

# Informe de proyecto - Incubadora

<b>Grupo Nº:</b>	<b>2</b>	<b>Año y División:</b>	<b>2025 - R4051</b>
<b>Integrantes:</b>	<b>1 - CARRETTONI, LUCIANO 2 - NANNI, FRANCO 3 - NOÉ, MAGDALENA CECILIA 4 - RIOS, ALEXANDER DANIEL</b>		



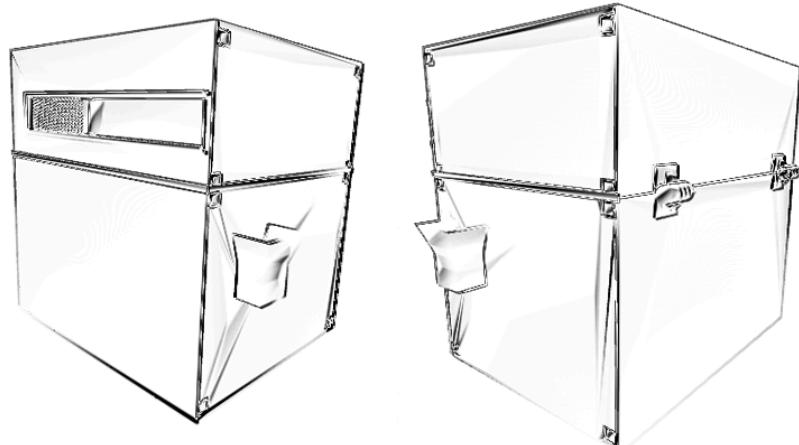
## Descripción general

La incubadora es un dispositivo que se encarga de autorregular los parámetros de temperatura y humedad, manteniéndolos en un rango óptimo para la incubación de huevos de gallina. Asimismo rota los huevos periódicamente por medio de un motor.

El proyecto presenta un display LCD y un decoder por el cual el usuario puede visualizar lecturas actuales de humedad, temperatura, motor y estado aproximado de incubación, el cual es un proceso de 21 días total.

También se puede configurar comunicación inalámbrica para obtener alarmas en caso de que un parámetro no pueda autorregularse, o el motor falle. Esta comunicación se realiza por medio de un bot de telegram.

Está desarrollado en torno a la placa STM32 Nucleo F446RE, la cual se conecta por medio de un PCB que regula las 4 tensiones utilizadas por los dispositivos del proyecto; 3v3,5,12 y 220.

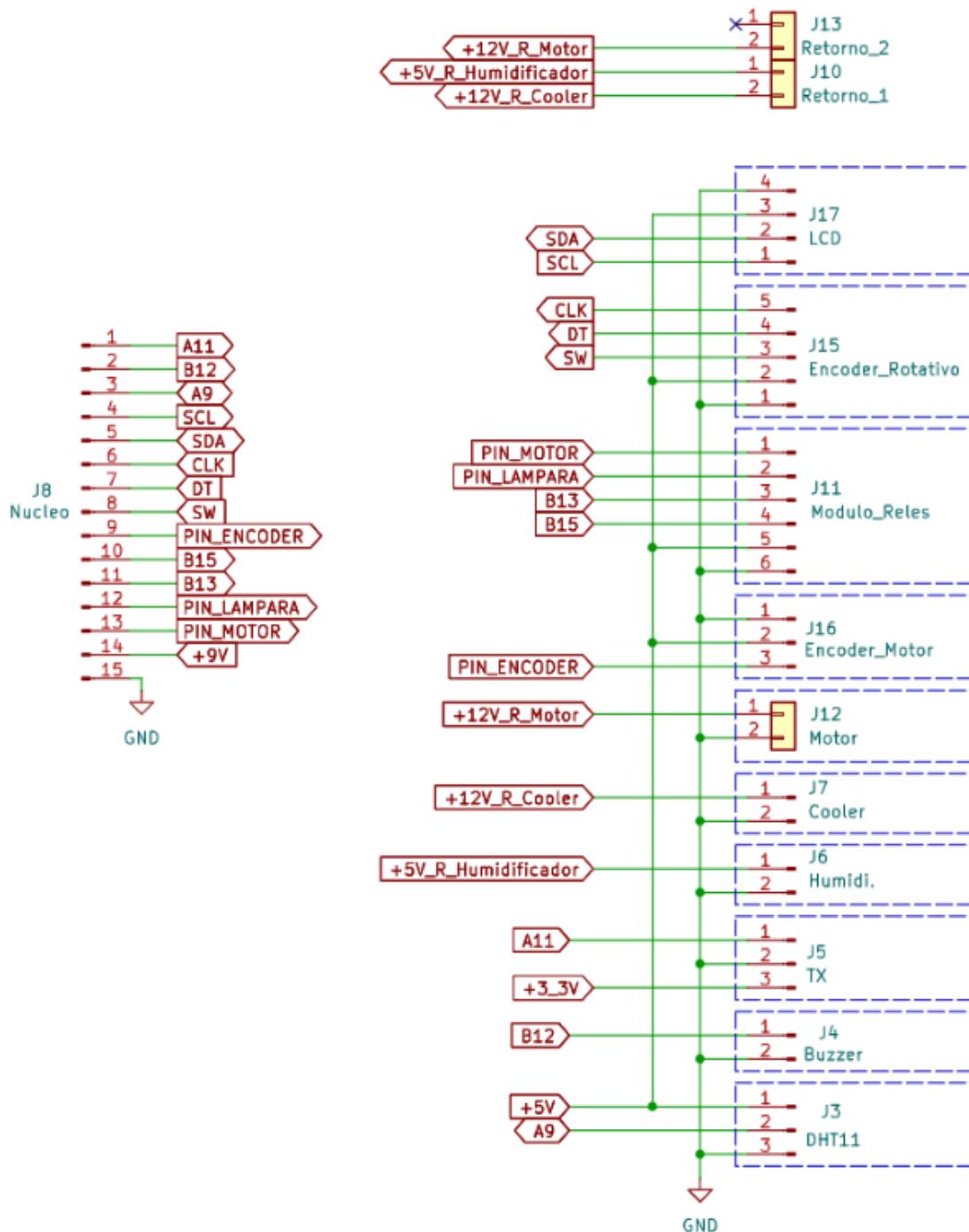


# Indice

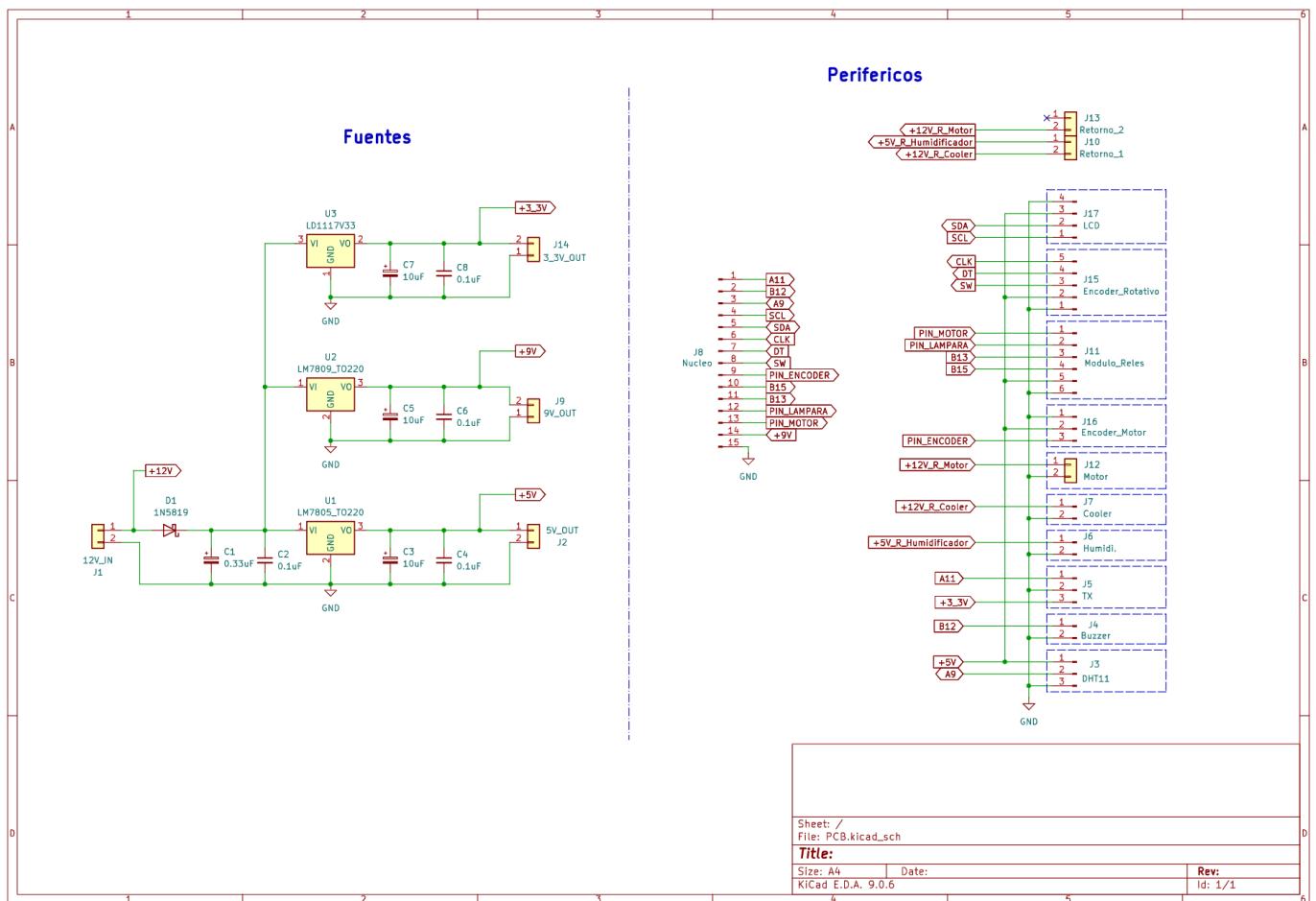
<b>Descripción general.....</b>	<b>0</b>
<b>Indice.....</b>	<b>1</b>
<b>Hardware.....</b>	<b>2</b>
Pinout.....	2
Esquematico.....	3
Descripción y circuito impreso.....	3
Diseño 3D.....	4
<b>Software.....</b>	<b>6</b>
Entorno de desarrollo.....	6
Flujo programa principal.....	6
Diagrama en bloques.....	7
<b>Dispositivos empleados.....</b>	<b>7</b>
Humedad y temperatura - DHT11.....	7
Autorregulación de parametros - Array Reles.....	8
Logica de control.....	9
Identificación en la incubadora.....	9
Humidificador.....	10
Cooler.....	10
Lampara.....	10
Comunicación inalámbrica - ESP32.....	10
ESP32 C3 Super Mini.....	10
Bot telegram.....	11
Alarma - Buzzer.....	11
Rotación - Motor y Decoder.....	12
Motor.....	12
Decoder.....	13
Interfaz de usuario - LCD y Decoder.....	13
LCD 16x2 I2C.....	14
Decoder KY-040.....	14
Tiempo - Memoria flash.....	14
<b>Referencias.....</b>	<b>15</b>
Repositorio.....	15
Datasheets.....	15

# Hardware

## Pinout



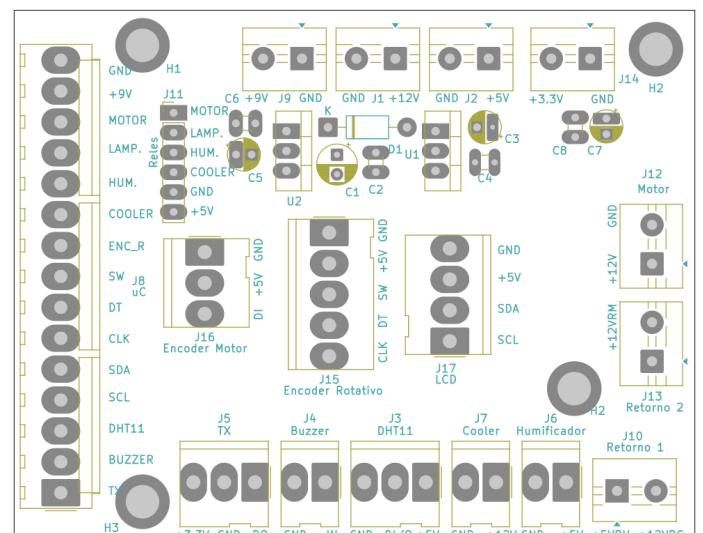
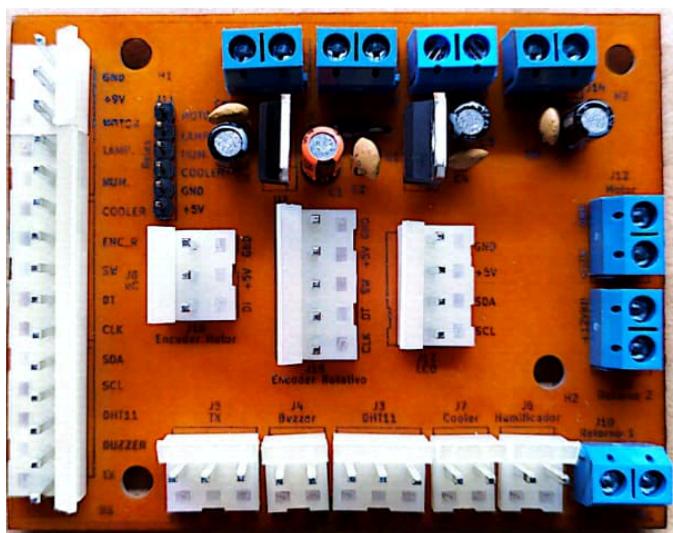
## Esquemático



## Descripción y circuito impreso

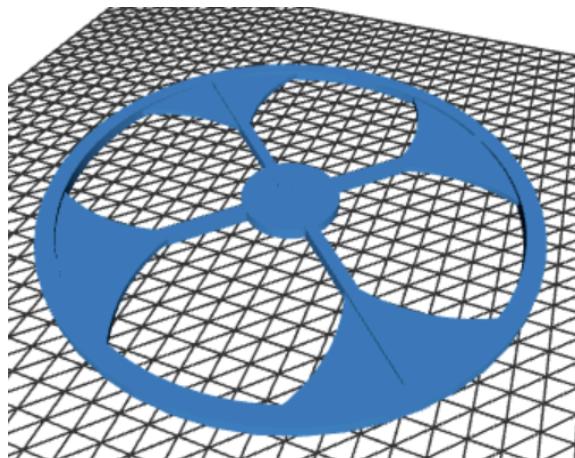
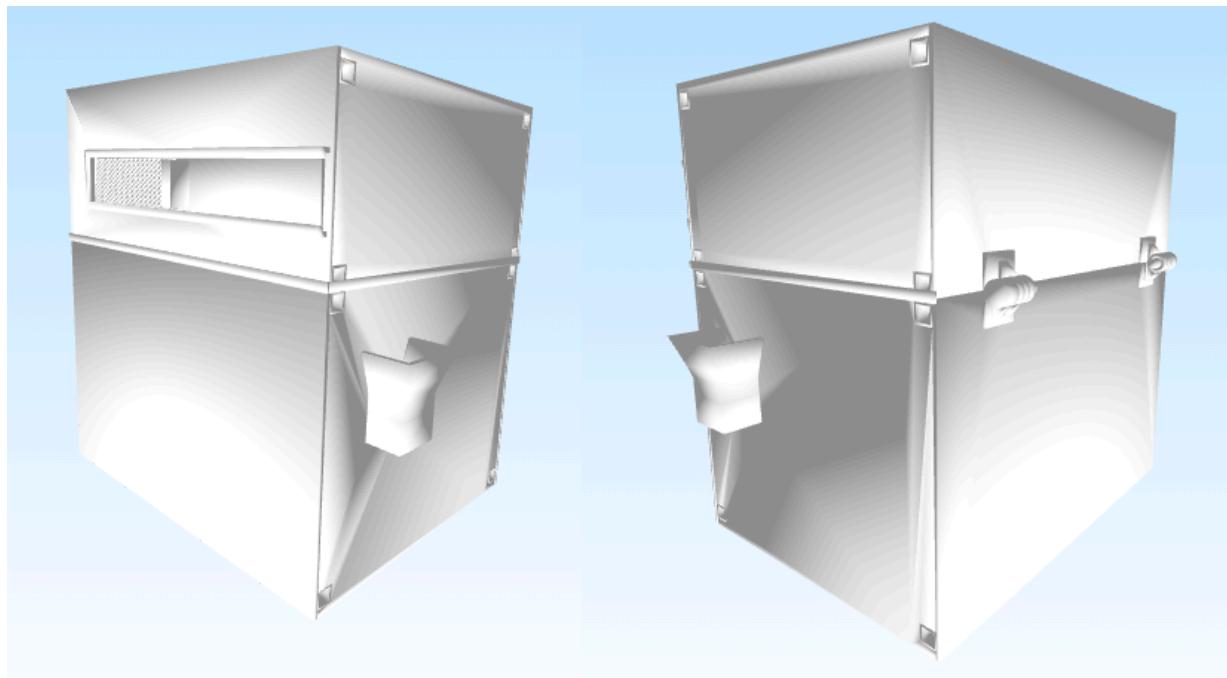
El PCB se encarga de alimentar y conectar los periféricos entre sí. Adapta 220v a las otras tres tensiones utilizadas: 12v, 5v y 3,3v.

Descargar el archivo [ibom.html](#) para obtener una lista de los materiales interactiva.

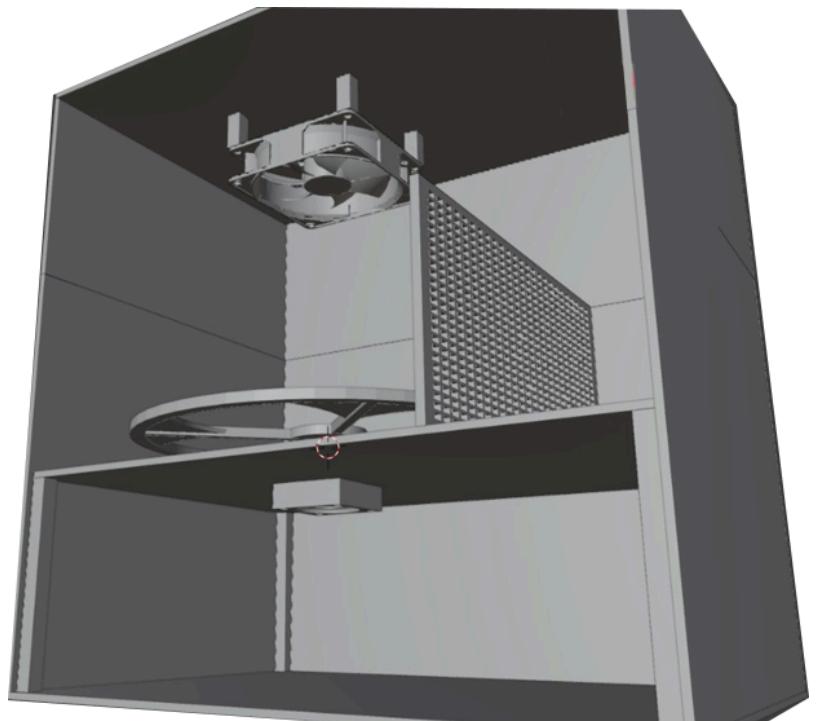


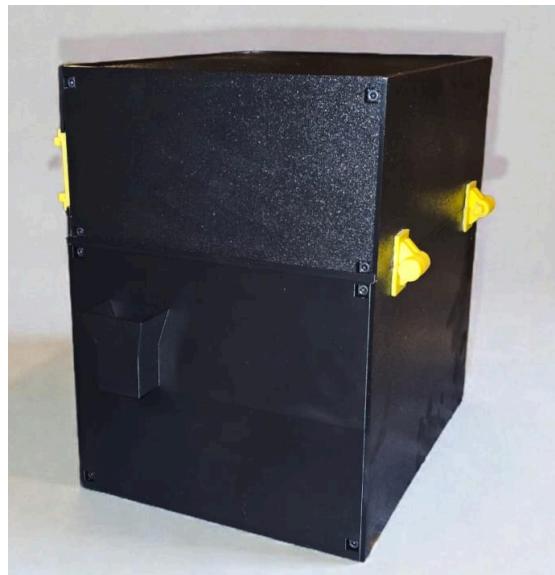
## Diseño 3D

El cuerpo de la incubadora consta de un cofre con una entrada lateral para cargar el humidificador y una ventana en la parte superior. En su interior cuenta con un carrousel en el cual se colocan los huevos. Este disco será rotado por el motor, que se encuentra en el compartimiento inferior junto con la mayoría de los componentes electrónicos.



Puede acceder a los archivos imprimibles por medio de [este link](#), o en la sección de Referencias.

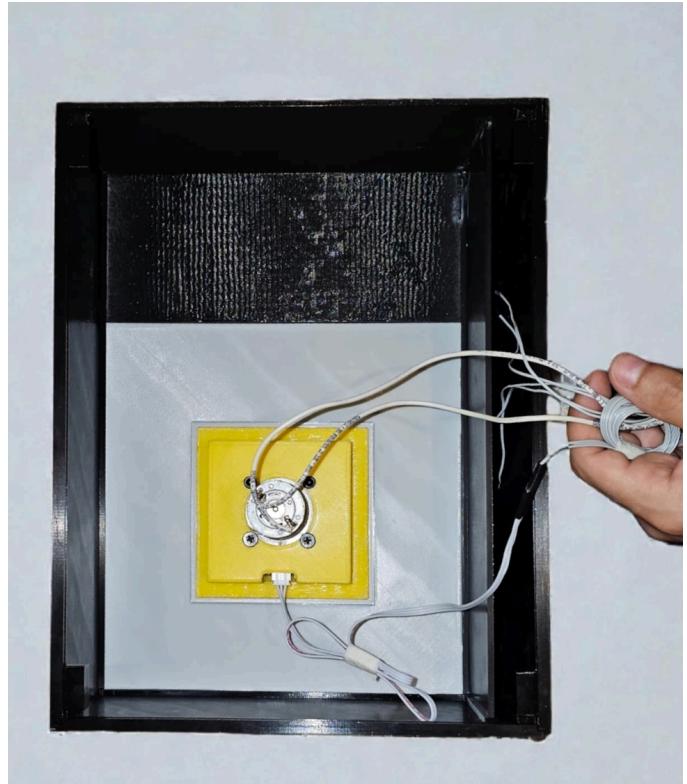




Exterior de la incubadora



Interior de la incubadora



Espacio inferior para la electrónica con el soporte del motor

# Software

## Entorno de desarrollo

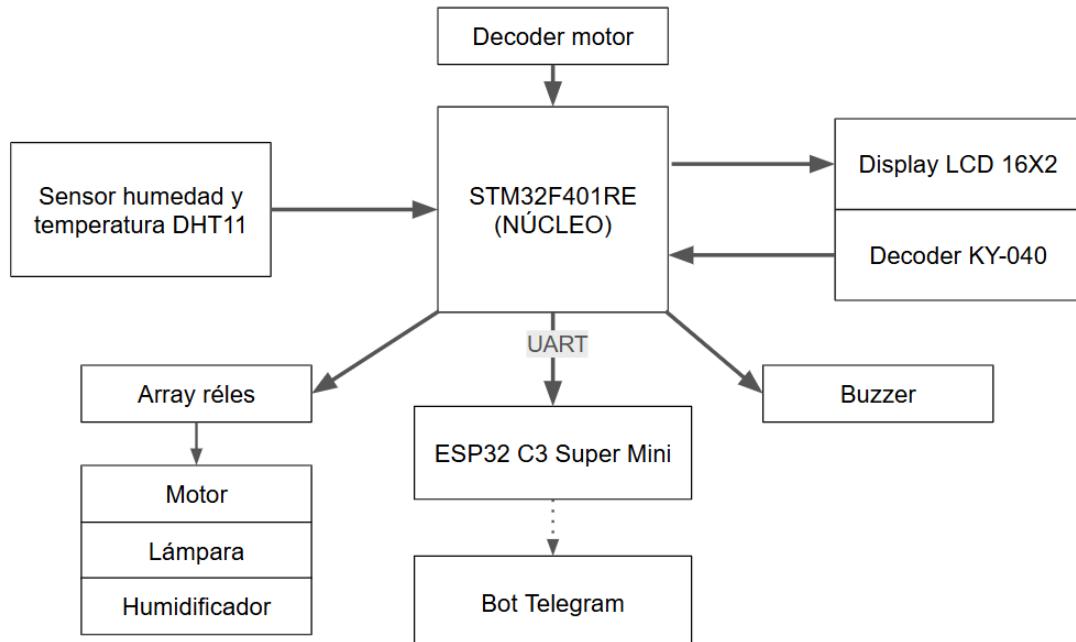
El proyecto fue desarrollado utilizando el entorno STM32CubeIDE de STMicroelectronics. El lenguaje de programación empleado es C estándar.

Se utiliza FreeRTOS, un sistema operativo en tiempo real, para gestionar de manera simultánea las tareas del sistema. La configuración específica del proyecto se encuentra en *freertos.c*.

## Flujo programa principal

1. Inicialización HAL y clock del sistema.
2. Configuración de periféricos (*GPIO*, *I2C*, *UART1*, *UART2*, *TIM1*).
3. Inicialización de módulos:
  - a. Pull-ups internas para protocolo I2C (Display).
  - b. Carga de la configuración desde Flash (*Load\_Config\_From\_Flash*).
  - c. Inicialización display.
  - d. Inicialización DHT11.
4. Inicialización de FreeRTOS: Creación de colas de mensajes (*menuQueueHandle*).
5. Creación de tareas:
  - a. *menuTask*: Interfaz de usuario y navegación.
  - b. *debounceTask*: Filtrado de rebotes del botón del encoder.
  - c. *MotorTask*: Cálculo de RPM a partir de pulsos del encoder del motor.
  - d. *ControlTask*: Lazo principal de control ambiental (temperatura/humedad).
  - e. *SensorTask*: Lectura periódica del DHT11, actualiza valor de temp. y humedad.
6. Creación de cola:
  - a. *menuQueueHandle*: Transmite cuando el decoder del menú se gira o pulsa.
7. Creación semáforo:
  - a. *xDHTMutex*: Asegura que sólo una función acceda al sensor DHT11 a la vez.
8. Inicio del planificador (*osKernelStart*).

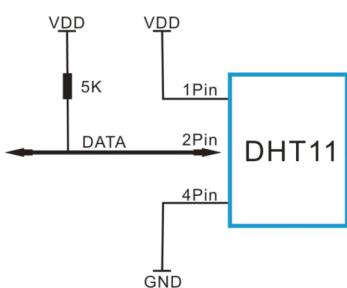
## Diagrama en bloques



## Dispositivos empleados

### Humedad y temperatura - DHT11

Los parámetros que se autorregulan son humedad y temperatura. Estos se obtienen por medio del sensor DHT11 (1). Dicho sensor presenta 3 pines, 2 para su alimentación de 3v3, y el tercero para la comunicación con el mismo. Es decir, la comunicación se realiza mediante una interfaz serial de un solo cable bidireccional. Se transmiten 40 bits de datos en 4ms.

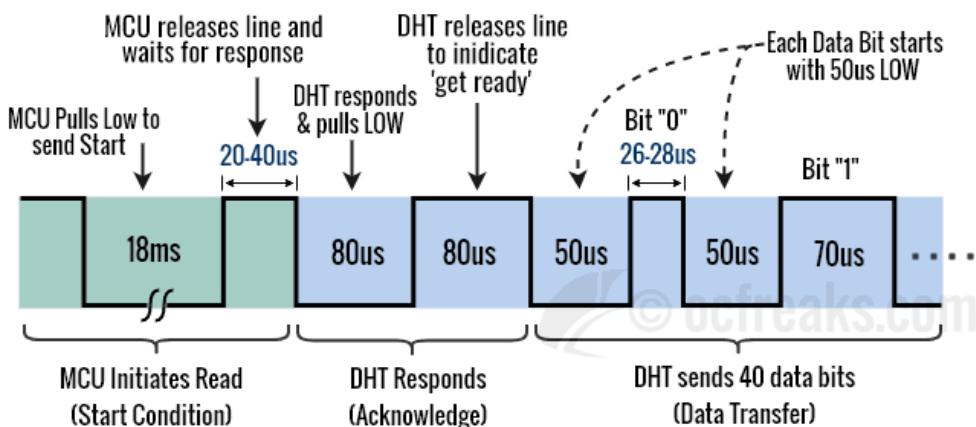


Para iniciar la comunicación primero se configura el pin data como output y se envía un pulso bajo por al menos 18ms. Luego el pin se cambia a input, y se espera la respuesta de un pulso bajo de 80us, seguido de un pulso alto de 80 us. Esto indica que el DHT11 escuchó exitosamente el pedido de datos, y está listo para transmitir.

Cada uno de los 40 bits de datos comienzan con un pulso bajo de 50us, seguido por un pulso alto que, según su duración,

determina si el bit es un 1 o un 0. 26-28 us significa '0', y 70 us significa '1'. Estos tiempos de pulso bajo y alto se miden con un timer interno de la placa de desarrollo (TIM1), y se guardan en un array para luego compararlos entre sí. Si el pulso bajo es mayor al pulso alto, significa que tal bit es un '0', y viceversa.

## DHT11 / DHT22 Protocol

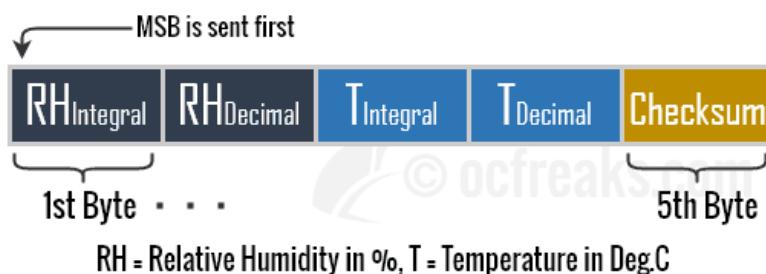


Para verificar la información se realiza un checksum con los primeros 4 bytes.

Al finalizar el sensor envía otro pulso bajo de 50 us y retorna a su estado idle de pulso alto.

Los 40 bits enviados corresponden a: humedad entera, humedad decimal, temperatura entera, temperatura decimal y el checksum.

## DHT11 / DHT22 Data Format

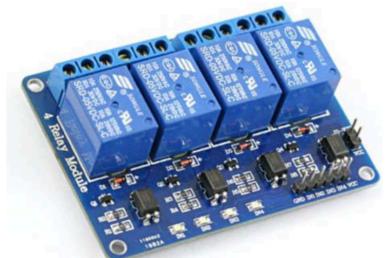


## Autorregulación de parámetros - Array Reles

Por medio de las lecturas del sensor de humedad y temperatura se establecen histéresis de acuerdo al estado en el que se encuentre la incubación:

Periodo [Días]	Temperatura ideal [°C]	Humedad [%]
0 - 18	37.5	50% - 55%
18 - 21	37.2	65% - 70%

Para controlar estos parámetros se utiliza una lámpara halógena, un cooler y un humidificador. Estos 3 dispositivos son controlados por la nucleo por medio de un array de relés.

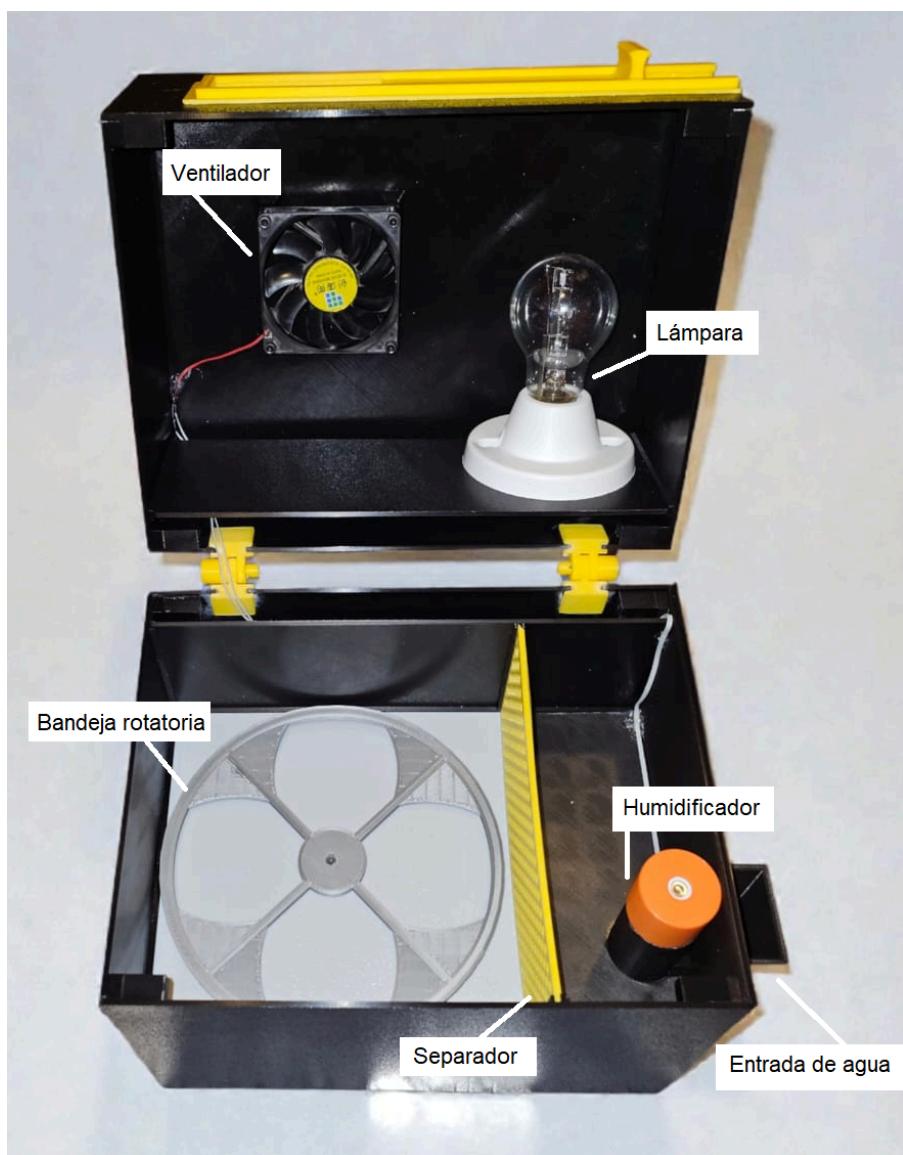


## Logica de control

El encendido y apagado de los mismos se encuentra en relay.c, controlado por medio del uso de la función `HAL_GPIO_WritePin`. La lógica de control de estos dispositivos se encuentra en logica.c, donde de acuerdo a la histéresis de humedad y temperatura se encienden y apagan los dispositivos con la función `control_parametros`. La temperatura tiene prioridad por sobre la humedad.

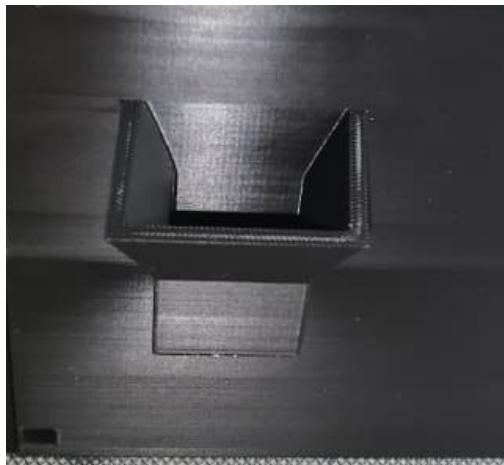
Los valores de humedad máxima y mínima, y temperatura máxima y mínima dictaminan las histéresis, y se cambian a partir del día 18 de incubación, diferenciando este cambio como etapa 1 y etapa 2. El cambio de etapa lo dictamina la memoria Flash, como se desarrolla más adelante.

## Identificación en la incubadora



## Humidificador

Se trata de un humidificador ultrasónico, el cual funciona vibrando una placa metálica rápidamente, atomizando el agua en una fina bruma que es dispersada por un ventilador, aumentando la humedad del ambiente sin calentar el agua. Este se alimenta con 5V. Para funcionar debe ser llenado con agua por la entrada lateral de la incubadora:



*Entrada externa para la carga de agua del humidificador.*

## Cooler

Un ventilador que se enciende tanto para disminuir la humedad y la temperatura. Se alimenta con 12V.

## Lámpara

Se trata de una lámpara halógena alimentada por 220V.

## Comunicación inalámbrica - ESP32

La incubadora es capaz de enviar alertas al celular del usuario por medio de un bot de telegram. A continuación se detalla el funcionamiento del sistema.

## ESP32 C3 Super Mini

Este microcontrolador recibe por el pin Rx (20) los mensajes a enviar por medio de telegram. Se conecta a una red wifi configurada por el usuario. El Usuario configura la red wifi a la cual estará conectada desde su celular, proceso el cual se inicia automáticamente al alimentar la placa.

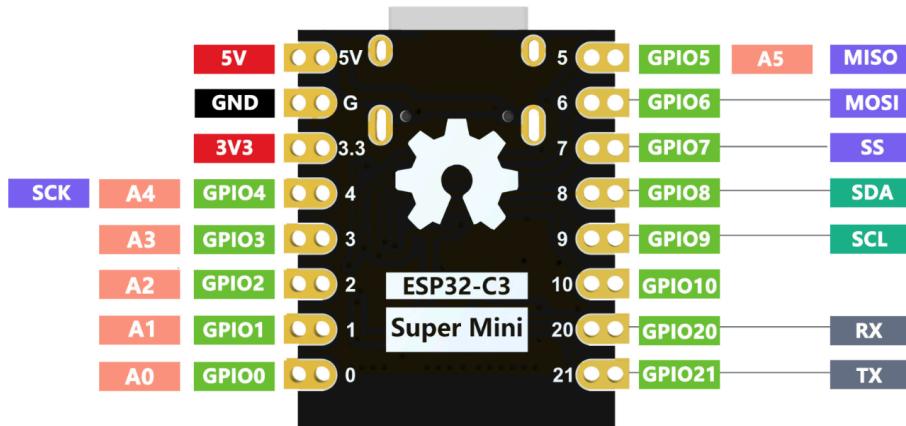
Acceder a ESP32-Telegram-Bri... :  
8.8.8.8

## WiFiManager

ESP32-Telegram-Bridge



Una vez configurado se encuentra chequeando constantemente si hay un nuevo mensaje a enviar. Este cuenta con su propio código C, aparte del de la placa núcleo. El mismo es alimentado con 3v3.



Para enviar los mensajes desde el bot se utiliza la biblioteca *CTBot.h*, donde con la función *sendMessage()* y la ID del chat se pueden enviar mensajes.

## Bot telegram

El bot es creado por medio de BotFather, la cual es una herramienta proporcionada por Telegram que permite la creación y configuración de bots personalizados. El usuario se conecta con el mismo por un grupo, en donde recibirá los mensajes enviados desde la nucleo, por medio del ESP32.

## Alarma - Buzzer

Si el dispositivo entra en alarma se encenderá un buzzer en la incubadora y enviará un mensaje por telegram. Se puede ingresar en alarma por la siguientes razones:

- La temperatura no se encuentra en un rango aceptable
- La humedad no se encuentra en un rango aceptable

Dicho buzzer es controlado por la nucleo de forma directa por un pin digital.

Existen 2 tipos de alarmas; alarma normal y alarma crítica.

La alarma normal es la primera que se dispara, sucede cuando la temperatura o humedad están fuera de rango por 2 minutos.

Luego la alarma crítica se activará cuando la temperatura o humedad se mantengan fuera de rango por 10 minutos.

Ante la activación de ambas alarmas se enviarán mensajes de telegram, especificando el tipo de error. En caso de que el problema se solucione el buzzer se apagará y se enviará un mensaje rectificando la alarma. En caso de haber ingresado en alarma crítica se recomienda al usuario verificar la incubadora en persona, aun cuando se haya salido de estado alarma.

Este tiempo fuera de valores aceptables se mide por `HAL_GetTick()` y las variables `alarma_temp_countdown` y `alarma_humedad_countdown` respectivamente. Si los valores se encuentran en histeresis, el counter se resetea. Si el counter supera los tiempos establecidos se ingresa en alarma o alarma crítica.

Dicho proceso se puede encontrar en `logica.c` con las funciones `reset_alarma_temp`, `reset_alarma_humedad` y `control_alarma`.

## Rotación - Motor y Decoder

La rotación de los huevos se hace por una estructura circular que rota horizontalmente por medio de un motor colocado en su centro. El motor se encuentra en un espacio inferior, que solo lo accede un técnico en caso de ser necesario. Allí también se encuentra el resto de la electrónica.



## Motor

El sistema utiliza un motor de corriente continua de 12V para realizar la rotación periódica de la bandeja de huevos. Se controla por medio de un relé. La tarea `Motortask` será llamada cuando gire el motor, una vez cada dos horas por default. Durante la etapa de eclosión (días 18-21) la rotación se detiene.

## Decoder

Para verificar que el motor está funcionando correctamente se implementó un sistema de detección por un sensor que genera pulsos por cada revolución del eje del motor. No es de especial interés conocer la velocidad del motor, pero si verificar que este esté rotando. La lectura de este decoder se utiliza para controlar cuánto rota el motor, evitando que los pollitos en incubación sean poco rotados o muy alterados.

## Interfaz de usuario - LCD y Decoder

Además de la comunicación inalámbrica el usuario interactuará con la incubadora por medio de una interfaz integrada a la incubadora. Consta de una pantalla LCD 16x2 y un encoder rotativo con pulsador, permitiendo la visualización del estado en tiempo real y la configuración de parámetros clave del ciclo de eclosión.

Por medio de una máquina de estados implementada en *i2c-lcd.c* se puede navegar por diferentes pantallas. Por default la incubadora siempre estará mostrando temperatura actual y objetivo, humedad actual, estado control de humedad, y día y hora del ciclo de incubación.

Pulsando el decoder se accede al Menú, desde el cual se encuentran las opciones:

- **Ver Sensores** Vuelve a la pantalla principal
- **Ciclo: INICIAR/PAUSAR** Inicia un nuevo ciclo o pausa el actual.
- **Configuracion**
  - **CFG: Desarrollo**: Ajusta objetivos para los primeros 18 días.
  - **CFG: Eclosión**: Ajusta objetivos para los últimos 3 días.
  - **Ciclo Total**: Cambia la duración total (por defecto 21 días).
  - **Ajustar Tiempo**: Permite corregir manualmente el día y hora actual del ciclo.
  - **Volver <-**: Retorna al menú
- **TEST** Permite apagar manualmente y de forma individual el cooler, humidificador, luz y motor.
- **Volver <-**: Retorna a pantalla principal

La tarea menuTask espera eventos como el giro de encoder o la pulsación del botón desde la cola menuQueueHandle. Se emplea una técnica de "update-on-change" en el Dashboard para evitar parpadeos innecesarios del LCD.





Ingresando al menú TEST y seleccionando el encendido manual de la luz.

## LCD 16x2 I2C

Se trata de un display 16x2 estándar de comunicación I2C alimentado por 5V.

## Decoder KY-040

Es un encoder rotacional que entrega información acerca de cuánto ha sido rotado y en qué dirección. Tiene un ángulo mecánico de 360° continuos. Así mismo funciona como un pulsador. Se alimenta con 5V.(2)

Rotando el mismo se puede navegar por el menú, scrolleando las opciones y seleccionando una presionando la perilla.



## Tiempo - Memoria flash

El tiempo de la incubación, proceso el cual es de aproximadamente 21 días, se gestiona por medio de la flash, borrando y sobreescritiendo el tiempo acumulado de incubación. Así se permite que en caso de un corte en la alimentación del dispositivo, este inicie desde la etapa de incubación en la que se encontraba previamente.

Este cálculo se encuentra en `utils.c`, en la función `Get_Current_Hour()` y `Get_Current_Minute()`. El cálculo se basa en el tiempo total acumulado desde el inicio del proceso, combinando el tiempo guardado previamente en memoria Flash (`sysData.saved_timestamp`) y el tiempo transcurrido desde el último arranque del sistema (`HAL_GetTick() - sysData.last_boot_tick`). Se mide en minutos a partir del contador de milisegundos del sistema `HAL_GetTick()`.

# Referencias

## Repositorio

Proyecto RTOS:

[https://github.com/aletbm/Incubadora\\_Agricola\\_PI\\_TD2\\_UTN\\_FRBA/tree/main/Incubadora\\_RTOS](https://github.com/aletbm/Incubadora_Agricola_PI_TD2_UTN_FRBA/tree/main/Incubadora_RTOS)

Código ESP32 C3 Super Mini:

[https://github.com/aletbm/Incubadora\\_Agricola\\_PI\\_TD2\\_UTN\\_FRBA/tree/main/esp32c3-telegram](https://github.com/aletbm/Incubadora_Agricola_PI_TD2_UTN_FRBA/tree/main/esp32c3-telegram)

Diseño PCB:

[https://github.com/aletbm/Incubadora\\_Agricola\\_PI\\_TD2\\_UTN\\_FRBA/tree/main/PCB](https://github.com/aletbm/Incubadora_Agricola_PI_TD2_UTN_FRBA/tree/main/PCB)

Diseño 3D:

[https://github.com/aletbm/Incubadora\\_Agricola\\_PI\\_TD2\\_UTN\\_FRBA/tree/main/modelo\\_3D](https://github.com/aletbm/Incubadora_Agricola_PI_TD2_UTN_FRBA/tree/main/modelo_3D)

## Datasheets

1. **DHT11 Sensor humedad y temperatura**  
<https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>
2. **KY-040 Decoder menu usuario**  
<https://www.handsontec.com/datasheets/module/Rotary%20Encoder.pdf>
3. **STM32 Nucleo (MB1319)**  
[https://github.com/aletbm/Incubadora\\_Agricola\\_PI\\_TD2\\_UTN\\_FRBA/blob/main/Manuales/stm32-nucleo64p-boards-mb1319-stmicroelectronics.pdf](https://github.com/aletbm/Incubadora_Agricola_PI_TD2_UTN_FRBA/blob/main/Manuales/stm32-nucleo64p-boards-mb1319-stmicroelectronics.pdf)
4. **ESP32 C3 Super Mini Modulo wifi**  
[https://github.com/aletbm/Incubadora\\_Agricola\\_PI\\_TD2\\_UTN\\_FRBA/blob/main/Manuales/ESP32-C3%20SuperMini%20datasheet.pdf](https://github.com/aletbm/Incubadora_Agricola_PI_TD2_UTN_FRBA/blob/main/Manuales/ESP32-C3%20SuperMini%20datasheet.pdf)
5. **Modulo 4 reles**  
[https://github.com/aletbm/Incubadora\\_Agricola\\_PI\\_TD2\\_UTN\\_FRBA/blob/main/Manuales/4Ch-relay.pdf](https://github.com/aletbm/Incubadora_Agricola_PI_TD2_UTN_FRBA/blob/main/Manuales/4Ch-relay.pdf)