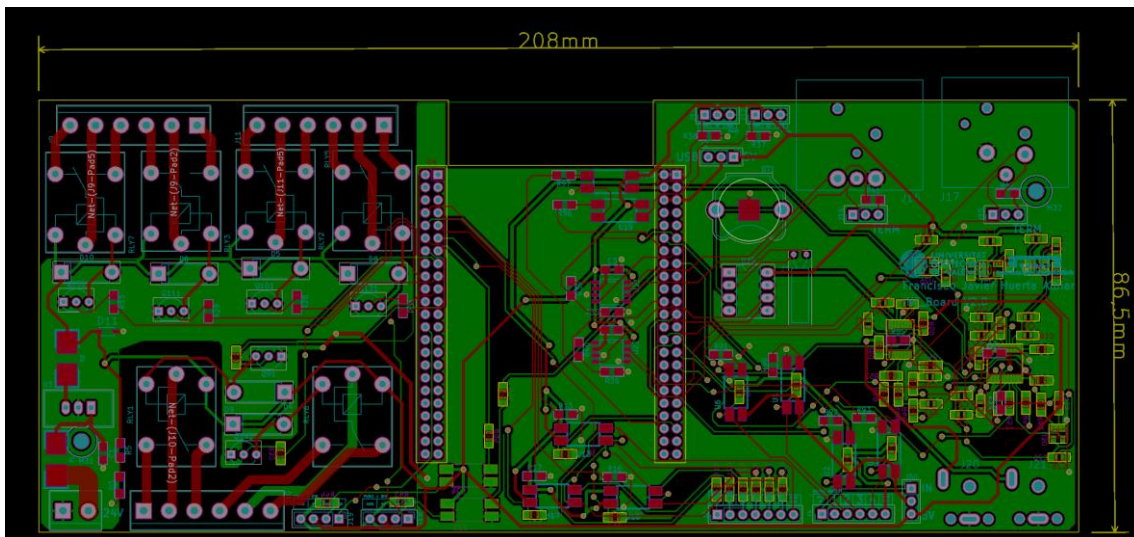


Ilustración 29 - Dimensiones interfaz hardware.

Por estos motivos se ha elegido la caja de montaje DIN de la marca CANDEBOSS con unas dimensiones exteriores de 90 x 212 mm y que permite el montaje de una PCB de un tamaño de 86,5 x 208mm.



Ilustración 30 - Encapsulamiento sistema de control para montaje en carril DIN (es.rs-online.com).

Los conectores se colocarán en las zonas superior e inferior de la PCB dado que es la configuración más utilizada en los montajes industriales.

Antes de entrar en el detalle de la explicación de cada uno de los subsistemas de la interfaz hardware se va a mostrar a continuación un diagrama explicativo de la PCB y su división en subsistemas:

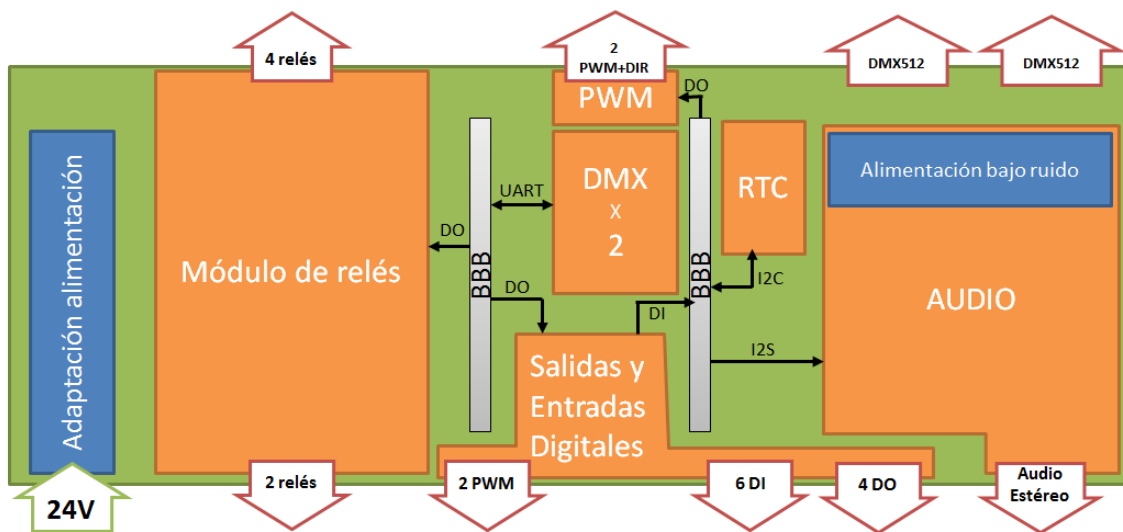


Ilustración 31 - Subsistemas de la interfaz hardware.

A continuación, se explicará cada una de los subsistemas de entradas y salidas que se han integrado en la interfaz hardware en función de su topología y se dará una visión de su posible aplicación para los sistemas animatrónicos.

5.3.1 Alimentación

Dado que se está diseñando un sistema electrónico, el punto del diseño de la alimentación es un punto fundamental. La elección de una alimentación correcta dependerá de las necesidades del sistema y de los componentes utilizados en él.

Tras el análisis del sistema completo, son necesarios tres niveles de alimentación principales en la interfaz hardware; 24 V DC, 5 V DC y 3.3 V DC.

Alimentación 24V

Este nivel de tensión es el más popular y utilizado en el terreno industrial por lo que también será el valor de la alimentación principal del sistema de control.

Para conseguir esta alimentación, partiendo de tensión de red 220 V AC, necesitaremos una fuente de alimentación externa al sistema con salida de 24 V DC. Dado que, como se ha mostrado en el apartado anterior, el sistema está pensado para ser montado en carril DIN, se ha buscado dentro de las fuentes industriales del mercado, una que cumpla las condiciones de:

- Tensión de entrada: 220V AC 50Hz.
- Tensión de salida 24V DC.
- Intensidad mínima 1A.
- Montaje en carril DIN TS35.

Con estas condiciones se ha seleccionado la fuente MRD-20-24 de la marca MEAN WELL. Esta fuente cumple con los requerimientos además de tener un precio aceptable y ser muy compacta.

Adicionalmente, la fuente proporciona entradas y salidas mediante borneras atornilladas, que son muy robustas para montajes industriales, un potenciómetro para ajuste fino del convertidor y por último, una salida de aviso en caso de error.



Ilustración 32 - Transformador MRD-20-24 de Mean Well (Datasheet).

Dado que se trata de un dispositivo externo, debe conectarse a la placa principal de interfaz hardware. Ésto se hará mediante un latiguillo que ira desde las borneras atornilladas de la fuente de alimentación hasta un bloque de terminales de dos polos y paso de 5.08mm. Los terminales macho van atornillado al latiguillo que sale de la fuente de alimentación y los terminales hembra van soldados a la PCB.

De esta manera queda un sistema de conexionado y des-conexionado rápido de la alimentación, lo suficientemente robusto para soportar vibraciones. Además, al haber seleccionado una fuente externa, podrá ser sustituida en el futuro por una que pueda entregar más intensidad en el caso de que el sistema lo necesite o asegurar una reparación rápida si falla en algún momento.

Alimentación 5V

Una vez se dispone de los 24 V en la PCB, se debe adecuar la entrada con protecciones y convertirla al siguiente nivel de tensión necesaria, 5 V, que será un de los principales niveles de alimentación de la electrónica de la PCB.

Para convertir la tensión de 24 V a 5 V se ha utilizado un convertidor DC/DC TSR 2-2450 de la compañía TRACOPOWER, que nos ofrece hasta 2 A la salida.



Pin-Out	
Pin	Single
1	+Vin
2	GND
3	+Vout

Ilustración 33 - Conversor TSR 2-2450 Traco Power (Datasheet).

Este convertidor, además de cumplir con las especificaciones, proporciona protección contra cortocircuito y ruido en la salida.

A la entrada se ha colocado un condensador de 22 μ F/50V como recomienda el datasheet para entradas mayores de 20V. Adicionalmente, se han colocados dos portafusibles a la entrada y salida del regulador para proteger al sistema de sobretensiones y que ésto no produzca un error en todo el sistema. Para la alimentación se ha elegido fusibles 145 OMNY-BLOK de 3 A.

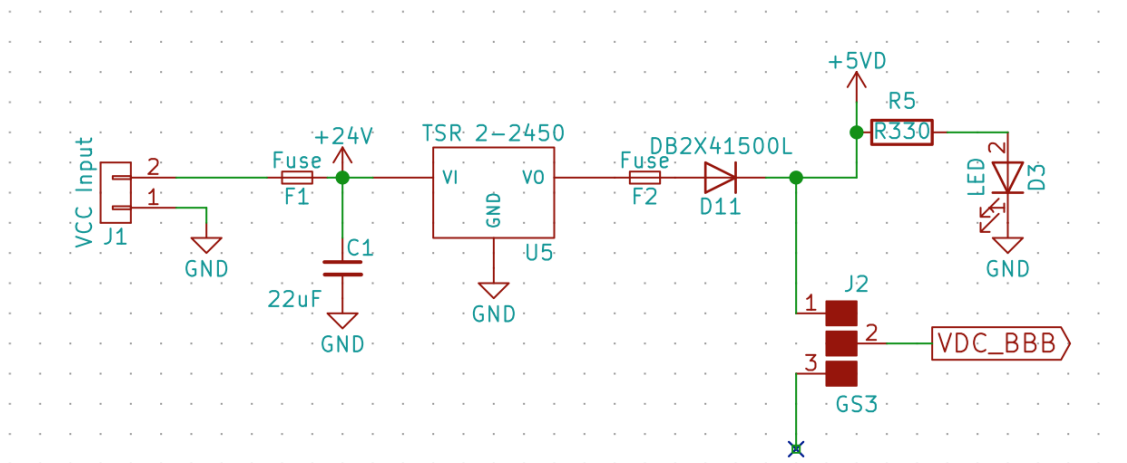


Ilustración 34 - Esquemático alimentación interfaz hardware.

A la salida también se ha añadido un diodo que asegura la protección contra corrientes inversas.

La alimentación de 5 V es necesaria para alimentar diferentes integrados que se verá a posteriori, pero el componente principal que cuelga de esta alimentación es el ordenador embebido. Éste requiere una alimentación mínima de 5V y 1A en algunos de sus puertos de entrada.

La BBB nos proporciona cuatro formas diferentes para alimentarla: por un conector específico, mediante una batería LiPo, por el USB y la última y la que se ha utilizado en el proyecto, mediante los pines 5 y 6 de la cabecera P9.

Por último, se ha añadido en este subconjunto un LED verde de manera que se pueda verificar de forma visual y rápida que el sistema está alimentado y funciona correctamente.

Alimentación 3,3V DC

A pesar de que la BBB tiene una tensión de alimentación de 5 V, su procesador trabaja con un nivel de tensión de 3,3 V, por lo que la BBB tiene internamente un convertidor DC/DC de 5V a 3,3V.

Debido a que el nivel de trabajo de la BBB para sus entradas y salidas es de 3.3V, muchos de los integrados que interactuarán con ella también necesitarán esta misma alimentación.

Para este nivel de alimentación utilizaremos el convertidor interno de la BBB que proporciona 3.3V con un máximo de 400mA en los pines 3 y 4 de la cabecera P9.

Adicionalmente, habrá en el subsistema de audio otros tres reguladores DC/DC de 5V a 3.3V pero esto se explicará en el apartado de salida de audio.

5.3.2 Entradas digitales

Una interfaz básica y que sin duda debe de contemplarse en cualquier sistema animatrónico son las entradas digitales. Estas permitirán al sistema la conexión de todo aquel dispositivo externo que tenga una respuesta digital basada en una nivel alto y bajo de tensión.

El GPIO del procesador de la BBB trabaja con un nivel de tensión de 3.3V que está muy limitado para los diferentes dispositivos disponibles en el mercado y mucho más en el caso de componentes industriales.

Por ello, cada entrada digital necesitará en la interfaz hardware un circuito que adecue las señales de entrada a los niveles lógicos de la BBB.

Como se ha comentado en el apartado 5, cuando se habla de dispositivos con salida digital los niveles de tensión más utilizados son 24 VDC en el terreno industrial y 5 VDC en otros dispositivos de consumo, por ello se harán los cálculos de este circuito con el fin de que se asegure un nivel alto para la BBB entre 5 y 24 VDC.

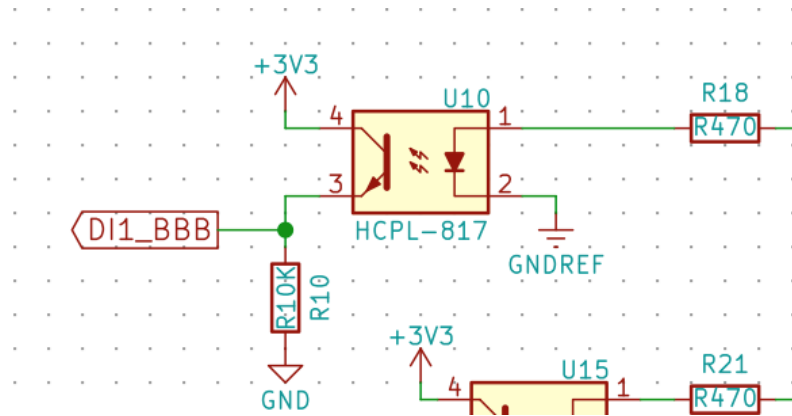


Ilustración 35 - Esquemático acondicionamiento entradas digitales.

Adicionalmente, dado que es un estándar para los PLC y con el fin de asegurar la protección de toda la interfaz hardware, se han opto-acoplado las entradas, de modo que un valor de tensión o intensidad incorrecto no dañe el resto de la PCB.

El valor de la resistencia R18 del ejemplo superior depende del optoaislador y de la tensión que se quiere aplicar a la entrada. Con el valor de corriente que debe circular por el transistor de salida y con ayuda del manual del optoaislador, obtendremos la corriente que debe pasar por el LED y la tensión que produce en el LED esa corriente. Esta corriente debe ser suficiente para saturar al transistor y no ser excesivamente grande para no quemar el LED ni tener un consumo absurdo [6]. Bajo estas premisas hacemos el siguiente cálculo:

$$V_{LED} = 1,2 V \quad I_{LED} = 20mA \quad I_{max}^{LED} = 50mA \quad I_{min}^{LED} = 5mA \quad V_{in} = 5 - 24V$$

$$R18_{max} = \frac{V_{max}^{in} - V_{LED}}{I_{LED}} = \frac{24 - 1,2}{20 \cdot 10^{-3}} = 1K14\Omega$$

$$R18_{min} = \frac{V_{max}^{in} - V_{LED}}{I_{max}^{LED}} = \frac{24 - 1,2}{50 \cdot 10^{-3}} = 456\Omega$$

$$R18 = 470\Omega$$

Una vez seleccionada la resistencia R18, se debe comprobar que con la entrada mínima en este caso 5 VDC, la intensidad será suficiente para activar el diodo:

$$V_{min}^{in} = I_{min}^{LED} \cdot R18 + V_{LED} = 5 \cdot 10^{-3} \cdot 470 + 1,2 = 3,55 \text{ V}$$

Ya sólo queda calcular la resistencia R10. Esta resistencia no debe ser muy baja, pues el transistor no entraría en saturación, ni demasiado alta, pues si a nivel alto la entrada del micro acepta algo de corriente, ésta podría provocar una caída de tensión en la resistencia que bajaría la tensión para el nivel alto [6].

$$V_{ce SAT} = 0,2V \quad \beta_{Transistor} = 1 \quad V_c = 3,3V$$

$$R10_{min} = \frac{V_c - V_{ce SAT}}{I_B \cdot \beta_{Transistor}} = \frac{3,3 - 0,2}{20 \cdot 10^{-3} \cdot 1} = 155 \Omega$$

Una vez verificado estos tres aspectos, se toman los valores 470 Ohm para todas las resistencias de entrada y 10K Ohm como resistencia “pull-down” para la entradas del procesador.

Por otro lado y para asegurar el aislamiento de las entras digitales respecto a la interfaz hardware, se requerirá en el conector de entrada de la masa de referencia que será diferente a la masa de todo el sistema.

Los conectores seleccionados para esta topología serán borneras atornilladas fijas de paso 2,54 mm, esto permitirá un conexionado robusto y a la vez flexible a la hora de hacer el montaje en un cuadro DIN.

Por último, se darán algunos ejemplos de los potenciales usos de estas entradas digitales:

- Detección de finales de carrera como límites mecánicos en movimientos.
- Detección de presencia de objetos con sensores capacitivos, inductivos u ópticos.
- Lectura de salidas digitales de encoders en articulación y ejes de giro.
- Detección de posición en pistones neumáticos para extremidades.

Éstos son algunos de los posibles usos que se les puede dar a estas entradas digitales, pero hay una infinidad de posibilidades según la necesidad del diseñador.

5.3.3 Salidas digitales

Otra interfaz fundamental, al igual que las entradas digitales, son las salidas digitales. Estas permitirán la activación o excitación de dispositivos externos que estén preparados con entradas digitales.

Como se ha explicado en el caso de las entradas digitales, el rango de trabajo del procesador de 3.3 VDC es muy limitado para interactuar con dispositivos externos como los industriales, por ello, se definirá un rango de salida igual que el anterior, entre 5 y 24 VDC, lo que permitirá un abanico mucho más amplio de dispositivos compatibles con el sistema. Para ello se ha integrado el siguiente sistema.

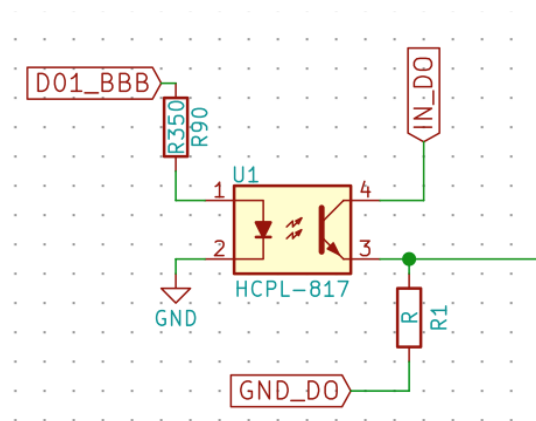


Ilustración 36 - Esquemático acondicionamiento salidas digitales.

Al igual que en el caso anterior, se ha utilizado un opto-acoplador para aislar las salidas. En este caso se debe de calcular R90, de modo que una salida del procesador a 3.3 VDC pueda excitar el diodo, y R1, para que limite la intensidad de la salida en todo el rango de IN_DO de modo que el transistor se pueda saturar correctamente.

$$V_{BBB} = 3,3V \quad I_{max}^{BBB} = 6mA \quad V_{LED} = 1,2V$$

$$R1 = \frac{V_{BBB} - V_{LED}}{I_{max}^{BBB}} = \frac{3,3 - 1,2}{6 \cdot 10^{-3}} = 350\Omega$$

El valor seleccionado para la resistencia de salida del procesador R90 es de 350 Ohm y el de la resistencia R1 que limita la intensidad es de 1,2K Ohm.

En este caso, para asegurar el aislamiento al igual que las entradas digitales, será necesario que se disponga de la masa de referencia (GND_DO) en el conector de entrada.

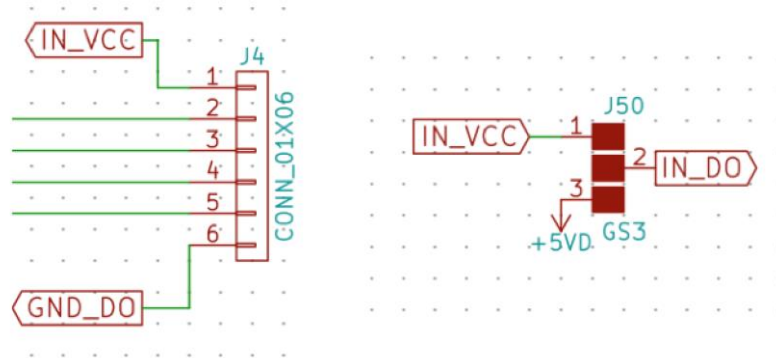


Ilustración 37 - Esquemático asignación de salidas digitales.

Adicionalmente se ha planteado un sistema para que el usuario pueda seleccionar el nivel de las salidas, así como para que no consuma corriente del sistema. Esto se consigue disponiendo de una entrada libre en el conector de las salidas digitales, donde el diseñador puede aportar el nivel de tensión deseado a nivel alto. Además, mediante un “jumper” se puede seleccionar el nivel de tensión 5VDC disponible en la interfaz hardware.

Por último, al igual que en el caso anterior, se han elegido unas borneras atornilladas de paso 2.54 mm para esta topología.

Esta topología da un gran abanico de posibles aplicaciones en animatrónica. A continuación nombramos algunas de ellas:

- Activación de solenoides de hasta 24 VDC para manejo de neumáticos de carga baja o media en extremidades.
- Activación de iluminación o indicadores.
- Enviar señales de coordinación con otros elementos de la escena.
- Activar válvulas para control de fluidos o gases que se pueden utilizar para efectos especiales en escena como llamaradas.

5.3.4 Relés

Los relés van un paso más allá que las salidas digitales. Mientras que las salidas digitales están limitadas a un rango de tensión continua entre 5 y 24 V, los relés no solo permiten un nivel de tensión mucho mayor, sino que también son capaces de manejar intensidades mucho más altas incluso para señales alternas.

Los relés permiten conmutar cualquier señal aportada por los conectores del sistema, tanto en configuración normalmente abierto (“normally open” o NO) como en normalmente cerrado (“normally close” o NC), de modo que se pueda elegir que la señal de control corte o cierre la señal de entrada por el terminal común.

Dado que la corriente que puede aportar el procesador como la tensión son muy bajas para poder excitar la bobina del relé, también será necesario un circuito de adaptación en la interfaz hardware que se explicará a continuación.

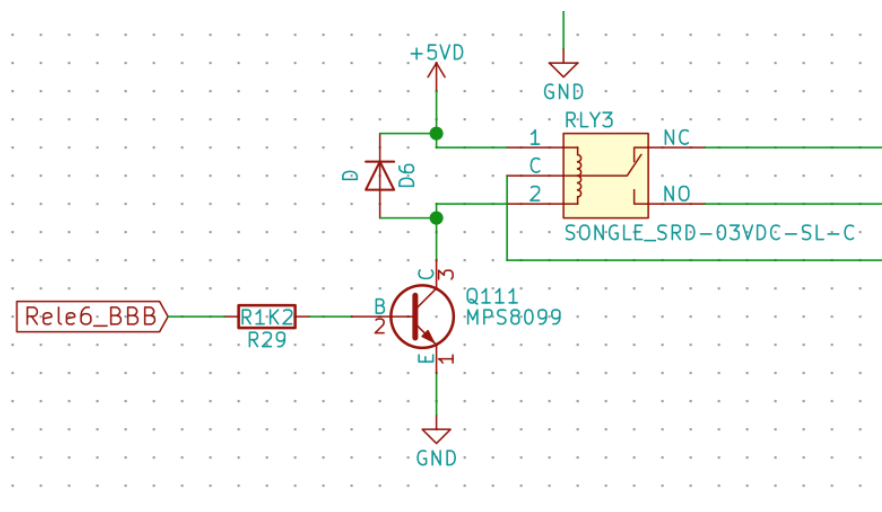


Ilustración 38 - Esquemático acondicionamiento relés.

Este circuito cuenta de dos partes principales, una es el relé, que como se ha comentado es el responsable de conmutar la señal de entrada y otro es el transistor encargado de conmutar la corriente para excitar la bobina del relé.

En primer lugar, se ha seleccionado un relé SONGLE_SRD que permite conmutar señales alternas de 240 VAC y 7 A o continuas de 28 VDC y 10 A. Además la bobina del relé se puede excitar con tensiones a partir de 3 VDC. Este cumple con las necesidades del proyecto pudiendo así conmutar señales de red eléctrica.

Respecto al transistor, se ha seleccionado un transistor MPS8099 que trabajará en régimen de corte y saturación para conmutar la corriente que activa la bobina del relé. Para asegurar este régimen de trabajo, se ha calculado la resistencia R29 de modo que se satura el transistor cuando la salida de la BBB está a 3.3 VDC.

$$I_{bobina} = 90mA \quad V_{CE\ sat} = 0,4V \quad V_{BE} = 0,8V \quad \beta = 75 \quad V_{BBB} = 3,3V$$

$$I_B > \frac{I_{bobina}}{\beta} = \frac{90 \cdot 10^{-3}}{75} = 1,2mA; \quad I_B \cong 2mA$$

$$R29 = \frac{V_{BBB} - V_{BE}}{I_B} = \frac{3,3 - 0,8}{2 \cdot 10^{-3}} = 1K25\Omega$$

El valor comercial que se tomara para la resistencia R29 será de 1.2 K Ohms.

Es importante puntualizar que se ha utilizado una alimentación de 5 VDC para la bobina a pesar de que soporta 3.3 VDC para optimizar la intensidad que consume la bobina.

Por último, el diodo D6 debe ponerse siempre en paralelo con la bobina del relé (para bobinas alimentadas con corriente continua). Este diodo se utiliza dado que al pasar el transistor de saturación a corte hay un cambio muy brusco de la corriente que pasa por el colector y por la bobina, el cual genera una tensión muy elevada en sus bornas. El diodo permite que al cortar el transistor, la corriente que pasa por la bobina siga circulando por el diodo y no dañe así la electrónica de la interfaz hardware.

Dado que en este caso se está tratando con señales de mayor tensión e intensidad, los conectores que se han seleccionado para esta topología son bloques de terminales atornillados con un paso de 5.08 mm que permiten una manipulación más segura gracias a la posibilidad de desconectar el bloque de terminales antes de manipularlo.

Algunas de las aplicaciones para este tipo de interfaz de hardware en animatrónica son:

- Conmutación de iluminación a 220 VAC.
- Activación de solenoides para válvulas de actuadores neumáticos con una gran carga.
- Computación de alimentación de motores para mecanismos que necesiten mucho par.

5.3.5 Salidas PWM

Una vez se dispone de entradas y salidas digitales que proporcionan un gran abanico de posibilidades en el control de periféricos, se debe pensar en una interfaz que permita control de elementos más complejos.

Como se ha visto en el apartado anterior, una de las interfaces fundamentales para el control de actuadores giratorios o motores es una señal PWM. Por ello, se introducirá esta interfaz en el diseño del presente proyecto. La señal PWM permitirá la gestión de servomotores, que proporcionan en sistemas animatrónicos movimiento de precisión.

Para implementar esta interfaz en el hardware se deben utilizar las salidas destinadas a este propósito (unidad PRU) del ordenador embebido, de modo que un módulo específico del procesador gestione la modulación de ancho de pulso dándole simplemente el ciclo de trabajo deseado.

Será necesaria la adaptación de esta señal ya que la BBB sólo proporciona salidas de 3.3 VDC y normalmente son requeridos 5 VDC a nivel alto para la mayoría de dispositivos.

Para ello se ha implementado el siguiente circuito en la interfaz hardware:

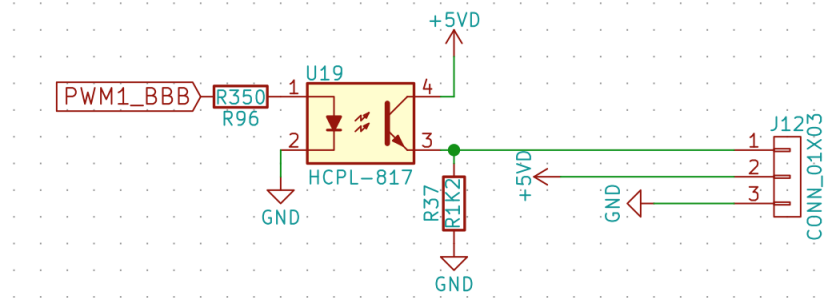


Ilustración 39 - Esquemático acondicionamiento salidas PWM.

Al igual que en las salidas digitales, se ha utilizado un octoacoplador para aislar el circuito del entorno al que se conecte y además permitirá conmutar la señal de 5VDC, adaptando así una señal PWM de 3.3 a 5VDC como es requerido.

El cálculo de las resistencias R94 y R43 es el mismo que el de las salidas digitales, por lo que los valores deben ser coincidentes.

Adicionalmente, dado que es un estándar para la gestión de servomotores, se han agrupado las salidas PWM en un conector de 3 pines junto una salida de 5VDC de alimentación y una masa. De este modo se dispondrá en este conector de todo lo necesario para gestionar el servo.

También se ha pensado en otra posible aplicación final, dándole a la función PWM la labor de transmitir a un variador la velocidad de giro de un motor. En este caso es necesario también una salida digital adicional para indicar al variador el sentido de giro del motor. En este caso, se han agrupado todas las señales en un conector de cuatro pines como se muestra a continuación.

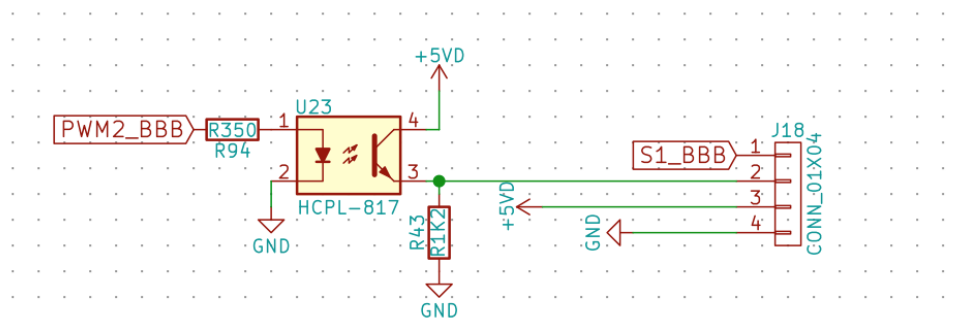


Ilustración 40 - Esquemático acondicionamiento salidas PWM con sentido de giro.

Con estos dos tipos de configuraciones se podrá sacar todo el partido a las señales PWM y ofrecer al usuario la flexibilidad para su aplicación en el sistema.

5.3.6 Salida de audio

En mundo del espectáculo y la cinematografía, además de los actores, hay dos elementos fundamentales en la escena, como son el sonido y la iluminación. En este apartado se explicará el hardware necesario que se ha implementado en la interfaz hardware para otorgar al sistema de la posibilidad de generar señales de sonido analógico.

Una de las limitaciones de la BBB es que no dispone de una salida de audio analógica directa, por lo que se debe buscar una alternativa para poder generar la interfaz necesaria.

Por otro lado, la BBB si dispone de una interfaz digital de imagen y audio HDMI que reaprovecharemos para poder conseguir la salida de audio analógica. El procesador de la BBB tiene un puerto de comunicación I2S que le permite transmitir señales de audio digitales a otros integrados como el de la gestión del HDMI.

El bus I2S es utilizado para comunicación de sonido digital entre integrados. Éste está basado en tres líneas de comunicación: reloj, selección de palabra y datos. A continuación se ilustra el conexionado y la transmisión de cada línea.

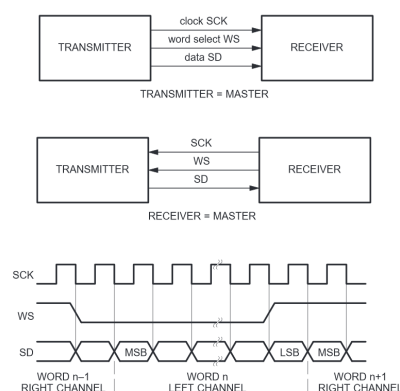


Ilustración 41 - Comunicación I2S (<http://iot-bits.com/interfacing-an-audio-codec-with-esp32>).

En la interfaz hardware utilizaremos una conversión digital analógica que convierta la señal I2S en una señal analógica. De este modo ya se dispondrá de una señal analógica de audio como es requerido, pero dado que el conversor da una señal analógica muy débil, será necesaria una segunda etapa de amplificación que adecue la señal de audio para la conexión con equipos periféricos.

A continuación se muestra el esquemático para el circuito de conversión digital analógico, donde se han conectado los pines 28, 29 y 31 de la cabecera P9 de la BBB al DAC.

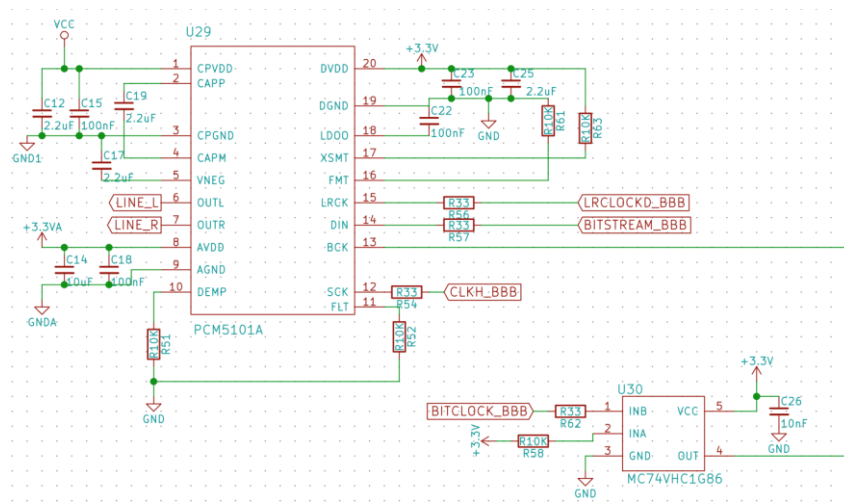


Ilustración 42 - Esquemático DAC.

Por último, se toman las señales analógicas de bajo nivel LINE_L y LINE_R y se pasan por el amplificador tal y como se muestra a continuación.

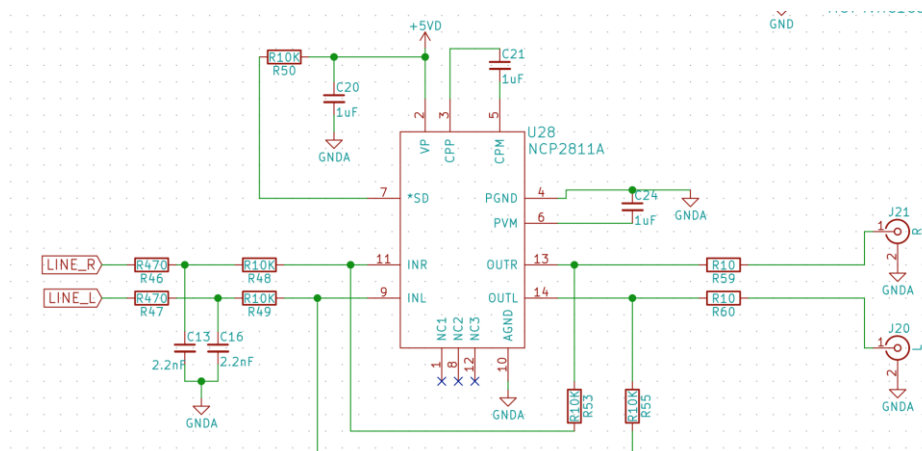


Ilustración 43 - Esquemático amplificador de audio.

Tras amplificar la señal, se pondrá a disposición del usuario a través de dos conectores RCA que son estándar para salidas de audio analógicas estéreo.

En esta interfaz hay que destacar que se ha implementado una alimentación específica debido a la naturaleza de la misma. Las señales de sonido son muy susceptibles a señales de ruido y afectan en gran medida a la calidad del audio a la salida, por ello, se han implementado tres convertidores REG101 DC/DC de 5VDC a 3,3VDC que son específicos para sistemas en los que evitar el ruido es fundamental.

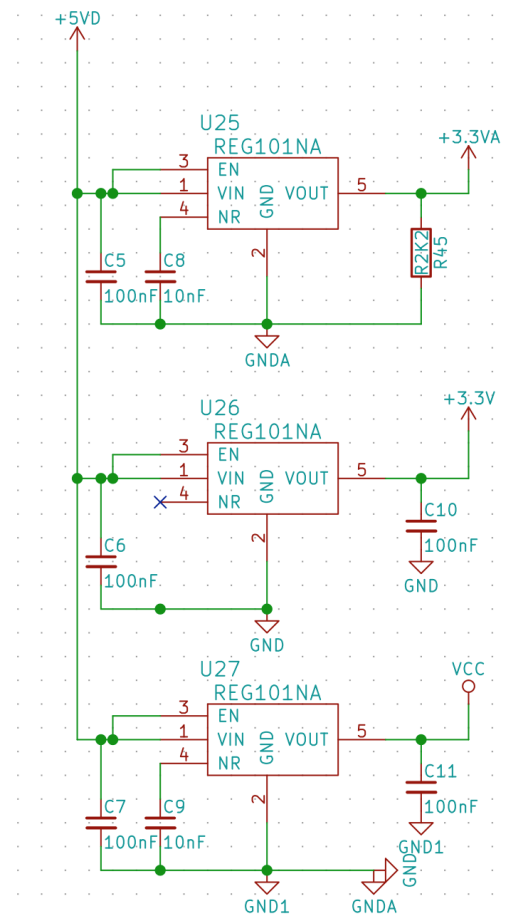


Ilustración 44 - Esquemático transformadores 3,3V de bajo ruido.

Se han implementado dos reguladores diferentes que independizan las alimentaciones analógicas y digitales del DAC de audio y un tercero que da soporte a la BBB para la alimentación general de integrados a 3,3 VDC.

5.3.7 Interfaz DMX

Por último, se expondrá una interfaz enfocada al segundo elemento principal de la escena, la iluminación. Como se ha mostrado en el apartado anterior la interfaz de comunicación más utilizada para la gestión de estos sistemas es el DMX512.

Dado que la capa física de este protocolo es RS485, el circuito que se ha implementado en la interfaz hardware está basado en un “transciever” MAX485 que convertirá las señales RX y TX del micro a una señal diferencial según el estándar de esta capa física.

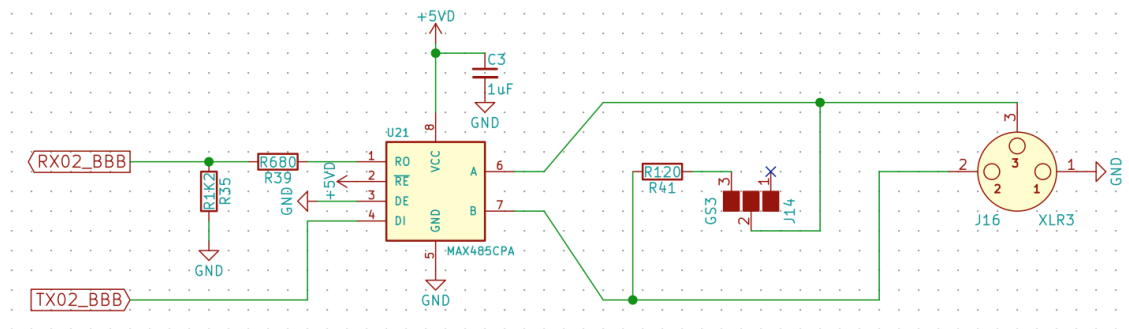


Ilustración 45 - Esquemático salida DMX.

El circuito seleccionado soporta entradas a nivel alto a partir de 2VDC, por lo que los 3.3VDC a los que trabaja la BBB serán suficientes para que sean reconocidos por el integrado. Por otro lado, el MAX485 tiene una salida a nivel alto de 5VDC, por lo que deberemos adaptar la señal RX mediante un divisor de tensión para que la salida máxima sea 3.3VDC.

Adicionalmente se añadira una resistencia de 120 Ohm con un “jumper” que permitirá activar o desactivar la resistencia terminadora del bus en el caso de que nuestra configuración lo necesite.

En el caso de este protocolo de comunicación, el conector estándar que se introducirá en la PCB es un conector XLR de tres contactos.

5.4 Distribución entradas/salidas del sistema

Una vez está definido todo el hardware necesario que se tiene que implementar en la interfaz con el sistema de procesamiento, y dado que el sistema que se quiere desarrollar en este proyecto debe ser lo más flexible posible, se debe de definir el máximo

número de entradas y salidas útiles posibles que acepte el sistema de procesado, teniendo en cuenta las necesidades y las limitaciones.

Como se ha comentado al principio del presente proyecto, no solo se debe de poner a disposición del usuario del hardware sino también de un software básico que permita la programación del sistema. Es por esto que las limitaciones no sólo vienen dadas por el hardware en si, sino que también el software que se va a implementar sobre la BBB que tiene sus propias limitaciones. En este caso el “runtime” de CODESYS que nos permitirá programar el sistema animatrónico y que se tratara en profundidad en el apartado seis, sólo soporta el acceso a algunas entradas y salidas de la BBB. A continuación, se muestran las entradas y salidas soportadas por CODESYS.

Soportado	P9		Soportado
GND	1	2	GND
DC_3.3V	3	4	DC_3.3V
VDD_5V	5	6	VDD_5V
SYS_5V	7	8	SYS_5V
PWR_BUT	9	10	SYS_RESETn
SI	11	12	SI
SI	13	14	SI
SI	15	16	SI
NO	17	18	NO
NO	19	20	NO
SI	21	22	SI
SI	23	24	SI
NO	25	26	SI
SI	27	28	NO
NO	29	30	NO
NO	31	32	NO
NO	33	34	NO
NO	35	36	NO
NO	37	38	NO
NO	39	40	NO
SI	41	42	SI
GND	43	44	GND
GND	45	46	GND

Soportado	P8		Soportado
DGND	1	2	DGND
NO	3	4	NO
NO	5	6	NO
SI	7	8	SI
SI	9	10	SI
SI	11	12	SI
SI	13	14	SI
SI	15	16	SI
SI	17	18	SI
SI	19	20	NO
NO	21	22	NO
NO	23	24	NO
NO	25	26	SI
NO	27	28	NO
NO	29	30	NO
NO	31	32	NO
NO	33	34	NO
NO	35	36	NO
NO	37	38	NO
NO	39	40	NO
NO	41	42	NO
NO	43	44	NO
NO	45	46	NO

Tabla 2 - Salidas soportadas por CODESYS.

Teniendo en cuenta esto y con las alternativas de configuración que proporciona el hardware de la Beaglebone Black, se han implementado las siguientes entradas y salidas:

- 6 entradas digitales entre 5 y 24V DC.
- 4 salidas digitales configurables.
- 2 salidas PWM.
- 2 salidas PWM + salida digital (control de motores con sentido de giro).
- 1 salida de audio analógica estéreo.
- 6 salidas de relé.
- 2 puertos DMX512.

Quedando la distribución sobre la Beaglebone de la siguiente manera:

P9			P8		
GND	1	2 GND	DGND	1	2 DGND
DC_3.3V	3	4 DC_3.3V	eMMC	3	4 eMMC
VDD_5V	5	6 VDD_5V	eMMC	5	6 eMMC
SYS_5V	7	8 SYS_5V	DI4_BBB	7	8 DI5_BBB
PWR_BUT	9	10 SYS_RESETn	DI6_BBB	9	10 DI3_BBB
RX02_BBB	11	12 DO1_BBB	DI2_BBB	11	12 Rele1_BBB
TX02_BBB	13	14 PWM1_BBB	PWM4_BBB	13	14 Rele2_BBB
DO2_BBB	15	16 PWM2_BBB	Rele3_BBB	15	16 Rele4_BBB
PINMUX I2C	17	18 PINMUX I2C	Rele5_BBB	17	18 Rele6_BBB
PINMUX I2C	19	20 PINMUX I2C	S1_BBB	19	20 eMMC
TX03_BBB	21	22 RX03_BBB	eMMC	21	22 eMMC
DO3_BBB	23	24 I2C_SCL	eMMC	23	24 eMMC
MCASPO(CLK_BBB)	25	26 I2C_SDA	eMMC	25	26 S2_BBB
DO4_BBB	27	28 MCASPO(BITSTREAM_BBB)	HDMI	27	28 HDMI
MCASPO(LRCLOCK_BBB)	29	30 MCASPO	HDMI	29	30 HDMI
MCASPO(BITCLOCK_BBB)	31	32 VADC	HDMI	31	32 HDMI
AIN4	33	34 AGND	HDMI	33	34 HDMI
AIN6	35	36 AIN5	HDMI	35	36 HDMI
AIN2	37	38 AIN3	HDMI(TX01_BBB)	37	38 HDMI(RX01_BBB)
AIN0	39	40 AIN1	HDMI	39	40 HDMI
DI1_BBB	41	42 PWM3_BBB	HDMI	41	42 HDMI
GND	43	44 GND	HDMI	43	44 HDMI
GND	45	46 GND	HDMI	45	46 HDMI

Tabla 3 - Asignación de funciones a la cabecera de la BBB.

Como se puede observar en la tabla, se han utilizado todas las salidas/entradas disponibles y 4 salidas específicas (P9_25,28,29,31) para el protocolo de comunicación I2S, que a pesar de que no sean soportadas/configurables, si permitirá generar señales de audio digitales.

Por último, se muestra un esquema de cómo se han implementado estas entradas y salidas en la interfaz hardware de manera que el usuario conozca la funcionalidad de cada conector y el uso de cada pin.

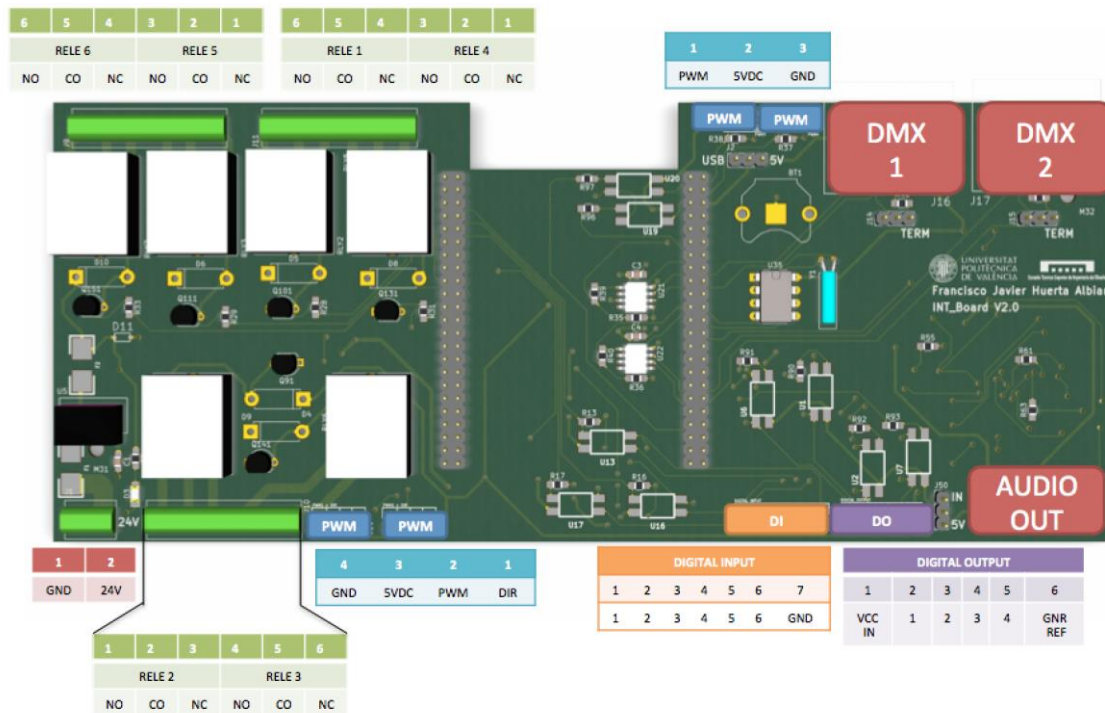


Ilustración 46 - Distribución entradas y salidas interfaz hardware.

Adicionalmente, confirmar que se han añadido 4 “jumpers” en el sistema que permiten las siguientes configuraciones:

- J50- Permite configurar el nivel de las salidas digitales:
 - IN: La salida a nivel alto es el nivel de tensión proporcionado por el pin 7 de la cabecera de salida digitales.
 - 5V: La salida a nivel alto son los 5V del sistema.
- J2- Define si la alimentación de la BBB son los 5V del sistema o se alimenta a través de USB.
- J14 & J14- Activan o desactivan la resistencia terminadora para el bus DMX.

6 ÁMBITO DE APLICACIÓN Y ALTERNATIVAS DE DISEÑO PARA EL SOFTWARE

Una vez está definido el hardware del sistema, se puede empezar a definir el software con el que se quiere programar el funcionamiento del mismo. Como comentado en apartados anteriores, la funcionalidad principal del sistema de control es gestionar las entradas y salidas para conseguir el comportamiento deseado del conjunto y esto no sería posible sin la implementación del software necesario.

Dado que uno de los objetivos del proyecto es verificar la funcionalidad del mismo y dejar un plataforma completamente lista para su programación he implementación, se deben definir algunos de los parámetros de programación del software.

En este apartado se dará una visión general de las diferentes formas de programar la BBB y los lenguajes de programación disponibles, para más tarde en el apartado siguiente, mostrar la solución adoptada en estos términos.

6.1 Plataformas de programación

Dado que la BBB ya dispone de un sistema operativo basado en Linux facilita en gran medida la compatibilidad con muchas plataformas de programación disponibles, o incluso, la programación de manera nativa.

A continuación, se trataran las condiciones de diseño para este componente del proyecto, y a su vez, las plataformas para la programación de sistemas embebidos basados en Linux más comunes.

6.1.1 Condiciones de diseño

El factor principal para todas las decisiones en términos de software debe ser, al igual que en el caso del hardware, la flexibilidad y la fácil implementación del mismo. Adicionalmente, en este caso también se debe tener en cuenta que la puesta en marcha o configuración de la plataforma tenga una curva de aprendizaje rápida y que proporcione al usuario un software funcional y robusto en un tiempo razonable.

Por otro lado y dado que a pesar de que se trate de un sistema automático de control, muchas veces será necesario la interacción con el usuario para modificar el proceso o para optimizar la programación, por lo que la posibilidad de la implementación de una interfaz gráfica también será valorada.

Por ello, se van a evaluar los siguientes aspectos de las diferentes alternativas:

- Dificultad de configuración y puesta en marcha.
- Diferentes lenguajes de programación soportados.
- Curva de aprendizaje de la plataforma.
- Posibilidad de desarrollo GUI.
- Se valorara positivamente que cumpla con estándares industriales.

6.1.2 Alternativas de diseño

Plataformas basadas en un “toolchain” y compilación cruzada

Hay diferentes plataformas en el mercado que permiten la configuración basada en un “toolchain” y que trabajan con el concepto de compilación cruzada.

Aquí aparecen dos conceptos muy importantes para los sistemas embebidos basados en sistemas operativos y que se definirán brevemente a continuación.

En primer lugar está el “toolchain”, éste es un conjunto de herramientas de desarrollo software que están unidas por un estándar (por ejemplo GCC). Las herramientas que se encuentran en este paquete son entre otras el compilador y “debugger”, que permitirán a la plataforma de programación crear un código y compilarlo para el sistema embebido deseado [15].

Dado que la mayoría de las veces la programación se hace en un dispositivo en el que la arquitectura software o hardware no es compatible con la plataforma que se quiere programar, es necesario utilizar el método de compilación cruzada.

La compilación cruzada consiste en la generación de un código ejecutable bajo una determinada arquitectura que es diferente a la encargada de su ejecución [20].

Esto permite al IDE (“Integrated Development Environment”) realizar la compilación en la plataforma hardware objetivo. De modo que toda la gestión y debug se muestra en ordenador del usuario a pesar de que el procesamiento y compilación se está haciendo en el sistema huésped [16].

A continuación se evalúan las características anteriormente nombradas:

-Dificultad de configuración y puesta en marcha: La configuración de un toolchain y hacer funcionar la compilación cruzada no es fácil, a pesar de que los IDE ofrece herramientas que facilitan la configuración.

-Lenguajes de programación: Esto depende directamente del “toolchain”, cada configuración del IDE basado en un “toolchain” sólo será compatible con un lenguaje de programación. Si se quiere cambiar el lenguaje de programación, se debe empezar desde cero con la configuración y programación.

-Curva de aprendizaje: Lenta

-Desarrollo GUI: La programación de GUIs con casi todas las IDEs es posible pero no dispone de objetos nativos de la plataforma, por lo que se deben buscar librerías orientadas a gráficos.

Programación en la nube (Cloud9 IDE)

Este tipo de plataforma de programación es cien por cien online y está basada también en el concepto de compilación cruzada, pero añadiendo a la fórmula que la plataforma ya está preparada y configurada para la programación en diferentes lenguajes y tan sólo tienes que indicar un IP local a la que referirse para poder compilar.

La plataforma de este tipo recomendada por la comunidad de la BBB es Cloud9 IDE, por lo que este proyecto se centrará en ella para el análisis de sus características [9].

-Dificultad de configuración y puesta en marcha: La configuración es muy sencilla, tan solo se tiene que seleccionar el lenguaje de programación deseado y dar una IP para que se produzca la conexión.

-Lenguajes de programación: La plataforma online está preparada para compilar los siguientes lenguajes: C, C++, PHP, Ruby, Perl, Python y JavaScript, entre otras.

-Curva de aprendizaje: Rápida

-Desarrollo GUI: Éste es el punto débil de las plataformas online, no están pensadas para el desarrollo de GUI y aunque no es imposible, no están optimizadas para ello.

Plataforma “SoftPLC” (CODESYS)

Para comprender del concepto de “SoftPLC” es necesario tener un poco de perspectiva sobre qué es un PLC o un PC.

Los PLC son dispositivos electrónicos surgidos en los años 70 para sustituir la lógica de relés en la industria. Tienen un microcontrolador integrado sobre el que normalmente corre un sistema operativo en tiempo real. Este micro garantiza que va a responder incluso en las situaciones más severas.

Por otro lado, el PC tiene un rol mucho más alejado de las máquinas, ya que normalmente lo relacionamos para trabajar más enfocados a las aplicaciones de oficina, domésticas o de entretenimiento [17].

Un SoftPLC es un software que se instala en una electrónica y emula las funcionalidades de un PLC. Esto, sumado a una interfaz hardware de entradas y salidas permite emular el funcionamiento de un PLC.

Sin embargo, los SoftPLC no sólo pueden correr sobre PC's. Estos programas en realidad pueden trabajar sobre cualquier placa electrónica con ciertos requisitos de memoria y velocidad. Por eso mismo, encontramos productos que pueden ser integrados en placas electrónicas como Raspberry Pi, Beaglebone o incluso pequeños SoC's [18].

Uno de los SoftPLC más utilizados en sistemas embebidos es CODESYS, y por ello lo tomaremos como referencia para la comparación con el resto de plataformas software.

-Dificultad de configuración y puesta en marcha: Tiene una dificultad media, dado que es necesario pre-acondicionar la plataforma embebida antes de empezar a programar de modo que se puede instalar el “runtime” que permite la ejecución del software y la programación de la misma.

-Lenguajes de programación: IL (Lista de instrucciones), ST (Texto estructurado), LD (Diagrama ladder), FBD (Diagrama de bloques de función), SFC (Bloques de función secuenciales), CFC (Continuous function chart).

-Curva de aprendizaje: Rápida

-Desarrollo GUI: Esta plataforma está preparada para diseñar GUI rápidas y funcionales gracias a sus herramientas con objetos prediseñados. Las GUIs desarrolladas con esta plataforma también son compatibles con la visualización online.

-Estandarización: a diferencia que el resto de opciones software mostradas en este proyecto, CODESYS cumple el estándar del IEEE IEC 61131 para PLC [19].

7 SOLUCIÓN EN TÉRMINOS DE SOFTWARE

Una vez se conocen las alternativas para las plataformas software con la que se puede programar la BBB, se elegirá la solución que pueda alcanzar todos los objetivos del proyecto.

El presente proyecto se decanta por la elección de una plataforma software basada en el concepto de SoftPCL, para ser más específico, la plataforma CodeSys del fabricante “3S-Smart Software Solutions GmbH” especializado en el desarrollo de software para automatización. A continuación, se expondrán las razones que han determinado la decisión.

Donde esta plataforma destaca notablemente sobre las demás, es en que está desarrollada y pensada según estándares industriales para PLCs (IEC 61131) lo que hace su integración con componentes industriales y su programación mucho más sencilla.

El concepto de PLC hace mucho más coherente el sistema a la hora de trabajar con componentes industriales, además de optimizar mucho la curva de aprendizaje ya que una vez instalado el “runtime” la interacción con la BBB es mucho más sencilla que con cualquier otra de las opciones.

Adicionalmente, CODESYS proporciona una gran flexibilidad a la hora de elegir plataforma hardware, de modo que en el futuro se puede escalar o cambiar el hardware de control de una manera muy rápida.

Cabe destacar la facilidad de desarrollo de GUIs. La plataforma dispone de herramientas avanzadas que permiten construir un panel de control en cuestión de minutos, ligando las variables del sistema a diferentes objetos prediseñados.

Estas GUIs diseñadas en CODESYS no sólo se pueden visualizar gracias a la salida HDMI de la BBB, sino que también, son compatibles para ser visualizadas a través de una web que está alojada en la BBB.



Ilustración 47 - Panel desarrollo GUI CodeSys.

Por último, los lenguajes de programación que son compatibles con CodeSys están claramente orientados a una aplicación industrial, lo que facilita el aprendizaje del manejo del sistema, ya que si el usuario domina el manejo de PLCs también estará mucho más familiarizado con estos lenguajes.

7.1 Programa de validación

Como se ha definido en el apartado de objetivos, el último objetivo de este proyecto es poner a disposición del usuario un programa básico que permita validar la mayoría de las interfaces implementadas en el hardware del sistema de control. En este apartado se mostrará este programa y se explicará el diseño del mismo.

El primer elemento que se va a explicar, es la configuración que se ha definido para el árbol del proyecto de CodeSys que se muestra a continuación y que se divide en cuatro partes principales que se explican a continuación:



Ilustración 48 - Árbol de proyecto CodeSys

1. En este bloque se encuentra la programación de cada una de las funciones o módulos POU (“Program Organization Unit”)[19], que contiene el proyecto. Se pueden observar las funciones de gestión del audio, control de salidas y entradas digitales (directas y por relés), comunicación DMX, control de servos y steppers.
2. En el segundo bloque, se definen las tareas que debe ejecutar el “runtime” de CodeSys en paralelo. En esta definición se le asigna a cada tarea sus características principales como son; tipo de ejecución (lineal o cíclica), periodo de ejecución y por último prioridad de ejecución.
3. Definición de la interfaz de usuario para el control del sistema animatrónico, tanto en su versión web como en la local.
4. Definición de la configuración del dispositivo hardware (BBB), especificando la función y características de cada entrada y salida.

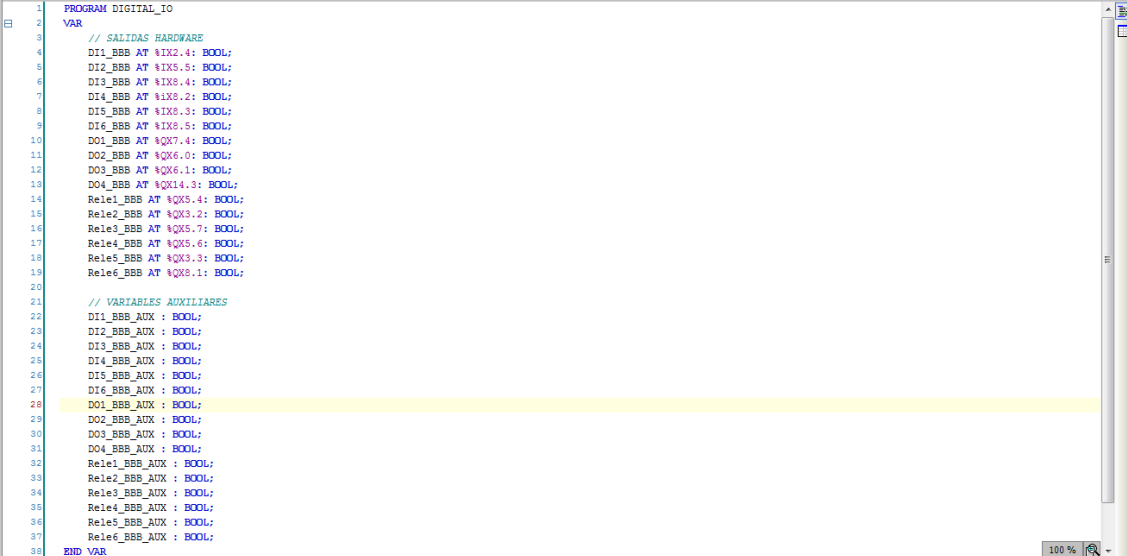
A continuación se muestra en detalle el contenido de cada uno de estos componentes del proyecto.

7.1.1 Funciones o módulos POU

En este módulo podemos encontrar dos tipos de funciones/POUs, las que se pueden ejecutar de manera nativa sobre el “runtime” de CodeSys y las que no. Una función nativa, es por ejemplo toda la gestión de entradas y salidas y una no nativa es la gestión de audio. A continuación se explica cada uno de los POU del sistema en detalle y su funcionamiento nativo o externo.

DIGITAL IO

Esta función es nativa y se programa de manera directa en CodeSys, de modo que el “runtime” se encarga de todo lo necesario para gestionar las salidas y leer las entradas. Esta función se ha programado de la siguiente manera.



```

1 PROGRAM DIGITAL_IO
2 VAR
3 // SALIDAS HARDWARE
4 D11_BBB AT %IX2.4: BOOL;
5 D12_BBB AT %IX5.5: BOOL;
6 D13_BBB AT %IX8.4: BOOL;
7 D14_BBB AT %IX8.2: BOOL;
8 D15_BBB AT %IX8.3: BOOL;
9 D16_BBB AT %IX8.5: BOOL;
10 DO1_BBB AT %QX7.4: BOOL;
11 DO2_BBB AT %QX6.0: BOOL;
12 DO3_BBB AT %QX6.1: BOOL;
13 DO4_BBB AT %QX14.3: BOOL;
14 Rele1_BBB AT %QX5.4: BOOL;
15 Rele2_BBB AT %QX3.2: BOOL;
16 Rele3_BBB AT %QX5.7: BOOL;
17 Rele4_BBB AT %QX5.6: BOOL;
18 Rele5_BBB AT %QX3.3: BOOL;
19 Rele6_BBB AT %QX8.1: BOOL;
20
21 // VARIABLES AUXILIARES
22 D11_BBB_AUX : BOOL;
23 D12_BBB_AUX : BOOL;
24 D13_BBB_AUX : BOOL;
25 D14_BBB_AUX : BOOL;
26 D15_BBB_AUX : BOOL;
27 D16_BBB_AUX : BOOL;
28 DO1_BBB_AUX : BOOL;
29 DO2_BBB_AUX : BOOL;
30 DO3_BBB_AUX : BOOL;
31 DO4_BBB_AUX : BOOL;
32 Rele1_BBB_AUX : BOOL;
33 Rele2_BBB_AUX : BOOL;
34 Rele3_BBB_AUX : BOOL;
35 Rele4_BBB_AUX : BOOL;
36 Rele5_BBB_AUX : BOOL;
37 Rele6_BBB_AUX : BOOL;
38 END_VAR

```

Ilustración 49 - Definición de variables de CodeSys.

Primeramente se asigna a la dirección de cada pin de la BBB (p.e. %IX2.4) una variable (p.e. D11_BBB). Adicionalmente se genera una variable auxiliar para cada una de estas entradas y salidas que serán las que se manejen por medio de la GUI.

```

1 // ENTRADAS DIGITALES
2 DI1_BBB_AUX := DI1_BBB;
3 DI2_BBB_AUX := DI2_BBB;
4 DI3_BBB_AUX := DI3_BBB;
5 DI4_BBB_AUX := DI4_BBB;
6 DI5_BBB_AUX := DI5_BBB;
7 DI6_BBB_AUX := DI6_BBB;
8 // SALIDAS DIGITALES
9 DO1_BBB_AUX := DO1_BBB;
10 DO2_BBB_AUX := DO2_BBB;
11 DO3_BBB_AUX := DO3_BBB;
12 DO4_BBB_AUX := DO4_BBB;
13 // RELES
14 Rele1_BBB_AUX := Rele1_BBB;
15 Rele2_BBB_AUX := Rele2_BBB;
16 Rele3_BBB_AUX := Rele3_BBB;
17 Rele4_BBB_AUX := Rele4_BBB;
18 Rele5_BBB_AUX := Rele5_BBB;
19 Rele6_BBB_AUX := Rele6_BBB;

```

Ilustración 50 - Función de gestión de entradas y salidas digitales.

Una vez están definidas todas las variables, la función de gestión de entradas y salidas digitales asignará, en el caso de las salidas, el valor que defina el usuario mediante la GUI como valor de salida de la BBB. Por ejemplo, cuando el usuario pulse el botón asignado a la variable DO1_BBB_AUX, esta función asignará un valor positivo a la variable de salida DO1_BBB, dando así la BBB un nivel alto en el pin asignado a esta variable.

En el caso de las entradas digitales esto funciona al contrario, siendo el valor de la variable asignada a la cabecera de la BBB la que define el valor del indicador de la GUI.

AUDIO

Este POU no es nativo, es decir, esta función no puede ser gestionada por el “runtime” de CodeSys. En estos casos CodeSys proporciona una alternativa que es la ejecución de comandos de sistema, de manera que se pueda gestionar todas las funciones disponibles en el ordenador embebido sin problema.

En este caso, CodeSys no proporciona herramientas que gestionen este tipo de archivos multimedia, por lo que se ha implementado por medio de comandos como se muestra a continuación.

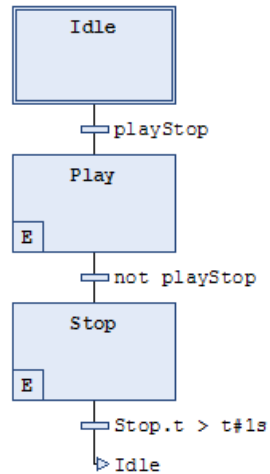


Ilustración 51 - GRAFCET función de audio.

En la función de audio se ha implementado un GRAFCET que es gestionado mediante la GUI. En el GRAFCET se pueden dar tres estados:

1. Estado de espera: la función está estática hasta que se le mande un comando por medio de la GUI.
2. Play: este estado manda el comando del sistema que inicia la reproducción del audio. El comando utilizado es “aplay”.
3. Stop: este estado manda el comando del sistema que para la reproducción del audio y tras un segundo vuelve al estado de espera. El comando utilizado es “pkill aplay”.

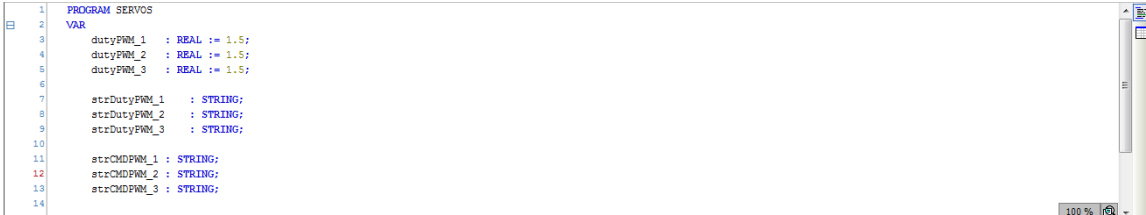
De esta manera el usuario será capaz de gestionar la reproducción de audio mediante CodeSys de una manera sencilla.

SERVOS

En este caso, a pesar de que se podría haber gestionado de manera nativa la generación de las señales PWM que manejan los servomotores, se ha decidido implementar la función de manera externa. El presente proyecto se ha decantado por esta alternativa por las siguientes razones:

- La generación de una PWM nativa se debería de haber gestionado por “timers” que consumen muchos recursos de procesado y necesitan un código más complejo.
- La BBB dispone de módulos hardware especializados en la gestión de este tipo de señales sin necesidad de consumir recursos (PRUs). Pero sólo son accesibles por medio de comandos de sistema.
- La calidad de la señal PWM generada por los módulos PRU es mucho mayor que mediante la generación nativa con “timers”.

Una vez justificada la decisión para la implementación de la función como externa, se va a detallar su diseño.



```
1 PROGRAM SERVOS
2 VAR
3     dutyPWM_1 : REAL := 1.5;
4     dutyPWM_2 : REAL := 1.5;
5     dutyPWM_3 : REAL := 1.5;
6
7     strDutyPWM_1 : STRING;
8     strDutyPWM_2 : STRING;
9     strDutyPWM_3 : STRING;
10
11     strCMDPWM_1 : STRING;
12     strCMDPWM_2 : STRING;
13     strCMDPWM_3 : STRING;
14
```

Ilustración 52 - Declaración variables POU SERVOS.

Lo primero que se ha hecho es la declaración de tres tipos de variables. Las variables del tipo “dutyPWM_*” que se inicializan a un valor de 1.5 pero que estarán asignadas a los “sliders” de la GUI.

La segunda variable a declarar del tipo “strDutyPWM_*” se encargará de almacenar el valor del ciclo de trabajo de la PWM en formato “String” de manera procesada y escalada para que el comando del sistema pueda ejecutarse de manera correcta.

En tercer lugar y último está la variable “strCMDPWM_*” que almacenará la cadena de caracteres que representa el comando del sistema para la selección del ciclo de trabajo de la señal PWM en el módulo PRU.

```

1 // SERVO PWM 1
2 strDutyPWM_1 := CONCAT(INT_TO_STRING(REAL_TO_INT(dutyPWM_1*10)), '00000 €');
3 strCMDPWM_1 := CONCAT('cmdPWM_1 ', strDutyPWM_1);
4 SysProcessExecuteCommand(strCMDPWM_1, ADR(cmdResult));
5

```

Ilustración 53 - Programación POU SERVOS.

Tras esto, la función simplemente tendrá que gestionar estas variables de la siguiente manera:

- Escalar el valor proporcionado por la GUI en la variable dutyPWM_1 y convertirla en una cadena de caracteres almacenada en la variable strDutyPWM_1.
- Concatenar la variable strDutyPWM1 al comando del sistema “cmdPWM_1” y almacenarlo en la variable strCMDPWM_1.
- Ejecutar la función para comandos del sistema “SysProcessExecute()”, con la variable strCMDPWM_1, seleccionando así de manera externa al “runtime” el ciclo de trabajo de la señal PWM 1.

STEPPER

El caso de este POU es especial dado que se trata de un módulo mixto en el que se han utilizado las funciones nativas de CodeSys para la gestión de la señal de salida que define el sentido de giro del motor y funciones externas para la gestión de la señal PWM que define su velocidad.

```

1 PROGRAM STEPPER
2 VAR
3     dutySTEPPER_1 : REAL := 1.5;
4     strDutySTEPPER_1 : STRING;
5     strCMDSTEPPER_1 : STRING;
6     dirSTEPPER_1 AT %QX2.6 : BOOL;
7     dirSTEPPER_AUX_1 : BOOL;
8     cmdResult : RTS_IEC_RESULT;
9 END_VAR
10

```

```

1 // STEPPER 1
2 dirSTEPPER_AUX_1 := dirSTEPPER_1;
3 strDutySTEPPER_1 := CONCAT(INT_TO_STRING(REAL_TO_INT(dutySTEPPER_1*10)), '00000 €');
4 strCMDSTEPPER_1 := CONCAT('cmdSTEPPER_1 ', strDutySTEPPER_1);
5 SysProcessExecuteCommand(strCMDSTEPPER_1, ADR(cmdResult));

```

Ilustración 54 - POU STEPPER.

Como se puede observar en la programación, el contenido de este módulo es la combinación de lo explicado en los módulos de STEPPER (señal PWM) y DIGITAL_IO (salida digital).

7.1.2 Ejecución de tareas

Como se ha explicado al comienzo del apartado, en CodeSys se debe de definir las tareas y modo de ejecución del programa. A continuación se muestra en forma gráfica y esquemática como se ha definido la ejecución de las tareas en el presente proyecto.

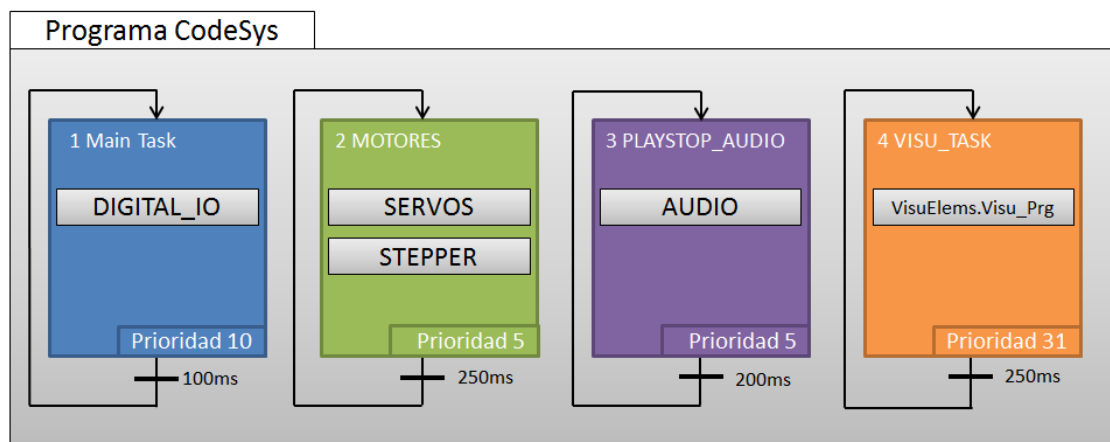


Ilustración 55 - Esquema ejecución programa CodeSys.

El programa está compuesto por 4 tareas diferentes con las siguientes características:

- Main Task
 - POU: DIGITAL_IO
 - Ejecución cíclica cada 100ms
 - Prioridad 10 de 31
- MOTORES
 - POU: SERVOS, STEPPER
 - Ejecución cíclica cada 250ms
 - Prioridad 5 de 31

- PLAYSTOP_AUDIO
 - POU: AUDIO
 - Ejecución cíclica cada 200ms
 - Prioridad 5 de 31

- VISU_TASK
 - POU: VisuElems.Visu_Prg
 - Ejecución cíclica cada 250ms
 - Prioridad 31 de 31

7.1.3 Interfaz gráfica o GUI

El bloque de la interfaz gráfica, se ha implementado como se muestra a continuación:

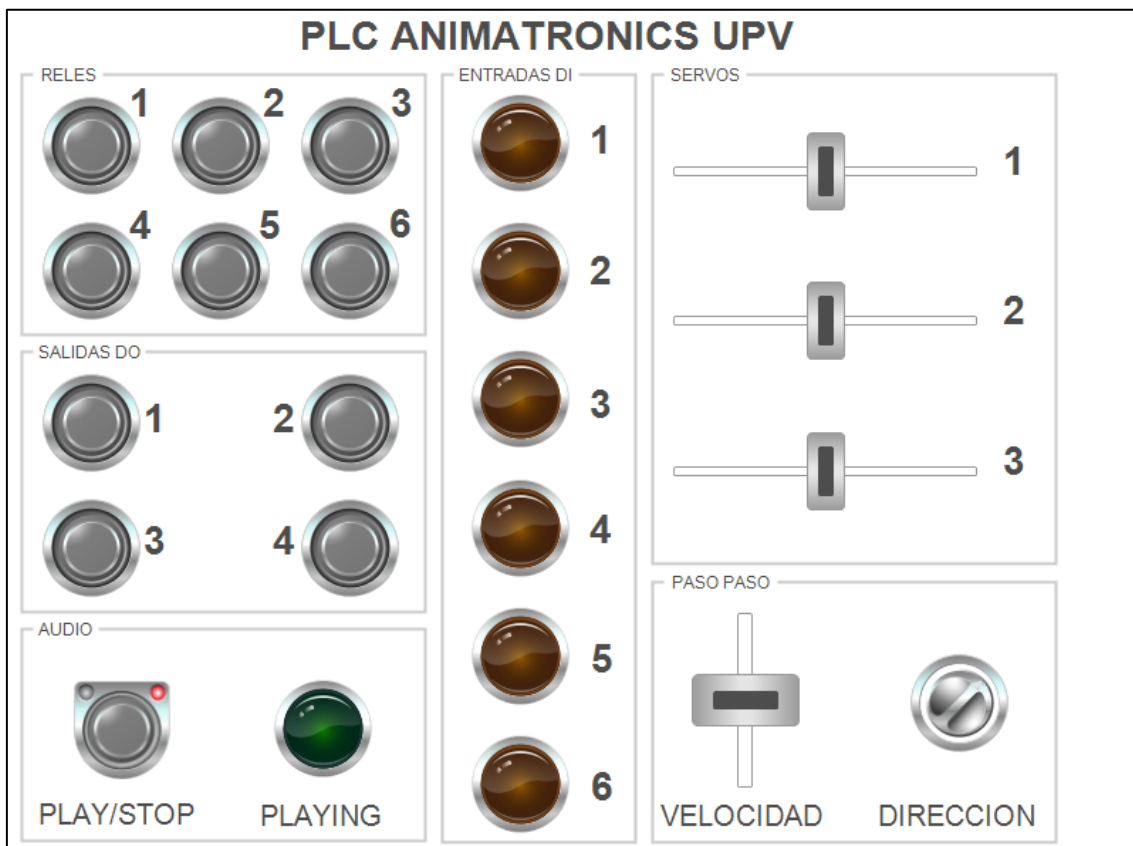


Ilustración 56 - GUI CodeSys de control del sistema.

En este panel se han implementado objetos gráficos pre-determinados de CodeSys a los que se les ha asignado una variable auxiliar que permite la interacción con las funciones programadas del bloque anterior.

La GUI permite el control de las salidas y entradas principales del sistema de la siguiente manera:

- Activación y desactivación de relés por medio de pulsadores.
- Activación y desactivación de las salidas digitales por medio de pulsadores.
- Gestión de la reproducción del audio mediante un botón de reproducción y parada y un indicador del estado de la misma.
- Información del estado de las entradas digitales mediante un indicador LED.
- Gestión del giro de servomotores mediante un “slider”.
- Gestión de la velocidad de giro y sentido de un motor paso a paso mediante un “slider” y un selector respectivamente.

La suma de todo esto permite que el diseñador y usuario puedan validar todas las entradas y salidas del sistema, verificando así su correcta implementación de una manera rápida e intuitiva.

7.1.4 Configuración dispositivo

En último lugar está el módulo de configuración del dispositivo. Esta parte es fundamental ya que es donde se define la configuración de la interfaz entre el “runtime” de CodeSys y el hardware de la BBB, de modo que el “driver” de CodeSys pueda gestionar de manera correcta todas las entradas y salidas. A continuación se muestra un esquema conceptual.

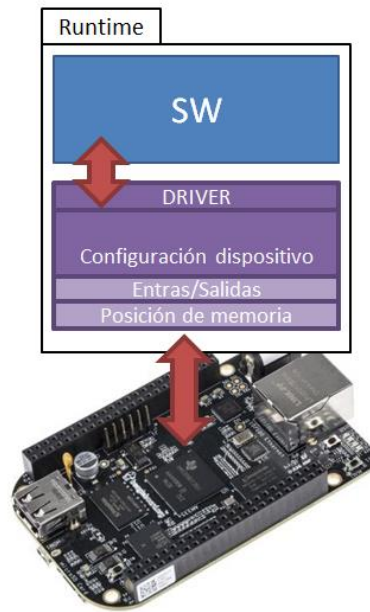


Ilustración 57 - Esquema runtime y driver CodeSys.

En la configuración se deben definir dos parámetros principales. En primer lugar se tiene que especificar para cada pin de la cabecera de la BBB si será utilizado como entrada o salida, de modo que el driver pueda configurar el hardware.

En segundo lugar está la asignación de un espacio de memoria a cada una de esas entradas y salidas, de modo que en la programación de los POU se pueda acceder a ellas pudiendo leer o modificar su valor.

La definición de estos parámetros le permite a CodeSys generar un “driver” que se encargue de la conexión entre el software y el hardware de la BBB.

8 RESULTADOS Y TEST

Para la validación del presente proyecto se analizará el funcionamiento de cada una de las partes principales del sistema por separado y de manera independiente, para más tarde, unirlos y verificar la funcionalidad del sistema en su conjunto. Por ello este apartado se dividirá en tres sub-apartados:

- Validación interfaz hardware.
- Validación función y configuración ordenador embebido.
- Validación sistema completo.

Esto permitirá detectar problemas de una manera más óptima al ser capaces de determinar la procedencia de una manera más sencilla.

A continuación, se pasa a desarrollar cada uno de estos sub-apartados en detalle.

8.1 Validación interfaz hardware

En este caso también se va a seguir una estrategia sectorial de montaje de la PCB, de modo que se irán soldando los componentes ligados a una función e inmediatamente después, y antes de pasar a los componentes de la siguiente función, se validará el correcto funcionamiento.

Para la validación se usarán fuentes de alimentación externas para simular entradas de diferente voltaje y polímetro para medir las salidas.

El resultado de los diferentes test se resumen a continuación en una tabla:

Interfaz hardware	
Descripción Test	Resultado
Alimentación 24V	Superado
Alimentación 5V	Superado
Alimentación 3,3V general	Superado
Alimentación 3,3V bajo ruido	Superado
Indicación encendido	Superado
Entrada digital 5V	Superado
Entrada digital 24V	Superado
Salida digital 5V	Superado

Salida digital voltaje externo	Superado
Salidas PWM	Superado
Salida Relés	Superado
RTC	Superado
DMX	-
Audio	-

Tabla 4 - Resumen validación interfaz hardware.

Dada la complejidad de las señales digitales de audio y DMX, no se han podido simular estas interfaces de manera independiente, por lo que no se han obtenido resultados concluyentes para estas interfaces en este test.

8.2 Validación ordenador embebido

De igual manera que en el apartado anterior, se van a aislar las funciones principales del ordenador embebido y se van a testear de manera independiente.

En la tabla siguiente se muestran los test realizados y los resultados obtenidos:

Ordenador embebido	
Descripción Test	Resultado
Configuración Runtime CODESYS	Superado
Conexión remota BBB	Superado
GUI HMDI	Superado
GUI on-line	Superado
Instalación imagen en eMMc	Superado
Entradas digitales	Superado
Salidas digitales	Superado
Salidas Relés	Superado
Salida PWM	Superado
Salida Audio I2S	Superado
Interfaz DMX	Superado
RTC	-

Tabla 5 - Resumen validación ordenador embebido.

Cabe destacar como se ha verificado la funcionalidad de la interfaz I2S, donde se ha podido comprobar, gracias a un analizador digital, las tramas del protocolo I2S que envía la BBB al reproducir una canción. En la siguiente ilustración se muestran los valores medidos:

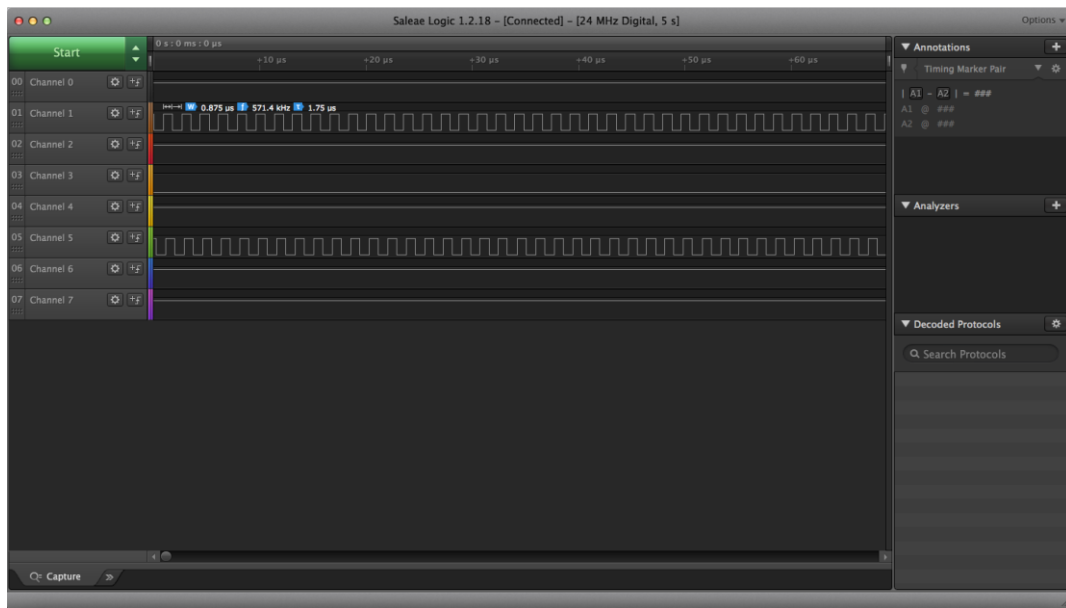


Ilustración 58 - Salida I2S BBB en silencio.

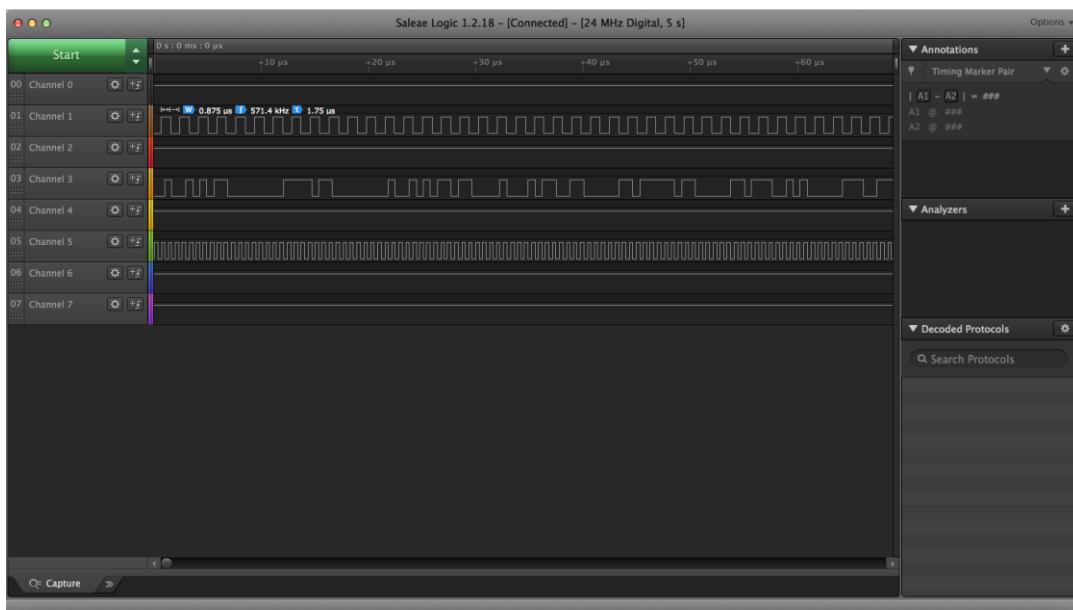


Ilustración 59 - Salida I2S BBB en reproducción.

En las imágenes se identifica claramente la diferencia en el envío de datos (ilustración 46 canal 3) con la canción en parado (ilustración 47 canal 3) y en reproducción.

8.3 Validación sistema completo

Por último la validación del sistema completo, donde se unen el ordenador embebido con la interfaz hardware.

Como en los apartados anteriores, mostramos un resumen de los resultados en forma de tabla:

Sistema completo	
Descripción Test	Resultado
Alimentación sistema	Superado
Entradas digitales 5V	Superado
Entradas digitales 24V	Superado
Salidas digitales 5V	Superado
Salidas digitales voltaje externo	Superado
Salidas relés	Superado
Salidas PWM	Superado
RTC	Superado
DMX	Superado
Audio	Superado

Tabla 6 - Resumen validación del sistema.

9 CONCLUSIONES

Para concluir se repasan todos los objetivos del presente proyecto y las soluciones adoptadas para alcanzarlos.

En primer lugar se ha diseñado todo el hardware necesario para el control del sistema animatrónico basado en dos partes principales, que son la BBB como ordenador embebido del sistema y responsable de la gestión y control y por otro lado la interfaz hardware con todos los componentes necesarios para que el ordenador embebido pueda interactuar con su entorno tal y como se especifica en el proyecto.

La BBB permite una implementación rápida y flexible con muchas plataformas diferentes, además de disponer de un gran número de entradas y salidas disponibles para su configuración.

En el caso de la interfaz hardware se implementado un PCB ocupando todas las entradas y salidas disponibles de la BBB y asignándoles una funcionalidad específica que sea coherente con el manejo de sistemas animatrónicos, como es el control de motores, pistones, salidas y entradas digitales, control de audio, etc. Cabe destacar de esta parte que se han implementado también mecanismos de seguridad como octoacopladores y relés que hacen al sistema más seguro y robusto.

En segundo lugar está el software, donde el presente proyecto se ha decantado por el sistema CODESYS basado en el concepto de "SoftPCL". Este sistema, que está desarrollado según estándares industriales, es óptimo para el control de elementos electromecánicos industriales y para su rápida implementación en la plataforma BBB.

Esta estandarización también aporta ventajas en el terreno de la flexibilidad y la curva de aprendizaje para su programación. Lo que ha permitido el desarrollo de una herramienta de validación rápida y eficaz para verificar todas las entradas/salidas de la plataforma.

Por último y tras los resultados de los test obtenidos, se puede concluir que los objetivos del presente proyecto se han alcanzado, dejando así a la disposición del

usuario una herramienta software y hardware muy flexible y económica para la implementación de sistemas animatrónicos.

Índice de ilustraciones

Ilustración 1 - Esquema sistema animatrónico	9
Ilustración 2 - Reloj astronómico de Praga (www.disfrutapraga.com).....	12
Ilustración 3 - Sistemas animatrónicos Walt Disney; Abraham Lincoln, pájaro Tiki, “9 tall dancing man” (www.thewaltdisneycompany.com).....	13
Ilustración 4 - Equipo Stan Winstons School con sistema animatrónico de la película Jurassic Park (www.stanwinstonschool.com).	13
Ilustración 5 - Sistema animatrónico humanoide FRED de Engineered Arts (www.engineeredarts.co.uk).....	14
Ilustración 6 - Esquema sistema animatrónico enfocado al TFG.	15
Ilustración 7 - Esqueleto sistema animatrónico Jurassic Park (www.stanwinstonschool.com).....	16
Ilustración 8 - Interpretación de una extremidad desde modelo biológico a implementación mecatrónica (Daniel Kuehn, ACTIVE SPINE AND FEET WITH INCREASED SENSING CAPABILITIES FOR WALKING ROBOTS).....	17
Ilustración 9 - Esqueleto y acabado estético de FRED (www.engineeredarts.co.uk). ...	18
Ilustración 10 - Esquema sistema de control TFG.	19
Ilustración 11 - Actuadores presentes en un modelo de extremidad (www.hindawi.com).	24
Ilustración 12 - Tipos variados de servos (www.superrobotica.com).....	24
Ilustración 13 - Esquema componentes de un servo (www.comofunciona.co.com). ...	25
Ilustración 14 - Relación ancho de pulso vs ángulo de giro de un servo 180° (www.jameco.com).	25
Ilustración 15 - rotor de jaula de ardilla. (Google Imágenes - CC BY ND)	27

Ilustración 16 - Esquema sistema neumático (www.quora.com).	28
Ilustración 17 - Funcionamiento sensor capacitivo (www.termoregulatory.pl).	29
Ilustración 18 - Funcionamiento sensor inductivo (www.temporizadores-sensores.blogspot.com).....	30
Ilustración 19 - Conjunto variado de finales de carrera industriales (www.library.automationdirect.com).	31
Ilustración 20 - Utilización de finales de carrear como protección de una extremidad mecánica.....	31
Ilustración 21 - Esquema funcionamiento de un Encoder (www.logicbus.com.mx).	32
Ilustración 22 - Conjunto variando de encoders para uso industrial (www.anaheimautomation.com).	32
Ilustración 23 - Arquitecturas DMX512 (PHILIPS Introduction to DMX 2008)..	33
Ilustración 24 - "String" DMX512(PHILIPS Introduction to DMX 2008).	34
Ilustración 25 - Aplicación DMX512 para control RGB iluminación (PHILIPS Introduction to DMX 2008).....	34
Ilustración 26 - Beaglebone Black (www.beagleboard.org).....	35
Ilustración 27 - Funcionalidad cabeceras BBB (es.mathworks.com).....	37
Ilustración 28 - Collage rail DIN.	38
Ilustración 29 - Dimensiones interfaz hardware.	38
Ilustración 30 - Encapsulamiento sistema de control para montaje en carril DIN (es.rs-online.com).....	39
Ilustración 31 - Subsistemas de la interfaz hardware.	39
Ilustración 32 - Transformador MRD-20-24 de Mean Well (Datasheet).....	41
Ilustración 33 - Conversor TSR 2-2450 Traco Power (Datasheet).	42

Ilustración 34 - Esquemático alimentación interfaz hardware.	42
Ilustración 35 - Esquemático acondicionamiento entradas digitales.	44
Ilustración 36 - Esquemático acondicionamiento salidas digitales.	46
Ilustración 37 - Esquemático asignación de salidas digitales.	47
Ilustración 38 - Esquemático acondicionamiento relés.	48
Ilustración 39 - Esquemático acondicionamiento salidas PWM.	51
Ilustración 40 - Esquemático acondicionamiento salidas PWM con sentido de giro. ...	51
Ilustración 41 - Comunicación I2S (http://iot-bits.com/interfacing-an-audio-codec-with-esp32).	52
Ilustración 42 - Esquemático DAC.	53
Ilustración 43 - Esquemático amplificador de audio.	53
Ilustración 44 - Esquemático transformadores 3,3V de bajo ruido.	54
Ilustración 45 - Esquemático salida DMX.	55
Ilustración 46 - Distribución entradas y salidas interfaz hardware.	58
Ilustración 47 - Panel desarrollo GUI CodeSys.	66
Ilustración 48 - Árbol de proyecto CodeSys	67
Ilustración 49 - Definición de variables de CodeSys.	68
Ilustración 50 - Función de gestión de entradas y salidas digitales.	69
Ilustración 51 - GRAFCET función de audio.	70
Ilustración 52 - Declaración variables POU SERVOS.	71
Ilustración 53 - Programación POU SERVOS.	72
Ilustración 54 - POU STEPPER.	72

Ilustración 55 - Esquema ejecución programa CodeSys.....	73
Ilustración 56 - GUI CodeSys de control del sistema.....	74
Ilustración 57 - Esquema runtime y driver CodeSys.....	76
Ilustración 58 - Salida I2S BBB en silencio.	79
Ilustración 59 - Salida I2S BBB en reproducción.....	79

Índice de tablas

Tabla 1 – Comparación técnica Beaglebone vs Raspberry Pi.....	22
Tabla 2 - Salidas soportadas por CODESYS.....	56
Tabla 3 - Asignación de funciones a la cabecera de la BBB.....	57
Tabla 4 - Resumen validación interfaz hardware.....	78
Tabla 5 - Resumen validación ordenador embebido.....	78
Tabla 6 - Resumen validación del sistema.....	80

Bibliografía

- [1] Rodney H, Gene P, (2003). *Animatronics: A Designer's Resource Guide*, Ian Hunter Publishers.
- [2] Anónimo (2001). Prague Astronomical Clock. www.pragueastronomicalclock.info (URL visitada 2018).
- [3] Disney (2017). The Birds, Beasts, and Beauty of Disney's Audio-Animatronics Characters. www.d23.com/audio-animatronics-disneyland-magic-kingdom-walt-disney-world (URL visitada 2018).
- [4] Disney (2019). Disney Research. www.la.disneyresearch.com (URL visitada 2018)
- [5] Kapse, .Y.D. y Sarangpure, P. R. (2016). Animatronic. *International Journal of Advanced Research in Computer and Communication Engineering*, 5,266-269.
- [6] Cbenso (2018). How to Make a Robot - Lesson 3: Making Sense of Actuators. www.robotshop.com/blog/en/how-to-make-a-robot-lesson-3-actuators-2-3703 (URL visitada 2019).
- [7] Jameco (2018). How Do Servo Motors Work?. www.jameco.com/jameco/ (URL visitada 2019)
- workshop/howitworks/how-servo-motors-work.html (URL).
- [8] Saeed N.(2011). *Introduction to Robotics: Analysis, Control, Applications*. John Wiley & Sons, Inc.
- [9] ENCODER (2016). ¿Qué es un Encoder? www.encoder.com/blog/encoder-basics/que-es-un-encoder (URL visitada 2019).
- [10] Phillips (2008). *Introduction to DMX*. BU Controls.
- [11] ESTA(2016). ANSI E1.27-1-2006. Entertainment Technology – Standard for Portable Control Cables for Use with ASI E1.11 (DMX512-A) and USITT DMX512/1990 Products

[12] Comisión Electrónica Internacional (2017). IEC/EN 60715. *Standardized mounting on rails for mechanical support of switchgear, controlgear and accessories*.

[13] DIN 46277-3. *Estándar perfil DIN*.

[14] Rufino Valor, J.R. (2010). *Entradas y salidas digitales de propósito general en los microcontroladores dsPIC30*. UPV

[15] Comunidad elinux (2018). Toolchains. <https://elinux.org/Toolchains> (URL visitada 2019).

[16] Comunidad GNU (2010). Cross-Compilation
[.www.gnu.org/software/automake/manual/html_node/Cross_002dCompilation.html](http://www.gnu.org/software/automake/manual/html_node/Cross_002dCompilation.html)
(URL visitada 2019).

[17] Cloud9 (2015). Getting Started with Cloud9. www.docs.c9.io/docs (URL).

[18] Satoshi (2017). ¿Qué es un SoftPLC?. www.opiron.com/2017/11/27/que-es-un-softplc-ventajas (URL visitada 2019).

[19] Organización de estándares IEEE (2016). IEC 1131. *Estándar autómatas programables (PLC)*.

[20] Emiliano P. López - Santa Fe. GNU/Linux embebido.
<http://linuxemb.wikidot.com/tesis-c3> (URL visitada 2019).

[21] J. Garrigos, (2011). *Motores de corriente Alterna*.