

## Resumen

El presente trabajo describe un emprendimiento personal en el que se desarrolló un dispositivo robótico de exploración **ambiental** controlable a distancia con las funciones básicas de desplazamiento, medición y reporte de parámetros ambientales tales como presión, temperatura, humedad y luminosidad en una primera versión. **No obstante, el** sistema cuenta con **una** una arquitectura base robusta y flexible sobre la cual otros sensores pueden ser adaptados para poder tomar lecturas de otros parámetros ambientales, **logrando** por tanto brindar una solución que ayuda a incrementar la oferta en robots exploradores en la industria argentina pudiendo ser utilizados por ejemplo en aplicaciones de minería, agro, explotación de petróleo, etc. Para la implementación **del trabajo** se utilizaron conceptos y herramientas **aprendidos en la especialización**, tales como buenas prácticas en diseño y desarrollo de firmware, la utilización de sistemas operativos de tiempo real como plataforma de ejecución base, protocolos de comunicaciones para sistemas **embebidos**, y técnicas y frameworks de testing para asegurar la calidad del producto final.

## Resumen

El presente trabajo describe un emprendimiento personal en el que se desarrolló un dispositivo robótico de exploración **ambiental**, controlable a distancia con las funciones básicas de desplazamiento, medición y reporte de parámetros ambientales tales como presión, temperatura, humedad y luminosidad en una primera versión. **El** sistema cuenta con una arquitectura base robusta y flexible sobre la cual otros sensores pueden ser adaptados para poder tomar lecturas de otros parámetros ambientales, **y logra** por tanto brindar una solución que ayuda a incrementar la oferta en robots exploradores en la industria argentina pudiendo ser utilizados por ejemplo en aplicaciones de minería, agro, explotación de petróleo, etc. Para la implementación se utilizaron conceptos y herramientas tales como buenas prácticas en diseño y desarrollo de firmware, la utilización de sistemas operativos de tiempo real como plataforma de ejecución base, protocolos de comunicaciones para sistemas **embebidos** y técnicas y frameworks de testing para asegurar la calidad del producto final.

# Índice general

<b>Resumen</b>	<b>I</b>
<b>1. Introducción general</b>	<b>1</b>
1.1. Motivación	1
1.2. Alcance y objetivos	1
1.3. Estado del arte	1
<b>2. Introducción específica</b>	<b>3</b>
2.1. Requerimientos	3
2.2. Tecnologías de hardware utilizadas	3
2.2.1. ESP32	3
2.2.2. DHT11	3
2.2.3. BMP280	4
2.2.4. Fotorresistor	4
2.2.5. Joystick analógico	5
2.2.6. Display	5
2.2.7. Motores	5
2.2.8. Integrados y otras utilidades	5
2.3. Tecnologías de software utilizadas	5
2.3.1. ESP-IDF	5
2.3.2. Docker	6
2.3.3. Visual Studio Code	6
2.3.4. Ubuntu	7
<b>3. Diseño e implementación</b>	<b>9</b>
3.1. Arquitectura de software del sistema	9
3.2. Arquitectura de hardware del sistema	9
3.3. Interfaz de usuario	9
<b>4. Ensayos y resultados</b>	<b>11</b>
4.1. Proceso de desarrollo y aseguramiento de calidad	11
4.2. Prototipos de los diferentes módulos	11
4.2.1. Prototipo de funcionalidad de medición de temperatura y humedad	11
4.2.2. Prototipo de funcionalidad de medición de presión	11
4.2.3. Prototipo de funcionalidad de medición de valor de luminosidad	12
4.2.4. Prototipo de funcionalidad de obtención de valores analógicos del joystick	12
4.2.5. Prototipo de funcionalidad de presentación de display	13
4.2.6. Prototipo de funcionalidad de control de motores DC	13
4.3. Tests de los diferentes módulos	13
4.4. Tests del producto final	13

# Índice general

<b>Resumen</b>	<b>I</b>
<b>1. Introducción general</b>	<b>1</b>
1.1. Motivación	1
1.2. Alcance y objetivos	1
1.3. Estado del arte	1
<b>2. Introducción específica</b>	<b>3</b>
2.1. Requerimientos	3
2.2. Tecnologías de hardware utilizadas	3
2.2.1. Espressif ESP32	3
2.2.2. DHT11	3
2.2.3. BMP280	4
2.2.4. Fotorresistor	4
2.2.5. Joystick analógico	4
2.2.6. Display	5
2.2.7. Motores	5
2.2.8. Integrados y otras utilidades	5
2.3. Tecnologías de software utilizadas	6
2.3.1. ESP-IDF	6
2.3.2. Docker	6
2.3.3. Visual Studio Code	6
2.3.4. Ubuntu	6
<b>3. Diseño e implementación</b>	<b>9</b>
3.1. Arquitectura de software del sistema	9
3.2. Arquitectura de hardware del sistema	9
3.3. Interfaz de usuario	9
<b>4. Ensayos y resultados</b>	<b>11</b>
4.1. Proceso de desarrollo y aseguramiento de calidad	11
4.2. Prototipos de los diferentes módulos	11
4.2.1. Prototipo de funcionalidad de medición de temperatura y humedad	11
4.2.2. Prototipo de funcionalidad de medición de presión	12
4.2.3. Prototipo de funcionalidad de medición de valor de luminosidad	12
4.2.4. Prototipo de funcionalidad de obtención de valores analógicos del joystick	13
4.2.5. Prototipo de funcionalidad de presentación de display	13
4.2.6. Prototipo de funcionalidad de control de motores DC	13
4.3. Tests de los diferentes módulos	13
4.4. Tests del producto final	13

VI

4.5. Reportes de testing . . . . .	13
4.6. Verificacion y validacion del producto . . . . .	13
4.7. Documentacion del producto . . . . .	13
<b>5. Conclusiones</b>	<b>15</b>
5.1. Conclusiones generales . . . . .	15
5.2. Próximos pasos . . . . .	15

VI

4.5. Reportes de testing . . . . .	13
4.6. Verificacion y validacion del producto . . . . .	14
4.7. Documentacion del producto . . . . .	14
<b>5. Conclusiones</b>	<b>15</b>
5.1. Próximos pasos . . . . .	15

## Capítulo 1

### Introducción general

#### 1.1. Motivación

El presente **proyecto** es un emprendimiento personal que **busca desarrollar** un dispositivo robótico de exploración ambiental controlable a distancia con las funciones básicas de desplazamiento, medición y reporte de parámetros ambientales tales como presión, temperatura, humedad y luminosidad.

#### 1.2. Alcance y objetivos

..

#### 1.3. Estado del arte

Los robots exploradores son dispositivos robotizados capaces de moverse de forma autónoma, y/o controlados a distancia, que han sido creados con el fin de reconocer y explorar un lugar o entorno donde una persona no pueda o deba acceder ya sea por motivos de capacidad, practicidad o seguridad. Por este motivo, en función de las necesidades de desplazamiento, existen diferentes sistemas de motricidad, como son por ejemplo, los bípedos, cuadrúpedos, con ruedas, tracción oruga, acuáticos/sumergibles, aéreos, etc. En cuanto a la forma de control, hay manejados por control remoto cableado o inalámbrico, habiendo equipos más **sofisticados** que gracias a aplicaciones de Inteligencia **Artificial** están **preparados** para desplazarse y tomar decisiones de forma autónoma. Algunos de los tipos de robots exploradores más conocidos son los espaciales, de minas, de rescate en catástrofes, de tuberías, acuáticos y/o submarinos, y de suelos.

## Capítulo 1

### Introducción general

Esta sección presenta la motivación, alcance y objetivos del producto en el marco del estado del arte de otros similares en la industria.

#### 1.1. Motivación

El presente **trabajo** es un emprendimiento personal **en el** que **se desarrolla** un dispositivo robótico de exploración ambiental controlable a distancia con las funciones básicas de desplazamiento, medición y reporte de parámetros ambientales tales como presión, temperatura, humedad y luminosidad.

#### 1.2. Alcance y objetivos

..

#### 1.3. Estado del arte

Los robots exploradores son dispositivos robotizados capaces de moverse de forma autónoma, y/o controlados a distancia, que han sido creados con el fin de reconocer y explorar un lugar o entorno donde una persona no pueda o deba acceder ya sea por motivos de capacidad, practicidad o seguridad. Por este motivo, en función de las necesidades de desplazamiento, existen diferentes sistemas de motricidad, como son por ejemplo, los bípedos, cuadrúpedos, con ruedas, tracción oruga, acuáticos/sumergibles, aéreos, etc. En cuanto a la forma de control, **los** hay manejados por control remoto cableado o inalámbrico, habiendo equipos más **sofisticados**, que gracias a aplicaciones de Inteligencia **Artificial**, están **preparados** para desplazarse y tomar decisiones de forma autónoma. Algunos de los tipos de robots exploradores más conocidos son los espaciales, de minas, de rescate en catástrofes, de tuberías, acuáticos y/o submarinos, y de suelos.

## Capítulo 2

### Introducción específica

Todos los capítulos deben comenzar con un breve párrafo introductorio que indique cuál es el contenido que se encontrará al leerlo. La redacción sobre el contenido de la memoria debe hacerse en presente y todo lo referido al proyecto en pasado, siempre de modo impersonal.

#### 2.1. Requerimientos

...

#### 2.2. Tecnologías de hardware utilizadas

##### 2.2.1. ESP32

ESP32 es una serie de microcontroladores de bajo costo y bajo consumo creado y desarrollado por Espressif Systems embebido en un chip con Wi-Fi integrado (2.4 GHz band) y Bluetooth de modo dual. Emplea dos cores Xtensa® 32-bit LX6 CPU. Incluye interruptores de antena integrados, balun de RF, amplificador de potencia, amplificador de recepción de bajo ruido, filtros, un co-procesador ULP (Ultra Low Power), módulos de administración de energía y varios periféricos. La placa utilizada para el desarrollo del presente trabajo es ESP32-WROOM-32D.



##### 2.2.2. DHT11

El DHT11 es un sensor digital de temperatura y humedad relativa de bajo costo y fácil uso. Integra un sensor capacitivo de humedad y un termistor para medir el aire circundante, y muestra los datos mediante una señal digital en el pin de datos (no posee salida analógica). Utilizado en aplicaciones académicas relacionadas al

## Capítulo 2

### Introducción específica

Esta sección presenta los requerimientos del producto así como una breve introducción técnica a las herramientas hardware y software utilizadas.

#### 2.1. Requerimientos

...

#### 2.2. Tecnologías de hardware utilizadas

##### 2.2.1. Espressif ESP32

ESP32 es una serie de microcontroladores de bajo costo y bajo consumo creado y desarrollado por Espressif Systems embebido en un chip con Wi-Fi integrado (2.4 GHz band) y Bluetooth de modo dual. Emplea dos cores Xtensa® 32-bit LX6 CPU. Incluye interruptores de antena integrados, balun de RF, amplificador de potencia, amplificador de recepción de bajo ruido, filtros, un co-procesador ULP (Ultra Low Power), módulos de administración de energía y varios periféricos. La placa utilizada para el desarrollo del presente trabajo es ESP32-WROOM-32D.



##### 2.2.2. DHT11

El DHT11 es un sensor digital de temperatura y humedad relativa de bajo costo y fácil uso. Integra un sensor capacitivo de humedad y un termistor para medir el aire circundante, y muestra los datos mediante una señal digital en el pin de datos (no posee salida analógica). Utilizado en aplicaciones académicas relacionadas al control automático de temperatura, aire acondicionado, monitoreo ambiental en agricultura y más.



control automático de temperatura, aire acondicionado, monitoreo ambiental en agricultura y más.



### 2.2.3. BMP280

El BMP280 es un sensor de presión barométrica absoluta, especialmente factible para aplicaciones móviles. Sus reducidas dimensiones y su bajo consumo permiten su implementación en dispositivos alimentados por batería como teléfonos móviles, módulos GPS o relojes. Se basa en la tecnología comprobada de sensor de presión piezorresistivo de Bosch que ofrece alta precisión y linealidad, así como estabilidad a largo plazo y alta robustez de EMC. El dispositivo está optimizado en términos de consumo de energía, puede ser utilizado con I2C o SPI, y ofrece una precisión absoluta de  $\pm 1$  hPa en lectura de presión y  $\pm 1,0$  °C para temperatura. Además, dada la precisión del dispositivo y al hecho de que la presión cambia con la altitud, también puede usarse como un altímetro con una precisión de  $\pm 1$  metro.



### 2.2.4. Fotoresistor

El fotoresistor es una resistencia eléctrica que varía su valor en función de la cantidad de luz que incide sobre su superficie. Cuando mayor sea la intensidad de luz que incide en la superficie del LDR o fotoresistor menor será su resistencia y en cuanto menor sea la luz que incida sobre éste mayor será su resistencia. Cuando el fotoresistor no está expuesto a radiaciones luminosas, los electrones están firmemente unidos en los átomos que lo conforman, pero cuando sobre él inciden radiaciones luminosas, esta energía libera electrones con lo cual el material se hace más conductor, y de esta manera disminuye su resistencia. Las resistencias LDR solamente reducen su resistencia con una radiación luminosa situada dentro de una determinada banda de longitudes de onda. El fotoresistor construido con sulfuro de cadmio son sensibles a todas las radiaciones luminosas visibles y las construidas con sulfuro de plomo solamente son sensibles a las radiaciones infrarrojas.



### 2.2.3. BMP280

El BMP280 es un sensor de presión barométrica absoluta, especialmente factible para aplicaciones móviles que puede ser utilizado con I2C o SPI. Permite alta precisión y linealidad, estabilidad a largo plazo, alta robustez de EMC a un muy bajo consumo.



### 2.2.4. Fotoresistor

El fotoresistor es una resistencia eléctrica que varía su valor en función de la cantidad de luz que incide sobre su superficie. Cuando el fotoresistor no está expuesto a radiaciones luminosas, los electrones están firmemente unidos en los átomos que lo conforman, por lo que alcanza su máxima resistencia eléctrica, y cuando sobre él inciden radiaciones luminosas, esta energía libera electrones con lo cual el material se hace más conductor, y se disminuye su resistencia.



### 2.2.5. Joystick analógico

El módulo de joystick analógico está construido sobre el montaje de dos potenciómetros en un ángulo de 90 grados. Los potenciómetros están conectados a una palanca corta centrada por resortes. Este módulo produce una salida de alrededor de 2,5 V cuando inicialmente se encuentra en posición de reposo (en el centro), mientras que en el trayecto del desplazamiento de la palanca hará que la salida varíe de 0V a 5V dependiendo de su dirección X e Y. Al conectar este módulo a un microcontrolador se puede leer un valor de alrededor de 512 en su posición de reposo mientras que al moverlo cambia entre 0 y 1023 dependiendo de su posición.



### 2.2.5. Joystick analógico

El módulo de joystick analógico está construido sobre el montaje de dos potenciómetros en un ángulo de 90 grados. Los potenciómetros están conectados a una palanca corta centrada por resortes. Este módulo produce una salida de alrededor de 2,5 V cuando inicialmente se encuentra en posición de reposo (en el centro), mientras que en el trayecto del desplazamiento de la palanca hará que la salida varíe de 0V a 5V dependiendo de su dirección X e Y. Al conectar este módulo a un microcontrolador se puede leer un valor de alrededor de 512 en su posición de reposo mientras que al moverlo cambia entre 0 y 1023 dependiendo de su posición.



### 2.2.6. Display

...pendiente...

### 2.2.7. Motores

...pendiente...

### 2.2.8. Integrados y otras utilidades

...pendiente...

## 2.3. Tecnologías de software utilizadas

### 2.3.1. ESP-IDF

Espressif proporciona recursos básicos de hardware y software para ayudar a los desarrolladores de aplicaciones a realizar sus ideas utilizando el hardware de la serie ESP32. El framework de software de Espressif está destinado al desarrollo



### 2.2.6. Display

El display consta de una pantalla de cristal líquido de 1602 caracteres, en un módulo de matriz de puntos para mostrar letras, números y caracteres, etc. Permite representar dos filas con hasta 16 caracteres en cada una y dado que se encuentra integrado a una interfaz adaptadora I2C puede ser controlado por este protocolo.



### 2.2.7. Motores

El motor DC (o corriente continua), pertenece a la clase de los electromotores y sirve principalmente para transformar la energía eléctrica en energía mecánica. Estos motores operan con un voltaje entre 3 y 6 Volts, corriente de 150 mA, permiten una velocidad de entre 90 y 200 RPM y un torque de entre 0.15Nm y 0.60Nm.



### 2.2.8. Integrados y otras utilidades

...pendiente...

de aplicaciones de Internet de las cosas (IoT) con Wi-Fi, Bluetooth, administración de energía y varias otras características del sistema. Sus componentes son:

1. Toolchain para compilar el código para ESP32
2. Build tools - con utilidades como CMake y Ninja para construir la aplicación completa para ESP32
3. ESP-IDF que esencialmente contiene la API de desarrollo (software base y librerías complementarias) para ESP32 y scripts para ejecutar Toolchain



### 2.3.2. Docker

Docker es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos. Docker utiliza características de aislamiento de recursos del kernel Linux, tales como cgroups y espacios de nombres (namespaces) para permitir que **contenedores livianos** independientes se ejecuten en paralelo de **manera** aislada evitando la sobrecarga de iniciar y mantener máquinas virtuales.



### 2.3.3. Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.

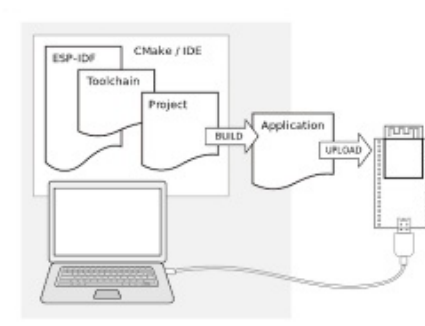


## 2.3. Tecnologías de software utilizadas

### 2.3.1. ESP-IDF

Espressif proporciona recursos básicos de hardware y software para ayudar a los desarrolladores de aplicaciones a realizar sus ideas utilizando el hardware de la serie ESP32. El framework de software de Espressif está destinado al desarrollo de aplicaciones de Internet de las cosas (IoT) con Wi-Fi, Bluetooth, administración de energía y varias otras características del sistema. Sus componentes son:

1. Toolchain para compilar el código para ESP32
2. Build tools - con utilidades como CMake y Ninja para construir la aplicación completa para ESP32
3. ESP-IDF que esencialmente contiene la API de desarrollo (software base y librerías complementarias) para ESP32 y scripts para ejecutar Toolchain



### 2.3.2. Docker

Docker es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos. Docker utiliza características de aislamiento de recursos del kernel Linux, tales como cgroups y espacios de nombres (namespaces) para permitir que **contenedores livianos** independientes se ejecuten en paralelo de **manera** aislada evitando la sobrecarga de iniciar y mantener máquinas virtuales.

### 2.3.3. Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.

### 2.3.4. Ubuntu

Ubuntu es una distribución Linux basada en Debian GNU/Linux y patrocinada por Canonical, que incluye principalmente software libre y de código abierto.



**2.3.4. Ubuntu**

Ubuntu es una distribución Linux basada en Debian GNU/Linux y patrocinada por Canonical, que incluye principalmente software libre y de código abierto. Puede utilizarse en ordenadores y servidores, está orientado al usuario promedio, con un fuerte enfoque en la facilidad de uso y en mejorar la experiencia del usuario.



...

Puede utilizarse en ordenadores y servidores, está orientado al usuario promedio, con un fuerte enfoque en la facilidad de uso y en mejorar la experiencia del usuario.

...

## Capítulo 3

### Diseño e implementación

#### 3.1. Arquitectura de software del sistema

...

#### 3.2. Arquitectura de hardware del sistema

...

#### 3.3. Interfaz de usuario

...

## Capítulo 3

### Diseño e implementación

Esta sección presenta los detalles técnicos del diseño e implementación de las diferentes funcionalidades del producto, la arquitectura hardware y software, y finalmente la interfaz de usuario para el control y reporte de las operaciones del robot.

#### 3.1. Arquitectura de software del sistema

...

#### 3.2. Arquitectura de hardware del sistema

...

#### 3.3. Interfaz de usuario

...

Capítulo 4

Ensayos y resultados

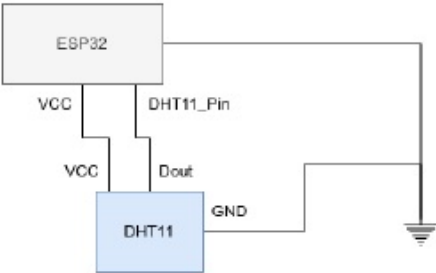
4.1. Proceso de desarrollo y aseguramiento de calidad

...

4.2. Prototipos de los diferentes modulos

4.2.1. Prototipo de funcionalidad de medición de temperatura y humedad

Para el desarrollo de este prototipo se utilizaron el framework ESP-IDF y como la librería de código de ESP-IDF Components Lib para la gestión del DHT11. A continuación se puede apreciar el conexionado del prototipo.



De acuerdo al pinout de ESP32 [ref] se configuraron los pines de acuerdo a la tabla a continuación.

Logic Name	Pin GPIO
Pin DHT11	2

4.2.2. Prototipo de funcionalidad de medicion de presion

Para el desarrollo de este prototipo se utilizaron el framework ESP-IDF y como la librería de código de ESP-IDF Components Lib para la gestión del BMP280. A continuación se puede apreciar el conexionado del prototipo.

Capítulo 4

Ensayos y resultados

Esta sección presenta los diferentes prototipos realizados para determinar la viabilidad de cada una de las funcionalidades provistas, la metodología de desarrollo, testing, y finalmente los entregables finales del trabajo.

4.1. Proceso de desarrollo y aseguramiento de calidad

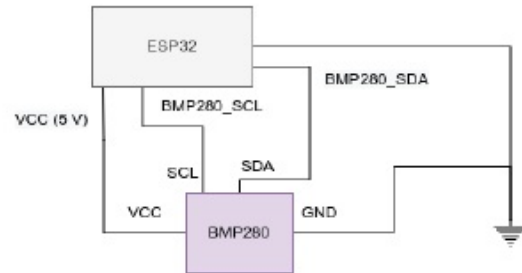
Para el proceso de desarrollo se utilizó TDD (Test Driven Development) con CMock, Ceedling y Unity como frameworks para abordar esto. Para la construcción de los casos de pruebas se utilizó Control Flow Test como técnica de caja blanca a fin de diseñar los tests unitarios y de integración. Adicionalmente, se configuró el entorno de desarrollo basado en Docker, por lo que se extiende la imagen de Expressif y se agregó el conjunto de utilidades ESP-IDF Components como parte de la misma. Luego de versionar esta imagen en el Docker Registry utilizado se configuró el servidor de integración continua para ejecutar las compilaciones usando esta imagen por cada push realizado. En la siguiente imagen se puede apreciar la infraestructura implementada para abordar el ciclo de desarrollo y despliegue.

[pendiente imagen]

4.2. Prototipos de los diferentes modulos

4.2.1. Prototipo de funcionalidad de medición de temperatura y humedad

Para el desarrollo de este prototipo se utilizó el framework ESP-IDF y la librería de código ESP-IDF Components que provee el soporte para gestionar el DHT11. A continuación se puede apreciar el conexionado del prototipo.

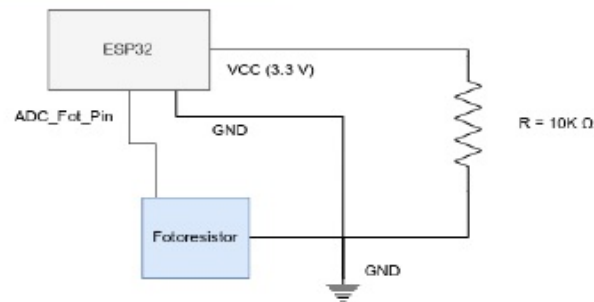


De acuerdo al pinout de ESP32 [ref] se configuraron los pines de acuerdo a la tabla a continuación.

Logic Name	Pin GPIO
BMP SDA	18
BMP280 SCL	19

#### 4.2.3. Prototipo de funcionalidad de medición de valor de luminosidad

Para el desarrollo de este prototipo se utilizó el siguiente conexionado y la librería de código ADC provista por ESP-IDF.

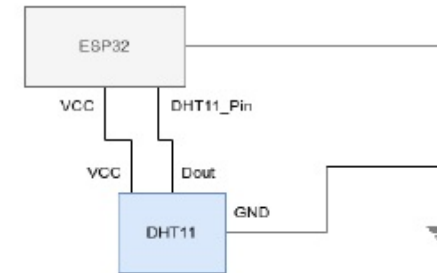


De acuerdo al pinout de ESP32 [ref] se configuraron los pines de acuerdo a la tabla a continuación.

Logic Name	Pin ADC	Pin GPIO
ADC_Fot_Pin	channel 0 - unit 2	4

#### 4.2.4. Prototipo de funcionalidad de obtencion de valores analogicos del joystick

Para el desarrollo de este prototipo se utilizó el siguiente conexionado y la librería de código ADC provista por ESP-IDF.

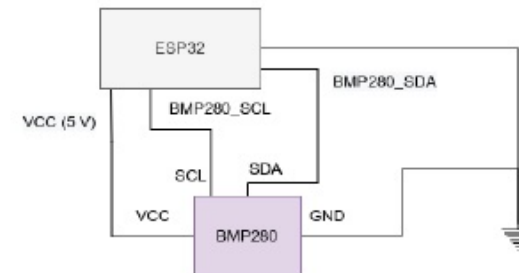


De acuerdo al pinout de ESP32 [ref] se configuraron los pines de acuerdo a la tabla a continuación.

Logic Name	Pin GPIO
Pin DHT11	2

#### 4.2.2. Prototipo de funcionalidad de medicion de presion

Para el desarrollo de este prototipo se utilizó el framework ESP-IDF y la librería de código ESP-IDF Components que provee el soporte para gestionar el BMP280. A continuación se puede apreciar el conexionado del prototipo.



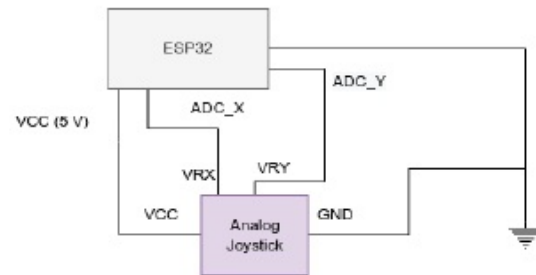
De acuerdo al pinout de ESP32 [ref] se configuraron los pines de acuerdo a la tabla a continuación.

Logic Name	Pin GPIO
BMP SDA	18
BMP280 SCL	19



## 4.3. Tests de los diferentes modulos

13



De acuerdo al pinout de ESP32 [ref] se configuraron los pines de acuerdo a la tabla a continuación.

Logic Name	Pin ADC	Pin GPIO
ADC <sub>X</sub>	channel 1 - unit 2	0
ADC <sub>Y</sub>	channel 7 - unit 1	35

## 4.2.5. Prototipo de funcionalidad de presentación de display

## 4.2.6. Prototipo de funcionalidad de control de motores DC

...

## 4.3. Tests de los diferentes modulos

...

## 4.4. Tests del producto final

...

## 4.5. Reportes de testing

...

## 4.6. Verificacion y validacion del producto

...

## 4.7. Documentacion del producto

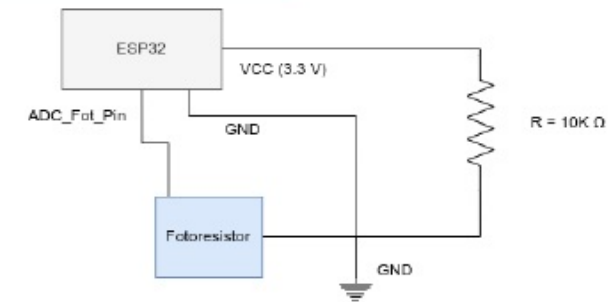
...

## 4.2. Prototipos de los diferentes modulos

13

## 4.2.3. Prototipo de funcionalidad de medición de valor de luminosidad

Para el desarrollo de este prototipo se utilizó el siguiente conexionado y la librería de código ADC provista por ESP-IDF.

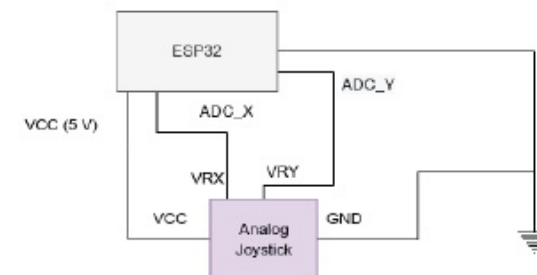


De acuerdo al pinout de ESP32 [ref] se configuraron los pines de acuerdo a la tabla a continuación.

Logic Name	Pin ADC	Pin GPIO
ADC Fot pin	channel 0 - unit 2	4

## 4.2.4. Prototipo de funcionalidad de obtencion de valores analogicos del joystick

Para el desarrollo de este prototipo se utilizó el siguiente conexionado y la librería de código ADC provista por ESP-IDF.



De acuerdo al pinout de ESP32 [ref] se configuraron los pines de acuerdo a la tabla a continuación.

Logic Name	Pin ADC	Pin GPIO
ADC X	channel 1 - unit 2	0
ADC Y	channel 7 - unit 1	35

**4.2.5. Prototipo de funcionalidad de presentación de display****4.2.6. Prototipo de funcionalidad de control de motores DC**

...

**4.3. Tests de los diferentes modulos**

...

**4.4. Tests del producto final**

...

**4.5. Reportes de testing**

...

**4.6. Verificacion y validacion del producto**

...

**4.7. Documentacion del producto**

...

## Capítulo 5

### Conclusiones

#### 5.1. Conclusiones generales

La idea de esta sección es resaltar cuáles son los principales aportes del trabajo realizado y cómo se podría continuar. Debe ser especialmente breve y concisa. Es buena idea usar un listado para enumerar los logros obtenidos.

Algunas preguntas que pueden servir para completar este capítulo:

- ¿Cuál es el grado de cumplimiento de los requerimientos?
- ¿Cuán fielmente se pudo seguir la planificación original (cronograma incluido)?
- ¿Se manifestó algunos de los riesgos identificados en la planificación? ¿Fue efectivo el plan de mitigación? ¿Se debió aplicar alguna otra acción no contemplada previamente?
- Si se debieron hacer modificaciones a lo planificado ¿Cuáles fueron las causas y los efectos?
- ¿Qué técnicas resultaron útiles para el desarrollo del proyecto y cuáles no tanto?

#### 5.2. Próximos pasos

Acá se indica cómo se podría continuar el trabajo más adelante.

## Capítulo 5

### Conclusiones

Conclusiones del trabajo...

- ¿Cuál es el grado de cumplimiento de los requerimientos?
- ¿Cuán fielmente se pudo seguir la planificación original (cronograma incluido)?
- ¿Se manifestó algunos de los riesgos identificados en la planificación? ¿Fue efectivo el plan de mitigación? ¿Se debió aplicar alguna otra acción no contemplada previamente?
- Si se debieron hacer modificaciones a lo planificado ¿Cuáles fueron las causas y los efectos?
- ¿Qué técnicas resultaron útiles para el desarrollo del proyecto y cuáles no tanto?

#### 5.1. Próximos pasos

Acá se indica cómo se podría continuar el trabajo más adelante.