



**Universidad**  
Zaragoza

## Trabajo Fin de Grado

Plataforma IoT para el desarrollo de una intervención  
electrónica colaborativa

IoT platform for the development of cooperative electronic  
intervention

Autor

**Francesco Ferrati Serrano**

Director

**José María López Pérez**

Escuela de ingeniería y arquitectura, Universidad de Zaragoza

2019

## Resumen

En el presente trabajo de fin de grado se muestra la colaboración realizada con la bailarina y coreógrafa Julia Zaccagnini y el Laboratorio de Danza y Nuevos Medios (DANM) del festival de danza en paisajes urbanos Trayectos. Se realiza con el objetivo de preparar un espectáculo de baile fruto de la colaboración entre danza y tecnología electrónica para el festival Trayectos 2019.

La intervención electrónica para la colaboración ha consistido en desarrollar un módulo inalámbrico con comunicación WiFi, capaz de detectar movimientos y activar diferentes actuadores según su programación. Se decidió que el diseño del hardware del módulo fuera objeto del TFG de Claudia Gonzalo y el diseño del firmware fuera objeto de este TFG. El desarrollo de la coreografía evoluciona en paralelo al desarrollo electrónico en clave de colaboración entre artista y tecnólogo. Para posibilitarla se establece una metodología que es considerada una parte capital en este TFG.

Debido a las circunstancias, el módulo desarrollado por Claudia no iba a estar listo para el festival y se tuvo que elaborar un plan de contingencia que consistía en un prototipo con menos funcionalidades, pero acorde a las necesidades principales.

A partir del festival, valorando la versatilidad del prototipo, se completa el sistema desarrollando un paquete de firmware que explore todas sus potencialidades tanto de sensorización como de comunicación.

Para facilitar su programación se ha desarrollado una librería en lenguaje C/C++ y se han programado diferentes posibles casos de uso.

# Índice

1.	GLOSARIO .....	1
2.	PREFACIO .....	2
	2.1 Origen del proyecto .....	2
	2.2 Motivación del proyecto .....	2
3.	INTRODUCCIÓN .....	2
	3.1 Estado del arte .....	2
	3.2 Antecedentes .....	4
	3.3 Objetivos .....	5
	3.4 Metodología y Herramientas .....	6
4.	COLABORACIÓN .....	9
	4.1 Fusión entre danza y electrónica .....	9
	4.2 Desarrollo de la colaboración .....	10
	4.2.1 Primera fase: presentaciones, planteamiento y comienzo de la colaboración .....	10
	4.2.2 Segunda fase: prueba y elección de componentes .....	12
	4.2.2.1 Primera sesión de trabajo .....	12
	4.2.2.2 Segunda sesión de trabajo .....	14
	4.2.2.3 Tercera sesión de trabajo .....	16
	4.2.3 Tercera fase: preparación espectáculo final .....	18
	4.2.3.1 Primera sesión de trabajo .....	20
	4.2.3.2 Segunda sesión de trabajo .....	20
	4.2.3.3 Tercera sesión de trabajo .....	20
	4.2.4 Cuarta fase: Actuación Festival Trayectos .....	21
5.	DESARROLLO DE LOS SISTEMAS ELECTRÓNICOS .....	25
	5.1 Hardware .....	26
	5.1.1 Módulo diseñado por Claudia Gonzalo .....	26
	5.1.2 Plan de contingencia .....	27
	5.1.2.1 Diseño esquemático .....	27
	5.1.2.2 Diseño PCB .....	35
	5.2 Firmware .....	39
	5.2.1 Comunicaciones .....	39
	5.2.2 Investigación sobre lenguajes de programación .....	40
	5.2.3 Diseño firmware .....	41
	5.2.3.1 Capas del firmware .....	41
	5.2.3.2 Librería TWM .....	42
	5.2.3.3 Casos de uso .....	54
6.	CONCLUSIONES .....	56
	6.1 Del proyecto .....	56
	6.2 Personales .....	56
7.	BIBLIOGRAFÍA .....	57
8.	ANEXOS .....	58

# 1. Glosario

Arduino Nano: es una placa de desarrollo de tamaño compacto. Posee los mismos componentes y capacidades que un Arduino UNO, tanto en potencia del microcontrolador como en conectividad. La pocas diferencias son que su tamaño es más reducido, que posee un conector Mini-USB, y que tiene un formato de pines header [1].

Arduino UNO: es una placa electrónica de desarrollo basada en el microcontrolador Atmega328P. Tiene 14 pines de entrada/salida digitales, 6 entradas analógicas, un cristal de 16Mhz, conexión USB, conector jack de alimentación, terminales para conexión ICSP y un botón de reseteo. Posee toda la electrónica necesaria para que el microcontrolador opere correctamente [2].

DANM: es el Laboratorio de Danza y Nuevos Medios, un proyecto que surge de la colaboración entre el festival Trayectos y Etopia Centro de Arte y Tecnología, junto a la Escuela de Ingeniería y Arquitectura de la Universidad de Zaragoza, en particular los Departamentos de Robótica, Ingeniería de Diseño y los alumnos del Grado de Ingeniería Electrónica y Automática [3].

Etopia: es el Centro de Arte y Tecnología de Zaragoza. Se trata de un edificio de nueva generación diseñado para albergar y promover los proyectos creativos y emprendedores más innovadores dentro del área de Milla Digital [4].

IMU: Inercial Measurement Units. Así se denominan los sistemas de medida inercial. Son dispositivos electrónicos que miden e informan acerca de la velocidad, orientación y fuerzas gravitacionales usando una combinación de acelerómetros y giróscopos [5].

IoT: Internet of things. Internet de las cosas es un concepto que se refiere a la interconexión digital de objetos a través de Internet [6]. El objetivo es el de poder mejorar la calidad de vida de las personas ya que se está desarrollando para poder conectar diferentes dispositivos en el hogar, la oficina, la calle o la fábrica. Se trata de un “mundo” en constante evolución y que apunta a asentarse en la vida cotidiana de las personas.

NodeMCU: es una placa de desarrollo basada en el microcontrolador ESP8266. Posee 32 pines, de los cuales, 17 de entrada/salida digitales, 1 con conversor ADC, y el resto se utilizan para la alimentación y control del microcontrolador. El ESP8266 permite la conexión WiFi de la placa [7].

RGB: es un término que se compone por las siglas de los términos “red”, “green” y “blue”, es decir, está relacionado con la representación de colores. Se trata de un modelo cromático mediante el cual es posible representar distintos colores a partir de la mezcla de estos tres colores primarios [8].

Hoverboard: es una tabla plástica puesta sobre dos ruedas. El usuario mantiene el equilibrio sobre el artefacto a través de sensores, como giroscopios, que además permiten controlar su movimiento [9].

Trayectos: es el Festival Internacional de Danza en paisajes urbanos de Zaragoza que se celebra cada año desde 2004 [10].

3DOF: Three Degrees Of Freedom. Tres grados de libertad, se refiere al movimiento libre en un espacio tridimensional, es decir, la posibilidad de poderse mover a lo largo de cada uno de los ejes independientemente de los otros [11].

## 2. Prefacio

### 2.1 Origen del proyecto

El presente proyecto tiene su origen en haber cursado la asignatura de Laboratorio de diseño electrónico, asignatura optativa de cuarto del grado de Ingeniería Electrónica y Automática de la Universidad de Zaragoza. Esta asignatura se basa en el desarrollo de un producto que contiene un sistema electrónico en colaboración interdisciplinar con los estudiantes de diseño industrial.

La parte de trabajo de los estudiantes de electrónica se basa en la investigación de tecnologías, elección de componentes, diseño, construcción y puesta a punto de la electrónica e integración en un prototipo funcional.

Uno de los profesores de esta asignatura es José María López Pérez, director de este TFG, que durante el curso propuso en clase a los alumnos si alguno estaba interesado a iniciar un nuevo proyecto y seguir aprendiendo sobre el desarrollo de sistemas electrónicos, pero esta vez colaborando con gente perteneciente a una disciplina muy diferente a la ingeniería.

### 2.2 Motivación del proyecto

El motivo principal por el que se ha seleccionado este tipo de proyecto es para aprender a colaborar con personas de otra disciplina diferente a la propia para poder llegar a un objetivo común. En este sentido, es una excelente oportunidad de aprendizaje trabajar con las creadoras de Trayectos, dada su experiencia en colaboraciones interdisciplinares. Además, un punto de partida motivador es la idea de que la tecnología y la danza son dos ámbitos muy diferentes, pero al mismo tiempo se intuye que su interacción puede llevar a resultados fascinantes.

El otro motivo, puramente técnico, por el que se ha escogido este tipo de proyecto, es que permite aprender respecto a todas las fases del desarrollo de sistemas electrónicos desde la definición de funciones hasta la puesta a punto, pasando por la selección de componentes óptimos, el diseño electrónico y el de la PCB. Además, este TFG puede ayudar a profundizar en el ámbito de la programación de un microcontrolador, sobre todo lo que concierne a la comunicación con otros dispositivos.

## 3. Introducción

### 3.1 Estado del arte

En los últimos años ha habido un notable crecimiento respecto a la búsqueda de nuevas tecnologías para poder sensorizar los movimientos del cuerpo.

Bajo este aspecto, la variedad de sensores acoplables en el cuerpo es elevada, pueden encontrarse sensores independientes que requieren un circuito externo para poder funcionar, o bien, se pueden encontrar incorporados a módulos compactos que incluyen el circuito necesario para poder almacenar los datos captados.

Un ejemplo claro es el módulo MPU 9250. Se trata de un sensor de movimiento de 9 ejes que incorpora en un mismo integrado un acelerómetro, un giróscopo y un magnetómetro. También posee un procesador interno para procesar y almacenar las mediciones de los sensores.



*Figura 3.1: Módulo MPU 9250 de InvenSense*

Una vez registrados los datos, es necesario poder analizarlos de alguna manera. La forma más cómoda es la de poder enviar estos datos a otro dispositivo (móvil, tablet o PC) a través de comunicación WiFi o Bluetooth. Para permitir esto, es necesario que el sensor esté controlado por un microcontrolador que reciba los datos y tenga la capacidad de poderlos enviar vía WiFi o Bluetooth.

Uno de los microcontroladores más nuevos y completos en el mercado que permite este tipo de comunicaciones es el ESP-WROOM-32.



*Figura 3.2: ESP-WROOM-32 de Espressif Systems*

En la danza, el caso más representativo de sensorización aplicada al cuerpo es el proyecto desarrollado por la diseñadora y bailarina Lesia Trubat con el cual ha creado las “E-Traces”, unas zapatillas con incorporadas pequeñas placas electrónicas en la punta y en el costado que sirven para registrar los movimientos de los bailarines [12].

Se registran los datos generados por el contacto del pie con el suelo, y se envían a una aplicación de móvil. Esto permite que los bailarines puedan ver en video sus movimientos para analizarlos y mejorar sus actuaciones [12].



*Figura 3.3: Zapatillas de danza E-Traces*

### 3.2 Antecedentes

Este TFG está enmarcado en una colaboración entre profesores y estudiantes de electrónica de la EINA con el Laboratorio de Danza y Nuevos Medios (DANM) del festival Trayectos.

La colaboración comenzó en el curso 2016-2017 y fueron participes los estudiantes de la asignatura de Laboratorio de Diseño Electrónico del Grado de Telecomunicaciones. En la asignatura los alumnos tratan de desarrollar diferentes tipos de proyectos basados en una temática común que aquel curso fue “*Intervenciones electrónicas en la danza*”. La colaboración consistió en diferentes sesiones de exposición y comunicación con el personal de Trayectos sobre montajes que incorporan tecnología y un análisis sobre los movimientos en la danza contemporánea. Una vez finalizadas estas sesiones los estudiantes propusieron sus proyectos que después desarrollaron y construyeron.

En 2018 se realizó un TFG basado en una intervención electrónica en la pieza “Formas que puedo nombrar” (coreografía de Alba Lorca) en la que la coreografía y la intervención electrónica se desarrollaron paralelamente dando lugar a un espectáculo en el que los bailarines vestían unos trajes que incorporaban la electrónica desarrollada que reaccionaba a su interacción. Fue presentado en el festival Trayectos de ese mismo año [13].



Figura 3.4: Representación de la pieza 'Formas que no puedo nombrar'

En 2019 se han desarrollado dos propuestas de colaboración: la primera, correspondiente a la pieza "Kn0w y0ur 4ngl3s" de la bailarina Julia Zaccagnini, que es objeto de este TFG y la segunda, objeto del TFG de Claudia Gonzalo, en colaboración con la bailarina Raquel Buil [14].



(a)



(b)

Figura 3.5: (a) Colaboración electrónica con Julia Zaccagnini (b) Colaboración electrónica con Raquel Buil

### 3.3 Objetivos

El objetivo principal es explorar las posibilidades de colaboración entre la electrónica y la danza a través del desarrollo de una intervención electrónica en una coreografía, ambas desarrolladas en forma cooperativa. Así se demuestra que la electrónica y la danza, siendo disciplinas diferentes, se pueden unir para realizar un objetivo común.

La función de la electrónica es la de reforzar el mensaje que quiere transmitir la bailarina, ayudar al espectador a entenderlo y generar un espectáculo artístico innovador. Por tanto, se va a diseñar y programar un módulo inalámbrico, con sensores para poder detectar movimientos y salidas que

provoquen actuaciones externas. El módulo estará controlado por un microcontrolador con posibilidad de comunicación WiFi y Bluetooth.

A nivel personal, el principal objetivo es poder aprender a través de este proyecto y adquirir nuevos conocimientos sobre el diseño electrónico y la programación. En segundo lugar, el objetivo es de ser capaz de colaborar con otras personas que pertenecen una disciplina diferente a la mía y juntos establecer una cancha común para alcanzar los resultados previstos.

### 3.4 Metodología y Herramientas

La metodología está basada en el trabajo cooperativo entre la bailarina y el estudiante de ingeniería electrónica para producir una pieza de danza contemporánea con intervención electrónica. En lugar de encargos desarrollados individualmente se trata de que ambas personas colaboren durante todo el proceso, exista un aprendizaje mutuo y se consiga un resultado innovador que no podría alcanzarse al trabajar por separado. En este sentido se considera que el proceso de creación es incluso más importante que el resultado final.

Se forma así un equipo interdisciplinar de danza y tecnología, en el que los participantes tendrán que aprender a comunicarse (ya que usan lenguajes distintos y tienen una concepción de la realidad distinta) y a trabajar cooperativamente para desarrollar un proyecto común.



*Figura 3.6: Equipo interdisciplinar formado por Julia Zaccagnini (bailarina) y Francesco Ferrati (estudiante ing. Electrónica)*

Esta metodología de base se ha ido materializando en las tareas de las que ha estado compuesto este proceso. Posteriormente se marcan temporalmente en un diagrama de Gantt:

- Reuniones iniciales con los departamentos que participaban en este proyecto: para conocerse y crear el grupo de trabajo, participaban el personal de Trayectos (cuya sede está en Etopia) y el de electrónica. Es importante señalar que Claudia Gonzalo también ha realizado su TFG en colaboración con Trayectos con otra bailarina, por ello estas primeras reuniones se hicieron de forma conjunta con ella.
- Primera fase de la colaboración: como primera toma de contacto entre las distintas partes de la colaboración, se realizó, a través de la plataforma Skype, una videoconferencia de presentación con Julia, la bailarina de esta colaboración. Se empezó a plantear el método de trabajo, las ideas que manejaba la bailarina para su coreografía y las posibilidades tecnológicas. En los meses sucesivos los contactos entre ambas partes eran continuos a través del correo electrónico, intercambiando ideas y planeando poco a poco cómo habría sido el espectáculo final.
- Presentaciones formales en Etopia: una serie de reuniones con Manon (la bailarina que inicialmente iba a participar en la colaboración de Claudia Gonzalo) donde se le mostraron los distintos componentes a disposición para que pudiera familiarizarse con ellos y ver el tamaño real. En estas reuniones mi trabajo era el de ayudar a Claudia. En el mes de mayo, con la llegada de Julia a Zaragoza, se habría realizado el mismo trabajo con ella.
- Investigación sobre posibles lenguajes de programación: se realizó un estudio entre los posibles lenguajes a utilizar para programar el módulo inalámbrico, intentando averiguar cuál era el apropiado para este proyecto.
- Segunda fase de la colaboración: una serie de reuniones iniciales en Etopia con Julia donde el objetivo era mostrar a la bailarina el material disponible para la colaboración para que pudiera familiarizarse con él y decidir conjuntamente qué utilizar para su pieza.
- Tercera fase de la colaboración: una serie de sesiones de trabajo con Julia, donde el objetivo era probar los componentes elegidos para la actuación. Estas pruebas servían para decidir cómo programar los distintos componentes y ver si seguían funcionando correctamente utilizándolos a la hora de bailar. También se realizó un ensayo abierto en Etopia, varios días antes de la actuación final, lo que conllevó a una serie de reuniones técnicas para decidir como montar el escenario.
- Plan de contingencia, diseño de esquemático y PCB: debido a circunstancias imprevistas que se explicaran posteriormente, fue necesario elaborar un plan de contingencia realizando el diseño del esquemático y consiguiente ruteo de la PCB de otro módulo inalámbrico con menos funcionalidades que el original.
- Montaje placas: una vez obtenidos todos los componentes del plan de contingencia, se procede a soldarlos sobre placa del nuevo módulo inalámbrico. Se montaron cuatro módulos.
- Actuación final en el festival Trayectos 2019: con el comienzo del festival de Trayectos, se realizó en Etopia la actuación final fruto de esta colaboración con la bailarina y coreógrafa Julia Zaccagnini.
- Desarrollo firmware, programación librería TIM: se desarrolla una librería, en lenguaje C/C++, compatible con el entorno Arduino IDE para poder controlar el módulo inalámbrico ideado para el plan de contingencia.
- Desarrollo firmware, programación casos de uso: se desarrollan tres programas en el entorno Arduino IDE correspondientes a los tres casos de uso posibles para el módulo inalámbrico ideado para el plan de contingencia.
- Documentación y puesta a punto de la memoria: se recopila información a lo largo de todo el proyecto para luego poder reflejarla de manera detallada en la memoria.

	<b>Meses</b>									
<b>Tareas del proyecto</b>	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE
Reuniones iniciales										
Presentaciones formales en Etopia										
Primera fase de la colaboración										
Investigación sobre posibles lenguajes de programación										
Segunda fase de la colaboración										
Tercera fase de la colaboración										
Plan de contingencia: diseño esquemático y PCB										
Montaje placas										
Actuación final festival Trayectos 2019										
Desarrollo firmware: programación librería TIM										
Desarrollo firmware: programación casos de uso										
Documentación y puesta a punto de la memoria										

Tabla 3.1: Diagrama de Gantt

Las herramientas utilizadas en el proyecto han sido las siguientes:

- Trello  
Trello es una herramienta de gestión de proyectos. Se compone de un *board* que contiene listas que corresponden a las tareas que hay que realizar en el proyecto. Los ítems dentro de las listas, llamados *cards*, sirven para insertar que hay que hacer en cada tarea pudiendo añadir imágenes, archivos adjuntos, y comentarios. Un *board* se puede compartir entre varios usuarios, y se permite que cualquier usuario pueda compartir archivos y comentarios en cualquier lista [15].  
Por tanto, esta herramienta ha sido útil para la organización del trabajo y la comunicación entre el estudiante y el director de este TFG.
- IDE Arduino  
Se trata de un entorno integrado de desarrollo de Arduino escrito en lenguaje de programación Java. El código se compone de dos funciones básicas, la primera se trata del *setup()* donde se encuentran las instrucciones de inicialización, y la segunda es el ciclo infinito *loop()* donde se encuentra la parte de programa que se va a ejecutar [16].  
Esta herramienta se ha utilizado para desarrollar la programación de todos los sistemas electrónicos realizados en el proyecto.
- Circuit Maker  
Se trata de un software de simulación de circuitos electrónicos, que facilita el diseño de circuitos tanto digitales como analógicos. Se ha elegido principalmente por ser gratuito, muy práctico y sencillo de utilizar. La única desventaja que tiene es que para utilizarlo se necesita conexión a Internet, lo que dificulta su funcionamiento si no se posee una buena conexión.  
Este programa se ha utilizado para hacer el esquemático del circuito y el consecuente diseño y ruteo de la PCB del módulo inalámbrico del plan de contingencia.

## 4. Colaboración

Como ya se ha explicado anteriormente, este TFG ha sido fruto de una colaboración entre profesores y estudiantes de electrónica de la EINA, la bailarina y coreógrafa Julia Zaccagnini y el Laboratorio de Danza y Nuevos Medios (DANM) del festival de danza en paisajes urbanos Trayectos.

En esta parte de la memoria se muestra de forma más detallada cómo se ha desarrollado. Se puede decir que esta colaboración se ha compuesto de cuatro fases y que ha concluido con la celebración del festival de Trayectos 2019.

### 4.1 Fusión entre arte y tecnología

El arte y la tecnología son dos disciplinas que se encuentran estrictamente relacionadas, a pesar de las diferencias que a priori tienen entre sí. La tecnología, que invade todos los órdenes de la vida, también es muy buscada en el mundo del arte, ya que abre posibilidades innovadoras que atraen fuertemente a

los artistas. En muchas ocasiones es posible realizar una determinada obra gracias a la tecnología que permite al artista llevarla a cabo. Igualmente, muchas nuevas tecnologías que mejoran o facilitan la vida cotidiana se han podido crear gracias a que han sido inspiradas por el arte. En este sentido, la relación entre ambas disciplinas ha ido evolucionando en los últimos tiempos tanto que en muchas ocasiones la tecnología posee una presencia tan importante en el arte, que es casi imprescindible para el desarrollo de esta. Por lo que respecta a la tecnología, desde su inicio se vinculó a diferentes formas de arte, algunas de estas ya están asentadas como es el caso de la fotografía o el vídeo. En la actualidad, la incorporación de las nuevas tecnologías ofrece a artistas y tecnólogos un extraordinario potencial de innovación.

En el caso de este TFG, la relación entre arte y tecnología viene reflejada en la intervención de la electrónica en el ámbito de la danza contemporánea. Esta colaboración pertenece al llamado arte digital, que se ha extendido a todas las facetas del arte incluso generando nuevas disciplinas.

## 4.2 Desarrollo de la colaboración

### 4.2.1 Primera fase: presentaciones, planteamiento y comienzo de la colaboración

La primera fase de la colaboración comienza, una vez tomada, por mi parte, la decisión de comenzar este TFG. Esta primera fase trató sobre todo de hacer reuniones para profundizar el conocimiento, intercambiar ideas y, principalmente, plantear el método de trabajo que se emplearía durante toda la colaboración.

Las primeras reuniones se realizaron en Etopia con Chema López, director de este proyecto, y Alba Lorca, responsable del Laboratorio de Danza y Nuevos Medios de Trayectos. Para realizar esta colaboración había dos bailarinas disponibles. Después de varias reflexiones y teniendo en cuenta lo que estaban realizando las dos bailarinas para el festival de Trayectos, se decidió de común acuerdo que la colaboración se realizaría con la bailarina argentina Julia Zaccagnini.

Esta decisión fue tomada porque cursando la asignatura de Laboratorio de Diseño Electrónico, estuve trabajando con sistemas electrónicos de comunicaciones con móviles y eso encajaba con la propuesta de Julia. Ella venía trabajando desde hace tiempo en una pieza, llamada “Kn0w y0ur 4ngl3s”, que se basaba sobre como impactan las tecnologías de comunicación sobre la constitución de las nuevas corporalidades en la contemporaneidad. En concreto, decía que era una coreografía centrada en el concepto Cuerpo-Identidad-Danza Postinternet, inspirada en el gesto del selfie y fabricada por medio de la idea de “Do it yourself”.



*Figura 4.1: Representación de la pieza “Kn0w y0ur 4ngl3s” de Julia Zaccagnini*

El programa del festival de Trayectos, que se iba a celebrar a finales de junio de 2019, establecía que los bailarines disfrutaran de un periodo de residencia previo al festival para desarrollar su trabajo. Julia estuvo en la residencia de Etopia durante los meses de mayo y junio. Por tanto, los primeros contactos con ella se establecieron por correo electrónico o por videoconferencia.

Se realizó una primera videoconferencia de presentación con Julia, a través de la plataforma Skype, para explicarle en qué consistía este TFG y para plantear el método de trabajo a seguir durante la colaboración. También se le explicó que, para aportar a la colaboración, se estaba desarrollando un pequeño módulo inalámbrico versátil, que tenía distintas funciones, como encender tiras de LEDs programables, activar sonidos, diversas entradas analógicas y además iba a tener comunicación WiFi.

Durante los meses sucesivos los contactos entre ambas partes eran continuos a través del correo electrónico, intercambiando ideas y planeando poco a poco cómo iba a ser el espectáculo final. Esta parte de la colaboración fue muy útil para entender cuál era la idea que Julia tenía sobre cómo enlazar el trabajo que había llevado a cabo hasta el momento, con el trabajo que se iba a desarrollar a través de esta colaboración.

Estos meses de constante comunicación fueron útiles para darse cuenta de que es muy importante llegar a un acuerdo que pueda satisfacer las necesidades de ambas partes, y sobre todo dejar claros al principio cuáles son los límites y capacidades de cada uno. Esto está relacionado con que, sobre todo durante estos primeros meses, Julia no conociendo todavía plenamente en qué consistía nuestro trabajo preguntaba si se podían realizar cosas que realmente no estaban a nuestro alcance o que no se disponía del material suficiente para llevarlo a cabo. Otras veces, pasaba lo contrario, que por nuestra parte se proponían ciertas ideas, que se pensaba que podían ser útiles a su trabajo, mientras que por su parte no despertaba interés.

No obstante, se coincidía en que cualquier tipo de idea era útil a la causa, por tanto, nunca se desechó ninguna antes de valorarla bien. Muchas veces alguna idea que aparentemente parecía imposible, a partir de ella se llegó a buscar una alternativa similar más simple que fuera posible de realizar.

Esta parte de la colaboración fue muy útil para generar ideas, comentarlas, ver si eran viables o si hacía falta buscar una alternativa similar y finalmente pensar como poder llevarlas a cabo con los medios que se tenían disponibles en el laboratorio. Todo esto ha servido para mejorar el

conocimiento entre ambas partes y sobre todo para llegar con las ideas más claras a la segunda fase de la colaboración.

Lo que se ha descrito muestra claramente el valor extraordinario de la colaboración interdisciplinar, clave para la consecución de resultados innovadores en cualquier campo.

#### 4.2.2 Segunda fase: prueba y elección de componentes

La segunda fase de la colaboración comenzó cuando Julia llegó a Zaragoza para disfrutar de dos meses (mayo y junio) de residencia en Etopia como establecido por la organización del festival de Trayectos, y consistió en varias sesiones de trabajo donde se enseñó a la bailarina el abanico de componentes que se tenían disponibles en el laboratorio para la colaboración, de modo que los pudiera probar y se hiciera una idea más exacta de cómo enfocar el trabajo. Este fue el punto de partida del desarrollo electrónico que se ha llevado a cabo durante toda la colaboración.

##### 4.2.2.1 Primera sesión de trabajo

En la primera sesión de trabajo que se realizó en Etopia, como todas las demás, después de un breve momento de presentación del grupo y metodología de trabajo por parte de Chema, se trató de enseñar a Julia el funcionamiento de algunos componentes que se tenían disponibles. Se le explicaron las funciones que iba a tener el módulo inalámbrico que se estaba desarrollando para la colaboración y se hicieron varias pruebas con algunos prototipos basados en Arduino, que intentaban simular a gran escala las variadas funciones que podía tener este módulo.

##### Prototipo con placa Arduino UNO que se ha desarrollado para la primera sesión:

###### **Funcionamiento:**

Se trata de un circuito montado en una protoboard, controlado por un Arduino UNO, al que están conectados un acelerómetro, un zumbador y aunque no aparezca en la figura 4.3 también se había conectado una tira de LEDs RGB direccionable. De modo que se había programado la placa Arduino para que cuando se produjera un movimiento sobre el eje X activara el pitido del zumbador, mientras que cuando se produjera el movimiento en el eje Y encendiera la tira de LEDs. El código desarrollado se encuentra en el Anexo 6.1.

###### **Componentes:**

- *Zumbador*

Se trata de un pequeño zumbador piezoeléctrico pasivo, que si se alimenta a una tensión comprendida entre 3.3 V y 5 V, permite generar cualquier sonido entre 2 KHz y 5 KHz.



Figura 4.2: Zumbador

- *Tiras de LEDs RGB direccionables*

Se trata de una tira de LEDs RGB programable con una tensión de alimentación de 12 V. Se hablará de este componente más detalladamente en el apartado 5.1.2.1 donde se explica el hardware del módulo inalámbrico diseñado para poder controlar este mismo tipo de tira de LEDs.

- *MPU 9250*

Se trata de un circuito integrado que contiene un acelerómetro, un giróscopo y un magnetómetro todos ellos de 3 DOF. Se hablará más detalladamente de este integrado en el apartado 5.1.2.1.

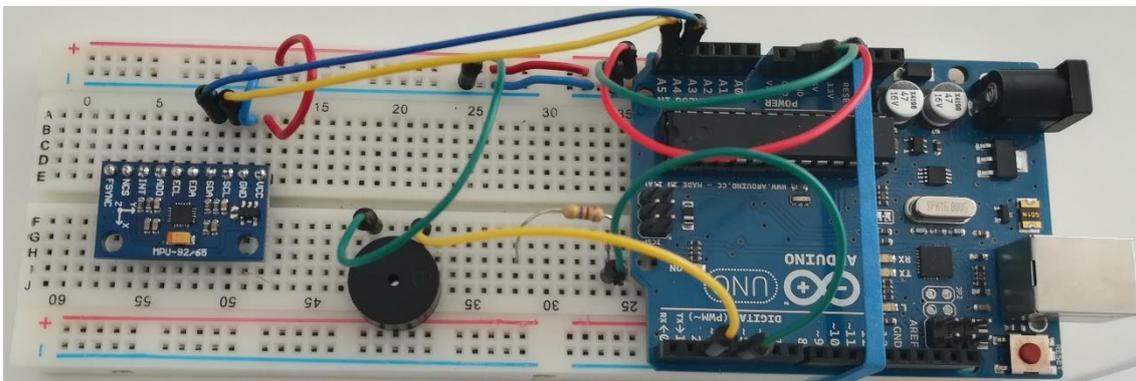
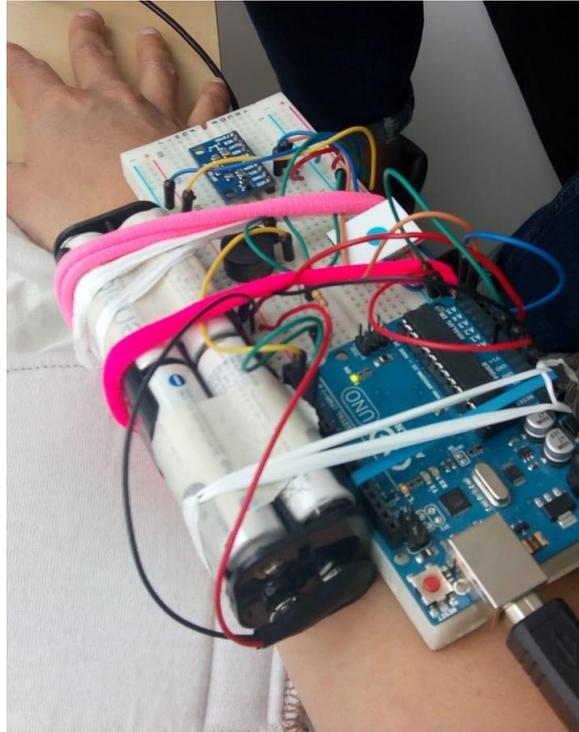


Figura 4.3: Protoboard realizada para la primera sesión de trabajo de la segunda fase

Todo esto se realizó, porque el módulo inalámbrico que se estaba desarrollando para la colaboración, iba a tener un acelerómetro similar al de esta protoboard, y se había pensado para que se pudieran conectar tiras de LEDs RGB direccionables y que además tuviera algunas salidas digitales como en este caso es la salida a la que está conectado el zumbador. Por tanto, este prototipo se colocó sobre el brazo de Julia para que, moviéndolo, el acelerómetro detectara el movimiento y, viendo los efectos que ello producía, poder enseñarle como podría funcionar el módulo inalámbrico.



*Figura 4.4: Colocación de la protoboard sobre el brazo de Julia*

#### *4.2.2.2. Segunda sesión de trabajo*

En la segunda sesión de trabajo de esta fase se trató de enseñar a Julia nuevas opciones. Al terminar la primera sesión, ella había dejado claro que no quería descartar nada y que quería probar todo lo que teníamos disponible y luego elegir lo más adecuado para acoplarlo a su trabajo. Se estuvo valorando la opción de poder colocar tiras de LEDs o zumbadores sobre su cuerpo. La opción quedó descartada desde el principio ya que la bailarina manifestaba la idea de que quería sentirse libre al bailar y colocar componentes sobre su cuerpo le habría resultado incómodo.

Por tanto, a partir de este punto, el trabajo estuvo enfocado en que, aparte el módulo inalámbrico, el resto de componentes estuvieran colocados para crear efectos en el escenario.

Esta sesión sirvió para enseñar a Julia nuevas opciones y hacer pruebas sobre cómo podría haber sido el escenario final. Se desarrollaron varios prototipos programados con Arduino.

*Prototipos que se desarrollaron para esta sesión:*

##### **Protoboard nº1**

##### ***Funcionamiento:***

El circuito de esta placa estaba controlado por un Arduino Nano, al que se conectaron un sensor ultrasónico y dos tiras de LEDs RGB direccionables. Se programó para que cuando se detectara algún obstáculo que se encontrara a menos de 100 cm del sensor, el microcontrolador activara una de las tiras de LEDs. A la segunda detección por parte del sensor ultrasónico, se apagaba la primera tira y se encendía la segunda. Una idea inicial que surgió para el espectáculo final, era que se podría haber instalado un circuito similar a este en algún punto del escenario, de modo que cuando la bailarina

pasara por ese punto se encendiera alguna luz presente en el propio escenario o incluso entre el público. El código desarrollado se encuentra en el Anexo 6.2.

### **Componentes:**

- *Tiras de LEDs RGB direccionables*

Dos tiras de LEDs RGB programables con una tensión de alimentación de 12V. Se hablará de este componente más detalladamente en el apartado 5.1.2.1.

- *Sensor ultrasónico HC-SR04*

Se trata de un sensor de distancia ultrasónico de alta precisión cuya tensión de alimentación es de 5V. Se ha elegido este tipo de componente porque es compatible con la placa Arduino. El sensor además de los pines de VCC y GND, posee otros dos pines llamados Trigger y Echo. Para poder saber a que distancia se encuentra un obstáculo de este sensor, se envía una señal en alto al pin Trigger durante unos pocos microsegundos y el sensor generará por el emisor un sonido de alta frecuencia que rebotará sobre el obstáculo y llegará al receptor del sensor. Durante el tiempo en el que el sensor genere este sonido, pondrá en estado de alto el pin Echo y cuando lo reciba de vuelta por el receptor lo pondrá en estado de bajo. Por tanto, midiendo el tiempo en el que el pin Echo permanece en el estado de alto y partiendo ese valor por la mitad, obtenemos el tiempo que el sonido ha tardado en llegar al obstáculo. Sabiendo el valor de la velocidad del sonido en el aire, podemos calcular a qué distancia se encuentra el obstáculo del sensor de la siguiente manera:

- *Cálculo distancia obstáculo:*

Sabiendo que la fórmula de la distancia es:

$$distancia = velocidad * tiempo$$

La distancia, en metros, a la que se encuentra el obstáculo la obtendremos a través de la siguiente expresión:

$$distancia\ obstaculo = V_{sonido} * \frac{Techo}{2}$$

*Vsonido: la velocidad del sonido en el aire (340 m/s)*

*Techo: es el tiempo durante el cual el pin Echo permanece en estado HIGH*

Siendo que Techo es del orden de microsegundos y la distancia a la queremos detectar el obstáculo es del orden de centímetros, pasamos la velocidad del sonido a unidades de cm/ $\mu$ s y obtenemos que:

$$distancia\ obstaculo\ (cm) = V_{sonido} * \frac{Techo}{2} = 0,034 * \frac{Techo}{2} = 0,017 * Techo$$

Todos estos cálculos y la medición del tiempo en estado en alto del pin Echo se realizan por software.



Figura 4.5: Sensor ultrasónico HC-SR04

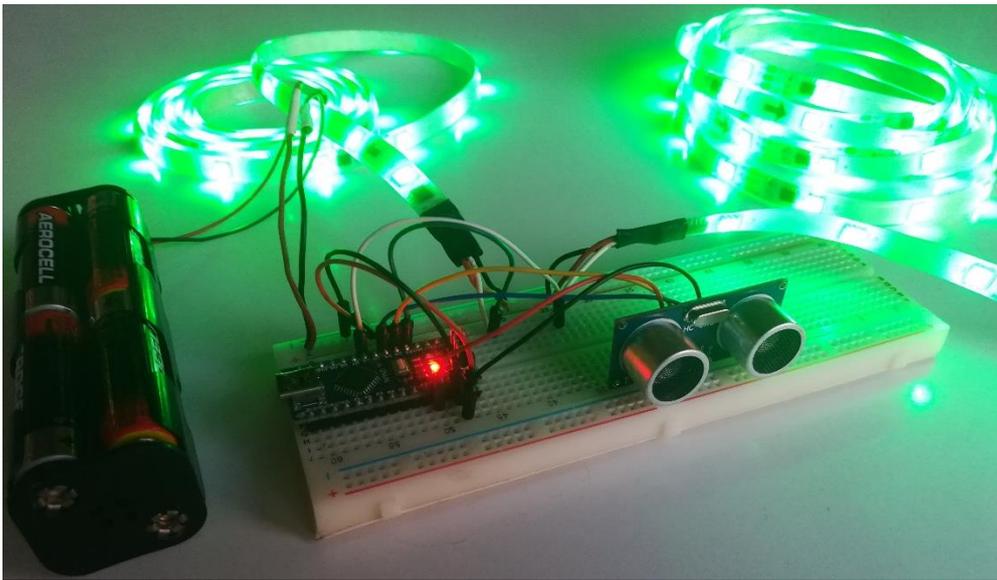


Figura 4.6: Protoboard nº1 realizada para la segunda sesión de trabajo de la segunda fase

Se realizaron otras dos protoboards para esta sesión. La descripción de los esquemas y el funcionamiento de cada una se encuentra en el Anexo 3.

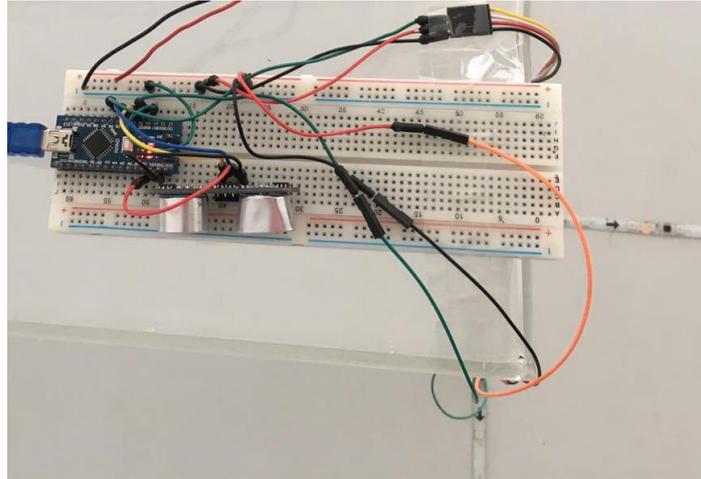
#### 4.2.2.3 Tercera sesión de trabajo

La tercera y última sesión de esta fase de la colaboración, después de haber mostrado a Julia todo el material disponible en las sesiones anteriores, sirvió para decidir conjuntamente qué componentes se iban a utilizar para el espectáculo final.

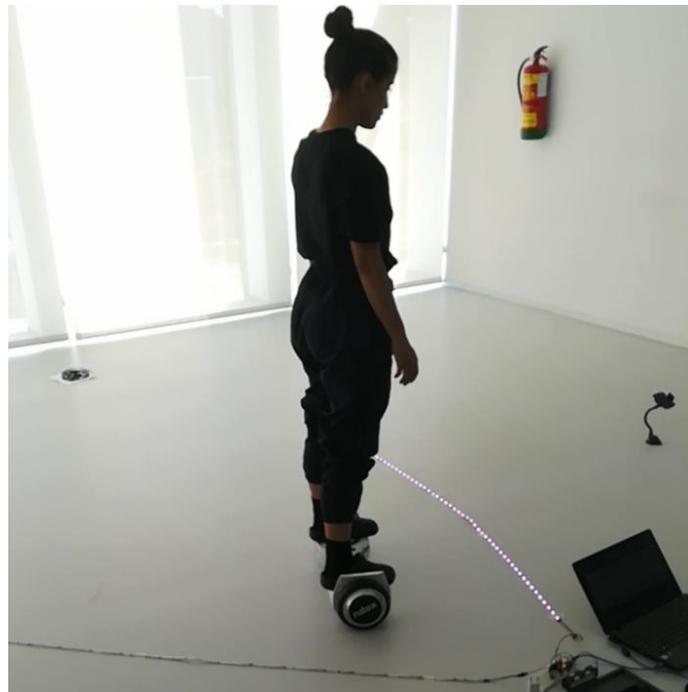
La elección final fueron dos tipos diferentes de placas:

- Placa con Arduino Nano, sensor ultrasónico y tiras de led direccionables

Se trata de la misma protoboard nº1 utilizada para la segunda sesión de trabajo de la segunda fase. La idea inicial era de utilizar cada detección que hiciera el sensor ultrasónico para encender alguna luz colocada en el público, pero finalmente Julia decidió que las tiras de LEDs fueran parte del escenario y que ella las pudiera utilizar para delimitar su espacio de baile.



*Figura 4.7: Placa con Arduino Nano, sensor ultrasónico y tiras de LEDs direccionable*



*Figura 4.8: Prueba funcionamiento placa y tiras de LEDs*

- *Placa con NodeMCU y MPU 9250*

Se trata de una placa similar a la protoboard nº1 mostrada en la primera sesión de la segunda fase. Se decidió seguir usando un acelerómetro, para que se pudieran simular las funciones que iba a tener el módulo inalámbrico que estaba en fase de desarrollo y porque desde el primer día de colaboración se pensaba que era importante poder detectar los movimientos de Julia a la hora de bailar. La novedad en esta placa, respecto a las anteriores, fue la sustitución de la placa de Arduino UNO por la placa NodeMCU, y la eliminación del zumbador y las tiras de LEDs, esto último por elección personal de la bailarina a la que no le interesaba ningún tipo de sonido ni luces por el cuerpo en su actuación.

La razón por la que se decidió sustituir la placa de Arduino UNO por la placa NodeMCU, es que se pensó que se podía generar algún efecto en el escenario a través de la detección de movimiento por parte del acelerómetro. Lo que permite la placa NodeMCU respecto de la placa Arduino UNO es reducir notablemente el tamaño de la placa, cosa de relevante importancia ya que tiene que molestar lo menos posible a la hora de bailar, y además permite comunicar con otros dispositivos a través de comunicación WiFi. Por tanto, la idea inicial era que la placa estuviera colocada sobre el brazo de Julia y durante el baile una vez que el acelerómetro detectara movimiento en cualquiera de los tres ejes, enviara los datos por I2C al NodeMCU y que este, según el valor de estos datos, se comunicara con otro dispositivo para hacer que se produjera algún efecto en el escenario.

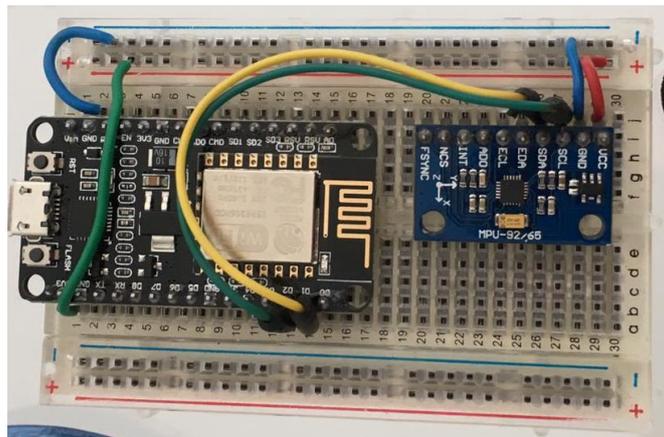


Figura 4.9: Placa con NodeMCU y MPU 9250

Se realizó esta placa de tal manera que se acercase lo máximo posible al funcionamiento del módulo inalámbrico que estaba todavía en fase de desarrollo, ya que permitía comunicación WiFi y un sensor para detectar movimientos. Además, se eligió controlarla con el NodeMCU porque su microcontrolador pertenece a la misma familia que el microcontrolador elegido para el módulo y por tanto el código realizado para esta placa se habría podido utilizar también para él.

#### 4.2.3 Tercera fase: preparación espectáculo final

Una vez decididas las placas a utilizar para la actuación final, comenzó la tercera fase de la colaboración. Esta fase consistió, en primer lugar, en desarrollar el firmware de las dos protoboards elegidas y luego en varias sesiones de trabajo en Etopia para probar y ajustar los efectos que se iban a incluir en la actuación final.

De acuerdo con lo decidido con Julia, el firmware de las dos placas elegidas se desarrolló de la siguiente manera:

- Placa con Arduino Nano, sensor ultrasónico y tiras de LEDs RGB direccionables

Funcionaba de manera que se pudieran activar dos tiras de LEDs cuando el sensor ultrasónico detectaba que había alguien a menos de 100 cm. Las tiras de LEDs se programaron para que se

encendieran de color verde. El programa se desarrolló de forma que el sensor fuera capaz de realizar únicamente tres detecciones:

1. Primera detección: la primera vez que se acercara Julia al sensor, se encendería una tira de LEDs que habría permanecido encendida durante toda la actuación, no se habrían atendido más detecciones durante los siguientes 30 s.
2. Segunda detección: la segunda vez que se acercara Julia al sensor, se habría encendido la segunda tira de LEDs, no se habrían atendido más detecciones durante los siguientes 30 s.
3. Tercera detección: la tercera vez que se acercara Julia al sensor, las dos tiras de LEDs habrían empezado a parpadear, a una frecuencia de 0.5 s, a partir de ese momento no se habrían atendido más detecciones.

El código desarrollado para esta placa se encuentra en el Anexo 6.5.

- [Placa con NodeMCU y MPU 9250](#)

El sentido conceptual de lo que se ha implementado tiene que ver con lo que Julia quería expresar en su espectáculo. Ella quería mostrar como la tecnología, a veces, afecta al comportamiento del cuerpo humano. Medir sus movimientos, almacenarlos, compartirlos, enviarlos a la nube y representarlos gráficamente, es una extensión del espectáculo expresada en forma tecnológica para enseñar como, a través de algo externo, es posible controlar el comportamiento del cuerpo.

Por tanto, el firmware se diseñó de manera que el sensor MPU 9250, cuando detectara movimiento en el eje X, midiera el valor de la aceleración en esa dirección, la pasara por comunicación I2C al NodeMCU que estaba conectado a la red WiFi de Etopia y mandaba los datos a una página web de Adafruit ([io.adafruit.com](http://io.adafruit.com)) donde era posible almacenar los datos y dibujar gráficas en tiempo real.

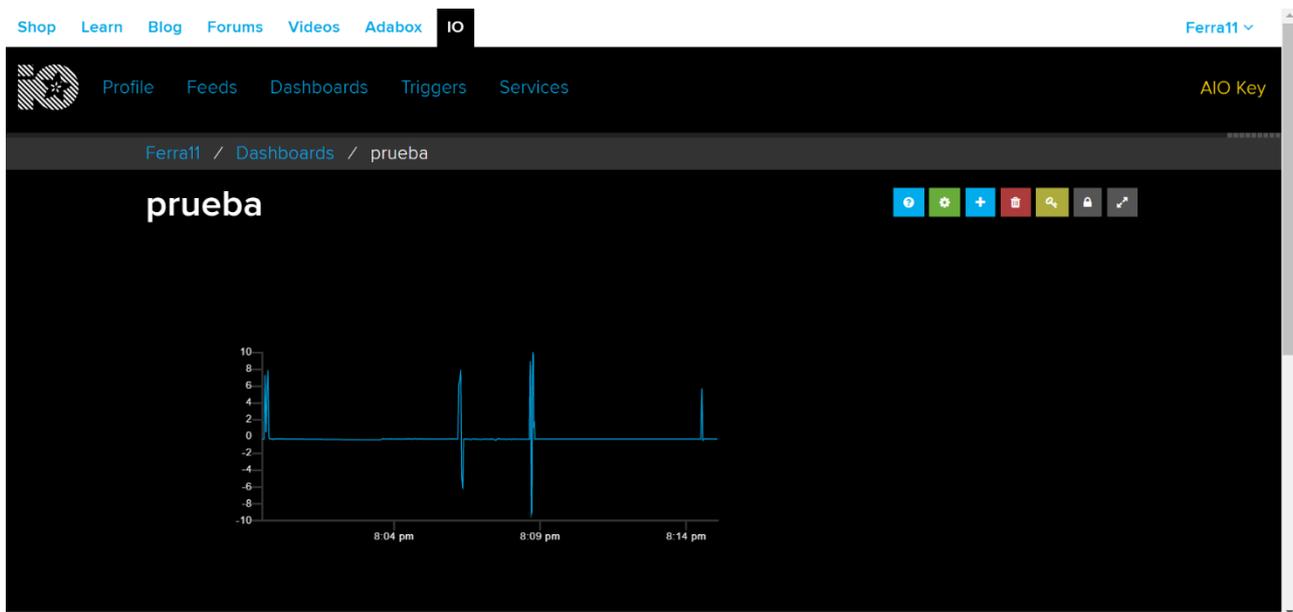


Figura 4.10: Pagina web [io.adafruit.com](http://io.adafruit.com)

Como explicado anteriormente, el firmware realizado para esta placa se habría podido utilizar para el módulo inalámbrico desarrollado para la colaboración, debido a que los microcontroladores de los dos sistemas pertenecen a la misma familia y al mismo fabricante. El código desarrollado para esta placa se encuentra en el Anexo 6.6.

#### *4.2.3.1 Primera sesión de trabajo*

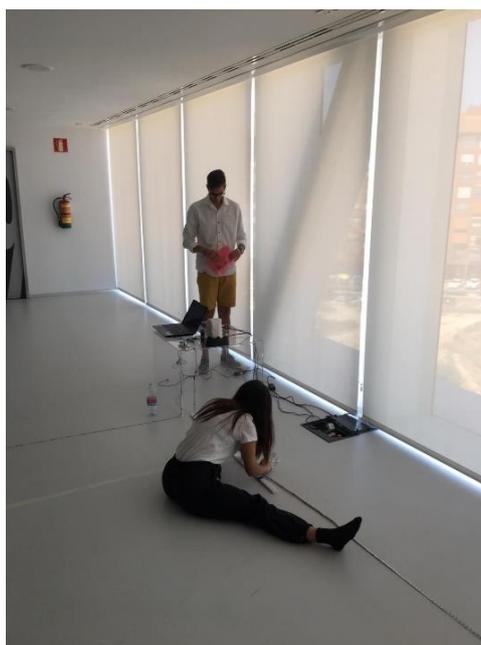
Después de haber desarrollado el firmware para las dos placas elegidas para la actuación final, se realizó la primera sesión de trabajo de esta fase para probar los movimientos y ajustar los tiempos para realizar las distintas detecciones con el sensor ultrasónico. La placa del Arduino Nano funcionaba a la perfección, mientras que la del NodeMCU daba problemas de conexión, se conectaba a la red WiFi de Etopia pero no conseguía a conectarse a la página web [io.adafruit.com](http://io.adafruit.com). Este problema no se pudo solucionar en el momento aun probando a cambiar la red WiFi a la que conectarse.

#### *4.2.3.2 Segunda sesión de trabajo*

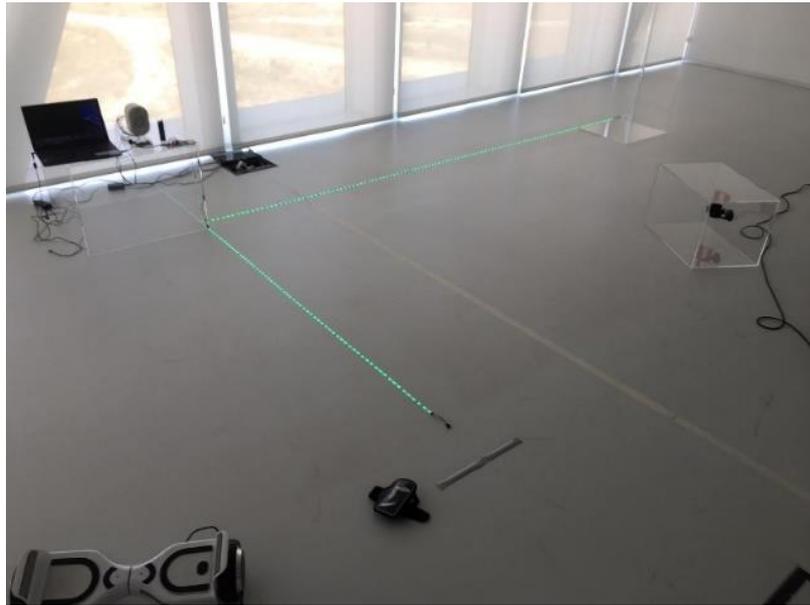
La segunda sesión de trabajo de esta fase se realizó para poder solucionar el problema de comunicación detectado en la primera sesión. Se intentó cambiar el módulo NodeMCU pero sin éxito. Se probó cambiando la red WiFi a la que conectarse, pero también sin éxito. Al final se decidió modificar el firmware inicial, partiendo del ejemplo oficial de las librerías de Adrafruit para el NodeMCU. Haciendo esto finalmente la comunicación comenzó a funcionar y ya se podían enviar los datos detectados con el MPU 9250 a la página web [io.adafruit.com](http://io.adafruit.com).

#### *4.2.3.3 Tercera sesión de trabajo*

La tercera y última sesión de esta fase, consistió en un ensayo general para probar la entera actuación y se realizó una semana antes respecto a cuando estaba programado el espectáculo final.



*Figura 4.11: Montaje del escenario para el ensayo general*



*Figura 4.12: Escenario del ensayo general*

#### 4.2.4 Cuarta fase: Actuación Festival Trayectos

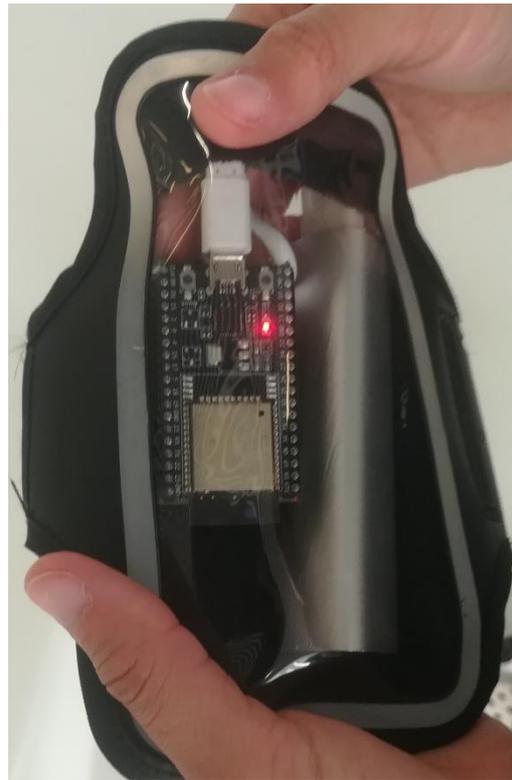
Después de todo el trabajo desarrollado hasta el momento llegó, la cuarta y última fase de la colaboración, la actuación final de Julia en el festival de Trayectos 2019.

Por razones que se explicaran posteriormente, el módulo inalámbrico inicialmente previsto no podía estar listo para la actuación final. Por tanto, fue necesario elaborar un plan de contingencia que consistía en desarrollar un nuevo módulo con menos funcionalidades que el original esperando que pudiera estar listo el día de la actuación.

Se hizo todo lo posible para tener preparado el nuevo módulo inalámbrico. La llegada de los componentes en los últimos días previos al comienzo del festival hizo que se pudiera montar un ejemplar que se entregó a la bailarina el mismo día de su actuación. Se planteó a Julia la opción de utilizar el módulo para la actuación y se realizaron varias pruebas para comprobar su correcto funcionamiento.



(a)



(b)

*Figura 4.13: (a) Pruebas para comprobar el funcionamiento del módulo (b) Foto del módulo*

Finalmente, Julia decidió utilizar el montaje realizado en protoboard debido que se sentía más segura al ser la placa con la que había realizado todos los ensayos.

Entonces, tal y como se había decidido entre la segunda y la tercera fase de la colaboración (apartados 4.2.2 y 4.2.3), la intervención electrónica para el espectáculo consistía en dos placas que fueron utilizadas por Julia en momentos diferentes de la actuación.



*Figura 4.14: Actuación final de Julia en el festival de Trayectos 2019*

La actuación se dividía en dos partes. En la primera parte la bailarina danzaba sobre un hoverboard, teniendo sobre su brazo izquierdo la placa con NodeMCU y acelerómetro. Como se ha explicado más detenidamente en el apartado 4.2.3, su función era la de detectar movimientos, registrar los datos y enviarlos mediante WiFi a una página web que recogía los datos y los dibujaba sobre una gráfica. El público podía observar la gráfica a través de una pantalla puesta en el escenario.

Para que la placa pudiese sostenerse sobre el brazo de Julia, se decidió insertarla en un brazaletes y de esta forma asegurar que estuviera bien protegida. La forma elegida de alimentar la placa fue a través de una batería portátil conectada al NodeMCU mediante su conector mini-USB. Se optó por este tipo de fuente de alimentación, en vez de una pila de 9V, por tener una conexión más segura a la hora de aguantar los movimientos bruscos de la bailarina.



*Figura 4.15: Placa utilizada en la actuación insertada en el brazalete*



*Figura 4.16: Primera parte de la actuación final de Julia*

En la segunda parte del espectáculo, Julia bajaba del hoverboard dejándolo a un lado del escenario junto con el brazaletes y empezaba a bailar libremente en el espacio delimitado por las tiras de LEDs colocadas en el suelo.



*Figura 4.17: Segunda parte de la actuación final de Julia*

La placa con el sensor ultrasónico y las tiras de LEDs RGB direccionables controlada por el Arduino Nano estaba colocada sobre un cubo de plástico transparente a lado de un ordenador cuya función era puramente estética. Durante el baile la bailarina pasaba por delante del sensor ultrasónico tres veces en tres diferentes momentos para que sucedieran los eventos decididos en las sesiones de trabajo comentadas previamente en el apartado 4.2.3.

La actuación concluyó con la bailarina subiendo otra vez sobre el hoverboard y su salida del escenario dejando al público con una sensación de incertidumbre.

## 5. Desarrollo de los sistemas electrónicos

Como comentado anteriormente, para la colaboración se ha tratado de diseñar y programar un módulo inalámbrico para poder utilizarlo en el festival de danza contemporánea Trayectos.

La idea inicial, era que la parte de diseño del esquemático y PCB del módulo fuera objeto del TFG de Claudia Gonzalo, mientras que la parte de diseño y programación del firmware fuera objeto de este TFG.

Debido a retrasos inesperados con el orden de componentes y ante la incertidumbre de que el módulo diseñado por Claudia podría no haber estado fabricado antes del comienzo del festival, se decidió elaborar un plan de contingencia que consistía en la creación de un nuevo módulo, con

menos funcionalidades que el módulo diseñado anteriormente, pero que pudiera estar listo para el comienzo del festival.

En esta parte de la memoria se describe de forma detallada cómo ha evolucionado la creación de este módulo y se van a comentar los distintos componentes elegidos para poderlo realizar.

También se va a explicar el diseño y estructura del firmware y cómo ha evolucionado el estudio sobre la elección del lenguaje de programación, punto que ha sido clave para adaptar la programación a las necesidades funcionales del nuevo módulo.

## 5.1 Hardware

En esta parte se va a mostrar de manera resumida en qué ha consistido el trabajo de Claudia Gonzalo para desarrollar el módulo inalámbrico que inicialmente se iba a utilizar para el festival.

Además, se van a explicar las decisiones tomadas respecto a la creación del módulo del plan de contingencia: elección de componentes para componer el esquemático, decisiones de colocación en la PCB y el consecuente ruteo de las pistas.

### 5.1.1 Módulo diseñado por Claudia Gonzalo

El módulo desarrollado tenía que ser capaz de detectar movimientos y actuar en función de estos. Se decidió que el módulo tuviera que tener cierta versatilidad, por ello se han elegido entradas y salidas de propósito general. A raíz de esto, para que fuera todavía más versátil, se ha dotado de un microcontrolador con conexión WiFi.

El módulo tenía que ser portable y de menor tamaño posible, pero sin renunciar a esa versatilidad.

Los componentes principales de este módulo son:

- ESP-WROOM-32:
  - Es el microcontrolador utilizado para controlar el sistema, elegido por su capacidad para comunicarse vía wifi.
  - Tiene 34 GPIO (General Purpose Input/Output) de diferentes tipos según su programación: digitales, ADC (Analog to Digital Converter), DAC (Digital to Analog Converter), capacitivas táctiles, para sensores Hall...
  - Tiene incorporados protocolos: SPI, I2C y UART. Se ha utilizado el protocolo de comunicación I2C para recibir la información que proviene del sensor bno055.
  - Gracias al bootloader se puede programar en diferentes lenguajes siendo los más destacados MicroPython y Arduino [18].
- Sensores:
  - Un sensor BNO055. Se trata de un sensor inercial absoluto de 9 ejes, acelerómetro, giróscopo y magnetómetro [19]. Se utiliza para detectar el movimiento de la bailarina.
  - Cinco sensores analógicos de flexión con la idea de poder sensorizar los movimientos de partes del cuerpo.
  - Un sensor analógico de pulso.

- Actuadores:
  - Tres salidas de propósito general controladas mediante transistores mofset y con diodos en antiparalelo por si los actuadores son cargas inductivas.
  - Un conector para poder controlar una tira de LEDs direccionable.

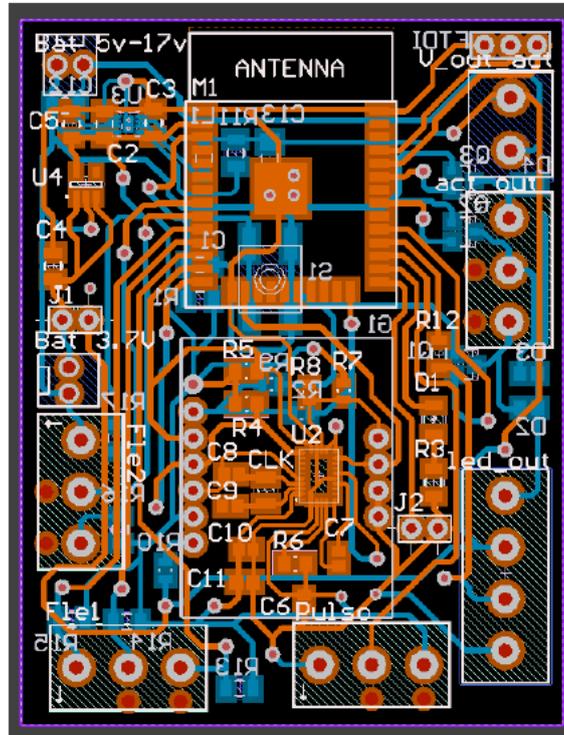


Figura 5.1: PCB del módulo inalámbrico diseñado por Claudia Gonzalo

### 5.1.2 Plan de contingencia

Debido a las circunstancias, como ya se ha explicado en el apartado anterior, fue necesario elaborar un plan de contingencia que se decidió incluir también como objeto de este TFG. Se tomó la decisión de crear un módulo inalámbrico más simple pero acorde a las necesidades.

Se procede a explicar cómo ha evolucionado el diseño de esquemático y PCB y la consecuente elección de componentes.

#### 5.1.2.1 Diseño del esquemático

El esquemático se divide en tres bloques: control, sensores y actuadores, alimentación. Se va a proceder a explicar cada bloque, por qué componentes está compuesto y cuál es su función en el circuito.

## Bloque de Control

El bloque de control se compone por la placa programable ESP32-DevKitC que se encarga de la comunicación WiFi y de controlar los varios sensores y actuadores del circuito.

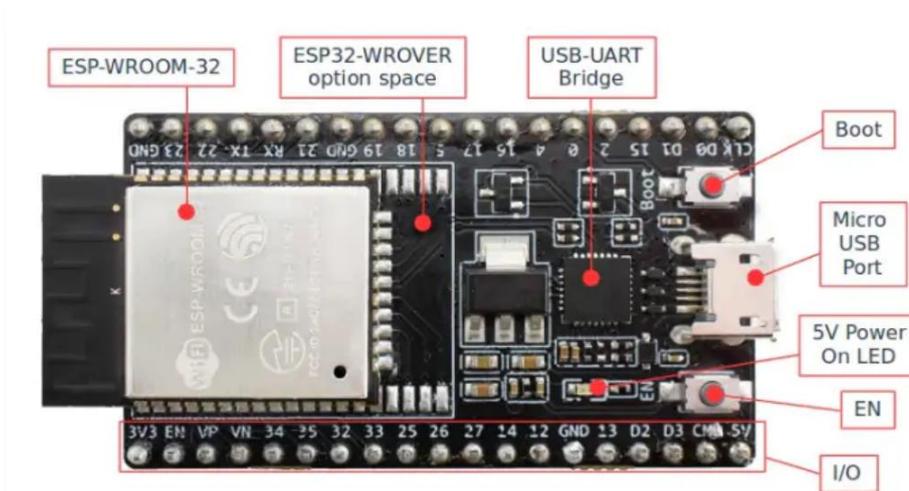


Figura 5.2: ESP32-DevKitC

El ESP32-DevKitC es una placa de desarrollo de Espressif Systems. Está basada en el microcontrolador ESP-WROOM-32 que permite tener comunicación WiFi y Bluetooth [20].

Tiene 32 GPIO (General Purpose Input/Output) de diferentes tipos según su programación: PWM, ADC, DAC etc. Por lo tanto, permite controlar diferentes tipos de periféricos [20].

Tiene incorporados protocolos: I2C, SPI, I2S y UART. Se utiliza el I2C para recibir información del sensor MPU 9250 [20].

Además, posee un botón de reset (EN) y una Micro USB Port para suministrar alimentación a la placa y a su vez permite la comunicación entre un ordenador y el ESP-WROOM-32. Por tanto, la forma de programarlo es mediante un cable Micro USB [20]. Los posibles lenguajes de programación se comentarán en el apartado 5.2.2.

También tiene la posibilidad de poderse alimentar por una fuente de alimentación externa a través de las patillas 5V y GND o 3.3V y GND [20]. En el circuito, se elige alimentar la placa a través de las patillas 5V y GND, debido a que la patilla 3.3V se utiliza para alimentar el sensor MPU 9250.

La decisión de elegir este tipo de placa se debe a que utiliza el mismo microcontrolador que el módulo desarrollado por Claudia Gonzalo, por tanto, todo el trabajo sobre el diseño del firmware realizado hasta el momento no habría sido desaprovechado.

Para poder insertarlo en el módulo inalámbrico, se han elegido 2 conectores hembra de 20 pines.

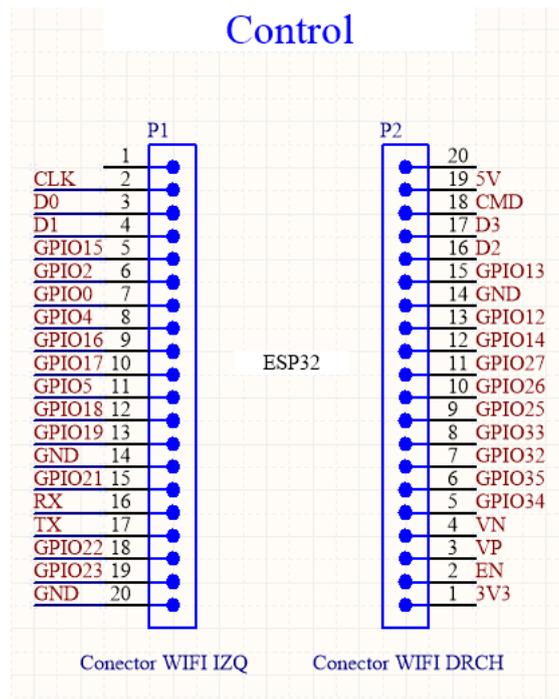


Figura 5.3: Esquemático bloque de control realizado con la herramienta CircuitMaker

#### Bloque de Sensores y Actuadores

El bloque de sensores y actuadores se puede dividir a su vez en tres bloques diferentes: bloque para el control de una tira de LEDs RGB direccionable y una entrada analógica, un bloque de tres salidas de propósito general y un sensor para detectar movimiento.

#### Bloque de control de tira de LEDs RGB direccionable y entrada analógica

Este bloque se compone por dos conectores macho J1 y J2 de tres pines cada uno. La función en el circuito de J1 es la de permitir conectar al módulo inalámbrico una tira de LEDs RGB direccionable, por tanto, dos pines se utilizan para suministrar alimentación a la tira de LEDs, mientras que el tercero se conecta al GPIO27 del ESP32-DevKitC que se configura como salida digital y sirve para controlar el encendido y el color de los LEDs de la tira.

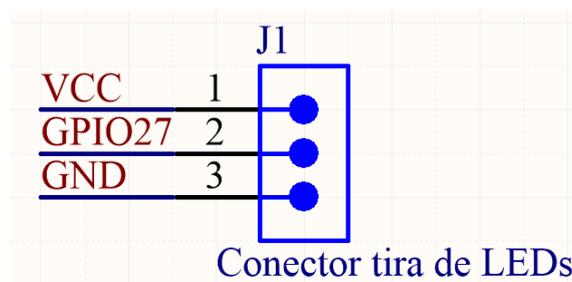


Figura 5.4: Esquemático tira de LEDs direccionable realizado con la herramienta CircuitMaker

- Tira de LEDs RGB direccionable:

Son tiras de LEDs de tipo RGB que poseen, por cada LED, un circuito integrado cuya función es permitir que sea programable.

Se ha elegido este tipo de tira de LEDs frente a las que no son direccionables porque, a través de su programación, es posible controlar el encendido y el color de cada LED de la tira. Además, tratándose de tiras digitales no se necesita añadir transistores para su control.

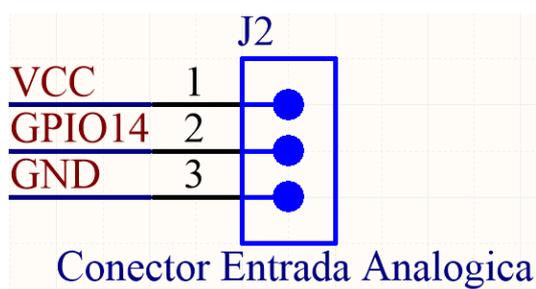
Su tensión de alimentación es de 12V, y se controla mediante un pin digital de salida del microcontrolador.

El color se elige mediante el modelo cromático RGB, asignando a cada color primario un valor entre 0 (nada) y 255 (todo). Modificando estos valores se pueden obtener colores diferentes.



*Figura 5.5: Tira de LEDs RGB direccionable*

Por otra parte, con J2 se quiere proporcionar al módulo inalámbrico una entrada de propósito general para poderle conectar algún sensor analógico como podrían ser un sensor de pulso o de flexión. Dos pines se utilizan para suministrar alimentación y el tercero se conecta al GPIO14 del ESP32-DevKitC que se configura como entrada analógica.



*Figura 5.6: Esquemático entrada analógica realizado con la herramienta CircuitMaker*

#### *Bloque de tres salidas de propósito general*

Se han incluido en el circuito tres salidas de propósito general controladas mediante transistores mofset. Se trata de tres circuitos iguales cuya función de cada uno es la de poder suministrar alimentación a una carga a través de la activación de una salida digital. Los tres circuitos se componen cada uno de un conector macho de dos pines. Un pin está conectado a VCC mientras

que el otro al terminal de drenaje de un transistor mosfet, cuya función es la de abrir y cerrar el camino a masa de la carga controlado por una salida digital del ESP32-DevKitC. Por tanto, el funcionamiento del circuito es el siguiente: conectando los otros dos terminales del transistor, la puerta a un GPIO del ESP32-DevKitC, que se configura como salida digital, y la fuente a masa. Cuando el GPIO se encuentra en estado alto, el transistor entrará en saturación, cerrará el camino a masa y la carga quedará alimentada, mientras que cuando el GPIO esté en estado bajo, el transistor entrará en corte y la carga quedará sin alimentación. La justificación de los GPIOs del ESP32-DevKitC elegidos se encuentra en el apartado 5.1.2.2.

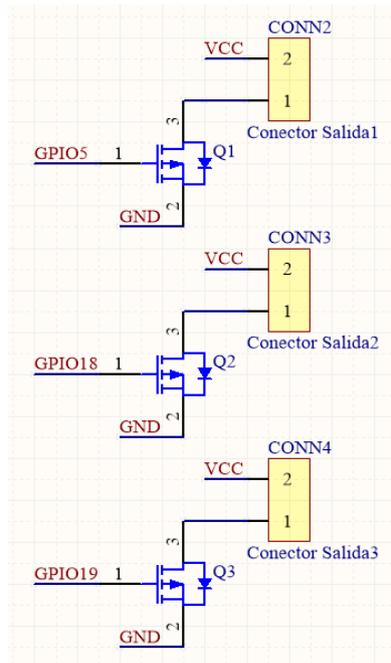


Figura 5.7: Esquemático bloque de tres salidas digitales transistorizadas realizado con la herramienta CircuitMaker

- Elección del transistor:

Se ha elegido un transistor de tipo mosfet para poderlo controlar en tensión sin necesitar resistencias auxiliares.

Debido a que la tensión de salida de los pines digitales del ESP32-DevKitC es de 3.3V, se necesitaba encontrar un transistor que tuviera una tensión umbral de puerta a fuente menor que esa tensión. El transistor elegido es el DMG230UK-7 de la marca Diodes Incorporated. Se trata de un transistor de canal N. Se ha elegido en base a las curvas características incluidas en su Datasheet [21]:

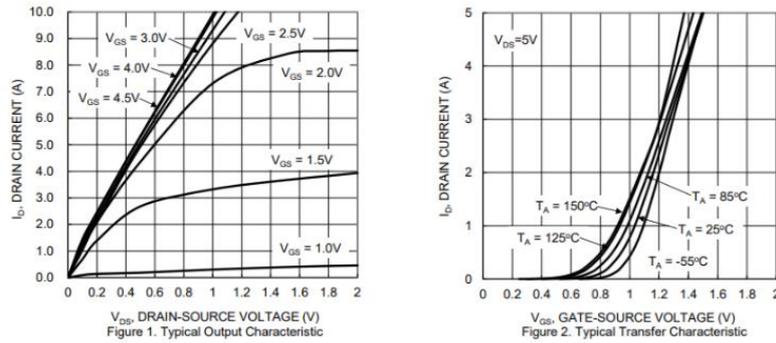


Figura 5.8: Curvas características del transistor DMG230UK-7

Como podemos observar en la gráfica de la izquierda, para una  $V_{GS}$  superior a 3 V el transistor es capaz de conducir varios Amperios, dependiendo de la carga, lo cual satisface las necesidades de activación.

Además se ha elegido este transistor por ser de tipo SMD, y debido a su reducido tamaño resultaba muy útil ya que la placa tenía que ser lo más pequeña posible.

#### Sensor para detectar movimiento

La función principal por la que se empezó a desarrollar el módulo inalámbrico era que fuera capaz de detectar movimientos de la bailarina a la hora de actuar. Por ello, se ha querido incorporar a este módulo el sensor MPU 9250.

El sensor MPU 9250 es un IMU de 9DOF de nueva generación fabricado por Invensense. Internamente incorpora en un mismo integrado un acelerómetro de 3DOF, un giroscopio de 3DOF, y un magnetómetro 3DOF [22].

Una de las razones por las que se ha elegido este tipo de componente es que la comunicación se puede realizar por I2C, por lo que es posible que se comunique con el microcontrolador elegido.

Además, la ventaja que tiene este sensor frente a otros IMUs con los mismos grados de libertad (DOF), es que incorpora un procesador interno capaz de realizar las mediciones de los tres sensores de forma simultánea. En otros componentes similares esta operación se debe realizar externamente, lo que supone una pérdida de rendimiento [22].

La tensión de alimentación puede variar entre 2.4 V a 5 V, y el consumo máximo es de 3.5 mA, con todos los sensores activados [22].

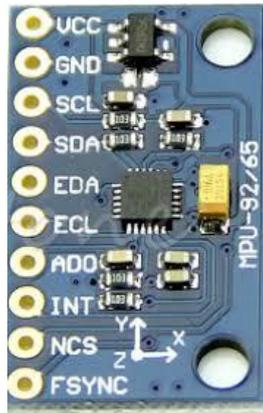


Figura 5.9: Sensor MPU 9250

Se conecta al circuito a través un conector hembra de diez pines. Para este proyecto, solamente se van a utilizar los cuatro primeros pines: los de alimentación y los necesarios para la comunicación con el microcontrolador. Para poderlo alimentar, el sensor se conecta a las patillas 3V3 y GND del ESP32-DevKitC que suministra una tensión de 3.3V. Mientras que los pines SCL y SDA se conectan respectivamente al GPIO22 y GPIO21, y sirven para establecer la comunicación I2C entre el ESP32-DevKitC y el sensor MPU 9250.

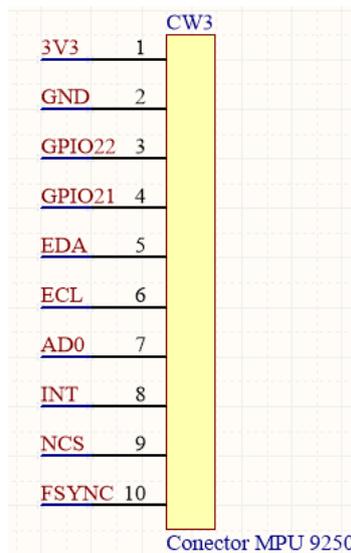


Figura 5.10: Esquemático sensor MPU 9250 realizado con la herramienta CircuitMaker

#### Bloque de Alimentación

La función de este bloque es la de suministrar la tensión y corriente necesarias a todo el circuito. El componente principal que marca un nivel de tensión necesario, ya que sin el módulo no podría funcionar, es el ESP32-DevKitC que requiere una tensión de 5V. El resto de cargas no afectan al funcionamiento del módulo si no están conectadas.

Debido a este aspecto es posible alimentar el módulo de dos maneras diferentes:

### 1. Primera opción:

Se trata de la opción principal. Es la recomendada cuando se quieren utilizar todos los sensores y actuadores del módulo y por tanto se suministrará alimentación al entero circuito. Para poder conectar la fuente de alimentación se ha elegido un conector macho de dos pines, a los cuales se conecta un regulador conmutado de tensión de 5V para poder alimentar así el ESP32-DevKitC.

- Regulador TR10S05:

El regulador elegido es el TR10S05 del fabricante XP Power. Este componente se ha elegido principalmente por tener una corriente de salida acorde con las necesidades del ESP32-DevKitC, por su alta disipación de potencia y por su tamaño reducido, cosa de relevante importancia para la placa que se ha desarrollado.



Figura 5.11: Regulador TR10S05

El fabricante en su datasheet [23], para que el regulador tenga un correcto funcionamiento, aconseja conectar tanto a la entrada como a la salida dos condensadores de filtrado. El condensador a la entrada es de 22 uF y es altamente recomendado, mientras que el que está conectado a la salida, de 47 uF, es opcional y sirve para mejorar la respuesta transitoria.

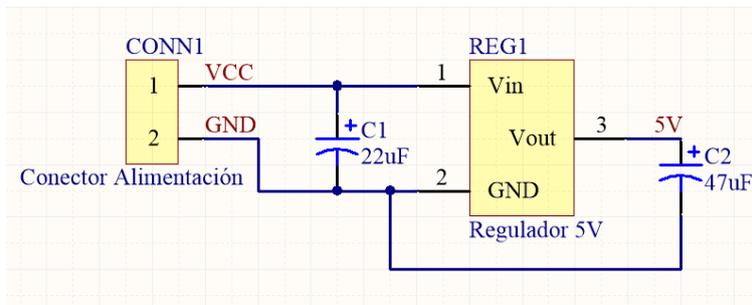


Figura 5.12: Esquemático bloque de alimentación realizado con la herramienta CircuitMaker

La tensión de alimentación puede oscilar entre 7V y 28V que es el rango de entrada admitido por el regulador elegido. Bajo este aspecto, hay que tener en cuenta que tanto el conector

de la tira de LEDs, el conector de la entrada analógica y los tres conectores correspondientes a las salidas digitales del módulo, se alimentan directamente desde la fuente de alimentación.

Por tanto, dependiendo de que cargas se conecten al módulo, habrá que elegir adecuadamente el nivel de tensión con el cual alimentar el circuito para evitar que se pueda dañar alguna.

## 2. Segunda opción:

Esta segunda opción permite utilizar únicamente el ESP32-DevKitC y el sensor MPU 9250, por lo tanto, no se podría conectar ningún tipo de carga al módulo inalámbrico. La forma de hacerlo sería alimentando el ESP32-DevKitC a través de su puerto Micro USB. Puede resultar útil, cuando fuera solo necesario recoger datos con el sensor y enviarlos vía WiFi.

El esquemático completo queda de la siguiente manera:

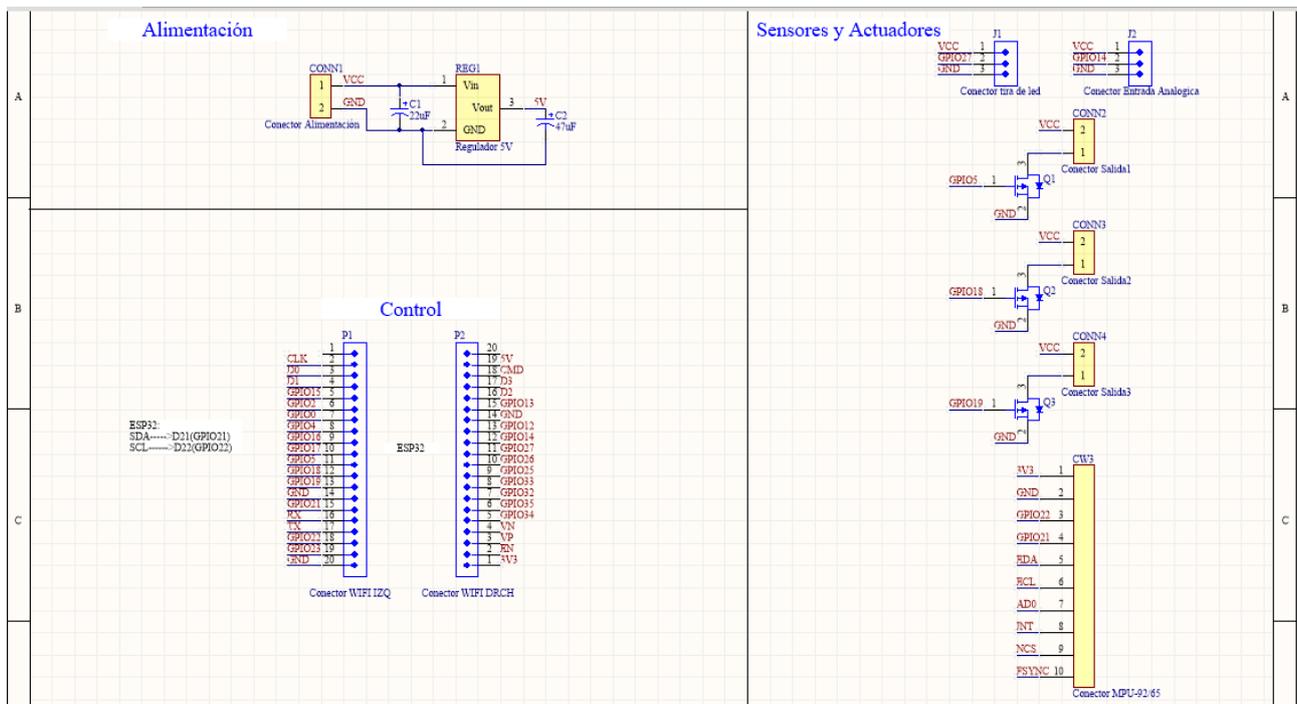


Figura 5.13: Esquemático módulo inalámbrico del plan de contingencia realizado con la herramienta CircuitMaker

### 5.1.2.2 Diseño PCB

La PCB de este proyecto se ha diseñado para que fuera lo más pequeña y lo más compacta posible, ya que se tenía que colocar sobre el cuerpo de la bailarina y por tanto era importante que molestara lo mínimo posible y que no entorpeciera sus movimientos a la hora de bailar. Los componentes y las pistas se han colocado tanto en la cara top como en la cara bottom para poder reducir al máximo el tamaño de la placa.

Las dimensiones de la placa se han establecido en base a las medidas del ESP32-DevKitC, ya que se trata del componente de mayor tamaño. El resto de componentes se han colocado de manera que la PCB no excediera del tamaño establecido.

La PCB se diseñó siguiendo unas normas de ruteo:

- Ancho de pista de 0.6 mm para señales generales.
- Ancho de pista de 0.8 mm para señales críticas (Alimentación).
- Pistas lo más rectas y cortas posible.
- Hacer esquinas de 45° y no de 90°.
- Todos los pads circulares para facilitar el trabajo de la máquina del laboratorio.
- Medidas de los pads y agujeros normalizados a los taladros del laboratorio.

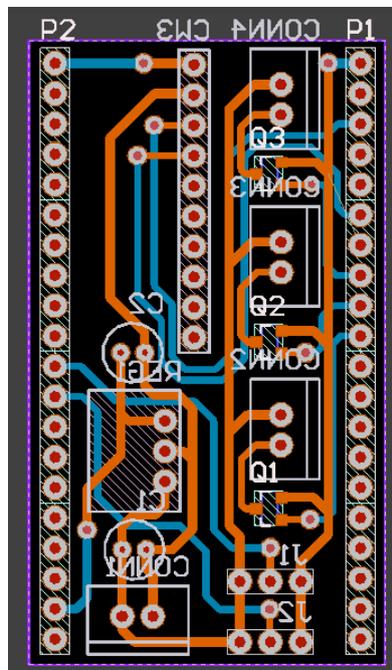


Figura 5.14: Ruteo completo sin planos de masa realizado con la herramienta CircuitMaker

### Cara Top

En la cara Top se han colocado, los dos conectores hembra P1 y P2 que sirven para insertar el ESP32-DevKitC. De este modo, ocupa toda la superficie de esta cara imposibilitando la colocación de otros componentes de tipo THD.

El área restante de la cara Top que se encuentra entre los conectores P1 y P2, se ha utilizado para el ruteo de las pistas de los componentes de la cara Bottom y para la colocación de los tres transistores mosfet Q1, Q2 y Q3 que, al tratarse de componentes de tipo SMD, por su reducido tamaño, se podían soldar debajo del ESP32-DevKitC.

Cabe comentar que los pines del ESP32-DevKitC para poder controlar los transistores se han elegido para que las pistas fueran lo más cortas posible. Siendo finalmente: el GPIO5 para Q1, el GPIO18 para Q2 y el GPIO19 para Q3.

#### Cara Bottom

El resto de componentes, siendo todos de tipo THD, se han colocado en la cara Bottom. Los componentes se han colocado en bloques funcionales: alimentación, sensor de movimiento y actuadores, para permitir que las pistas fueran lo más cortas posible.



*Figura 5.15: Bloques funcionales*

La colocación de los componentes por bloques funcionales también ayuda a minimizar el ruido electromagnético que es un factor importante a tener en cuenta a la hora de diseñar una PCB [24]. Además, se ha prestado particular atención a cerrar los bucles de retorno de las corrientes mediante planos de masa tanto en la cara Top como en la cara Bottom.

Bajo este último aspecto, como podemos observar en la figura 5.18 (a), la antena del ESP32-DevKitC necesaria para la comunicación WiFi, queda fuera de los límites de la placa para evitar cualquier tipo de interferencia.

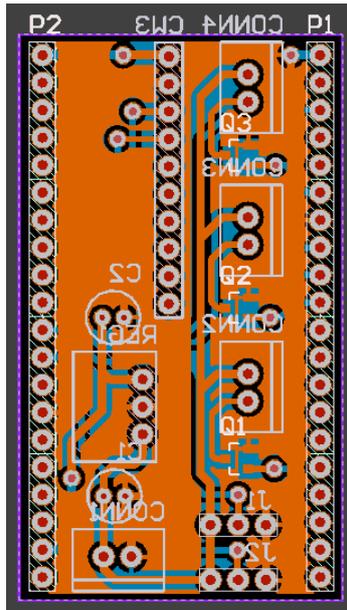
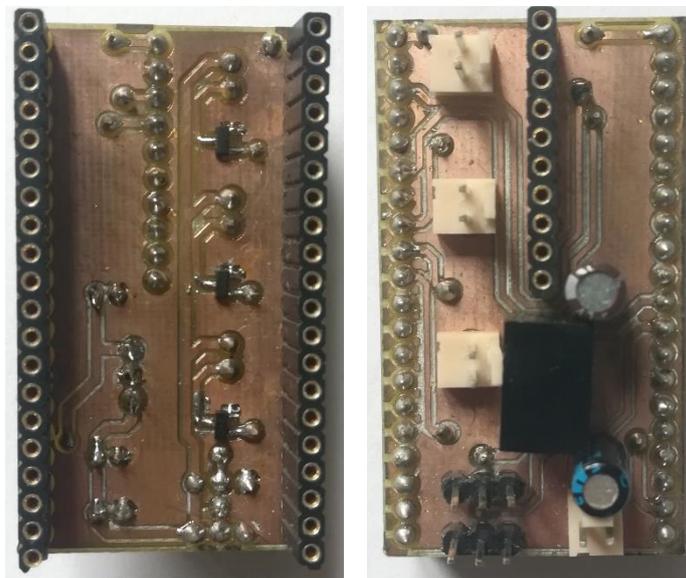


Figura 5.16: Ruteo completo con planos de masa realizado con la herramienta CircuitMaker

Se fabricaron y montaron cuatro unidades, aquí se muestran las imágenes relativas a la cara Top y a la cara Bottom:



(a)

(b)

Figura 5.17: (a) Cara Top (b) Cara Bottom

Quedando el módulo montado de la siguiente manera:

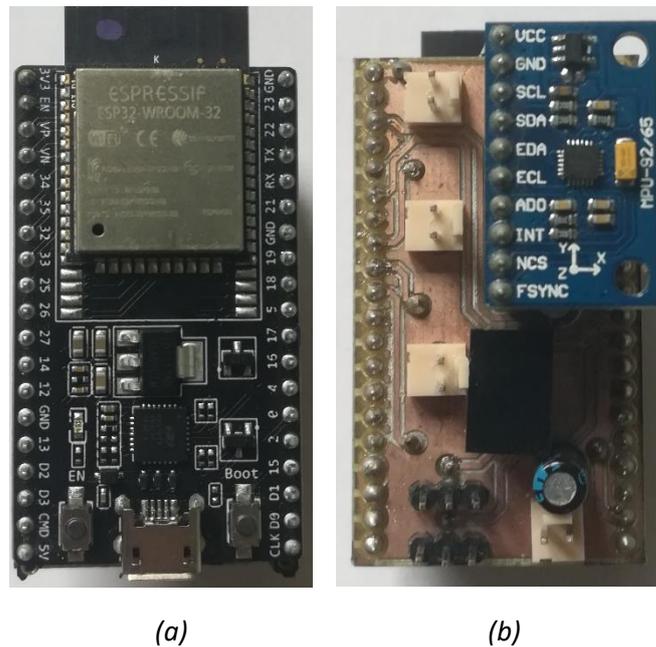


Figura 5.18: (a) Cara Top módulo montado (b) Cara Bottom módulo montado

## 5.2 Firmware

Aunque el festival de Trayectos ya había terminado, se decidió seguir con el trabajo ya que el módulo desarrollado, dispone de una gran potencia para sensorizar movimientos y la posibilidad de comunicación WiFi ofrece mucha versatilidad para cualquier tipo de aplicación. Por tanto, se decidió diseñar un firmware que pudiese aprovechar a pleno las posibilidades del módulo.

### 5.2.1 Comunicaciones

Para llegar a entender plenamente como se ha desarrollado el firmware del módulo inalámbrico, hay que conocer previamente como funciona y de que se compone una red de comunicaciones WiFi ya que el módulo se ha programado para que sea capaz de comunicarse con otros dispositivos a través de este tipo de comunicaciones.

Como en todo proceso de comunicación, se requiere de un emisor, un mensaje, un medio y un receptor. Una red de comunicaciones es un conjunto de dispositivos que pueden funcionar como emisores o receptores y están interconectados para compartir información y recursos.

Los conceptos de red WiFi, modelo TCP/IP y sistema de sockets necesarios para entender como se ha realizado la programación del módulo se explican en el Anexo 4.

## 5.2.2 Investigación sobre lenguajes de programación

Una de las decisiones a tomar para comenzar a desarrollar el firmware fue la de elegir un adecuado lenguaje de programación para poder aprovechar al máximo las funciones del microcontrolador utilizado.

Acorde con lo explicado anteriormente el microcontrolador a programar era el ESP-WROOM-32 cuyo fabricante es Epressif Systems.

Se analizan a continuación los lenguajes de programación posibles para poder discriminar los más adecuados y los que conviene descartar con criterio y proceder a la elección.

- Programación con Comandos AT:

Se trata de un tipo de programación de bajo nivel que se utiliza cuando el microcontrolador no tiene un firmware instalado. Se descarta por ser la opción más tediosa.

- Programación con el SDK original de Expressif Systems en lenguaje C++:

Se trata del software desarrollado por el propio fabricante para poder compilar programas sobre los microcontroladores que fabrica. Era una opción adecuada, pero con el inconveniente de que es un software de pago, por tanto quedó descartada.

- Programación por scripts de LUA:

La ventaja de este tipo de programación es que permite hacer programas relativamente complejos con pocas líneas de código.

El inconveniente es que el micro tiene muy poca memoria de programa disponible para este tipo de programación y en cuanto se acumulan unas pocas líneas de código da fallo de memoria. Debido a este inconveniente esta opción quedó descartada.

- Programación con lenguaje Arduino:

Se trata del lenguaje de programación más sencillo debido a la multitud de información de internet y la cantidad de librerías disponibles que simplifican la programación. Esta opción habría sido la adecuada a falta de otra mejor.

- Programación con MicroPython:

Se trata de un intérprete del lenguaje Python que se utiliza para poder programar microcontroladores.

La ventaja de MycroPython respecto al resto de lenguajes es que permite escribir códigos más sencillos, y ofrece muchas librerías que simplifican notablemente los programas para la comunicación WiFi. Este aspecto era de relevante importancia por los objetivos propuestos al principio del proyecto y ha sido el motivo por el cual se ha elegido este lenguaje de programación.

Debido a que el estudio sobre los posibles lenguajes de programación se realizó antes de que llegaran los componentes del módulo, los primeros programas se probaron sobre la placa NodeMCU ya que, como se ha explicado en apartados anteriores, tiene como microcontrolador el ESP8266 que es la

versión anterior del que se ha utilizado para el módulo inalámbrico (ESP-WROOM-32). Por tanto, el código desarrollado habría sido portable del uno al otro.

Por desgracia, debido a una serie de problemas utilizando MicroPython, reflejados en el Anexo 5.1, se tomó la decisión de cambiar de lenguaje de programación eligiendo definitivamente Arduino.

### 5.2.3 Diseño Firmware

Para el plan de contingencia se fabricaron cuatro módulos iguales. Debido a esto, el diseño del firmware ha sido orientado a que varios módulos se puedan comunicar a través de un sistema de comunicación por sockets basada en la estructura servidor-cliente. Así pues, uno será el servidor y el resto de módulos serán los clientes.

La función del servidor será la de escuchar los mensajes enviados por los clientes de manera que estos puedan, a través del mensaje, decir cuál de los actuadores del servidor quieren controlar (tira de LEDs o salidas digitales) o bien, enviarle los datos registrados por sus sensores.

También cabe la posibilidad que el módulo pueda funcionar de manera autónoma sin estar conectado a ninguna red de comunicaciones.

Para facilitar el uso y la programación del módulo para cualquier tipo de aplicación, se ha desarrollado una librería en lenguaje C/C++ compatible con Arduino IDE. Su estructura se ha diseñado para dar una utilidad de carácter general a cada componente del módulo, programando de manera debida las diferentes capas del firmware.

#### 5.2.3.1 Capas del firmware

Las tres diferentes capas del firmware son las que se encuentran entre el hardware del módulo y el código con el cual se va a programar:

- **Capa HAL:** Hardware Abstraction Layer. Se trata de la capa que agrupa los modos de comunicación entre el hardware del módulo y el microcontrolador.
- **Capa de driver:** esta capa agrupa todas las funciones que utiliza la capa de aplicación para manejar los protocolos de la capa HAL.
- **Capa de aplicación:** es la capa que corresponde al programa que implementa la aplicación del usuario y que queda compilado sobre el microcontrolador.

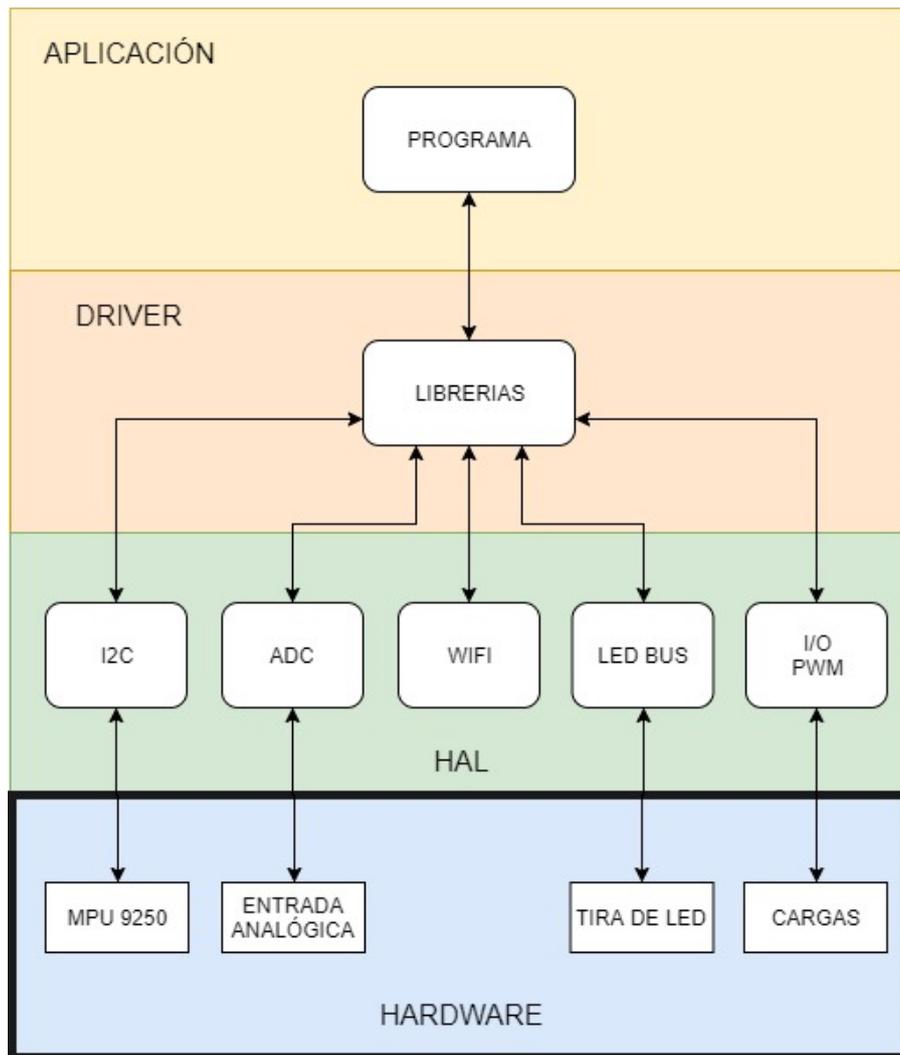


Figura 5.19: Capas del firmware

La mejor manera para entender este último concepto es a través de un ejemplo. Tomamos como ejemplo el caso del sensor MPU 9250. Si se quiere conocer el valor de aceleración en cualquiera de los tres ejes, en el programa (capa de aplicación) habrá que llamar a la función correspondiente de la librería de este componente (capa de driver). A su vez esta función hará uso del protocolo I2C (capa HAL), con el cual el sensor se comunica con el microcontrolador, para leer el valor de aceleración deseado.

### 5.2.3.2 Librería TWM

La librería TWM (Trayectos Wireless Module) se ha desarrollado para facilitar la programación y permitir cierta abstracción haciendo que el código sea más sencillo de hacer y de entender.

Se trata de una librería escrita para que se pueda utilizar en un entorno Arduino IDE. En el directorio de esta librería se incluyen los siguientes archivos:

- Un archivo TWM.h o encabezado de C, donde están declaradas todas las funciones de la librería.

- Un archivo TWM.cpp (código de C++) donde se encuentra el código de todas las funciones de la librería.

La de Arduino es una programación orientada a objetos, por tanto, cada librería compatible con este entorno de programación posee una clase que se suele denominar con el mismo nombre de la librería en la que se encuentra. Por tanto, la librería desarrollada posee una clase llamada TWM. En el interior de una clase, se declaran objetos y funciones. Para poder utilizar las funciones de una cierta librería dentro de un programa (o dentro de otra librería), hace falta incluir esa librería dentro del programa. Además, también es necesario declarar en el, el objeto perteneciente a la clase de esa librería. En el programa se realizará la llamada a la función, aplicando la función al objeto declarado.

En el archivo TWM.h, cuyo código está disponible en el Anexo 6.10.1, aparte de declarar la clase, el objeto y las funciones de la librería, se han incluido otras librerías cuyas funciones se utilizan en su interior.

- Librería Arduino: se incluye para poder utilizar todas las funciones compatibles con el entorno Arduino IDE.
- Librería MPU9250: se incluye para poder utilizar todas las funciones relativas al sensor de movimiento del módulo inalámbrico. El objeto de esta librería se denominará imu (Inercial Measurement Units).
- Librería Adafruit\_NeoPixel: se incluye para poder utilizar todas las funciones que permiten programar tiras de LEDs direccionables. El objeto de esta librería se denominará pixel.
- Librería WiFi: se incluye para poder utilizar todas las funciones relativas a la comunicación WiFi del módulo inalámbrico. Esta librería incluye en su interior otras dos librerías, WiFiServer y WiFiClient. El objeto de la librería WiFiClient se denominará client.

Algunas funciones de esta librería se han programado para que el módulo se pueda utilizar en un sistema de comunicación por sockets, donde pueda actuar tanto de servidor como de cliente. Por tanto, en el interior del archivo TWM.h, se ha declarado el protocolo de comunicación ideado para que un módulo que actúa como servidor y pueda comunicarse otro módulo que actúa como cliente y viceversa. El buffer, o mensaje, que se envía está compuesto por diez datos, cuyo significado se ilustra a continuación:

NAME	CODE	PIN	VALUE	G	R	B	AI_VALUE	S_VALUE	BUFFER SIZE
------	------	-----	-------	---	---	---	----------	---------	-------------

*Figura 5.20: Buffer de comunicación servidor-cliente*

- NAME: es el nombre del módulo que envía el mensaje, sirve para que el módulo que recibe sepa quien le está enviando el mensaje. Se trata de un dato de tipo char. Si diferentes módulos estuvieran colocados por el cuerpo, un ejemplo de nombres que se podrían asignar para reconocer cada módulo sería: 'M' = Mano o 'P' = Pierna.
- CODE: es el código que sirve para que el módulo que recibe el mensaje sepa que datos de los siguientes tomar en consideración. Este dato puede tomar valores entre 1 y 4 siendo:
  - CODE = 1: para establecer valor de una salida digital.
  - CODE = 2: para establecer el color de la tira de LEDs.

- CODE = 3: para obtener el valor de la entrada analógica del módulo que envía el mensaje.
  - CODE = 4 para obtener el valor del sensor de movimiento del módulo que envía el mensaje.
- PIN: es el número de pin de la salida digital de la cual se desea alterar el estado (las tres salidas digitales del módulo corresponden a los pines 5, 18 y 19).
  - VALUE: es el estado que se quiere establecer para el pin (1 = HIGH o 0 = LOW).
  - G: corresponde a la cantidad de color verde de la tira de LEDs, puede tomar valores entre 0 y 255.
  - B: corresponde a la cantidad de color rojo de la tira de LEDs, puede tomar valores entre 0 y 255.
  - R: corresponde a la cantidad de color azul de la tira de LEDs, puede tomar valores entre 0 y 255.
  - AI\_VALUE: es el valor de la entrada analógica del módulo que envía el mensaje.
  - S\_VALUE: es el valor del sensor de movimiento del módulo que envía el mensaje.
  - BUFFER\_SIZE: es el tamaño del mensaje que se envía.

En el archivo TWM.cpp se encuentran programadas todas las funciones de la librería. Se va a proceder a explicar cómo se han programado y para qué sirven. Las funciones se han dividido en base a los componentes que se quieren controlar. El código está disponible en el Anexo 6.10.2.

Las funciones correspondientes al control de las salidas digitales del módulo son las siguientes:

- `void TIM::set_Mosfet(int pin_mosfet, int value)`

Sirve para establecer el estado (HIGH o LOW) de una de las salidas digitales del módulo. Como parámetros de entrada recibe: el número de pin cuyo estado se quiere alterar (pines posibles: 5, 18 o 19) y el valor de ese estado (1 = HIGH o 0 = LOW).

*Estructura:*

En su interior se utiliza la función `digitalWrite(pin_mosfet, value)` de la librería de Arduino, esta accede a los registros de entrada/salida del microcontrolador y establece al pin seleccionado el estado deseado.

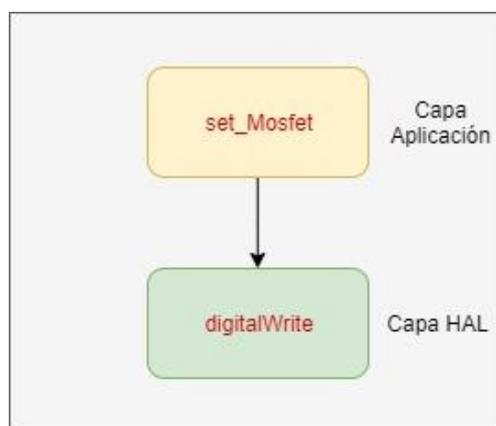


Figura 5.21: Estructura función `set_Mosfet`

- `void TIM::set_Mosfet_Server(WiFiClient client, char nombre, int pin_mosfet, int value)`

En el caso de que varios módulos se comunicaran a través de un sistema por sockets, esta función podría utilizarse en el código de los módulos que se programan como clientes.

Sirve para que un cliente pueda decidir el estado (HIGH o LOW) de una de las salidas digitales del módulo que hace de servidor.

Como parámetros de entrada recibe: el objeto de la clase WiFiClient, para poder utilizar sus funciones para enviar el mensaje, y el nombre que se quiere asignar al cliente (Ejemplo: 'P' = Pierna).

Además, también recibe como parámetros el pin cuyo estado se quiere alterar (pines disponibles: 5, 18 o 19) y el valor de ese estado (1 = HIGH o 0 = LOW). De esta manera el cliente enviará esta información al servidor que al recibirla asignará a la salida elegida el estado deseado.

*Estructura:*

En el interior de la función se compone el mensaje (o buffer) con los parámetros que ha recibido de entrada y se asigna al dato CODE el valor 1, que corresponde a que se quiere controlar una salida digital.

Para enviar el mensaje al servidor se utiliza la función `client.write(buffer, Protocol::BUFFER_SIZE)` de la librería WiFiClient. Esta función accede a los registros WiFi del microcontrolador que se encargan de enviarlo al servidor. Una vez escrito el mensaje sobre los registros, a través de la función `client.flush()` se espera hasta que se hayan enviado todos los datos del buffer y finalmente se cierra la conexión con el servidor a través de la función `client.stop()`.

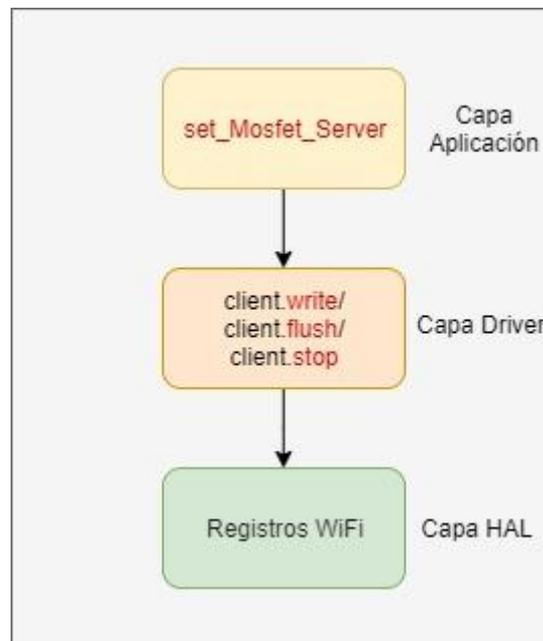


Figura 5.22: Estructura función `set_Mosfet_Server`

- `void TIM::set_Mosfet_OK(WiFiClient client)`

En el caso de que varios módulos se comunicaran a través de un sistema por sockets, esta función podría utilizarse en el código del módulo que se programa como servidor.

Sirve para responder al cliente una vez se haya recibido su mensaje. Como parámetro de entrada recibe el objeto de la clase WiFiClient para poder utilizar sus funciones.

*Estructura:*

En su interior se utiliza la función `client.println("<MOSFET_OK>")` de la librería de WiFiClient, esta accede a los registros WiFi del microcontrolador que se encargan de enviar la respuesta al cliente.

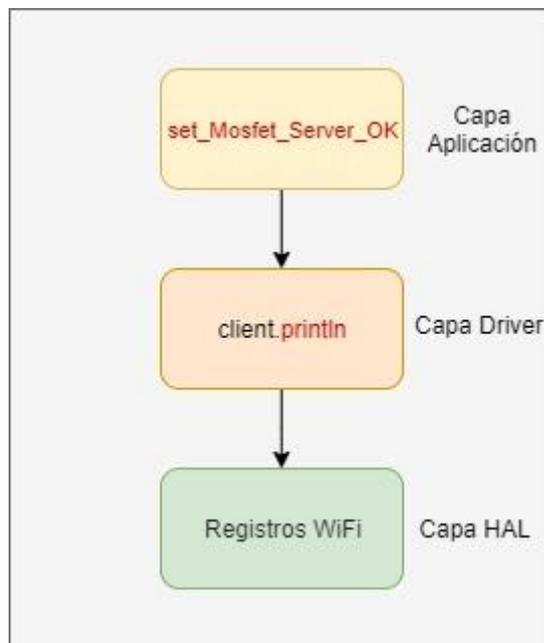


Figura 5.23: Estructura función `set_Mosfet_Server_OK`

Las funciones correspondientes al control de la tira de LEDs del módulo son las siguientes:

- `void TIM::set_Tira_Led(int n_led, int G, int R, int B)`

Sirve para establecer el color de cada pixel de la tira de LEDs del módulo. Como parámetros de entrada recibe el número de pixel cuyo color se quiere establecer y la cantidad de cada color primario.

*Estructura:*

En su interior se utiliza la función `pixel.setPixelColor(n_led,G,R,B)` de la librería Adafruit\_NeoPixel que permite programar tiras de LEDs direccionables, esta función accede a los registros del circuito integrado que posee cada pixel de la tira de LEDs y se encarga de dejar grabado el color del pixel. Después se llama a la función `pixel.show()`, que da la orden al circuito integrado de encender el pixel que controla.

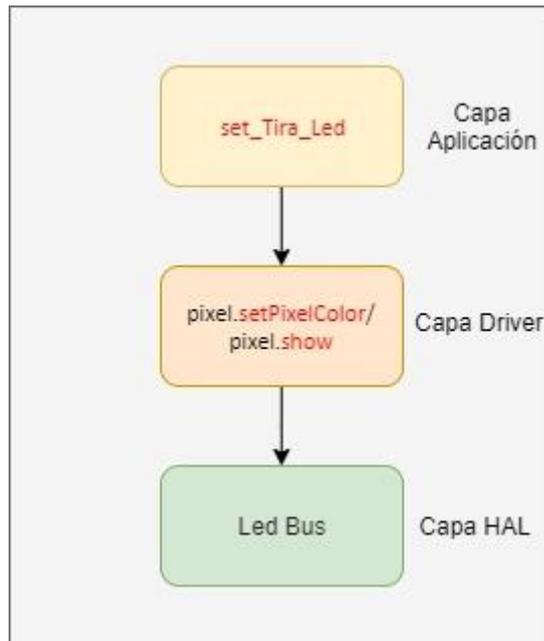


Figura 5.24: Estructura función `set_Tira_Led`

Con esta función se tuvieron algunos problemas que se comentan en el Anexo 5.2.

- `void TIM::set_Tira_Led_Server(WiFiClient client, char nombre, int G, int R, int B)`

En el caso de que varios módulos se comunicaran a través de un sistema por sockets, esta función podría utilizarse en el código de los módulos que se programan como clientes.

Sirve para que un cliente pueda decidir el color de la tira de LEDs del servidor. Como parámetros de entrada recibe el objeto de la clase `WiFiClient`, para poder utilizar las funciones de esta librería para enviar el mensaje, y el nombre que se quiere asignar al cliente. Además, también recibe como parámetro la cantidad de cada color primario que se quiere dar a la tira de LEDs. De esta manera el cliente enviará esta información al servidor que al recibirla asignará a la tira el color deseado.

*Estructura:*

En su interior compone el mensaje con los parámetros que ha recibido de entrada y asigna al dato `CODE` el valor 2, que corresponde a que se quiere controlar la tira de LEDs. El proceso de comunicación es el mismo que el que ya se explicó anteriormente para la función `set_Mosfet_Server`.

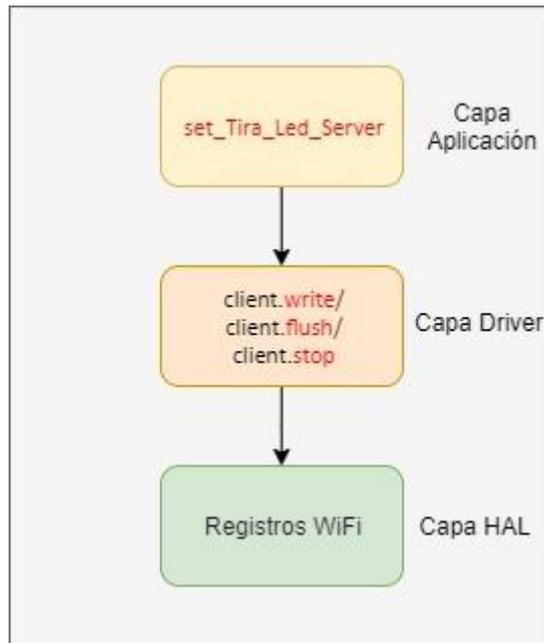


Figura 5.25: Estructura función `set_Tira_Led_Server`

- `void TIM::set_Tira_Led_OK(WiFiClient client)`

En el caso de que varios módulos se comunicaran a través de un sistema por sockets, esta función podría utilizarse en el código del módulo que se programa como servidor.

Sirve para responder al cliente una vez se haya recibido su mensaje. Como parámetro de entrada recibe el objeto de la clase `WiFiClient` para poder utilizar sus funciones.

*Estructura:*

En su interior llama a la función `client.println("<LED_OK>")` y el proceso de comunicación es el mismo que se ha explicado para la función `set_Mosfet_OK`.

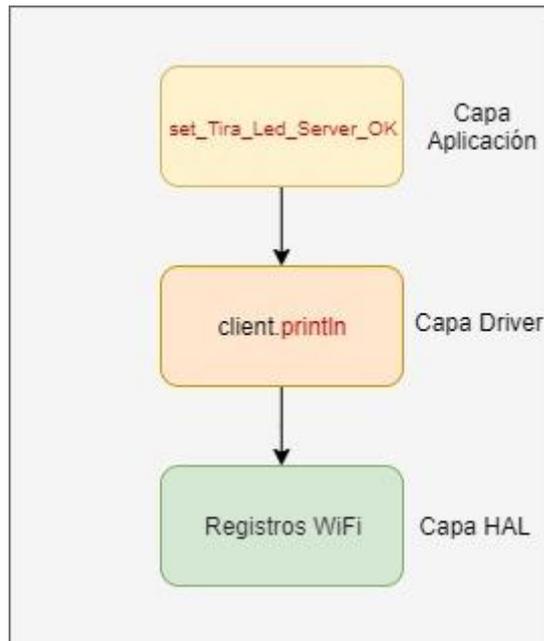


Figura 5.26: Estructura función `set_Tira_Led_Server_OK`

Las funciones correspondientes a la obtención y envío del valor a la entrada analógica del módulo son las siguientes:

- `int TIM::get_AI_Value()`:

Sirve para obtener el valor en la entrada analógica del módulo. Dado que el microcontrolador tiene un convertor ADC de 12 bits, esta función devolverá un valor entero comprendido entre 0 y 4095.

*Estructura:*

En su interior llama a la función `analogRead(14)` de la librería de Arduino, esta accede al registro del convertor ADC del pin 14 del microcontrolador y lee el valor deseado.

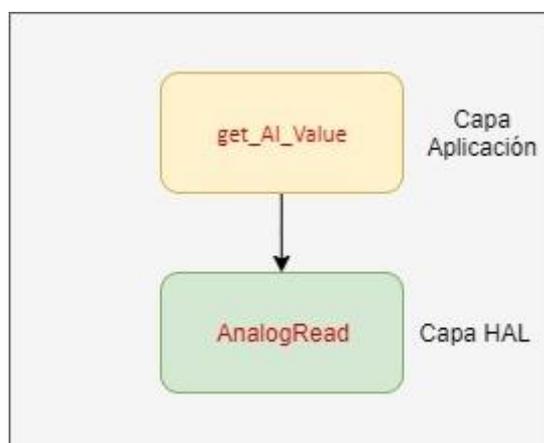


Figura 5.27: Estructura función `get_AI_Value`

- `void TIM::send_AI_Value_to_Server(WiFiClient client, char nombre)`

En el caso de que varios módulos se comunicaran a través de un sistema por sockets, esta función podría utilizarse en el código de los módulos que se programan como clientes.

Sirve para enviar el valor en la entrada analógica del cliente al servidor. Como parámetros de entrada recibe el objeto de la clase `WiFiClient`, para poder utilizar las funciones de esta librería para enviar el mensaje, y el nombre que se quiere asignar al cliente.

*Estructura:*

En su interior compone el mensaje con los parámetros que ha recibido de entrada. Asigna al dato `CODE` el valor 3, que corresponde a que se envía el valor en la entrada analógica, y al dato `AI_VALUE` el valor obtenido por la función `get_AI_Value()` anteriormente explicada. El proceso de comunicación es el mismo que el que ya se explicado anteriormente para la función `set_Mosfet_Server`.

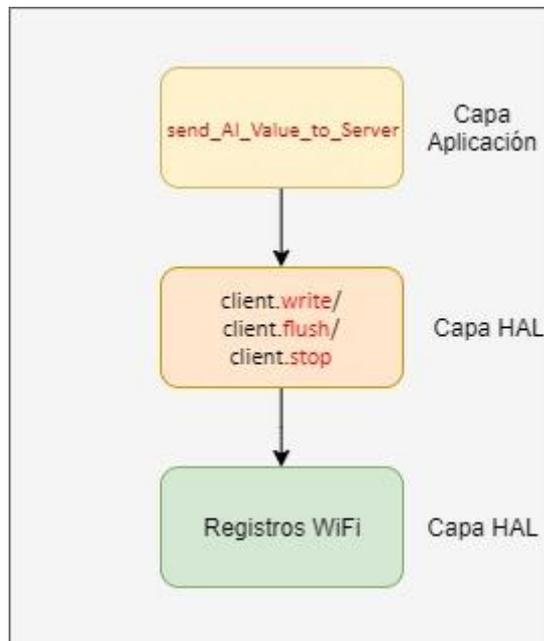


Figura 5.28: Estructura función `send_AI_Value_to_Server`

- `void TIM::get_AI_Value_OK(WiFiClient client)`

En el caso de que varios módulos se comunicaran a través de un sistema por sockets, esta función podría utilizarse en el código del módulo que se programa como servidor.

Sirve para responder al cliente una vez se haya recibido su mensaje. Como parámetro de entrada recibe el objeto de la clase `WiFiClient` para poder utilizar sus funciones.

*Estructura:*

En su interior llama a la función `client.println("<AI_VALUE_OK>")` y el proceso de comunicación es el mismo que se ha explicado para la función `set_Mosfet_OK`.

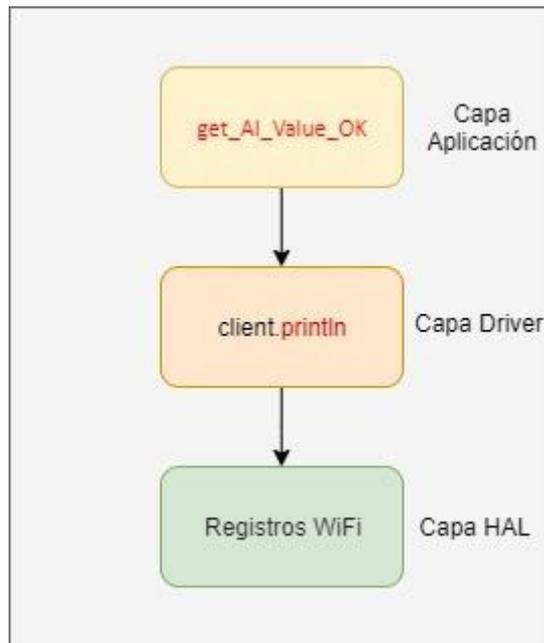


Figura 5.29: Estructura función `get_AI_Value_OK`

Las funciones correspondientes a la obtención y envío del valor del sensor de movimiento del módulo son las siguientes:

- `float TIM::get_Sensor(MPU9250 imu, char eje)`

Sirve para obtener el valor de aceleración detectado por el sensor de movimiento del módulo. Como parámetros de entrada recibe el objeto `imu` de la clase `MPU9250`, para poder utilizar sus funciones, y el eje en el cual se quiere obtener el valor.

*Estructura:*

En su interior se utiliza la función `imu.readSensor()` de la librería de `MPU9250`, está a través del protocolo de comunicación `I2C`, lee los registros donde están guardados los valores medidos por el acelerómetro, giróscopo y magnetómetro del sensor. Según el eje que se haya pasado como parámetro se llamará a la función correspondiente para obtener el valor de aceleración deseado.

Ejemplo: `eje = 'x'`, el valor se obtendrá a través de la llamada a la función `imu.getAccelX_mss()`.

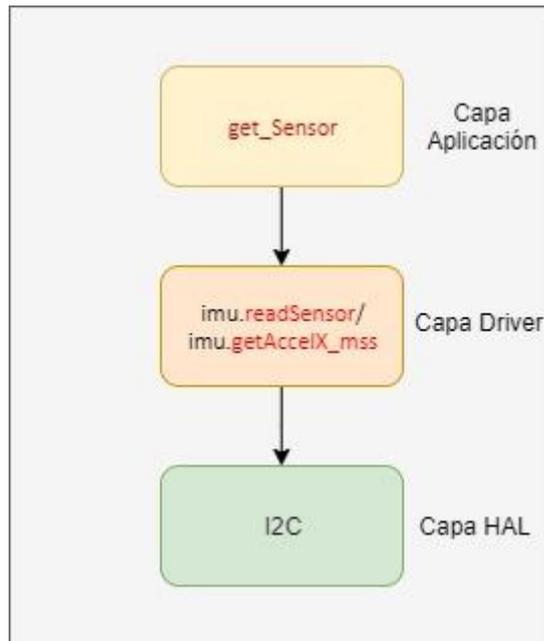


Figura 5.30: Estructura función `get_Sensor`

- `void TIM::send_Sensor_Value_to_Server(WiFiClient client, char nombre, MPU9250 imu, char eje)`

En el caso de que varios módulos se comunicaran a través de un sistema por sockets, esta función podría utilizarse en el código de los módulos que se programan como clientes.

Sirve para enviar el valor de aceleración del sensor de movimiento del cliente al servidor. Como parámetros de entrada recibe el objeto de la clase `WiFiClient`, para poder utilizar las funciones de esta librería para enviar el mensaje, y el nombre que se quiere asignar al cliente.

Además, recibe también como parámetros el objeto `imu` de la clase `MPU9250` que y el eje en el cual se quiere obtener el valor del sensor.

*Estructura:*

En su interior compone el mensaje con los parámetros que ha recibido de entrada. Asigna al dato `CODE` el valor 4, que corresponde a que se envía el valor del sensor de movimiento y al dato `S_VALUE` el valor obtenido por la función `get_Sensor(imu, eje)`; anteriormente explicada. El proceso de comunicación es el mismo que el que ya se explicado anteriormente para la función `set_Mosfet_Server`.

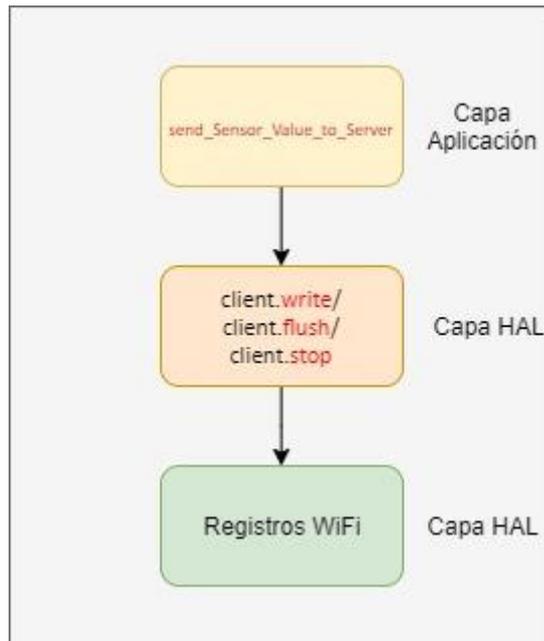


Figura 5.31: Estructura función `send_Sensor_Value_to_Server`

- `void TIM::get_Sensor_OK(WiFiClient client)`

En el caso de que varios módulos se comunicaran a través de un sistema por sockets, esta función podría utilizarse en el código del módulo que se programa como servidor.

Sirve para responder al cliente una vez se haya recibido su mensaje. Como parámetro de entrada recibe el objeto de la clase `WiFiClient` para poder utilizar sus funciones.

*Estructura:*

En su interior llama a la función `client.println("<SENSOR_VALUE_OK>")` y el proceso de comunicación es el mismo que se ha explicado para la función `set_Mosfet_OK`.

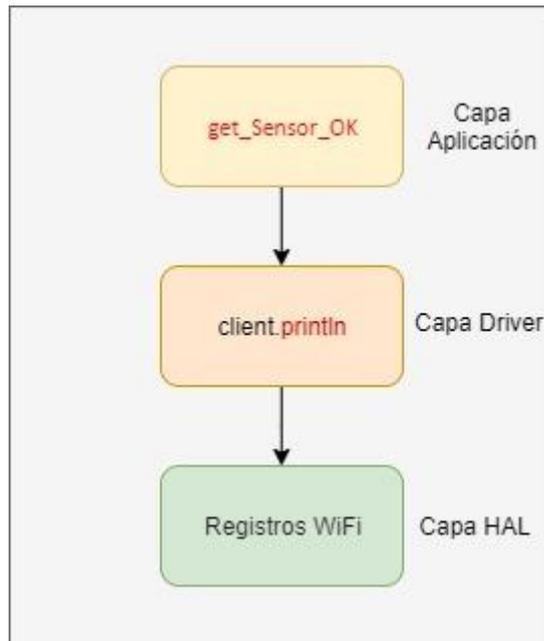


Figura 5.32: Estructura función `get_Sensor_OK`

### 5.2.3.3 Casos de uso

Se han desarrollado tres diferentes programas de Arduino IDE que corresponden a tres posibles casos de uso que puede tener el módulo inalámbrico. Los tres posibles casos de uso son:

- CASO 1: el módulo no está conectado a ninguna red WiFi y tiene un funcionamiento autónomo según su programación. El código está disponible en el Anexo 6.7.



Figura 5.33: Funcionamiento autónomo (CASO 1)

- CASO 2: se utilizan más módulos a la vez, todos están conectados a una misma red WiFi. Un módulo actúa como servidor y escucha al resto de módulos que tienen función de clientes. Cada cliente, mediante el protocolo ideado para este tipo de aplicación, comunica para controlar un actuador del servidor y recibe por su parte una respuesta afirmativa cada

vez que esto sucede. Los códigos del servidor y de los clientes están disponibles en el Anexo 6.8.

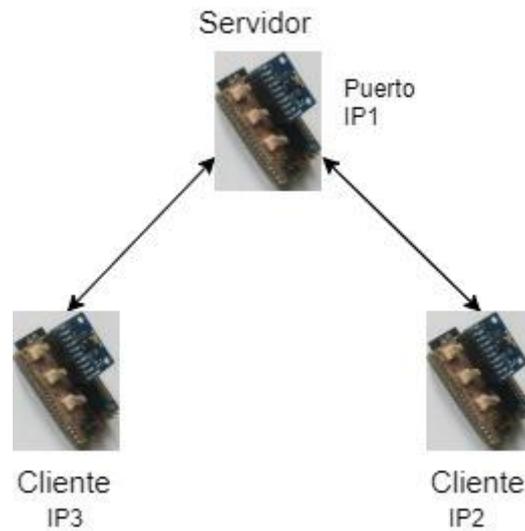


Figura 5.34: Estructura servidor-cliente CASO 2

- CASO 3: también se utilizan más módulos a la vez, estando conectados todos a una misma red WiFi. La diferencia respecto del CASO 2 es que todos los módulos tienen función de clientes, mientras el que actúa como servidor es un PC. El servidor se ha realizado mediante un script en lenguaje Python que se ejecuta sobre el PC. Los clientes pueden atender las ordenes que llegan desde el servidor cuando quiere controlar alguno de sus actuadores, o bien, pueden enviarle la información recogida por sus sensores. Una vez llegada la información, el PC es capaz de registrarla y mostrarla mediante gráficas. Los códigos del servidor y de los clientes están disponibles en el Anexo 6.9.

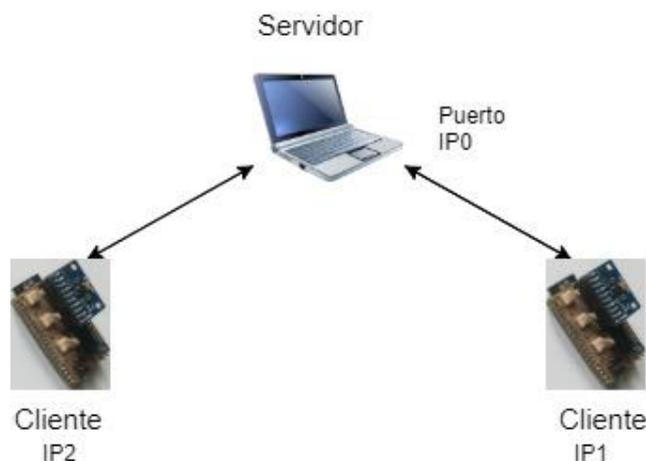


Figura 5.35: Estructura servidor-cliente CASO 2

## 6. Conclusiones

### 6.1 Del proyecto

Este proyecto ha sido muy intenso y lleno de satisfacción no obstante todas las dificultades que ha habido durante los meses en los que se ha desarrollado.

En cuanto al desarrollo electrónico, se ha sabido superar el problema causado por imposibilidad de tener el módulo inalámbrico original para el festival, elaborando un plan de contingencia que ha dado muy buenos resultados y ha permitido que el proyecto pudiera seguir y alcanzar los objetivos establecidos al principio.

Además, se han alcanzado los objetivos propuestos respecto de la programación del módulo, desarrollando varios posibles casos de uso y proporcionándole mucha versatilidad permitiendo que pueda utilizarse no solo para la danza sino para cualquier tipo de aplicación.

Respecto de la colaboración con Trayectos, en particular con Julia, se ha conseguido formar un equipo de trabajo multidisciplinar intentando llegar siempre a un acuerdo satisfactorio para ambas partes y afrontando todos los problemas que puede dar una colaboración de trabajo como esta, donde los cambios repentinos son frecuentes, se realizan muchas pruebas y no todo funciona siempre a la primera. Por tanto, a menudo era necesario dedicar más tiempo a solucionar errores o intentar buscar otras soluciones, que dedicarlo a seguir avanzando.

Pero el sacrificio y la dedicación por parte de todos los que han participado en el proyecto, ha permitido poder llegar a crear un espectáculo artístico innovador acorde con lo que se quería conseguir.

### 6.2 Personales

Después de haber finalizado este proyecto, puedo decir que estoy muy satisfecho con el trabajo realizado por haber cumplido los objetivos prefijados y porque me ha dado modo de poder adquirir nuevos conocimientos y mejorar mucho en varios aspectos de la electrónica, como el diseño electrónico y la programación, pero también en aspectos que no están relacionados con ella.

A través de este trabajo he podido mejorar respecto a todas las fases del desarrollo de sistemas electrónicos desde la definición de funciones hasta la puesta a punto, pasando por la selección de componentes óptimos, el diseño electrónico y el de la PCB. Además, he conseguido profundizar todo lo que concierne a la programación de un microcontrolador, he sabido desenvolverme con diferentes tipos de lenguajes de programación y sobre todo he adquirido nuevos conocimientos en el ámbito de la comunicación con otros dispositivos.

Ha sido una oportunidad de poder desarrollar un proyecto diferente a todo lo que había realizado hasta el momento en la carrera. Me he tenido que enfrentar a un trabajo que no es como una práctica donde hay que seguir un guión que te indica que hay que hacer y cómo, sino que he tenido que tomar decisiones siendo consciente de que podían influir sobre el trabajo y si me equivocaba habría tenido que dar un paso hacia atrás y volver a empezar todo de nuevo. Esta experiencia ha sido muy útil para aprender a tomar la iniciativa y saber defender mis ideas, pero a la vez saber adecuarme a las necesidades y consejos del resto de las partes implicadas en el proyecto.

En cuanto a la colaboración con Trayectos, he tenido la suerte de poder trabajar con personas con mucha experiencia en este tipo de colaboraciones interdisciplinarias que me han dado la posibilidad de aprender relacionarme con gente que pertenece a una disciplina diferente a la mía. Esto ha enriquecido mi formación como persona y como profesional y me ha proporcionado nuevas competencias para trabajar y desenvolverme en un mundo diverso.

Tras finalizar todo el proyecto puedo decir que ha superado mis expectativas, y ha sido gracias a las personas con las que he trabajado que me han permitido adquirir muchos más conocimientos de los que esperaba.

## 7. Bibliografía

- [1] Arduino Nano. [Consulta: ]. Disponible en: <https://arduino.cl/arduino-nano/>
- [2] Arduino UNO. [Consulta: ]. Disponible en: <http://arduino.cl/arduino-uno/>
- [3] Trayectos, Laboratorio de danza y nuevos medios. [Consulta: ]. Disponible en: <http://www.danzatrayectos.com/laboratorio-de-danza-y-nuevos-medios/>
- [4] ETOPIA, Centro de Arte y Tecnología. [Consulta: ]. Disponible en: <https://www.zaragoza.es/sede/servicio/equipamiento/5105>
- [5] Unidad de medición inercial. [Consulta: ]. Disponible en: [https://es.wikipedia.org/wiki/Unidad\\_de\\_medici%C3%B3n\\_inercial](https://es.wikipedia.org/wiki/Unidad_de_medici%C3%B3n_inercial)
- [6] Internet de las cosas. [Consulta: ]. Disponible en: [https://es.wikipedia.org/wiki/Internet\\_de\\_las\\_cosas](https://es.wikipedia.org/wiki/Internet_de_las_cosas)
- [7] Luis Llamas, NodeMCU la popular placa de desarrollo con ESP8266. [Consulta: ]. Disponible en: <https://www.luisllamas.es/esp8266-nodemcu/>
- [8] RGB para qué es esto y para que se utiliza en informática. [Consulta: ]. Disponible en: <https://www.profesionalreview.com/2019/01/20/rgb-que-es/>
- [9] Qué es el "hoverboard": Revisamos la evolución electrónica del skate. [Consulta: ]. Disponible en: <https://www.emol.com/noticias/Tecnologia/2015/11/25/760961/Hoverboard.html>
- [10] Trayectos, danza en paisajes urbanos. [Consulta: ]. Disponible en: <http://www.danzatrayectos.com/en/about-us/>
- [11] Seis grados de libertad. [Consulta: ]. Disponible en: [https://es.wikipedia.org/wiki/Seis\\_grados\\_de\\_libertad](https://es.wikipedia.org/wiki/Seis_grados_de_libertad)
- [12] "E-Traces", la tecnología para bailarines que reúne a la tecnología con la danza. [Consulta: ]. Disponible en: <https://zonafranca.mx/cultura-y-entretenimiento/e-traces-la-tecnologia-para-bailarines-que-reune-a-la-tecnologia-con-la-danza>
- [13] Trayectos, Laboratorio de danza y nuevos medios, edición 2018. [Consulta: ]. Disponible en: <http://www.danzatrayectos.com/portfolio/laboratorio-de-danza-y-nuevos-medios-2018/>
- [14] Trayectos, Laboratorio de danza y nuevos medios, edición 2019. [Consulta: ]. Disponible en: <http://www.danzatrayectos.com/portfolio/laboratorio-de-danza-y-nuevos-medios-iii/>

- [15] Cómo utilizar Trello para organizar tu vida casi al completo. [Consulta: ]. Disponible en: <https://es.gizmodo.com/como-organizar-toda-tu-vida-utilizando-trello-1684529913>
- [16] Arduino IDE. [Consulta: ]. Disponible en: [https://es.wikipedia.org/wiki/Arduino\\_IDE](https://es.wikipedia.org/wiki/Arduino_IDE)
- [17] Datasheet BC547. [Consulta: ]. Disponible en: <https://www.sparkfun.com/datasheets/Components/BC546.pdf>
- [18] Espressif, Datasheet ESP\_WROOM-32. [Consulta:]. Disponible en: [https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf)
- [19] Bosch, Datasheet BNO055. [Consulta:]. Disponible en: [https://www.bosch-sensortec.com/bst/products/all\\_products/bno055](https://www.bosch-sensortec.com/bst/products/all_products/bno055)
- [20] ESP32-DevKitC V4 Getting Started Guide. [Consulta:]. Disponible en: <https://docs.espressif.com/projects/esp-idf/en/latest/hw-reference/get-started-devkitc.html>
- [21] Diodes Incorporated, Datasheet DMG2302UK. [Consulta:]. Disponible en: <https://www.mouser.it/datasheet/2/115/DMG2302UKQ-1479702.pdf>
- [22] Luis Llamas, usar Arduino con los IMU de 9DOF MPU-9150 y MPU-9250. [Consulta:]. Disponible en: <https://www.luisllamas.es/usar-arduino-con-los-imu-de-9dof-mpu-9150-y-mpu-9250/>
- [23] XP Power, Datasheet TR10S05 [Consulta:]. Disponible en: [https://www.mouser.it/datasheet/2/942/SF\\_TR10-1508831.pdf](https://www.mouser.it/datasheet/2/942/SF_TR10-1508831.pdf)
- [24] Casas R., López J.M. (2018), Diseño de circuitos electrónicos.
- [25] Protocolos de comunicación en red. [Consulta:]. Disponible en: <https://mastermoviles.gitbook.io/tecnologias2/protocolos-de-comunicacion-en-red>
- [26] Protocolos de comunicación en red: Sockets. [Consulta:]. Disponible en: <https://mastermoviles.gitbook.io/tecnologias2/protocolos-de-comunicacion-en-red#sockets>

## 8. Anexos

Anexo 1: Lista de componentes

Anexo 2: Planos de esquemático y PCB del plan de contingencia

Anexo 3: Protoboards no utilizadas

Anexo 4: Sistemas de comunicación

Anexo 5: Problemas durante el proyecto

Anexo 6: Códigos