

ESCUELA ESPECIALIZADA EN INGENIERÍA ITCA-FEPADE



"ROBOT DE LIMPIEZA CONTROLADO REMOTAMENTE"

TRABAJO DE INVESTIGACIÓN PARA OPTAR AL GRADO
DE INGENIERO EN MECATRÓNICA

POR

JOSÉ FRANCISCO JOYA CHÁVEZ
JOSÉ LUIS MARAVILLA MARTÍNEZ
ÁLVARO RUTILIO MULATO CHÁVEZ
VICTORIA MICHELL GONZÁLEZ POCASANGRE
ANDREA MARIELA FLORES ESCALÓN

JULIO 2022

SANTA TECLA, LA LIBERTAD, EL SALVADOR, C.A.

**ESCUELA ESPECIALIZADA EN INGENIERÍA ITCA-FEPADE
ESCUELA DE INGENIERÍA EN MECATRÓNICA**

NOMBRE DE AUTORIDADES

RECTORA

ELSY ESCOLAR SANTODOMINGO

VICERRECTOR ACADEMICO

CARLOS ALBERTO ARRIOLA MARTÍNEZ

DIRECTOR DE ESCUELA

MARIO ALFREDO MAJANO

COORDINADOR INGENIERÍA EN MECATRÓNICA

BLADIMIR ARNOLDO ALVARENGA HENRÍQUEZ

ASESOR DE TRABAJO DE INVESTIGACIÓN

RENE MAURICIO HERNANDEZ ORTIZ

ESCUELA ESPECIALIZADA EN INGENIERÍA ITCA-FEPADE
ESCUELA DE INGENIERÍA EN MECATRÓNICA

ACTA DE APROBACIÓN DE PROYECTO DE GRADUACIÓN

En la escuela de ingeniería en mecatrónica de la Escuela Especializada en Ingeniería ITCA-FEPADE, en la mañana del viernes 22 de julio de 2022.

Reunidos los suscritos miembros del Jurado Examinador del Trabajo de Graduación titulado:

"ROBOT MÓVIL DE LIMPIEZA CONTROLADO REMOTAMENTE"

Presentada por los estudiantes:

- Andrea Mariela Flores Escalón
- Victoria Michell González Pocasangre
- José Francisco Joya Chávez
- José Luis Maravilla Martínez
- Álvaro Rutilio Mulato Chávez

Para optar al grado de: Ingeniero en Mecatrónica (RESPECTIVAMENTE)

HACE CONSTAR QUE: Habiendo revisado y evaluado en forma individual su contenido escrito de conformidad con el Reglamento de Graduación.

ACORDARON DECLARARLA: APROBADO SIN OBSERVACIONES

APROBADO CON OBSERVACIONES

REPROBADO

No habiendo más que hacer constar, damos por finalizada la presente acta que firmamos, entregando el original.

ASESOR
RENE MAURICIO HERNANDEZ

JURADO 1
BLADIMIR ARNOLDO ALVARENGA

JURADO 2
EVER SIGFREDO ABREGO

Escuela Especializada en Ingeniería ITCA-FEPADE

Sistema Bibliotecario

CARTA DE CESIÓN DE DERECHOS DE AUTOR

Nosotros: José Luis Maravilla Martínez con número de DUI: 05663733-7; José Francisco Joya Chávez con número de DUI: 05538989-9; Andrea Mariela Flores Escalón con número de DUI: 05686180-2; Álvaro Rutilio Mulato Chávez con número de DUI: 05791270-3; Victoria Michell González Pocasangre con número de DUI: 05789395-9, estudiantes de la carrera de ingeniería en Mecatrónica de la Escuela Especializada en ingeniería ITCA-FEPADE

Manifestamos:

1) Que somos los autores del trabajo de investigación que lleva por título: "Robot móvil de limpieza controlado remotamente" y que en adelante denominaremos la obra, presentado como requisito de graduación de la carrera antes mencionada, el cual fue dirigido y asesorado por el ingeniero: Rene Mauricio Hernández Ortiz, quien se desempeña como docente investigador de la escuela de ingeniería en mecatrónica.

2) Que la obra es una creación original y que no infringe los derechos de propiedad intelectual, ni los derechos de publicidad, comerciales, de propiedad industrial u otros, y que no constituye una difamación, ni una invasión de la privacidad o de la intimidad, ni cualquier injuria hacia terceros.

3) Nos responsabilizamos ante cualquier reclamo que se le haga a la Escuela Especializada en ingeniería [TCA-FEPADE, en este sentido.

4) Que estamos debidamente legitimados para autorizar la divulgación de la obra mediante las condiciones de la licencia de Creative Commons.

Reconocimiento (cc by)

Reconocimiento - Compartir (cc by -sa)

Reconocimiento - SinObraDerivada (cc by -nd)

Reconocimiento - NoComercial (cc by-nc)

Reconocimiento - NoComercial - Compartirigual (cc by-nc-sa)

Reconocimiento -NoComercial-SinObraDerivada (cc by-nc-nd)

De acuerdo con la legalidad vigente.

5) Que conocemos y aceptamos las condiciones de preservación y difusión, establecidas en la Especializada en ingeniería TTCA-FEPADE.

En atención a lo antes expuesto solicitamos:

Que la obra quede depositada en las condiciones establecidas en la licencia de difusión anteriormente seleccionada, por lo tanto y con base a los artículos 5, 7 y 8 de la Ley de Propiedad Intelectual; cedemos los derechos de autor de orden patrimonial.

José Maravilla

José Joya

Andrea Flores

Álvaro Mulato

Victoria González

Santa Tecla, 22 de julio de 2022

AGRADECIMIENTOS

A ITCA.

POR HABERNOS PERMITIDO FORMARNOS, POR LA DISCIPLINA Y LA INCULCACIÓN DE VALORES, GRACIAS POR EL TIEMPO, POR PRESTAR LAS INSTALACIONES Y EQUIPOS PARA LLEVAR A CABO UN PROYECTO QUE HOY SE VE CONCRETADO.

A LA ESCUELA DE INGENIERÍA EN MECATRÓNICA.

POR LOS CONOCIMIENTOS ADQUIRIDOS A LO LARGO DE NUESTRA CARRERA POR MEDIO DE LOS DOCENTES Y POR LA OPORTUNIDAD DE CONOCER COMPAÑEROS DE ESTUDIO QUE A LO LARGO DE ESTOS AÑOS SE CONVIRTIERON EN AMIGOS.

A NUESTRO ASESOR ING. RENE MAURICIO HERNÁNDEZ.

POR LA CONFIANZA Y EL APOYO BRINDADO EN LA REALIZACIÓN DE ESTE PROYECTO.

RESUMEN

El proyecto robot de limpieza, será construido en ayuda al área de mantenimiento de limpieza del ITCA FEPADE Sede Central, el cual está basado en la implementación de tecnologías de robótica y automatización. Para la elaboración del robot de limpieza se ha considerado que tendrá movimiento y control desde una estación remota, para lo cual se utilizará un teléfono móvil que tendrá un amplio rango de alcance, ya que tendrá accesibilidad de conectarse desde cualquier área dentro de las instalaciones del ITCA FEPADE Sede Central, para posteriormente poder controlar su mecanismo.

Dentro de las funciones del robot de limpieza, se implementará un sistema de aspiración y cepillos giratorios con la capacidad de barrer superficies, a su vez tendrá un sistema de protección anticollisiones que podrá evitar un posible golpe cambiando de dirección cuando encuentre un obstáculo en su alrededor, esto será posible debido a la instalación de diferentes sensores que le permitirán detectar paredes y otros objetos.

CONTENIDO CAPITULAR

Contenido

NOMBRE DE AUTORIDADES	ii
ACTA DE APROBACIÓN DE PROYECTO DE GRADUACIÓN.....	iii
CARTA DE CESIÓN DE DERECHOS DE AUTOR.....	iv
AGRADECIMIENTOS.....	v
RESUMEN.....	vi
CAPITULO I	1
PLANTEAMIENTO DEL PROBLEMA	1
1.1. Antecedentes.....	4
1.2. Justificación.....	7
1.3. Objetivo general	8
1.4. Objetivos específicos.....	8
CAPITULO II	9
MARCO TEÓRICO.....	9
2.1. ¿Qué es un robot?.....	10
2.1.1 Robots de Limpieza.....	10
2.2. IoT.....	12
2.2.1. Diseñando aplicaciones IoT con Blynk.....	12
2.2.2. Cambios de Blynk 0.1	14
2.2.3. Productos de Blynk.....	16
2.2.4. Servicios.....	19
2.2.5. Comandos de librería Blynk.....	21
2.3. Software de Arduino.....	27
2.3.1. IDE de Arduino.....	27
2.4. Lenguaje de programación de Arduino.....	30
2.4.1. ¿Qué es un programa de Arduino?.....	31
2.5. Arduino Mega	32
2.5.1. Hoja de datos Arduino Mega.....	33
2.5.2. Data pin.....	34
2.5.2.2. Digitales	34

2.6.	Comunicación de datos Wi-Fi.....	36
2.6.1.	Ancho de banda.....	36
2.7.	ESP32.....	37
2.7.1.	Funciones ESP32.....	37
2.7.2.	Diferentes formas de programar.....	38
2.7.3.	Pinout de la placa ESP32 de 30 pines.....	40
2.7.4.	Especificaciones ESP32.....	41
2.8.	Arduino Mini Pro.....	43
2.9.	Shields para Arduino.....	46
2.9.1.	¿Qué es un shield?.....	47
2.9.2.	Motor Shield L293D.....	48
2.10.	MPU 6050.....	52
2.10.1.	I2C.....	54
2.10.2.	¿Qué es un MPU 6050?.....	57
2.10.3.	Comandos MPU 6050 con Arduino.....	57
2.11.	Sensores.....	60
2.11.1.	¿Qué es un sensor?.....	61
2.11.2.	Sensor ultrasónico.....	65
2.12.	Displays para Arduino.....	68
2.12.1.	Tipos de Display.....	69
2.13.	Baterías.....	70
2.13.1.	Tipos de batería.....	71
2.14.	Servomotor.....	73
2.14.1.	¿Cómo funciona un servomotor?.....	74
2.14.2.	Hoja de datos servomotor MG996R.....	76
2.15.	Motor DC.....	77
2.15.1.	Estructura y modo de funcionamiento del motor DC.....	78
2.15.2.	Hoja de datos motor DC.....	80
2.15.3.	Torque.....	80
CAPITULO III.....		84
ESQUEMAS DE CIRCUITOS ELECTRÓNICOS, DIBUJOS Y PLANOS DE ESTRUCTURA DE SOPORTE, CÓDIGOS DE PROGRAMACIÓN.....		84

3.1.	Esquemas de circuitos electrónicos	85
3.1.1.	Shield Arduino Mega.....	85
3.1.1.1.	Diagrama esquemático.....	85
3.1.1.2.	Diseño del circuito impreso (PCB)	85
3.1.1.3.	Diseño del circuito impreso (PCB)	86
3.1.2.1.	Diagrama esquemático.....	86
3.1.2.2.	Diseño del circuito impreso (PCB).....	87
3.1.2.3.	Shield entrada y salida Arduino Mega: Diseño del circuito impreso (PCB)	87
3.1.3.	Regulador de 12 a 5V.....	87
3.1.3.1.	Diagrama esquemático	87
3.1.3.2.	Diseño del circuito impreso (PCB)	88
3.1.3.3.	Diseño del circuito impreso (PCB)	88
3.1.4.	Shield para aspiradora.....	88
3.1.4.1.	Diagrama esquemático	88
3.1.4.2.	Diseño del circuito impreso (PCB)	89
3.1.4.3.	Diseño del circuito impreso (PCB)	89
3.1.5.	Shield Pro Mini OLED.....	89
3.1.5.1.	Diagrama esquemático	89
3.1.5.2.	Diseño del circuito impreso (PCB)	90
3.1.5.3.	Diseño del circuito impreso (PCB)	90
3.2.	Códigos de programación.....	91
3.2.1.	Código de programación Pantalla Oled con Arduino Mini.....	91
3.2.2.	Código de programación IoT con ESP32 y Blynk.....	92
3.2.3.	Código de programación Arduino Mega y Sistema de Control.....	93
3.3.	Dibujos y planos de estructura de soporte.....	94
3.3.1.	Ensamble general	94
3.3.2.	Explotado general	95
3.3.3.	Listado general de piezas	96
CAPITULO IV.....		97
MANUALES ROBOT DE LIMPIEZA		97
4.1.	Manual de instalación Drivers CH340 Y CP210X.....	98

4.1.1. Manual de instalación Driver CH340	98
4.1.2. Manual de instalación Driver CP210X.....	101
4.2. Manual de usuario de APP Blynk	104
4.3. Manual y guía rápida de instalación cámara inalámbrica.....	110
4.5. Manual de mantenimiento	123
CAPITULO V	129
ALCANCES Y LIMITACIONES.....	129
5.1. Alcances	130
5.2. Limitaciones.....	130
CAPITULO VI	131
CONCLUSIONES Y RECOMENDACIONES.....	131
6.1. Recomendaciones	132
6.2. Conclusiones	133
BIBLIOGRAFÍA.....	134
ANEXOS.....	138
ANEXO 1.....	138
• Diseñar proyecto en Blynk io.....	138
ANEXO 2.....	142
Crear un proyecto en plataforma Blynk.....	143
Arduino conectado por USB.....	158
ANEXO 3.....	161

CAPITULO I

PLANTEAMIENTO DEL PROBLEMA

Planteamiento del problema

Desde hace muchos años en la institución ITCA FEPADE Sede Central, se ha implementado un sistema de limpieza manual, en el cual los empleados encargados de saneamiento se encargan de realizar esta función en horarios específicos de limpieza.

La importancia de la limpieza en todos los entornos es algo esencial, la actividad de la limpieza es una acción de las más importantes e imprescindibles que puedan realizarse en todos los espacios que habitamos, sean del tipo que sean y en cualquier ámbito al que nos podamos referir, doméstico, laboral, público, es esencial para la buena salud de las personas.

En todas las partes del mundo las organizaciones para la salud prestan especial atención a la importancia de la limpieza en todos los entornos, ya que consideran que el mantener ambientes limpios e higiénicos en cualquier tipo de establecimiento de una manera permanente, es uno de los elementos más importantes y esenciales para la salud de las personas que los habitan. La limpieza es la primera acción que nos puede asegurar evitar la presencia de bacterias, parásitos y virus, que puedan ser contaminantes para las personas, así como la acumulación de cualquier tipo de suciedad o basura.

Actualmente la institución ITCA FEPADE Sede Central, realiza contrataciones con una empresa en el área de limpieza para poder realizar estas labores en todo el campus, en las aulas y en otras áreas, lo cual le genera un costo demasiado alto a la institución para la realización de esta acción, además de eso como sabemos, el esfuerzo humano para realizar una misma acción no es muy exacto, debido a problemas como cansancio, aburrimiento, fatiga, desesperación y otras acciones que se dan en el ámbito laboral al realizar una acción demasiadas veces por un humano.

El coronavirus en el mundo vino a revolucionar la manera de aseo, por lo cual hoy en día es muy importante la higiene en los ambientes de trabajo y de estudio, para evitar enfermedades como el Covid-19, y es por ello por lo que hay que tener sistemas de limpieza más eficaces.

La limpieza como lo repetimos anteriormente es algo fundamental para la salud, tanto como para los estudiantes, docentes y empleados, por lo cual la realización correcta de ello debe ejecutarse de manera idónea y evitar un porcentaje de error a la hora de realizar la limpieza, debido a la acumulación de bacterias.

¿Es posible construir un robot con la capacidad de limpiar pasillos y que sea comandado a larga distancia?

1.1. Antecedentes

La automatización ha revolucionado aquellas áreas en las que se ha introducido, y apenas hay un aspecto de la vida moderna que no se haya visto afectado por ella.

El término se usa ampliamente en un contexto de fabricación, pero también se aplica fuera de la fabricación en relación con una variedad de sistemas en los que hay una sustitución significativa de la acción mecánica, eléctrica o computarizada por el esfuerzo y la inteligencia humanos.

La tecnología de automatización ha madurado hasta un punto en el que se han desarrollado otras tecnologías a partir de ella y han logrado un reconocimiento y un estatus propio. La robótica es una de estas tecnologías; es una rama especializada de la automatización en la que la máquina automatizada posee ciertas características antropomórficas o similares a las humanas.

La robótica es una tecnología de automatización que ha recibido una atención considerable desde aproximadamente 1960. se basa en dos tecnologías relacionadas: control numérico y teleoperadores. El control numérico (NC) es un método para controlar los ejes de las máquinas herramienta por medio de números que han sido codificados en cinta de papel perforado u otro medio. Fue desarrollado a finales de la década de 1940 y principios de la de 1950. La primera máquina herramienta de control numérico se demostró en 1952 en los Estados Unidos en el Instituto Tecnológico de Massachusetts (MIT). La investigación posterior en el MIT condujo al desarrollo del Lenguaje APT (Herramientas Programadas Automáticamente) para la programación de máquinas herramienta.

Un teleoperadores un manipulador mecánico que es controlado por un humano desde una ubicación remota. El trabajo inicial en el diseño de teleoperadores se remonta al manejo de materiales radiactivos a principios de la década de 1940. En una implementación típica, un ser humano mueve un brazo y una mano mecánicas en un lugar, y el manipulador duplica estos movimientos en otro lugar.

La robótica industrial puede considerarse una combinación de tecnologías de control numérico y teleoperador. El control numérico proporciona el concepto de una máquina industrial programable, y la tecnología de teleoperadores aporta la noción de un brazo mecánico para realizar un trabajo útil. El primer robot industrial se instaló en 1961 para descargar piezas de una operación de fundición a presión. Su desarrollo se debió en gran parte a los esfuerzos de los estadounidenses.

Además de las aplicaciones de fabricación de la tecnología de automatización, ha habido logros significativos en áreas como las comunicaciones, el transporte, las industrias de servicios y los productos de consumo.

Los productos de consumo que van desde automóviles hasta pequeños electrodomésticos se han automatizado en beneficio del usuario. Los hornos de microondas, las lavadoras, las secadoras, los refrigeradores, las grabadoras de video y otros electrodomésticos modernos suelen contener un microprocesador que funciona como controlador de computadora para el dispositivo.

Basado en los ejemplos anteriores cabe mencionar el uso de la tecnología de automatización para la invención de los robots de limpieza. Los primeros indicios de los robots de limpieza surgen en 1996 con la aparición de la primera aspiradora robótica, elaborada por una empresa sueca fabricante de electrodomésticos.

Funcionó bastante bien, pero tenía problemas de colisión con objetos y como se detenía a poca distancia de las paredes y otros objetos, dejaba pequeñas áreas que no se limpiaban. Su producción está discontinuada. La empresa de tecnología Dyson de Gran Bretaña dio a conocer su variante DC06 en 2001, pero era demasiado costosa y nunca entró en producción en masa.

Como sugieren los datos y la información mencionada y como estudiantes de ingeniería en mecatrónica, nos hemos enfocado en aplicar los conocimientos adquiridos en automatización, para implementar nuestro proyecto de diseño y construcción de un robot móvil de limpieza controlado remotamente, de forma que este pueda ser utilizado en las instalaciones de ITCA-FEPADE Sede Central.

El uso del robot de limpieza aportará diferentes beneficios y ventajas de forma general ya que la limpieza es un factor muy importante en los diferentes edificios y áreas de las instalaciones de la Sede Central de ITCA FEPADE y es una actividad que se realiza a diario por un personal designado.

Se ahorrará mucho tiempo con la ayuda de un robot de limpieza controlado remotamente, ya que, una vez realizada la configuración, el robot será capaz de realizar el trabajo de limpieza, lo cual optimizará el tiempo para que el personal pueda hacer otras actividades de aseo a lo largo de la jornada.

1.2. Justificación

El uso de tecnologías ha venido a revolucionar la vida cotidiana del ser humano, los avances tecnológicos se han vuelto una parte esencial en nuestras vidas, como sistemas automatizables de limpieza que nos ayudan en las labores cotidianas.

Hoy en día tener un robot de limpieza no es algo extraño, las empresas dedicadas a la robótica se han encargado de hacer sistemas automáticos para mejorar la calidad de vida del ser humano, entre ellos se encuentran los robots de limpieza, los cuales tienen capacidades de poder limpiar áreas enteras como casas, oficinas, laboratorios, etc. En respuesta al requerimiento actual se plantea el diseño y creación de un robot móvil que tenga la capacidad de hacer la limpieza en los edificios internos del campus de ITCA FEPADE Santa Tecla, con el objetivo de minimizar el recurso humano, dándole una mayor autonomía, y eficacia al sistema actual de aseo interno.

La implementación de un robot de limpieza en el campus del ITCA FEPADE, no solo vendría a beneficiar aspectos económicos, sino también, en la calidad de higiene del suelo, donde transitan muchos estudiantes como docentes, dando una facilidad en la forma de operabilidad del robot, y evitar contagios de enfermedades del personal de limpieza que diariamente se encuentra expuesta a diferentes variantes de virus. El robot de limpieza tendrá la capacidad de moverse en todo el campus, de esta manera podremos garantizar que la limpieza no sólo se realice en una sección, sino también, en los edificios correspondientes según el operario tenga indicado. Este proyecto debido a su tecnología le permitirá al operador poder controlar el robot desde cualquier punto con acceso a internet, para posteriormente ejecutar las órdenes programadas. El mundo poco a poco va evolucionando de una manera autónoma, y es de mucha importancia ser parte de este proceso.

1.3. Objetivo general

- Diseñar y construir un robot móvil de limpieza e implementar tecnologías de control inalámbrico para la movilidad del robot dentro del Campus ITCA FEPADE Santa Tecla.

1.4. Objetivos específicos

- Implementar un sistema comunicación entre el robot y el dispositivo de mando mediante la utilización y configuración de la plataforma blynk.
- Realizar un programa por medio de la plataforma Arduino con la capacidad de controlar los sistemas internos del robot como actuadores, sensores y motores.
- Implementar un sistema electrónico de control de carga para la administración de batería del robot.
- Rediseñar planos de estructura mecánica y de ensamble de piezas del robot.
- Implementar manual de uso y manual de mantenimiento del proyecto.
- Diseñar e implementar un sistema de control atractivo y fácil para la manipulación del accionamiento del robot por medio de una app móvil.

CAPITULO II

MARCO TEÓRICO

2.1. ¿Qué es un robot?

Máquina automática programable capaz de realizar determinadas operaciones de manera autónoma y sustituir a los seres humanos en algunas tareas, en especial las pesadas, repetitivas o peligrosas; puede estar dotada de sensores, que le permiten adaptarse a nuevas situaciones.

2.1.1 Robots de Limpieza

Los robots de limpieza cuentan con programación inteligente y un limitado sistema de limpieza. Pueden incluir operación manual a través de controles remotos o modo automático. Sus características de limpieza varían según diferentes modelos.

Modelo	Año	Descripción
Electrolux AB "Trilobite" 	1996	Tenían buen funcionamiento, pero chocaban con objetos, se detenían cerca de las paredes y dejaban algunas áreas sin limpiar.
Dyson "DC06" 	2001	Nunca se construyó en masa debido a su alto costo de producción.

<p>iRobot "Roomba"</p> 	<p>2002</p>	<p>Podía cambiar de dirección cuando se encontró con un obstáculo, detectó puntos sucios en el piso y detectó caídas pronunciadas para evitar que se cayera por las escaleras.</p>
<p>iRobot "Scooba"</p> 	<p>2005</p>	<p>Es el opuesto al Roomba ya que en lugar de aspirar friega los suelos. Su objetivo era simplificar las tareas de los consumidores.</p>
<p>A partir de 2002 empezaron a aparecer variantes de los modelos anteriores.</p>		
<p>Dyson "Dyson 360 eye"</p> 	<p>2015</p>	<p>Equipado con una cámara de 360 grados proporciona una mejor navegación.</p>
<p>Roborock S6 MaxV</p> 	<p>2020</p>	<p>Este modelo emplea inteligencia artificial.</p>

2.2. IoT

La Internet de las cosas (IoT) describe la red de objetos físicos ("cosas") que llevan incorporados sensores, software y otras tecnologías con el fin de conectarse e intercambiar datos con otros dispositivos y sistemas a través de Internet. Estos dispositivos van desde objetos domésticos comunes hasta herramientas industriales sofisticadas. Con más de 7 mil millones de dispositivos IoT conectados en la actualidad, los expertos prevén que este número aumentará a 10 mil millones para los 2020 y 22 mil millones para el 2025.



2.2.1. Diseñando aplicaciones IoT con Blynk

Comenzar a diseñar aplicaciones para la Internet de las cosas (IoT) puede ser difícil debido a la multitud de opciones y posibilidades, tanto a nivel de hardware como de software, así como en lo que respecta a protocolos de comunicación y aplicaciones. Afortunadamente tenemos a Blynk, que nos facilita enormemente la tarea y nos permite introducirnos en esta tecnología de una forma simple y sencilla.



HTTP, MQTT, REST, GET, JSON, etc. La tecnología de IoT puede ser apabullante para quien recién se inicia debido al amplio abanico de posibilidades que tiene el desarrollador para llevar a la realidad su proyecto. Sin embargo, no es imprescindible zambullirse a fondo en este mar de siglas y acrónimos para poner en marcha un sencillo proyecto de IoT y comenzar a adquirir los conceptos fundamentales de la tecnología. Por suerte, podemos contar con herramientas muy sencillas de utilizar que nos permiten poner en funcionamiento un proyecto en muy poco tiempo,

sin necesidad de adentrarnos en los detalles finos. Una de estas herramientas es Blynk.

¿Qué es Blynk?

Blynk es una plataforma que provee soluciones para el desarrollo de aplicaciones de IoT. Su funcionamiento se basa en una app, que puede utilizarse tanto en teléfonos Android como iOS, que se comunica con el hardware a través de los servidores de Blynk. La app permite diseñar un “panel de control” con una amplia variedad de elementos que posibilitan enviar y recibir información hacia y desde el hardware, el que debe incluir alguna de las numerosas librerías especiales provistas por Blynk, disponibles para múltiples dispositivos tales como Arduino, Raspberry o placas basadas en ESP8266 y ESP32. Estas librerías resuelven la mayor parte de los detalles de la comunicación con la app, permitiendo que el desarrollador se enfoque en los detalles de su proyecto.



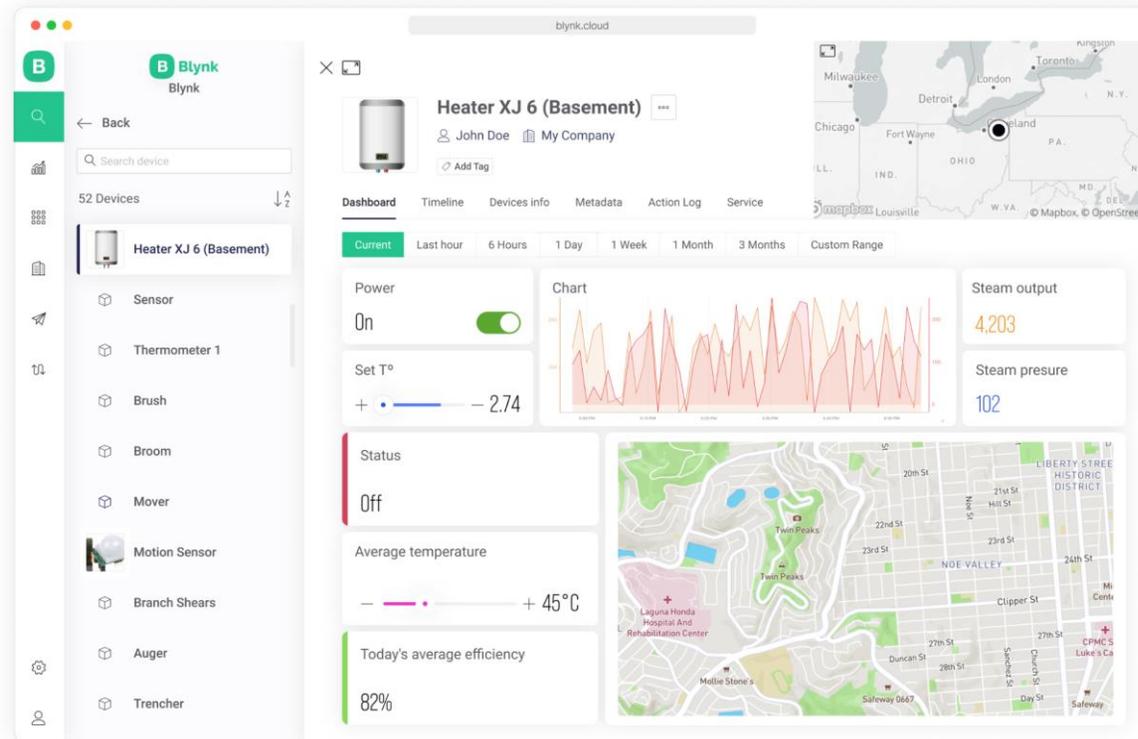
2.2.2. Cambios de Blynk 0.1

Rasgos	Diferencias
Apps	
Web Dashboard	Nuevo Blynk.Console
Android App	-Legacy Blynk requiere Android 4.2 o posterior -Blynk IoT requiere Android 5 o posterior
iOS App	-Legacy Blynk requiere iOS 9 o posterior -Blynk IoT requiere iOS 14.1 o posterior
Gestión de dispositivos	
Asistentes de voz	Integración de Alexa y Google Home (próximamente)
Aprovisionamiento y creación de perfiles de dispositivos	Nuevo en Blynk.Edgent y Blynk.Apps
Actualizaciones de firmware por aire (Blynk.Air)	Nuevo en Blynk.Edgent y Blynk.Console
El registro de eventos	Nuevo en Blynk.Console y Blynk.Apps
Gestión de notificaciones	Nuevo en Blynk.Console
Gestión de dispositivos y usuarios	Nuevo en Blynk.Console
Roles y permisos de usuario	Nuevo en Blynk.Console
Uso compartido de dispositivos (mediante código QR)	Reemplazado por invitar a los usuarios
Conectividad	
Bluetooth 2.0 SPP	No disponible en nuevo Blynk (todavía)
Bluetooth de baja energía (BLE)	No disponible en nuevo Blynk (todavía)
Widgets	
16 nuevos widgets: botón de icono, botón de imagen, control deslizante de pasos, control deslizante de pasos verticales, control de pendiente, botón de interruptor,	Solo en Blynk 2.0

control deslizante de nivel, control deslizante de nivel con interruptor, indicador radial, indicador mejorado, rampa de gradiente, icono, botón de enlace, botón de enlace de imagen, alias	
Widget de puente	Reemplazado por Automatizaciones y Rule Engine (próximamente)
Widget de eventos	Reemplazado por Automatizaciones. Ahora funciona en iOS y Android
Widget de temporizador	Reemplazado por automatizaciones
Widget de RTC	No se necesita widget. Usar API de tiempo
Widget de gancho web	Reemplazado por la configuración de WebHook en Blynk.Console
Notificaciones, correo electrónico, widgets de Twitter	Reemplazado por eventos de plantilla
Widget de terminal	Funciona solo en tiempo real. Historial no compatible
Widget de tabla	No disponible en nuevo Blynk (todavía)
Selector de dispositivos	Reemplazado con mosaicos de dispositivos
Proyectos multidispositivo	Reemplazado con mosaicos de dispositivos y tableros (próximamente)
Informes	Movido a Blynk.Console Informes programados aún no disponibles
Propiedad de conjunto mínimo/máximo de widget	Los campos mínimo/máximo del widget se movieron a la configuración del flujo de datos
Nube	
Servidor local	Disponible sólo en el plan de marca blanca
Servidor empresarial privado	Sigue disponible para clientes de Marca Blanca
Modelo de precios	
Blynk Energy	Reemplazado con planes de suscripción

2.2.3. Productos de Blynk

2.2.3.1 Blynk.Console



Blynk.Console es una aplicación web rica en funciones que puede ser utilizada por diferentes tipos de usuarios. La funcionalidad principal de este producto es:

1. Configuración del funcionamiento de los dispositivos conectados en la plataforma + ajustes de la aplicación.
2. Gestión de dispositivos, sus datos, usuarios, organizaciones y ubicaciones.
3. Supervisión y control remoto de dispositivos.

2.2.3.2. Blynk.Apps



Blynk.Apps es una aplicación móvil nativa multifuncional para iOS y Android que cumple estas funciones principales:

1. Monitoreo y control remoto de dispositivos conectados que funcionan con la plataforma Blynk.
2. Configuración de la interfaz de usuario móvil durante las etapas de creación de prototipos y producción.
3. Automatice el trabajo de los dispositivos conectados.

Blynk.apps es una solución de marca blanca.

2.2.3.3. Blynk.Edgent

Edge + Agent = Edgent

```
main.cpp
1 // Fill-in information from your Blynk Template here
2 #define BLYNK_TEMPLATE_ID      ""
3 #define BLYNK_DEVICE_NAME     ""
4 #define BLYNK_FIRMWARE_VERSION "0.1.0"
5 #define APP_DEBUG
6
7 #include "BlynkEdgent.h"
8
9 void setup()
10 {
11     delay(100);
12     BlynkEdgent.begin();
13 }
14
15 void loop() {
16     BlynkEdgent.run();
17 }
```

Blynk.Edgent es una biblioteca integrada liviana que se ejecuta en más de 400 modelos de hardware compatibles

Las funciones principales de Blynk.Edgent son:

1. Reclamación de dispositivos y aprovisionamiento de Wi-Fi (poner el dispositivo en línea y autenticarlo con un usuario determinado).
2. Gestión de conectividad (WiFi, Celular, Ethernet).
3. Transferencia de datos entre el dispositivo y la nube.
4. API para trabajar con características específicas de Blynk.Apps y Blynk.cloud
5. Actualizaciones de firmware inalámbricas (para modelos de hardware seleccionados).

2.2.3.4. Blynk.Cloud

Blynk.Cloud es una infraestructura de servidor: el corazón de la plataforma Blynk IoT. La nube es responsable de unir todos los componentes de la plataforma.

Blynk también ofrece servidores privados.

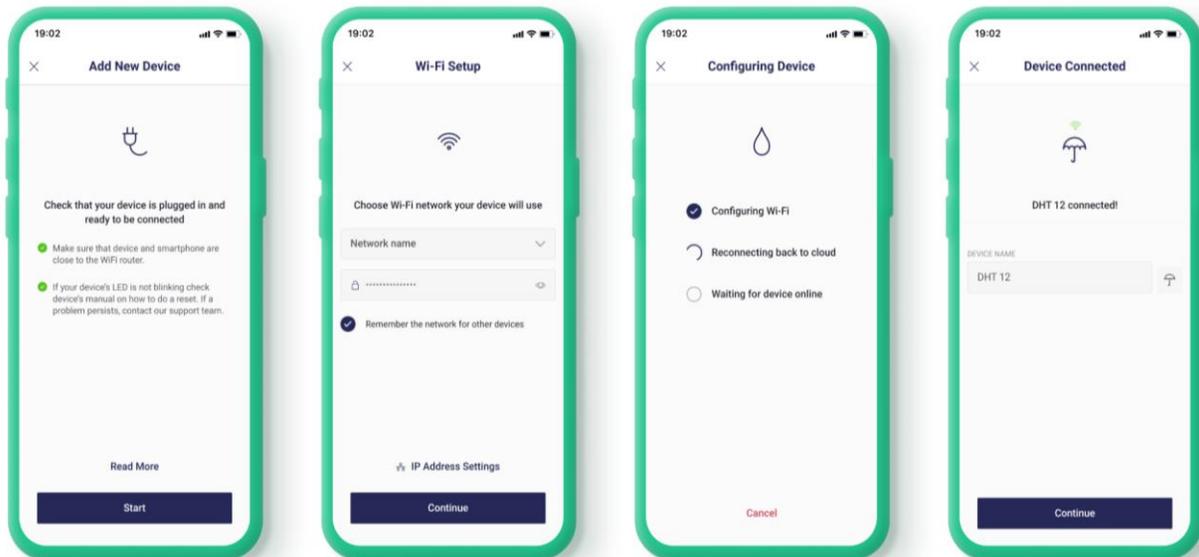
2.2.4. Servicios

Los servicios (o microservicios) de Blynk son módulos de software que funcionan en todos los productos y realizan un determinado conjunto de funciones.

2.2.4.1 Blynk.Inject

Un microservicio para:

- Reclamación de propiedad de dispositivos por parte de usuarios y organizaciones.
- Aprovisionamiento de dispositivos con credenciales WiFi para que puedan conectarse a las redes WiFi del usuario final.
- Gestión de tokens de autenticación
- Flujo de UX para guiar a los usuarios finales a través del proceso de reclamo y aprovisionamiento con la ayuda de Blynk.Apps

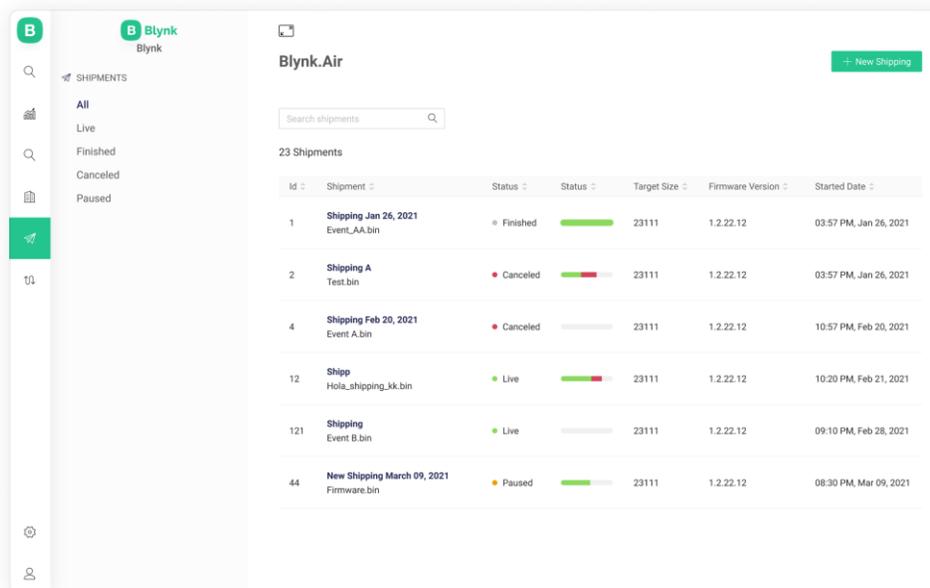


2.2.4.2. Blynk.R

Un microservicio para la gestión de usuarios, que cubre:

- Registro de usuarios con invitaciones.
- Gestión de roles y permisos.
- Multi Alquiler.

2.2.4.3. Blynk.Air



The screenshot displays the Blynk.Air dashboard. On the left, there is a sidebar with a search icon and a list of filters: SHIPMENTS, All, Live, Finished, Canceled, and Paused. The main content area is titled 'Blynk.Air' and features a search bar for shipments. Below the search bar, it indicates '23 Shipments' and displays a table with the following columns: Id, Shipment, Status, Status (with a progress bar), Target Size, Firmware Version, and Started Date. The table contains six rows of shipment data.

Id	Shipment	Status	Status	Target Size	Firmware Version	Started Date
1	Shipping Jan 26, 2021 Event_AA.bin	Finished	<div style="width: 100%;"></div>	23111	1.2.22.12	03:57 PM, Jan 26, 2021
2	Shipping A Test bin	Canceled	<div style="width: 50%;"></div>	23111	1.2.22.12	03:57 PM, Jan 26, 2021
4	Shipping Feb 20, 2021 Event A.bin	Canceled	<div style="width: 0%;"></div>	23111	1.2.22.12	10:57 PM, Feb 20, 2021
12	Shipp Hola_shipping_kk.bin	Live	<div style="width: 50%;"></div>	23111	1.2.22.12	10:20 PM, Feb 21, 2021
121	Shipping Event B.bin	Live	<div style="width: 0%;"></div>	23111	1.2.22.12	09:10 PM, Feb 28, 2021
44	New Shipping March 09, 2021 Firmware bin	Paused	<div style="width: 50%;"></div>	23111	1.2.22.12	08:30 PM, Mar 09, 2021

Un microservicio para actualizaciones inalámbricas de firmware responsable de:

- Gestión de campañas de envío de actualizaciones inalámbricas de firmware.

Instalación de nuevo firmware en los dispositivos perimetrales.
(Blynk.io, s.f.)

2.2.5. Comandos de librería Blynk

Comando Blynk	Descripción
Configuración	
Blynk.begin()	<p>Tiene múltiples parámetros para diferentes modelos de hardware y también depende del tipo de conexión. Siga los bocetos de ejemplo para su modelo de hardware específico.</p> <ol style="list-style-type: none"> 1. Conexión a la red (WiFi, Ethernet, etc.) 2. Llamada de Blynk.config(...) para configurar el token de autenticación, la dirección del servidor, etc. 3. Intenta conectarse al servidor una vez (puede bloquearse durante más de 30 s)
Blynk.config()	<p>Le permite administrar la conexión de red. Puede configurar su tipo de conexión (WiFi, Ethernet, ...) usted mismo y luego llamar a:</p> <pre>Blynk.config(auth, server, port);</pre> <p>o solo</p> <pre>Blynk.config(auth);</pre> <p>NOTA: Después Blynk.config(...) de llamar, su hardware aún no está conectado al servidor. Intentará conectarse hasta que llegue a la primera instancia Blynk.run() o Blynk.connect() rutina. Para omitir la conexión al servidor o para desconectarse manualmente, llame Blynk.disconnect() después de la configuración.</p>
Blynk.connectWiFi(ssid, pass);	<p>Úselo para configurar convenientemente la conexión WiFi:</p> <p>Nota: para conectarse a redes WiFi abiertas, establezca pass en una cadena vacía (" ").</p>

Gestión de conexiones	
Blynk.connect()	<p>Estas funciones seguirán intentando conectarse al servidor Blynk. Vuelve true cuando está conectado, false si se ha alcanzado el tiempo de espera. El tiempo de espera predeterminado es de 30 segundos.</p> <pre>bool result = Blynk.connect();</pre>
Blynk.disconnect()	Desconecta el hardware del servidor Blynk
Blynk.connected()	Regresa true cuando el hardware está conectado al servidor Blynk, false si no hay una conexión activa al servidor Blynk.
Blynk.run()	<p>Esta función debe llamarse con frecuencia para procesar los comandos entrantes y realizar el mantenimiento de la conexión Blynk.</p> <p>Por lo general, se llama en void loop() {}.</p>
Virtual pins control	
<pre>BLYNK_WRITE(V0) { }</pre>	<p>Función especial que se activa automáticamente cada vez que el servidor le dice a su dispositivo que el valor de su pin virtual ha cambiado. Este cambio normalmente ocurriría cuando se presiona el botón del widget en la aplicación.</p> <p>No se puede usar dentro de ningún ciclo o función. Es una función independiente.</p> <p>Nota: las variable dentro de esta función, no será posible utilizar afuera, excepto utilizar el comando return;</p>

<pre>int virtual_pin_value = param.asInt();</pre>	<p>Obtiene el parámetro de valor del pin virtual y lo almacena en la variable entera local llamada virtual_pin_value.</p> <ul style="list-style-type: none"> - param.asInt() para variable tipo entero. - param.asFloat() para variable tipo flotante. - param.asDouble() para variable tipo doble. - param.asStr() para variable tipo cadena. <p>Nota: las variable dentro de esta función, no será posible utilizar afuera, excepto utilizar el comando <code>return virtual_pin_value;</code></p>
<pre>Blynk.virtualWrite(vPin, value)</pre>	<p>La llamada virtualWrite intenta enviar el valor a la red inmediatamente.</p> <p>Nota:</p> <ul style="list-style-type: none"> - Use BlynkTimer cuando use este comando para enviar datos. De lo contrario, su hardware se desconectará del servidor. - Para pines virtuales con números > 127, la V128 sintaxis no está disponible. Utilice un número de PIN virtual simple, por ejemplo: <code>Blynk.virtualWrite(128, "abc");</code>
<pre>BLYNK_WRITE_DEFAULT()</pre>	<p>Redefine el controlador para todos los pines que no están cubiertos por BLYNK_WRITE funciones personalizadas.</p> <pre>BLYNK_WRITE_DEFAULT(){ int pin = request.pin; int value = param.asInt(); }</pre>

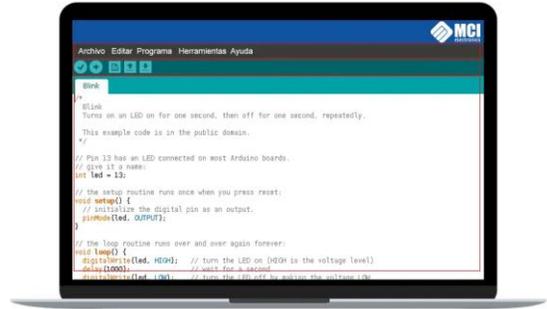
<pre>Blynk.syncAll()</pre>	<p>Solicita todos los valores más recientes almacenados en el servidor para todos los widgets. Todos los valores y estados de los pines analógicos/digitales/virtuales se establecerán en el último valor almacenado. Cada pin virtual generará un evento BLYNK_WRITE().</p> <pre>BLYNK_CONNECTED() { Blynk.syncAll(); }</pre>
<pre>Blynk.syncVirtual(vPin)</pre>	<p>Este comando actualiza el PIN virtual individual al último valor almacenado en el servidor. Cuando se usa, BLYNK_WRITE se llama al controlador correspondiente.</p> <pre>Blynk.syncVirtual(V0);</pre> <p>Para actualizar varios pines, use:</p> <pre>Blynk.syncVirtual(V0, V1, V6, V9, V16);</pre>
<pre>Blynk.setProperty(vPin, "property", value)</pre>	<p>Este comando permite cambiar las propiedades del widget.</p> <p>También se admite el cambio de algunas de las propiedades del widget desde el lado del hardware.</p> <p>Por ejemplo, puede cambiar el color del widget LED según una condición:</p> <pre>//change LED color Blynk.setProperty(V0, "color", "#D3435C");</pre> <pre>//change LED label Blynk.setProperty(V0, "label", "My New Widget Label");</pre> <pre>//change MENU labels Blynk.setProperty(V0, "labels", "Menu Item 1", "Menu Item 2", "Menu Item 3");</pre>

Temporizador	
BlynkTimer	<p>Es importante enviar datos en intervalos y mantener el ciclo void() lo más limpio posible. BlynkTimerle permite enviar datos periódicamente con intervalos dados sin interferir con las rutinas de la biblioteca Blynk Blynk Timer hereda SimpleTimer Library , una biblioteca bien conocida y ampliamente utilizada para cronometrar múltiples eventos en hardware. BlynkTimer está incluido en la biblioteca Blynk de forma predeterminada y no es necesario instalar SimpleTimer por separado o incluir SimpleTimer.h</p> <ul style="list-style-type: none"> - Un solo BlynkTimer objeto permite programar hasta 16 temporizadores. - Compatibilidad mejorada con tableros como Arduino 101, Intel Galileo, etc. - Cuando un temporizador tiene problemas para ejecutarse varias veces (debido a un bloqueo loop), simplemente omite todos los intervalos perdidos y llama a su función solo una vez. Esto difiere de SimpleTimer, que podría llamar a su función varias veces en este escenario.
Evento de registro	
Blynk.logEvent ()	<p>Los eventos se utilizan para rastrear y registrar eventos importantes que suceden en sus dispositivos. Los eventos también pueden desencadenar diferentes tipos de notificaciones que se pueden enviar por correo electrónico, como notificaciones automáticas a la aplicación Blynk o como SMS.</p> <ul style="list-style-type: none"> - Blynk.logEvent("hello"); - Blynk.logEvent("hello", "Hello World,");

	<p>Nota: Para crear un evento se debe configurar también en Blynk.Console:</p> <ol style="list-style-type: none"> 1. Haga clic en Agregar nuevo evento. 2. Nombra el primer evento con el nombre Hello. 3. Haga clic en Crear (el evento aparecerá en la lista de la pestaña Eventos). 4. Haga clic en Guardar para guardar y aplicar los cambios realizados (seleccione Actualizar dispositivo activo esta vez).
Reloj RTC	
<pre>Blynk.sendInternal("rtc", "sync");</pre>	<p>Para sincronizar la hora de su dispositivo con la nube de Blynk, use este comando</p> <p>Por ejemplo, puede usarlo justo después de que el dispositivo se haya conectado a la nube con el comando BLYNK_CONNECTED():</p> <pre>BLYNK_CONNECTED() { Blynk.sendInternal("rtc", "sync"); } BLYNK_WRITE(InternalPinRTC) { long t = param.asLong(); Serial.print("Unix time: "); Serial.print(t); Serial.println(); }</pre>

2.3. Software de Arduino

Para empezar a programar la placa Arduino es necesario descargar un IDE (Integrated Development Environment). El IDE es un conjunto de herramientas de software que permiten a los programadores desarrollar y grabar todo



el código necesario para hacer que nuestro Arduino funcione como queramos. El IDE de Arduino nos permite escribir, depurar, editar y grabar nuestro programa (llamados "sketches" en el mundo Arduino) de una manera sumamente sencilla, en gran parte a esto se debe el éxito de Arduino, a su accesibilidad.

2.3.1. IDE de Arduino

El área de trabajo puede ser dividida en 5 grandes partes. De arriba abajo son: la barra de menús, la barra de botones, el editor de código, la barra de consola de mensajes, y la barra de estado.



La mayoría del tiempo estaremos trabajando en el editor de código, donde desarrollaremos nuestros proyectos.

A los elementos que más les sacaremos provecho son:

Verificar: Este botón realiza dos funciones: comprueba que no haya error en nuestro código, y si no hay problemas, lo compila.

Subir: Este botón lo utilizamos luego de "Verificar". Su función es cargar en la memoria del microcontrolador el programa que hemos escrito.

Nuevo: Crea un nuevo sketch vacío.

Abrir: Despliega un menú con todos los sketches disponibles para abrir. Podemos abrir nuestros propios sketches como la gran cantidad de sketches que viene listos como ejemplos para probar, clasificados en categorías dentro del menú.

Guardar: Guarda el código de nuestro sketch en un fichero, el cual tendrá la extensión ".ino". Podemos guardar estos ficheros donde queramos, pero el IDE Arduino nos ofrece una carpeta específica para ello, la carpeta "Arduino" en "Documentos", ahí se creará una carpeta con el nombre de tu sketch, así el IDE evita que se mezclen los archivos de los distintos sketch.

Monitor Serial: Abre el monitor serial. Nos permite ver información transmitida desde nuestro Arduino por el puerto de comunicación serial

Menú contextual: Esta pestaña se ubica bajo el botón de monitor serial, nos permite abrir nuevas pestañas. Abrir nuevas pestañas es de suma utilidad cuando tenemos códigos tan largos que necesitamos dividirlo en partes para trabajar más cómodamente. Es así porque todas las nuevas pestañas abiertas forman parte del mismo proyecto que la primera pestaña original. Lo más habitual es utilizar pestañas separadas para la definición de funciones, constantes o variables globales.

En la barra de menú hallaremos opciones más tradicionales que de igual manera son útiles de conocer: "Archivo", "Editar", "Programa", "Herramientas" y "Ayuda".

Archivo: Ofrece acciones estándar como crear un nuevo documento sketch, abrir uno existente, guardarlo, cerrarlo, cerrar el IDE, etc. También añade otras funciones interesantes, como "Ejemplos" donde podemos acceder a los sketches de ejemplo que vienen de serie con el IDE y gracias a la entrada "Proyectos" podemos acceder a nuestros propios sketches guardados en las diferentes subcarpetas que hay dentro de la carpeta "Arduino".

Editar: Además de ofrecer acciones estándar como deshacer y rehacer, cortar, copiar y pegar texto, seleccionar todo el texto o buscar y reemplazar texto, podemos ver otras acciones interesantes. Por ejemplo, gracias a la entrada "copiar al foro" podemos copiar el código de nuestro sketch al portapapeles de nuestro sistema operativo en una forma que es especialmente adecuada para pegarlo directamente en la foto oficial de Arduino. También podemos "copiar como HTML" el cual copia en una forma especial para pegarlo en páginas webs genéricas.

Programa: en este menú se ofrece la acción de verificar/compilar nuestro sketch, la de abrir la carpeta donde está guardado el fichero ".ino" que se está editando en este momento, la de añadir en una nueva pestaña un nuevo fichero de código a nuestro sketch y la de importar librerías.

Herramientas: En este menú se ofrecen diferentes herramientas variadas, como la posibilidad de auto formatear el código para hacerlo más legible, la posibilidad de guardar una copia de todos los sketches del proyecto actual en formato .zip, la posibilidad de abrir el monitor serie, etc. Otras herramientas mucho más avanzadas son por ejemplo la

entrada "Programador:" que podemos usar para seleccionar un programador externo y poder grabar el sketch en memoria a través de dicho programador o "Quemar bootloader", útil cuando queramos grabar un nuevo bootloader en el microcontrolador de la placa.

Ayuda: desde este menú podemos acceder a varias secciones de la página web oficial de Arduino que contienen diferentes artículos, tutoriales y ejemplos de ayuda. No se necesita internet para consultar dichas secciones ya que esta documentación se descarga junto con el propio IDE, por lo que su acceso se realiza en forma local.

Monitor Serial: El monitor serial es una ventana del IDE que nos permite desde nuestro computador enviar y recibir datos textuales a la placa Arduino usando el cable USB (el cual utiliza la conexión serie). Para enviar datos, simplemente hay que escribir el texto deseado en la caja de texto que aparece en su parte superior y presionar el botón "enviar". Por otro lado, los datos recibidos provenientes de la placa serán mostrados en la sección central del "monitor serial".

Es importante elegir la misma velocidad de transmisión que se haya especificado en el programa ejecutado en la placa, porque si no, los caracteres transferidos no serán reconocidos correctamente.

2.4. Lenguaje de programación de Arduino

Una vez que tengamos instalado y configurado el IDE de Arduino en nuestro ordenador, es hora de empezar a escribir el código. La codificación de la plataforma Arduino no requiere ningún esfuerzo. Hay un montón de bibliotecas para hacer las cosas más fáciles.

El programa escrito en código fuente de Arduino se llama sketch. Contiene todas las instrucciones que debe realizar la placa Arduino,

las configuraciones iniciales de pines, y la inclusión de librerías entre otras cosas.

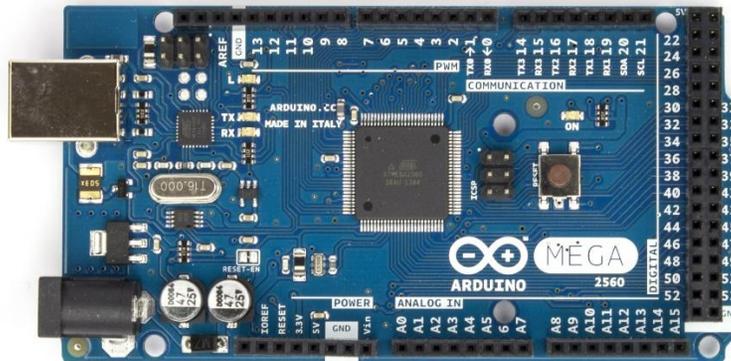
El lenguaje de programación Arduino se basa en un lenguaje de programación de hardware muy simple llamado Processing, que es muy similar al C y al C++. Pero tiene algunas funciones incorporadas además de las funciones normales utilizadas en el C común. Después de compilar el sketch en el IDE, debe ser cargado en la placa Arduino para su ejecución.

2.4.1. ¿Qué es un programa de Arduino?

El código de Arduino debe contener como mínimo dos funciones básicas: `void setup()` y `void loop()`. Estos dos bloques de código son los principales elementos de cualquier código de Arduino. Se conocen como funciones. Pronto les echaremos un vistazo más de cerca.

Todas las declaraciones se hacen en `void setup ()`. El nombre contiene la palabra `'setup'` que debería revelar su propósito. Los comandos reales para realizar y el proceso de cálculo están codificados dentro de la función `void loop ()`. El "loop" significa que el sketch se ejecutará infinitamente.

2.5. Arduino Mega



Arduino es una marca de microcontroladores mundialmente conocida, es un proyecto open Source que pone a disposición del usuario una amplia gama de dispositivos basados en el microcontrolador ATmega.

El Arduino mega es el microcontrolador más amplio y capaz de Arduino; posee 54 pines digitales que funcionan como entrada/salida, 16 entradas análogas, un cristal oscilador de 16 MHz, una conexión USB, un botón reset y una entrada para la alimentación de la placa.

La comunicación entre la computadora y Arduino se produce a través del puerto serie. Posee un convertidor USB-serie, por lo que sólo se necesita conectar el dispositivo a la computadora utilizando un cable USB como el que utilizan las impresoras.

2.5.1. Hoja de datos Arduino Mega

- **Procesador ATmega2560**
 - Hasta 16 MIPS de rendimiento a 16 MHz
 - 256 bytes (de los cuales 8k se utilizan para el gestor de arranque)
 - EEPROM de 4k bytes
 - 8k bytes de SRAM interna
 - Registros de trabajo de uso general de 32 x 8
 - Contador en tiempo real con oscilador separado
 - Cuatro canales PWM de 8 bits
 - Cuatro USART seriales programables
 - Interfaz serial SPI de controlador/periférico
- **ATmega16U2**
 - Hasta 16 MIPS de rendimiento a 16 MHz
 - Memoria Flash ISP de 16k bytes
 - EEPROM de 512 bytes
 - 512 bytes SRAM
 - USART con modo solo maestro SPI y control de flujo de hardware (RTS/CTS)
 - Interfaz serie SPI maestro/esclavo
- **Modos de suspensión**
 - Inactivo
 - Reducción de ruido ADC
 - Ahorro de energía
 - Apagado
 - En espera
 - Espera extendida
- **Energía**
 - Conexión USB
 - Adaptador externo de CA/CC
- **I/O**
 - 54 digitales
 - 16 analógico
 - 15 salida PWM

2.5.2. Data pin

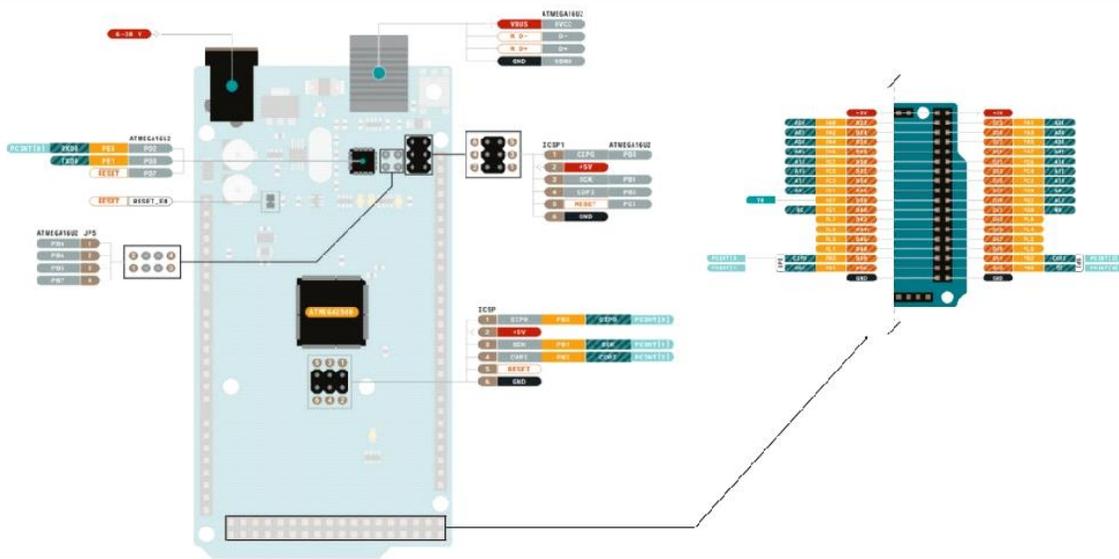
2.5.2.1. Analógicos

Pin	Function	Type	Description
1	NC	NC	Not connected
2	IOREF	IOREF	Reference for digital logic V - connected to 5V
3	Reset	Reset	Reset
4	+3V3	Power	+3V3 Power Rail
5	+5V	Power	+5V Power Rail
6	GND	Power	Ground
7	GND	Power	Ground
8	VIN	Power	Voltage Input
9	A0	Analog	Analog input 0/GPIO
10	A1	Analog	Analog input 1/GPIO
11	A2	Analog	Analog input 2/GPIO
12	A3	Analog	Analog input 3/GPIO
13	A4	Analog	Analog input 4/GPIO
14	A5	Analog	Analog input 5/GPIO
15	A6	Analog	Analog input 6/GPIO
16	A7	Analog	Analog input 7/GPIO
17	A8	Analog	Analog input 8/GPIO
18	A9	Analog	Analog input 9/GPIO
19	A10	Analog	Analog input 10/GPIO
20	A11	Analog	Analog input 11/GPIO
21	A12	Analog	Analog input 12/GPIO
22	A13	Analog	Analog input 13/GPIO
23	A14	Analog	Analog input 14/GPIO
24	A15	Analog	Analog input 15/GPIO

2.5.2.2. Digitales

Pin	Function	Type	Description
1	D21/SCL	Digital input/I2C	Digital input 21/I2C Dataline
2	D20/SDA	Digital input/I2C	Digital input 20/I2C Dataline
3	AREF	Digital	Analog Reference Voltage
4	GND	Power	Ground
5	D13	Digital/GPIO	Digital input 13/GPIO
6	D12	Digital/GPIO	Digital input 12/GPIO
7	D11	Digital/GPIO	Digital input 11/GPIO
8	D10	Digital/GPIO	Digital input 10/GPIO
9	D9	Digital/GPIO	Digital input 9/GPIO
10	D8	Digital/GPIO	Digital input 8/GPIO
11	D7	Digital/GPIO	Digital input 7/GPIO
12	D6	Digital/GPIO	Digital input 6/GPIO
13	D5	Digital/GPIO	Digital input 5/GPIO
14	D4	Digital/GPIO	Digital input 4/GPIO

15	D3	Digital/GPIO	Digital input 3/GPIO
16	D2	Digital/GPIO	Digital input 2/GPIO
17	D1/TX0	Digital/GPIO	Digital input 1/GPIO
18	D0/TX1	Digital/GPIO	Digital input 0/GPIO
19	D14	Digital/GPIO	Digital input 14/GPIO
20	D15	Digital/GPIO	Digital input 15/GPIO
21	D16	Digital/GPIO	Digital input 16/GPIO
22	D17	Digital/GPIO	Digital input 17/GPIO
23	D18	Digital/GPIO	Digital input 18/GPIO
24	D19	Digital/GPIO	Digital input 19/GPIO
25	D20	Digital/GPIO	Digital input 20/GPIO
26	D21	Digital/GPIO	Digital input 21/GPIO



Arduino Mega Pinout

2.6. Comunicación de datos Wi-Fi

La tecnología inalámbrica es la que hace posible la capacidad de comunicarse entre dos o más entidades (ordenadores o PC, Smartphone, Tv, etc.) separadas a una distancia e incluso en movimiento, sin el uso de cables de ningún tipo, ni otros medios físicos. Existen varias tecnologías inalámbricas diferentes para conseguir la comunicación inalámbrica entre aparatos.

La tecnología inalámbrica consta de un emisor de las ondas electromagnéticas que transmiten la información codificada y un receptor que recoge la información de esas ondas, la descodifica y recibe la información.

Una red de transmisión de datos vía Wi-Fi consiste en un sistema que permite que diferentes dispositivos electrónicos se conecten a las redes de comunicación a través de un punto de acceso de red inalámbrica (hotspot). Dicho punto de acceso tiene un alcance limitado, siendo mayor al aire libre que en interiores.

2.6.1. Ancho de banda

Es la máxima cantidad de datos transmitidos a través de una conexión a Internet en cierta cantidad de tiempo.

El ancho de banda por lo general se confunde con la velocidad de Internet cuando en realidad es el volumen de información que se puede enviar a través de una conexión en una cantidad medida de tiempo, calculado en megabits por segundo (Mbps).

El ancho de banda también es un concepto clave en muchas otras áreas tecnológicas. Por ejemplo, en el procesamiento de señales se usa para describir la diferencia entre las frecuencias superior e inferior en una transmisión como una señal de radio, y se mide típicamente en hercios (Hz).

2.7. ESP32

El módulo ESP32 es una solución de Wi-Fi/Bluetooth todo en uno, integrada y certificada que proporciona no solo la radio inalámbrica, sino también un procesador integrado con interfaces para conectarse con varios periféricos. Está diseñado más comúnmente para dispositivos móviles, tecnología portátil y aplicaciones IoT (Internet de las cosas).

Ahora hay varios modelos diferentes, los cuales están disponibles con conectividad Wi-Fi y Bluetooth combinada, o sólo conectividad Wi-Fi.

2.7.1. Funciones ESP32

ESP32 tiene muchas aplicaciones cuando se trata de Internet de las cosas (IoT). Estas son solo algunas de las funciones de IoT para las que se utiliza el chip:

- **Redes:** la antena Wi-Fi del módulo y el doble núcleo permiten que los dispositivos integrados se conecten a enrutadores y transmitan datos.
- **Procesamiento de datos:** incluye el procesamiento de entradas básicas desde sensores analógicos y digitales hasta cálculos mucho más complejos con un RTOS o un SDK sin sistema operativo.
- **Conectividad P2P:** crea una comunicación directa entre diferentes ESP y otros dispositivos mediante la conectividad IoT P2P.
- **Servidor Web:** Accede a páginas escritas en HTML o lenguajes de desarrollo.

2.7.1.1. Aplicaciones ESP32

Los módulos ESP32 se encuentran comúnmente en los siguientes dispositivos IoT:

- Dispositivos industriales inteligentes, incluidos los controladores lógicos programables (PLC).
- Dispositivos médicos inteligentes, incluidos monitores de salud portátiles.
- Dispositivos de energía inteligentes, incluidos HVAC y termostatos.
- Dispositivos de seguridad inteligentes, incluidas cámaras de vigilancia y cerraduras inteligentes.

2.7.2. Diferentes formas de programar

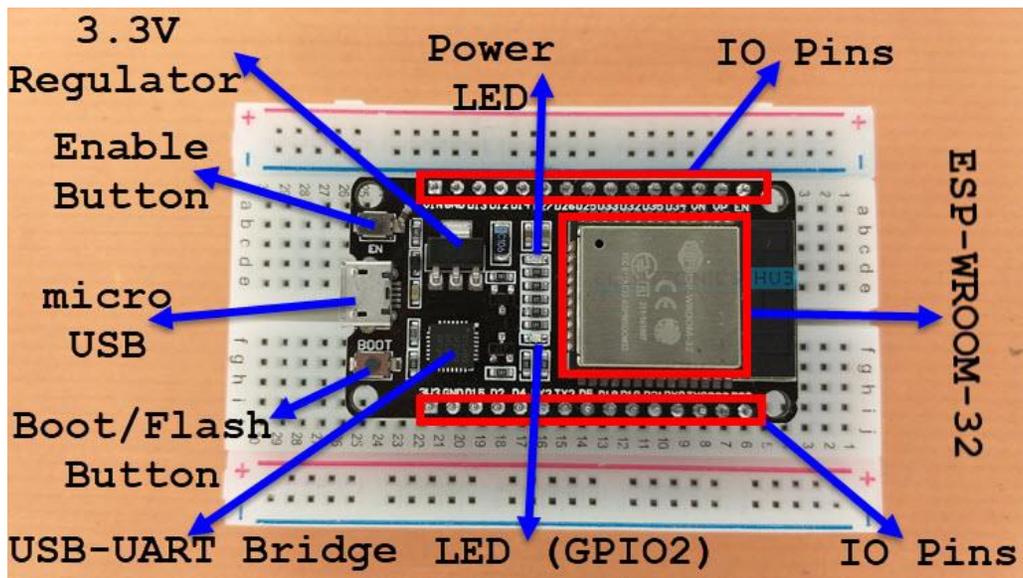
Un buen hardware como el ESP32 será más fácil de usar si se puede programar (escribir código) en más de una forma. El ESP32 admite múltiples entornos de programación.

Algunos de los entornos de programación comúnmente utilizados son:

- IDE de Arduino
- PlatformIO IDE (código VS)
- lua
- micropitón
- Espressif IDF (marco de desarrollo de IoT)
- JavaScript

Disposición

La siguiente imagen muestra el diseño de una placa de desarrollo ESP32

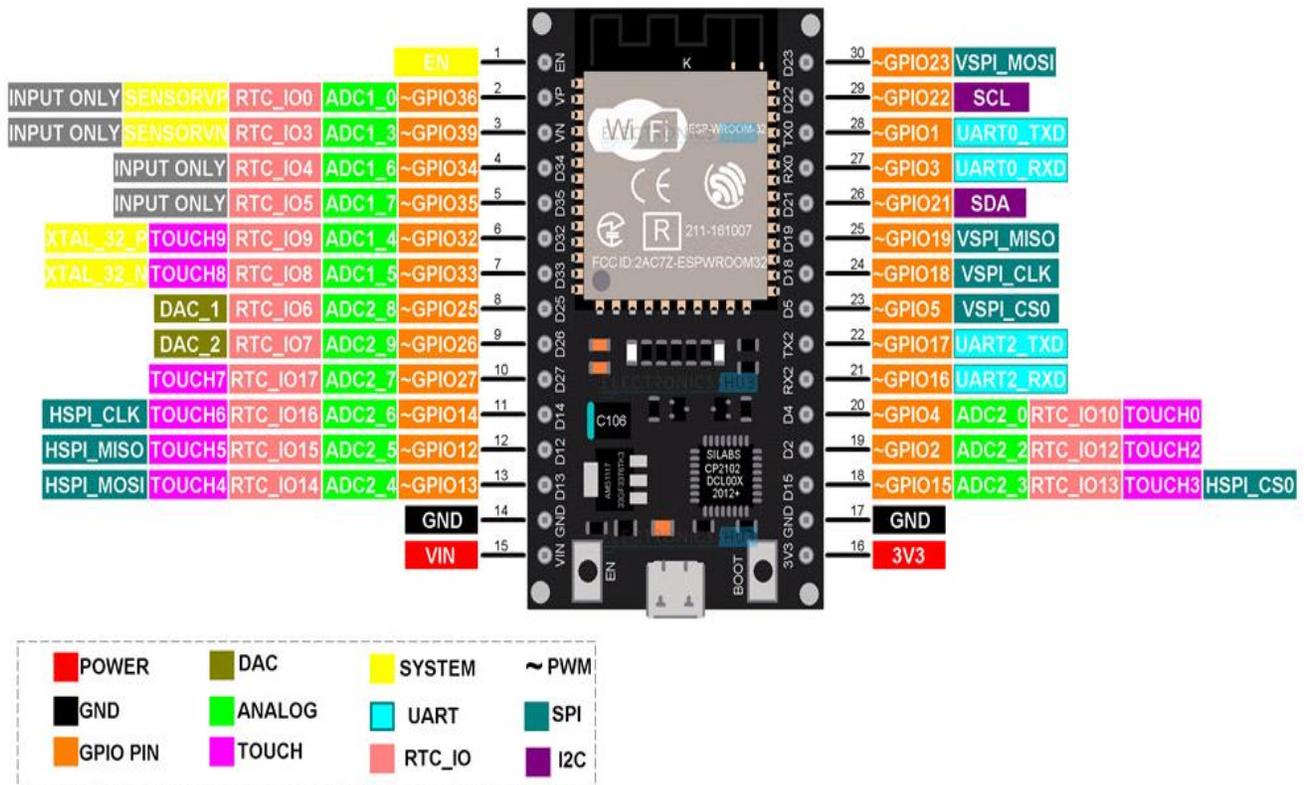


La placa ESP32 consta de lo siguiente:

- Módulo ESP-WROOM-32
- Dos filas de pines IO (con 15 pines en cada lado)
- CP2012 USB-Puente UART IC
- Conector micro-USB (para alimentación y programación)
- Circuito integrado regulador AMS1117 3.3V
- Botón Habilitar (para Restablecer)
- Botón de arranque (para flashear)
- LED de encendido (rojo)
- LED de usuario (azul, conectado a GPIO2)
- Algunos componentes pasivos

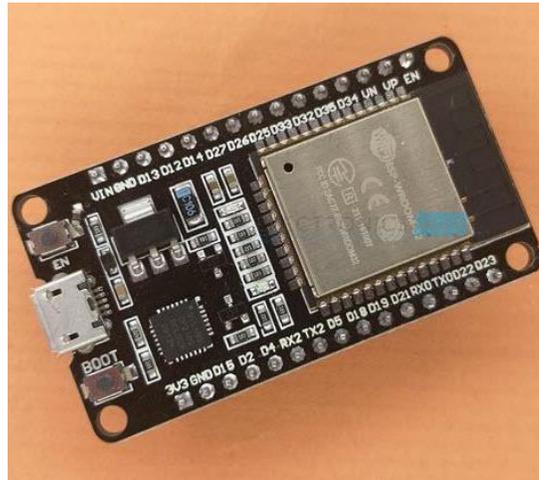
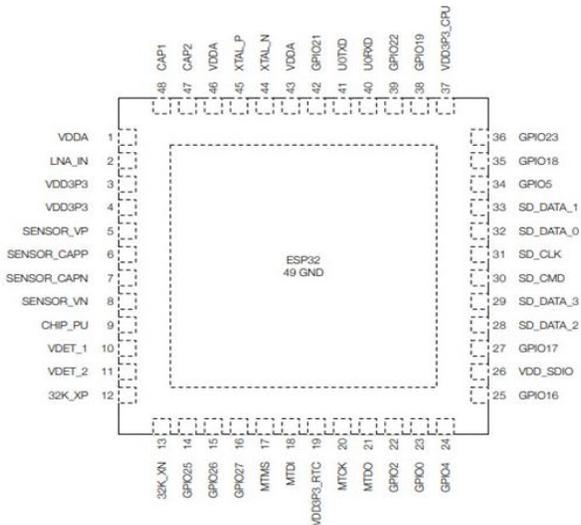
Un punto interesante sobre el USB-to-UART IC es que sus pines DTR y RTS se utilizan para configurar automáticamente el ESP32 en el modo de programación (siempre que sea necesario) y también descansar la placa después de la programación.

2.7.3. Pinout de la placa ESP32 de 30 pines



2.7.4. Especificaciones ESP32

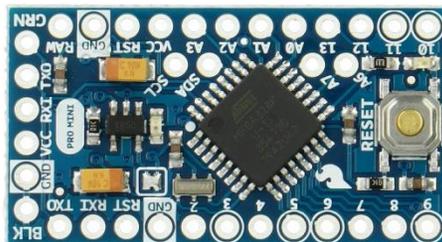
GPIO Pin	Pin on ESP32	Information
0	-	Pulled HIGH. Connected to BOOT Button
1	TX0	Do not use while TXing
2	YES	Pulled LOW
3	RX0	Do not use while RXing
4	D4	Pulled LOW
5	D5	Pulled HIGH
6	-	Connected to SPI Flash IC
7	-	Connected to SPI Flash IC
8	-	Connected to SPI Flash IC
9	-	Connected to SPI Flash IC
10	-	Connected to SPI Flash IC
11	-	Connected to SPI Flash IC
12	D12	Pulled LOW. Boot fails if pulled HIGH as it sets voltage of internal voltage regulator.
13	D13	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPICLK, HS2_CLK, SD_CLK, EMAC_TXD2
14	D14	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI, HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3
15	D15	Pulled HIGH
16	RX2	UART2 RX
17	TX2	UART2 TX
18	D18	GPIO10, SD_DATA3, SPIWP, HS1_DATA3, U1TXD
19	D19	GPIO11, SD_CMD, SPICS0, HS1_CMD, U1RTS
21	D21	I2C SDA
22	D22	I2C SCL
23	D23	GPIO15, ADC2_CH3, TOUCH3, MTDO, HSPICS0, RTC_GPIO13, HS2_CMD, SD_CMD, EMAC_RXD3
25	D25	GPIO0, ADC2_CH1, TOUCH1, RTC_GPIO11, CLK_OUT1, EMAC_TX_CLK
26	D26	GPIO4, ADC2_CH0, TOUCH0, RTC_GPIO10, HSPICLK, HS2_DATA1, SD_DATA1, EMAC_TX_ER
27	D27	GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT
32	D32	-
33	D33	GPIO21, VSPIHD, EMAC_TX_EN
34	D34	Digital Input Only. No Digital Output.
35	D35	Digital Input Only. No Digital Output.
36	YES	Digital Input Only. No Digital Output.
39	YES	Digital Input Only. No Digital Output.



- Microprocesador LX6 de 32 bits de uno o dos núcleos con frecuencia de reloj de hasta 240 MHz.
- 520 KB de SRAM, 448 KB de ROM y 16 KB de RTC SRAM.
- Admite conectividad Wi-Fi 802.11 b/g/n con velocidades de hasta 150 Mbps.
- Compatibilidad con las especificaciones Classic Bluetooth v4.2 y BLE.
- 34 GPIO programables.
- Hasta 18 canales de SAR ADC de 12 bits y 2 canales de 8 bits DAC
- La conectividad serial incluye 4 x SPI, 2 x I²C, 2 x I²S, 3 x UART.
- Ethernet MAC para comunicación LAN física (requiere PHY externo).
- 1 controlador host para SD/SDIO/MMC y 1 controlador esclavo para SDIO/SPI.
- Motor PWM y hasta 16 canales de LED PWM.
- Arranque seguro y cifrado flash.
- Aceleración de hardware criptográfico para AES, Hash (SHA-2), RSA, ECC y RNG

2.8. Arduino Mini Pro

Esta placa se desarrolló para aplicaciones e instalaciones donde el espacio es primordial y los proyectos se realizan como instalaciones permanentes. Pequeño, disponible en versiones de 3,3 V y 5 V, alimentado por ATmega328P.



El Arduino Pro Mini es una placa de microcontrolador basada en el ATmega328P.

Tiene 14 pines de entrada/salida digital (de los cuales 6 se pueden usar como salidas PWM), 6 entradas analógicas, un resonador incorporado, un botón de reinicio y orificios para montar cabezales de pines. Se puede conectar un encabezado de seis pines a un cable FTDI o a una placa de conexión Sparkfun para proporcionar alimentación USB y comunicación a la placa.

El Arduino Pro Mini está destinado a la instalación semipermanente en objetos o exposiciones. La placa viene sin cabezales premontados, lo que permite el uso de varios tipos de conectores o soldadura directa de cables. El diseño de pines es compatible con Arduino Mini.

Hay dos versiones del Pro Mini. Uno funciona a 3,3 V y 8 MHz, el otro a 5 V y 16 MHz.

El Arduino Pro Mini fue diseñado y fabricado por SparkFun Electronics.

microcontrolador	ATmega328P *
Fuente de alimentación de la placa	3,35 -12 V (modelo de 3,3 V) o 5 - 12 V (modelo de 5 V)
Voltaje de funcionamiento del circuito	3,3V o 5V (según modelo)
Pines de E/S digitales	14
Pines PWM	6
UART	1
SPI	1
I2C	1
Pines de entrada analógica	6
Interrupciones externas	2
Corriente CC por pin de E/S	40mA
Memoria flash	32 KB de los cuales 2 KB utilizados por el gestor de arranque *
SRAM	2 KB *
EEPROM	1 KB *
Velocidad de reloj	8 MHz (versiones de 3,3 V) o 16 MHz (versiones de 5 V)

Las placas más antiguas estaban equipadas con ATmega 168 con estas especificaciones:

Memoria flash: 16 KB

SRAM: 1 KB

EEPROM: 512 bytes

Cada uno de los 14 pines digitales del Pro Mini se puede usar como entrada o salida, utilizando las funciones pinMode, digitalWrite y digitalRead. Funcionan a 3,3 o 5 voltios (según el modelo). Cada pin puede proporcionar o recibir un máximo de 40 mA y tiene una resistencia pull-up interna (desconectada por defecto) de 20-50 kOhms. Además, algunos pines tienen funciones especializadas:

Serie: 0 (RX) y 1 (TX). Se utiliza para recibir (RX) y transmitir (TX) datos en serie TTL. Estos pines están conectados a los pines TX-0 y RX-1 del encabezado de seis pines.

Interrupciones externas: 2 y 3. Estos pines se pueden configurar para activar una interrupción en un valor bajo, un flanco ascendente o descendente, o un cambio de valor. Consulte la función adjuntar Interrupción para obtener más información.

PWM: 3, 5, 6, 9, 10 y 11. Proporcione una salida PWM de 8 bits con la función analogWrite.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estos pines admiten la comunicación SPI que, aunque la proporciona el hardware subyacente, actualmente no está incluida en el lenguaje Arduino.

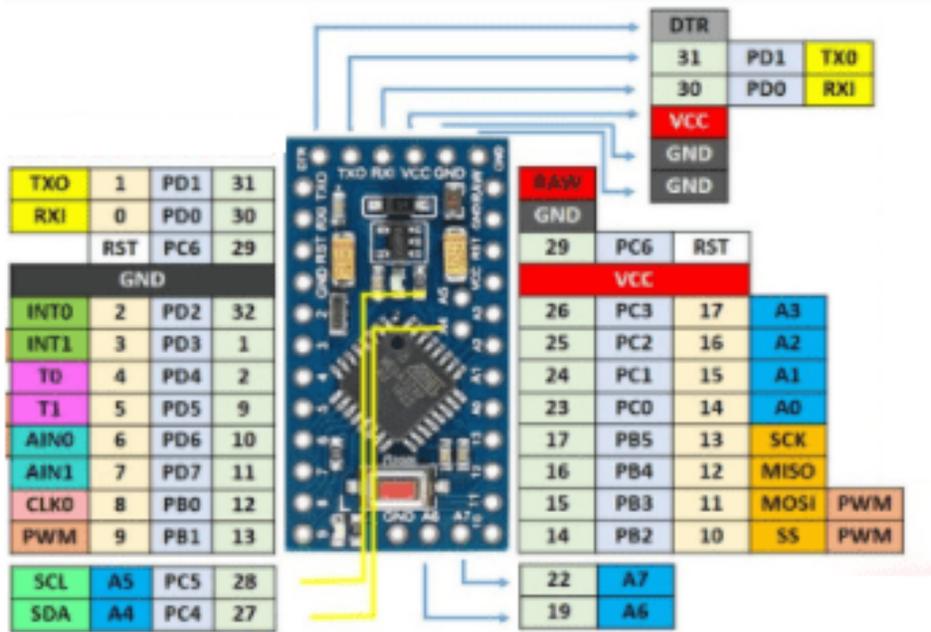
LED: 13. Hay un LED incorporado conectado al pin digital 13. Cuando el pin tiene un valor ALTO, el LED está encendido, cuando el pin tiene un valor BAJO, está apagado.

El Pro Mini tiene 8 entradas analógicas, cada una de las cuales proporciona 10 bits de resolución (es decir, 1024 valores diferentes). Cuatro de ellos están en los encabezados en el borde del tablero; dos (entradas 4 y 5) en agujeros en el interior del tablero. Las entradas analógicas miden de tierra a VCC. Además, algunos pines tienen una funcionalidad especializada:

I2C: A4 (SDA) y A5 (SCL). Admite comunicación I2C (TWI) utilizando la biblioteca Wire.

Hay otro pin en el tablero:

Reiniciar. Lleve esta línea a BAJO para restablecer el microcontrolador. Por lo general, se usa para agregar un botón de reinicio a los escudos que bloquean el que está en el tablero.



2.9. Shields para Arduino

Usualmente para proyectos de robótica, automatización y otros proyectos que requieran de una placa de microcontrolador como Arduino, no es posible conectar directamente ciertos componentes a estos controladores o son difíciles de adaptar, por ejemplo: motores DC que necesitan de

circuitos externos para recibir la corriente requerida por dicho motor, la cual no puede ser brindada por la placa Arduino.

En Arduino se implementa una amplia diversidad de shields para facilitar el manejo de algunos componentes como: control de motores, pantallas LCD, conexión WIFI, conexión a internet, etc.

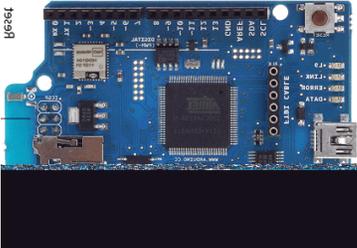
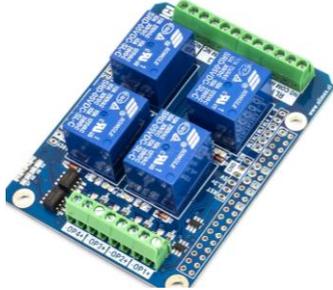
2.9.1. ¿Qué es un shield?

Existen placas electrónicas o placas de circuito impreso llamadas shields, estas placas cumplen la función de ampliar la capacidad de la PCB Arduino y facilitar el control de algunos componentes a utilizar en nuestros proyectos. Otra función de las placas shields es el acoplamiento compacto que existe entre dicha placa y la placa Arduino, ya que la conexión se hace mediante el acople directo de los pines de ambas placas, además, de esta misma manera es posible apilar más de un shield colocándolos uno sobre otro.

Generalmente para cada placa shield suele venir asociada una librería, facilitando el trabajo a la hora de crear los códigos para el control de nuestros dispositivos mediante las placas shield y la placa principal Arduino

En la siguiente tabla se presentan algunos ejemplos de shield para Arduino:

Nombre	Descripción	Representación grafica
Ethernet Shield v4.0	La principal función del shield ethernet V4.0, es que permite que una placa arduino se conecte a interne basándose en el chip ethernet Wiznet W5100.	 A photograph of an Arduino Ethernet Shield v4.0. The board is green and populated with various electronic components. On the left side, there is a blue Ethernet port. The central part of the board features a large black integrated circuit (the Wiznet W5100 chip) and several smaller components like capacitors and resistors. On the right side, there are two RJ45 ports. The board is designed to be connected to an Arduino Uno or similar board via its pin headers.

microSD Shield	<p>Este shield permite tener al Arduino un almacenamiento masivo, este puede utilizarse para el registro de datos o proyectos similares.</p>	
Graphic LCD4884 Shield	<p>El <u>Graphic LCD4884 Shield</u> proporciona una pantalla de 48x48 píxeles. Tiene la capacidad de mostrar lenguaje inglés, chino e imágenes.</p>	
Arduino WiFi Shield	<p>Este shield permite a la placa Arduino el acceso a internet mediante las librerías Wifi, además puede leer y escribir en una tarjeta SD mediante la librería SD.</p>	
Relay Shield	<p>El 4-relay shield es una placa que tiene integrado 4 relés mediante los cuales la placa puede encender y apagar ciertos dispositivos como TV, equipos de sonidos, luces, etc.</p>	
Motor shield L293D	<p>Este shield nos facilita el control de motores como motores DC, servo motores y motores paso a paso.</p>	

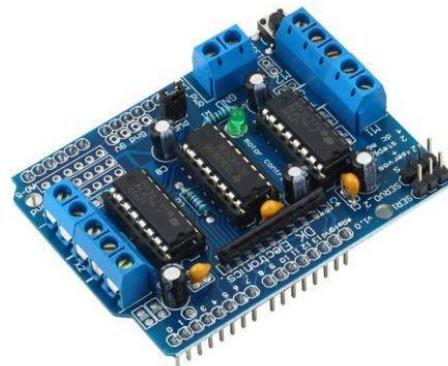
2.9.2. Motor Shield L293D

El Motor Shield L293D es una placa complementaria de Arduino que nos ayuda a controlar fácilmente distintos tipos de motores como: motores DC, motores paso a paso y servomotores.

En este shield se pueden conectar dos servos motores y cuatro motores DC o dos motores paso a paso, cada canal de conexión para los motores

tiene una corriente máxima de 1.2A y su rango de funcionamiento con respecto al voltaje está entre los 4.5V y 25V.

Esta placa está basada en la implementación de 3 circuitos integrados L293D. Un circuito integrado L293D se comporta como un puente H doble de alta corriente. Los denominados puentes H consisten en un circuito a base de transistores y generalmente son utilizados para darle un cambio de giro a motores DC.



2.9.2.1. Control de servo motor, utilizando Motor shield L293D

Comando	Descripción
<code>#include <Servo.h></code>	Incluye la librería "Servo", esto facilita el control de un servo motor por medio de arduino.
<code>Servo myservo;</code>	Este comando indica que se está creando un objeto de tipo Servo. (myservo)
<code>myservo.attach ();</code>	Vincula la variable o el objeto Servo a una patilla de salida de Arduino.
<code>myservo.write();</code>	Por medio de este comando indicamos cuántos grados gira el servo motor (0°-180°).

2.9.2.2. Control de motor paso a paso, utilizando Motor shield L293D

1. Instalar librerías necesarias:

- **Librería para shield L293D.**

link de descarga: [GitHub -](#)

[adafruit/Adafruit Motor Shield V2 Library: v2!](#)

Comando	Descripción
#include <AFmotor.h>	Incluye la librería "AFmotor.h", esta librería facilita el manejo de motores DC y motores paso a paso, con el uso del shield L293D.
AF_stepper motor (256,1);	En este comando se define un motor paso a paso, además se configura el número de pasos por revolución (256), y los canales a usar del shield. Estos pueden ser 1 (canales 1 y 2) o 2 (canales 3 y 4).
motor.setSpeed ();	Con este comando configuramos la velocidad de los parámetros del motor.
motor.step (2000, FORWARD, SINGLE);	Con el comando ".step ", podemos conjurar el número de pasos que dará el motor, el sentido del giro de motor ("FORWARD" ADELANTE, "BACKWARD" ATRÁS), y el método de excitación que pueden ser: SINGLE , DOUBLE , INTERLEAVE, MICROSTEP.
delay();	En este comando indicamos un retardo. El valor que se proporcione en este comando será comprendido en valores de milisegundos, es decir, si en este comando colocamos un valor de 100, esto indica que habrá un retardo de 100 milisegundos.

2.9.2.3. Control de motor DC, utilizando Motor shield L293D

1. Instalar librerías necesarias:

- **Librería para shield L293D.**

link de descarga: [GitHub -](#)

[adafruit/Adafruit Motor Shield V2 Library: v2!](#)

Comando	Descripción
<pre>#include <AFmotor.h></pre>	Incluye la librería "AFmotor.h", esta librería facilita el manejo de motores DC y motores paso a paso, con el uso del shield L293D.
<pre>AF_DCMotor motor1 (1,MOTOR12_64KHz);</pre>	En este comando se crea un objeto o variable para el motor a controlar, además se establece el canal a utilizar del shield (1, 2, 3 y 4). También se establece la frecuencia PWM, en el caso que no se especifique, se usará 1KHz de forma predeterminada. Las frecuencias para los canales 1 y 2 son: 64KHz, 8KHz, 62KHz, 1KHz. La frecuencia en los canales 3 y 4 pueden ser : 64KHz, 8KHz, 1KHz.
<pre>Motor1.setSpeed (200);</pre>	Con este comando configuramos la velocidad de los parámetros del motor.

motor.run (FORDWARD);	Con el comando ".run", configuramos el modo de funcionamiento del motor. Por ejemplo que el giro sea hacia adelante (FORDWARD), que el giro del motor sea para atrás (BACKWARD) o que se detenga cortándole la energía al motor (RELEASE).
motor.run (BACKWARD);	
motor.run (RELEASE);	

2.10. MPU 6050

Gracias a la tecnología actualmente, se puede encontrar una gran diversidad de sensores, unos más complejos que otros, ya sea por su principio de funcionamiento, por señal de salida, por la magnitud a medir, etc. Además de los tipos de sensores mencionados, existen sensores que nos permiten medir más de una magnitud física, como es el caso del sensor de movimiento MPU 6050.

Primeramente, para conocer el sensor MPU 6050 y cómo funciona, necesitamos saber el significado de los siguientes términos:

- **IMU: inertial measurement unit**, son unidades de medida de inercia de un cuerpo, este es un dispositivo electrónico que mide e informa la velocidad, orientación y fuerza gravitacional de un dispositivo.
- **DOF:** estas siglas nos indican el grado de libertad de un dispositivo (**Degree of freedom**), se puede interpretar como el número de parámetros que definen el estado de un objeto, por ejemplo: un tren de uniones rígidas entre la locomotora y vagones está delimitada a un solo grado de libertad, ya que la distancia de este varía solamente a lo largo de la vía.

- **DMP: Digital Motion Processor** o procesador de movimiento digital en español, es un dispositivo que permite procesar complejos algoritmos de capturas de movimientos, es decir que acondiciona las capturas de movimiento en señales digitales, para que estas posteriormente sean interpretadas en un microcontrolador.
- **Aceleración:** es denominada aceleración, a la variación de la velocidad por unidad de tiempo y se expresa de la siguiente manera:

$$a = \frac{dv}{dt}$$

En la segunda ley de Newton se presenta que, la aceleración de un cuerpo con masa constante es proporcional a la fuerza que actúa sobre dicho cuerpo. Para esta definición, la aceleración se expresa de la siguiente manera:

$$a = \frac{F}{m}$$

- **Acelerómetro:** la definición de la aceleración en la segunda ley de newton es la utilizada por el acelerómetro MEMS (sistema microelectromecánico), este MEMS consiste es un sistema similar al de masa-resorte, el cual nos permite medir la aceleración.

A pesar de que no recibe movimiento el acelerómetro, este estará censando la aceleración de la gravedad. Además, de esta aceleración podríamos obtener algunas mediciones indirectas como lo es la velocidad y el desplazamiento, si integramos la aceleración por el tiempo, obtendremos la velocidad, si realizamos nuevamente una integración, pues entonces obtendremos el desplazamiento. Para realizar estas mediciones indirectas, debemos conocer tanto la velocidad como la posición inicial del cuerpo.

Velocidad angular: al cambio del desplazamiento angular por unidad de tiempo, se le conoce como velocidad angular, es decir la rapidez que tiene un cuerpo para girar alrededor de un eje. La velocidad angular se expresa de la siguiente forma:

$$\omega = \frac{d\theta}{dt}$$

- **Giroscopio:** existen diferentes tipos de giroscopios por ejemplo, los giroscopios que implementan un MEMS, estos giroscopios miden la velocidad angular a partir del efecto llamado Coriolis, consiste en que un cuerpo vibratorio tiende a seguir vibrando en el plano que gira su apoyo, el efecto Coriolis provoca que el objeto vibratorio induzca una fuerza (F_c), y a partir de esta se puede determinar la velocidad angular que existe en un objeto.

Al igual que en el acelerómetro, con un giroscopio podemos obtener mediciones indirectas, si integramos la velocidad angular con respecto al tiempo, podríamos obtener el desplazamiento angular (posición angular si se sabe la posición inicial del giro).

2.10.1. I2C

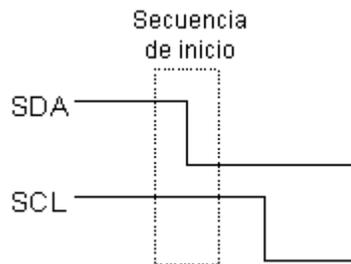
El protocolo **Inter Integrated Circuit** o mayormente conocido como I2C, es un protocolo de comunicación serial para intercambiar datos entre dos o más dispositivos. El intercambio de información entre dispositivos es en serie y síncrona, se hace mediante dos cables llamados: SDA (Serial Data) y SCL (Serial Clock). Estos cables trabajan con una configuración de tipo pull-up.

- o SDA: utilizado para el intercambio de datos.
- o SCL: Utilizado como señal de reloj.

Este protocolo se basa en la comunicación entre un dispositivo maestro (generalmente microcontroladores) y múltiples dispositivos esclavos (generalmente sensores), tanto el dispositivo maestro como el esclavo,

pueden enviar y recibir datos, pero solo el maestro puede enviar señales de SCL.

La comunicación entre maestro y esclavo se inicia cuando el maestro deja la señal de datos (SDA) en bajo y la señal de reloj (SCL) en alto, es decir solo el maestro puede iniciar la comunicación, ya que solo el maestro puede enviar señales de reloj.



Al iniciar la comunicación, el maestro envía una secuencia de 9 bits, de los cuales los primeros 7 bits (A0-A6) componen la dirección del dispositivo que se desea seleccionar (esclavo), el bit 8 presenta la operación que se quiere realizar con el esclavo, estas operaciones pueden ser de lectura (R) o escritura (W), si el bit 8 está en un estado lógico bajo será una operación de escritura, en caso contrario (bit 8 en estado lógico alto) la operación será de tipo lectura.

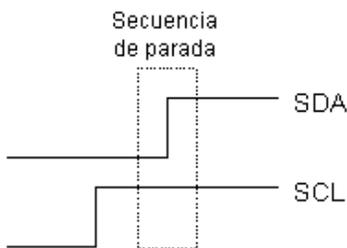
Si la operación se encuentra en modo escritura, el dispositivo maestro envía datos al dispositivo esclavo. En el caso contrario, si se encuentra en el modo lectura, el dispositivo maestro genera pulsos de reloj para que el dispositivo esclavo pueda enviarle los datos.

El bit 9, es utilizado para que el esclavo responda al dispositivo maestro, si el bit está en bajo, se está confirmando al dispositivo maestro que la dirección del dispositivo esclavo es correcta, que reconoce la solicitud y que está en condiciones de comunicarse. A este bit de reconocimiento se denomina bit ACK.

Luego de esto el dispositivo maestro envía un byte en la señal de reloj, la cual corresponde a los datos a transferir (D0-D7), ya sea que esté en modo de lectura o modo de escritura. Seguidamente tenemos un noveno bit que corresponde al reconocimiento del dato transferido (ACK).



Finalmente, el maestro puede decidir dejar libre el bus (cortar comunicación) enviando una condición de parada, esta se da cuando deja las señales de reloj y datos en alto.



Si el maestro decide seguir transfiriendo datos, puede generar otra condición de inicio en lugar de una condición de parada. A este nuevo inicio se conoce como "bit inicio repetido".

Ejemplo de una comunicación I2C.



Ahora que ya tenemos claro estos términos podemos describir el sensor MPU 6050 y su funcionamiento.

2.10.2. ¿Qué es un MPU 6050?

El MPU 6050 es un sensor IMU básico y es compatible con microcontroladores como Arduino. Este sensor tiene integrado un giroscopio MEMS (sistemas microelectromecánicos) de 3 ejes, un acelerómetro MEMS de 3 ejes y un DMP.

Este dispositivo es una pequeña pieza que nos ayuda al procesamiento de movimientos. Las piezas integran el algoritmo MotionFusion para 9 ejes pueden incluso acceder a magnetómetros externos u otros sensores a través de un bus I2C auxiliar maestro. Los pines SCL y SDA tienen una resistencia pull-up en placa para una conexión directa al microcontrolador o Arduino.

El MPU 6050 es un sensor DOF de 6 grados de libertad, es decir un sensor (IMU) de 6 ejes. Este sensor es muy preciso ya que consta de un conversor ADC (conversor análogo digital) de 16 bits que convierte los datos a un valor digital, el sensor se comunica con el Arduino por medio del protocolo I2C.

Este sensor consta de dos direcciones de comunicación que son, 0x58 y 0x59.

2.10.3. Comandos MPU 6050 con Arduino

1. Librería para sensor MPU 6050 (desarrollada por Jeff Rowberg).

link de descarga: [GitHub - ZHomeSlice/Simple MPU6050: Going Live](#)

2. Librería para comunicación I2C

link de descarga: [GitHub - jrowberg/i2cdevlib: I2C device library collection for AVR/Arduino or other C++-based MCUs](#)

Comando	Descripción
#include "MPU6050.h"	Se incluye la librería "MPU6050.h", para facilitar el control del sensor MPU6050.
#include "I2Cdev.h"	Este comando incluye la librería I2C, esta librería nos ayuda a tener una comunicación entre un dispositivo y la placa Arduino por medio de dos pines, por medio del protocolo de comunicación I2C. Para poder usar esta librería también necesitamos el uso de una librería llamada "Wire".
#include "Wire.h"	En este comando se incluye la librería, la cual es una librería estándar de Arduino, esta librería permite a la Arduino poder comunicarse con dispositivos por medio de la comunicación I2C.
MPU6050 sensor;	En este comando se crea un objeto para el sensor MPU6050 (sensor). Además se especifica la dirección de conexión, si no se especifica la dirección, se toma la dirección 0x58 como dirección predeterminada.
Serial.begin (9600);	Este comando inicia la comunicación del puerto serial.
Wire.begin ();	Inicia la librería "Wire" (inicia comunicación I2C)

<pre>sensor.testConecction ()</pre>	<p>Este comando nos ayuda a saber el estado del sensor (conectado o si existe algún error)</p>
<pre>ax_o= sensor.getXAccelOffset() ; ay_o= sensor.getYAccelOffset() ; az_o=sensor.getZAccelOff set();</pre>	<p>Con este comando podemos conocer o leer el offset del acelerómetro que tiene el sensor MPU 6050 en cualquiera de sus ejes (x, y, z).</p>
<pre>ax_o= sensor.getXGyroOffset(); ay_o= sensor.getYGyroOffset(); az_o= sensor.getZGyroOffset();</pre>	<p>En este comando se puede conocer o leer el offset del giroscopio que tiene el sensor MPU 6050 en cualquiera de sus ejes (x, y, z).</p>
<pre>sensor.getAcceleration(& ax, &ay, &az)</pre>	<p>Podemos leer la aceleración del sensor con este comando, las aceleraciones de cada uno de los ejes se guardan en las variables (ax, ay, az).</p>
<pre>sensor.getRotation (&gx, &gy, &gz)</pre>	<p>Se puede leer la rotación del sensor con este comando, la rotación en cada uno de los ejes se guarda en las variables (gx, gy, gz).</p>

<pre>Serial.print ();</pre>	<p>Este comando permite escribir en el monitor serial, ya sea un valor de una variable, o texto. Si se desea expresar el valor de una variable en el monitor serial, simplemente se escribe la variable entre los paréntesis. Por el contrario, si se desea presentar un texto o una palabra como tal, este se debe escribir entre comillas dentro del paréntesis, por ejemplo:</p> <pre>Serial.print ("hola mundo");</pre>
<pre>sensor.setXAccelOffset (); sensor.setYAccelOffset (); sensor.setZAccelOffset ();</pre>	<p>Por medio de este comando podemos ajustar el offset de aceleración del sensor MPU6050 en cada uno de sus ejes (x, y, z).</p>
<pre>sensor.setXGyroOffset (); sensor.setYGyroOffset (); sensor.setZGyroOffset ();</pre>	<p>Por medio de este comando podemos ajustar el offset de giroscopio del sensor MPU6050 en cada uno de sus ejes (x, y, z).</p>

2.11. Sensores

En la construcción de máquinas automatizadas y semiautomatizadas, es casi indispensable la implementación de dispositivos que permita a estas máquinas interactuar con su medio y a partir de esto hacer una

toma de decisiones, ya sea decisiones para el funcionamiento principal de la máquina o decisiones que ayuden a resguardar nuestro equipo de algún peligro.

Dependiendo para qué función esté construida una máquina, puede haber distintos tipos de cambios físicos en su entorno, como, por ejemplo: cambio de temperatura, presencia y cambios de humedad, presencia de humo, cambios de niveles de líquidos, cambio de posición de objetos, etc.

Gracias a la avanzada tecnología de hoy en día, se pueden encontrar una amplia gama de sensores de distintos tipos, modelos y tamaños que ayudan a solventar las necesidades de los diferentes tipos de máquinas automatizadas para medir cambios físicos en su entorno.

2.11.1. ¿Qué es un sensor?

Un sensor es un dispositivo que, a partir de un estímulo ya sea químicas o físicas, presenta una respuesta que normalmente es en forma de señal eléctrica, es decir, los sensores nos ayudan a detectar y medir cambios físicos que se puedan producir, para posteriormente procesar los datos obtenidos.

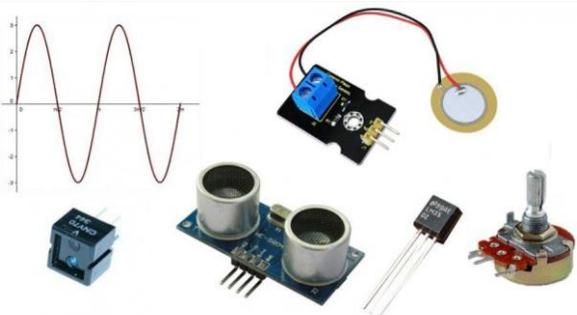
Todo tipo de sensor necesita de una calibración para ser utilizado, en dicha calibración se debe establecer una relación entre la variable a medir y la señal de salida convertida.

Algunos términos importantes que se utilizan en torno al tema de sensores son: exactitud, precisión, rango de funcionamiento, velocidad de respuesta y fiabilidad.

- **Exactitud:** es denominada como la cualidad de ajustarse o acercarse al valor verdadero.

- **Precisión:** este término hace referencia a la cercanía de varias medidas entre sí.
- **Rango de funcionamiento:** es el rango de medición que pueda realizar el sensor, es decir, desde que mínima cantidad puede medir y hasta que máxima cantidad es posible de medir con el instrumento.
- **Velocidad de respuesta:** es el tiempo que se tarda el sensor en dar una respuesta, a partir del momento de la alteración física percibida.
- **Fiabilidad:** el sensor debe ser altamente fiable, es decir que no debe estar sujeto a fallos frecuentes presentados en su funcionamiento.

Existen una gran diversidad de sensores, y estos se pueden clasificar por: señales de salida, energía, principio de funcionamiento y por magnitud a medir.

Clasificación de sensores por señal de salida		
Nombre	Descripción	Descripción gráfica
Análogos	Estos sensores proporcionan una señal eléctrica analógica, es decir un rango de valor de voltaje o corriente.	
Digitales	Estos sensores proporcionan una señal de salida en forma de bist de estado (0-1).	

Clasificación de sensores por energía

Nombre	Descripción	Descripción gráfica
Pasivos	Emiten la señal a partir de la transformación de energía medida.	
Activos	Estos sensores necesitan energía de alimentación para medir y emitir una señal de salida.	

Clasificación de sensores por principio de funcionamiento

Nombre	Descripción	Descripción gráfica
Sensores resistivos	En estos sensores, existe una resistencia en la cual cambia su valor resistivo con respecto a la variable a medir.	
Sensores inductivos	Estos sensores están diseñados para generar un campo electromagnético, cuando un objeto metálico entra en ese campo se inducen corrientes de la superficie en el objeto	

	metálico a medir. Es decir mide la presencia de objetos metálicos.	
Sensores capacitivos	Este sensor se basa en el principio de un capacitor, en el cual la capacidad de almacenamiento eléctrico varía dependiendo de la constante dieléctrica que se encuentre entre dos placas energizadas. Con este sensor podemos saber si existe la presencia de un objeto de cualquier material.	
Sensores generadores	Los sensores generadores son sensores que generan señales eléctricas a partir de las magnitudes que miden.	

Clasificación de sensores según la magnitud a medir		
Tipo de magnitud	Sensor	Señal de salida
Posición lineal o angular	Finales de carrera	todo - nada
	Potenciómetros	Analógica
	Encoder (absolutos incrementales)	Digital
Pequeños desplazamientos o deformaciones	Transformadores diferenciales	Analógica
	Galgas extensiométricas	Analógica
Velocidad lineal o angular.	Dinamos tacométricas	Analógica
	Encoder	Digital
	Detectores inductivos	Digital
Aceleración	Acelerómetros	Analógica
	Sensores de velocidad +calculador	Digital

Fuerza y par	Medición indirecta mediante galgas o transformadores diferenciales.	Analógicas
Nivel	Flotador +detector de desplazamiento	Analógicas
	Capacitivos	Analógicas
	Ultrasónicos	Digital
Presión	Membrana + detector de desplazamiento	Analógica
	Piezo eléctrico	Analógica
Caudal	Presión diferencial (diagramas /tubos de Venturi)	Analógica
	De turbina	Analógica
	Magnético	Analógica
Temperatura	Termostato	Todo - nada
	Termopares	Analógica
	Termoresistencias (PT100)	Analógica
	Resistencias NTC	Analógica
	Resistencias PTC	Analógica
	Pirómetros	Analógica
Sensores de presencia o proximidad.	Inductivos	Todo - nada
	Capacitivos	Todo - nada
	Ópticos (células fotoeléctricas)	Todo - nada
	Ultrasonido	Analógico
Sistema de visión artificial.	Cámara de video y tratamiento de imagen	Procesamiento por puntos o píxeles.
	Cámara CCD	

2.11.2. Sensor ultrasónico

Estos sensores de ultrasonido son sensores de proximidad, los cuales miden la distancia de objetos mediante ondas ultrasónicas. Estas ondas ultrasónicas son emitidas por el sensor mediante su elemento emisor, al chocar la onda con un objeto, ésta regresa y es captada por el elemento receptor.



El tiempo que se tarda la onda en hacer esta trayectoria es convertido en una señal eléctrica, para que luego esta señal sea interpretada por algún controlador en el cual se realizan ciertas operaciones para determinar la distancia.

Existen diversos tipos de sensores ultrasónicos, entre ellos tenemos el módulo de sensor ultrasónico HC-SR04. Este sensor es ideal para usar con los microcontroladores Arduino y PIC.

La comunicación entre el controlador (Arduino) y el módulo HC-SR04 se realiza mediante los pines llamados Triger y Echo.

El pin Trig o disparo, es donde el módulo recibe la señal del microcontrolador para realiza un disparo que emite 8 pulsos de ultrasonido (40KHz) mediante el piezoeléctrico emisor, luego de esto las ondas viajan en el aire y rebotan en el objeto, el sonido de rebote es captado por el piezoeléctrico receptor, el pin echo manda una señal de alto (5v) por un tiempo que es iguala al tiempo en que demoró la onda desde ser emitida hasta que regresó al sensor.

Dado que el sensor nos envía un pulso al microcontrolador, este pulso se mantiene un tiempo equivalente a la trayectoria del sonido emitido, y este tiempo está dado en microsegundos, necesitamos hacer un pequeño cálculo para encontrar la distancia.

Teniendo la velocidad a la que viaja el sonido y el tiempo que le tomó en toda la trayectoria, podríamos encontrar la variable deseada (distancia) con la fórmula de velocidad.

$$V_s = \frac{d}{t}$$

Despejando para d tendríamos:

$$d = V_s t$$

La velocidad del sonido es 344 m/s, pero el valor de tiempo que recibimos está dado en μs , además para conveniencia, determinaremos la distancia en cm. Para esto realizamos las siguientes conversiones:

$$V_s = 344 \frac{m}{s} * \frac{100cm}{m} * \frac{1s}{1x10^6 \mu s} = 0.0344cm/\mu s$$

Teniendo este valor tendrías la fórmula:

$$d = (0.0344cm/\mu s)(t)$$

Con la fórmula anterior obtendremos la distancia total de la trayectoria, es decir ida y vuelta, pero solo necesitamos la distancia desde el sensor hasta el objeto, para esto simplemente dividimos entre dos, la ecuación nos queda:

$$d = \frac{\left(0.0344 \frac{cm}{\mu s}\right)(t)}{2}$$

2.11.2.1. Comandos usados para medición de distancia con sensor ultrasónico HC-RS04.

Comando	Descripción
Serial.begin (9600);	Este comando inicia la comunicación del puerto serial.

<pre>PinMode (12, OUTPUT); PinMode (trig, OUTPUT);</pre>	<p>En este comando especificamos el modo en que estará configurado un pin, ya sea OUTPUT (salida) o INPUT (entrada). Si para el pin ya se definió una variable, esta se puede poner a cambio del número del pin.</p>
<pre>digitalWrite (trig, LOW);</pre>	<p>Con este comando activamos o desactivamos (5V-0V) la salida de un pin, primero se define el pin (si el pin deseado a configurar tiene una variable asignada, se puede colocar dicha variable) y luego el estado del pin, ya sea "HIGH" para activarlo, o "LOW" para desactivarlo.</p>
<pre>pulseIn (pin, value); pulseIn (pin, value, timeout);</pre>	<p>Este comando lee el tiempo de un pulso en un pin específico y lo devuelve en microsegundos. Se puede usar simplemente especificando el pin a usar y el tipo de pulso a medir, ya sea HIGH (alto) O LOW (bajo). Además de las anteriores, si se desea, se puede configurar un tiempo (timeout) que espera a que el pulso se complete, si el pulso no se completa en este tiempo establecido, el comando brindará un valor igual a 0.</p>
<pre>delayMicroseconds ();</pre>	<p>En este comando indicamos un retardo. El valor que se proporcione en este comando será comprendido en valores de microsegundos, es decir, si en este comando colocamos un valor de 100, esto indica que habrá un retardo de 100 microsegundos.</p>

2.12. Displays para Arduino

Los displays LCD son elementos muy útiles en la mayoría de los proyectos para poder visualizar información sin tener conectado a nuestro Arduino un ordenador, es por ello que vamos a hacer un pequeño repaso a los diferentes tipos de displays LCD que podemos encontrarnos y que son 100% compatibles con Arduino. El conocer diferentes displays y sus

características nos será muy útil a hora de decidir qué display usaremos nuestros proyectos.

la
en



2.12.1. Tipos de Display

En el mercado podemos encontrar muchos tipos de displays diferentes, estas diferencias están marcadas fundamentalmente por el tipo de display, que va a determinar cómo van a ser mostrados los datos y por la conexión que se usa para mostrar los datos en el display.

- Display LCD de líneas
- OLED
- TFT LCD
- Pantalla de papel electrónico
- Pantalla 5110
- Pantalla de 7 segmentos
- OLED Transparente
- Display LCD en matriz de puntos

2.12.1.1. Pantalla OLED SH1106

Las pantallas OLED se destacan por su gran contraste, mínimo consumo de energía y buena calidad de imagen. El Display Oled 1.3" SPI 128*64 SH1106 posee una resolución de 128*64 píxeles, permitiendo controlar cada píxel individualmente y mostrar tanto texto como gráficos. Además, por ser de tipo OLED no necesita de retroiluminación (Backlight) como

los LCD, lo que hace que su consumo de energía sea mucho menor y aumenta su contraste.

La interface de comunicación es de tipo SPI y puede trabajar con sistemas desde 3V hasta 5V gracias a su regulador y convertidores en la placa. Debemos tener en cuenta que los pines SPI son diferentes para cada modelo de Arduino, por lo que debemos revisar el Pinout de nuestro Arduino para saber cuáles son los pines SPI.

Para manejar la pantalla es necesario utilizar un microcontrolador con al menos 1K de RAM, este espacio cumple la función de buffer para el display. El controlador de la pantalla es el SH1106 (similar al SD1306), con una librería lista para usarse en Arduino. La librería permite mostrar texto, mapas de bits, píxeles, rectángulos, círculos y líneas. A pesar de usar 1K de RAM, el funcionamiento es muy rápido y el código es fácilmente portable a distintas plataformas de microcontroladores.

2.13. Baterías

Una batería eléctrica, también llamada pila o acumulador eléctrico, es un artefacto compuesto por celdas electroquímicas capaces de convertir la energía química en su interior en energía eléctrica. Así, las baterías generan corriente continua y, de esta manera, sirven para alimentar distintos circuitos eléctricos, dependiendo de su tamaño y potencia.

Las baterías poseen una capacidad de carga determinada por la naturaleza de su composición y que se mide en amperios-hora (Ah), lo que significa que la pila puede dar un amperio de corriente a lo largo de una hora continua de tiempo. Mientras mayor sea su capacidad de carga, más corriente podrá almacenar en su interior.

2.13.1. Tipos de batería

TIPO	ENERGIA/PESO	TENSION POR ELEMENTO (V)	DURACIÓN (NÚMERO DE RECARGAS)	TIEMPO DE CARGA	AUTODESCARGA POR MES (% DEL TOTAL)
Plomo	30-40 Wh/kg	2V	1000	8-16h	5%
Ni-Fe	35-55 Wh/kg	1,2V	+ de 10,000	4-8h	10%
Ni-Cd	48-80 Wh/kg	1,25	500	10-14h*	30%
Ni-Mh	60-120 Wh/kg	1,25V	1000	2h-4h*	20%
Li-ion	110-160 Wh/kg	3,16V	4000	2h-4h	25%
LiPo	100-130 Wh/kg	3,7V	5000	1h-1,5h	10%

2.13.1.1. Batería de polímero de litio (LiPo)

La batería de polímero de litio es un componente electrónico que ha ido ganando popularidad para el desarrollo proyectos de robótica y radiocontrol. Como ofrece más ventajas de las pilas de NiMH y NiCd, son una fuente de alimentación más confiable.

Ya que existe gran variedad de marcas y modelos, es importante saber cómo interpretar la información de una batería de polímero de litio. La mayor va a indicar el voltaje (3.7v), el amperaje (mAh), la Clasificación C y las siglas 3S1P o 3S2P (celdas + packs conectados en serie). Las LiPo pueden incluir varias celdas, aumentando la potencia de la pila. Existen baterías LiPo con los siguientes valores de voltaje:

- Una celda y 3.7v
- Dos celdas de 3.7, equivalentes a 7.4v

- Tres celdas de 3.7, iguales a 11.1v
- Cuatro celdas de 3.7, llegando a 14.8v

Corriente máxima o capacidad de una LiPo

Este valor se refiere al tiempo durante el cual la pila puede dar energía y está expresado como miliamperios hora (mAh). Mientras más grande sea este número, mayor será la capacidad de la batería. Esto significa que usted podrá alimentar más mecanismos por un lapso mayor. El único inconveniente de tener más capacidad es que el tamaño también aumentará.

Cálculo de la capacidad de una LiPo

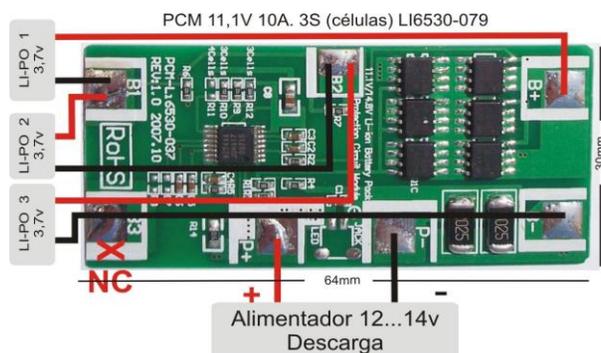
La fórmula para saber la capacidad de una batería de polímero de litio es sencilla. Basta tener los datos de la clasificación C y los miliamperios para multiplicarlos. Un ejemplo: 20C x 3500 mAh = 70,000 mA. Luego divide entre mil para simplificar, dando 70 mA.

Mientras que esta es la capacidad máxima, no se recomienda llevar la pila a este límite. Hacerlo podría disminuir su vida útil, provocar que se caliente o incluso que explote.

Clasificación C o constante de descarga

En la etiqueta de una LiPo aparece un valor seguido de una C. Esto se refiere a la tasa de descarga o corriente máxima que puede ofrecer. Algunas pilas también incluyen picos de descarga, que duran entre 10-30 seg.

Diagrama esquemático de una batería de polímero de litio de 12v



2.14. Servomotor

Un servomotor es un actuador rotativo o motor que permite un control preciso en términos de posición angular, aceleración y velocidad, capacidades que un motor normal no tiene. En definitiva, utiliza un motor normal y lo combina con un sensor para la retroalimentación de posición.

Pero, los servomotores no son en realidad una clase específica de motor, sino una combinación de piezas específicas, que incluyen un motor de corriente continua o alterna, y son adecuados para su uso en un sistema de control de bucle cerrado.

Por lo que una definición más exacta de un servomotor sería la de un servomecanismo de bucle cerrado que utiliza la retroalimentación de posición para controlar su velocidad de rotación y posición. La señal de control es la entrada, ya sea analógica o digital, que representa el comando de posición final para el eje.

Por otro lado, el codificador o encoder sirve como sensor, proporcionando retroalimentación de velocidad y posición. En la mayoría de los casos, sólo se informa de la posición. La posición final se informa al controlador y se compara con la entrada de posición inicial, y luego, si hay una discrepancia, se mueve el motor para llegar a la posición correcta.

Hay varios tipos de servos como los que utilizan motores de corriente continua y detección de posiciones a través de un potenciómetro y también utilizan un control de gran potencia, lo que significa que el motor se mueve a la velocidad máxima hasta que se detiene en la posición designada.

En cambio, los servos para uso industrial disponen de sensores de posición y velocidad, así como de algoritmos de control proporcional-integral-derivativo, lo que permite llevar el motor a su posición de forma rápida.

Este tipo de dispositivos están disponibles en una gran variedad de tipos, formas y tamaños. El término servo fue utilizado por primera vez en 1859 por Joseph Facort, que implementó un mecanismo de retroalimentación para ayudar a controlar los timones de un barco con vapor.

2.14.1. ¿Cómo funciona un servomotor?

Los servomotores se controlan enviando un pulso eléctrico de ancho variable, o modulación de ancho de pulso (PWM), a través del cable de control. Hay un pulso mínimo, un pulso máximo y una frecuencia de repetición.

Por lo general, un servomotor sólo puede girar 90° en cualquier dirección para un movimiento total de 180° . La posición neutra del motor se define como la posición en la que el servo tiene la misma cantidad de rotación potencial tanto en el sentido de las agujas del reloj como en el sentido contrario.

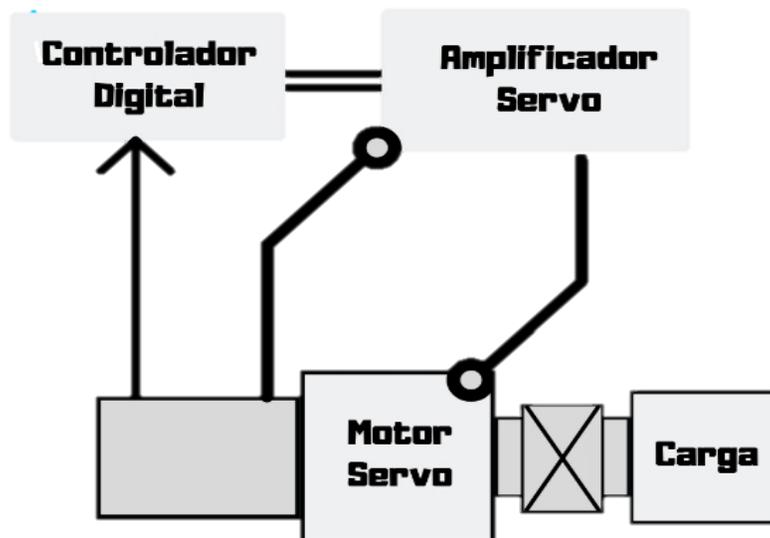
El PWM enviado al motor determina la posición del eje, y se basa en la duración del pulso enviado a través del cable de control; el rotor girará a la posición deseada.

El servomotor espera ver un pulso cada 20 milisegundos (ms) y la longitud del pulso determinará hasta dónde gira el motor. Por ejemplo, un pulso de 1.5ms hará que el motor gire a la posición de 90° .

Si el tiempo es inferior a 1,5 ms, se mueve en sentido contrario a las agujas del reloj hacia la posición de 0°, y si el tiempo es superior a 1,5 ms, el servo girará en sentido de las agujas del reloj hacia la posición de 180°.

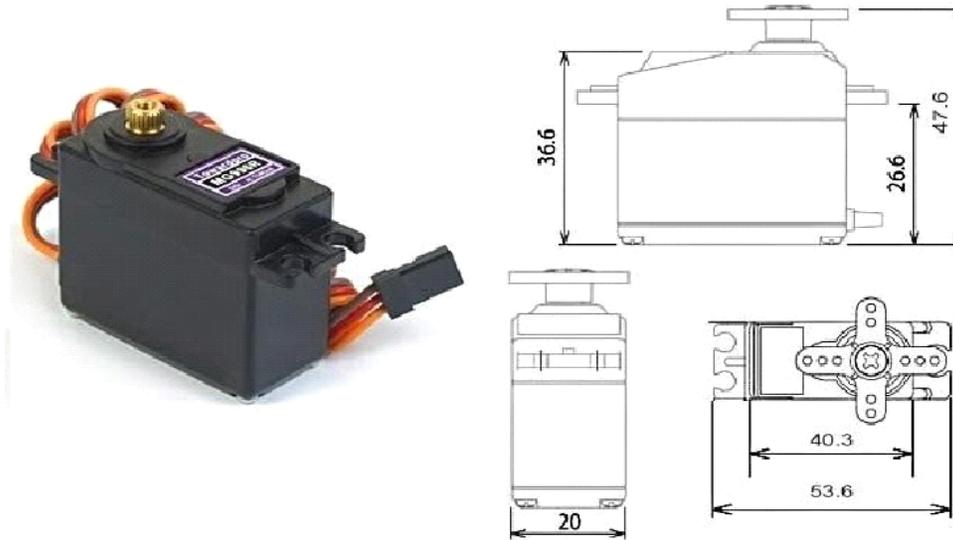
Cuando se les ordena a los servos que se muevan, estos se moverán a la posición y mantendrán esa posición. Si una fuerza externa empuja contra el servo mientras el servo mantiene una posición, el servo se resistirá a salir de esa posición.

La cantidad máxima de fuerza que puede ejercer el servo se denomina par de torsión del servo. Sin embargo, los servos no mantendrán su posición para siempre; el pulso de posición debe repetirse para indicar al servo que se mantenga en posición.



2.14.2. Hoja de datos servomotor MG996R¹

MG996R High Torque Metal Gear Dual Ball Bearing Servo



This High-Torque MG996R Digital Servo features metal gearing resulting in extra high 10kg stalling torque in a tiny package. The MG996R is essentially an upgraded version of the famous MG995 servo, and features upgraded shock-proofing and a redesigned PCB and IC control system that make it much more accurate than its predecessor. The gearing and motor have also been upgraded to improve dead bandwidth and centering. The unit comes complete with 30cm wire and 3 pin 'S' type female header connector that fits most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum and Hitec.

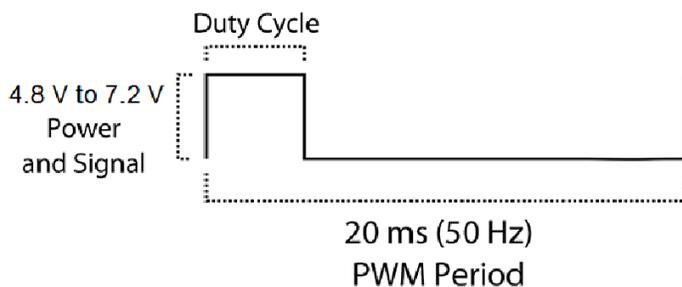
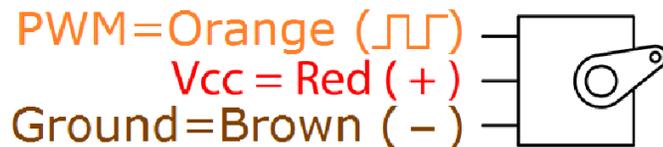
This high-torque standard servo can rotate approximately 120 degrees (60 in each direction). You can use any servo code, hardware or library to control these servos, so it's great for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. The MG996R Metal Gear Servo also comes with a selection of arms and hardware to get you set up nice and fast!

Specifications

- Weight: 55 g
- Dimension: 40.7 x 19.7 x 42.9 mm approx.
- Stall torque: 9.4 kgf·cm (4.8 V), 11 kgf·cm (6 V)
- Operating speed: 0.17 s/60° (4.8 V), 0.14 s/60° (6 V)

MG996R Servo Datasheet pdf (2014) - Bearing Servo. Equivalent,
Catalog Sitio web: <https://datasheetspdf.com/pdf/942981/ETC/MG996R/1>

- Operating voltage: 4.8 V a 7.2 V
- Running Current 500 mA – 900 mA (6V)
- Stall Current 2.5 A (6V)
- Dead band width: 5 μ s
- Stable and shock proof double ball bearing design
- Temperature range: 0 $^{\circ}$ C – 55 $^{\circ}$ C



2.15. Motor DC

El motor DC, también llamado motor de corriente continua, pertenece a la clase de los electromotores y sirve principalmente para transformar la energía eléctrica en energía mecánica. La mayoría de las formas de construcción del motor DC se basa en fuerzas magnéticas y dispone de mecanismos internos de tipo electrónico o electromecánico.

También característico de los motores de corriente continua convencionales es el acumulador, que cambia periódicamente la dirección del flujo de corriente dentro del motor. Una modificación del motor DC clásico que resulta más potente en numerosos ámbitos de aplicación es el motor DC sin escobillas, que prescinde de contactos de roce y ofrece así, entre otras cosas, una vida útil más larga.

A pesar de la creciente utilización de motores AC y de la gran rivalidad con ellos, que, como motores trifásicos, también se consideran electromotores, los motores DC todavía tienen una importancia enorme

hoy en día. Gracias a sus características, (especialmente la regulación de revoluciones y par motor, que es precisa y extremadamente variable), se utilizan en diversas aplicaciones industriales. Ejemplo de ello son los servo - accionamientos de la serie PMA de Harmonic Drive AG, compuestos por un motor de corriente continua altamente dinámico con codificador incremental y la caja de reductor PMG. Son ideales para aplicaciones de la industria de semiconductores, la técnica de la medicina y para ser utilizados en máquinas de medición y de control.

2.15.1. Estructura y modo de funcionamiento del motor DC

La estructura del motor DC clásico sigue un modo de construcción sobrio y emplea pocos componentes. Los elementos principales son el estator y el rotor. Como componente fijo, inmóvil, el estator suele estar compuesto de un electroimán o, (sobre todo en máquinas pequeñas), de un imán permanente. En el interior del estator, con el rotor, también denominado ancla, se encuentra un componente giratorio que en los motores DC convencionales también está hecho con un electroimán. Los motores de corriente continua con la asignación descrita de estator y rotor se denominan de rotor interior o central, mientras que la estructura contraria se corresponde al denominado motor de rotor o inducido exteriores.

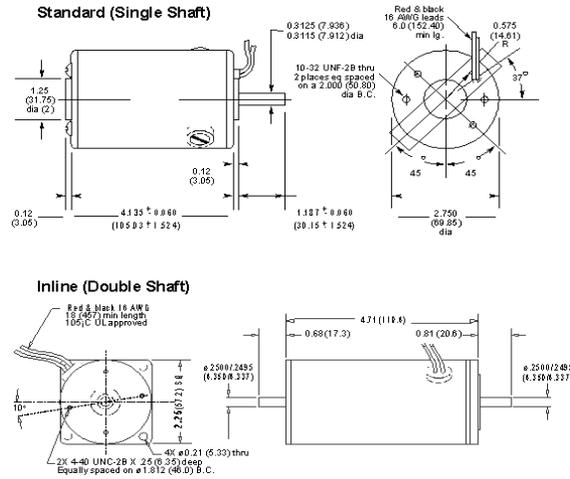
Las bobinas del rotor se conectan mediante un conmutador. Este sirve como alternador de polos y alberga contactos de roce que adquieren una forma tipo cepillo y están fabricados con un material eléctricamente conductor. Los materiales más apreciados son grafito y, dependiendo de la aplicación específica del motor, materiales enriquecidos con metal. En servicio, los contactos de roce son determinados para el funcionamiento del motor DC convencional. Puesto que, cuando fluye corriente continua a través de la bobina del rotor o a través del rotor, este se convierte en un electroimán y genera, gracias a las

características del estator, fuerzas magnéticas. Debido a que los polos del mismo signo se rechazan y los polos diferentes se atraen, el resultado es un movimiento giratorio del rotor, movimiento este que principalmente terminaría en un ámbito neutral. Para garantizar una rotación continua debe producirse una reversión periódica de la dirección de la corriente. Mediante la intervención de las escobillas, el acumulador del motor de corriente continua realiza estas inversiones de polos periódicas.

En lo que respecta a la construcción, la diferencia de motores DC con escobillas también se puede profundizar según el tipo de conmutación de ancla y de bobina de estator. En la máquina de conexión principal, también conocida como máquina de conexión en serie, el inductor (bobina del agente) y la bobina del rotor están conectados en serie, esto es, uno después de otro, lo cual crea la base de la alimentación de corriente alterna. La pieza contrapuesta o contraparte es la máquina de conexión paralela, en la que ambas bobinas se conectan paralelamente. Una combinación de máquina de conexión en serie y máquina de conexión paralela se encuentra en el motor combinado o compuesto (compound). Dependiendo de la configuración da soporte a un modo de funcionamiento diferente y se beneficia de las ventajas de ambos tipos de construcción.

2.15.2. Hoja de datos motor DC

D 24 Volt DC Motor Specifications



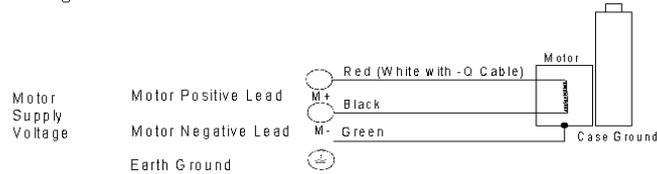
Electrical Data

Rated Voltage	V	24
Max. Continuous Current	A	4.5
Max. Operating Voltage	V	36
Inductance	mH	2.0
K_t Torque Constant ($\pm 10\%$)	oz-in/A [N-m/A]	8.9 [0.062]
K_v Voltage Constant ($\pm 10\%$)	V/kRPM	6.5
Winding Resistance @ Ambient	Ohms	1.0

Mechanical Data

Continuous Stall Torque	oz-in [N-m]	40 [0.28]
No-load Speed at Rated Voltage	RPM	3600
No-load Current	A	0.5
Rotor Inertia	oz-in- 2 [kg-cm 2]	0.018 [1.3]
Max. Winding Temperature	$^{\circ}$ F [$^{\circ}$ C]	180 [82]

Motor Wiring



2.15.3. Torque

El torque se considera una magnitud vectorial, siendo más precisos, lo podríamos definir como el momento de fuerza o momento dinámico. Es decir, es la medida de la fuerza que se aplica a una varilla y que sirve para hacer girar un objeto. Este momento de fuerza se mide en newtons/metro.

Muchas veces se confunde el término de torque con el de potencia, aunque están relacionados significan cosas diferentes. El torque mide

las revoluciones por minuto, y la potencia hace referencia a la velocidad del vehículo.

Es decir, el torque hace que el motor pueda mover el vehículo, sin afectar a la velocidad a la que luego vaya a desplazarse dicho vehículo. Por lo tanto, un camión muy pesado y con mucha carga, debe tener un torque mucho más fuerte que el que necesitaría un turismo.

Para determinar el torque usamos la siguiente fórmula.

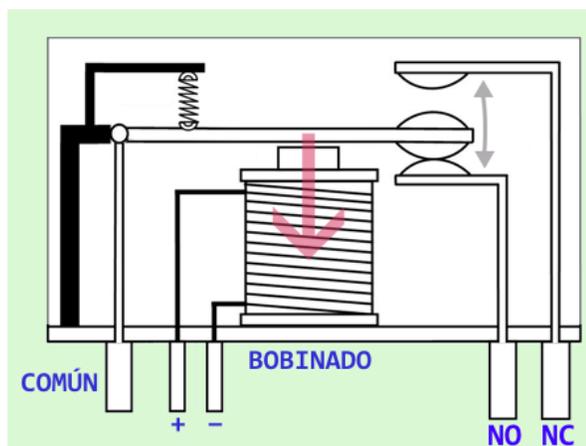
$$Hp = (Torque)(RPM)$$

$$\frac{HP}{RPM} = Torque$$

2.16. Modulo relé

Un relé es un interruptor mecánico operado eléctricamente que se puede encender o apagar, dejando pasar la corriente o no, y se puede controlar con voltajes bajos, como los 5V utilizados en la alimentación de un Arduino.

«NO» corresponde al inglés Normally Open (normalmente abierto) y «NC» a Normally Closed (normalmente cerrado). Un relé desactivado tiene unidos entre sí los contactos COMÚN y NC, y cuando se lo activa aplicando corriente a su bobina, quedan unidos entre sí los contactos COMÚN y NO.



Controlar un módulo de relé con el Arduino es lo mismo que controlar cualquier otra salida digital. El consumo de corriente de una de sus entradas es el mismo que se necesita para encender un led.

Aquí vemos cómo los relés están marcados en la parte de abajo, del mismo modo que se puede ver el símbolo con la disposición de los contactos del relé en la salida de alto voltaje del módulo:



El módulo de relé debe ser alimentado con 5V, lo que es apropiado para usar con un Arduino. Hay otros módulos de relé que funcionan con 3,3V, ideal para ESP32, ESP8266 y otros microcontroladores, y también hay modelos de 12V.



2.16.1. Conexión Módulo Relé

La siguiente figura muestra la función de los contactos del módulo de relé.



Los seis pines en el lado izquierdo del módulo del relé controlan voltaje alto, y los pines en el lado derecho se conectan al componente que entrega bajo voltaje, como los pines digitales de un Arduino u otro microcontrolador.

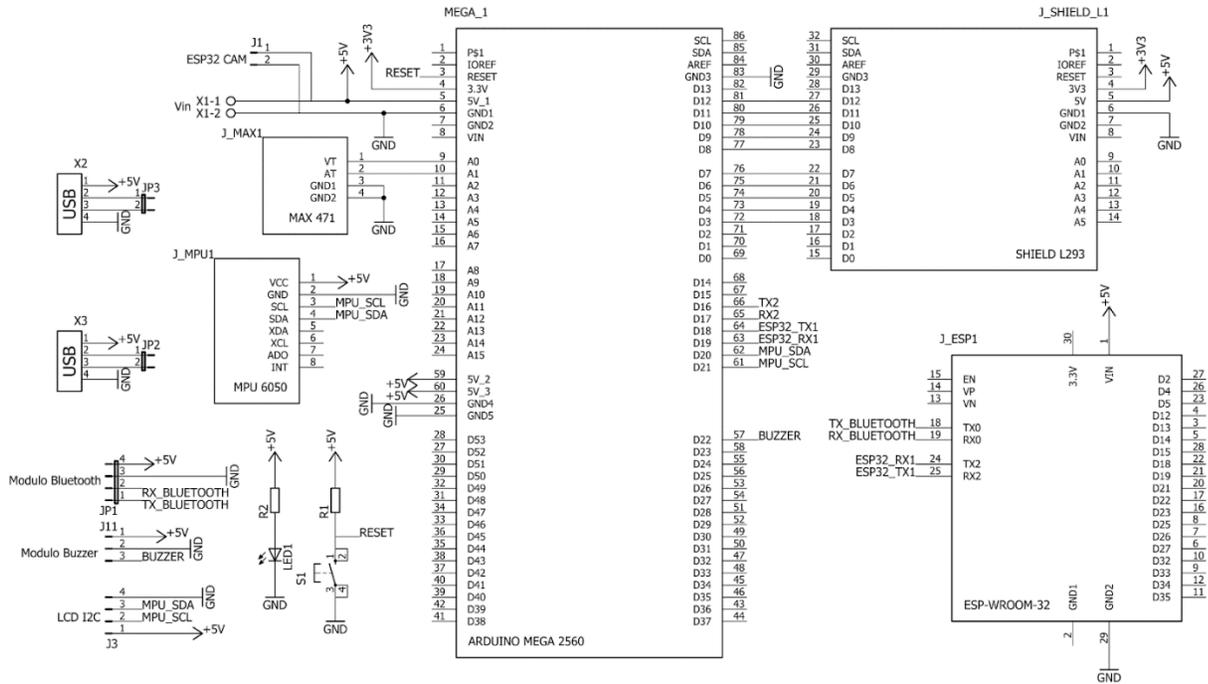
CAPITULO III

ESQUEMAS DE CIRCUITOS ELECTRÓNICOS, DIBUJOS Y PLANOS DE ESTRUCTURA DE SOPORTE, CÓDIGOS DE PROGRAMACIÓN

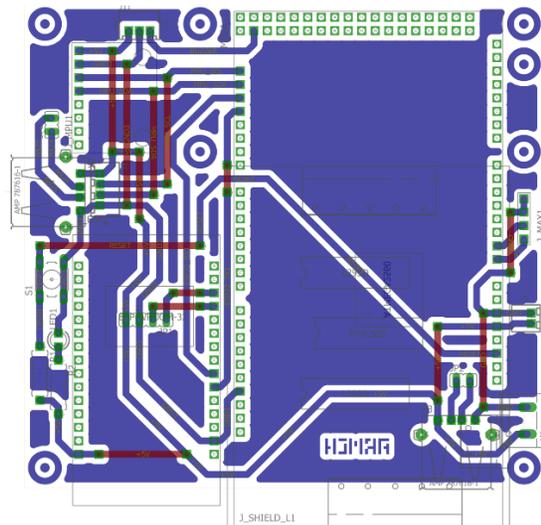
3.1. Esquemas de circuitos electrónicos

3.1.1. Shield Arduino Mega

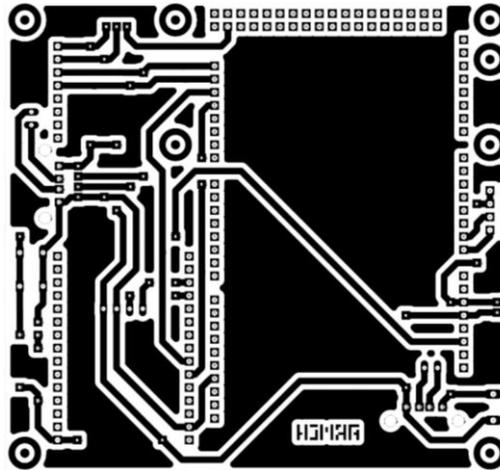
3.1.1.1. Diagrama esquemático



3.1.1.2. Diseño del circuito impreso (PCB)

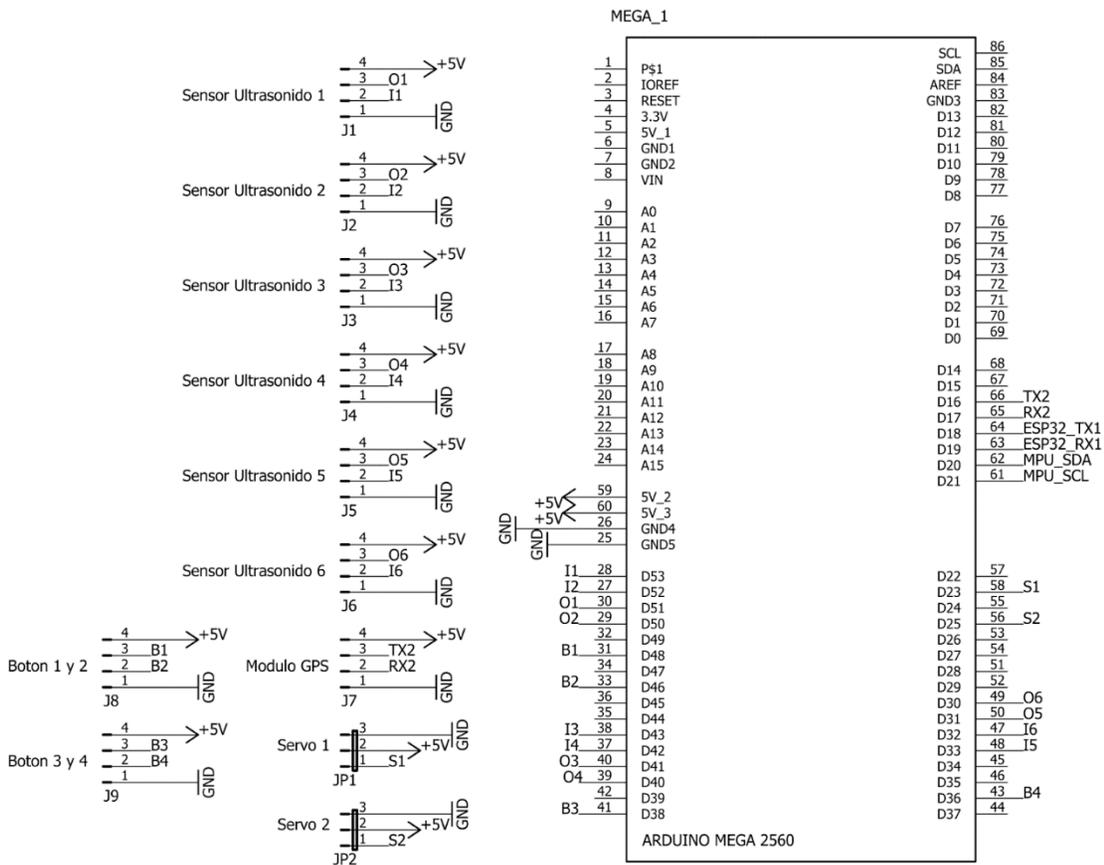


3.1.1.3. Diseño del circuito impreso (PCB)

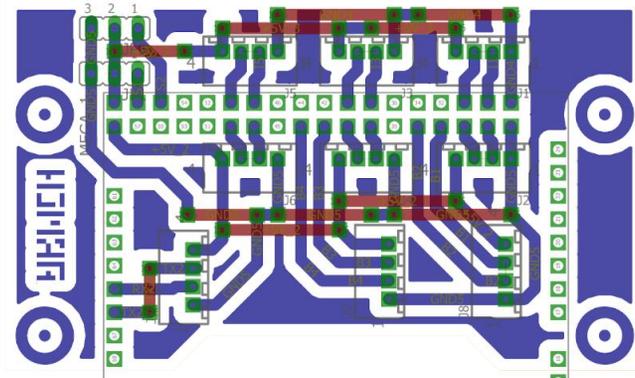


3.1.2. Shield entradas y salidas Arduino

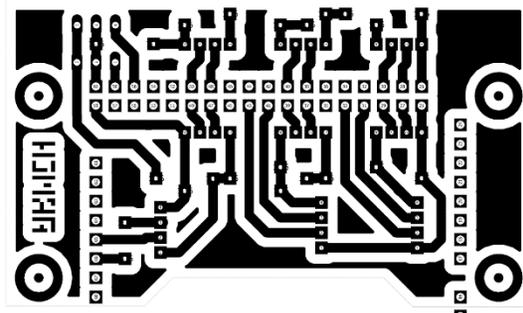
3.1.2.1. Diagrama esquemático



3.1.2.2. Diseño del circuito impreso (PCB)

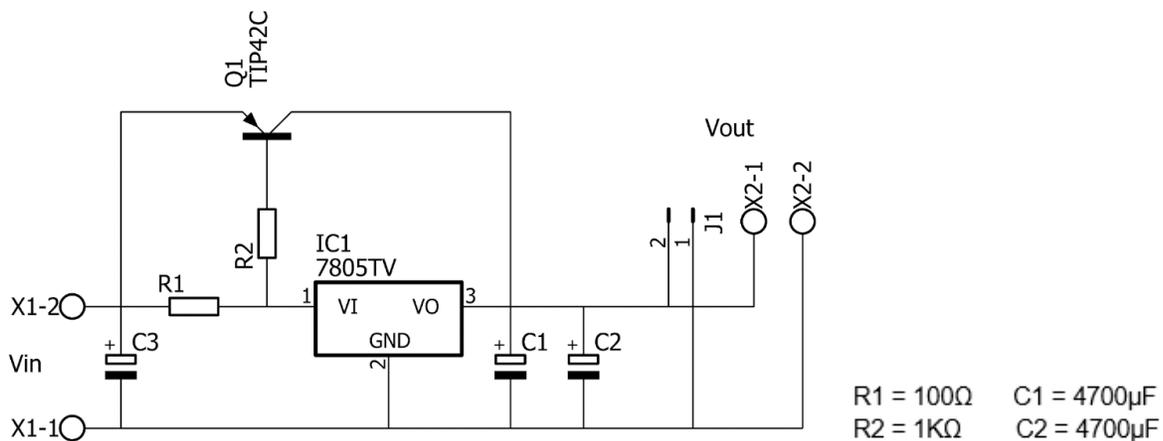


3.1.2.3. Shield entrada y salida Arduino Mega: Diseño del circuito impreso (PCB)

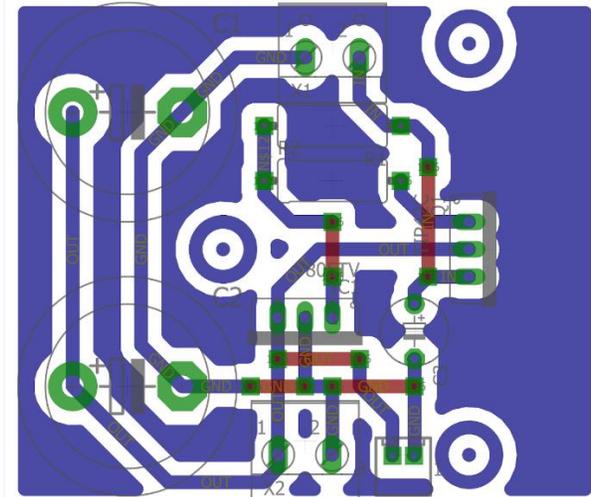


3.1.3. Regulador de 12 a 5V

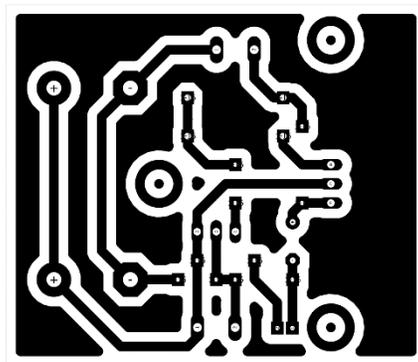
3.1.3.1. Diagrama esquemático



3.1.3.2. Diseño del circuito impreso (PCB)

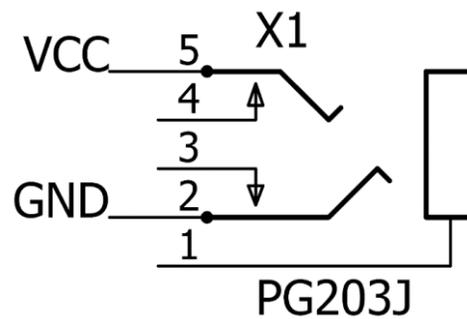


3.1.3.3. Diseño del circuito impreso (PCB)

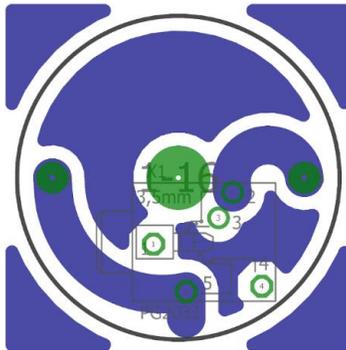


3.1.4. Shield para aspiradora

3.1.4.1. Diagrama esquemático



3.1.4.2. Diseño del circuito impreso (PCB)

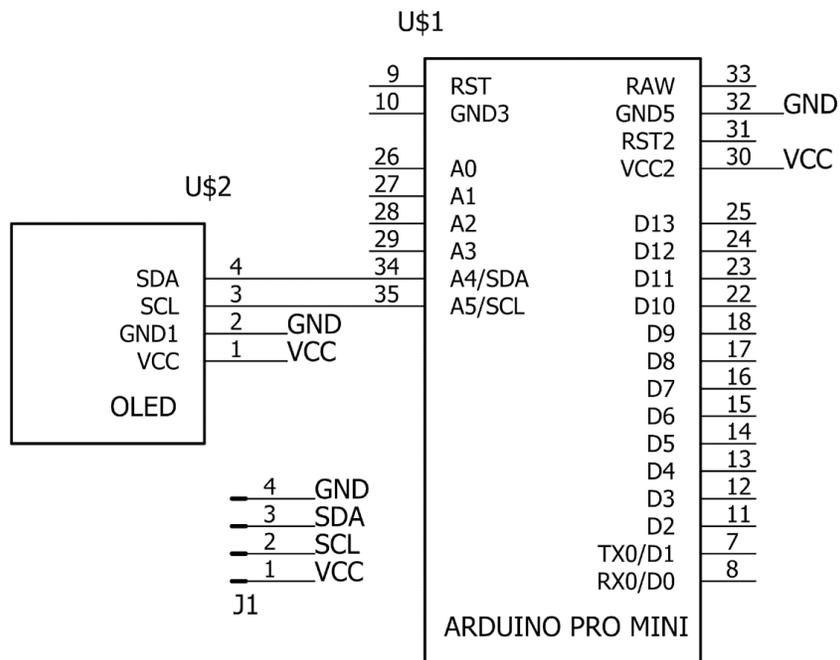


3.1.4.3. Diseño del circuito impreso (PCB)

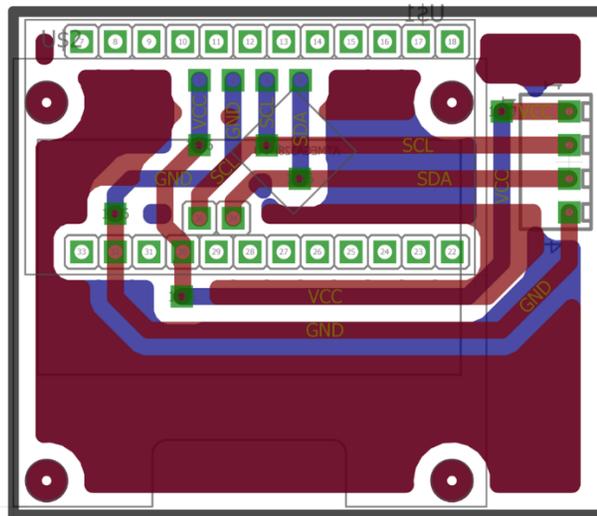


3.1.5. Shield Pro Mini OLED

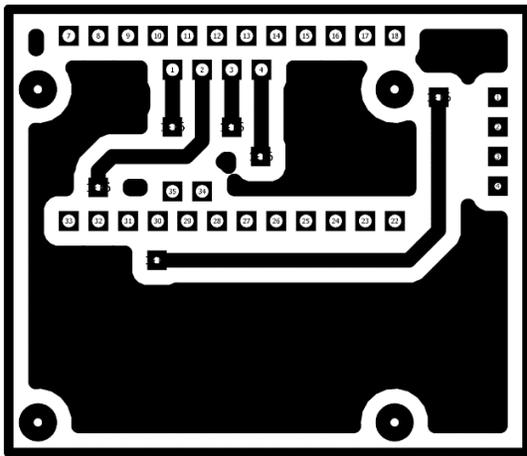
3.1.5.1. Diagrama esquemático



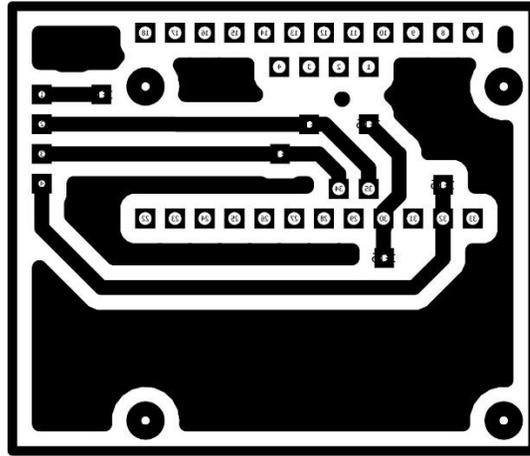
3.1.5.2. Diseño del circuito impreso (PCB)



3.1.5.3. Diseño del circuito impreso (PCB)



Abajo



Arriba

3.2.2. Código de programación IoT con ESP32 y Blynk.

El código de programación consta de 196 líneas, la lógica del código es base a llamada de funciones, en el cual nos comunicamos con la base de datos de Blynk y posterior mandamos ordenes al Arduino Mega.

```
2  * Proyecto Robot de Limpieza
3  * Grupo:1
4  *Codigo: Comunicación IoT con Esp32 y Blynk
5  * Autores:
6  * Flores Escalón, Andrea Mariela
7  * Gonzales Pocasangre, Victoria Michell
8  * Joya Chávez, José Francisco
9  * Maravilla Martínez, José Luis
10 * Mulato Chavés, Álvaro Rutilio
11 */
12
13 #define BLYNK_PRINT Serial
14
15 /* Fill-in your Template ID (only if using Blynk.Cloud) */
16 /*Estos codigos son muy importantes para la comunicación entre blynk y el hardware, para poder
17 la comunicación por medio de características de comunicación entre la nube y nuestro ESP32
18 */
19 #define BLYNK_TEMPLATE_ID "TMPL2_8sJ82B" //Se le coloca el ID del proyecto brindado por Blynk
20 #define BLYNK_DEVICE_NAME "Control 2" //El nombre del proyecto en la plataforma
21 #define BLYNK_AUTH_TOKEN "3Qpnf6-812tIKL-EuS7irruq2M5y2EU" //Un Authtoken de seguridad para proteger nuestra comunicación
22
23 #include <WiFi.h> //Se incluye la librería WiFi para poder comunicarnos con nuestro internet portatil
24 #include <WiFiClient.h> //Se incluye la librería WiFiClient para poder comunicarnos con el servidor
25 #include <BlynkSimpleEsp32.h> //Se incluye la librería BlynkSimpleEsp32 para comunicarnos con la base de datos de Blynk
26 #include "Separador.h"
27
28 Separador code;
29
30 //Variables para establecer comunicación e inicializar librerías
31 char auth[] = BLYNK_AUTH_TOKEN; //El Authtoken que nos permitira conectarnos al servidor de blynk
32 char ssid[] = "ROBOT DE LIMPIEZA"; //El nombre de la red a conectar
33 char pass[] = "ProyectoX_2022"; //Contraseña de la red
34
35 #define RXp2 16 //comunicacion RX Serial 2
36 #define TXp2 17 //Comunicacion TX Serial 2
37
38 int timer;
39
40 //-----Variables de procesos-----
41 int value;
42 int Data1;
43 int barra;
44
45 bool result;
46 int state;
47 int state_2;
48 char receive_data;
49 String batt;
50 String Str_Flon;
51 String lat;
52 String lon;
53 int x;
54 String gps;
55 float Fiat;
56 float Flon;
57 WidgetMap MyMap(V10);
58
59
60
61 void setup()
62 {
63
64   Serial.begin(115200); //Iniciamos el Serial a 115200 baudios
65   Serial2.begin(9600, SERIAL_8N1, RXp2, TXp2); //Se inicializa el Serial 2 a 9600 baudios para comunicación con arduino
66   Blynk.begin(auth, ssid, pass); //Se inicializa la librería de blynk
67
68 }
69
70 void loop() {
71
72   result = Blynk.connected(); //Verificamos que el esp32 se encuentra conectado a la nube
```

Para más información ir ANEXO 3.

3.2.3. Código de programación Arduino Mega y Sistema de Control.

El código consta de 858 líneas de programación, la lógica del programa es en base a llamada de funciones, el cual se encarga de tener lectura de todos los sensores y dar actuación a los sistemas de potencia.

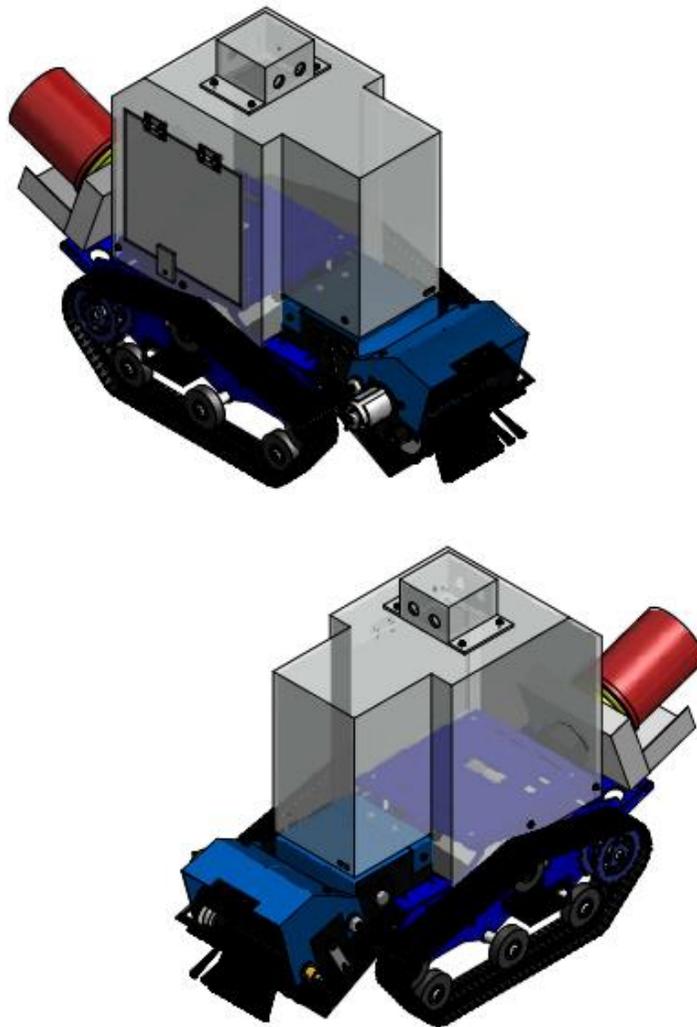
```
1  /* Escuela Especializada en Ingeniería ITCA FEPADE
2  * Proyecto Robot de Limpieza
3  * Grupo:1
4  * Código: Control del sistema de potencia y control de sensores
5  * Autores:
6  * Flores Escalón, Andrea Mariela
7  * Gonsales Pocasangre, Victoria Michell
8  * Joya Chávez, José Francisco
9  * Maravilla Martínez, José Luis
10 * Mulato Chavés, Álvaro Rutilio
11 */
12
13 #include <AFMotor.h>
14 #include <Servo.h>
15 #include <Wire.h>
16 #include "I2Cdev.h"
17 #include "MPU6050.h"
18 #include "Wire.h"
19
20 AF_DCMotor motor_R(4);
21 AF_DCMotor motor_L(3);
22 AF_DCMotor motor_clean(2);
23 AF_DCMotor motor_aspiradora(1);
24
25 Servo position_clean;
26 Servo servo_camera;
27
28 MPU6050 sensor;
29
30 int mov_auto;
31 int contador;
32
33 //-----VARIABLE DE COMUNICACION MPU6050
34
35 // Valores RAW (sin procesar) del acelerometro y giroscopio en los ejes x,y,z
36
37 //-----VARIABLE DE COMUNICACION MPU6050
38
39 // Valores RAW (sin procesar) del acelerometro y giroscopio en los ejes x,y,z
40 int ax, ay, az;
41 int gx, gy, gz;
42 unsigned long tiempo_prev;
43 float dt;
44 float ax_m_s2;
45 float ay_m_s2;
46 float az_m_s2;
47 float gx_deg_s;
48 float gy_deg_s;
49 float gz_deg_s;
50 float accel_ang_x;
51 float accel_ang_y;
52 float ang_x, ang_y;
53 float ang_x_prev, ang_y_prev;
54
55 //-----VARIABLE DE COMUNICACION I2C
56 byte enviar[2];
57 int wire_data;
58 int32_t loop_timer;
59 unsigned long timer;
60 unsigned long timer2;
61 int mode=1;
62 const int CONTROL_REMOTE=1;
63 const int LIMPIEZA_MANUAL=2;
64 const int SMART_CLEAN=3;
65 const int BAJA_BATERIA=4;
66 const int COLISION=5;
67 //-----VARIABLE DE POTENCIA
68 int vel_1 = 200;
69 int vel_2 = 200;
70 int vel_izquierda = 240;
```

Para más información ir ANEXO 3.

3.3. Dibujos y planos de estructura de soporte

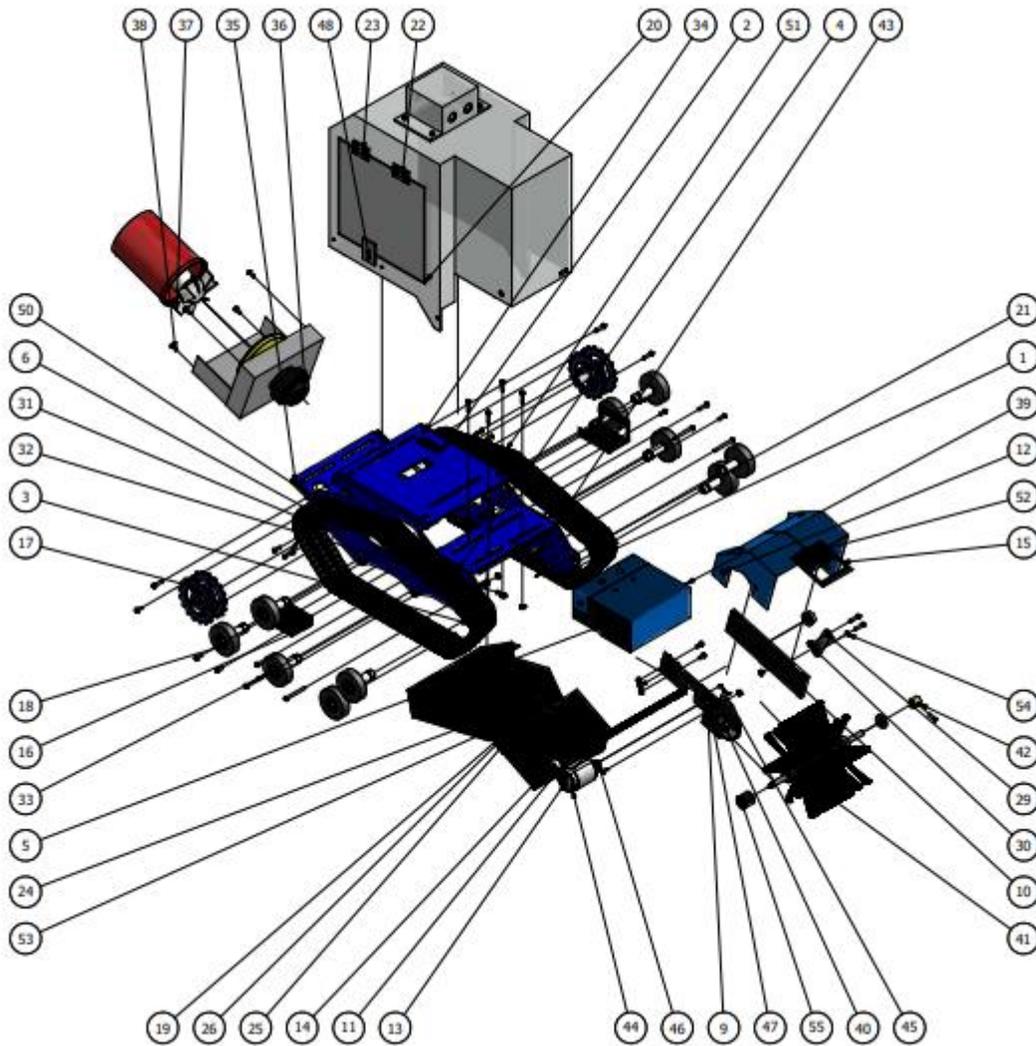
En la presente se adjuntarán planos pieza por pieza en memoria USB y link de Google Drive.

3.3.1. Ensamble general



Integrantes		ESCUELA ESPECIALIZADA EN INGENIERÍA ITCA - FEPADE		
Flores Escalón Andrea Mariela		Especialidad: Ingeniería Mecatrónica		
Gonzalez Pocasangre Victoria Michell		Proyecto de Innovación: ROBOT DE LIMPIEZA		
Joya Chávez José Francisco				
Maravilla Martínez José Luis				
Mulato Chávez Álvaro Rutilio				
		SIZE C	DWG NO Ensamble	
		SCALE 1/2	UNIDADES: mm	SHEET 1 OF 75

3.3.2. Explotado general



Integrantes		ESCUELA ESPECIALIZADA EN INGENIERÍA ITCA - FEPADE	
Flores Escalón Andrea Mariela		Especialidad: Ingeniería Mecatrónica	
Gonzalez Pocasangre Victoria Michell		Proyecto de Innovación: ROBOT DE LIMPIEZA	
Joya Chávez José Francisco			
Maravilla Martínez José Luis			
Mulato Chávez Álvaro Rutilio			
		SIZE C	DWG NO Explotado general
		SCALE 0.25:1	UNIDADES: mm SHEET 2 OF 75

3.3.3. Listado general de piezas

LISTA DE PIEZAS		
ELEMENTO	CTDAD	Nº DE PIEZA
1	1	Soporte lateral derecho
2	1	Base 1
3	1	Soporte lateral izquierdo
4	1	Base 2
5	1	Protector de servo
6	2	Brazo escobilla 1
9	1	Brazo escobilla 2
10	1	Soporte escobillas
11	5	Arandela
12	1	Protector escobillas
13	1	Tornillo sin fin
14	3	Tuerca tornillo sin fin
15	1	Soporte de sensor frontal
16	2	Soporte de sensores laterales
17	2	Engrane de llanta completo
18	10	Rueda completa
19	1	Depósito
20	1	Estructura 1
21	6	Pin de bisagra
22	6	Bisagra centro
23	6	Bisagra laterales
24	1	Tapadera
25	1	Bandeja de depósito
26	1	Pasador parte 1
27	1	Pasador parte 2
28	1	Pin de pasador
29	1	Union de bandeja
30	1	Ancla de union de bandeja
31	20	Tornillo general 2
32	46	Tuerca de tornillo general
33	4	Tornillo general 1
34	14	Tornillo de proyecto
35	1	Base 3
36	1	Caja aspiradora
37	1	Soporte de motor
38	25	Tornillo general 3
39	1	Perno
40	1	Engranaje de escobilla
41	1	Escobilla completa
42	1	Seguro de escobillas
43	4	Ancla de deposito
44	1	Motor DC 2
45	2	Tornillo de motor DC 2
46	1	Engrane de motor
47	1	Engrane de escobillas 2
48	1	Puerta
50	90	Cadena con pin I
51	90	Cadena con pin D
52	4	Remache de protector de escobillas
53	22	Remache de deposito
54	1	Arandela de plástico
55	1	Goma sola

CAPITULO IV

**MANUALES ROBOT DE
LIMPIEZA**

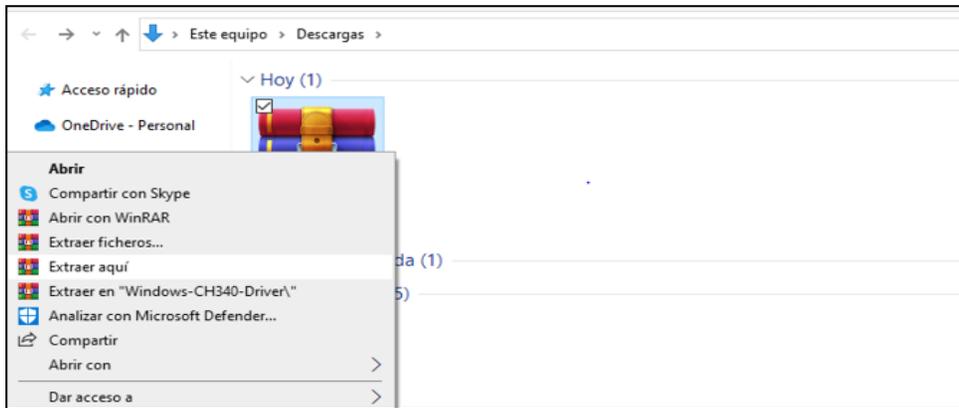
4.1. Manual de instalación Drivers CH340 Y CP210X

4.1.1. Manual de instalación Driver CH340

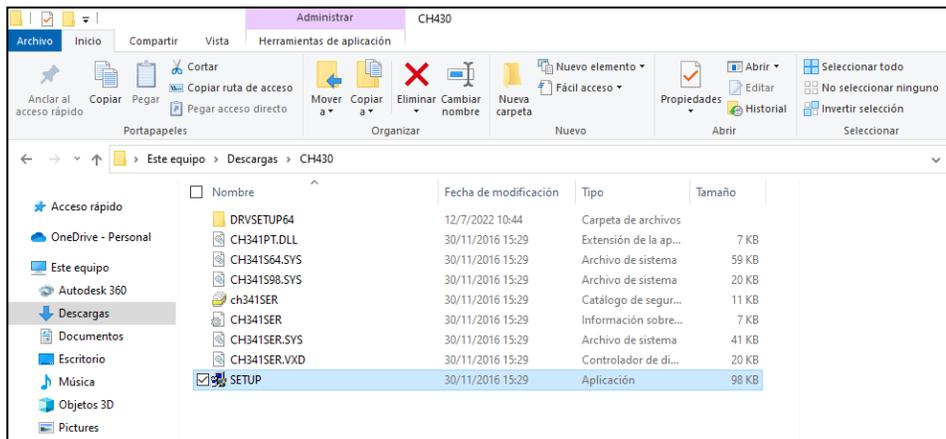
Es necesario realizar la instalación de este driver, en caso de que se presente la dificultad al utilizar el controlador genérico de Arduino Mega y que este no sea reconocido por el software de Arduino, para solucionar este problema deben realizarse los siguientes pasos:

- Paso 1: Nos dirigimos a la carpeta en la que se encuentra el archivo del instalador con el nombre "Windows-CH340-Driver".

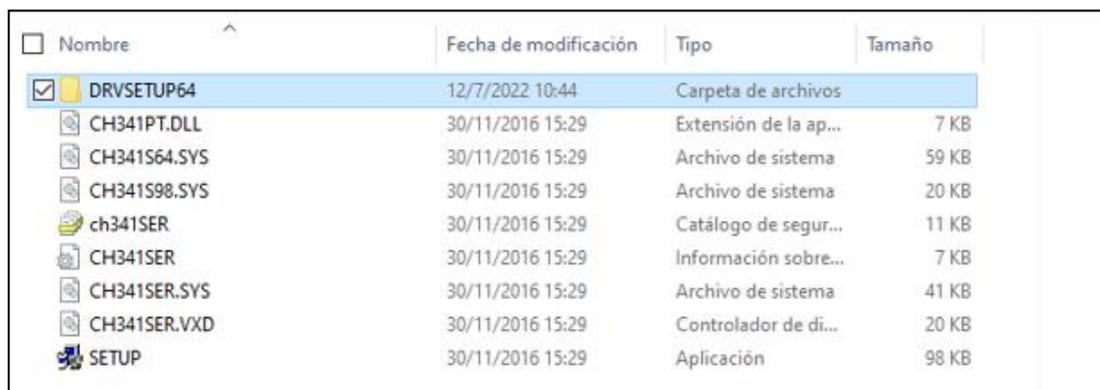
Extraeremos los documentos de instalación de la siguiente manera, ubicamos el archivo comprimido y hacemos clic derecho, luego seleccionamos la opción "extraer aquí".



Nos aparecerán los siguientes archivos



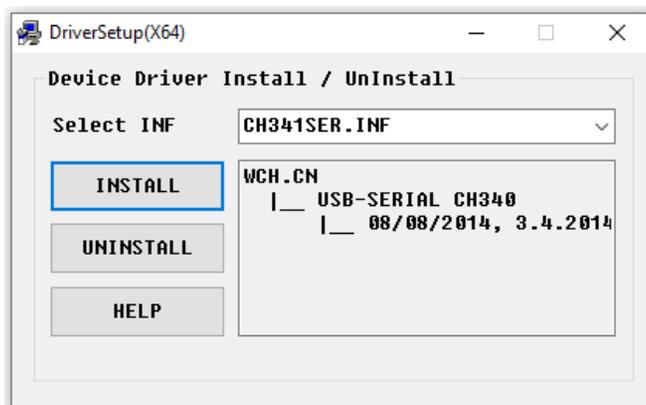
- Paso 2: En caso de que nuestro sistema operativo sea de 32 bits dar doble clic en el documento "SETUP" y en caso de que nuestro sistema operativo sea de 64 bits, abrir la carpeta DRVSETUP64.



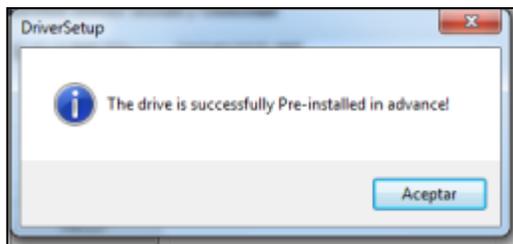
- Paso 3: Hacer clic derecho al documento que aparecerá adentro de la carpeta y luego seleccionar "Ejecutar como administrador".



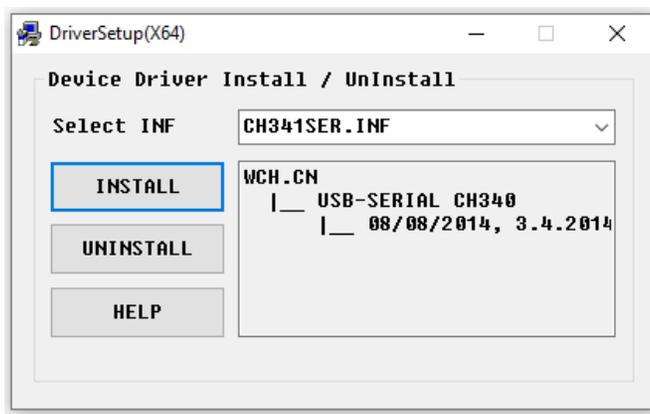
- Paso 4: Luego nos aparecerá un mensaje diciendo "¿Desea permitir que este programa realice cambios en el equipo?", nosotros seleccionaremos "SI", luego nos aparecerá esta sub-pantalla y hacemos clic en la opción "INSTALL"



- Paso 5: Luego de unos segundos nos aparecerá la siguiente pantalla y hacemos clic en aceptar.



- Paso 6: Finalmente cerramos el menú principal del instalador.

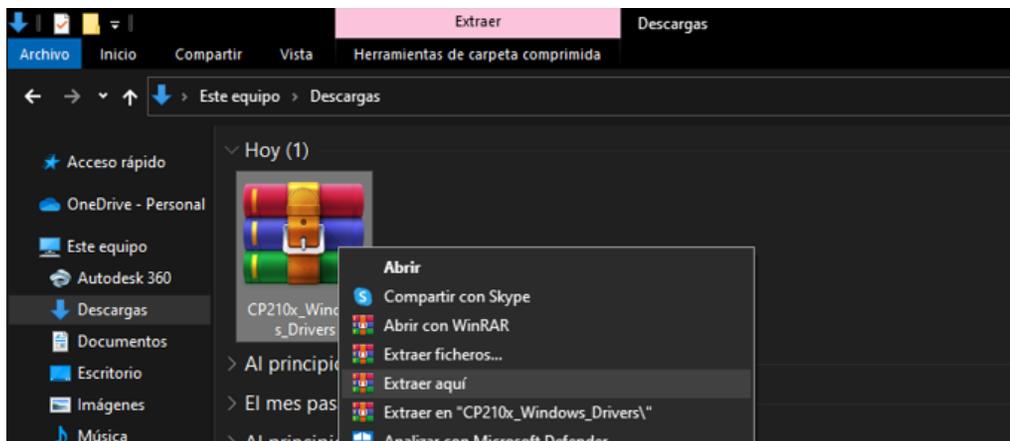


4.1.2. Manual de instalación Driver CP210X

Los controladores del puerto COM virtual (VCP) del puente USB a UART CP210x son necesarios para el funcionamiento del dispositivo como un puerto COM virtual para facilitar la comunicación del host con los productos CP210x. A continuación, se presentará una serie de pasos a seguir para realizar la instalación del driver CP210X.

- Paso 1: Nos dirigimos a la carpeta en la que se encuentra el archivo del instalador con el nombre "CP210x_Windows_Drivers".

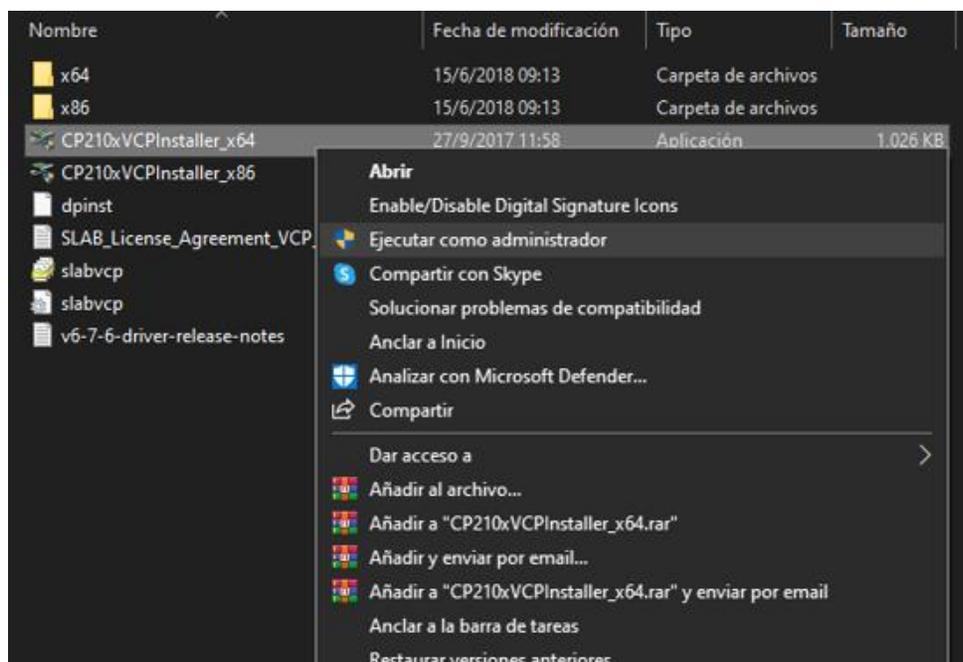
Extraeremos los documentos de instalación de la siguiente manera, ubicamos el archivo comprimido y hacemos clic derecho, luego seleccionamos la opción "extraer aquí"



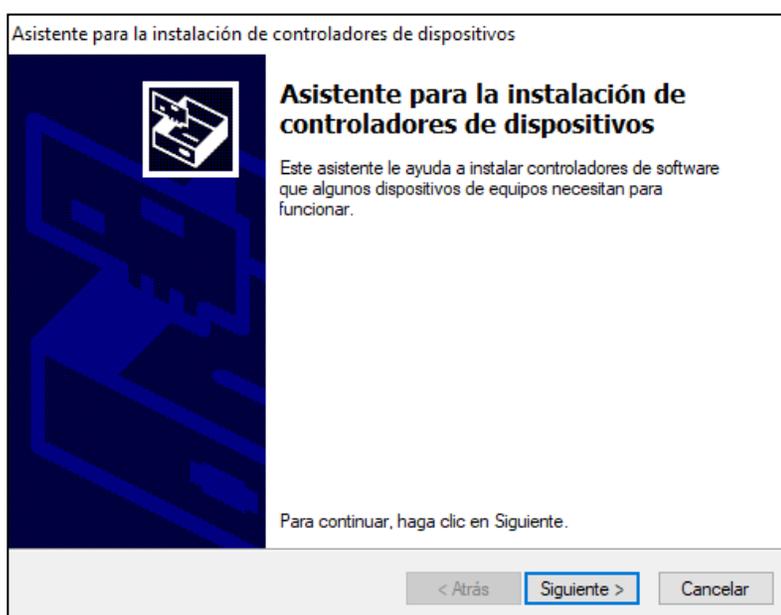
Nos aparecerán los siguientes archivos:

Nombre	Fecha de modificación	Tipo	Tamaño
x64	15/6/2018 09:13	Carpeta de archivos	
x86	15/6/2018 09:13	Carpeta de archivos	
CP210xVCPInstaller_x64	27/9/2017 11:58	Aplicación	1,026 KB
CP210xVCPInstaller_x86	27/9/2017 11:58	Aplicación	903 KB
dpinst	27/9/2017 11:45	Documento XML	12 KB
SLAB_License_Agreement_VCP_Windows	27/9/2017 11:46	Documento de te...	9 KB
slabvcp	1/6/2018 14:35	Catálogo de segur...	11 KB
slabvcp	1/6/2018 14:35	Información sobre...	8 KB
v6-7-6-driver-release-notes	15/6/2018 12:51	Documento de te...	16 KB

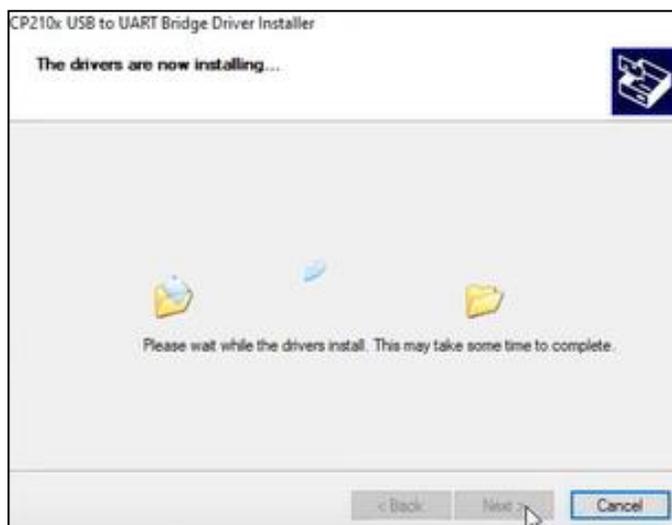
- Paso 2: En este caso utilizaremos el archivo de instalación de 64 bits, hacer clic derecho al documento que aparecerá adentro de la carpeta con el nombre "CP210xVCPInstaller_x64" y luego seleccionar "Ejecutar como administrador".



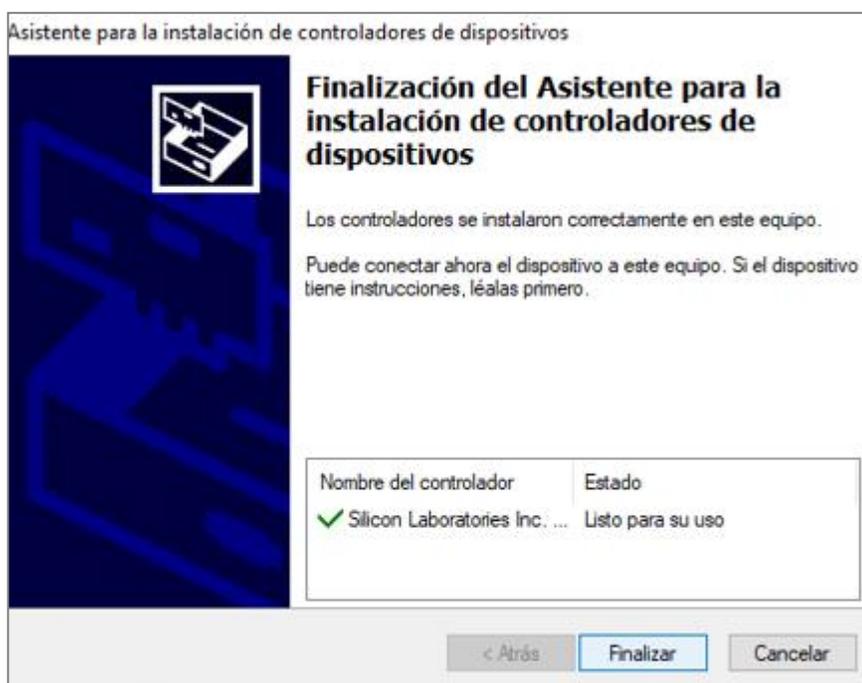
- Paso 3: Luego nos aparecerá un mensaje diciendo "¿Desea permitir que este programa realice cambios en el equipo?", nosotros seleccionaremos "SI", y posteriormente aparecerá el siguiente cuadro de diálogo del asistente de instalación y haremos clic en la opción que dice "Siguiente"



- Paso 4: Posteriormente aparecerá la siguiente pantalla que nos indica que nuestro driver se está instalando, esperaremos unos segundos mientras se completa la instalación y luego haremos clic en "Siguiente"



- Paso 6: Luego de que se complete la instalación, nos aparecerá el siguiente mensaje del asistente de instalación y será necesario hacer clic en la opción "Finalizar", con ello ya tendremos instalado nuestro Driver CP210x,



4.2. Manual de usuario de APP Blynk

Conexión

Paso 1: Descarga la aplicación

Busque y descargue la App "Blynk IoT" desde Play Stores o Apple Stores, o bien, escanee el siguiente código QR para obtenerla.



Android



iOS

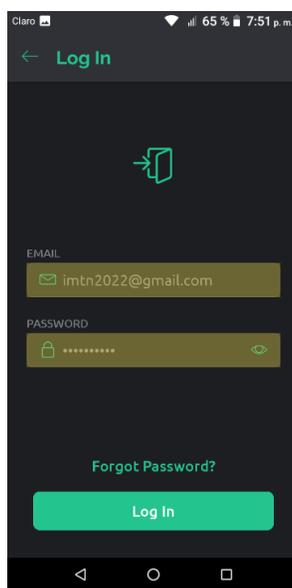
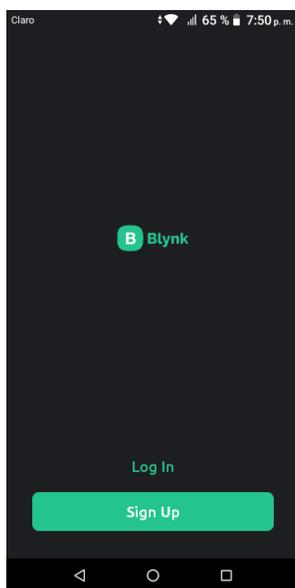
Paso 2: Entrar a la app

Paso 3: Registrarse

Para acceder el control de administrador del robot de limpieza utilice la cuenta del proyecto

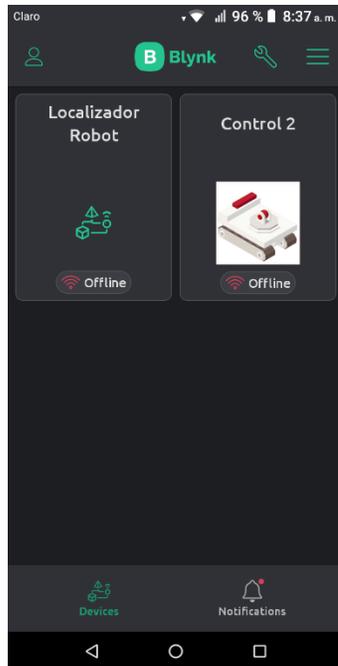
Correo: **imtn2022@gmail.com**

Contraseña de blynk: **imtn-@2022**

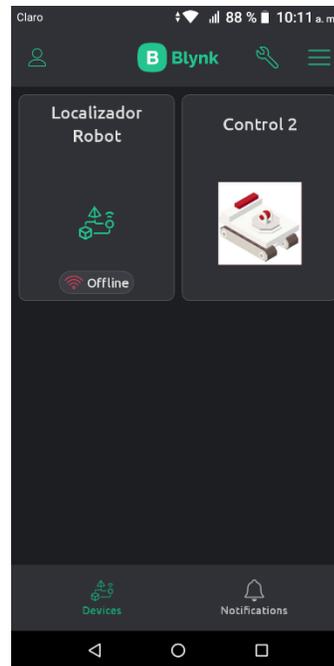


Paso 4: Encender el robot

Paso 5: Observar que el Robot se encuentre conectado en "Control 2"

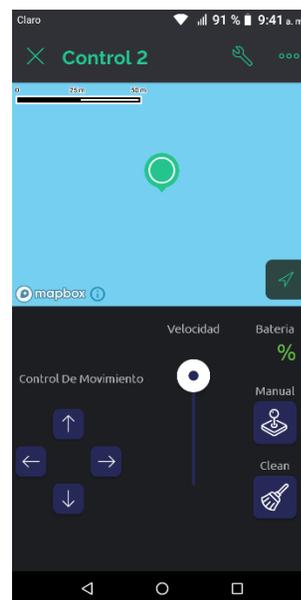
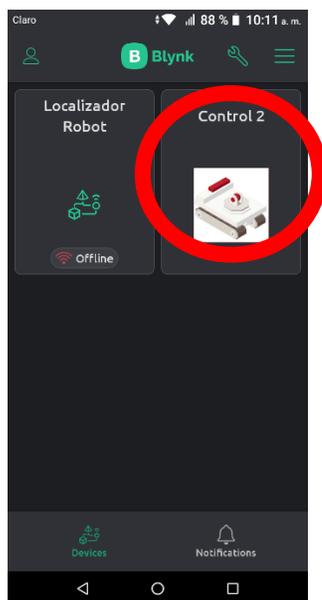


Fuera de línea

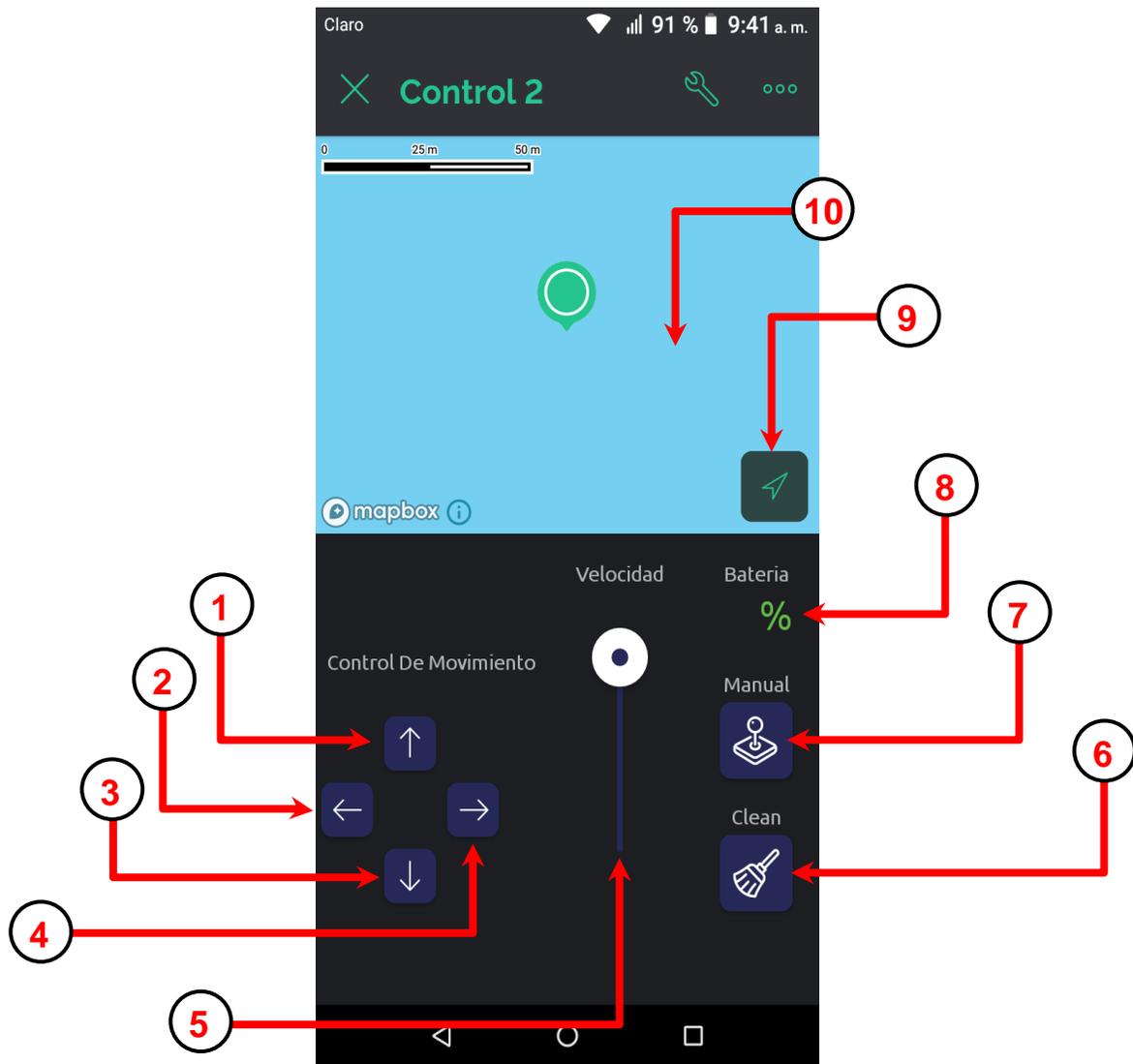


En línea

Paso 6: Dar clic en "Control 2" para ingresar al panel de control del Robot de Limpieza



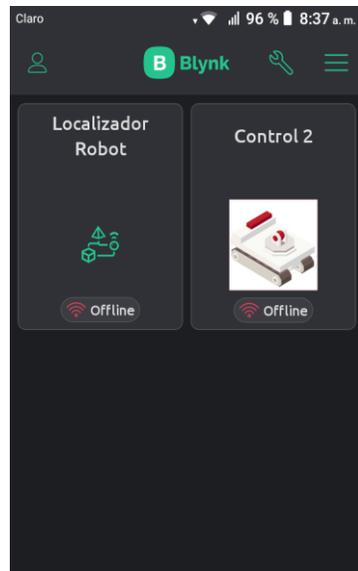
▪ Interfaz



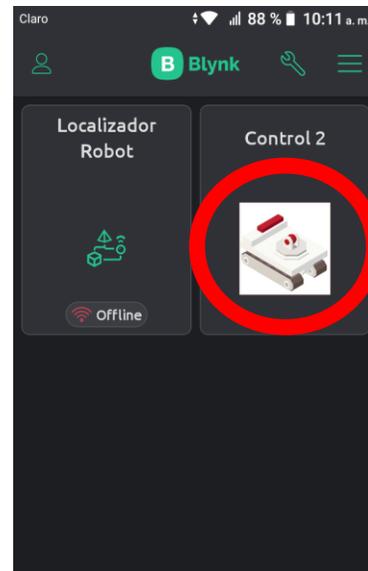
- 1- Botón Adelante (switch)
- 2- Botón izquierda (push)
- 3- Botón Retroceso (switch)
- 4- Botón derecha (push)
- 5- Barra de velocidad
- 6- Botón modo automático (push)
- 7- Botón modo manual (push)
- 8- Indicador de batería
- 9- Botón para localizar mi dispositivo (Si está activado mi ubicación del celular)
- 10- Mapa para visualizar los dispositivos (Si incluye modulo GPS).

▪ Proceso

- Paso 1: Encender el robot
- Paso 2: Observar que el Robot se encuentre conectado en el proyecto "Control 2"

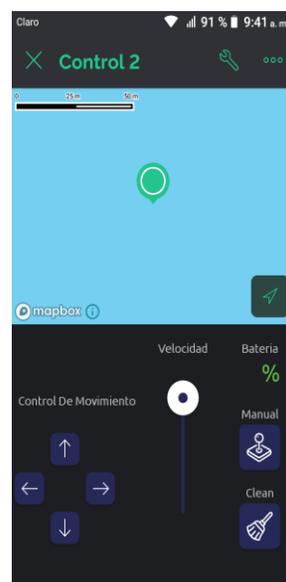
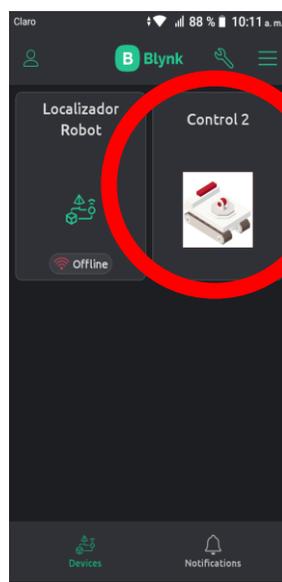


Fuera de línea



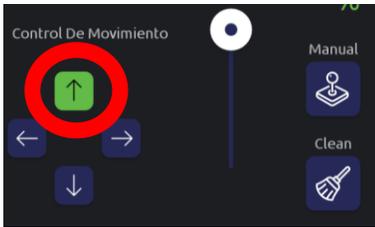
En línea

- Paso 3: Dar clic en "Control 2" para ingresar el tablero de Robot de Limpieza



▪ Descripción de los elementos

1- Botón Adelante (switch)



Este botón nos permite dar la orden de desplazarse hacia adelante, este tipo de botón es switch, además si el robot se acerca a un obstáculo, el sensor enviará la una señal al Arduino para detener el Robot. Para volver a enviar la orden de moverse deberá desactivar el botón y volver activar.

2- Botón izquierda (push)

Este botón nos permite dar la orden de desplazarse hacia la izquierda, este tipo de botón es push, deberá mantener presionado el botón.

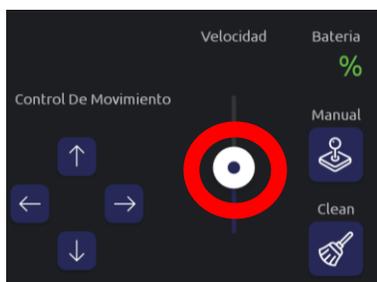
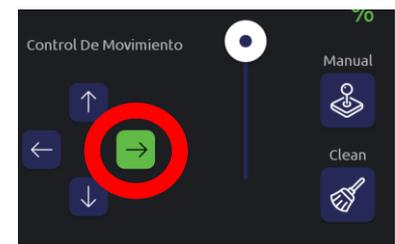


3- Botón Retroceso (switch)

Este botón nos permite dar la orden de desplazarse hacia atrás, este tipo de botón es switch, además si el robot se acerca a un obstáculo, el sensor enviará la una señal al arduino para detener el Robot. Para volver a enviar la orden de moverse deberá desactivar el botón y volver activar.

4- Botón derecha (push)

Este botón nos permite dar la orden de desplazarse hacia la derecha, este tipo de botón es push, deberá mantener presionado el botón.



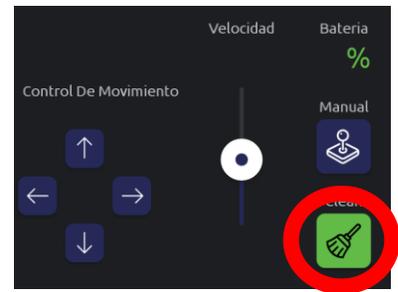
5- Barra de velocidad

Este widget nos permite ajustar las velocidades del motor del sistema oruga.

6- Botón modo automático (push)

Este botón nos permite usar el modo de limpieza automática.

- Baja el brazo, activa la escobilla y la aspiradora
- El robot realiza una limpieza de manera automática.



7- Botón modo manual (push)

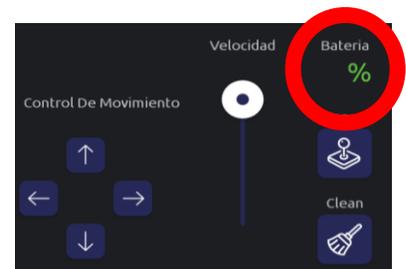
Este botón nos permite usar el modo de limpieza manual.

- Baja el brazo, activa la escobilla y la aspiradora
- El robot se maneja de manera manual para realizar la limpieza.



8- Indicador de batería

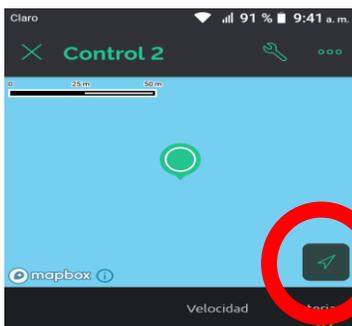
Este widget nos permite mostrar el porcentaje de la batería.



9- Botón para localizar mi dispositivo

(Si está activado mi ubicación del celular)

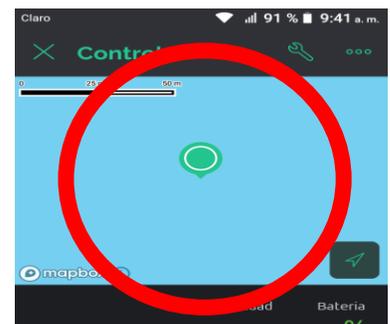
El widget de mapa nos permite localizar nuestro dispositivo, para ver, necesitamos activar la ubicación del celular. Luego presionar el botón



10- Mapa para visualizar los dispositivos

(Si incluye modulo GPS)

El widget de mapa nos permite localizar nuestro dispositivo, y hardware del robot, el módulo GPS deberá enviar las coordenadas al servidor blynk para visualizar en el mapa.



4.3. Manual y guía rápida de instalación cámara inalámbrica

Especificaciones técnicas:

Marca: Lloyd's

Modelo: LC-1220

Cámara inalámbrica: 5 Vcc, 1 A

Adaptador de ca/cc

Entrada: 100-240V ~ 50/60 Hz - 0,2 A

Salida: 5 Vcc 1 A

Favor leer la presente guía de instalación antes de operar la cámara

1. Diagrama



2. Funciones principales:

- Almacenamiento en la nube.
- Grabación en alta resolución (1080P).
- Cámara con rotación y zoom (PTZ).
- Lente: Ángulo 85".
- Compatible con iOS y Android.
- Dos vías de audio.
- Monitoreo remoto.
- Internet inalámbrico.
- Soporta tarjeta de memoria (hasta 64GB). - Visión nocturna.
- Transmisión en vivo.
- Detección de movimiento.

3. Conexión:

Descarga de la aplicación.

Busque y descargue la APP "Lloyd'sSmart" desde Play Store o Apple Store, o bien, escanee el siguiente código QR para obtenerla.



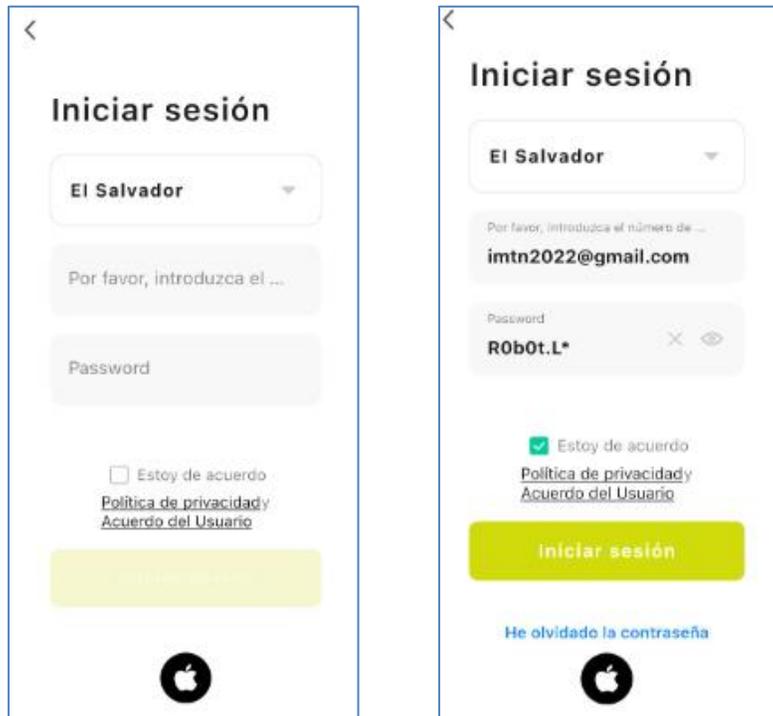
4. Registro:

Al ingresar a la App aparecerá la siguiente pantalla y seleccionaremos la opción "Iniciar sesión con cuenta ya existente".

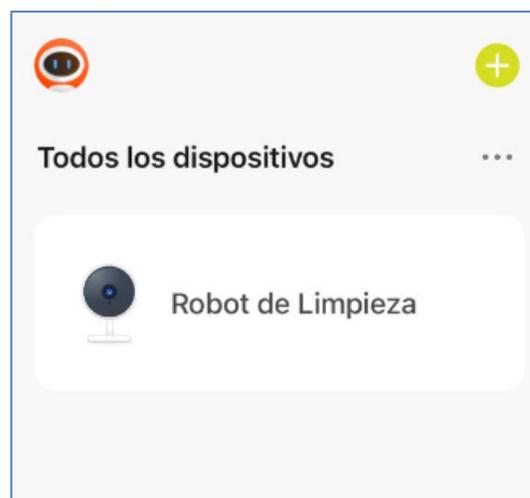


Posteriormente aparecerá la opción para ingresar nuestros datos de inicio de sesión, ingresaremos nuestro correo y contraseña.
Correo: [imtn2022@gmail.com/](mailto:imtn2022@gmail.com)
Contraseña: R0b0t.L*

Luego hacemos clic en el recuadro en el que indicamos que estamos de acuerdo con las políticas de privacidad, posteriormente hacemos clic en "iniciar sesión"



Finalmente aparecerá la siguiente interfaz y seleccionaremos la opción "Robot de limpieza"



5. Configuración:

Notas importantes:

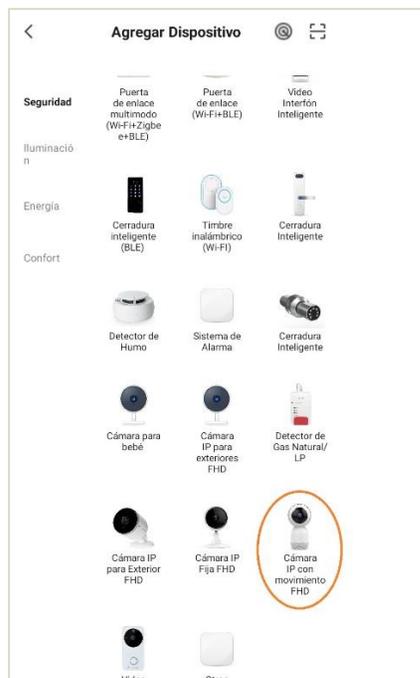
Este dispositivo solo soporta frecuencias de banda de 2.4GHz. NO puede conectarse a redes de 5GHz. Cuando se encuentre configurando el dispositivo con su red WiFi, mantenga de preferencia su teléfono celular y el dispositivo cercanos al router para acelerar la velocidad de configuración y conexión.

Para iniciar:

- Ingrese la contraseña establecida para iniciar sesión.
- El dispositivo sólo puede agregarse a una cuenta de la APP, si ya ha sido vinculado a una cuenta no podrá ser agregado a otra cuenta.
- El dispositivo será desvinculado en cuanto usted lo elimine a través de la aplicación. Se reiniciará automáticamente y emitirá un sonido "Beep" cada 30 segundos, lo cual indicará que se encuentra en modo de configuración.

6. Proceso:

Paso 1. Abra la aplicación "Lloyd'sSmart" y haga clic en "+" en la esquina superior derecha o en "Añadir un dispositivo". Después seleccione el tipo de dispositivo, en este caso es una "Cámara IP con movimiento FHD".



Paso 2. Asegúrese de que el dispositivo está encendido en modo de configuración y haga clic en "siguiente paso" para continuar.



Paso 3. Elija red WiFi a la que desee añadir la cámara e introduzca la contraseña de la red. Haga clic en "Confirmar". Coloque el código QR que le aparecerá en la pantalla de su teléfono frente al lente de la cámara para su lectura.



Paso 4. Haga clic cuando haya escuchado el tono de aviso y espere a que termine la configuración de la cámara.



Paso 5. Cuando muestre la conexión al 100% el dispositivo ha sido añadido. Usted podrá cambiar el nombre del dispositivo, compartirlo y comenzar a disfrutar de sus funciones local o remotamente.

7. Funciones:

- Inclinación y paneo

Usted puede mover automáticamente la cámara al deslizar el dedo en la pantalla de visualización de su teléfono celular.

- Comunicación de dos vías

La cámara cuenta con micrófono y altavoz para lograr una intercomunicación completa con la misma desde su teléfono móvil.

- Detección de movimiento

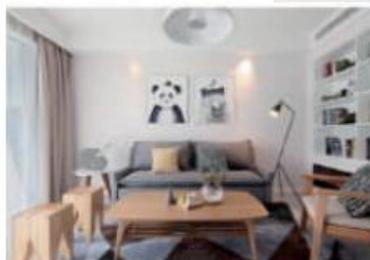
Su sensor integrado detecta el movimiento y envía notificaciones a la aplicación.

- Grabación

Soporta 24h de grabación continua con alta capacidad de almacenamiento en tarjeta de memoria y reproducción desde la APP.

- Día y noche

Incluso en completa oscuridad, su tecnología de visión nocturna permite observar claramente en cualquier momento.



Día



Noche

8. Advertencias de uso:

Intente mantener la cámara fuera de espacios metálicos o bloqueados por muebles u homos de microondas que obstruyan las señales de sonido, video y datos. Procure instalar en los sitios más cercanos a la cobertura WiFi de su hogar u oficina.

9. Preguntas frecuentes:

P: En el proceso de conexión de red, la barra no llega al 100%, ¿es un error?

R: La cámara únicamente puede conectarse a redes de 2.4GHz, por favor asegúrese que esa es la frecuencia de red e intente de nuevo.

P: ¿Añadir dispositivos repetidamente puede causarles fallas?

R: Si en algún momento un dispositivo presenta problemas de conexión, es recomendable reiniciarlo e intentar añadirlo de nuevo.

P: El dispositivo no muestra imágenes correctamente.

R: Observe si la red funciona correctamente. Coloque la cámara cerca del router, si no mejora reinicie el dispositivo y añádalo nuevamente.

P: ¿Cómo puedo cambiar el dispositivo a otro router?

R: Primero reinicie el dispositivo y a continuación bórralo desde la aplicación. Posteriormente configúrelo nuevamente desde la aplicación.

P: ¿Por qué no obtengo notificaciones en mi teléfono?

R: Asegúrese que la aplicación está activa en su teléfono y que la función de recordatorios está funcionando.

P: ¿Por qué el dispositivo no reconoce la tarjeta de memoria?

R: Si ocurrió un corto en el suministro de energía, se recomienda reinsertar la tarjeta SD. Observe que la tarjeta tenga formato FAT32.

4.4. Manual de uso cargador de batería

En primer lugar, es importante mencionar que la batería utilizada es de polímero de litio (LiPo), así mismo es necesario saber cómo interpretar la información de una batería de polímero de litio. Encontraremos datos como el voltaje (11.1V), el amperaje (2200mAh), la clasificación C (50C) y el número de celdas que se identificará con la letra S (3s).



En la siguiente imagen se presentarán los componentes que posee el cargador de batería.



1. Cable de alimentación
2. Cargador de batería
3. Conector de celdas
4. Cable de alimentación del cargador a la batería.

Especificaciones técnicas

- Marca: Protek RC
- Modelo: PTK-8522
- Voltaje de entrada de AC: 100V-240V
- Voltaje de entrada de CC: 11V-18V
- Potencia de carga: 100W
- Corriente de carga: 10A
- Potencia de descarga: 5W
- Corriente de descarga: 24 -
- Admite: batería LiPo, LiFe, LiMV, NiCD, NiMH y PB

ADVERTENCIA:

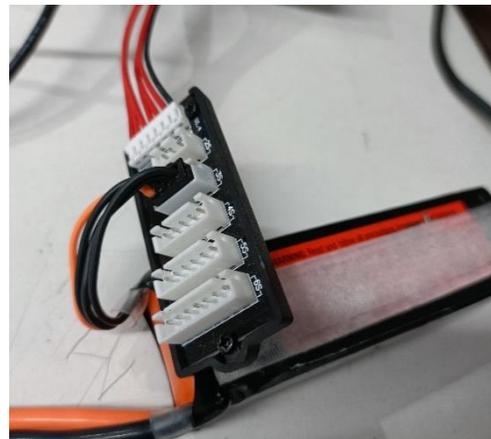
- No cargue baterías no recargables.
- No apto para pilas desechables.
- Mantener alejado del calor, materiales inflamables, polvo, humedad y vibraciones.
- Nunca deje el cargador desatendido mientras está en uso.
- Lea atentamente el manual del usuario antes de utilizar este cargador.
- El incumplimiento de las instrucciones puede causar daños o lesiones.

Para cargar nuestra batería será necesario realizar los siguientes pasos:

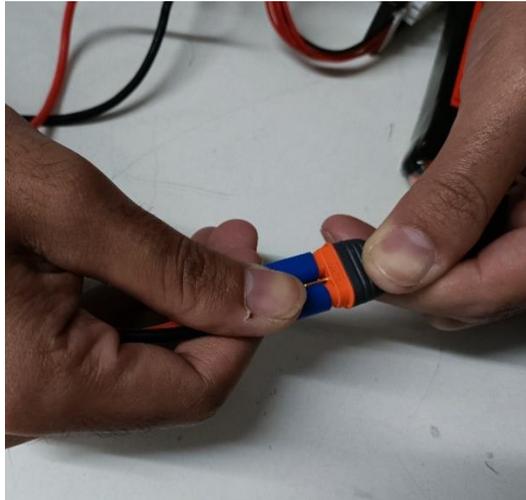
- Paso 1. Conectaremos los cables de alimentación a nuestro cargador, como se muestra a continuación:



- Paso 2: Adaptar al conector correspondiente a 3 celdas (3S)



- Paso 3. Conectar terminales de la batería con terminales de salida del cargador y luego conectar el cargador al tomacorriente (120V)



- Paso 4. Al encender la pantalla, aparecerán los siguientes mensajes, presionar el botón de "Start" para continuar.



- Paso 5. Seleccionar el tipo de batería a usar (LiPo) y luego presionar start para continuar.



- Paso 6. En primer lugar, es necesario seleccionar el tipo de carga (con los botones +, -), en nuestro caso seleccionar "BAL-CHG" para equilibrar la carga en las celdas. Luego presionando el botón de

Start una vez y posteriormente presionando los botones (+. -) configurar las características de la batería a cargar (n° de celdas, capacidad de carga y amperaje) y finalmente presionar el botón start cuando se encuentre lista la configuración de cada uno de los parámetros.



- Paso 7. Al tener lista la configuración de nuestra batería, mantener presionado el botón de start por 5 segundos y aparecerá la siguiente pantalla que nos muestra el voltaje actual de la batería, la capacidad de carga, el tiempo de carga en minutos y segundos, y la corriente faltante para llegar a sus características ideales.



- Paso 8. Al finalizar la carga, emitirá un sonido que nos indica que la batería ya se encuentra lista y en ese momento presionar el botón Stop



- Paso 9. Desconectar el cargador del tomacorriente de alimentación.

4.5. Manual de mantenimiento

EMPRESA:	MASDE		
LISTA DE INSPECCIÓN SISTEMA MECANICO			
Sección:		Instalación: Campus ITCA- FEPADE sede Santa Tecla	
Nivel: 1		Máquina: Robot móvil de limpieza	
Fabrica: ITCA_FEPADE		No Inv.:	
M	A	INTERVALO	ACTIVIDAD
x		Diariamente	Verificar que no haya piezas sueltas
X		diariamente	Verificar el estado de la bandeja de recolección
x		Una vez por semana	Verificar lubricación en partes móviles
x		Diario	Verificar que no haya obstrucción en las llantas
x		Una vez por semana	Verificar que no haya partes sin pintura de la carcasa
	x	diario	Verificar el estado de los visores acrílicos que no estén rotos o agrietados
	x	Una vez por semana	Verificar el estado de las escobillas
x		Una vez por semana	Verificar el funcionamiento de la aspiradora
	x	Una vez por mes	Verificar estado de la batería
	X	Una vez por mes	Verificar el ajuste y fijación de la cámara
	x	Una vez por mes	Verificar que la banda de oruga no presente daño
X		Una vez por semana	Verificar bisagras de la compuerta en carcasa de acrílico
x		Una vez por semana	Verificar ajuste de tornillos
	x	Una vez por semana	verificar movimiento manual para eje Y
	x	Una vez por semana	verificar movimiento manual para eje x
x		Una vez por sema	Verificar la sujeción de la pantalla
	x	Una vez por sema	Verificar sujeción y estado de motores
Operario:		Cambios:	
Hoja: De:		Fecha:	
Responsable:			

EMPRESA: ITCA-FEPADE	HOJA DE CHEQUEO SISTEMA MECANICO		
ESTADO	SI	NO	
Se encuentran pernos sin ajustar			
Se encuentran lubricadas las piezas móviles			
Existe alguna pieza suelta			
Existe algún visor de acrílico roto o agrietado			
La compuerta de acrílico abre y cierra sin problema			
Al presionar las teclas para mover el eje Y este se mueve normalmente			
Al presionar las teclas para mover el eje X este se mueve normalmente			
Se encuentran en buen estado los ejes de los servomotores			
Se encuentra en buen estado la banda de oruga			
Existe obstrucción en las ruedas			
Se ha dañado la pintura de la carcasa			
Hay obstrucciones en el visor acrílico			
Hay abolladuras en la carcasa metálica			

EMPRESA		MASDE			CODIGO		
LISTA DE SERVICIO: SISTEMA MECANICO							
Sección		Instalación:			Edificio:		
:		ITCA FEPAD E Sede			Fabrica: ITCA-		
Nivel:1		Central			FEPAD E		
No posición: N/A		Máquina: Robot de limpieza móvil					
M	A	INTERVALO	ACTIVIDAD	No VECES	Herramienta a usar	Cantidad	Artículos usados
	X	Semanal	Lubricación de piezas móviles	3	Aceitera		Aceite lubricante DW 40
	X	Semanal	Ajuste de pernos		Destornilladores		
X		Semanal	Limpieza de bandejas recolectoras	2	Brocha		Limpiador liquido
	X	Mensual	Limpieza de carcasa metálica	1			Paños de limpieza
	X	Mensual	Limpieza de carcasa acrílica	1	Contact cleaner		Paños de limpieza
Operario:		Cambios:			Hoja:		De:
				Fecha:		Responsable:	

EMPRESA:		MASDE	
LISTA DE INSPECCIÓN SISTEMA ELECTRICO/ELECTRONICO			
Sección:		Instalación: Campus	
ITCA- FEPADE sede Santa Tecla			
Nivel: 1		Máquina: Robot móvil de limpieza	
Fabrica: ITCA_FEPADE			
No Inv.:			
M	A	INTERVALO	ACTIVIDAD
	x	Diariamente	Verificar el estado cables de alimentación si están bien conectados y no estén dañados
x		Una vez por semana	Verificar los cables de comunicación entre placas, que no estén dañados o cortados
x		Diario	Verificar el funcionamiento de la pantalla
x		Diario	Verificar el funcionamiento de la cámara
	x	Diario	Verificar el funcionamiento de indicadores LED
	x	Una vez por semana	Verificar el estado de los sensores ultrasónicos
x		Una vez por semana	Verificar el funcionamiento correcto de la placa de Arduino
	x	Una vez por mes	Verificar estado de la batería
	x	Una vez por semana	Verificar el suministro de voltaje que sea el correcto
	x	Una vez por semana	Verificar la corriente consumida de la batería
Operario:		Cambios:	
Hoja:		De:	
Responsable:		Fecha:	

EMPRESA: ITCA-FEPADE	HOJA DE CHEQUEO SISTEMA ELECTRICO/ELECTRÓNICO		
ESTADO	SI	NO	
Existe algún cable cortado o desconectado			
Se encuentra polvo o suciedad en alguna placa			
Existe algún pin de placa roto o faltante			
Los botones funcionan correctamente			
Las placas electrónicas se encuentran en buen estado, sin quemaduras o grietas			
Existe algún led quemado			
Existen capacitores quemados			
Se encuentran en buen estado los servomotores			
La cámara graba y transmite con normalidad			
El suministro de voltaje es adecuado			
El amperaje consumido es el adecuado			
Los sensores ultrasónicos funcional con normalidad			
La pantalla se muestra con normalidad			
Hay retrasos en la comunicación con el robot			

EMPRESA		MASDE			CODIGO		
		LISTA DE SERVICIO: SISTEMA ELECTRICO/ELECTRONICO					
Sección		Instalación:		Edificio:			
:		ITCA FEPADE		Fabrica: ITCA-			
Nivel:1		Sede Central		FEPADE			
No posición: N/A		Máquina: Robot de limpieza móvil		No de Inventario:			
M	A	INTERVAL O	ACTIVIDAD	No VECES	Herramien ta que usar	Canti dad	Artículos usados
X			Limpieza de sensores ultrasónicos	6	Brocha		Alcohol isopropíli co
X			Limpieza de placas	5	Brocha		Alcohol isopropíli co
X			Reconexión de sensores		Borneros, tester		
X			Cambio de sensores		Borneros, tester		
Operario:		Cambios:		Hoja:		De:	
		Fecha:		Responsable:			

CAPITULO V

ALCANCES Y LIMITACIONES

5.1. Alcances

- Ensamblar un robot que será capaz de desplazarse en los pasillos exteriores y dentro de los edificios del campus de ITCA-FEPADE.
- Diseñar e implementar un control remoto o aplicación para que el robot pueda ser operado por una persona.
- El robot contará con sensores de proximidad para evitar colisiones con muros, gradas o grietas en el suelo.
- El robot contará con sistema de alarma para notificar colisiones graves u otros estados de emergencia.
- Agregar al robot la función de trituración.
- Implementar un sistema de registros de operaciones del robot.
- Realizar un manual de mantenimiento para el proyecto.
- Realizar un manual de ensamblaje para el proyecto.
-

5.2. Limitaciones

- El robot se verá limitado a solo desplazarse en la planta baja de los edificios y pasillos exteriores del campus, debido a que no pueden subir escaleras por sí mismo.
- La mayoría de los componentes para el ensamblaje son de importación.
- La batería tiene un tiempo de duración limitado de aproximadamente 60 minutos.
- El robot no podrá desplazarse sobre superficies con piedras considerablemente grandes en comparación a las dimensiones del robot.
- El robot no podrá desplazarse sobre grama (ejem. Cancha del campus ITCA-FEPADE).
- En caso de sufrir una colisión fuerte un usuario deberá intervenir para alinear al robot.

CAPITULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1. Recomendaciones

- Evitar su uso en clima lluvioso y húmedo
- No someter a terreno hostil; como hierba alta, lodo, grava, corrientes de agua, etc.
- Se recomienda lubricar las partes móviles regularmente
- Se recomienda hacer ciclo de carga completa a la batería para mantenerla en buen estado y evitar que llegue a descargarse por completo.
- No exponer los elementos electrónicos a líquidos u ambientes húmedos.
- Mantener despejada y limpia la zona que cubre la cámara
- Verificar el estado de los sensores de forma regular para asegurar el funcionamiento completo de forma óptima.
- Dar mantenimiento planeado regularmente y aplicar mantenimiento correctivo de ser necesario.
- Cambiar las escobillas de limpieza cuando el desgaste sea considerable.
- Vaciar regularmente la bandeja de almacenamiento de basura.
- Cambiar el filtro de la aspiradora con regularidad.
- Vaciar regularmente el filtro de la aspiradora.
- No agregar elementos pesados sobre la estructura, ya que aumenta la carga que debe desplazar, por lo tanto, se fuerzan más sus motores y batería.
- Evitar colisiones, ya que pueden causar daño permanente a la estructura metálica o a los elementos electrónicos del robot.
- No ajustar demasiado la banda de oruga para evitar su desgaste.

6.2. Conclusiones

- A través de nuestro proyecto se logró investigar, aprender y aplicar conocimientos sobre sistemas innovadores de comunicación a través de red inalámbrica, demostrando sus ventajas sobre sistemas más conocidos, convencionales y de mayor uso.
- El robot utiliza un sistema de control comunicación desarrollado en la plataforma Blynk con las ventajas que esta ofrece.
- Se construyó un robot móvil, capaz de desplazarse por el campus de ITCA-FEPADE sede central y de limpiar los corredores y pasillos de forma eficiente.
- Se implementó exitosamente la plataforma Arduino para controlar los circuitos internos, actuadores, motores y sensores.
- La aplicación para el control del robot es atractiva visualmente y fácil de usar.
- Se aplicaron conocimientos adquiridos en toda la carrera de ingeniería en mecatrónica.

BIBLIOGRAFÍA.

- Tolocka, E. (7 de mayo de 2020). *Diseñando aplicaciones IoT con Blynk*. profe Tolocka. <https://www.profetolocka.com.ar/2020/05/07/disenando-aplicaciones-iot-con-blynk/>
- Blynk.io. (s.f.). *documentation*. <https://docs.blynk.io/en/>
- Diosdado, R. (5 de mayo de 2016). *Display's para Arduino*. Zona Marker. <https://www.zonamaker.com/arduino/modulos-sensores-y-shields/tipos-de-lcd-para-arduino>
- Ephesos Software. (s.f.). *6 pantallas para generar datos desde tu Arduino*. <https://es.ephesossoftware.com/articles/diy/6-displays-to-output-data-from-your-arduino.html>
- Crew Noticias Móviles. (s.f.). *7 pantallas para la salida de datos de su Arduino*. <https://noticiasmoviles.com/7-pantallas-para-la-salida-de-datos-de-su-arduino/>
- Naylampmechatronics. (s.f.). *Tutorial LCD con I2C, controla un LCD con solo dos pines*. https://naylampmechatronics.com/blog/35_tutorial-lcd-con-i2c-controla-un-lcd-con-solo-dos-pines.html
- Anonimo . (2022). *Invention of Robotic Vacuum Cleaners*. 12/04/2022, de vacuumcleanerhistory.com Sitio web: vacuumcleanerhistory.com
- Arduino S.r.l . (2021). *Arduino Mega 2560 Rev3*. 12/04/2022, de Arduino S.r.l Sitio web: <https://store-usa.arduino.cc/products/arduino-mega-2560-rev3?selectedStore=us>
- Arduino S.r.l. (2022). *Arduino Software*. 12/04/2022, de Arduino s.r.l Sitio web: <https://www.arduino.cc/en/software>
- Helloauto. (2021). *Torque*. 12/04/2022, de Hello Insurance group, Compañía de seguros. S.A Sitio web: <https://www.helloauto.com/glosario/torque#:~:text=El%20torque%20se%20considera%20una,se%20mide%20en%20newtons%2Fmetro.>

- hiniuns; Moises Diez. (2019). Cálculo de torque para elección de un motor . 12/04/2022, de Foros de Electrónica Sitio web: <https://www.forosdeelectronica.com/threads/calculo-de-torque-para-eleccion-de-un-motor.163071/>

Fuentes de información (shields-shield L293D):

- Juan Carlos macho. (2022). Motor shield V1 y motores paso a paso. 11/04/2022, de PROMETEC.NET Sitio web: <https://www.prometec.net/motor-shield-stepper/>
- Fouad:Robotocist. (2019). Arduino servo motor control with motor. 11/04/2022, de Project HUB Sitio web: https://create.arduino.cc/projecthub/Fouad_Robotocist/arduino-servo-motor-control-with-motor-driver-shield-l293d-f60a7c
- <https://sensoricx.com/electronica-de-control/circuito-integrado-l293d-que-es-para-que-sirve-y-como-funciona/>
- Jorge lobo. (2019). EchidnaShield el Arduino shield para la educación . 11/04/2022, de Programafacil.com Sitio web: <https://programafacil.com/blog/arduino-blog/echidnashield-arduino-shield-para-educacion/>
- Equipo Techmake. (2020). Empezando con Arduino - 5C: Motor shield. 10/04/2022, de Tech make Sitio web: <https://techmake.com/blogs/tutoriales/empezando-con-arduino-5c-motor-shield>
- Java T Point . (2021). Arduino Shields . 10/04/2022, de Java T Point Sitio web: <https://www.javatpoint.com/arduino-shields>

Fuentes de información (sensores- sensor HC-RS04):

- Geek Factory team. (2014). HC-SR04 sensor de distancia ultrasónico . 10/04/2022, de Geek Factory Sitio web: <https://www.geekfactory.mx/tienda/sensores/hc-sr04-sensor-de-distancia-ultrasonico/>
- Geek Factory team. (2014). HC-SR04 con Arduino sensor de distancia ultrasónico. 10/04/2022, de Geek Factory Sitio web: <https://www.geekfactory.mx/tutoriales-arduino/hc-sr04-con-arduino-sensor-de-distancia-ultrasonico/#>

Fuentes de información (MPU 6050):

- Naylamp Mechatronics. (2016). Tutorial MPU6050, acelerómetro y giroscopio . 12/04/2022, de Naylamp Mechatronics Sitio web: https://naylampmechatronics.com/blog/45_tutorial-mpu6050-acelerometro-y-giroscopio.html

- Vicente García . (2015). Sensor MPU6050. 12/04/2022, de Electrónica practica aplicada Sitio web: <https://www.diarioelectronicohoy.com/blog/sensor-mpu6050>
- Robots Argentina. (2019). Descripcion y funcionamiento del BUS I2C. 12/04/2022, de Robots Argentina Sitio web: <https://robots-argentina.com.ar/didactica/descripcion-y-funcionamiento-del-bus-i2c/>
- Wikipedia, La enciclopedia libre. (2020). Giroscopio de estructura vibrante. 1270472022, de Wikipedia, La enciclopedia libre Sitio web: https://es.wikipedia.org/wiki/Gir%C3%B3scopo_de_estructura_vibrante

ESP32:

- Ravi Teeja. (2021). Introducción a ESP32. 10/04/2022, de Electronics Hub Sitio web: <https://www-electronicshub-org.translate.goog/getting-started-with-esp32/? x tr sl=en& x tr tl=es& x tr hl=es-419& x tr pto=sc>
- Ida Hubschmann. (2019). ESP32 for IoT: A complete guide. 10/04/2022, de Nabto Sitio web: <https://www.nabto.com/guide-to-iot-esp-32/#:~:text=ESP32%20is%20the%20name%20of,%E2%80%9CESP32%E2%80%9D%20by%20the%20manufacturer.>

Eficiencia compañías de internet

- Javier Orellana. (2020). El Salvador: Telefónicas cumplen hasta en 80% calidad de conexión. 11/04/2022, de El Economista Sitio web: <https://www.eleconomista.net/actualidad/El-Salvador-Telefonicas-cumplen-hasta-en-80--calidad-de-conexion-20201123-0005.html>

Redes Wifi:

- Areatecnologia . (----). Tecnología inalámbrica. 07/04/2022, de Areatecnologia Sitio web: <https://www.areatecnologia.com/informatica/tecnologia-inalambrica.html>

Pull up y Pull Down:

- Luis del valle Hernández . (2018). Resistencia pull up y pull down. 07/04/2022, de programafacil.com Sitio web: <https://programafacil.com/blog/arduino-blog/resistencia-pull-up-y-pull-down/>

Baterías de Polímero de Litio:

- Arturo valle. (2020). Todo lo que debería saber sobre la batería de polímero de litio (LiPo). 07/04/2022, de UNIT electronics Sitio web: <https://uelectronics.com/todo-lo-que-deberia-saber-sobre-la-bateria-de-polimero-de-litio-lipo/>

Servo motor MG996R¹

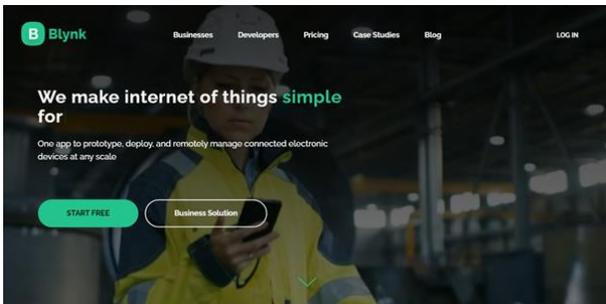
- MG996R Servo Datasheet pdf (2014) - Bearing Servo. Equivalent, Catalog Sitio web: <https://datasheetspdf.com/pdf/942981/ETC/MG996R/1>

ANEXOS

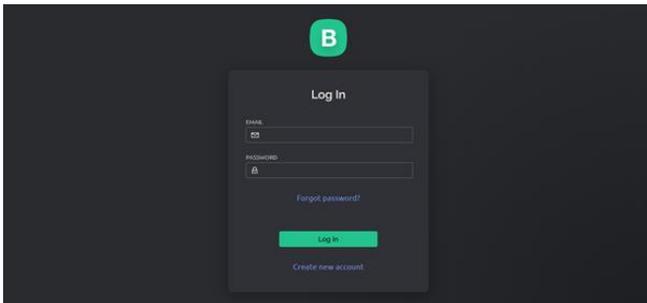
ANEXO 1

- Diseñar proyecto en Blynk io

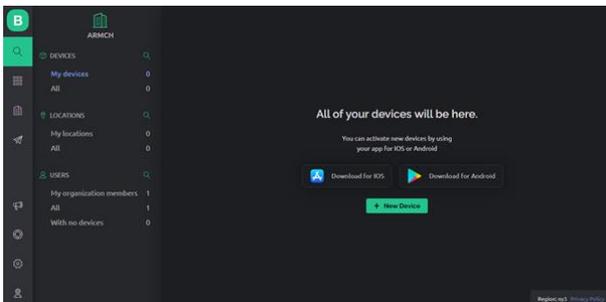
Paso 1: Ingresar a la página oficial de Blynk io.



Paso 2: Crear una cuenta en Blynk io.

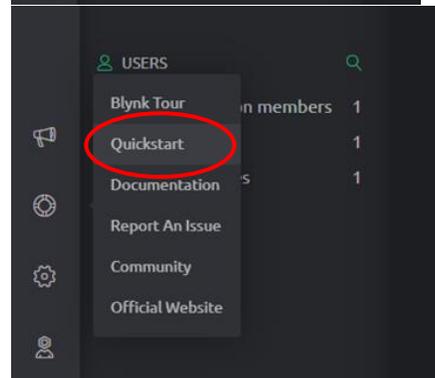
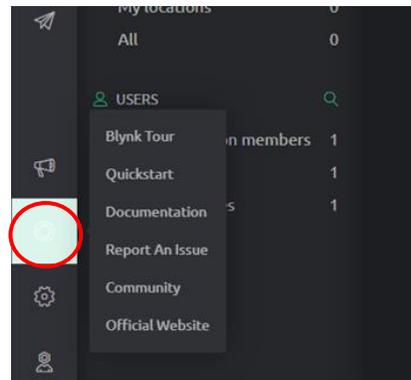


Paso 3: Crear un proyecto.

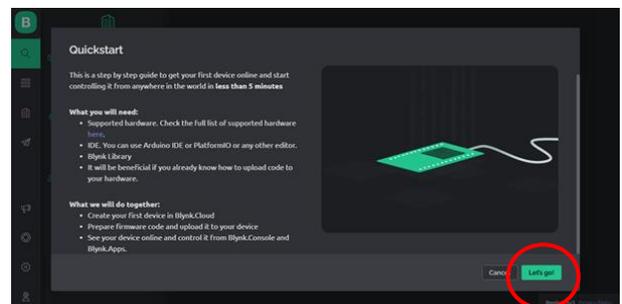


Podemos usar un proyecto predeterminado de Blynk:

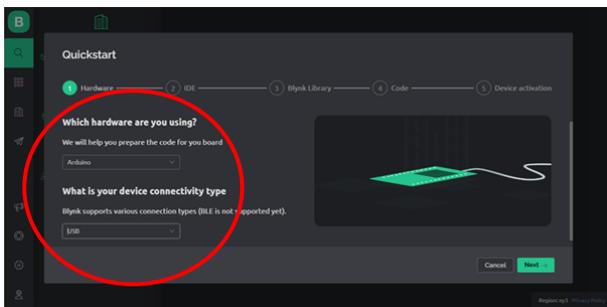
1- Buscamos la herramienta Quickstart.



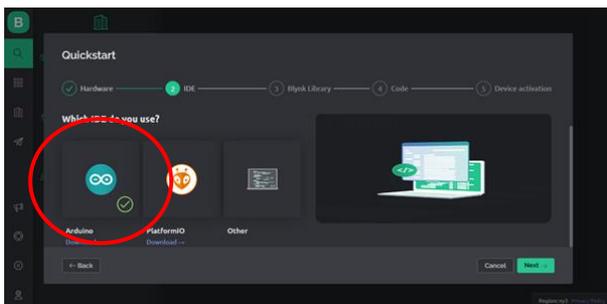
2- Dar click en Let's go!



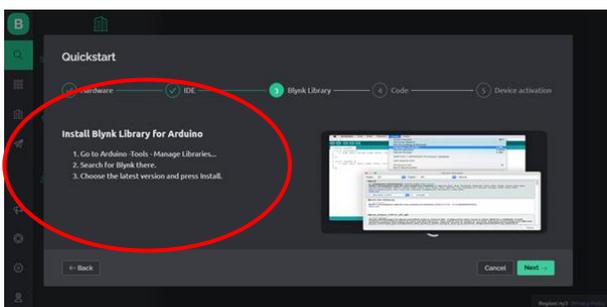
3- Seleccionar Hardware.



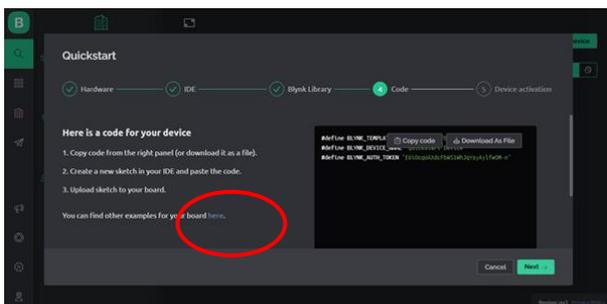
4- Seleccionar IDE.



5- Instalar librería Blynk siguiendo instrucciones de la plataforma.



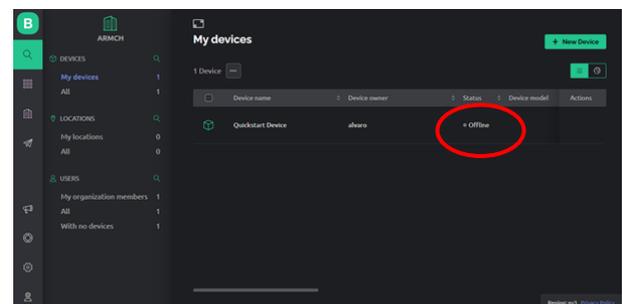
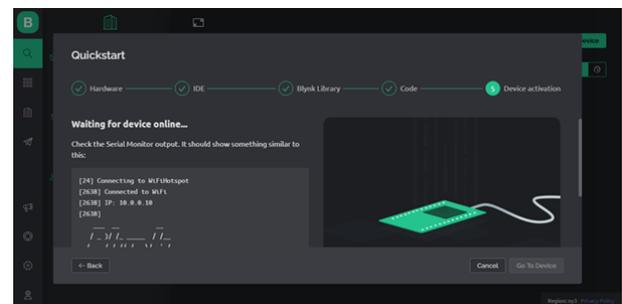
6- Dar click here.



7- Verificar Hardware y copiar código.

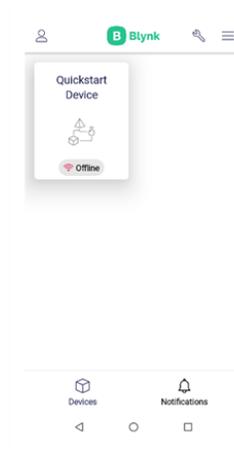


8- Cargar el código al Hardware y verificar que se haya conectado a la plataforma.



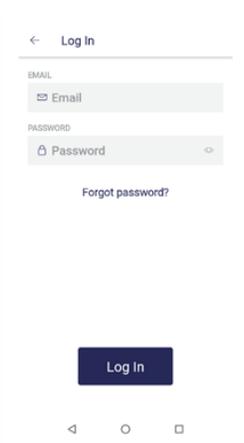
Nota: Dependerá el tipo de conexión del hardware con la plataforma.

Paso 4: Descargar app.



Podemos observar si el hardware está en línea.

Paso 5: Ingresar la cuenta de Blynk.



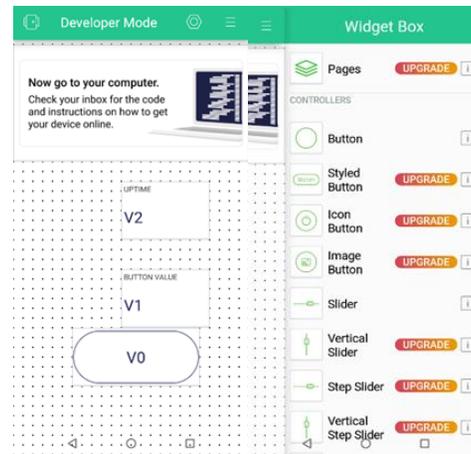
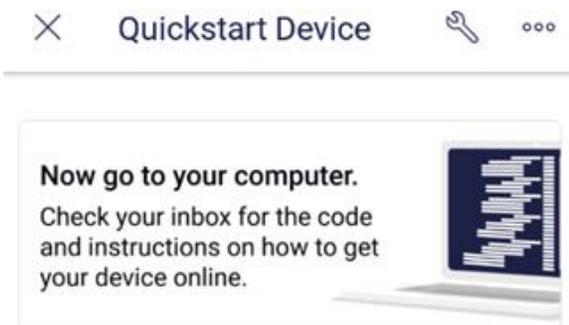
Paso 6: Dar clic Quickstart Device.

Paso 7: Controlar y monitorear.



Este proyecto nos mostrará un mensaje si el dispositivo está conectado, además de poder activar un botón y recibir el estado del botón, también muestra el tiempo en que el dispositivo está en línea.

Para modificar el tablero debemos dar click a la herramienta.



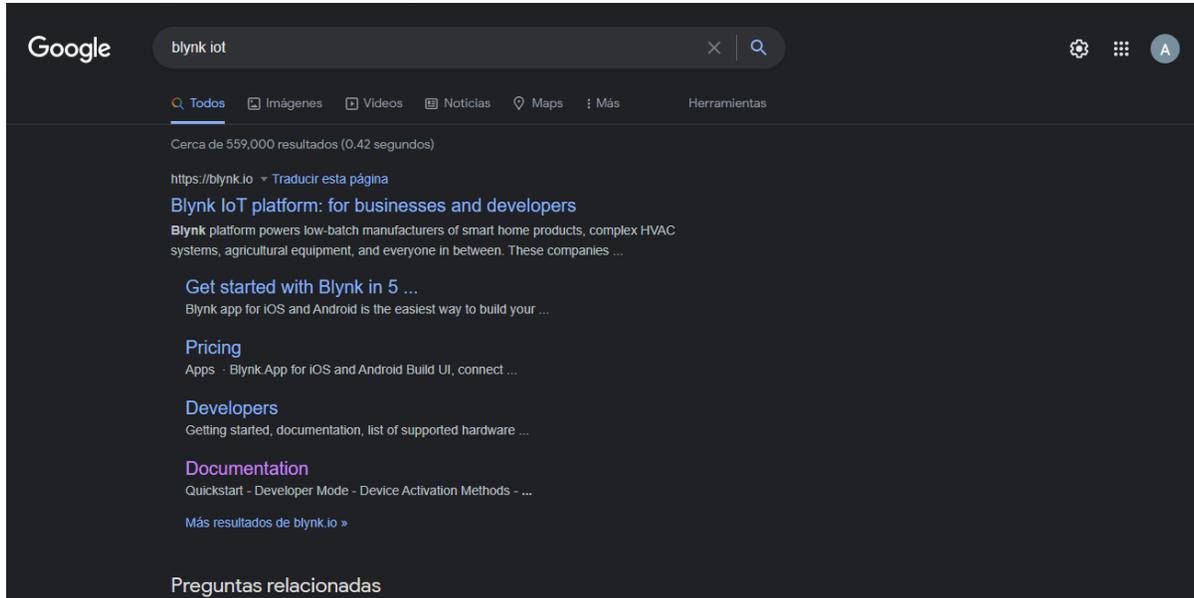
En esta sección podemos modificar el tablero, además agregar, eliminar y mover widgets.

Para encontrar más información lo podemos encontrar documento Blynk.

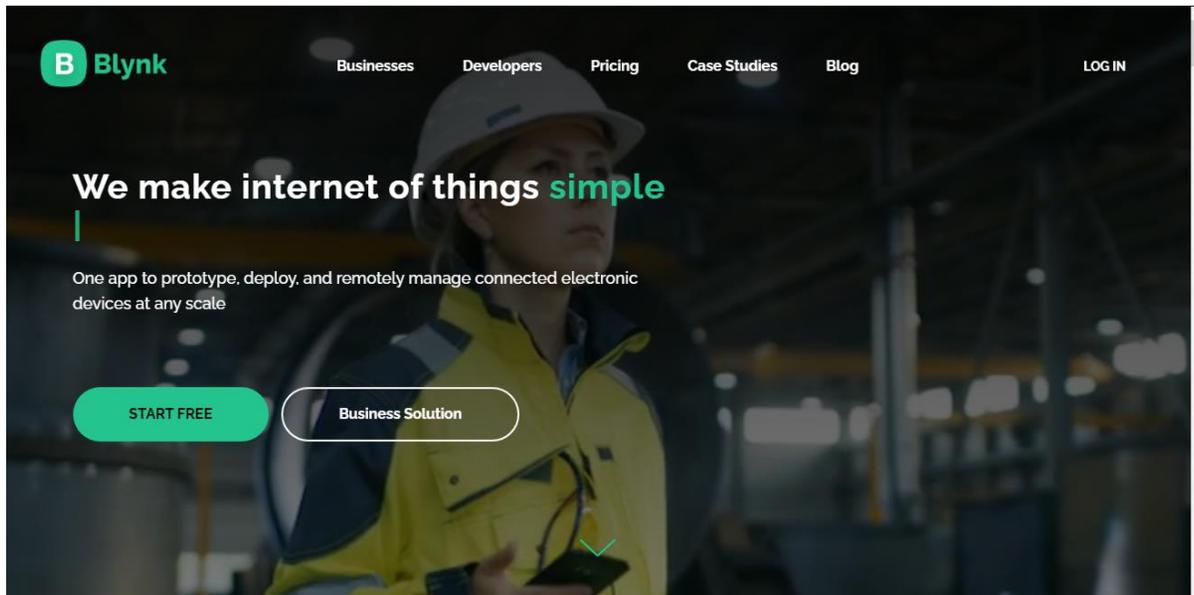
ANEXO 2

Ingresar blynk

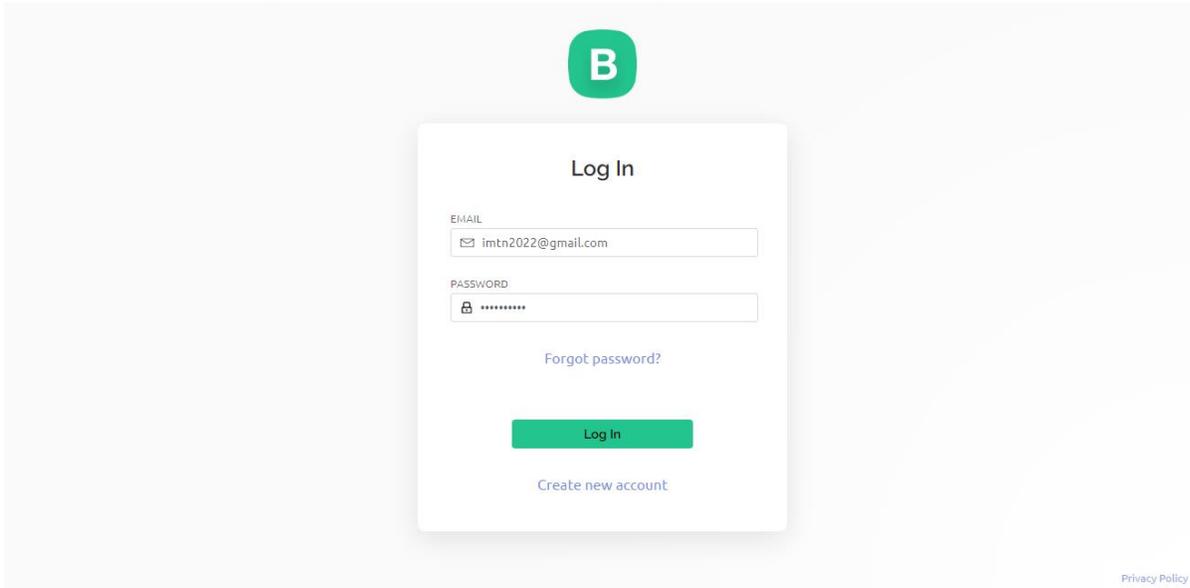
1- Buscar blynk iot en el navegador web



2- Iniciar sesión



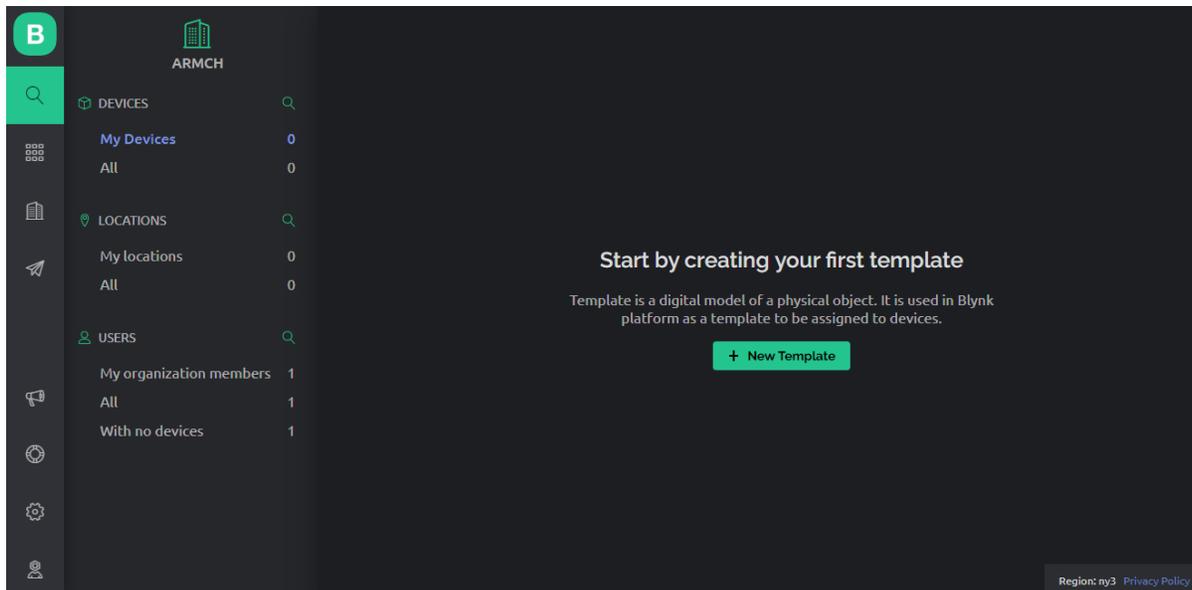
3- Ingresar la cuenta del proyecto para acceder el control del robot de limpieza
Correo: **imtn2022@gmail.com** Contraseña de blynk: **imtn-@2022**



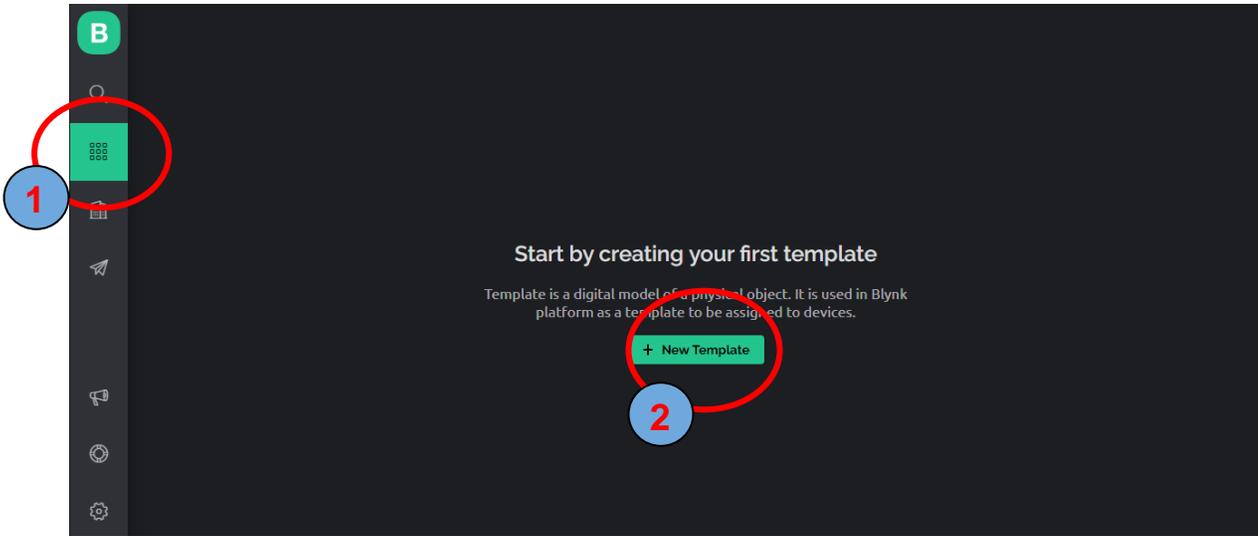
Nota: Esta cuenta es de administrador

Crear un proyecto en plataforma Blynk

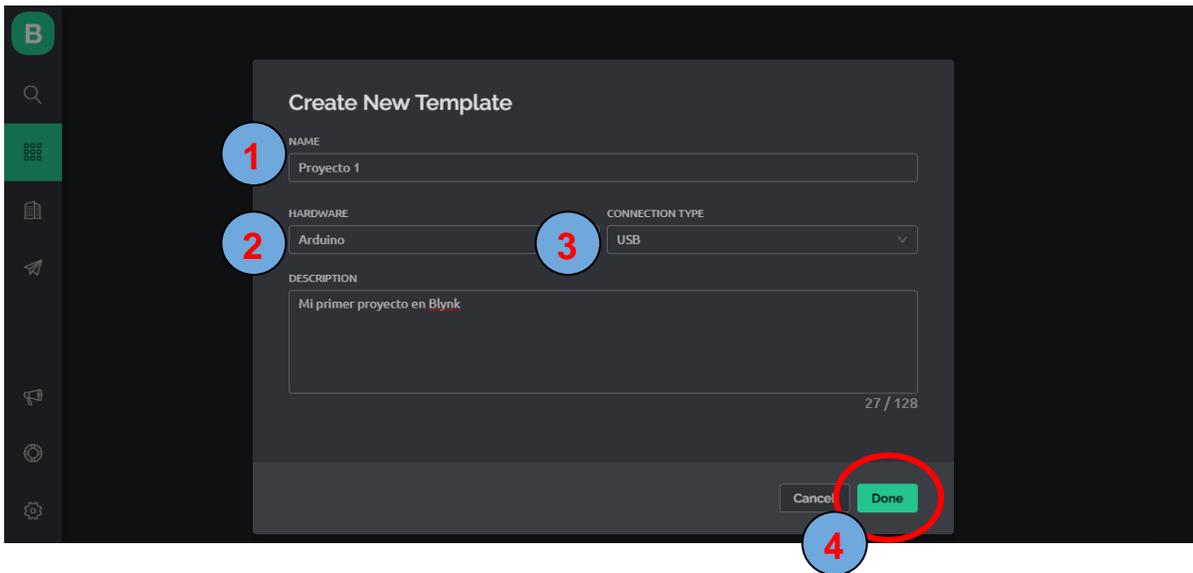
1- Ingresar a la consola de Blynk

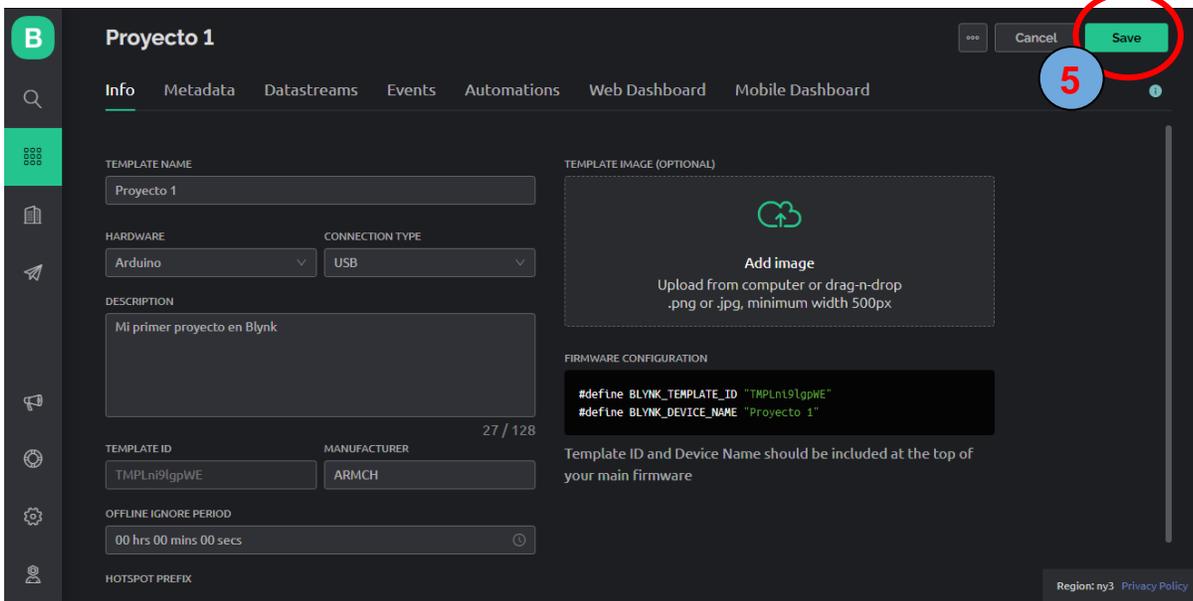


2- Crear un nuevo template

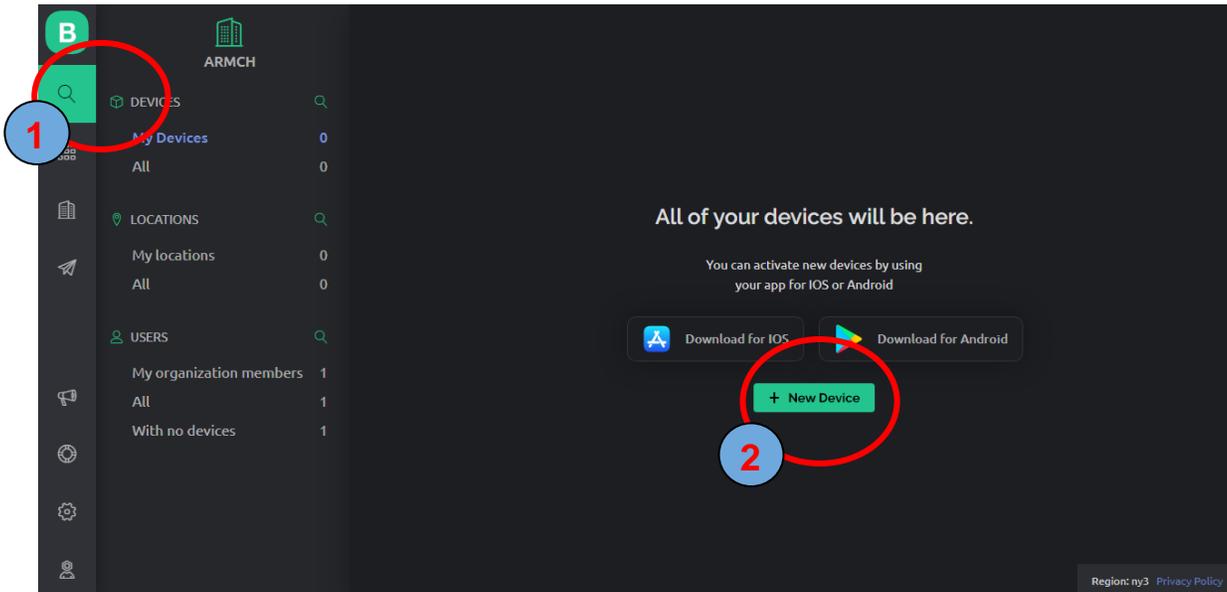


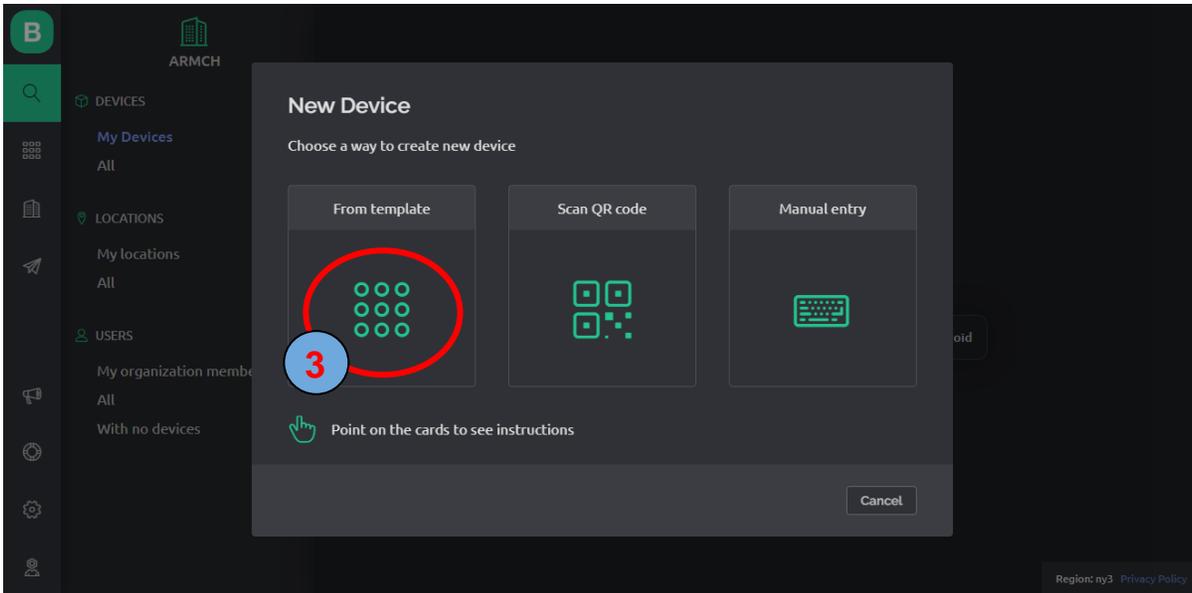
3- Completar los datos como el nombre del proyecto, hardware, tipo de conexión



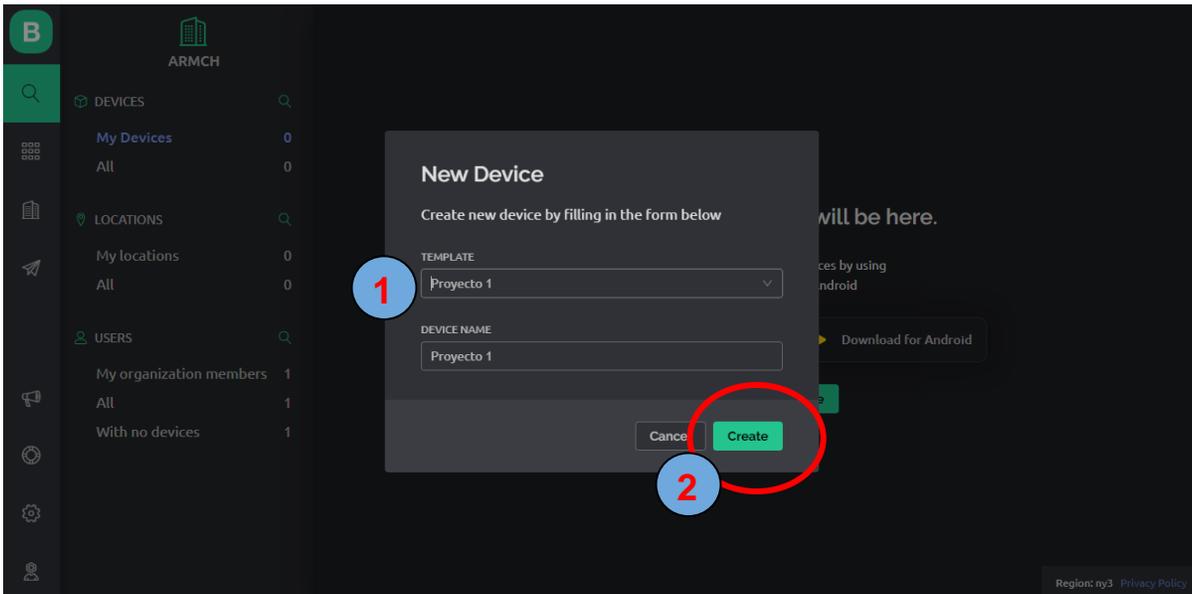


4- Crear un nuevo devices

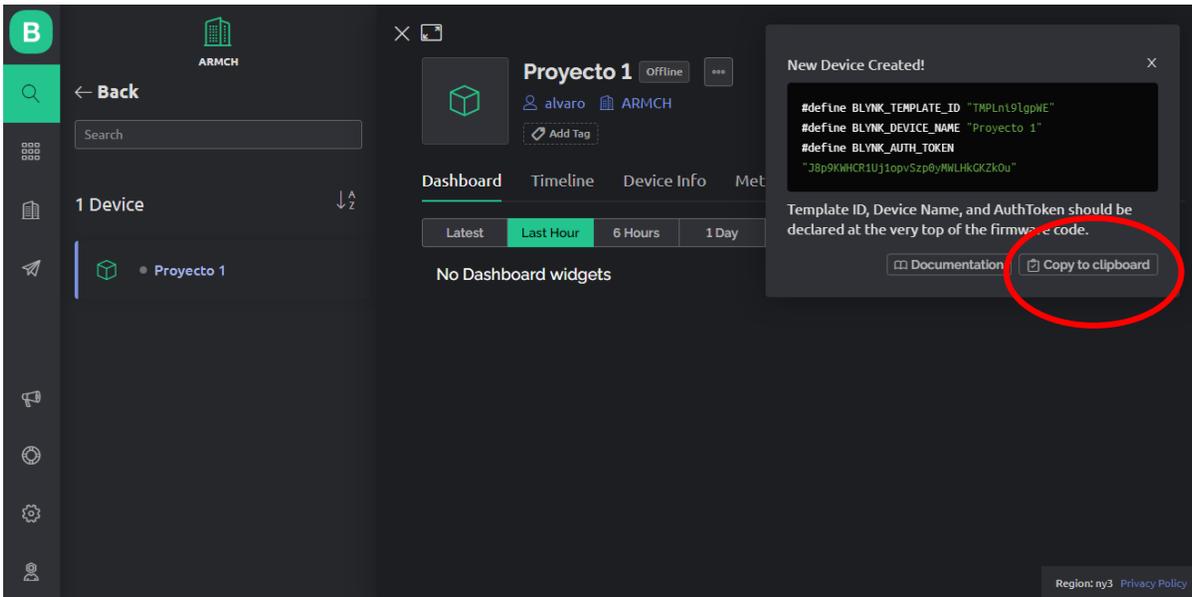




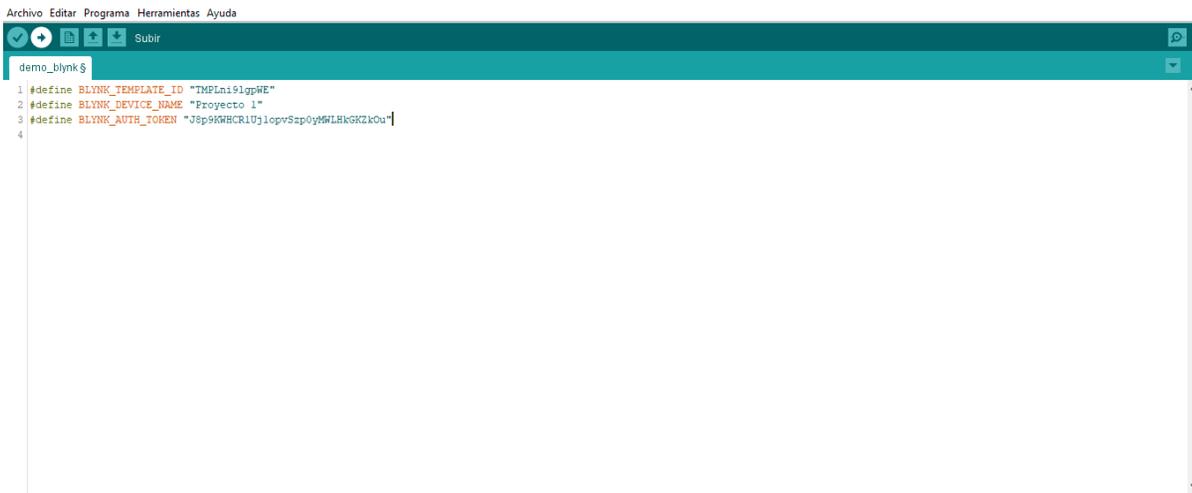
5- Selecciona el template creado



6- Copiar el template ID, devices name, Auth token



7- Pegar el template ID, devices name, Auth token en Arduino IDE



8- Buscar en el correo para copiar un ejemplo

Blynk

21 mar 2022, 16:04 ☆

Hi there, Let's get your first device connected! We hope you already know how to add libraries and upload sketches to your device using Arduino IDE, ...

Blynk <robot@blynk.cloud>

21 mar 2022, 17:31 ☆ ↩️ ⋮

para mí ▾

🌐 inglés ▾ > español ▾ Traducir mensaje

Desactivar para: inglés x

Hi there,

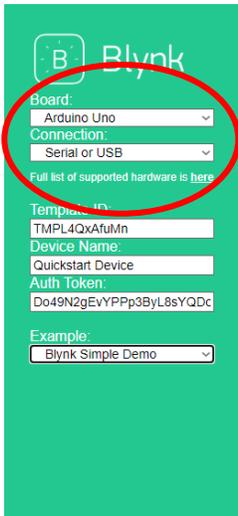
Let's get your first device connected!

We hope you already know how to add libraries and upload sketches to your device using Arduino IDE, PlatformIO, or any other one.

Now follow these steps to prepare your device:

1. Install Blynk library for [Arduino IDE](#) or for [PlatformIO](#)
2. We've prepared a ready-to-use code for your device. Copy it [here](#)
3. If you're using a WiFi device, find the following code lines in the provided sketch and replace it with your WiFi credentials:

9- Selecciona el hardware y el tipo de conexión



```

/*****
 * This is a simple demo of sending and receiving some data.
 * Be sure to check out other examples!
 *****/

// Template ID, Device Name and Auth Token are provided by the Blynk.Cloud
// See the Device Info tab, or Template settings
#define BLYNK_TEMPLATE_ID "TMPL4Qx4FuMn"
#define BLYNK_DEVICE_NAME "Quickstart Device"
#define BLYNK_AUTH_TOKEN "Do49N2gEvYPPp3ByL8sYQDgDnrqQ1CK"

// Comment this out to disable prints and save space
#define BLYNK_PRINT SwSerial

#include <SoftwareSerial.h>
SoftwareSerial SwSerial(10, 11); // RX, TX

#include <BlynkSimpleStream.h>

char auth[] = BLYNK_AUTH_TOKEN;

BlynkTimer timer;

// This function is called every time the Virtual Pin 0 state changes
BLYNK_WRITE(V0)
{
  // Set incoming value from pin V0 to a variable
  int value = param.asInt();

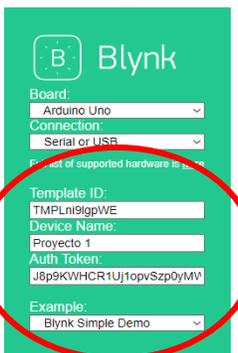
  // Update state
  Blynk.virtualWrite(V1, value);
}

```

Please give us a Github star! ☆ Star copy example

WARNING! Some sketches may contain errors.

10- Pegar el template ID, devices name, Auth token



```

/*****
 * This is a simple demo of sending and receiving some data.
 * Be sure to check out other examples!
 *****/

// Template ID, Device Name and Auth Token are provided by the Blynk.Cloud
// See the Device Info tab, or Template settings
#define BLYNK_TEMPLATE_ID "TMPLn191gpWE"
#define BLYNK_DEVICE_NAME "Proyecto 1"
#define BLYNK_AUTH_TOKEN "J8p9KWHCR1Uj10pvS2p0yMV"

// Comment this out to disable prints and save space
#define BLYNK_PRINT SwSerial

#include <SoftwareSerial.h>
SoftwareSerial SwSerial(10, 11); // RX, TX

#include <BlynkSimpleStream.h>

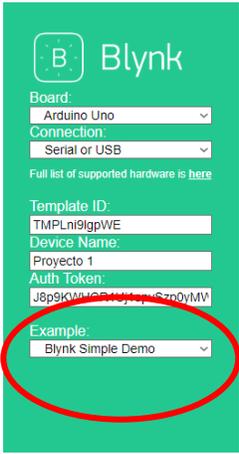
char auth[] = BLYNK_AUTH_TOKEN;

BlynkTimer timer;

```

Please give us a Github star! ☆ Star copy example

11- Seleccionar un ejemplo



The screenshot shows the Blynk web interface. On the left, there is a green sidebar with the Blynk logo and various configuration options: Board (Arduino Uno), Connection (Serial or USB), Template ID (TMPLn9l9ppWE), Device Name (Proyecto 1), and Auth Token (J8p9KWHCR1Uj10pvSzp0yMV). The 'Example' dropdown menu is highlighted with a red circle, and 'Blynk Simple Demo' is selected.

```
/*****  
This is a simple demo of sending and receiving some data.  
Be sure to check out other examples!  
*****/  
  
// Template ID, Device Name and Auth Token are provided by the Blynk.Cloud  
// See the Device Info tab, or Template settings  
#define BLYNK_TEMPLATE_ID "TMPLn9l9ppWE"  
#define BLYNK_DEVICE_NAME "Proyecto 1"  
#define BLYNK_AUTH_TOKEN "J8p9KWHCR1Uj10pvSzp0yMVLHKGKZkOu"  
  
// Comment this out to disable prints and save space  
#define BLYNK_PRINT SwSerial  
  
#include <SoftwareSerial.h>  
SoftwareSerial SwSerial(10, 11); // RX, TX  
  
#include <BlynkSimpleStream.h>  
  
char auth[] = BLYNK_AUTH_TOKEN;  
  
BlynkTimer timer;  
  
// This function is called every time the Virtual Pin 0 state changes  
BLYNK_WRITE(V0)  
{  
  // Set incoming value from pin V0 to a variable  
  int value = param.asInt();  
  
  // Update state  
  Blynk.virtualWrite(V1, value);  
}  
  
// This function is called every time the device is connected to the Blynk.Cloud  
BLYNK_CONNECTED()  
{  
  // Phanna Moh link Button message to "Congratulations!"  
}
```

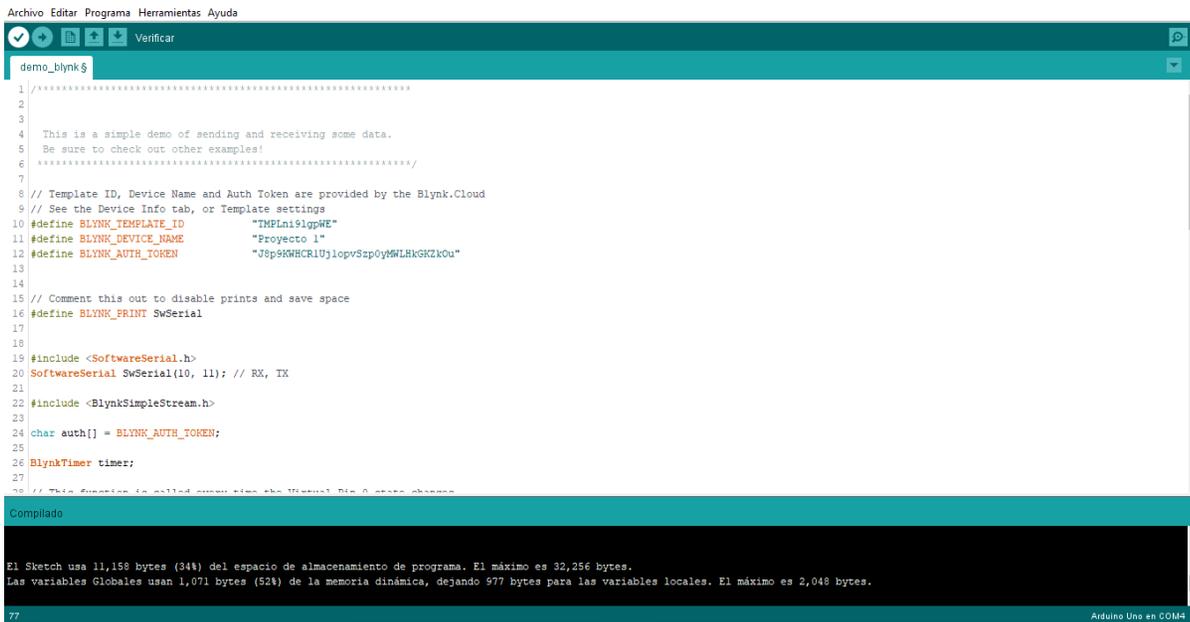
12- Copiar el código del ejemplo



The screenshot shows the Blynk web interface with the same configuration as in step 11. The 'Example' dropdown menu is still set to 'Blynk Simple Demo'. On the right side of the code editor, the 'copy example' button is circled in red.

```
/*****  
This is a simple demo of sending and receiving some data.  
Be sure to check out other examples!  
*****/  
  
// Template ID, Device Name and Auth Token are provided by the Blynk.Cloud  
// See the Device Info tab, or Template settings  
#define BLYNK_TEMPLATE_ID "TMPLn9l9ppWE"  
#define BLYNK_DEVICE_NAME "Proyecto 1"  
#define BLYNK_AUTH_TOKEN "J8p9KWHCR1Uj10pvSzp0yMVLHKGKZkOu"  
  
// Comment this out to disable prints and save space  
#define BLYNK_PRINT SwSerial  
  
#include <SoftwareSerial.h>  
SoftwareSerial SwSerial(10, 11); // RX, TX  
  
#include <BlynkSimpleStream.h>  
  
char auth[] = BLYNK_AUTH_TOKEN;  
  
BlynkTimer timer;  
  
// This function is called every time the Virtual Pin 0 state changes  
BLYNK_WRITE(V0)  
{  
  // Set incoming value from pin V0 to a variable  
  int value = param.asInt();  
  
  // Update state  
  Blynk.virtualWrite(V1, value);  
}  
  
// This function is called every time the device is connected to the Blynk.Cloud  
BLYNK_CONNECTED()  
{  
  // Phanna Moh link Button message to "Congratulations!"  
}
```

13- Pegar el código en Arduino IDE



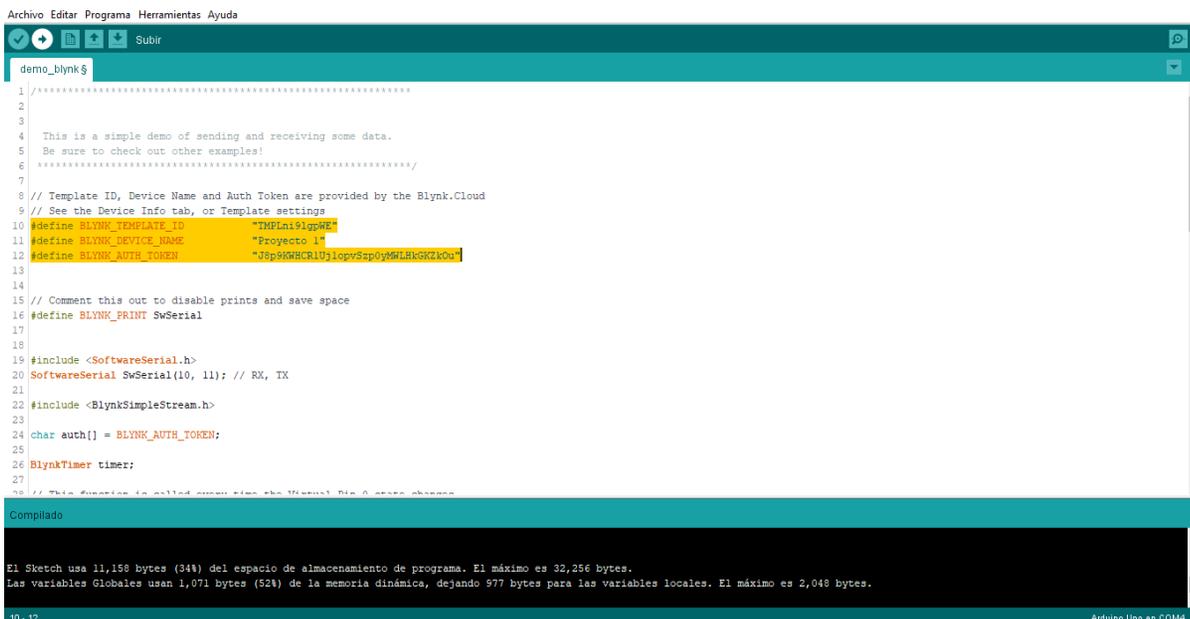
```
1 //*****
2
3
4 This is a simple demo of sending and receiving some data.
5 Be sure to check out other examples!
6 *****/
7
8 // Template ID, Device Name and Auth Token are provided by the Blynk.Cloud
9 // See the Device Info tab, or Template settings
10 #define BLYNK_TEMPLATE_ID "TMPLn191gpWE"
11 #define BLYNK_DEVICE_NAME "Proyecto 1"
12 #define BLYNK_AUTH_TOKEN "Jsp9KwGCR1Uj1opvSzpy9WLEkGK2kOu"
13
14
15 // Comment this out to disable prints and save space
16 #define BLYNK_PRINT SwSerial
17
18
19 #include <SoftwareSerial.h>
20 SoftwareSerial SwSerial(10, 11); // RX, TX
21
22 #include <BlynkSimpleStream.h>
23
24 char auth[] = BLYNK_AUTH_TOKEN;
25
26 BlynkTimer timer;
27
28 // This function is called every time the Micro: Bit is powered on
```

Compilado

El Sketch usa 11,159 bytes (34%) del espacio de almacenamiento de programa. El máximo es 32,256 bytes.
Las variables Globales usan 1,071 bytes (52%) de la memoria dinámica, dejando 977 bytes para las variables locales. El máximo es 2,048 bytes.

77 Arduino Uno en COM4

14- Verifica el template ID, devices name, Auth token del proyecto sean los correcto



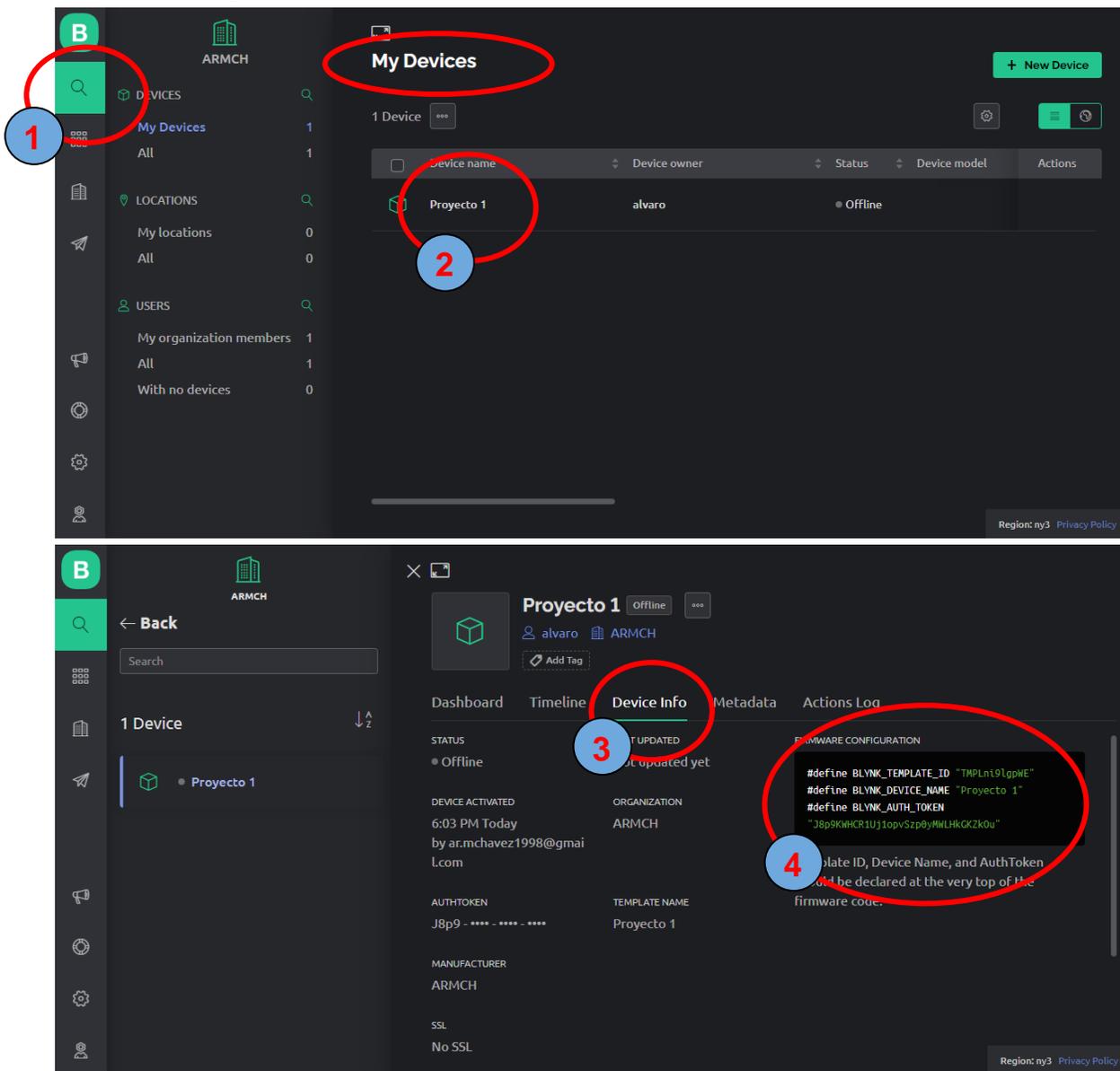
```
1 //*****
2
3
4 This is a simple demo of sending and receiving some data.
5 Be sure to check out other examples!
6 *****/
7
8 // Template ID, Device Name and Auth Token are provided by the Blynk.Cloud
9 // See the Device Info tab, or Template settings
10 #define BLYNK_TEMPLATE_ID "TMPLn191gpWE"
11 #define BLYNK_DEVICE_NAME "Proyecto 1"
12 #define BLYNK_AUTH_TOKEN "Jsp9KwGCR1Uj1opvSzpy9WLEkGK2kOu"
13
14
15 // Comment this out to disable prints and save space
16 #define BLYNK_PRINT SwSerial
17
18
19 #include <SoftwareSerial.h>
20 SoftwareSerial SwSerial(10, 11); // RX, TX
21
22 #include <BlynkSimpleStream.h>
23
24 char auth[] = BLYNK_AUTH_TOKEN;
25
26 BlynkTimer timer;
27
28 // This function is called every time the Micro: Bit is powered on
```

Compilado

El Sketch usa 11,159 bytes (34%) del espacio de almacenamiento de programa. El máximo es 32,256 bytes.
Las variables Globales usan 1,071 bytes (52%) de la memoria dinámica, dejando 977 bytes para las variables locales. El máximo es 2,048 bytes.

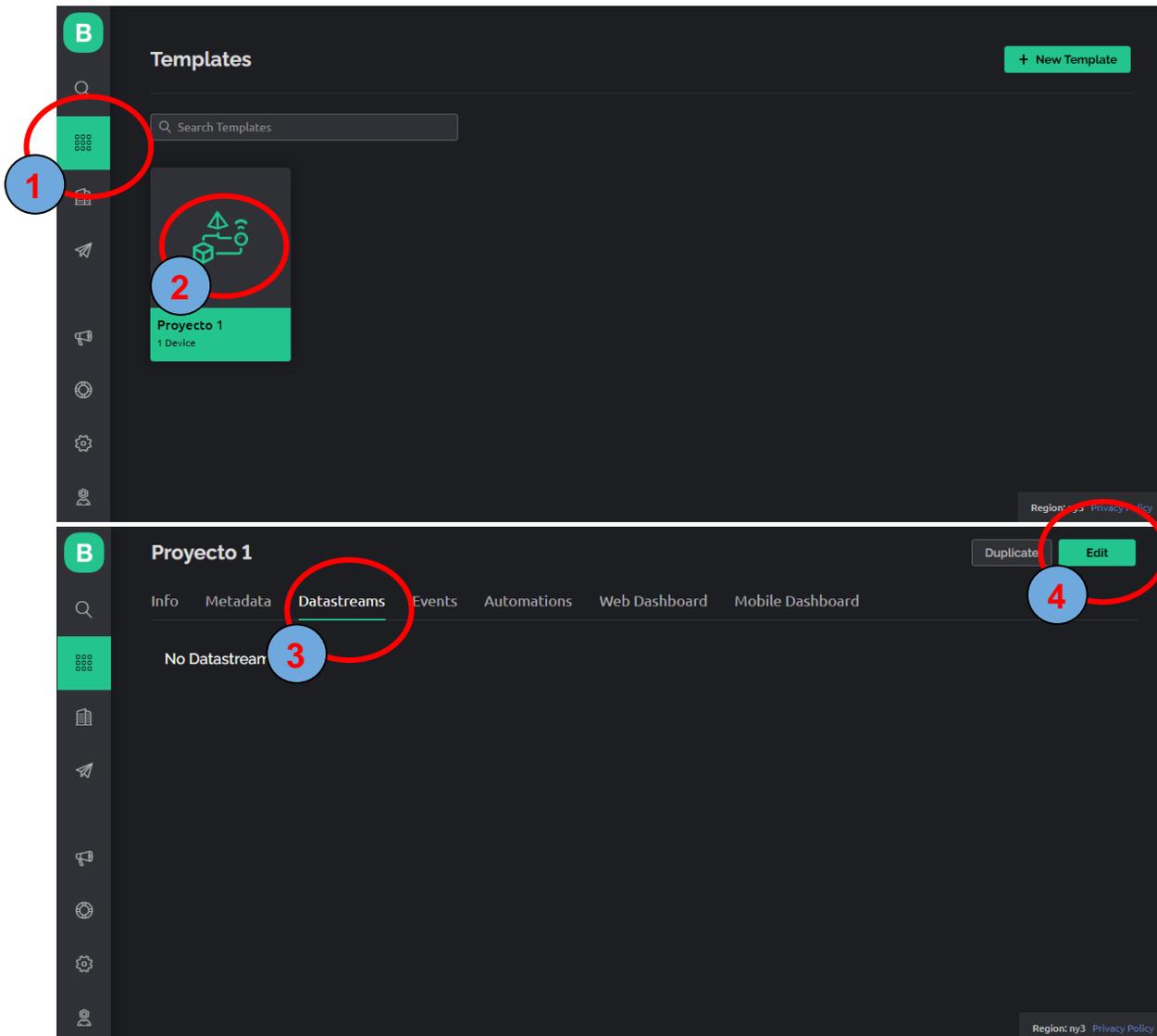
10-12 Arduino Uno en COM4

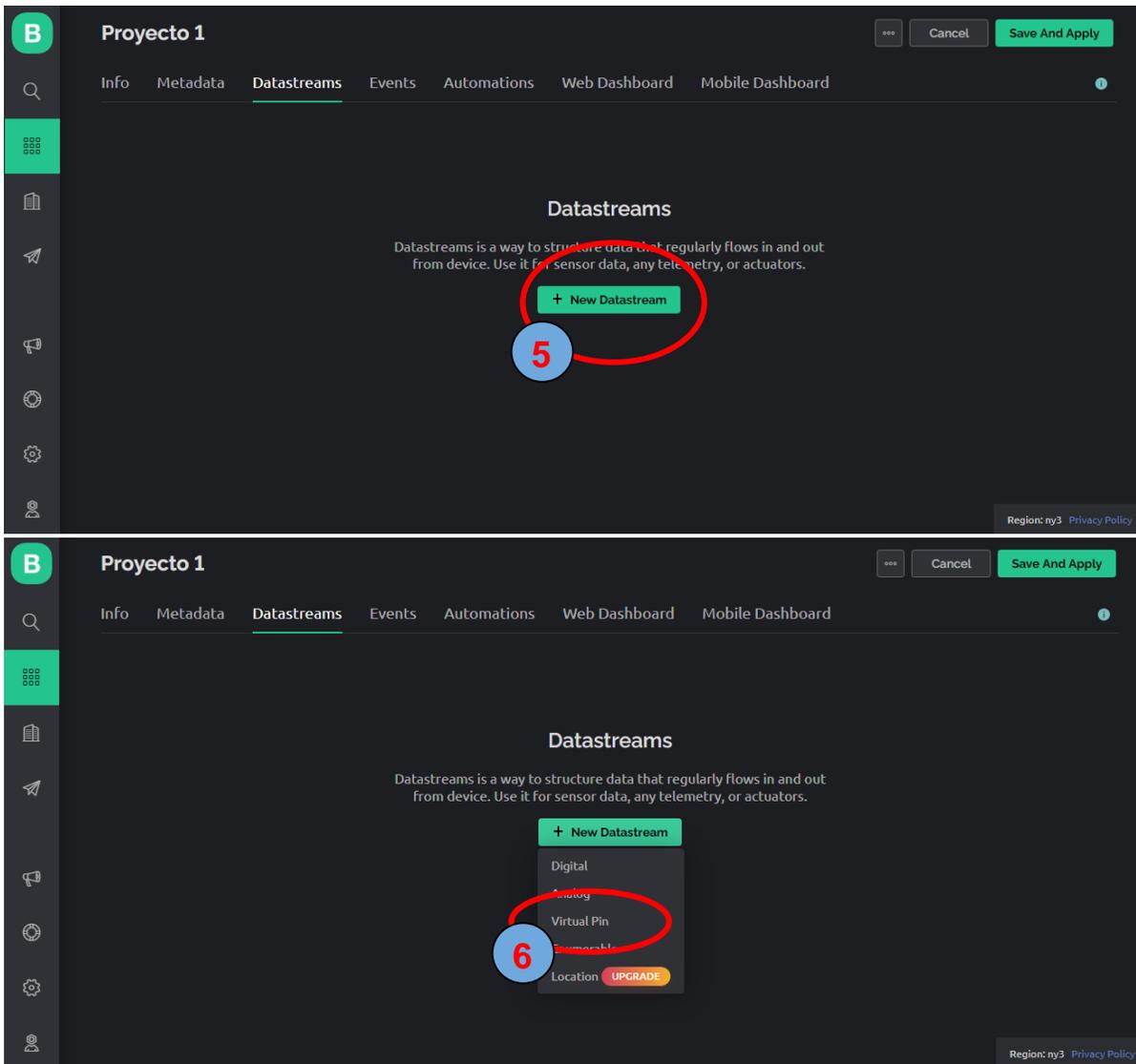
15- Verifica en consola de blynk el template ID, devices name, Auth token en My devices



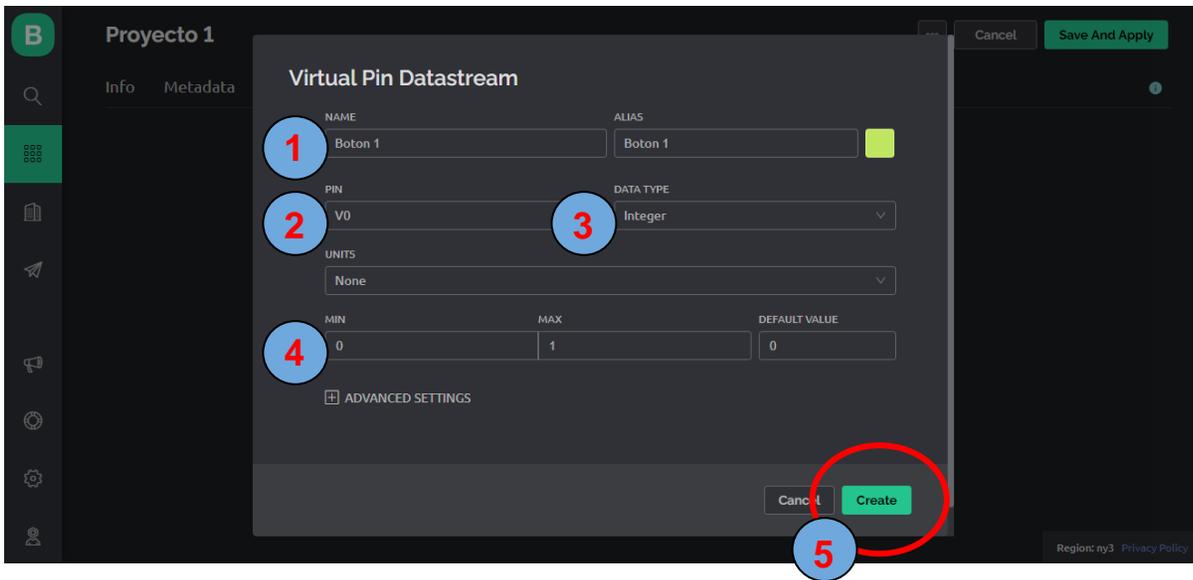
16- Si es diferente copia template ID, devices name, Auth token dando click y pegando en Arduino IDE

17- Crear variables en blynk

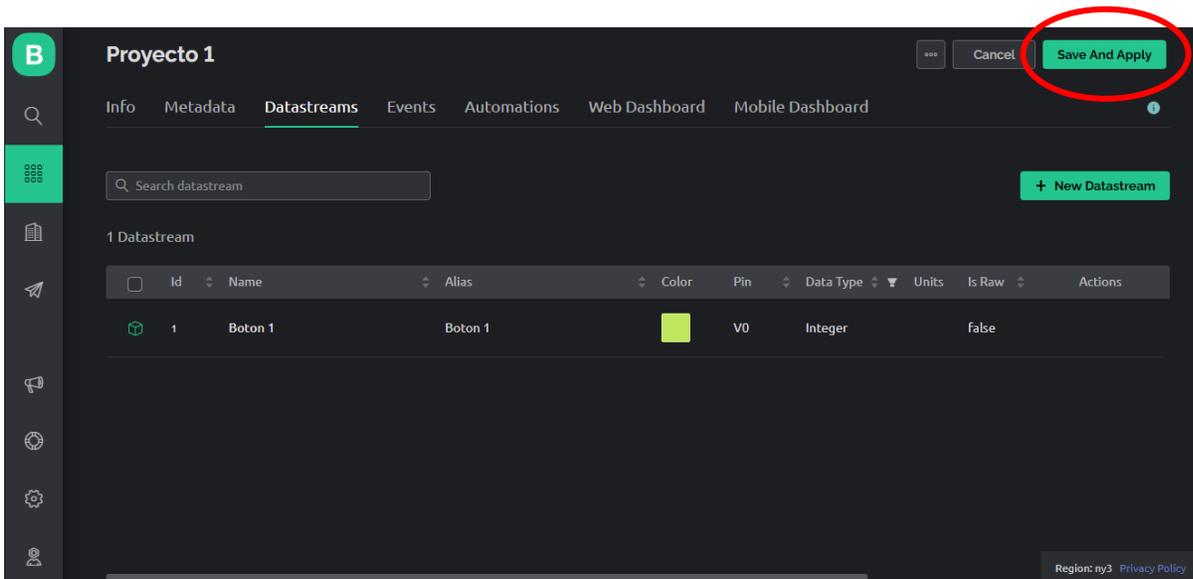




18- Asignar nombre a la variable, seleccionar un pin virtual (máximo 255), tipo de variable, rango de valor

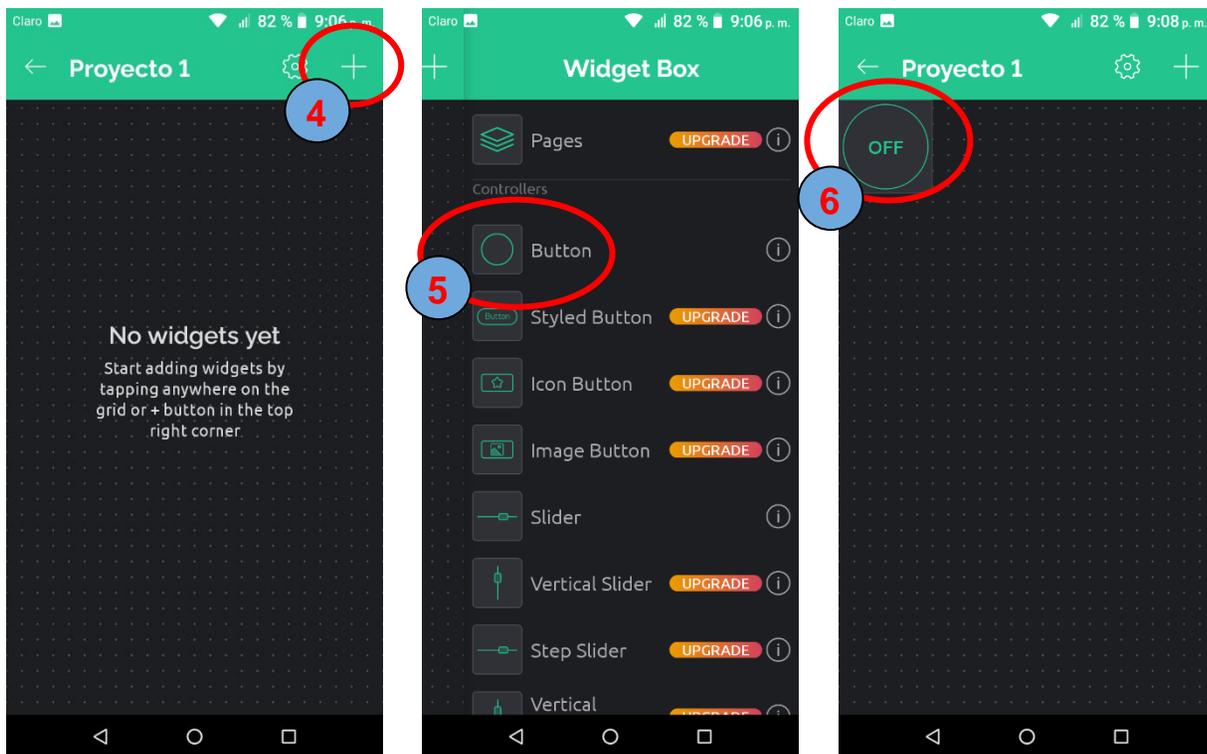
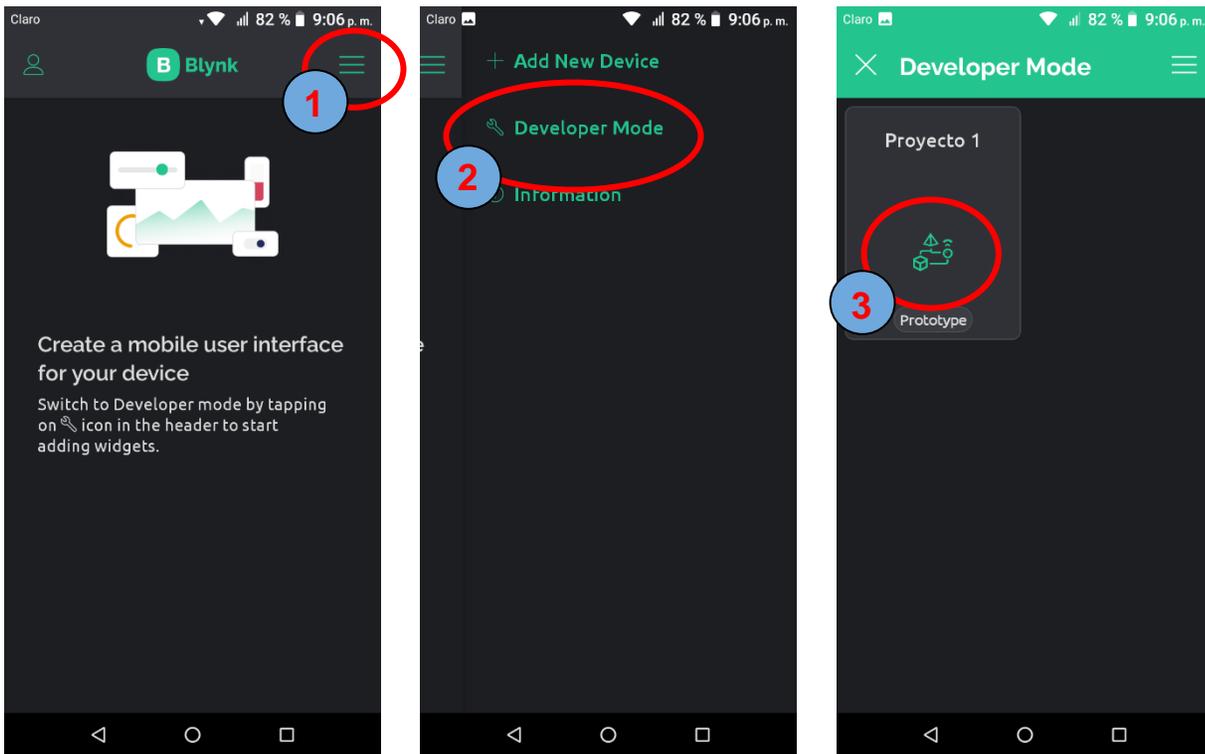


19- Guardar la configuración

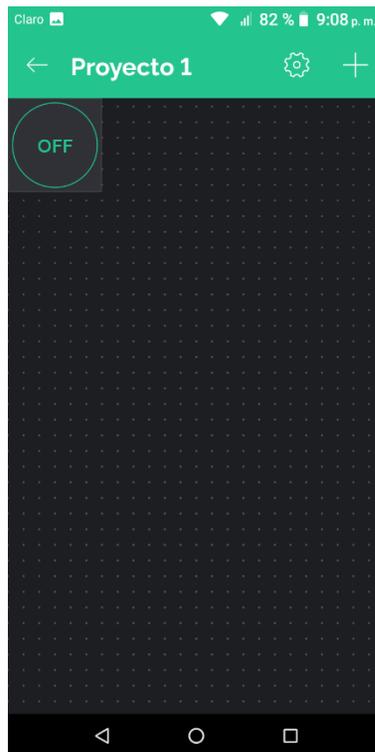
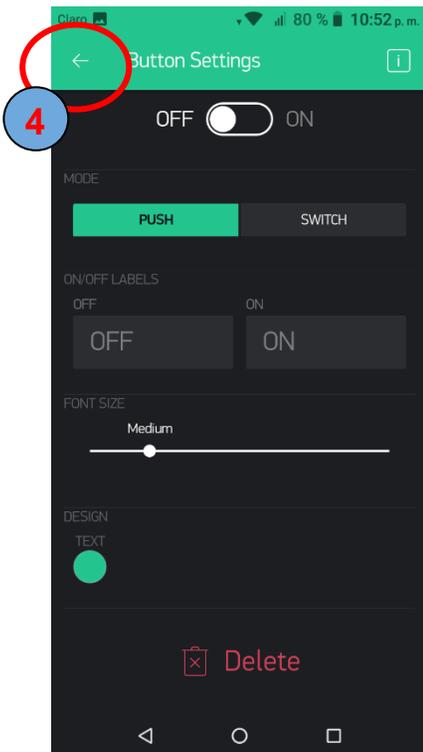
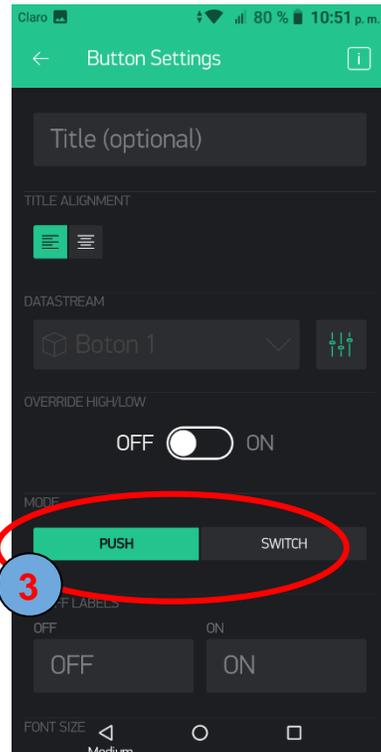
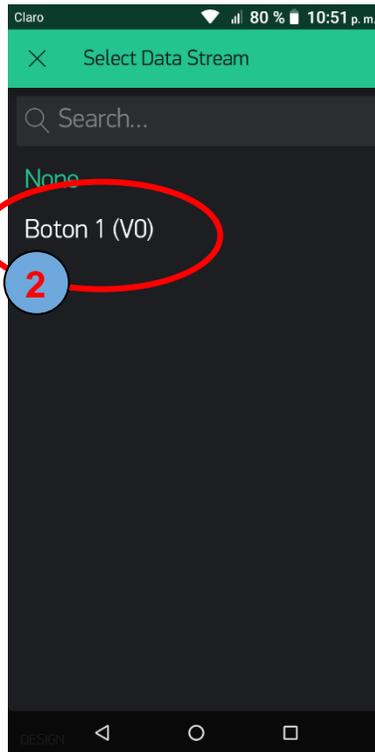
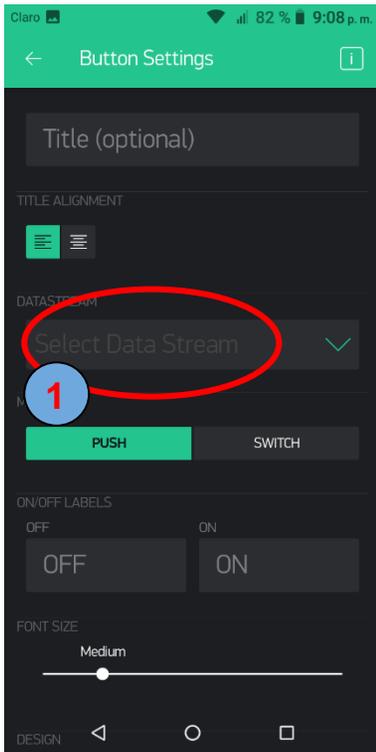


Nota: Es mejor usar pin virtual, porque el tiempo de respuesta es rápida

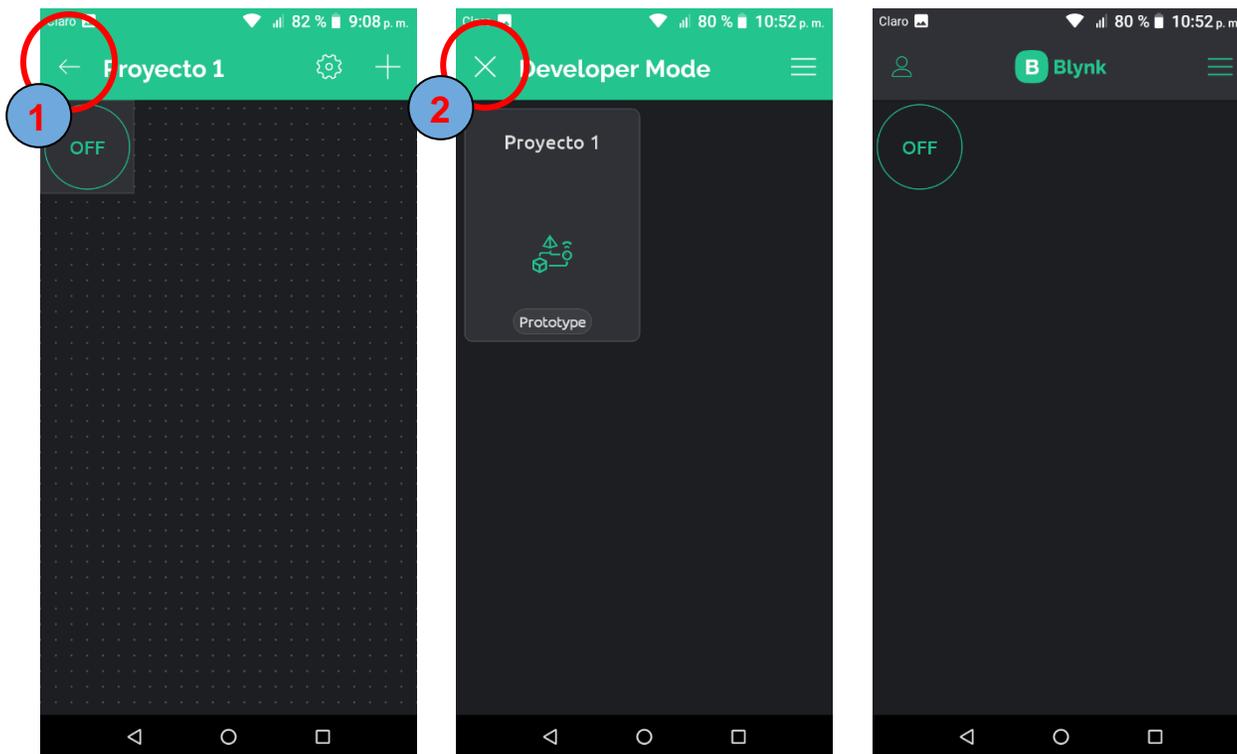
20- Crear nuestro tablero en la App de Blynk IoT



21- Configurar botón

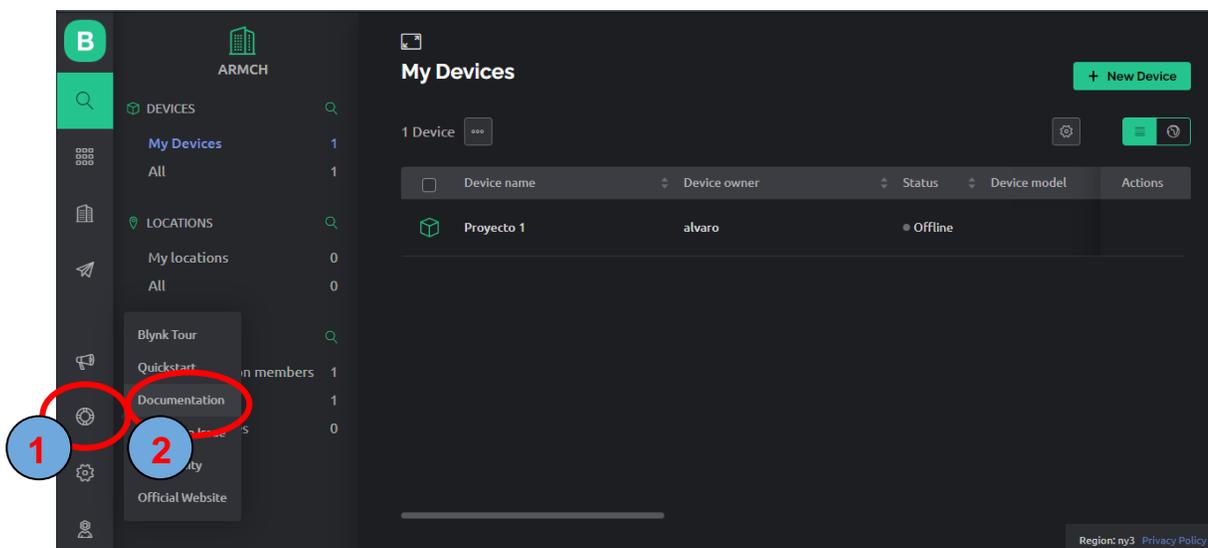


22- Salir modo desarrollador



Ahora puedes controlar el botón si está conectado el Arduino

Para obtener más información podemos encontrarlo en consola de blynk la documentación



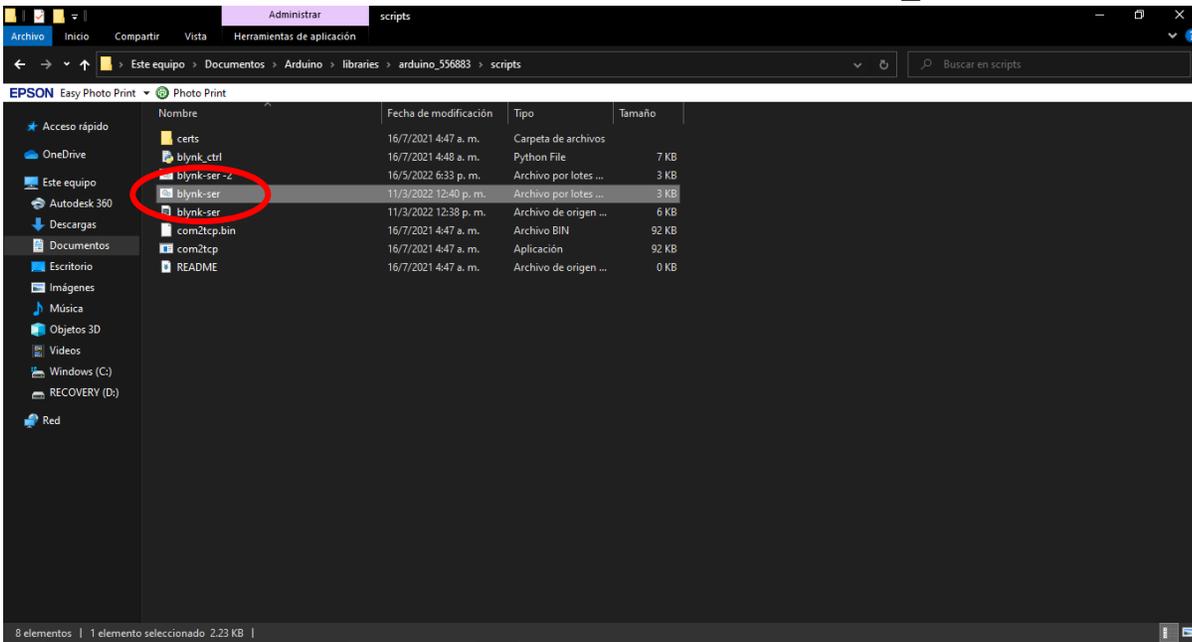
Arduino conectado por USB

Para conectar nuestro Arduino al servidor de blynk, necesitaremos de una computadora con acceso a internet.

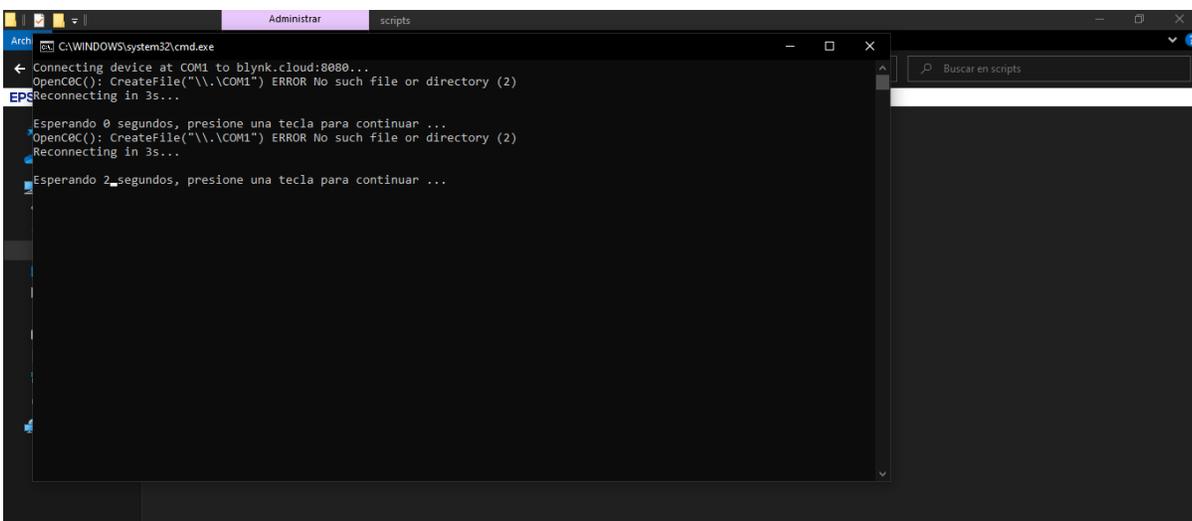
1- Buscar la carpeta de la librería de Blynk

Por lo general lo buscaremos en:

C:\Users\HP\Documents\Arduino\libraries\arduino_556883\scripts



2- Abrir el archivo **blynk-ser** para conectar nuestro arduino al servidor



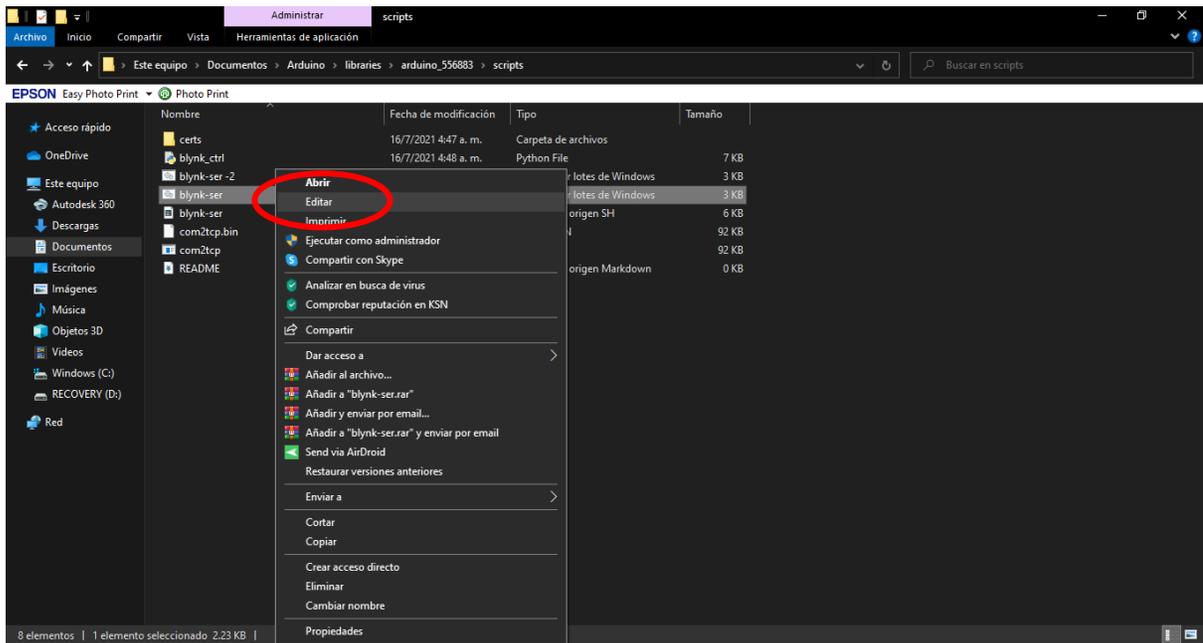
Nota: Puedes crear un acceso directo en escritorio para no tener dificultad para buscarlo

Si no logra conectarse debemos asegurarnos:

- El arduino este conectado a nuestra computadora
- La computadora tenga acceso a internet
- El archivo **blynk-ser** esté correctamente configurado
-

Verificar la configuración **blynk-ser**:

- Dar click derecho, en editar



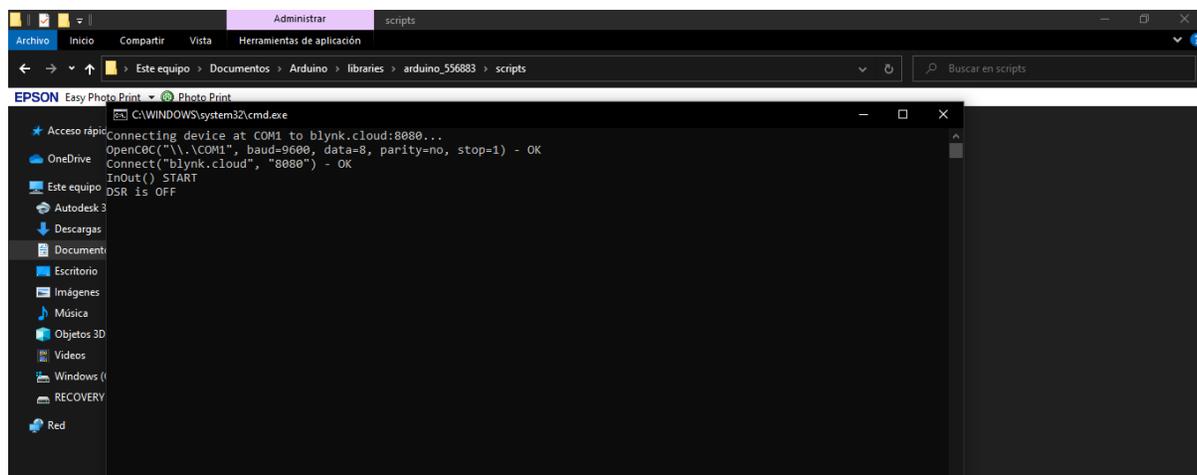
- Verificar que la dirección del servidor sea blynk.cloud



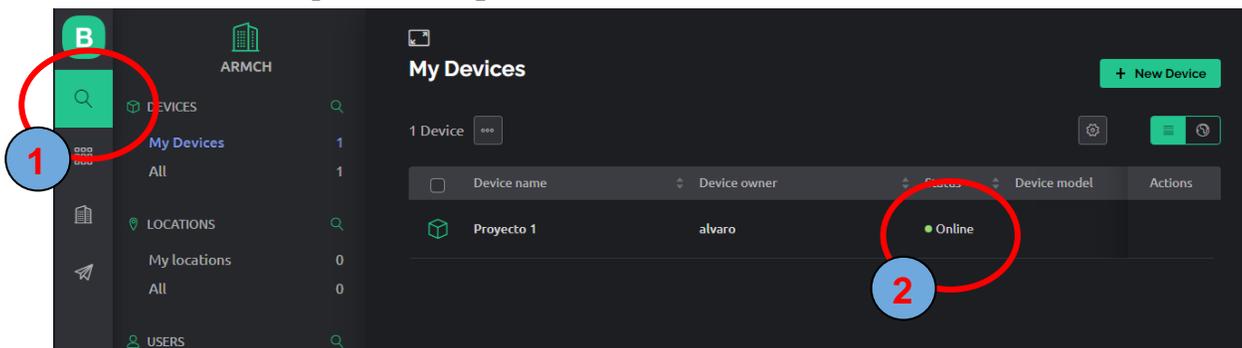
```
:usage
echo.
echo. This script redirects serial communication to the server.
echo.
echo. You can specify port, baud rate, and server endpoint like this:
echo. blynk-ser.bat -c ^<serial port> -b ^<baud> -s ^<server> -p ^<port>
echo.
echo. The defaults are:
echo. -c /dev/ttyUSB0 (on Linux)
echo. COM1 (on Windows)
echo. /dev/tty.usbserial (on OSX)
echo. -b 9600
echo. -s blynk.cloud
echo. -p 8080
echo.
echo. If the specified serial port is not found, it will ask to enter another one.
echo. The script also tries to reestablish connection if it was lost.
goto:eot
```

Nota: Por defecto el puerto es COM1, debemos editarlo según el puerto que esté conectado nuestro arduino, podemos crear una copia de este archivo, guardarlo según el puerto que hemos elegido.

3- Abrir el archivo **blynk-ser** para conectarse a blynk deberá mostrarse así



4- Verificar que en Blynk esté conectado



Nota: Para desconectarse o cargar nuevo programa debemos cerrar el archivo **blynk-ser**.

ANEXO 3

El proyecto contará con una serie de información que será encontrada en una memoria de 16 Gb, con el nombre "Robot de Limpieza, la cual contendrá toda la información correspondiente al proyecto, como planos, códigos y diseños de PCB de circuitos electrónicos, como sus archivos editables.

También dicha información se podrá encontrar en un enlace de Google Drive, el cual contendrá la información anteriormente descrita.

Link Google Drive:

<https://drive.google.com/drive/folders/1ujkpIjbLRveeQK242TrhWGzw8EDmwPVq?usp=sharing>