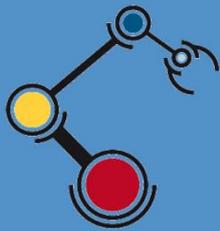




Escuela  
Politécnica  
Superior

# Control de Brazo Robótico mediante Ros en Plataformas de bajo coste



Máster Universitario en Automática  
y Robótica

## Trabajo Fin de Máster

Autor:

Adrián Peña Pablos

Tutor/es:

Vicente Morell Giménez



Universitat d'Alacant  
Universidad de Alicante

## Agradecimientos

En primer lugar, me gustaría dar las gracias a la empresa *IRYS Control* por darme la oportunidad de realizar este proyecto y por toda la ayuda que me han ofrecido, con la cual no habría sido posible la realización de este proyecto.

Agradezco también a mi tutor Vicente Morell Giménez por la atención y ayuda recibida por su parte.

Por último, agradezco a mi madre, a mi hermana, a mi padre, a mi pareja, familia y amigos por el apoyo recibido por todos ellos día tras día durante todo este tiempo. Muchas gracias por estar siempre a mi lado.

## Resumen

Este trabajo de integración de tecnologías se centra en la realización de una tarea cotidiana como puede ser preparar y servir café usando plataformas de bajo coste. Este trabajo contempla la elección y/o diseño del brazo robótico para llevar a cabo esa tarea y su programación/control desde una plataforma de bajo coste que permita monitorizar los elementos necesarios. Además se propone integrar el sistema ya funcionando en una interfaz gráfica desde la cual mandar las órdenes de preparar los cafés y recibir avisos.

Para ello, será necesario buscar, estudiar, elegir e implementar los elementos necesarios para que el sistema funcione y pueda realizar su función. Se necesitara elegir una la plataforma entre todas las posibles opciones del mercado. Un brazo robótico que cumpla los requisitos para realizar su tarea. Y los elementos complementarios para ayudar la gestión, control y monitorización del proyecto.

Se trabajará con el software ROS, se implementaran comunicaciones entre distintos controladores, se diseñara un esquema eléctrico para la instalación y se realizaran diferentes pruebas.

# Índice

## Contenido

1.	Introducción.....	11
1.1.	Motivación.....	12
1.2.	Objetivos del trabajo.....	13
1.3.	Estructura de la memoria.....	14
2.	Cuerpo del trabajo.....	15
2.1.	Introducción.....	15
2.2.	Marco teórico y estado del arte.....	16
2.2.1.	Robótica.....	16
2.2.3.	Domótica.....	26
2.2.4.	Plataformas de bajo coste.....	29
2.2.5.	CODESYS.....	35
2.2.6.	Modbus.....	36
2.2.7.	Descripción del sistema.....	39
2.2.8.	Ejemplos de brazos robóticos en la domótica.....	47
2.3.	Objetivos.....	49
2.4.	Metodología.....	50
2.4.1.	Configuración BeagleBone Black.....	51
2.4.2.	Protocolo SSH.....	52
2.4.3.	Instalación ROS.....	52
2.4.4.	Open VPN.....	52
2.4.5.	Botonera.....	54
2.4.6.	CODESYS para BeagleBone Black.....	57
2.5.	Cuerpo del trabajo.....	58

2.5.1. Nodos del sistema.....	58
2.5.2. Modos de funcionamiento .....	62
2.5.3. Interfaz gráfica.....	65
2.5.4. Resultados.....	68
3. Conclusiones.....	71
3.1. Posibles mejoras.....	71
4. Apéndices .....	73
4.1. Comandos ROS desde terminal.....	73
REFERENCIAS .....	75
Anexos.....	77
Esquemas Eléctricos .....	77

## LISTA DE ILUSTRACIONES

Ilustración 1: Articulación lineal. ....	19
Ilustración 2: Articulaciones rotacionales. ....	19
Ilustración 3: Los 6 grados de libertad. ....	21
Ilustración 4: Esquema de comunicación entre nodos.....	24
Ilustración 5: Características de la domótica. ....	26
Ilustración 6: BeagleBone Black ..... 29	29
Ilustración 7: Pin Out BBB..... 30	30
Ilustración 8: Raspberry Pi. .... 32	32
Ilustración 9: Raspberry Pi 2B pinout ..... 32	32
Ilustración 10: Adafruit Feather HUZZAH ESP8266 ..... 33	33
Ilustración 11: Adafruit Feather HUZZAH ESP8266 pinout..... 34	34
Ilustración 12: ROBOT SNAM6800..... 39	39
Ilustración 13: Robot por piezas..... 40	40
Ilustración 14: Posicionamiento servo mediante PWM. .... 42	42
Ilustración 15: Esquema de potencia ..... 43	43
Ilustración 16: Bomba de vacío. .... 44	44
Ilustración 17: Sensor de nivel ..... 45	45
Ilustración 18: Detector de obstáculos IR ..... 46	46
Ilustración 19: Café X..... 48	48
Ilustración 20: Arquitectura Centralizada..... 50	50
Ilustración 21: Arquitectura desarrollada en el proyecto..... 51	51
Ilustración 22: Conexión remota del usuario con la BeagleBone. .... 53	53
Ilustración 23: Botonera ..... 54	54
Ilustración 24: Esquema eléctrico de la botonera..... 55	55

Ilustración 25: Botonera por detras .....	56
Ilustración 26: Nodos del sistema.....	58
Ilustración 27: Msg Botonera. ....	59
Ilustración 28: Msg Articulaciones .....	60
Ilustración 29: Diagrama de flujo con los pasos a seguir por el robot .....	64
Ilustración 30: Pantalla principal de la página web. A) Hora, B) Tiempo, C) Temperatura interior, D) Potencia; E) Iluminación, F) Climatización, G) Inicio sesión, H) Menú, I) Logo Empresa .....	66
Ilustración 31: Pantalla Robot .....	67
Ilustración 32: Sistema Robótico.....	68
Ilustración 33: Robot con cápsula .....	69

## LISTA DE TABLAS

Tabla 1: Información sobre la BeagleBone Black extraída de beagleboard.org .....	31
Tabla 2: Tabla Modbus.....	62
Tabla 3: Rango de giro de las articulaciones.....	63

<b>ACRÓNIMOS</b>	<b>DESCRIPCIÓN EN CASTELLANO</b>	<b>DESCRIPCIÓN EN INGLES</b>
<b>E/S</b>	Entradas y salidas	Input and Output
<b>BBB</b>	BeagleBone Black	BeagleBone Black
<b>HTTP</b>	Protocolo de Transferencia de hipertexto	Hypertext Transfer Protocol
<b>VPN</b>	Red Privada Virtual	Virtual Private Network
<b>OS</b>	Sistema operativo	
<b>SSH</b>		Secure Shell
<b>ROS</b>	Sistema Operativo Robótico	Robot Operating System

# Capítulo 1: Introducción

---

## 1. Introducción

Hoy en día, las viviendas están repletas de sistemas informáticos, eléctricos y mecánicos que nos hacen la vida más fácil. En casi todos los hogares disponemos de sistemas de calefacción, iluminación, seguridad, teléfono, internet, televisión que, hasta la aparición de la automatización del hogar, funcionaban de manera independiente.

La automatización del hogar (también conocido como domótica) se refiere al control automático y eléctrico de las funciones y actividades del hogar. Se emplean varios sistemas de control, los cuales pueden incluir el control centralizado de cerraduras de seguridad en puertas, electrodomésticos, ventanas, iluminación, cámaras de vigilancia y sistemas HVAC (Calefacción, ventilación y aire acondicionado).

La automatización del hogar ha aumentado muchísimo en popularidad en los últimos años. Una de las mayores ventajas es el ahorro energético con la programación inteligente del sistema, consiguiendo un sistema más eficiente. La programación de la luminosidad o de la climatización son unas de las medidas para conseguir el ahorro energético deseado. Otra de las ventajas más importante de un hogar automatizado es la facilidad con la que la funcionalidad se puede administrar en una variedad de equipos: Ordenadores, móviles, tabletas, etc.

Por tanto, se puede definir la domótica como conjunto de sistemas y tecnologías capaces de automatizar y controlar una vivienda, oficina o similar de manera inteligente para aportar seguridad, confort y bienestar, comunicación y ahorro energético.

## 1.1. Motivación

Como se ha explicado antes, la Automatización del hogar o domótica está en auge. Con lo cual surge este proyecto de implementar un sistema robótico capaz de realizar una tarea tan sencilla y diaria como poner un café. Esta tarea se va a implementar mediante el control de un brazo robótico usando un dispositivo de bajo coste y el software de ROS.

La oficina de IRYS Control, empresa dedicada al sector de la automatización industrial, en la cual actualmente trabajo, me ofreció la oportunidad de realizar este proyecto. Con la motivación de poder implementar un sistema robótico para realizar una tarea cotidiana.

Esta motivación es gracias al aumento de la domótica y robots de servicios. Ya que actualmente la mayoría de fabricantes de robots, están invirtiendo y adaptando sus brazos y robots a mundo del servicio. Consiguiendo grandes avances en este ámbito.

A raíz de este proyecto, se espera la familiarización con el software de ROS y su implementación en un dispositivo real. Se va a realizar el proceso de investigación, elección, instalación, programación y comprobación de robot.

## 1.2. Objetivos del trabajo

Como se ha definido antes, el objetivo principal del proyecto es que el robot, sea capaz de realizar la tarea cotidiana de servir cafés en la oficina de IRYS Control. Para conseguir lograr el objetivo se van a tener que lograr unas metas. Estas metas serán las siguientes:

- Definir las tareas que realizará el robot para servir los cafés. Que pasos debe seguir el robot para realizar su función.
- Elegir/diseñar un brazo robótico capaz de realizar esas tareas. Se debe elegir un brazo acorde con los requisitos necesarios para la tarea.
- Controlar los movimientos de ese brazo robótico usando la plataforma ROS integrada en un dispositivo de bajo coste. Además se deberá poder monitorizar los aspectos relevantes del sistema (si queda agua, si quedan capsulas, etc.)
- Se deberá utilizar un dispositivo de bajo coste (Arduino/RaspberryPi/BeagleBone). para controlar el brazo robótico y acceder a los sensores. Para esta elección habrá que estudiar las placas disponibles y elegir la mejor opción para el sistema.
- Se diseñara una interfaz gráfica para el control y monitorización del sistema. Esta interfaz estará integrada en el sistema demótico de la oficina.

### 1.3. Estructura de la memoria

Este proyecto se compone de 4 capítulos bien diferenciados aportando claridad al documento.

- En el capítulo 1 (actual) se describe una introducción al proyecto, la motivación a realizarlo y los principales objetivos.
- El capítulo 2, cuerpo del proyecto, donde se redacta el marco teórico y estado del arte. En este capítulo se enfoca en los orígenes de la robótica, automatización, domótica y una introducción a la BeagleBone Black. A parte de explicar los elementos más técnicos del sistema. Y por último el cuerpo del trabajo
- El capítulo 3. La conclusión, donde se describirán tanto las pruebas como los resultados del trabajo. Además se comentaran las posibles mejoras sobre el sistema.

## 2. Cuerpo del trabajo

### 2.1. Introducción

En los últimos años se ha incrementado considerablemente la domótica en nuestros hogares. Dispositivos que se ocupan de las tareas del hogar o que las hacen más fáciles. Como pueden ser enchufes o pulsadores inteligentes, que apaguen y enciendan las luces de casa, suban o bajen las persianas, etc... Estos dispositivos ahora son más accesibles por su precio, instalación y puestas en marchas más sencillas. Dentro de estos dispositivos se encuentran los robots domésticos, también llamados robots de servicios [1].

Algunos de estos robots domésticos tienen la finalidad, como ya hemos contado antes, de facilitar las tareas del hogar, mejorando la calidad de vida y administración del tiempo de los usuarios. Por otro lado, se encuentran los robots que se emplean para el ocio o entretenimiento. Aunque a estos robots es posible asignarles tareas de seguridad, atención a personas mayores y vigilancia.

Los fabricantes mejoran o desarrollan nuevos modelos día tras día, sacando una gran variedad de productos y ampliándose el rango de labores a realizar. Tanto en tareas especiales como en más generales.

Hay multitud de ejemplos de estos robots domóticos, como pueden ser el Roomba de iRobot, el cual es un robot especializado en limpieza de suelos: <https://www.irobot.es/roomba/serie-600>

La conga de Winrobot, es otro robot destinado a la limpieza del hogar. Este especialmente se destina a la limpieza de cristales. Utiliza una potente bomba de succión que le permite mantenerse en posición vertical:

<https://www.storececotec.com/es/aspiradores-limpiacristales/778-conga-windroid-870-connected.html>

Otro ejemplo es el robot Pepper de Aldebran Robotics. Este robot es un humanoide programable y diseñado para interactuar con personas. Su tecnología le permite detectar tanto el lenguaje verbal como el facial, para reconocer el estado emocional de cada interacción [2].

<https://aliverobots.com/robot-pepper/>

## 2.2. Marco teórico y estado del arte

### 2.2.1. Robótica

Se tiene una idea aproximada de lo que es la robótica, se sabe de sus aplicaciones y su potencial, sin embargo, no se conoce el origen de la robótica, ni del significado de la palabra robot. La robotización se encuentra en nuestra vida y lo diferente que es a nosotros en cuanto al aspecto físico.

Según la Real Academia de la Lengua Española, un robot es: “Máquina o ingenio electrónico programable, capaz de manipular objetos y realizar operaciones antes reservadas solo a las personas” (Real Academia de la lengua española, 2018).

Una definición de robótica puede ser el afán del ser humano de buscar alternativas para facilitar y optimizar cualquier tarea productiva, mediante el diseño, construcción, manipulación y programación de robots.

La robótica como se conoce hoy en día, se originó hace miles de años. Pero el nombre de robot surgió mucho después del origen de los autómatas. Antiguamente los robots eran conocidos como autómatas, ya que la robótica no era reconocida como una ciencia.

#### 2.2.1.1. Historia de la robótica

El ser humano tiene la tendencia de mirar siempre al futuro, da igual la época que sea, siempre nos imaginamos lo que está por venir. En esta época imaginamos el futuro con coches voladores, viviendo en otros planetas o con la inteligencia artificial. Pero la robótica siempre ha existido en todas las generaciones.

En la antigua Grecia, en concreto en el año 200 a.c, se puede encontrar los primeros indicios de lo que se podría considerar robótica, que además es uno de los instrumentos más interesantes de la historia. En el año 1900 se encontraron las piezas de un mecanismo bastante complejo, constituido con 82 engranajes. Estas piezas formaban el primer ordenador analógico de la historia, capaz de predecir con bastante exactitud las posiciones astrológicas durante 19 años.

Desde esa momento hemos tenido algunos ejemplos más de robótica, pero siempre girando en torno al concepto de autómata. Un autómata es una máquina que imita la figura y

movimientos de un ser animado.

Estos primeros autómatas surgen en la Edad antigua, más en concreto con las civilizaciones de egipcios y griegos. Estos fueron los primeros en desarrollar la ingeniería y mecánica. Estas civilizaciones conocían los conceptos básicos de mecánica como la rueda, el engranaje, la palanca, además del conocimiento sobre la hidráulica y neumática, para la eficiencia de las construcciones de la época. En el siglo 1 d.c, Herón de Alejandría redactó el primer manuscrito sobre la robótica y se describieron los primeros autómatas. El teodolito, un aparato que era capaz de medir distancias, ángulos y desniveles.

En 1921 es la primera vez en la historia que aparece el nombre de robot. Que viene de la palabra “robota”, en checo significa “trabajo forzado”. Durante la Segunda Guerra Mundial se crearon una gran variedad de mecanismos de control y pilotaje automático. Estas máquinas empezaron a perder la forma humana, llegando así a la actualidad y a los robots industriales. El primer robot industrial se implementó en el 1961. El cual se instaló en una cadena de montaje de " General Motors". Este robot transportaba las piezas fundidas en los moldes hasta la cadena de montaje y las soldaba al chasis del vehículo.

Si se habla de la robótica, es necesario hablar también de la Automatización. Ya que la robótica se encuentra dentro de la rama de Automatización. En los procesos de Automatización se encuentra por ejemplo el paletizado, tarea que se puede realizar mediante un robot.

#### 2.2.1.2. ¿Qué es la Automatización?

La automatización (del griego auto, ‘guiado por uno mismo’) es el uso de elementos o procesos informáticos, mecánicos y electromecánicos, básicamente engranajes, palancas, pequeños motores, relés, contadores y temporizadores, que operan con la mínima, incluso sin intervención del ser humano. La automatización normalmente se utiliza para optimizar y mejorar el funcionamiento de procesos industriales, pero puede utilizarse para la automatización de estadios, infraestructuras de las ciudades, hogares (domótica), etc [3].

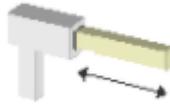
El primer ejemplo de automatización real fue el Telar automático de Joseph Marie Jacquard, quien revolucionó la industria del textil, que en el año 1745 utilizó tarjetas perforadas para automatizar el proceso de tejer. Con la llegada de la máquina de Jacquard

ayudo a los avances para dar con la revolución industrial, la revolución industrial no solo cambio por el uso de máquinas de vapor, sino también cambio la forma de trabajar de las plantas de producción al dividir los procesos. La división de del trabajo permitió que varios obreros trabajasen sobre una parte del proceso de una pieza mientras otros compañeros se encargarían del siguiente proceso. Al optimizar el área y el proceso, se generó la producción en cadena. Este cambio fue importante para la automatización ya que permitía saber en qué procesos podrían ser mejorados con máquinas y cuáles no [4].

El siguiente paso de la automatización llego con la evolución de las computadoras digitales ya que incremento el número de áreas en las que podía realizarse la automatización. Las computadoras digitales con la combinación de velocidad, poder de cómputo, precio y tamaño, fueron apareciendo en la industria en la década de los años 1960. Desde ese momento las computadoras digitales tomaron el control de la mayoría de tareas simples, repetitivas y especializadas. Uno de los sectores que se beneficiaron del estas computadoras digitales fue el sector automotriz e incluso fueron los primeros en utilizar lo que se considera la primera computadora para la automatización, el autómeta programable (PLC). Los primeros PLC consistían en tres componentes, la tarjeta procesadora, la memoria y la tarjeta que resolvía la lógica de control relacionada con los programas en esquemas de contactos. La primera vez que se utilizó el termino PLC “Programmable Logic Controller” fue en 1971 por los ingenieros de Allen-Bradley dándole el nombre “Bulletin 1774 PLC” al autómeta programable .

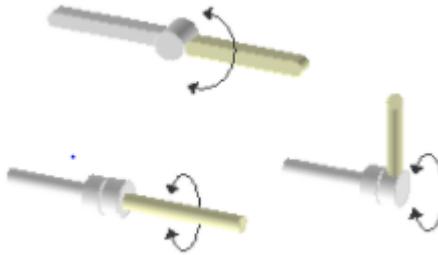
### 2.2.1.3. Clasificación de los Robots

Un robot manipulador está compuesto de una secuencia de elementos estructurales rígidos, denominados enlaces o eslabones. Estos enlaces están conectados entre sí mediante juntas o articulaciones, permitiendo el movimiento relativo de cada dos eslabones consecutivos. Existen dos tipos de articulaciones dependiendo de su movimiento. La articulación lineal (deslizante, traslacional o prismática), si un eslabón desliza sobre un eje solidario al eslabón anterior (Ilustración 1) [5].



**Ilustración 1: Articulación lineal.**

El otro tipo de articulación es la rotacional, en este caso, es el eslabón que gira entorno a un eje solidario al eslabón anterior (Ilustración 2).



**Ilustración 2: Articulaciones rotacionales.**

El conjunto de eslabones y articulaciones se denomina cadena cinemática. Una cadena cinemática es abierta sí, todos los eslabones están conectados exclusivamente al anterior y al siguiente, siendo el primer, el que este fijo a un soporte, y el último, el extremo final. En el extremo es posible conectar una herramienta o actuador final. Este actuador es una herramienta, la cual permite al robot realizar una tarea especial, como puede ser soldar, transportar, pintar, etc. El punto más significativo del elemento terminal se denomina punto terminal (PT). En el caso de una pinza, este punto se encuentra en el centro de sujeción de la herramienta.

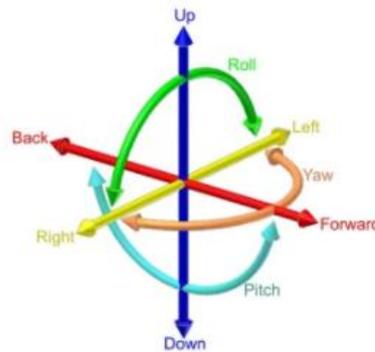
Esta estructura del robot, viene definida por la configuración general del robot y la aplicación a realizar. Esta estructura puede ser metamórfica. Este concepto, se ha introducido para incrementar la flexibilidad funcional de un robot a través del cambio de su configuración por sí mismo. Esta configuración permite diferentes tipos, desde los más básicos (cambio de efector final), hasta los más complejos, que permiten el cambio de algunos de sus elementos estructurales. Los dispositivos y mecanismos que pueden agruparse bajo la denominación genérica de robot, tal como se ha indicado, son muy diversos. Con lo cual se ha realizado una subdivisión compuesta de poliarticulados,

móviles, androides, zoomórficos e híbridos.

- **Poliarticulados:** los robots de esta división tienen la característica común de ser robots fijos, es decir, no se pueden desplazar por los ejes. En esta clasificación se encuentran los robots manipuladores, robots cartesianos y robots industriales.
- **Móviles:** Estos robots, a diferencia de los anteriores, tienen la capacidad de desplazamiento mediante carros o plataformas. Estos robots son capaces de desplazar piezas de un punto a otro de una cadena, pudiendo incluso sortear obstáculos mediante la inteligencia artificial.
- **Androides:** Estos son los robots más reconocidos por la gente, ya que intentan simular la forma y los movimientos del cuerpo humano, parcial o totalmente. Estos robots androides aún están poco evolucionados y sin utilidad práctica. El aspecto más complejo de estos dispositivos es la locomoción bípeda.
- **Zoomórficos:** Esta clase también podría incluir a los androides, ya que su principal característica es sus sistemas de locomoción, que imitan a diversos seres vivos. En esta existen dos tipos: caminadores y no caminadores. Este último grupo se encuentra muy poco desarrollado. En cuanto al grupo de caminadores, son muy numerosos y están siendo objeto de muchos experimentos. Ya que algunos de ellos llegan a ser verdaderos todoterrenos pudiendo adaptarse a cualquier medio.
- **Híbridos:** Esta es una difícil clasificación, que se encuentra entre algunas de las anteriores, o en combinación de varias. Por ejemplo, un robot segmentado articulado y con ruedas, es al mismo tiempo, un robot móvil y un robot zoomórfico.

El movimiento en un espacio tridimensional, es decir, la capacidad de moverse hacia arriba/abajo, izquierda/derecha, delante/atrás, combinados con la rotación sobre tres ejes perpendiculares. Este movimiento a lo largo de cada uno de los ejes es independiente de

los otros, y cada uno es independiente de la rotación sobre cualquiera de los ejes, el movimiento de hecho tiene seis grados de libertad.



**Ilustración 3: Los 6 grados de libertad.**

También es posible clasificar los robots por sus grados de libertad. Este número normalmente se refiere al número de un solo eje de rotación de las articulaciones en el brazo, donde un mayor número indica una mayor flexibilidad en un robot. Esta medida es muy práctica, ya que, mide la capacidad de posicionamiento del robot.

Se denomina grado de libertad (g.d.l) a cada una de las coordenadas independientes, que son necesarias para describir el sistema mecánico de un robot. Es decir, describir la posición y orientación en un espacio.

Para el control y el estado del robot es necesario saber la posición de un punto del robot respecto de un sistema de coordenadas externo y fijo. Y el movimiento del brazo cuando los actuadores aplican sus fuerzas. El análisis mecánico de un robot se realiza mediante el estudio cinemático, el cual estudia sus movimientos, o atendiendo las fuerzas y momentos que actúan sobre los eslabones, estudio dinámico.

### 2.2.2. ROS

ROS (Robotic Operating System) es un middleware robótico, es decir, una agrupación de frameworks, creado para facilitar la programación de robots a través de una gran variedad de plataformas robóticas. La gran ventaja de ROS es que es un sistema de código abierto. Este sistema está compuesto de herramientas librerías y convenciones. La programación de robots antes de ROS era una tarea bastante compleja ya que era necesario tener un gran conocimiento de varios temas para solucionar los problemas triviales que surgían en la programación del robot.

ROS fue creado en 2007 bajo el nombre de switchyard por el laboratorio de inteligencia artificial de Stanford. Desde 2008 ROS se convirtió en una entidad formal a cargo de Willow Garage y en 2009 se publica el primer artículo en el cual se presenta ROS como un sistema operativo de código abierto.

Actualmente ROS se desarrolla en un entorno colaborativo, el cual cualquier persona con conocimientos sobre el mundo de la robotica puede ayudar a desarrollar funcionalidades, gracias a que ROS es de código abierto. Con lo cual, ROS se ha creado de manera modular para que los usuarios puedan programar sus proyectos con los paquetes que necesiten de forma sencilla. Este software es libre bajo términos de licencia BSD. Esta licencia permite libertad para uso comercial e investigador. Las contribuciones de los paquetes en ros-pkg están bajo una gran variedad de licencias diferentes.

En este momento ROS solo se puede ejecutar en sistemas basados en Unix (Ubuntu – Linux) y en W10 mediante virtualización. El software para ROS se prueba principalmente en sistemas Ubuntu y Mac OS X, aunque la comunidad ROS ha estado contribuyendo con soporte para Fedora, Gentoo, Arch Linux y otras plataformas Linux.

#### 2.2.2.1. Arquitectura ROS

A continuación se va a explicar cómo está compuesta la arquitectura del sistema ROS para poder comprender mejor el funcionamiento de este sistema. Este aparatado se compone de 3 secciones:

- Sistema de archivos
- Grafo computacional
- Comunidad

En el sistema de archivos está organizado en carpetas, las cuales agrupan archivos encargados de desarrollar las distintas funciones. La carpeta más importante de ROS es el Packages.

- **Packages** (Paquetes) es el contenido mínimo necesario para crear un programa. Esta es la unidad principal para organizar el software de ROS. Un paquete puede contener dependencias, los archivos de configuración o procesos en tiempo de ejecución (nodos). Dentro de esta carpeta se encuentra el Manifests, el cual,

contiene la información del paquete. Esta información puede ser tanto las dependencias como las licencias.

- **Manifests** son los que proporcionan la información de un paquete, como puede ser el nombre de este, la versión, licencias, descripción y dependencias. El archivo donde se gestionan los manifest es manifests.xml.
- Los repositorios son una colección de paquetes que comparte un sistema VCS común.
- **Message** (msg) type, son mensajes que manda un proceso a otro proceso. Existen varios tipos de mensajes predeterminados, y también es posible crear nuevos tipos de mensajes.
- **Service** (srv) types, son las descripciones de los servicios, estos servicios definen las estructuras de datos petición/respuesta para la comunicación entre nodos.

El apartado de grafo computacional es la red peer-to-peer de procesos en ROS. En esta red todos los procesos están conectados, provocando que cada nodo pueda enviar y recibir datos entre sí. Los conceptos básicos de este apartado son los nodos, maestro, servidor de parámetros, mensajes, etc...

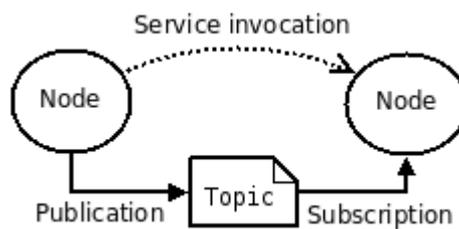
Los nodos son ejecutables que realizan interacciones con los otros nodos. Para ello, utilizan el parameter server, messages y topics. Ros está diseñado para ser modular, es decir poder separar la programación por funcionalidades, consiguiendo un código mucho más sencillo. Los nombres de los nodos son únicos, identifican a cada nodo para poder realizar las comunicaciones con los otros nodos. Un nodo Ros se pueden escribir en C++, con la librería roscpp, o con la librería rospy en Phyton.

El Maestro en ROS proporciona toda la información de registros de nombres para que los nodos sean identificados. Sin el Maestro los nodos no podrían encontrarse, intercambiar mensajes o invocar servicios.

El parameter server, permite almacenar los datos de manera centralizada. Actualmente formar parte del Master.

Messages (msg) se utilizan para realizar la comunicación entre nodos. Un msg es simplemente una estructura de datos, en la cual se admiten los tipos primitivos (booleano, entero, punto flotante, etc.), como las matrices. Pueden ser matrices y estructuras anidadas. Estos messages son publicados por topics.

Los topics son los buses de comunicación para transmitir messages entre los nodos. Estas comunicaciones son del tipo publicación/suscripción. Los nodos se suscriben a un topic para recibir la información o enviarla. Pueden existir varios publicadores y suscriptores simultáneos para un mismo tema y un solo nodo puede publicar y/o suscribirse a múltiples temas. Los topics pueden transmitir sin la necesidad de tener una conexión directa. En la ilustración 4 se muestra como es el esquema de comunicación entre nodos mediante topics, ya que hay un nodo publicando y otro suscrito a ese mismo topic.



**Ilustración 4: Esquema de comunicación entre nodos.**

Servicios: las comunicaciones entre nodos mediante topics son unidireccionales, es decir el suscriptor no responder al publicador. Los servicios son del tipo solicitud respuesta entre nodos. Estos servicios se definen mediante un par de estructuras de mensajes, una para la solicitud y otra para la respuesta. En la ilustración 3 se puede observar que la comunicación cuando se solicita un servicio es directa entre los nodos, sin ningún topic de por medio.

Las Bags son un formato para guardar y reproducir datos de mensajes. Se pueden utilizar para visualizar lo que sucedió en una ejecución. Se suele utilizar como depuración de código, para desarrollar y probar algoritmos.

El maestro de ROS actúa como un servicio, es decir, almacena información de registro de temas y servicios para los nodos. Estos nodos se comunican con el maestro para reportar su información de registro. Además, pueden recibir información sobre otros nodos registrados y realizar conexiones según corresponda. El maestro también hará devoluciones de llamada a estos nodos cuando cambie esta información de registro, lo que permite a los nodos crear conexiones dinámicamente a medida que se ejecutan nuevos nodos.

El protocolo más común utilizado en ROS se llama TCPROS, que utiliza sockets TCP/IP estándar.

En el siguiente apartado trata sobre la comunidad de ROS. Esta comunidad juega un papel crucial para ROS, en ella se desarrolla las nuevas distribuciones del sistema. Estos recursos se incluyen:

**Las distribuciones:** Son las colecciones de versiones de ROS disponibles para ser instaladas.

**Repositorios:** Ros se basa en una red federada de repositorios de código, donde se permite a los diferentes usuarios poder crear su propio robot y acceder a diferentes ejemplos de robots.

**La Wiki de ROS:** La Wiki de la comunidad de ROS es el foro principal para encontrar la información fundamental sobre ROS. Cualquiera puede registrarse para contribuir con su propia documentación, actualizaciones o correcciones, tutoriales y más.

**Contacto por email:** Creado para resolver las dudas por las personas implicadas en el desarrollo de ROS. Esta función está disponible en páginas de respuestas como puede ser: ROS answers [6].

### 2.2.3. Domótica

El término domótica viene de la unión de las palabras “Domus” (en la latín significa casa) y autónomo (palabra en griego, “que funciona por sí solo”). Según la asociación española de domótica e inmótica, “La domótica es el conjunto de tecnologías aplicadas al control y automatización inteligente de la vivienda, que permite una gestión eficiente del uso de la energía, que aporta seguridad y confort, además de comunicación entre el usuario y el sistema”. La domótica es por lo tanto un conjunto de tecnologías aplicadas al control y la automatización de la vivienda o edificación de cualquier tipo, tratando de buscar servicios de ahorro energético, seguridad, confort, bienestar y comunicación como representa la ilustración 5.



Ilustración 5: Características de la domótica.

Un sistema domótico es capaz de recibir información proveniente de unos sensores o entradas, procesarla y actuar sobre salidas o actuadores. El sistema puede acceder a redes exteriores de comunicación o información.

Los primeros pasos de la domótica pertenecen a los años 80 a nivel comercial, pero cuando realmente empezó a comercializarse para el uso en casas urbanas fue en los años 90, coincidiendo con la evolución y despliegue de internet. En los primeros países donde se implementó la domótica fue en Japón, Estados Unidos y algunos países del norte de Europa. A medida que fue pasando el tiempo fueron desarrollándose nuevos sistemas domóticos. Estas mejoras eran totalmente autónomas, sin buscar en ningún momento la comunicación con otros dispositivos de hogar. Pero a partir de la entrada de internet en la

mayoría de los hogares se cambió la forma de interpretar la domótica, buscando integrar los sistemas eléctricos y electrónicos y una comunicación integral de todos los dispositivos del hogar.

La llegada de la domótica a España fue gracias a los organismos e instituciones que, asumiendo un riesgo económico considerable, apostaron por esta tecnología. Las principales entidades son: Ministerio de Fomento, CEDOM (Comité Español para la Gestión Técnica de Edificios y Viviendas), EIBA (Asociación del Bus de Instalaciones Europeas), AFME (Asociación de Fabricantes de Materiales Eléctricos), IDEA (Instituto para la diversificación y ahorro de Energía).

El avance de la tecnología ha proporcionado que hoy en día los sistemas domóticos son relativamente baratos, comparados con los de hace algunos años. La domótica ofrece una gran versatilidad y gradualidad habiendo viviendas poco automatizadas con pocos sensores y actuadores, a viviendas con todo automatizado. Otra ventaja es la modularidad, ya que se puede dividir el sistema en varios subsistemas como la climatización, iluminación, etc.

Los servicios que ofrece la domótica se pueden agrupar en diferentes categorías: ahorro energético, seguridad, confort y comunicaciones. A menudo existen situaciones que es difícil clasificar un servicio estrictamente en una sola porque pueden englobar a varias a la vez.

El ahorro energético es algo que en un primer momento puede no notarse a corto plazo, pero a largo plazo, sí. Para que esta aplicación surta efecto, no es necesario cambiar los componentes instalados en la vivienda sino un uso adecuado tanto de tiempo como de potencia para reducir su consumo. De acuerdo con los cálculos del IDEA, es posible ahorrar más de 50% del consumo eléctrico de un hogar en función de múltiples variables. Otra medida puede ser la programación, planificación y zonificación de la climatización, automatizar el control de los toldos para proteger la vivienda del sol, apagar la iluminación en zonas deshabitadas o derivar el funcionamiento de algunos aparatos eléctricos a horas de tarifa reducida.

En el ámbito de seguridad, lo que se pretende es evitar los robos como minimizar los daños en la vivienda ante cualquier percance. Este tipo de situaciones puede referirse a detección de gases, agua, humos, etc. También existen funcionalidades más innovadoras, como el botón del pánico, que al accionarlo se ilumina completamente la casa y envía una señal de alarma, la simulación de presencia, actúa sobre la iluminación, persianas, toldos, etc para

que desde fuera la casa parezca habilitada, claves y niveles de acceso configurables, o llaves de acceso por proximidad.

Para ofrecer un mayor confort y bienestar se engloban todos los servicios que mejoren las comodidades de vida dentro de la vivienda, o incluso fuera de ella. Dichas actuaciones pueden ser de carácter tanto pasivo como activo o mixto. Las funciones de confort ayudan a realizar o reducir las tareas de la casa, ofreciendo la posibilidad de controlar todos los dispositivos instalados desde un dispositivo central o un dispositivo móvil (ordenador o móvil con conexión a internet) automatizando y simplificando así su uso y gestión. También permite controlar o automatizar el apagado/encendido de cada punto de luz, regular la iluminación en función de la luz exterior o regular la temperatura a nuestro gusto.

En las comunicaciones lo buscado es la capacidad para establecer conexiones con la vivienda para saber el estado de la misma sin necesidad de estar en ella y poder modificarlas. Al tener conectados todos los sistemas de comunicación, permite tener el control remoto de todos los dispositivos tanto dentro como fuera de la vivienda, recibir informes de consumo y costes en tiempo real.

Actualmente, estos son los cuatro ámbitos en los que se centra principalmente la domótica. Sin embargo, la tecnología va evolucionando y van apareciendo nuevas necesidades y sistemas que se pueden incorporar a las viviendas inteligentes. A pesar de todas las ventajas y comodidades que la domótica nos brinda, el número de hogares domotizados es todavía bajo con respecto al número de viviendas totales, aunque la bajada de precios está ayudando a que cada vez se implementen más estos servicios. No cabe duda que en un futuro la domótica conquistará todos los hogares, haciendo la vida más fácil, confortable, respetuosa con el medio ambiente y segura.

## 2.2.4. Plataformas de bajo coste

El proyecto necesita de un procesador para realizar todo el control sobre los elementos dependientes del sistema. En el mercado existen una multitud de estas tarjetas de bajo coste. Cada una con unas características y diseñadas para unas tareas u otras. En este apartado se va a comentar tres tarjetas disponibles para la realización del proyecto. Eligiendo una final. Se han elegido estos microprocesadores por su precio, ya que deben de ser de bajo coste y por su disponibilidad en mi empresa. Estas tres plataforma son la BeagleBone Black, raspberry pi 2 y Arduino ESP8266

### 2.2.4.1. BeagleBone Black

La BeagleBone Black (BBB) es una tarjeta de bajo coste desarrollado por Texas Instrument con arquitectura ARM Cortex A8. Como otras plataformas del mismo tipo, está pensada para experimentar y aprender a programar el procesador y a acceder a los periféricos mediante la creación propia de software y hardware. Beagleboard nace como un foro de desarrolladores software y hardware, donde pueden intercambiar ideas, conocimientos y experiencia (Ilustración 6).

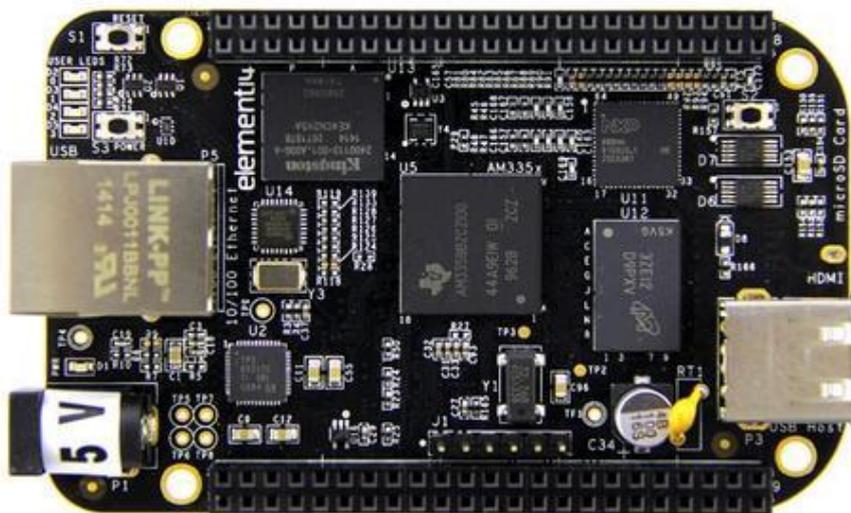


Ilustración 6: BeagleBone Black

El procesador de la BeagleBone Black es Sitara AM3359, desarrollado por TI y que contiene una CPU ARM Cortex-A8 trabajando a una frecuencia de reloj de 1 GHz. EN cuanto a la memoria, contiene 512MB de SDRAM DDR3 y 4GB de memoria flash (eMMC, embedded MultiMedia card) en la que trae instalado por defecto el sistema operativo Angstrom. También cuenta con un potente procesador gráfico que soporta video de alta definición de hasta 1080p de resolución [7].

La BBB dispone diferentes formas de conexionado. Cuenta con un puerto miniUSB 2.0 en modo cliente para la alimentación, este mismo puerto es también USB 2.0 normal para conectar periféricos y un puerto 10/100 Ethernet con conector RJ45, ranura para conectar tarjetas de memorias microSD, HDMI y 92 pines de expansión divididos en dos. Mediante los 92 pines, se dispone de 65 entradas/salidas digitales, que también pueden ser configuradas en 8 modos diferentes, como GPIO, UART, PWM, SPI, I2C (Ilustración 7). También es posible adaptarle diversos accesorios compatibles llamados Capes. Estos accesorios son desde pantallas táctiles LCD de varias pulgadas a módulos GPS o WiFi.

P9				P8			
7	1	2	7	7	1	2	7
DGND	1	2	DGND	DGND	1	2	DGND
VDD 3.3	3	4	VDD 3.3	MMCL_DAT6(1)	3	4	GPIO_39
VDD 5V	5	6	VDD 5V	MMCL_DAT2(1)	5	6	GPIO_35
SYS 5V	7	8	SYS 5V	TIMER4(2)	7	8	GPIO_67
PWR_BUT	9	10	SYS_RESETN	TIMERS(2)	9	10	GPIO_68
UART4_RX(6)	GPIO_30	28	GPIO_60	GPIO_45	13	12	GPIO_44
UART4_TX(6)	GPIO_31	29	GPIO_40	GPIO_23	9	14	GPIO_26
PWM_TRIPZ_IN	GPIO_48	16	GPIO_51	PWM_2B(4)	GPIO_47	15	GPIO_46
I2C-1_SCL(2)	GPIO_4	87	GPIO_5	PWM0_SYNCO	GPIO_27	11	GPIO_65
I2C-2_SCL(3)	GPIO_13	95	GPIO_12	PWM_2A(4)	GPIO_22	8	GPIO_63
UART2_TX(1)	GPIO_3	85	GPIO_2	MMCL_CLK(2)	GPIO_62	32	MMCL_DAT5(1)
PWM0_SYNCO	GPIO_49	17	GPIO_15	MMCL_DAT4(1)	GPIO_36	4	MMCL_DAT1(1)
SPI-1_D0(3)	GPIO_117	107	GPIO_14	MMCL_DAT0(1)	GPIO_32	0	GPIO_61
SPI-1_SCL(3)	GPIO_110	100	GPIO_113	LCD_VSYNC(0)	GPIO_86	56	LCD_PCLK(0)
AIN4	33	34	GPIO_112	LCD_HSYNC(0)	GPIO_87	57	LCD_AC_BIAS(0)
AIN6	35	36	GPIO_112	LCD_DATA14(0)	GPIO_10	54	LCD_DATA15(0)
AIN2	37	38	GPIO_112	LCD_DATA13(0)	GPIO_9	53	LCD_DATA11(0)
AIN0	39	40	GPIO_112	LCD_DATA12(0)	GPIO_8	52	LCD_DATA10(0)
GPIO_20	41	42	GPIO_7	LCD_DATA8(0)	GPIO_78	48	LCD_DATA9(0)
DGND	43	44	DGND	LCD_DATA6(0)	GPIO_76	46	LCD_DATA7(0)
DGND	45	46	DGND	LCD_DATA4(0)	GPIO_74	44	LCD_DATA5(0)
				LCD_DATA2(0)	GPIO_72	42	LCD_DATA3(0)
				LCD_DATA0(0)	GPIO_70	40	LCD_DATA1(0)

Ilustración 7: Pin Out BBB

La BBB trae instalada en la memoria flash el sistema operativo Angstrom. Pero no es el único sistema operativo (OS) que soporta. Es posible sobrescribir la memoria flash e instalar Debian, Android, Ubuntu, Gentoo, Minix o ArchLinux entre otros (Tabla 1).

Modelo BeagleBone	Black
Precio	45\$
Procesador	AM335x ARM Cortex-8
Máxima velocidad del procesador	1GHz
Entradas Analógicas	7(1.8V)
Entradas/salidas digitales	65(3.3V)
Memoria	512MB DDR3. 4GB on-board storage using eMMC, microSD card slot
USB	MiniUSB 2.0 client port, USB 2.0 host port
Video	MicroHDMI
Audio	MicroHDMI
Periféricos	4x UART, 8x PWM/Timers, LCD, GPMC, MMC1, 2x SPI, 2x I2C,A/D converter, 2xCAN Bus.

Tabla 1: Información sobre la BeagleBone Black extraída de [beagleboard.org](http://beagleboard.org)

#### 2.2.4.2. Raspberry pi 2

Raspberry Pi es una placa de microordenador de pequeñas dimensiones, la cual tiene multitud de utilidades. El origen de esta placa se encuentra en el año 2009 en Reino Unido, donde nació una organización llamada Fundación Raspberry Pi. El objetivo de esta organización era animar a los niños a aprender informática en las escuelas. Aunque la mayoría de escuelas y niños pueden acceder a un ordenador, no es un recurso asequible al que puedan acceder todos los centros ni todos los países. Lo que intenta Raspberry pi es que este problema sea más sencillo. Con esta tarjeta es posible realizar proyectos de electrónica como tareas básicas de cualquier ordenador, por ejemplo, navega en la red, hojas de cálculo, etc... [8]

Raspberry Pi 2B (ilustración 8) es la segunda generación de Raspberry Pi, este modelo salió al mercado reemplazando el 1B+ en febrero del 2015. Es aspecto físico y medidas son idénticas a su antecesor pero mejora el rendimiento de la versión anterior. Esta placa es bastante completa, cuenta con un procesador ARMv7, 4 puertos USB, conexión HDMI, Ethernet y mucho más. Debido al procesador, es posible cargarle distribuciones de ARM GNU/Linux [7].



Ilustración 8: Raspberry Pi1.

La placa cuenta con un conector Ethernet RJ 45 (10/100 Mbps), los cuatro puertos USB son 2.0, dos conectores para cables planos de 15 vías (CSI y DSI) para poder añadir módulos de cámara o una pantalla táctil, un jack de 3.5mm A/V para audio y video compuesto, una ranura para tarjetas microSD, conector microUSB para la alimentación de 5V a 2A y 40 pines GPIO (Ilustración 9). La placa tiene unas medidas de 86 mm por 56 mm con un peso de 45 gramos [9].

Raspberry Pi 2 Model B (38 Header)					
GPIOPIN	NAME		NAME	GPIOPIN	
	3.3 VDC Power	⚡	5.0 VDC Power		
<b>8</b>	GPIO 8 SDA1 (I2C)	⚡	5.0 VDC Power		
<b>9</b>	GPIO 9 SCL1 (I2C)	⚡	Ground		
<b>7</b>	GPIO 7 GPCLK0	⚡	GPIO 15 Tx0 (UART)	<b>15</b>	
	Ground	⚡	GPIO 16 Rx0 (UART)	<b>16</b>	
<b>0</b>	GPIO 0	⚡	GPIO 1 PCM_CLKPWMB	<b>1</b>	
<b>2</b>	GPIO 2	⚡	Ground		
<b>3</b>	GPIO 3	⚡	GPIO 4	<b>4</b>	
	3.3 VDC Power	⚡	GPIO 5	<b>5</b>	
<b>12</b>	GPIO 12 MOSI (SPI)	⚡	Ground		
<b>13</b>	GPIO 13 MISO (SPI)	⚡	GPIO 6	<b>6</b>	
<b>14</b>	GPIO 14 SCLK (SPI)	⚡	GPIO 10 CE0 (SPI)	<b>10</b>	
	Ground	⚡	GPIO 11 CE1 (SPI)	<b>11</b>	
<b>30</b>	SDA0 (I2C ID EEPROM)	⚡	SCL0 (I2C ID EEPROM)	<b>31</b>	
<b>21</b>	GPIO 21 GPCLK1	⚡	Ground		
<b>22</b>	GPIO 22 GPCLK2	⚡	GPIO 26 PWM0	<b>26</b>	
<b>23</b>	GPIO 23 PWM1	⚡	Ground		
<b>24</b>	GPIO 24 PCM_FSPWM1	⚡	GPIO 27	<b>27</b>	
<b>25</b>	GPIO 25	⚡	GPIO 28 PCM_DIN	<b>28</b>	
	Ground	⚡	GPIO 29 PCM_DOUT	<b>29</b>	

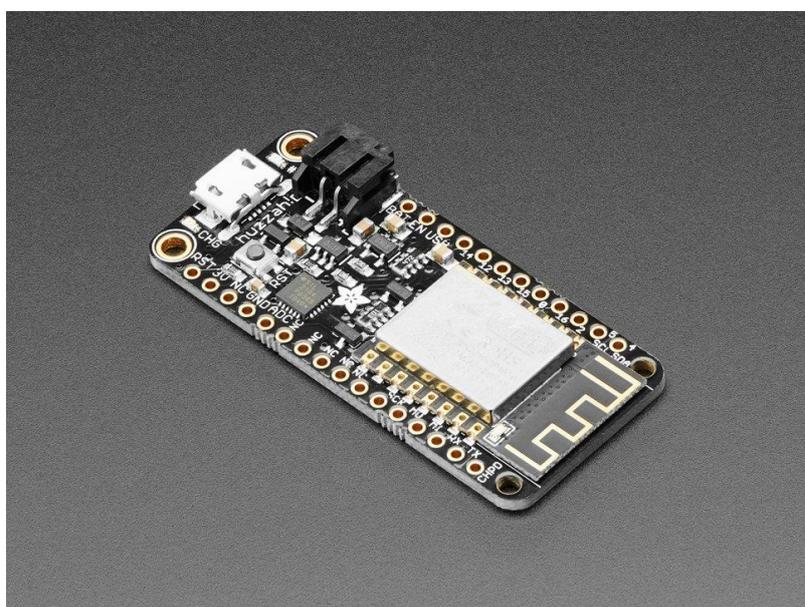
**Attention!** The GPIO pin numbering used in this diagram is intended for use with WiringPi / Pi4J. This pin numbering is not the raw Broadcom GPIO pin numbers.

<http://www.pi4j.com>

Ilustración 9: Raspberry Pi 2B pinout

### 2.2.4.3. Arduino ESP8266

Por último, contamos con el ESP8266, el cual, es un microcontrolador con WIFI incorporado. Esta placa es un Soc (System on Chip). La empresa creadora de este chip es Espressif, una empresa de Shanghái, China. Salió a la venta en 2014 con una versión más básica. A mediados de 2016 la empresa anuncio una nueva versión, la ESP8285, con la gran diferencia que incorpora una memoria Flash de 1MB integrada dentro del propio chip. El ESP8266 no dispone de esta memoria y necesita de un módulo externo para almacenar los programas. En este proyecto se estudió la placa Adafruit Feather HUZZAH ESP8266 (ilustración 10) [10].



**Ilustración 10: Adafruit Feather HUZZAH ESP8266**

Adafruit Feather HUZZAH ESP8266 es una placa de desarrollo con el chip ESP8266 incorporado. Esta placa es ‘todo en uno’ con USB integrado y carga de batería. Este chip ESP8266 cuenta con una frecuencia de reloj de 80 MHz y lógica de 3.3V. Es posible programar el microcontrolador utilizando el software IDE de Arduino. También tiene un conector de baterías de litio de 3.7V y carga de batería incorpora, aparte de la alimentación desde el micro USB a 5V. La placa mide 51 mm por 23 mm por 8mm, con un peso de 9,7 gramos, 4MB de memoria Flash, WiFi 802.11 b/g/n 9 pines (Ilustración 11), que pueden ser utilizados como I2C o SPI, una entrada analógica de 1V como máximo y botón de reset [11].

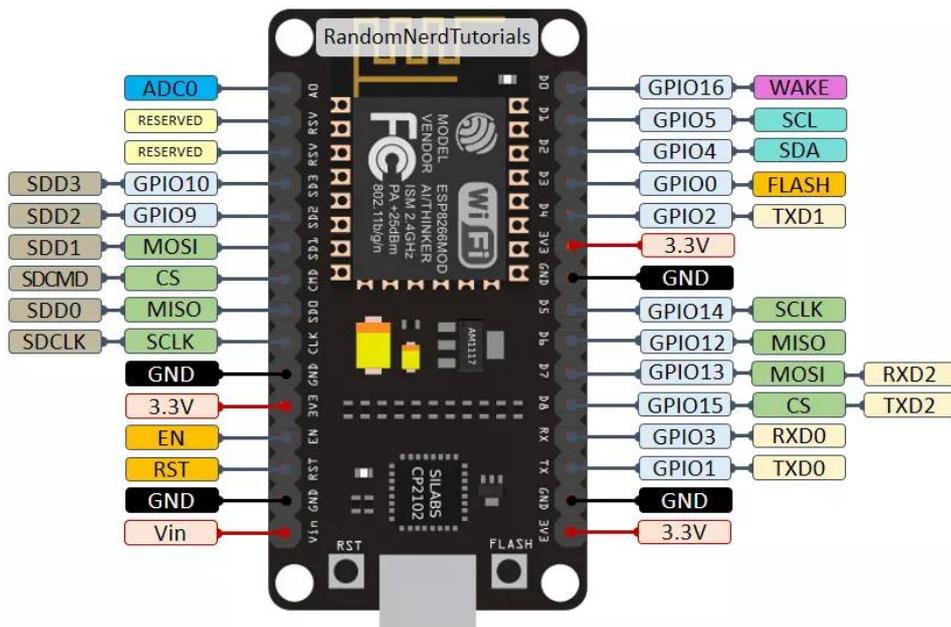


Ilustración 11: Adafruit Feather Huzzah ESP8266 pinout

#### 2.2.4.4. Comparativa

Después del estudio de las plataformas disponibles para realizar este proyecto, nos hemos decantado por una por varias razones. En el mercado se encuentran muchas más opciones pero solo son estas tres las que se han estudiado.

El sistema necesita 7 salidas digitales con el modo de trabajo de PWM, para el movimiento de las 6 articulaciones del robot. En la raspberry PI únicamente cuenta con 4 salidas PWM, a diferencia de la BeagleBone Black que incorpora en su I/O 7 PWM. Y el microchip ESP8266 no cuenta con ninguna salida PWM. Este no sería un problema, ya que, en el mercado existen extensiones o módulos con PWM y sería posible añadir estos módulos por comunicaciones I2C a nuestra placa. Respecto al precio, la placa más barata es la ESP8266, después la Raspberry Pi y luego la BeagleBone Black.

Al final, la plataforma de bajo coste es la BBB, ya que el encuentro más robusta, porque es posible cargarle el sistema operativo en su memoria, a diferencia de la Raspberry Pi que necesitas una microSD introducida con el SO. Además la BBB cuenta con los 7 PWM y no es necesario ningún modulo extra.

### 2.2.5. CODESYS

CODESYS es un entorno de desarrollo para la programación de controladores conforme con el estándar industrial internacional IEC 61131-3. El término CODESYS (Controller Development Systems). CODESYS es desarrollado y comercializado por la empresa de software alemán 3S-Smart de Soluciones de Software situado en la ciudad bávara de Kempten. La primera Versión (1.0) fue creada en 1994. CODESYS es una plataforma con un lenguaje unificado para la programación de autómatas y controladores.

El Software es muy utilizado por empresas del mundo de la Automatización Industrial que lo utilizan en sus equipos de automatización: BeckHoff, Festo, EATON, KEBA, IFM, LENZE, Schneider Electric, ABB, ESA.

CODESYS utiliza el estándar de programación de PLC IEC 61131-3 con los cinco lenguajes de programación lo que asegura que la programación cumpla la norma. Los 5 lenguajes de programación que utiliza CODESYS son:

- Lenguaje de Contactos (LD- Ladder Diagram): Permite al programador combinar los contactos de relé y las bobinas. Es el lenguaje de Programación de PLC por excelencia.
- Diagrama de bloque de funciones (FBD- Function Block Diagram): Permite al usuario programar rápidamente, tanto expresiones como en lógica booleana.
- Texto estructurado (ST - Structured Text): Es similar a la programación en C.
- Lista de instrucciones (IL – Instruction List): Es un lenguaje de programación parecido al lenguaje ensamblador.
- Bloques de función secuenciales (SFC – Sequential Function Chart): Es conveniente para los procesos de programación secuencial.
- CFC (Continuous Function Chart): Es un editor de EBD libre. Es un editor orientado a FBD donde las conexiones entre las entradas, salidas y operadores se fijan automáticamente. Todas las cajas se pueden colocar libremente, lo que permite programar ciclos de retroalimentación provisional sin variables. Es un lenguaje que no está definido en la norma IEC.

CODESYS dispone una versión DEMO 100% operativa que permite aprender a programar autómatas basados en el lenguaje IEC 61131-3. El simulador integrado permite testear y simular los programas, tanto del PLC como de pantallas. CODESYS Depictor permite insertar simulación 3D. También dispone de editor HMI para programar interfaz gráficos para pantallas o terminales de operador. Esto permite poder realizar el programa del autómata y su correspondiente interfaz gráfico en un mismo software. También es posible la visualización en web con CODESYS WebVisu.

### 2.2.6. Modbus

Modbus es un protocolo de comunicación industrial. Este protocolo se ha convertido en un protocolo de comunicaciones estándar en la industria. El objetivo de este protocolo es la transmisión de información entre distintos equipos electrónicos conectados a un mismo bus. Modbus utiliza una arquitectura de maestro/esclavo. El maestro inicia las comunicaciones preguntando datos al esclavo, que siempre responde en función de la pregunta hecha por el maestro. Cada dispositivo posee una dirección única. Los comandos básicos del maestro Modbus modifican el valor de alguno de sus registros o solicitar el contenido de dicho registro. Las funciones básicas del protocolo Modbus son lectura de bits, lectura de palabras, escritura de bits y escritura de palabras.

Modbus es un protocolo de aplicación, por lo cual se puede implementar sobre diferentes capas físicas. Las dos versiones en las que podemos implementar este protocolo son, TCP/IP, que se implementara con cables de red, y RTU y ASCII el cual esta implementado por cables serie.

#### 2.2.6.1. Modbus TCP/IP

Modbus TCP/IP es una variante utilizada para comunicarse sobre una red Ethernet, conectándose a través del puerto 502 y normalmente usando comunicación half-duplex sobre una conexión dada. Es decir, no hay beneficio en enviar solicitudes adicionales sobre una conexión única mientras espera una respuesta está pendiente. Esta variante no requiere suma de verificación (checksum).

La trama de esta modalidad está compuesta por:

- Identificador de la transacción con longitud de 2 bytes.
- Identificador del protocolo con longitud de 2 bytes.
- Campo de longitud, indica el número de bytes de la trama. Tiene 2 bytes de longitud.
- Identificador de unidad, es la dirección del esclavo. 1 byte de longitud.
- Código de función, es el número de la función a utilizar. 1 byte de longitud.
- Byte de datos, datos de la respuesta o comandos.

#### 2.2.6.2. Modbus RTU

Modbus RTU es la implementación más común de Modbus en la industria. Se utiliza la comunicación serie y hace uso de una representación binaria compacta de los datos para el protocolo de comunicación. La capa física normalmente a utilizar en esta variante de Modbus es RS-485. El formato de trama es la siguiente:

- Inicio, es un tiempo de silencio de al menos 28 bits.
- Dirección, dirección del esclavo. 8 bits de longitud.
- Función, código de la función con una longitud de 8 bits.
- Datos, los datos a enviar.
- CRC, verificación de redundancia cíclica. Es decir, una suma de comprobación.

Fin, un tiempo de silencio de 28 bits.

#### 2.2.6.3. Modbus ASCII

Modbus ASCII se utiliza en la comunicación serie, normalmente por cable RS-485 y hace uso de caracteres ASCII para el protocolo de comunicación. La trama utilizada por Modbus ASCII es:

- Inicio, comienza con “:”. 1 byte.
- Dirección, dirección del esclavo o estación. 2 byte.
- Función, código de la función a solicitar. 2 byte.

- Datos, datos a enviar.
- LRC, suma de verificación de redundancia longitudinal. 2 byte de longitud.
- Fin.

### 2.2.7. Descripción del sistema

En este apartado se procede a describir los elementos implicados en el sistema. Estos elementos están escogidos ya sea por sus características que son acordes con las necesarias para el proyecto o porque se encontraban disponibles en la oficina de IRYS Control y es posible su utilización en el proyecto.

#### 2.2.7.1. Robot SNAM6800

El robot elegido para el proyecto es el Robot SNAM6800, que consta de 6 grados de libertad. El brazo está compuesto de 6 articulaciones, todas ellas rotacionales. El control de las articulaciones se realiza mediante servomotores (TIANKONGRC TD-8120MG), menos las dos últimas articulaciones que tienen unos servos más pequeños. A su vez, estos servos reciben el control desde la BeagleBone por PWM. Además el brazo cuenta con un sistema de vacío, con una herramienta final para la palatización. Esta herramienta consta de 4 ventosas, en las esquinas. El sistema de vacío se actúa a partir de dos motores de corriente continua (LY370DPM30 3V a 6V).

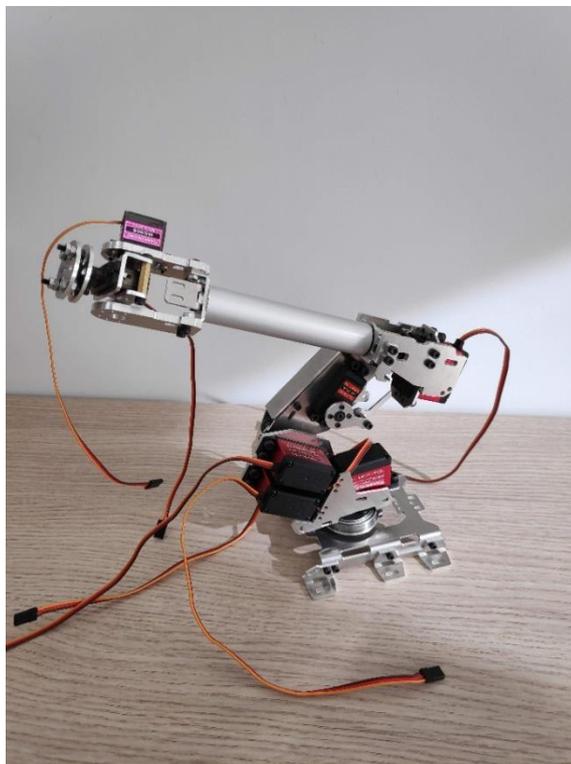


Ilustración 12: ROBOT SNAM6800

Los materiales del brazo son de aluminio y plástico. Cada articulación esta actuada por un solo servo, menos la 2 articulación, que consta de 2 servos. Los primeros 5 servomotores son de 20KG y los dos últimos de 9g. El peso total del Robot es de unos 1064 g.

La elección del robot estaba marcada por dos requisitos, uno de ellos era de presupuesto y la otra de potencia. Es decir, necesitaba un brazo robótico por menos de 200€ y que su capacidad de carga fuera de 200g El requisito de la capacidad de carga a levantar, es necesario para poder abrir y cerrar la tapa donde se introduce la capsula de café en la cafetera. Esta tapa es rígida y hay que realizar fuerza para cerrarla con la capsula dentro. Teniendo en cuenta estas características y buscando en internet, me decante por este brazo que rondaba los 140€ y en las especificaciones indicaba una capacidad de carga de 300g. El robot fue comprado por la página de internet de Aliexpress.

#### 2.2.7.2. Montaje del brazo robótico.

El brazo robótico llego desmontado y sin ningún manual de instrucciones (Ilustración 13). Este estaba separado por las piezas de aluminio, los tornillos, tuercas y arandelas, los servomotores y el sistema de vacío, con todos sus elementos. Para la instalación del robot me ayude de la página: <https://www.makerbuying.com/17675>. En esta página web se muestran fotos con las piezas a utilizar por cada eslabón y como enlazarlos entre sí. La instalación del robot fue simple y sencilla.



Ilustración 13: Robot por piezas.

### 2.2.7.3. Servomotor

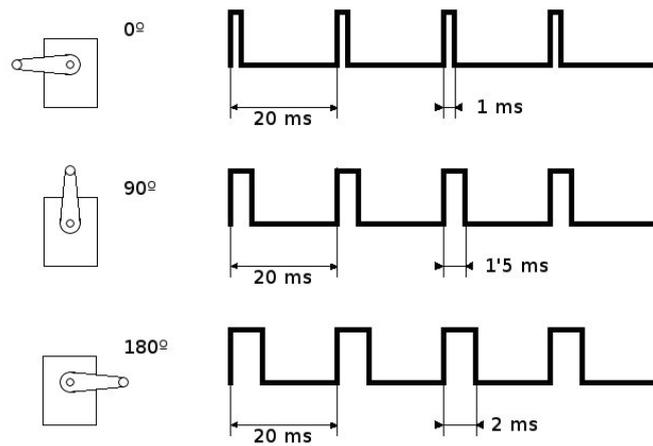
Un servomotor; también llamado servo, es un actuador eléctrico compuesto de un motor eléctrico, en la mayoría de caso de corriente continua, una reductora y la electrónica de control. La funcionalidad del servo es girar a la posición deseada dentro de su rango de operación, y permanecer estable en esa posición. El control de dichos servos es por posición. Es muy común encontrarse servomotores en la industria de la robótica, ya que, los servos tienen una precisión máxima. En la mayoría de servos no gira su eje 360°, lo más normal es que solo giren 180° hacia la izquierda o derecha, es decir ida y vuelta.

Los servomotores tienen una gran aplicación dentro de la industria. Como se ha explicado antes, dentro de la robótica, zoom de cámara de fotos, puertas automáticas, sistemas de producción, en el timón de los aviones, etc...

El servo contiene cuatro elementos clave para su funcionamiento:

- Un motor eléctrico: es el encargado de general el movimiento a través de su eje.
- Un sistema de regulación: Este sistema está compuesto de engranajes, actuando sobre el motor regulando la velocidad y el par. Mediante estos engranajes, que en la mayoría de casos son ruedas dentadas, se aumenta o disminuye la velocidad o el par.
- Sistema de control. El circuito electrónico encargado de controlar el movimiento del motor mediante pulsos eléctricos.
- Potenciómetro: se encuentra conectado al eje del motor e informa en todo momento de la posición del eje del motor. Este potenciómetro es una simple resistencia eléctrica variable.

El principal sistema de control de estos servos es por la modulación por anchura de pulsos (PWM). El PWM se utiliza para controlar la referencia de posición del servo. Dependiendo de la anchura del pulso, el servo se posicionara en un ángulo u o en otro. Como se muestra en la ilustración 14, cuando el PWM tiene una anchura de 1ms el servo motor se encuentra en la posición 0°. Mientras que cuando se cambie el pulso a 2ms de ancho, este servo se colocara en el ángulo 180°. Siendo 1ms la posición mínima y 2ms la máxima posición.



**Ilustración 14: Posicionamiento servo mediante PWM.**

Cabe destacar que es necesario mantener el PWM activo con la frecuencia deseada para que el servo mantenga la posición. De este modo, si se ejecutara alguna fuerza externa sobre el eje del motor, como pudiera ser el peso del brazo, este intentara resistir a esa fuerza manteniendo la posición deseada.

Este PWM, en el proyecto actual, será generado por la BeagleBone Black. Para el control del brazo robótico, al tener 7 servomotores independientes, es necesario 7 PWM, un PWM para cada servo. Ya que es necesario mover cada articulación independientemente de las otras seis. Es posible este control, gracias a que la BBB cuenta con 7 PWM de salida. Estas salidas digitales son necesarias declararlas como PWM y no como outputs normales.

### 2.2.7.4. Alimentación del sistema

Una de las partes más importantes del sistema es la alimentación del él. Ya que sin tensión o con fallos el sistema no funcionaria. Para este proyecto se ha optado por disponer de dos alimentaciones distintas. Una para alimentar la propia BeagleBone Black, y otra para la potencia de los servomotores y bombas de vacío.

Se ha decido esta separación de la alimentación ya que si se alimentan los servos por el pin de 5V de la BeagleBone, cuando hace un esfuerzo produce una sobrecarga en la salida, apagando la BeagleBone.

Por ello, se va a tener 5V de potencia, 5V y 3.3V de control, con los que se alimentaran los sensores necesarios. En la Ilustración siguiente (Ilustración 15) se muestra el esquema de potencia del sistema.

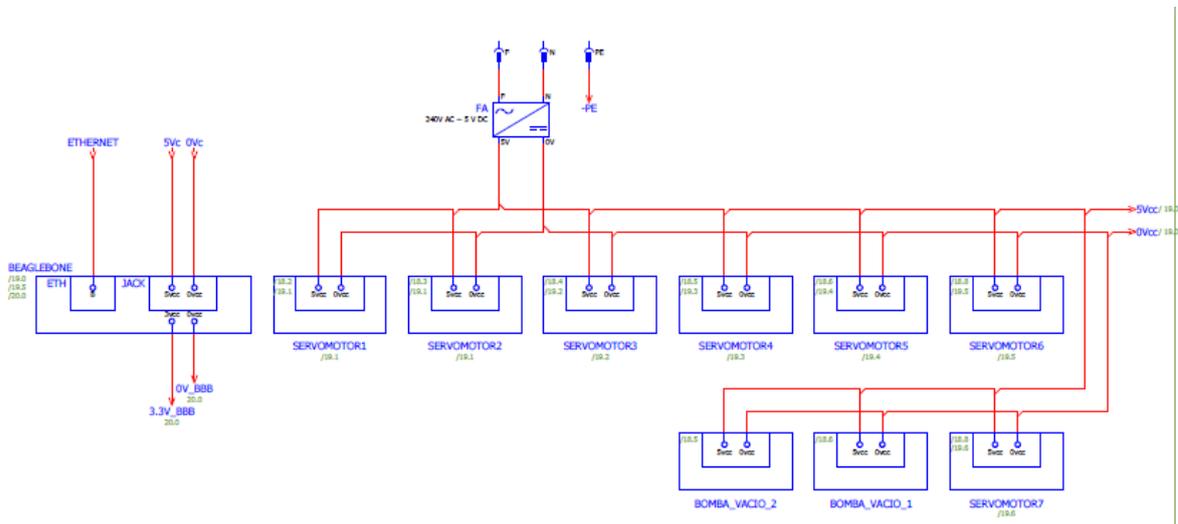


Ilustración 15: Esquema de potencia

#### 2.2.7.5. Sistema de vacío

Como ya se ha comentado antes, el brazo robótico cuenta con un sistema de vacío. El cual genera una presión en las ventosas de la herramienta, pudiendo agarrar objetos. Para poder realizar el vacío actúan dos motores eléctricos de corriente continua. Realizan la función de una bomba de vacío. Estos sistemas son muy comunes en el mundo de la industria, sobre todo en las tareas de transporte de granulados, como pueden ser el plástico, grano, etc... En nuestro proyecto es un sistema de vacío a escala pequeña, ya que no necesitamos un gran vacío para nuestra aplicación. En nuestro caso, contamos con dos bombas del tipo LY370DPM30 (Ilustración 16), con una tensión de alimentación de 3V a 6V y una corriente de 0.26A a 0.38A respectivamente.



Ilustración 16: Bomba de vacío.

Al alimentar la bomba de vacío, ya sea con 3V o 6V, estas mediante el giro de su eje, succionara aire por su entras y lo expulsara por la salida. Generando un vacío en la entrada, donde se encontrara el sistema de vacío. La entrada de la bomba se conecta a las 4 ventosas con un tubo de plástico. Las cuatro ventosas se encuentran enseriadas.

#### 2.2.7.6. Sensor de nivel

Otra de las funciones del sistema, es la monitorización del sistema. En este caso se quiere conocer el estado del tanque de agua de la cafetera. La mejor opción será un sensor que pudiera medir la cantidad de agua que contiene el tanque. Un ejemplo es un aforador, dispositivo que normalmente mide el tanque de gasolina de los vehículos. Este dispositivo normalmente es una entrada analógica. En el mercado la mayoría de estos elementos son bastante caros, con lo que son descartados para este proyecto.

En nuestro caso, necesitamos un elemento que nos indique cuando queda poca agua, así poder avisar con una alarma y que los usuarios de la cafetera repongan el agua necesaria. Para esta funcionalidad existe una gran variedad de sensores. Al estar trabajando con una placa de bajo coste y tener una alimentación de 5V, necesitamos buscar elementos que puedan trabajar o sean alimentados a esa tensión.

El elemento elegido es un sensor de nivel de líquido, el cual, mediante una boya cuando llega a un extremo u al otro cierra un contacto indicando que el nivel del líquido se encuentra lleno o vacío. Este sensor se encuentra con distintas medidas para poder adaptarlo a nuestro tanque de agua (Ilustración 17).



Ilustración 17: Sensor de nivel

### 2.2.7.7. Detector de capsulas

Otro de los elementos a monitorizar del sistema, es el alimentador de capsulas de café. También es un aspecto importante el avisar cuando es necesario recargar las capsulas de café o transmitir un fallo cuando ya no queda ninguna.

Existen multitud de elementos para detectar este estado. Un ejemplo es un sensor capacitivo. Estos sensores se usan para detectar plásticos (nuestro caso, ya que las capsulas de café son de este material), así como otros tipos de aislantes [12].

El problema de estos elementos es que necesitan una alimentación externa, normalmente, de 24V, y en nuestro sistema únicamente contamos con una alimentación de 5V como máxima. En este caso necesitaríamos otra alimentación de 24V. Con lo cual, este sensor quedo descartado.

La opción más interesante es un sensor de infrarrojos, también conocido como detector de obstáculos IR (Infrarrojos). Este sensor detecta la presencia de un objeto mediante la reflexión que produce en la luz. El uso de la luz infrarroja es simplemente para que no sea visible por el ser humano. Este sensor está compuesto de un luz IR y un fotodiodo, el cual recibe la luz reflejado por el obstáculo. Estos sensores suelen devolver un valor digital, este valor cambia cuando se supera cierto umbral (Ilustración 18) [13].

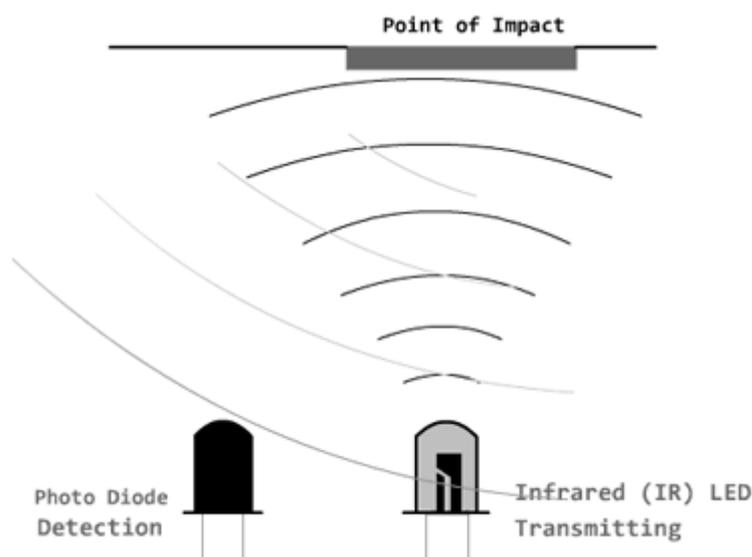


Ilustración 18: Detector de obstáculos IR

Estos sensores suelen tener un precio reducido, y su alimentación suele ser de 5V. Con lo cual es la opción a elegir para esta función

### 2.2.8. Ejemplos de brazos robóticos en la domótica

Como se ha explicado en los apartados anteriores, existen varias categorías de robots. En concreto se va a tratar de brazos robóticos, pero aún más específico, en brazos robóticos en la domótica. En este ámbito se pueden encontrar una variedad de proyectos relacionados con estos tipos de robots.

#### 2.2.8.1. Robot montado en el techo

Toyota ha presentado un proyecto de robótica para el hogar con una idea diferente a las demás. Colgar los robots del techo de la casa para poder desplazarse con mayor facilidad. Toyota lo que propone es que su robot se mueva a lo largo de la casa por railes en el techo. De este modo, el robot no molesta ni obstaculiza a los habitantes del hogar, ni tener que hacerle espacio en el suelo para que pueda pasar. Esta idea fue recogida de Japón, ya que, se encuentran con muchos hogares con espacio reducido donde se busca aprovechar al máximo el espacio. El problema, que esta idea requiere de una compleja y espaciosa instalación de la casa [14].

Se han realizado varias pruebas en una casa de demostración, creada específicamente para este robot. Toyota consiguió que el robot se moviese por la cocina limpiara la mesa y colocara los objetos en su sitio sin problemas. Se adjunta un enlace donde se puede ver el robot comentado junto a varios proyectos de brazos robóticos de Toyota realizando tareas del hogar:

<https://www.youtube.com/watch?v=u6bulfbFj04>

De momento, solo son prototipos de Toyota, aún falta demostrar que son capaces de trabajar en casas reales y no en espacios diseñados para su demostración. Veremos si en un tiempo este proyecto sigue adelante, y el robot colgante se convierte en una parte mas del hogar.

### 2.2.8.2. Café X

Cuatrocientos años más tarde de la primera cafetería, se estrenó Café X, la primera cafetería robótica en Estados Unidos. Este nuevo local, está revolucionando el modelo establecido de cafetería tradicional. Se ha inaugurado en San Francisco y es la primera cafetería que no emplea a baristas a para servir los cafés, sino a robots [15].

Henry Hu fue el diseñador de este proyecto con la inversión de Thiel Fellowship, pudiendo así abrir Café X. Hu adquirió un brazo robótico industrial de 6 ejes de libertad de la marca Mitsubishi (Ilustración 19).



Ilustración 19: Café X

Este local cuenta con un sistema automatizado, el cual controla las máquinas de café y las coordina con los movimientos del robot. Es posible la personalización del café, pudiendo elegir la cantidad de leche, tipo de café, sabores adicionales y otra multitud de elecciones. A demás, la elección de toda esta personalización como del pago, es mediante una aplicación móvil compatible con iOS y Android. El robot es capaz de servir entre 100 y 120 tazas por hora, unos números bastante altos a comparación de otras cafeterías tradicionales. A parte de que este robot trabaja las 24 horas del año [16].

Se comparte un enlace donde se puede ver el funcionamiento de esta cafetería: <https://www.youtube.com/watch?v=t42XCACHUN8>

### 2.3. Objetivos

Como ya se ha comentado anteriormente, el objetivo principal del proyecto es realizar la tarea cotidiana de poner cafés. Que nuestro brazo robótico, prepare la cafetera, coja la capsula de café, la introduzca en la cafetera y que encienda la cafetera. Para conseguir este resultado será necesario programar, configurar, diseñar o implementar varios elementos y tecnologías.

En cuanto a la BeagleBone Black, es necesaria su configuración. Necesitamos instalarle un sistema operativo acorde con la versión de ros a utilizar. Configurar las salidas digitales a utilizar como PWM. Y por último, se va a implementar una VPN, mediante el software OPENVPN, dentro de la BBB, en este caso la BBB será configurada como cliente. Esta implementación se realiza para el posible control, mantenimiento o programación de la BeagleBone desde fuera de la red de la oficina.

Respecto al robot, como se ha mencionado anteriormente, se ha realizado la instalación del robot pieza a pieza. La instalación de los servomotores en el robot y su calibración. Se ha diseñado una misma base para poder anclar el robot y la cafetera consiguiendo estabilidad al robot.

Para la alimentación de los servos, bombas de vacío y I/O de la BBB se ha diseñado un circuito eléctrico. Este circuito eléctrico se instalara encima de la BBB como una capa de la propia BeagleBone. Para ello también se han dibujado unos esquemas eléctricos mediante el software Eplan.

Por último, está la interfaz gráfica del sistema, para el control y monitorización. En este proyecto, se utilizara otra BeagleBone Black con el software de automatización Codesys instalado en ella. Ya que gracias a este software es posible crear un visualizador gráfico y realizar las comunicaciones con el cerebro del sistema, la BBB principal. Esta comunicación es mediante modbus TCP/IP. Siendo la BBB principal quien pida o escriba la información en la secundaria, y esta última, la represente en la interfaz gráfica.

## 2.4. Metodología

La BeagleBone Black, es el cerebro del sistema, la cual le da sentido común a todo el sistema. Esta recibe la información del entorno la gestiona y la convierte acciones. El sistema tiene una arquitectura centralizada, ya que es el maestro y los demás elementos envían su información a este. El cerebro la gestiona teniendo en cuenta el estado del entorno y genera las órdenes oportunas para los actuadores y su interfaz. (Ilustración 20).

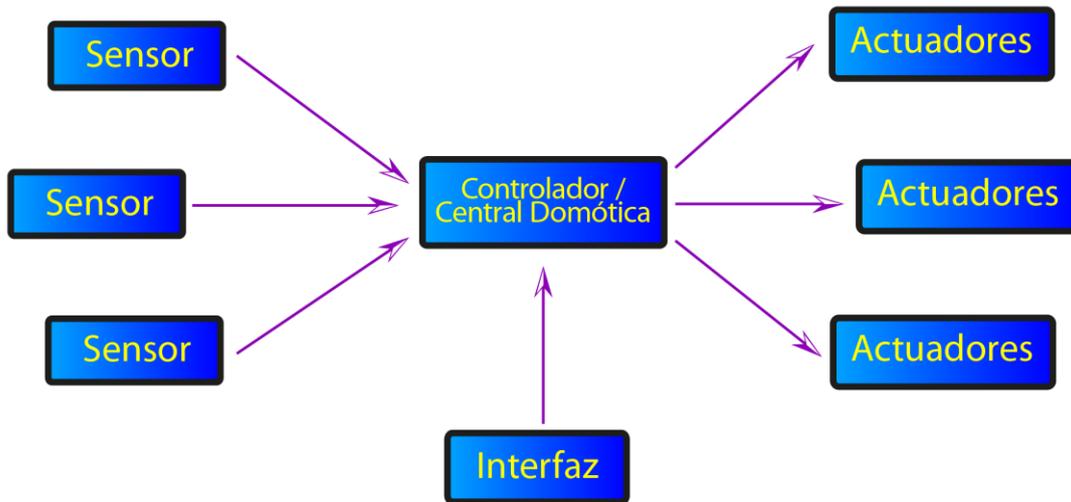
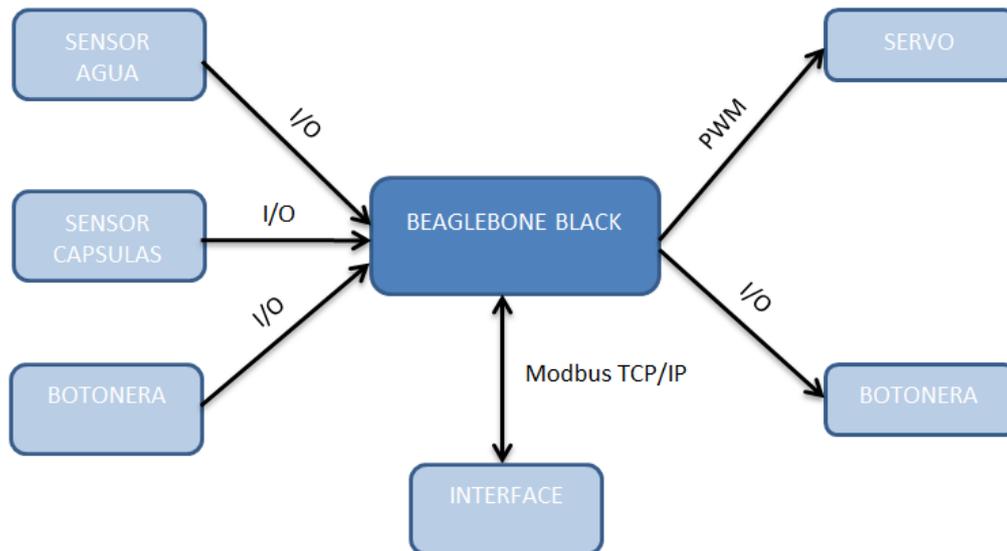


Ilustración 20: Arquitectura Centralizada.

Nuestro sistema solo tiene un controlador centralizador, donde reside la inteligencia del sistema. El sistema dispone de una botonera para el control del robot en local. El robot conectado a la BeagleBone mediante PWM de los servomotores. Sensores para conocer el estado del almacenaje de las capsulas de café y del tanque de agua de la cafetera (Ilustración 21).



**Ilustración 21: Arquitectura desarrollada en el proyecto.**

#### 2.4.1. Configuración BeagleBone Black

Para la configuración de la BBB, lo primero es la instalación del sistema operativo adecuado para la versión de ROS. En nuestro caso se instalara el sistema operativo Ubuntu 18.04.4 LTS. Primero es necesario descargar la imagen del SO deseado de internet, grabarla mediante el programa balenaEtcher el SO en una microSD Introducir la microSD en la BBB y se instalara y guardara el sistema operativo en la BeagleBone. Se ha instalado un SO sin escritorio, es decir, la Beaglebone no tiene un escritorio en sí, todo se hace desde un terminal y por comandos. Para poder realizar las modificaciones es necesario acceder a ella, esto se realizara por el protocolo Secure Shell (SSH). Se configurara una IP estática para poder conectarnos siempre con la misma IP.

### 2.4.2. Protocolo SSH

El protocolo SSH es un protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor y que permite a los usuarios conectarse a un host remotamente.

Al no tener monitor, ratón y teclado es necesario el programa Putty, para poder conectarse a la BBB con este protocolo. Esto permite configurar el hardware de la BBB, poder configurar el I/O, insta y configurar la VPN e instalar y programar ROS.

### 2.4.3. Instalación ROS

En este sistema se ha instalado ROS Melodic Morenia 1.14.10. Se ha seguido la recomendación por parte de ROS para instalar esta versión de ROS en el SO de Ubuntu 18.04. Para la instalación de ROS, mediante Putty y por comandos, se han seguidos los pasos a instalar de la página oficial de ROS: <http://wiki.ros.org/melodic/Installation>.

### 2.4.4. Open VPN

OpenVPN es un software, de libre uso, que permite la conexión segura de recursos de red locales, desde ubicaciones remotas haciendo uso de las conexiones públicas de internet. OpenVPN utiliza la tecnología VPN (Virtual Private Network), esta tecnología es usada para conectar una o más computadoras a una red privada utilizando internet.

La BeagleBone funciona como OpenVPN cliente, es decir se conecta a un servidor VPN. Este servidor está contratado por la empresa y está fuera de la oficina, y a este servidor es posible conectarse con un dispositivo fuera de la oficina con acceso a internet y los certificados del servidor.

De este modo, la BeagleBone está conectada a la red de la oficina y al servidor VPN permitiendo que un dispositivo, fuera de la oficina con acceso a internet, se puede conectar a la BeagleBone. EL servidor VPN crea un túnel seguro entre la BeagleBone y el usuario. La utilización de este servidor VPN fuera de la oficina es por el hecho de que cuando algún usuario quiera conectarse a la interfaz del control de la oficina, únicamente pueda acceder a la interfaz y así denegando el acceso a la red privada de la oficina (Ilustración

22).

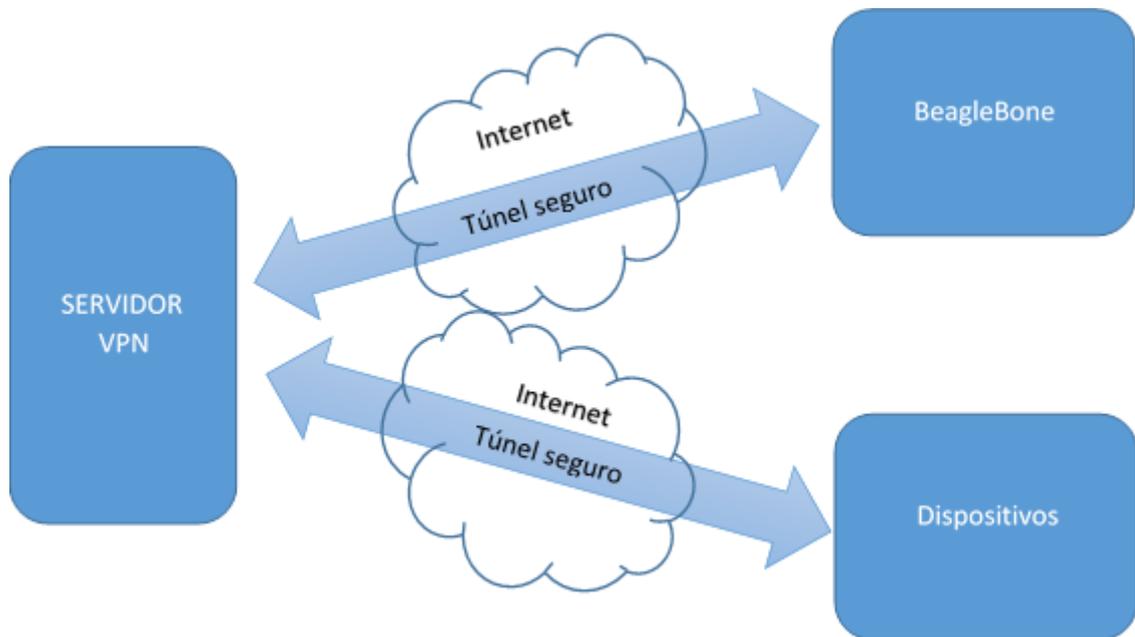


Ilustración 22: Conexión remota del usuario con la BeagleBone.

La instalación de OpenVPN en la BeagleBone se realiza con comandos en la consola del terminal. Se utiliza la conexión SSH para poder ejecutar comandos en la consola del terminal. A la BeagleBone hay que darle los certificados del servidor copiando unos archivos en su repositorio.

#### 2.4.5. Botonera

El control sobre el sistema se puede realizar remotamente, es decir, mediante alguna interfaz gráfica o por comandos desde un PC con ROS instalado y acceso a la red del sistema. O por otro lado el control desde local, con una botonera al lado del robot, con selectores, pulsador, seta de emergencia y leds (Ilustración 23).

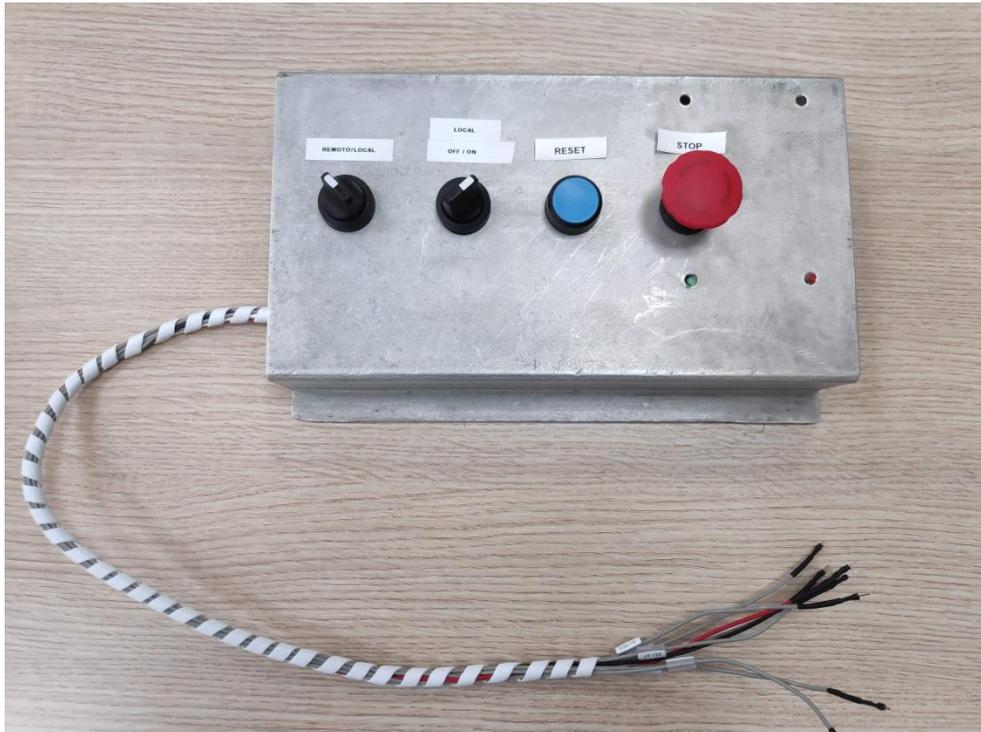


Ilustración 23: Botonera

Esta botonera va a contar con un selector, mediante el cual, se va a poder elegir el modo de control del sistema. Este selector cuenta con dos posiciones. Remoto, como se ha comentado antes, para controlar el robot por comandos o interfaz gráfica. Local, con este modo, el sistema se controla por el otro selector. El segundo selector hace la función de START y STOP del sistema en modo local, con lo cual, si el sistema se encuentra en modo local, la única forma de arrancar el sistema sería desde este pulsador. Pasar el selector de STOP a STAR, el robot empezara su función. Y si el selector pasa de STAR a STOP, el robot parara.

El pulsador que cuenta la botonera se utiliza para el RESET del sistema. Este botón reseteará las alarmas y fallos que puedan salir durante el funcionamiento del robot. Estos fallos y alarmas pueden ser falta de agua, falta de capsulas de café o algún problema con el robot. Al salir algún fallo, será necesario resetear el sistema para poder arrancarlo de nuevo.

Los leds marcarán el estado del sistema. Son dos led, rojo y verde. El led rojo permanecerá fijo cuando el sistema tenga un fallo, parpadeará cuando haya alguna alarma. Y el led verde estará encendido cuando el sistema esté en marcha.

Por último, la seta de emergencia, el elemento de seguridad del sistema. Este elemento al ser pulsado parará el sistema de inmediato. Esta seta está diseñada para la seguridad de las personas a utilizar el robot y que puedan pulsarla antes de que se produzca ningún accidente.

Esta botonera ha sido diseñada con elementos disponibles en la oficina. Ya sean los selectores como el pulsador, la seta y los leds. Para facilitar la instalación eléctrica y el mantenimiento del sistema se ha dibujado los esquemas eléctricos de esta botonera (ilustración 24).

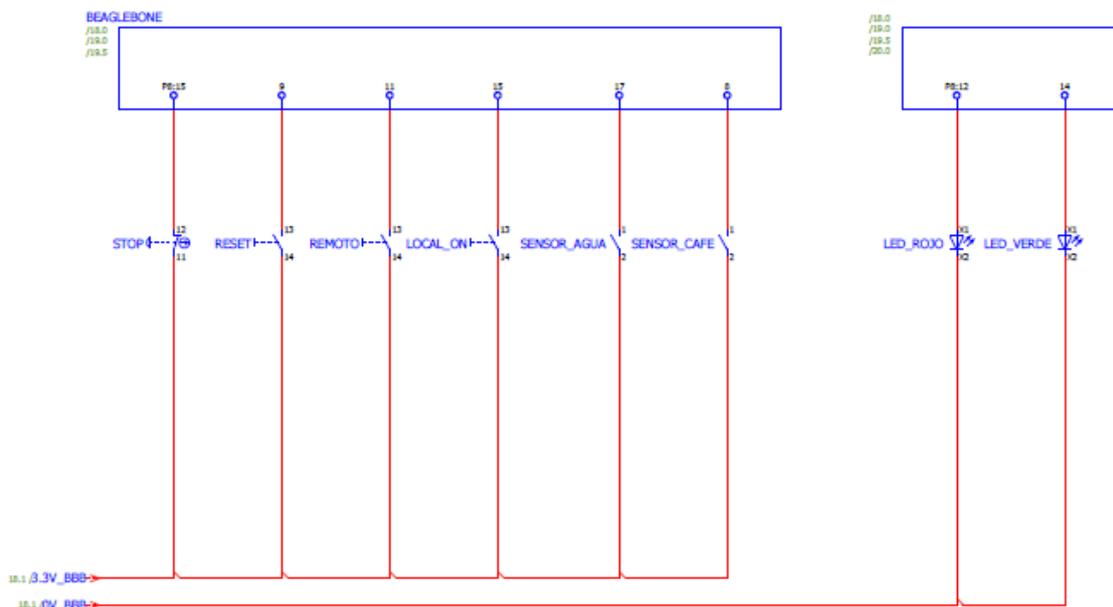


Ilustración 24: Esquema eléctrico de la botonera

Los elementos están alimentados a 3.3V de la propia BeagleBone. Esta placa ha sido diseñada y creada exclusivamente para el proyecto, en cuanto al tema eléctrico ha sido necesario el cableado de la manguera, el taladrado de la placa para los elementos, la conexión eléctrica y el soldado de los pines finales para la conexión a los pines de las BeagleBone Black (Ilustración 25).

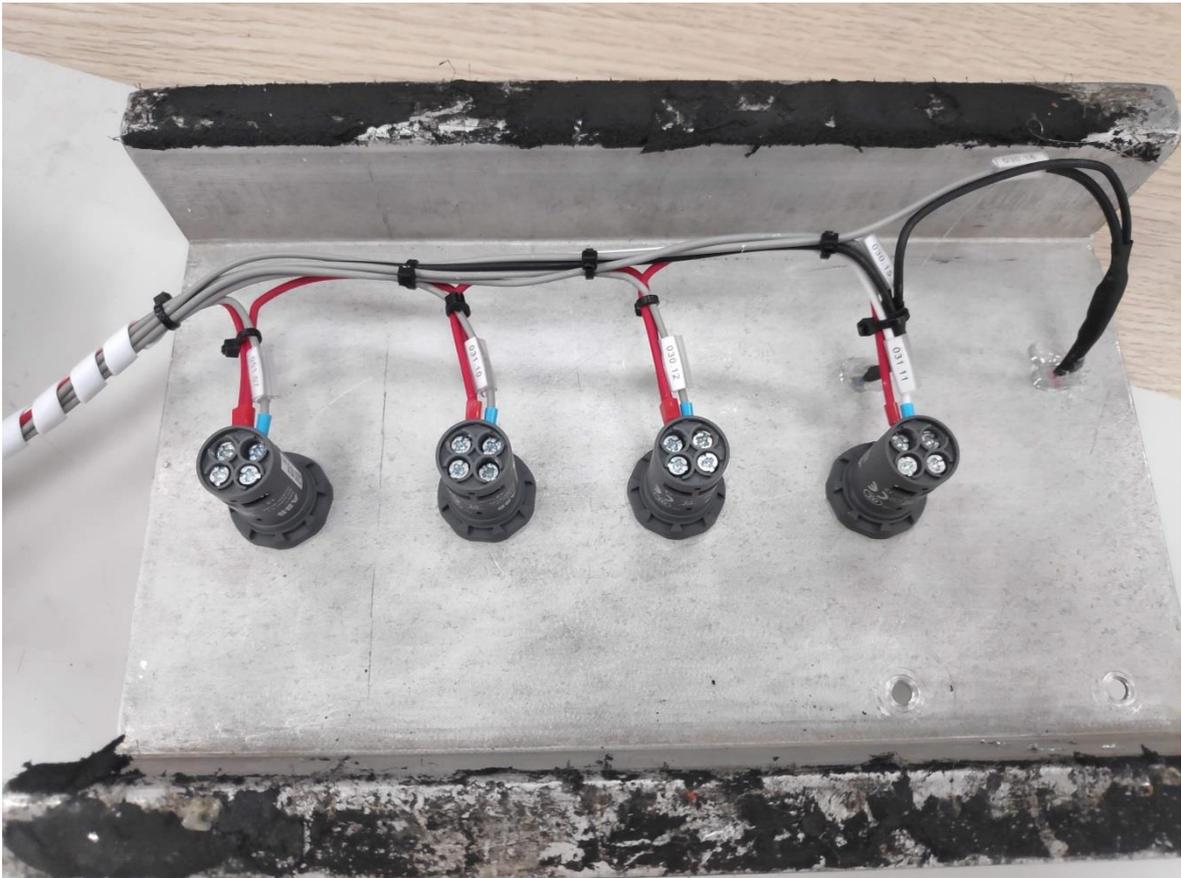


Ilustración 25: Botonera por detras

#### 2.4.6. CODESYS para BeagleBone Black

En este caso, es necesario instalar el sistema operativo Debían 9.5. Hay que actualizar la BBB para que la misma BBB pueda ejecutar el archivo generado por CODESYS con el código de programación. La actualización se realiza en el software de CODESYS, indicando la dirección IP, el usuario y contraseña de la BeagleBone para que pueda conectarse a ella y ejecute la actualización. Esto solo es posible con algunos sistemas operativos, por eso la utilización de Debian 9.5 en este proyecto.

En el software de CODESYS para poder utilizar la BeagleBone es necesario instalar un paquete con licencia, el cual admite las funciones de:

- CODESYS EtherCAT maestro
- CODESYS Modbus TCP maestro/esclavo
- CODESYS Modbus RTU maestro/esclavo
- CODESYS WebVisu
- CODESYS SoftMotion CNC+Robotics
- CODESYS OPC UA Server
- CANopen

En el proyecto no se utilizarán todas las funciones disponibles, ya que no son necesarias para el objetivo del proyecto. Únicamente se utilizaran las funciones de Modbus TCP, en función de esclavo y el WebVisu.

## 2.5. Cuerpo del trabajo

En el apartado anterior se han descrito los elementos del sistema y la configuración de los diferentes dispositivos para la realización del proyecto. En este apartado se verán las soluciones de programación sobre el robot. En el desarrollo del capítulo actual es importante resaltar las características del sistema, además de los modos de funcionamiento.

La programación del proyecto se desarrolla en la Beaglebone Black. Esta programación se ha creado en el lenguaje Python. La programación del sistema se puede dividir en cuatro partes, siendo estas partes, la programación de las entradas y salidas digitales, el control del sistema, la programación de los actuadores del robot y la última, la comunicación con la interfaz gráfica.

### 2.5.1. Nodos del sistema

Para la unión de las tres partes del sistema se utilizara el software de ROS, el cual mediante mensajes y servicios se conectaran las diferentes secciones. Para ello se ha creado un nodo principal, que será el nodo maestro. En este nodo llegara la información del exterior, se administrara y se enviara a otro nodo para que se ejecute. Otro nodo creado es el que administrara la botonera, tanto los selectores, pulsador, seta como los leds. El tercer nodo será quien administre los PWM de los servomotores del robot. Y el cuarto nodo es quien realiza la comunicación Modbus TCP/IP con el sistema encargado de la interfaz gráfica (Ilustración 26).

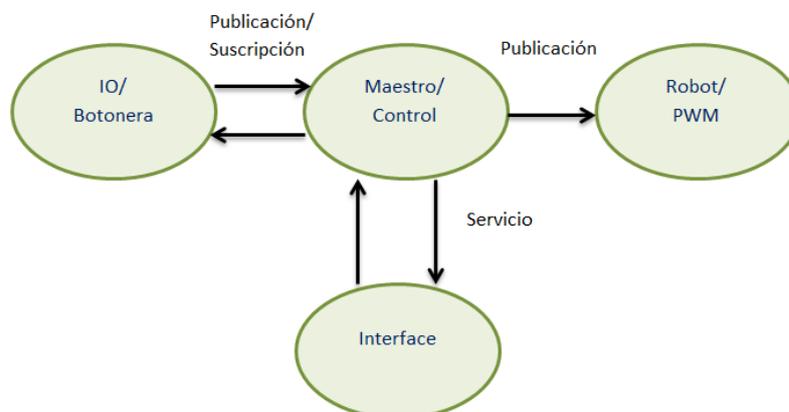


Ilustración 26: Nodos del sistema.

### 2.5.1.1. Nodo Maestro.

El nodo maestro es el cerebro del sistema. Es el quien recibe la información y quien la administra. Este nodo está suscrito a varios tópicos para recibir la información externa y publica otros tópicos para realizar las acciones necesarias. Dentro de este nodo esta toda la programación de control del sistema. Este se encarga de definir la posición del robot en cada momento. Es decir, enviar por un tópico la posición angular de cada una de las articulaciones del robot. A demás controla el sistema de vacío, pero este control es muy simple, únicamente es activar la electroválvula cuando no se necesite vacío en el sistema.

Desde este nodo, además del control del sistema, se gestionan las alarmas. Estas alarmas pueden venir del dispensador de cápsulas o del tanque de agua. Estas alarmas necesitan ser resetadas para borrarlas. Únicamente serán borradas cuando no siga activo el evento que la provoca y se pulsa el botón de reset.

También envía y recibe la información por un servicio al nodo de la interfaz gráfica.

### 2.5.1.2. Nodo Botonera.

Este nodo administra las entradas y salidas digitales de la botonera. La cual es utilizada para la declaración del funcionamiento del robot. Como ya se ha comentado anteriormente, esta botonera cuenta con dos selectores, un pulsado, una seta de emergencia y dos leds. El estado de la botonera esta subdividido en dos mensajes. Las entradas digitales y las salidas digitales.

En un mensaje se encuentran el estado de los seccionadores, pulsador, reset y stop. Estos estados son un bit cada uno y en otro mensaje el estado de los dos leds.

```
bool remote
bool local_on
bool reset
bool stop
```

Ilustración 27: Msg Botonera.

Este nodo publica un topic con el estado de las entradas digitales de la botonera. Y está suscrito a otro topic, recibiendo el estado de los led. El nodo lee las entradas digitales e inmediatamente las guarda en 4 variables booleanas. Estas variables se envían mediante el topic y el msg de la botonera al nodo maestro. A parte, el nodo maestro envía la acción

sobre los leds para indicar el estado del sistema.

### 2.5.1.3. Nodo Robot.

Este nodo actúa sobre los servomotores del robot. Está suscrito al topic de las articulaciones del robot, es decir, recibe el ángulo de cada una de las articulaciones y lo transforma en señal PWM. Para ello es necesario inicializar los PWM, declarar el tiempo de trabajo mínimo y máximo del PWM para cada servo respecto a sus características. En nuestro caso es el mismo para todos. Luego hay que convertir los ángulos en tiempo de trabajo del PWM. Todos estos tiempos que se hablan son en ms.

El topic está compuesto por 7 enteros del tipo int8 (Ilustración 28). Cada entero para el ángulo de un servo. Hay que tener en cuenta, que el robot tiene dos servos para una misma articulación. Y estos servos giran al contrario, es decir, se encuentran invertidos. Con lo cual, el rango de trabajo de un ángulo será de 0° a 180° y el otro de 180° a 0°. Mientras un servo se encuentra a 180° el otro está en 0°. Esta inversión se realiza en este nodo. El nodo recibe el ángulo que necesita la articulación y transforma el segundo ángulo.

```
#mensaje para cada angulo de las articulaciones
int16 art1
int16 art2
int16 art3
int16 art4
int16 art5
int16 art6
```

Ilustración 28: Msg Articulaciones

#### 2.5.1.4. Nodo Interfaz.

Este es el último nodo del sistema. Es el encargado de pasarle la información al sistema de la interfaz gráfica. Esta comunicación, como ya se ha comentado por encima anteriormente, se realiza por Modbus TCP. En este sistema la BBB principal será el esclavo y la segunda el maestro, Cliente/servidor. En este nodo por este protocolo se leerá el estado de una variable, y por otro lado se escribirá otras 7 variables. Para ello, es necesario ubicar estas variables en direcciones de memoria de la BeagleBone (Tabla 2). Desde la interfaz se leerán los estados y se escribirán los comandos. Esto se realiza en el software de Codesys. Mientras que en la BeagleBone principal habrá que configurar el protocolo Modbus como servidor en Python.

<b>Variable</b>	<b>Dirección</b>	<b>Tamaño</b>	<b>Bit</b>
<b>Remoto</b>	159	Bit	0
<b>Local_On</b>	159	Bit	1
<b>Reset</b>	159	Bit	2
<b>STOP</b>	159	Bit	3
<b>Led_Verde</b>	159	Bit	4
<b>Led_Rojo</b>	159	Bit	5
<b>Sensor_Agua</b>	159	Bit	6
<b>Sensor_Capsulas</b>	159	Bit	7
<b>Paso</b>	160	Word	
<b>Articulación 1</b>	161	Word	
<b>Articulación 2</b>	162	Word	
<b>Articulación 3</b>	163	Word	

<b>Articulación 4</b>	164	Word	
<b>Articulación 5</b>	165	Word	
<b>Articulación 6</b>	166	Word	
<b>Start_remoto</b>	83	Bit	0

**Tabla 2: Tabla Modbus**

Para la parametrización de la función de Modbus, se define la IP de esclavo, su ID si fuera necesario, en este caso no lo es, ya que esto solo se utilizan cuando se realiza una conversión de Modbus TCP/IP a Modbus RTU, el timeout, la función, en nuestro caso, utilizaremos la función 1 para leer n booleanos y la función 5 para escribir un bool y la dirección inicial de las variables.

### 2.5.2. Modos de funcionamiento

El sistema tiene dos modos distintos de funcionamiento, modo local y modo remoto. El movimiento del robot es lo mismo en los dos casos, pero la forma de darle marcha al sistema no.

En el caso del modo local, el robot hará su función cuando el primer selector se encuentre en local y el segundo pase de OFF a ON. En ese instante, y si el sistema no tiene ningún fallo, el robot realizará sus movimientos. Cuando el selector se encuentra en local, no es posible mover el robot desde la interfaz, y al contrario cuando está en remoto.

Al arrancar el sistema el robot ira a una posición inicial. Ya en esta posición realizara los siguientes movimientos para realizar su tarea, volviendo a su posición inicial al terminar el ciclo.

Al contar con un servomotores de 180°, el robot tiene un área de trabajo. También se cuenta con que no todos los servomotores pueden girar completamente su rango de trabajo, ya sea por obstaculización de otros eslabones o de la cafetera (Tabla 3).

Articulación	Rango de giro (°)
1	0-180
2	90-180
3	120-180
4	0-180
5	60-180
6	0-180

Tabla 3: Rango de giro de las articulaciones

Teniendo en cuenta estos rangos de posiciones de las articulaciones, se van a declarar unos pasos a seguir que debe realizar el robot para completar su tarea. Al arrancar el sistema el robot se debe colocar en la posición inicial. El siguiente paso será levantar la tapa para poder introducir la capsula.

Con la tapa levantada y si el dispensador de cápsulas de café tiene cápsula, el robot debe coger la cápsula, para ello el robot se colocara delante del dispensador y a la altura de la cápsula. Esta será la segunda posición. Cuando se encuentre en esta posición, se activará la electroválvula para crear vacío en el sistema. Con el vacío activado y en la posición dos, se muevo a la tercera posición, esta posición es junto donde se coge la cápsula. Con la cápsula cogida el robot debe volver a la segunda posición.

Ahora el robot se ubicara justo encima de la cafetera, alineando la cápsula cogida con el hueco para ella. En esta posición, se desactiva la electroválvula abriéndose el circuito de vacío y soltando la cápsula. Con la capsula ya metida en la cafetera, se debe bajar la tapa. Este paso es uno de los más críticos, ya que es cuando debe realizar la mayor fuerza. Para ello, el robot apoyara el antebrazo en tapa y gracias a que la segunda articulación tiene dos servomotores, debe poder bajar la tapa.

Con la capsula dentro y la tapa bajado, queda activar la cafetera. Para ello, se colocara encima del botón y será apretado. El último paso del robot, será colocarse otra vez en la posición de inicio. Terminando así su ciclo y completando su tarea (Ilustración 29).

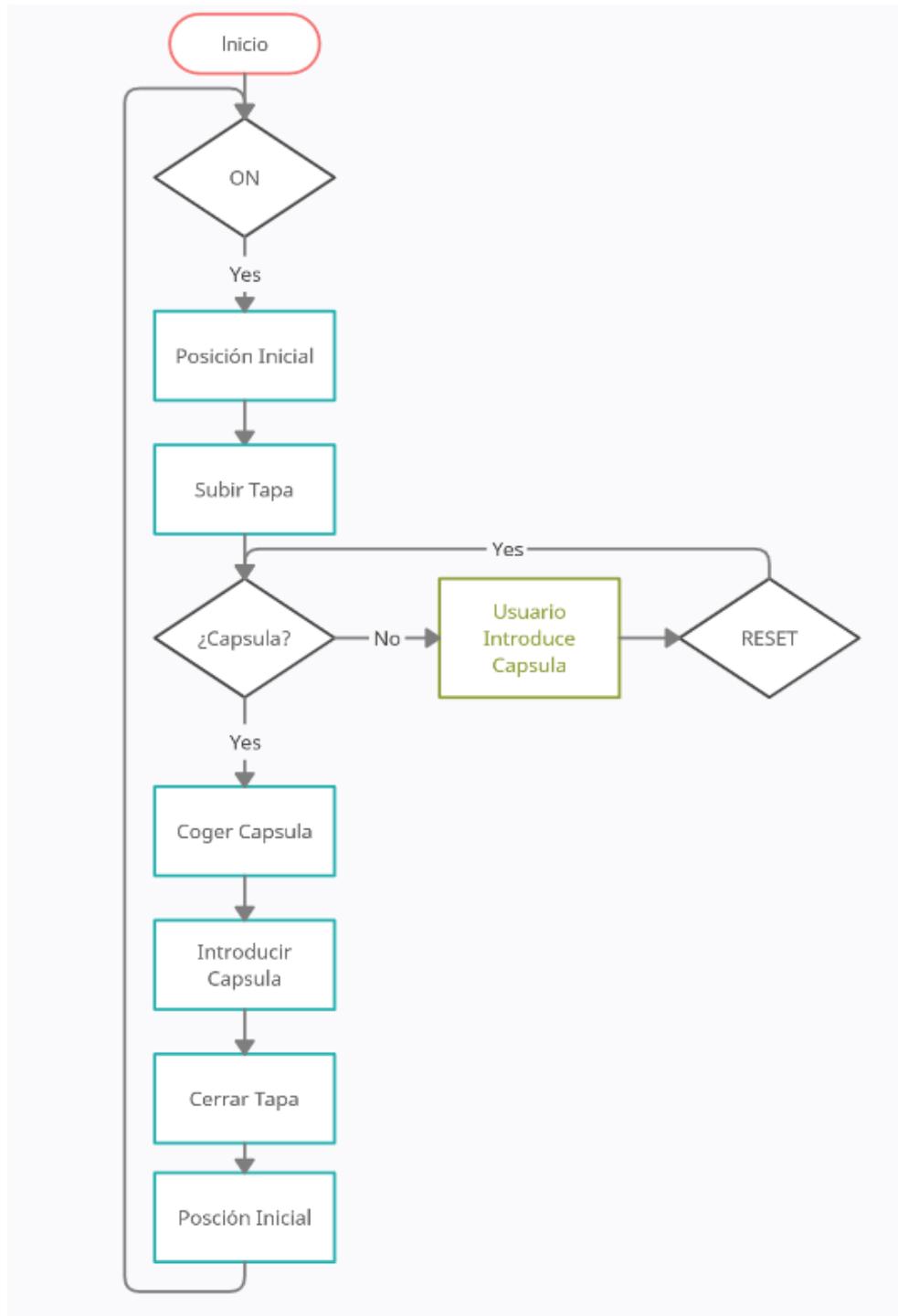


Ilustración 29: Diagrama de flujo con los pasos a seguir por el robot

### 2.5.3. Interfaz gráfica.

Para la interfaz gráfica, como ya se ha hablado anteriormente, se ha desarrollado mediante el software Codesys V3. En este caso, este control se ha implementado dentro de, la ya existente, interfaz gráfica de la oficina. En este control y monitorización de la oficina, en la cual, se pueden controlar tanto la iluminación como la climatización y la energía consumida en esta. Gracias a esta nueva pantalla, el control y monitorización del sistema robótico se va a poder realizar desde un dispositivo como puede ser un PC o un teléfono móvil.

Gracias al software de Codesys la visualización del sistema, es posible acceder a ella desde un navegador. Para ello hay que configurarlo desde el propio software de Codesys, y una vez cargado el programa en la segunda BeagleBone, introducir la IP y el puerto 9090 de la propia tarjeta.

#### 2.5.3.1. Pantalla principal

La pantalla principal es la primera pantalla que aparece al acceder a la dirección de la visualización web. Para poder acceder a esta web es necesario estar conectado a la misma red de la BeagleBone Black, ya que es la BBB quien genera esta dirección. Es preciso conectarse a la red de la oficina para poder estar conectado con la BBB.

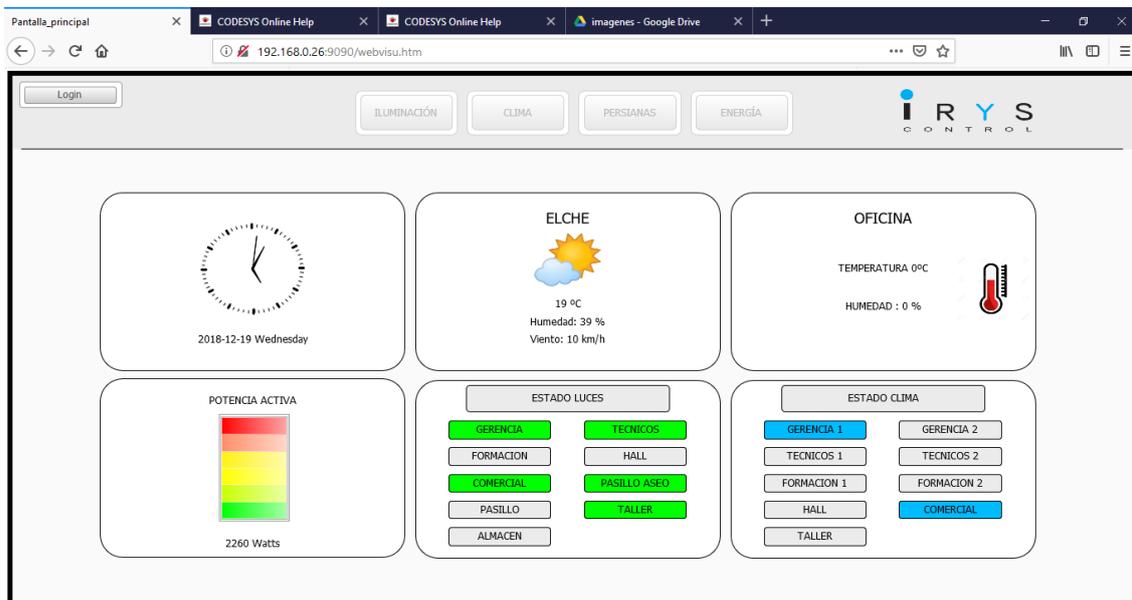
Hay dos opciones para conectarse a la red, mediante WiFi o por VPN. Esta segunda opción es necesaria cuando el usuario está fuera del entorno de cobertura de la red de la oficina. Ya que la oficina cuenta con su propio servidor VPN, aparte del servidor VPN que está conectado la BeagleBone. Este último servidor, es un servidor contratado por la empresa.

La pantalla principal (Ilustración 30) muestra la información principal de la oficina. La parte superior de la visualización está dividida en tres secciones. La esquina superior derecha muestra el logo de la empresa Irys Control (Ilustración 30i). La parte central superior son los botones de cambio de visualización (Ilustración 30h), éstos botones son iluminación, que al presionar sobre él, cambiaría de visualización a la pantalla de iluminación. Los demás botones son climatización, persianas y energía. Son los 4 apartados principales del proyecto.

En el ángulo superior izquierdo se encuentra el botón de inicio de sesión (Ilustración 29g),

lo cual es necesario para poder acceder a las posteriores visualizaciones. Existen distintos tipos de usuario. Diseñado para que no todos los usuarios puedan realizar todas las funciones del sistema o cambiar parámetros. También está el botón para cerrar sesión, el nombre del usuario y el botón para administrar el usuario.

Las 6 divisiones en el centro de la pantalla exponen diferentes informaciones. La ventana arriba a la izquierda muestra la fecha y hora del día actual (Ilustración 30a). La siguiente ventana muestra la información sobre el clima de Elche (Ilustración 30b). La última ventana de la fila superior expone la temperatura y humedad de la oficina (Ilustración 30c), mediante un mismo sensor de temperatura y humedad. La ventana inferior derecha (Ilustración 30f), define las zonas dónde hay máquina interior y en azul aquellas zonas que la unidad interior está en marcha. La siguiente información es similar pero con la iluminación (Ilustración 30e), siendo las zonas en verde las salas iluminadas. La última ventana expone la potencia activa consumida por la oficina en tiempo real (Ilustración 30d).



**Ilustración 30: Pantalla principal de la página web. A) Hora, B) Tiempo, C) Temperatura interior, D) Potencia; E) Iluminación, F) Climatización, G) Inicio sesión, H) Menú, I) Logo Empresa**

Para la realización de la nueva pestaña del Robot se ha utilizado la pestaña de persianas, ya que actualmente esta función del sistema está inoperativa.

### 2.5.3.2. Pantalla Robot

Esta pestaña sigue la filosofía del resto de interfaz. Es decir, el control del usuarios arriba a la izquierda, en el control el navegador de la visualización y a la derecha el logo y la pantalla principal.

A la izquierda se encuentra el estado del sistema, donde podemos ver si está el robot en movimiento o parado, si tiene alguna alarma o fallo y en que paso de su tarea esta. A demás de poder ponerlo en marcha y pararlo cuando se encuentra en remoto.

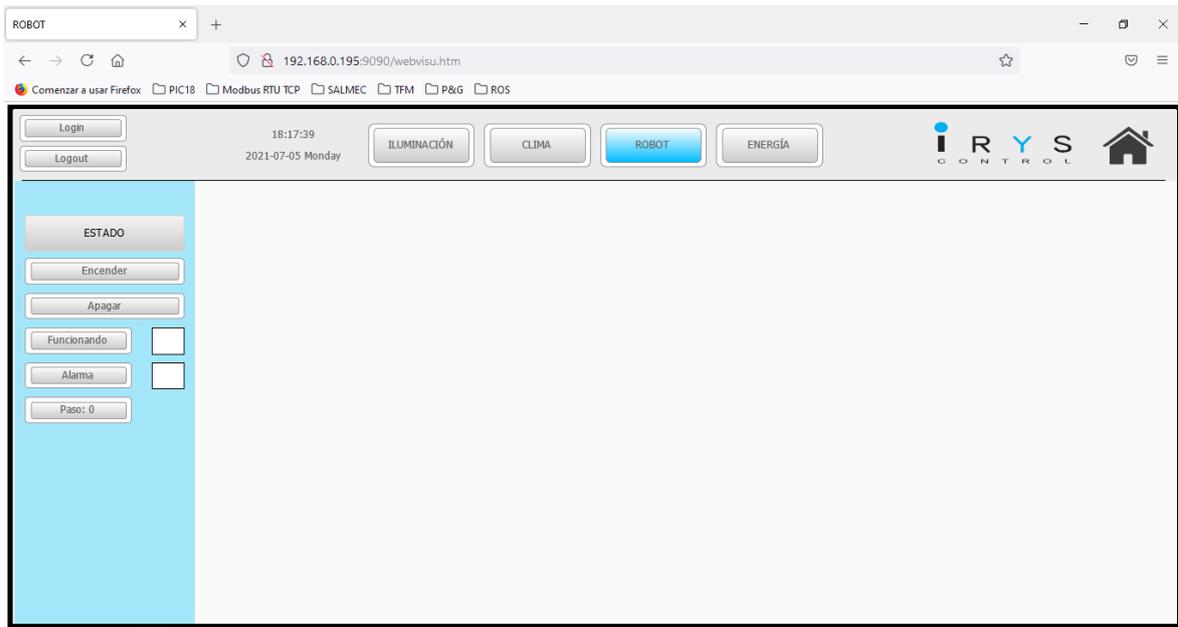


Ilustración 31: Pantalla Robot

#### 2.5.4. Resultados

Después de la investigación y elección se ha conseguido montar todo el sistema, tanto el robot y cafetera, como el dispensador y la parte eléctrica. Con el sistema montado y listo se realizaron las pruebas.

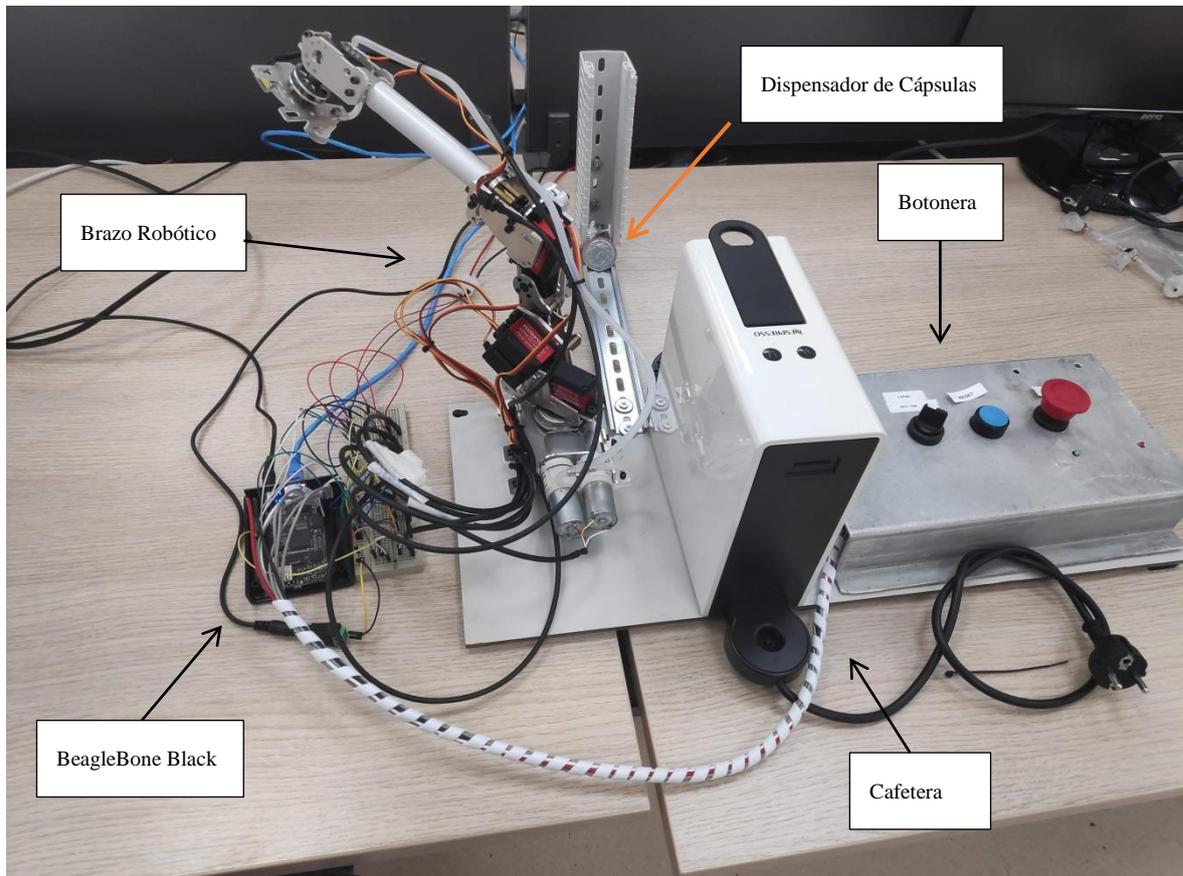
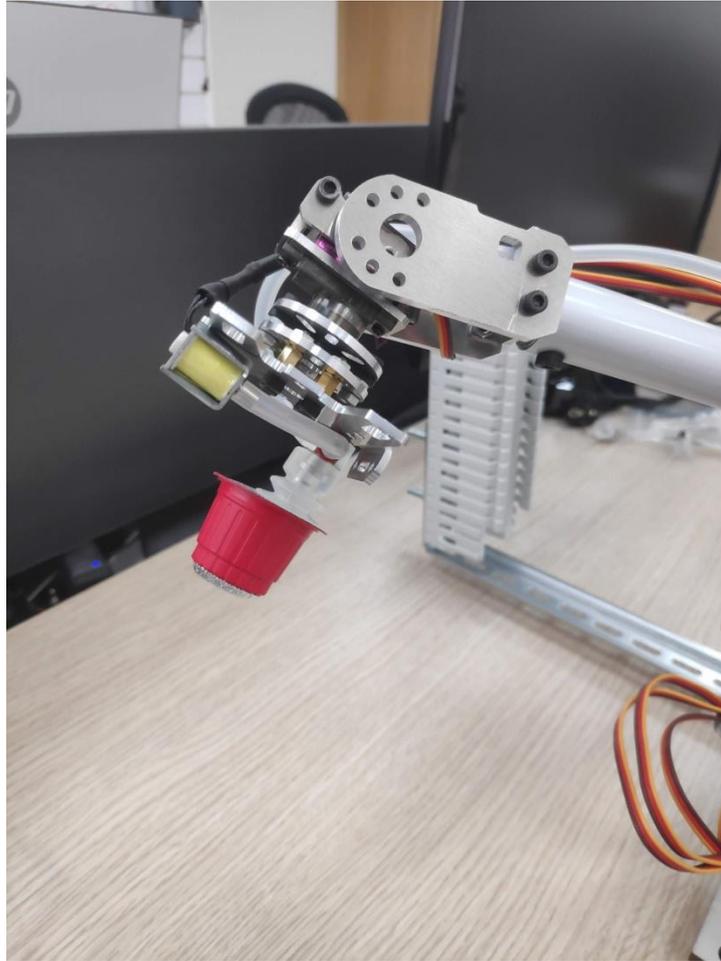


Ilustración 32: Sistema Robótico

La primera prueba ha sido el agarre de la capsula y comprobar el funcionamiento de las bombas y electroválvula, es decir, el correcto funcionamiento del sistema de vacío.

El robot inicialmente contaba con 4 ventosas en la herramienta final. Esto era un problema para el agarre de la capsula, ya que únicamente hay contacto entre la cápsula y una ventosa. Con lo cual, no es posible crear esa succión. La solución de este pequeño inconveniente fue, retirar las tres ventosas restantes y dejar una única ventosa. Una solución simple pero sencilla, consiguiendo la función que necesitábamos, coger la cápsula de café (Ilustración 33).



**Ilustración 33: Robot con cápsula**

En cuanto al robot, se ha probado y demostrado que se mueve sin problemas y llegando a la posición deseada. Hemos conseguido mover todas las articulaciones individualmente al ángulo deseado y realizar una trayectoria entre dos puntos. De los pasos que tenía asignado el robot para realizar su tarea. Se consiguió que el robot se colocara en la posición inicial, cogiera la capsula y la depositara en su lugar dentro de la cafetera.

La parte de levantar y bajar la tapa no se ha podido implementar. Se necesitan realizar algunas pruebas más o realizar algunas mejoras en el sistema mecánico para el levantamiento de la tapa. Con las estimaciones iniciales y las características del robot, se estimaba que iba a ser capaz de realizar esta función,

El sistema arranca tanto en modo local, como en modo remoto, desde la interfaz gráfica diseñada para el sistema. Y las comunicaciones con la segunda BeagleBone funcionan correctamente y sin problemas, pudiendo monitorizar y controlar el sistema robótico.

La botonera también realiza su trabajo correctamente, pudiendo controlar el robot con sus selectores e indicando el estado de este con sus dos leds. Estos led, se encienden cuando existe una alarma. Estas alarmas son producidas por los sensores de agua y del alimentador de cápsulas. Cuando desaparece el error, el led parpadea indicando que ya no existe la alarma y es posible rearmar el sistema. Al pulsar la seta de emergencia el sistema se para de inmediato, saltando una alarma de paro de emergencia.

Se ha creado un drive donde se han subido videos e imágenes del proyecto y de las pruebas realizadas.

[https://drive.google.com/drive/folders/1yCur9\\_1KxbKLQ35yStpiSax-SoPCHBqA?usp=sharing](https://drive.google.com/drive/folders/1yCur9_1KxbKLQ35yStpiSax-SoPCHBqA?usp=sharing)

### 3. Conclusiones

Como conclusión se ha podido instalar un brazo robótico en una plataforma de bajo coste como es la BeagleBone Black. La interfaz realiza su función de control y monitorización del sistema. Existen varias mejoras que se podrían realizar y que se tienen en cuenta en el apartado de trabajos futuros.

Aunque no se haya podido realizar la tarea completa que se tenía prevista, el sistema sí que ha sido capaz de realizar la mayoría de pasos, y con la ayuda de un usuario es capaz de poner un café.

La BeagleBone Black ha sido capaz de ejecutar varias tareas a la vez, como son ROS, OpenVPN y las comunicaciones TCP/IP. Mostrando que es posible instalar un sistema robótico en una plataforma de bajo coste como puede ser esta BeagleBone.

#### 3.1. Posibles mejoras

Una vez finalizado este Trabajo de Fin de Master, el sistema debe mejorar, ya que es posible implementar algunos nuevos elementos para darle mayor funcionalidad al sistema.

Una de ellas es que intentar de alguna manera que el robot pueda poner y quitar el vaso donde se dispense el café. Y con ello, que pudiera poner un número de cafés.

Se podría añadir varios dispensadores de cafés, cada uno con un tipo distinto, y el usuario elegir el café a su gusto,

Otra mejora posible es la instalación de una elongación de la tapa de café para que el robot no tenga que realizar tanta fuerza.

También sería posible mejora, como ya se ha comentado anteriormente, la de la interfaz gráfica. Se puede realizar una animación del propio robot dependiendo del paso en el que se encuentra.

Aparte de este visualizador por medio de un buscador, sería posible el diseño de app para dispositivos Android. Ya que sería más cómodo para los usuarios tener una app en el dispositivo móvil

En cuanto a los elementos de monitorización se puede mejorar el sistema de avisos del dispensador de cápsulas poniendo otro sensor un poco más arriba, es decir, que detecte cuando únicamente queden 3 cápsulas. Dejando que el sistema trabaje, pero alertando al usuario que dentro de poco se van a acabar las cápsulas.

Con el reciente aumento de popularidad de los sistemas de reconocimiento de voz como el dispositivo Alexa de Amazon, un trabajo para el futuro sería la implementación el reconocimiento de voz. Ya que con esta nueva aplicación se podría realizar los comando de poner café sin ningún tipo de dispositivo. Para conseguir esta función, sería necesario definir los comandos a recibir, y luego transformar el comando en respuesta sobre el sistema. Esta nueva interfaz entre el usuario y el sistema mejoraría bastante el confort y comodidad de los usuarios de este brazo robótico. Este sistema sería el proyecto que personalmente más me gustaría instalar y que en un futuro cercano se pueda realizar, lo vería como una función innovadora y con bastante utilidad para la vida cotidiana.

## 4. Apéndices

### 4.1. Comandos ROS desde terminal

Todo el proyecto ha sido desarrollado desde el software Putty. Con lo cual, todo se ha configurado y programado mediante scripts o comandos. En ros existe el paquete rosbash, el cual nos permite movernos entre paquetes de manera sencilla.

Al trabajar con ROS es importante conocer los comandos básicos en el terminal para poder trabajar más fácil y rápidamente.

Estos son unos de los comandos más utilizados para este proyecto.

#### 4.1.1. Rostopic

Los comandos que empiezan por rostopic dan información sobre los topics [17].

- **Rostopic echo /topic:** Muestra el contenido de los mensajes por pantalla.
- **Rostopic info /topic:** Muestra información sobre determinado topic activo, los topics a los que publica, los servicios, a quienes está suscrito.
- **Rostopic find tipoDeMensaje:** Encuentra topics por su tipo de mensaje.
- **Rostopic hz /topic:** Muestra el ratio de publicación de determinado topic.
- **Rostopic list:** Muestra información de los topics activos.
- **Rostopic pub /topic tipo parametros:** Publica en determinado topic, mediante línea de comandos.

#### 4.1.2. Rosnode

Todos los comandos comenzados por rosnode aportan información sobre los nodos que se están ejecutando actualmente en el sistema [18].

- **Rosnode info nombreDeNodo:** Da información sobre el nodo
- **Rosnode kill nombreDeNodo:** Mata a un nodo en ejecución.
- **Rosnode list:** Muestra por pantalla todos los nodos activos.
- **Rosnode ping nombreDeNodo:** Prueba la conexión de un nodo.
- **Rosnode cleanup:** Elimina la información de registro de los nodos que no son

accesibles.

## 4.2. Rosmsg

Este comando aporta información sobre los mensajes [19].

- **Rosmsg show:** Muestra los campos que componen el mensaje.
- **Rosmsg list:** Muestra una lista de todos los mensajes.
- **Rosmsg package:** Muestra todos los mensajes en el paquete.

## REFERENCIAS

- [1] Estos robots domésticos te harán la vida más fácil en casa. 12/2019  
[https://www.65ymas.com/sociedad/tecnologia/robots-domesticos-que-pueden-ayudarte-en-casa\\_558\\_102.html](https://www.65ymas.com/sociedad/tecnologia/robots-domesticos-que-pueden-ayudarte-en-casa_558_102.html).
- [2] Pepper <https://aliverobots.com/robot-pepper/>
- [3] Anónimo (2019). Automatización Industrial, Wikipedia, Recuperado el 10/2018, de [https://es.wikipedia.org/wiki/Automatizaci%C3%B3n\\_industrial#cite\\_note-1](https://es.wikipedia.org/wiki/Automatizaci%C3%B3n_industrial#cite_note-1)
- [4] Logicbu. (2018). Que es la automatización. 11/2018, de Logicbus Sitio web: <http://www.logicbus.com.mx/automatizacion.php>
- [5] Inteligencia Artificial. De sitio web:  
<https://freedomforlife.wordpress.com/grados-de-libertad/>
- [6] ROS <https://www.ros.org/>
- [7] Gerald Coley. (2013). BeagleBone Black System Reference Manual. 11/2018, de beagleboard.org Sitio web: [https://cdn-shop.adafruit.com/datasheets/BBB\\_SRM.pdf](https://cdn-shop.adafruit.com/datasheets/BBB_SRM.pdf)
- [8] De cero a maker: todo lo necesario para empezar con Raspberry Pi, 07/2018 de <https://www.xataka.com/makers/cero-maker-todo-necesario-para-empezar-raspberry-pi>
- [9] <https://naylorlampmechatronics.com/raspberry-pi/151-raspberry-pi-2-b.html>
- [10] Análisis de la Raspberry Pi 2.  
<https://descubrearduino.com/analisis-de-la-raspberry-pi-2/>
- [11] ESP8266 todo lo que necesitas saber del módulo WiFi para Arduino.  
<https://programarfácil.com/podcast/esp8266-wifi-coste-arduino/>
- [12] ¿Sabe cómo funcionan los sensores capacitivos? 07/2020  
<https://www.sepia.mx/sabe-como-funcionan-los-sensores-capacitivos/>
- [13] Detector de obstáculos con sensor infrarrojo y Arduino. 05/2016  
<https://www.luisllamas.es/detectar-obstaculos-con-sensor-infrarrojo-y-arduino/>

[14] El último robot de Toyota se cuelga del techo para no ocupar espacio mientras hace tareas del hogar. 10/2020

<https://www.xataka.com/robotica-e-ia/ultimo-robot-toyota-se-cuelga-techo-para-no-ocupar-espacio-hace-tareas-hogar>

[15] La primera cafetería robótica del mundo abre sus puertas en San Francisco. 02/2017

<https://www.lavanguardia.com/comer/sitios/20170201/413903120085/cafeteria-x-san-francisco-primera-cafeteria-robotica-mundo.html>

[16] Los robots llegan a las cafeterías, y éste es capaz de servir hasta 120 tazas cada hora sin descanso. 01/2017. <https://www.xataka.com/robotica-e-ia/los-robots-llegan-a-las-cafeterias-y-este-es-capaz-de-servir-hasta-120-tazas-cada-hora-sin-descanso>

[17] Rostopic. <http://wiki.ros.org/rostopic>

[18] Rosnode. <http://wiki.ros.org/rosnode>

[19] Rosmsg. <http://wiki.ros.org/rosmsg>

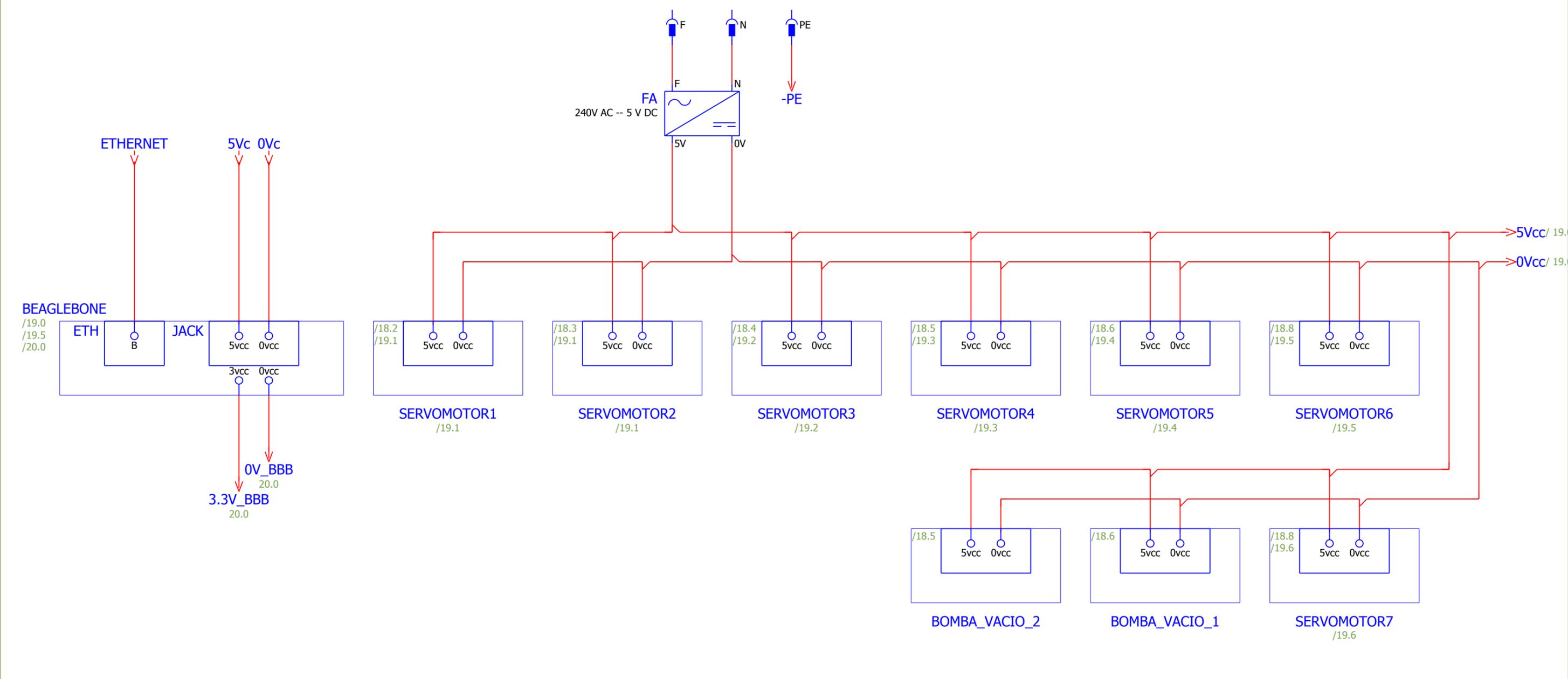
Anexos

Esquemas Eléctricos

# CONTROL DE BRAZO ROBÓTICO MEDIANTE ROS EN PLATAFORMA DE BAJO COSTE

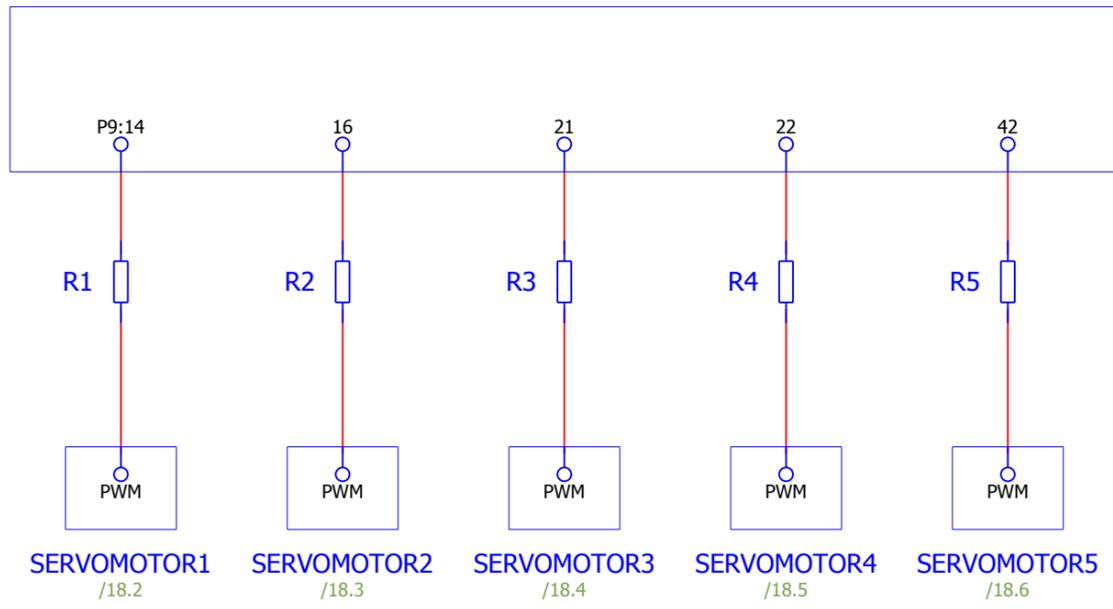
## TFM

			Fecha	16/06/2021	Proyecto	Portada		Cliente	TFM	Ubicación	Elche	
			Ed.	Tecnico	CONTROL DE BRAZO ROBOTICO MEDIANTE ROS						Pág.	1
Modificación	Fecha	Nombre	Original							V00001	Total	14



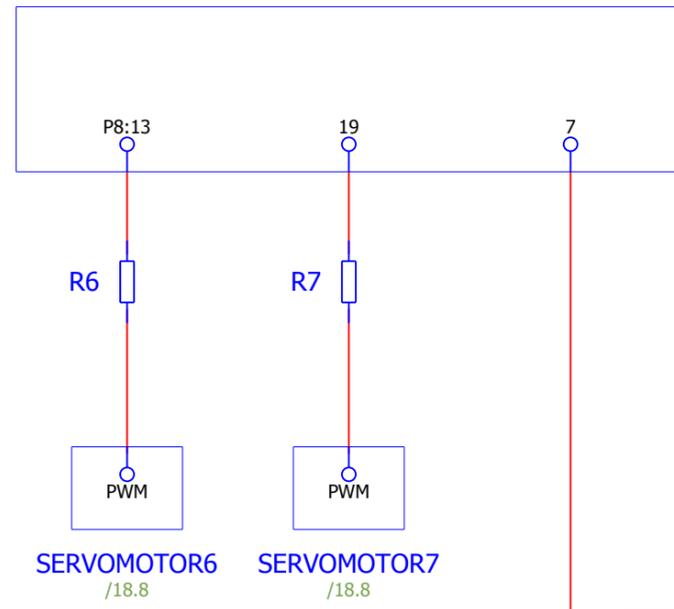
BEAGLEBONE

/18.0  
/19.5  
/20.0



BEAGLEBONE

/18.0  
/19.0  
/20.0



18.9 /5Vcc->

18.9 /0Vcc->

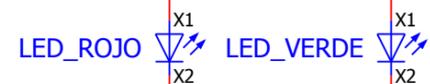
Fecha	29/06/2021	Proyecto			Control		Cliente	TFM	Ubicación	Elche	
Ed.	Tecnico	CONTROL DE BRAZO ROBOTICO MEDIANTE ROS								Pág.	19
Appr										Total	14
Modificación	Fecha	Nombre	Original					V00001			

BEAGLEBONE

/18.0  
/19.0  
/19.5



/18.0  
/19.0  
/19.5  
/20.0



18.1 /3.3V\_BBB->

18.1 /0V\_BBB->

Fecha	29/06/2021	Proyecto	CONTROL DE BRAZO ROBOTICO MEDIANTE ROS		Control		Ciente	TFM	Ubicación	Elche
Ed.	Tecnico									V00001
Modificación	Fecha	Nombre	Original							