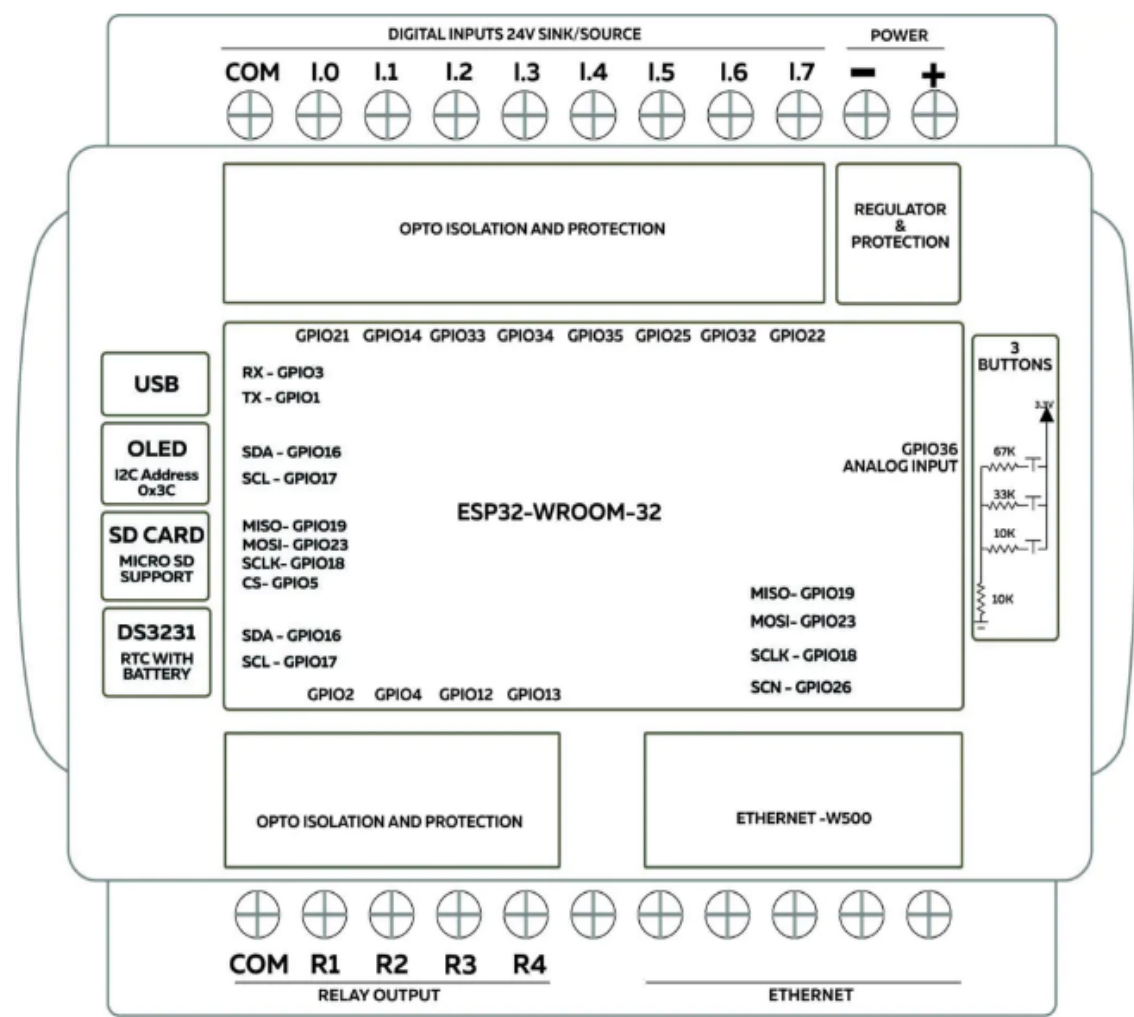


NORVI ENET-AE06-R – Ficha de Datos

Características del Producto



- ESP32-WROOM32
- Módulo con pantalla OLED integrada de 0,96
- Soporte de tarjeta micro SD
- DS3231 RTC con batería de respaldo
- Ethernet a través de W5500
- Botón incorporado en el panel frontal
- Entradas digitales
- Salidas de relé
- Montaje en carril DIN

Ampliaciones Compatibles

- Salida de Relé
- Entrada Digital
- Salida Analógica

Inicio

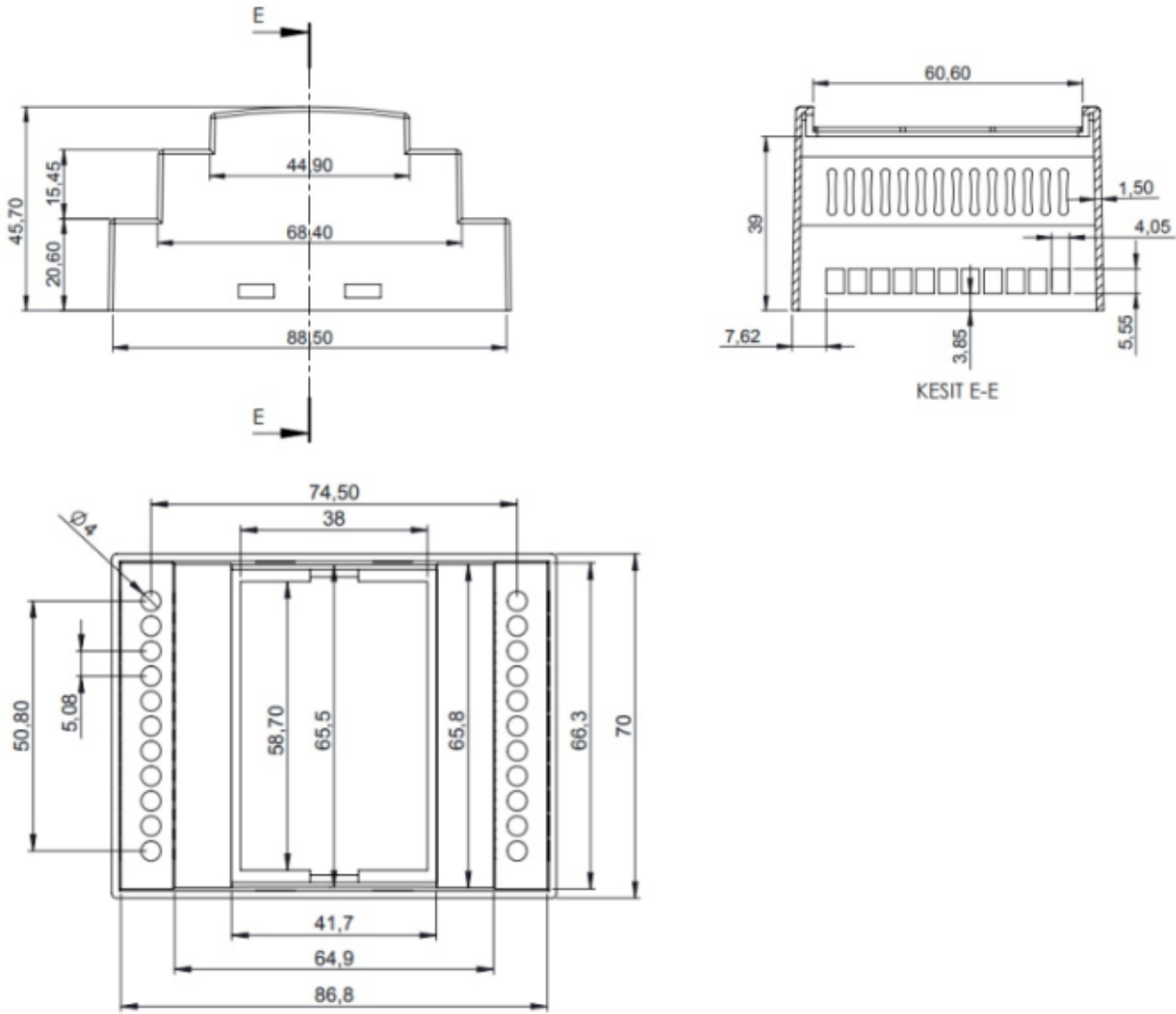
Gama de Producto	NORVI ENET
Tipo de Producto	Controlador Programable
Certificaciones	EN 61131-2:2007 EN 61010-1:2010+A1:2019 EN IEC 61010-2-201:2018 2014/30/EU- Compatibilidad Electromagnética (EMC) Anexo III, Parte B, Modulo C
Tensión Nominal de Alimentación	24V DC
Comunicación	WiFi / Bluetooth
Entradas y Salidas	8 x Entradas Digitales 4 x Salidas Relé
Pantallas e Indicadores Visuales	0,96 Pantalla OLED e indicadores

Complementarios

Código Unificado de Producto	NORVI ENET-AE06-R
Números de Pieza de Producto	NORVI ENET-AE06-R

Características Mecánicas

Carcasa	NORVI 204
Montaje / Método de Instalación	RIEL DIN / PESTAÑAS DE MONTAJE
Tipo de Terminal	SCREW TERMINAL
Disposición de los terminales	Top and Bottom
Largo	90.50 mm
Altura	56.60 mm
Ancho	60.60 mm



Entorno

Grado de Protección IP	IP20
Altitud de Funcionamiento	0–2000 metros
Temperatura de Funcionamiento	– -10... +85° C (14...185 °F)
Altitud de Almacenaje	0–3000 metros
Resistencia a Golpes	15 gn para 11ms
Resistencia a Descargas Electrostáticas	kV en contacto 8 kV en el aire
Resistencia a Campos Electromagnéticos	10 V/m (80 MHz 1GHz) 3 V/m (1.4 MHz 2 GHz) 1 V/m (2 MHz 3 GHz)

Características Eléctricas

Dispositivos Alimentados por la Red

Tensión nominal de alimentación (V)	24V DC
Consumo de Corriente (mA)	400mA
Fuente de Alimentación Recomendada	1A, 24V DC

Procesamiento

SOC / MCU	ESP-WROOM32
Flash Memory	4MB
ROM	448 KB
SRAM	520 KB
PSRAM	NO DISPONIBLE

Periféricos

Botones Incorporados

Botón 1 Pin	GPIO36 Entrada Analógica Nivel 1
Botón 2 Pin	GPIO36 Entrada Analógica Nivel 2
Botón 3 Pin	GPIO36 Entrada Analógica Nivel 3

Compatible con Tarjetas micro SD

Tipo de tarjeta	microSD
Interfaz	SPI
SD CARD CS	GPIO5
MISO	GPIO19
MOSI	GPIO23
SCLK	GPIO18
Detección SD	NO CONECTADO

RTC Interno

Chip RTC	DS3231
Tipo de Batería de Respaldo	CR2032
Interfaz	I2C
Dirección I2C	0x68
SCL Pin	GPIO17
SDA Pin	GPIO16

Conexiones GPIO del RTC

Pantalla OLED

Controlador de Pantalla	SSD1306
Tamaño de Pantalla	0.96 inch
PinSCL	GPIO17
Pin SDA	GPIO16
RESET Pin	NO CONECTADO

ENTRADAS y SALIDAS

Entradas Digitales

Numero de Entradas Digitales	8
Polaridad de Entradas Digitales	Ambos, Receptor de Corriente y Fuente (Sink and Source)
Tensión Máxima de Entrada Digital	32V DC
Tensión Mínima de Entrada Digital	18V DC
Frecuencia Máxima de Conmutación	1 kHz
Disposición de los Terminales	Digital Input 0 – GPIO21 Digital Input 1 – GPIO14 Digital Input 2 – GPIO33 Digital Input 3 – GPIO34 Digital Input 4 – GPIO35 Digital Input 5 – GPIO25 Digital Input 6 – GPIO32 Digital Input 7 – GPIO22

Salidas Relé

Número de Salidas de Relé	4
Tipo de Salida de Relé	Normalmente Abierto / SPST (Single-Pole, Single-Throw) / Electromecánico
Corriente Nominal de Contacto (Resistiva)	5 A 30V DC/250V AC
Tensión Máxima de Contacto	270V AC, 125V DC
Frecuencia Máxima de Conmutación	60 Hz
Disposición de los Terminales	Relay Output 0 – GPIO2 Relay Output 1 – GPIO4 Relay Output 2 – GPIO12 Relay Output 3 – GPIO13

Canales de Comunicación

Ethernet SPI - W5500

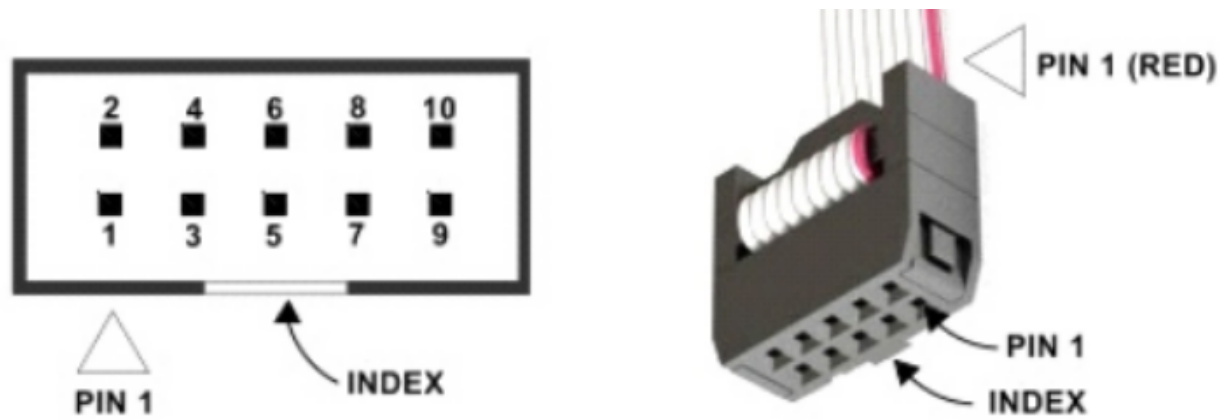
Transceptor	W5500
Velocidad	10BaseT/100BaseTX
Admite Negociación Automática	Si
Tamaño del Buffer TX/RX	Memoria interna de 32 Kbytes
Protocolos TCP/IP Cableados Compatibles	TCP, UDP, ICMP, IPv4, ARP, IGMP, and PPPoE
Número de tomas independientes simultáneas	8
SCSn	GPIO26
MISO	GPIO19
MOSI	GPIO23
SCLK	GPIO18
RSTn	GPIO27

Mapa GPIO

GPIO	Descripción	Uso
0	Puerto de Expansión	NRST
1	Puerto de Expansión	TX0
2	conectado al LED integrado	Salida Relé: 0
3	Puerto de Expansión	RX0
4	conectado al LED integrado	Salida Relé 1
5	Tarjeta micro SD	CS
6	conectado a la memoria flash SPI integrada	
7	conectado a la memoria flash SPI integrada	
8	conectado a la memoria flash SPI integrada	
9	conectado a la memoria flash SPI integrada	
10	conectado a la memoria flash SPI integrada	
11	conectado a la memoria flash SPI integrada	
12	conectado al LED integrado	Salida Relé 2
13	conectado al LED integrado	Salida Relé 3
14	Emite señal PWM al arrancar	Entrada Digital 1
15	Puerto de Expansión	Pin 6
16	Puerto de Expansión, I2C	SDA
17	Puerto de Expansión, I2C	SCL
18	microSD Card	SCLK
19	microSD Card	MISO
20		
21	sólo entrada	Entrada Digital 0
22	sólo entrada	Entrada Digital 7

23	Tarjeta micro SD	MOSI
24		
25	sólo entrada	Entrada Digital 5
26	RTC	SCSN
27		RSTn
28		
29		
30		
31		
32	sólo entrada	Entrada Digital 6
33	sólo entrada	Entrada Digital 2
34	sólo entrada	Entrada Digital 3
35	sólo entrada	Entrada Digital 4
36		Botones
37		
38		
39	sólo entrada	

Puerto de Expansión



PIN	ESP32 Conexión
1	No Conectado
2	TX0
3	5V
4	RX0
5	BOOT GPIO0
6	GPIO15
7	3.3V
8	SCL GPIO17
9	Tierra
10	SDA GPIO16

NORVI ENET-AE06-R – Guía del Usuario

Programación

El NORVI ENET-AE06-R tiene un mini puerto USB para la conexión en serie con el SoC para la programación. Se puede utilizar cualquier IDE de programación compatible con ESP32 para programar el controlador. Sigue esta Guía para programar controladores NORVI basados en ESP32 con el IDE Arduino.

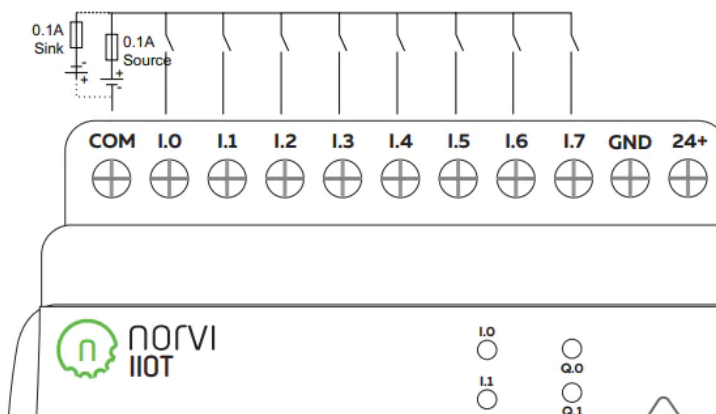
SoC: ESP32-WROOM32

Programming Port: USB UART

Entradas Digitales

Cableado de Entradas Digitales

Las entradas digitales del NORVI-ENET-AE06-R pueden configurarse como conexiones de origen y de destino. La polaridad inversa de la entrada digital debe suministrarse al terminal común.



NORVI-ENET-AE06-R Cableado de entrada digital

Programación de Entradas Digitales

La lectura del GPIO correspondiente del ESP32 da el valor de la Entrada Digital. Cuando las entradas están en estado OFF, el GPIO pasa a HIGH, y cuando la entrada está en estado ON, el GPIO pasa a LOW.

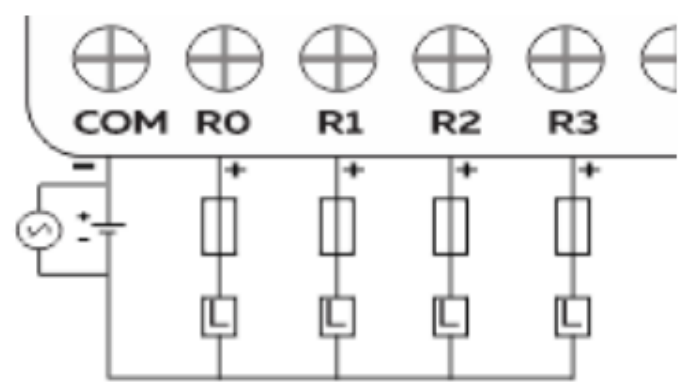
Consulte la tabla de asignación GPIO en la hoja de datos para la entrada digital GPIO

```
#define INPUT1 21

void setup() {
  Serial.begin(115200);
  Serial.println("Device Starting");
  pinMode(INPUT1, INPUT);
}

void loop() {
  Serial.print(digitalRead(INPUT1));
  Serial.println("");
  delay(500);
}
```

Salida Relé



NORVI ENET-AE06-R Cableado de salida de relé

Programación de Salidas de Relé

La lectura del GPIO correspondiente del ESP32 proporciona el valor de la salida del relé o transistor. Consulte la tabla de asignación de GPIO en la hoja de datos para el GPIO de salida de relé / transistor.

```
#define OUTPUT1 02

void setup() {
  Serial.begin(115200);
  Serial.println("Device Starting");
  pinMode(OUTPUT1 , OUTPUT);
}

void loop() {
  digitalWrite(OUTPUT1, HIGH);
  delay(500);
  digitalWrite(OUTPUT1, LOW);
  delay(500);
  Serial.println("");
  delay(500);
}
```

Pantalla OLED Integrada

Controlador de Pantalla	SSD1306
Comunicación	I2C
Dirección del Modulo	0x3C
Resolución	128 x 64

0,96 Especificaciones de la pantalla OLED

Consulte la tabla de asignación de GPIO en la hoja de datos para el GPIO I2C de la pantalla OLED. Librería soportada por el [Adafruit_SSD0306 Library](#). Wire.begin (SDA, SCL) es necesario para inicializar I2C en los pines correctos.

Programación Pantalla OLED

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
```

```
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
#define OLED_RESET -1 // Reset pin # (or -1 if sharing Arduino reset pin) Adafruit_SSD1306 display(SCREEN_WIDTH,
SCREEN_HEIGHT, &Wire, OLED_RESET);

void setup() {
  Wire.begin(16,17);
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    // Address 0x3C for 128x64 Serial.println(F("SSD1306 allocation failed"));
    for(;;); // Don't proceed, loop forever
  }
  // Show initial display buffer contents on the screen -
  // the library initializes this with an Adafruit splash screen.
  display.display(); delay(2000); // Pause for 2 seconds
}

void loop() {
}
```

Botones Incorporados

Modo de Lectura	ADC (Conversión Analógico a Digital)
Analógico IO (In-Out)	GPIO36

Conexión GPIO de los botones

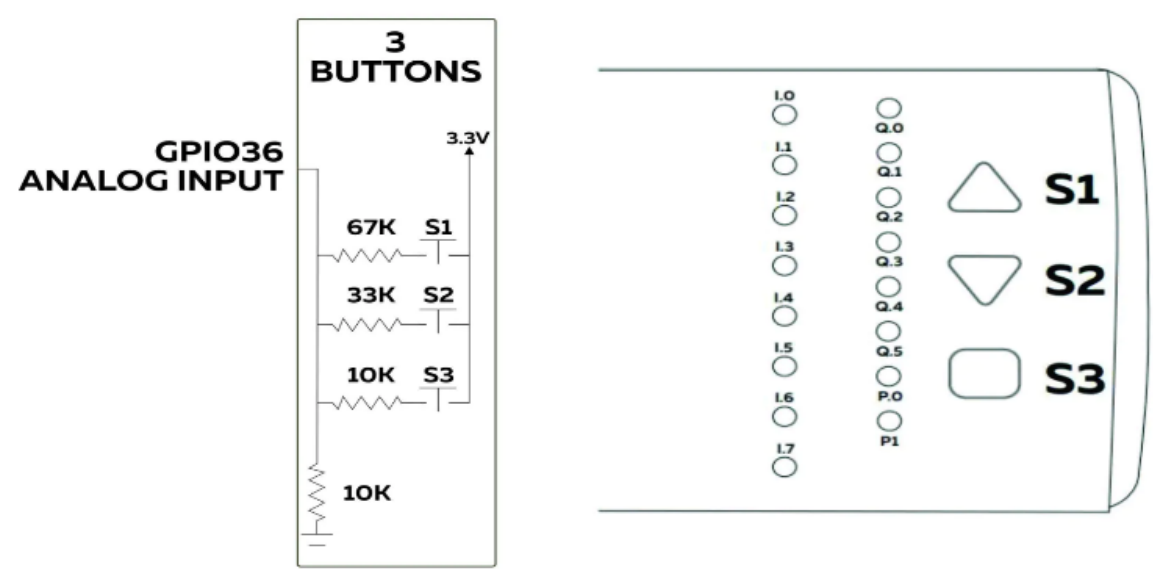


Diagrama Interno del Botón Incorporado

Programación de Botones

```
#define buttonPin 36
int  buttonState = 0;

void setup() {
  Serial.begin(9600);
  pinMode(buttonPin, INPUT);
}

void loop() {
  Serial.print("Button: ");
  buttonState = analogRead(buttonPin);
  delay(50);
  Serial.print(analogRead(buttonPin));
  Serial.print("\tAnalog: ");
  delay(1000);
}
```

RTC Interno

Chip RTC	DS3231
Tipo de Batería de Respaldo	CR2032
Interfaz	I2C
Dirección I2C	0x68
Pin SCL	GPIO17
Pin SDA	GPIO16

Conexiones GPIO del RTC

Programación del RTC

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include "RTClib.h"

RTC_DS3231 rtc;
char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

void setup() {
  Serial.begin(9600);
  if (! rtc.begin()) {
    Serial.println("Couldn't find RTC");
    while (1);
  }
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;); // Don't proceed, loop forever
  }
  rtc.adjust(DateTime(__DATE__, __TIME__));
  display.display();
  delay(2);
  display.clearDisplay();
  display.clearDisplay();
  display.setTextColor(WHITE); //display.startscrollright(0x00, 0x0F);
  display.setTextSize(2);
  display.setCursor(0,5);
  display.print(" Clock ");
  display.display();
  delay(3000);
}

void loop() {
  DateTime now = rtc.now();
  display.clearDisplay();
```

```
display.setTextSize(2);
display.setCursor(75,0);
display.println(now.second(), DEC);

display.setTextSize(2);
display.setCursor(25,0);
display.println(":");

display.setTextSize(2);
display.setCursor(65,0);
display.println(":");
display.setTextSize(2);
display.setCursor(40,0);
display.println(now.minute(), DEC);

display.setTextSize(2);
display.setCursor(0,0);
display.println(now.hour(), DEC);

display.setTextSize(2);
display.setCursor(0,20);
display.println(now.day(), DEC);

display.setTextSize(2);
display.setCursor(25,20);
display.println("-");

display.setTextSize(2);
display.setCursor(40,20);
display.println(now.month(), DEC);

display.setTextSize(2);
display.setCursor(55,20);
display.println("-");

display.setTextSize(2);
display.setCursor(70,20);
display.println(now.year(), DEC);

display.setTextSize(2);
display.setCursor(0,40);
display.print(daysOfTheWeek[now.dayOfTheWeek()]);
display.display();
}
```

Soporte para Tarjeta Micro SD

SD CARD CS	GPIO5
MISO	GPIO19
MOSI	GPIO23
SCLK	GPIO18

Conexiones GPIO de la Tarjeta SD

Programación de Tarjeta SD

```
#include <SD.h>
#define PIN_SPI_CS 05 // The ESP32 pin GPIO5
File myFile;

void setup() {
  Serial.begin(9600);
  if (!SD.begin(PIN_SPI_CS)) {
    Serial.println(F("SD CARD FAILED, OR NOT PRESENT!"));
    while (1); // stop the program
  }
  Serial.println(F("SD CARD INITIALIZED.));
  if (!SD.exists("esp32.txt")) {
    Serial.println(F("esp32.txt doesn't exist. Creating esp32.txt file..."));
    // create a new file by opening a new file and immediately close it
    myFile = SD.open("esp32.txt", FILE_WRITE);
    myFile.close();
  }
  // recheck if file is created or not
  if (SD.exists("esp32.txt"))
    Serial.println(F("esp32.txt exists on SD Card.));
  else
    Serial.println(F("esp32.txt doesn't exist on SD Card.));
}

void loop() {
}
```

USB y REINICIO

