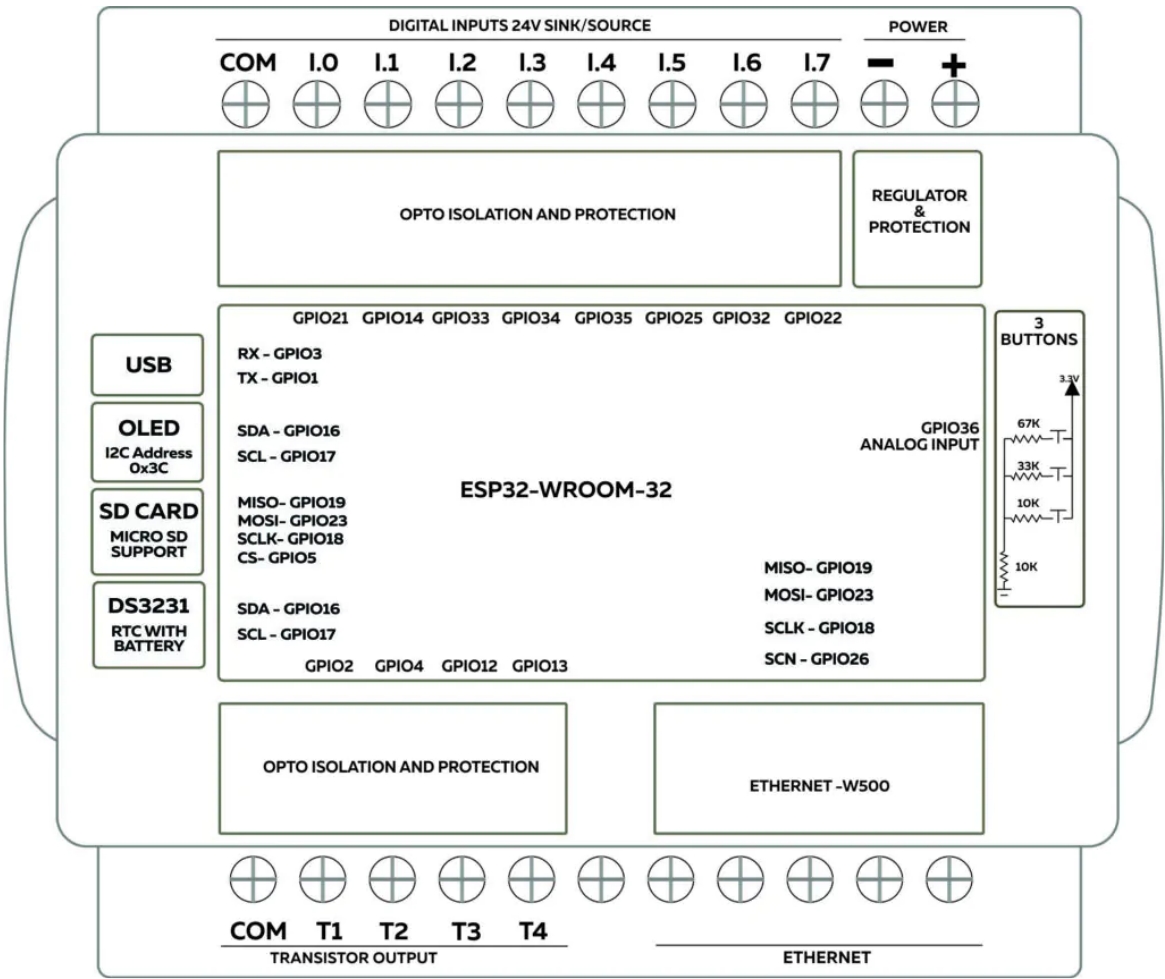


NORVI ENET-AE06-T – FICHA TÉCNICA

Características del Producto



- Módulo ESP32-WROOM32
- Pantalla OLED de 0.96 integrada
- Soporte para Tarjeta Micro SD
- RTC DS3231 con respaldo de batería
- Ethernet a través de W5500
- Botón integrado en el panel frontal
- Entradas Digitales
- Salidas a Transistor
- Montaje en Carril DIN

Expansiones Soportadas

- Salidas de transistor
- Entrada digital
- Salida analógica

Inicio

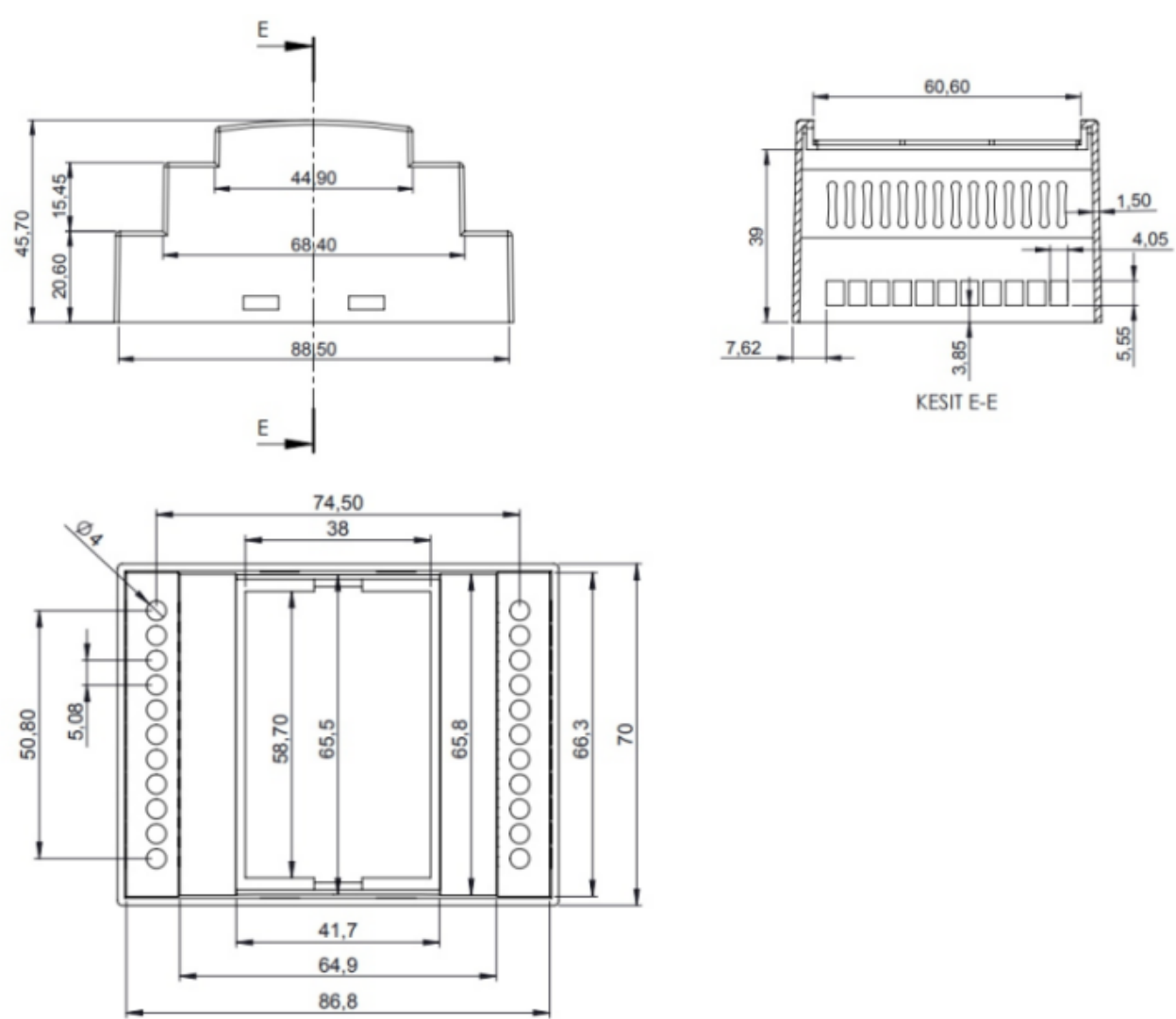
Gama del Producto	NORVI ENET
Tipo de Producto	Controlador Programable
Certificaciones	EN 61131-2:2007 EN 61010-1:2010+A1:2019 EN IEC 61010-2-201:2018 2014/30/EU- Compatibilidad Electromagnética (EMC) Anexo III, Parte B, Módulo C
Tensión de alimentación	24V DC
Comunicación	WiFi / Bluetooth
Entradas y Salidad	8 x Entrada Digital 4 x Salida Transistor
Pantallas e indicadores visuales	0.96 Pantalla OLED Display e Indicadores

Complementarios

Producto Código unificado	NORVI ENET-AE06-T
Números de Pieza de Producto	NORVI ENET-AE06-T

Características Mecánicas

Carcasa	NORVI 204
Montaje / Método de Instalación	RIEL DIN / PLACAS DE MONTAJE
Tipo de Terminal	TERMINAL DE TORNILLOS
Disposición de terminales	Arriba y Abajo
Largo	90.50 mm
Alto	56.60 mm
Ancho	60.60 mm



Entorno

Grado de Protección IP	IP20
Altitud de Operatividad	0–2000 metros
Temperatura de Operación	– –10 ... +85° C (14...185 °F)
Altitud de Almacenaje	0 – 3000 metros
Resistencia a Golpes	15 gn para 11ms
Resistencia a Descargas Electrostáticas	4kV en contacto 8kV en 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 Red

Tensión de Alimentación (V)	24V DC
Consumo 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 Integrados

Botón 1 Pin	GPIO36 Nivel de entrada analógica 1
Botón 2 Pin	GPIO36 Nivel de entrada analógica 2
Botón 3 Pin	GPIO36 ANivel de entrada analógica 3

Compatible con tarjeta micro SD

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

RTC Interno

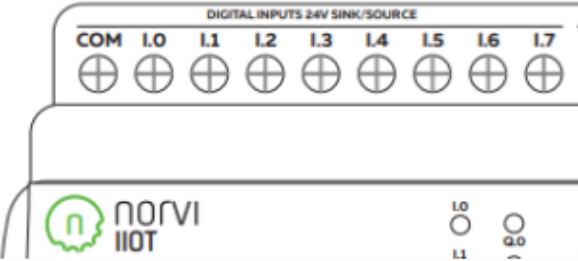
Chip RTC	DS3231
Tipo de Batería de respaldo	CR2032
Interfaz	I2C
Dirección I2C	0x68
Pin SCL	GPIO17
Pin SDA	GPIO16

Pantalla OLED

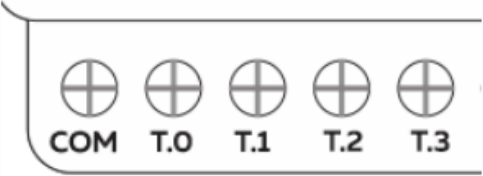
Controlador de Pantalla	SSD1306
Tamaño de Pantalla	0.96 pulgadas
Pin SCL	GPIO17
Pin SDA	GPIO16
RESET Pin	NO CONECTADO

ENTRADAS y SALIDAS

Entradas Digitales

Numero de Entradas Digitales	8
Polaridad de Entrada Digital	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 terminales	Entrada Digital 0 – GPIO21 Entrada Digital 1 – GPIO14 Entrada Digital 2 – GPIO33 Entrada Digital 3 – GPIO34 Entrada Digital 4 – GPIO35 Entrada Digital 5 – GPIO25 Entrada Digital 6 – GPIO32 Entrada Digital 7 – GPIO22 

Salidas Transistor

Numero de Salidas Transistor	4
Tipo de Salida Transistor	COLECTOR ABIERTO
Maximum Sink/Source Current (mA)	100mA
Tensión Máxima Aplicable	36V DC
Frecuencia Máxima de Conmutación	1 kHz
Disposición de Terminales	Salida Transistor 0 – GPIO2 Salida Transistor 1 – GPIO4 Salida Transistor 2 – GPIO12 Salida Transistor 3 – GPIO13 

Canales de Comunicación

Ethernet SPI – W5500

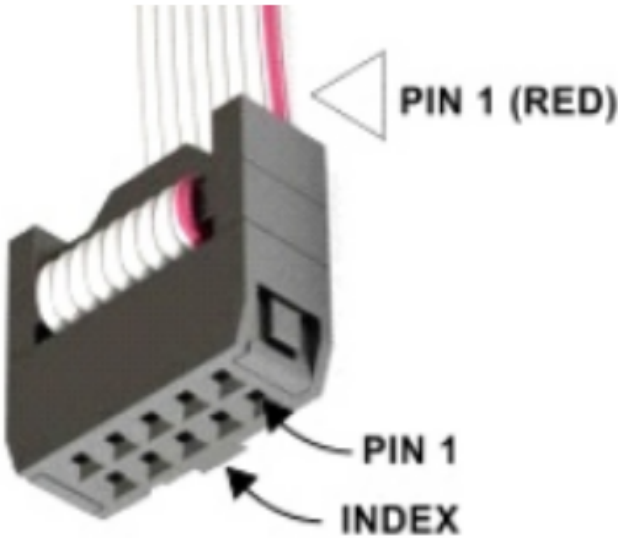
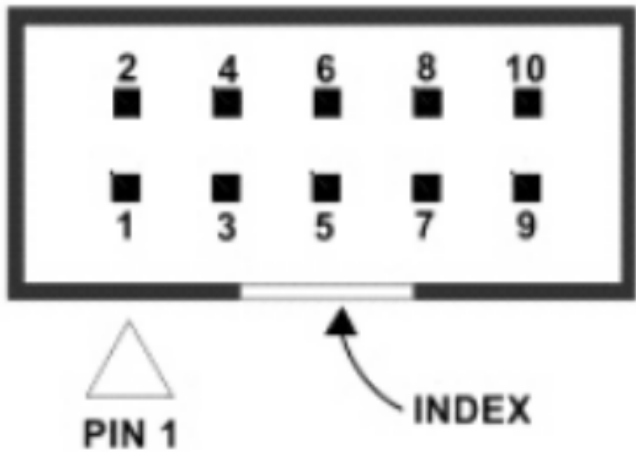
Transceptor	W5500
Velocidad	10BaseT/100BaseTX
Admite Negociación Automática	Si
TX/RX Buffer Size	Internal 32 Kbytes Memory
Protocolos TCP/IP Cableados Compatibles	TCP, UDP, ICMP, IPv4, ARP, IGMP, PPPoE
Número de tomas Independientes Simultáneas	8
SCSn	GPIO26
MISO	GPIO19
MOSI	GPIO23
SCLK	GPIO18
RSTn	GPIO27

Mapa GPIO

GPIO	Descripción	Uso
0	Puerto de Expansión	NRST
1	Puerto de Expansión	TX0
2	connected to on-board LED	Salida Transistor 0
3	Puerto de Expansión	RX0
4	conectado al LED integrado	Salida Transistor 1
5	Tarjeta micro SD	CS
6	conectado a la flash SPI integrada	
7	conectado a la flash SPI integrada	
8	conectado a la flash SPI integrada	
9	conectado a la flash SPI integrada	
10	conectado a la flash SPI integrada	
11	conectado a la flash SPI integrada	
12	conectado al LED integrado	Salida Transistor 2
13	conectado al LED integrado	Salida Transistor 3
14	emite una señal PWM en el arranque	Entrada Digital 1
15	Puerto de Expansion	Pin 6
16	Puerto de Expansión, I2C	SDA
17	Puerto de Expansion, I2C	SCL
18	Tarjeta microSD	SCLK
19	Tarjeta microSD	MISO
20		
21	sólo entrada	Entrada Digital 0
22	sólo entrada	Entrada Digital 7

23	microSD Card	MOSI
24		
25	sólo entrada	Digital Input 5
26	RTC	SCSN
27		RSTn
28		
29		
30		
31		
32	sólo entrada	Digital Input 6
33	sólo entrada	Digital Input 2
34	sólo entrada	Digital Input 3
35	sólo entrada	Digital Input 4
36		Buttons
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-T – GUIA DEL USUARIO

Programación

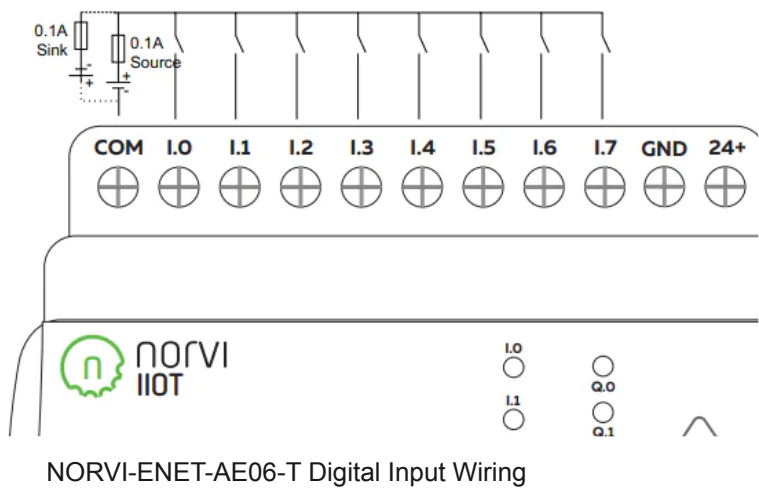
El NORVI ENET-AE06-T tiene un mini puerto USB para la conexión en serie con el SoC para la programación. Se puede utilizar cualquier IDE de programación compatible con ESP32 para programar el controlador. Siga esta Guía para programar controladores NORVI basados en ESP32 con el IDE Arduino.

SoC: ESP32-WROOM32
Puerto de Programación: USB UART

Entradas Digitales

Cableado de Entradas Digitales

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



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

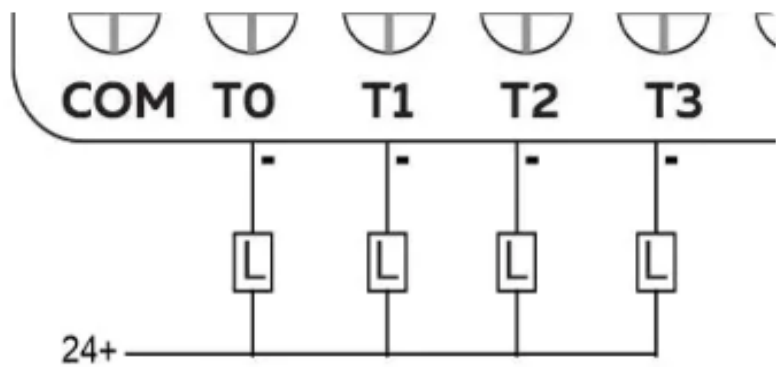
```
#define INPUT1 21

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

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


Salida Transistor

Cableado de Salidas Transistor



NORVI ENET-AE06-T Transistor Output Wiring

Programación de Salidas Transistor

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

Biblioteca respaldada 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
```



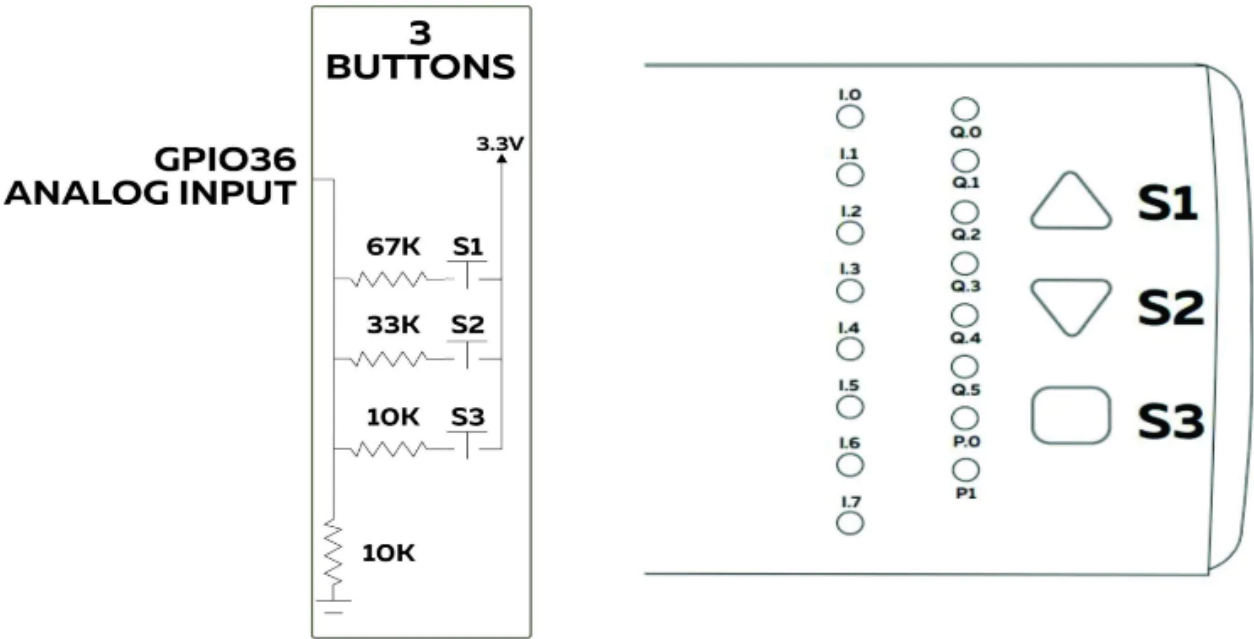
```
// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
#define OLED_RESET -1 // Reset pin # (or -1 if sharing Arduino reset pin) Adafruit_SSD1306 display(SCREEN_WIDTH,
SCREEN_HEIGHT, &Wire, OLED_RESET);
void setup() {
  Wire.begin(16,17);
  if(!display.begin(SSD1306_SWITCHCAPVCC, 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 Integrados

Modo de Lectura	ADC (Conversión Analógica a Digital)
IO Analógico	GPIO36

GPIO Conexión de botones



Botón incorporado esquema interno

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

RTC Chip	DS3231
Backup Battery Type	CR2032
Interface	I2C
I2C Address	0x68
SCL Pin	GPIO17
SDA Pin	GPIO16

Conexiones GPIO del RTC

Programación 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();
}
```

Compatible con Tarjetas Micro SD

TARJETA SD CS	GPIO5
MISO	GPIO19
MOSI	GPIO23
SCLK	GPIO18

Conexiones GPIO de la tarjeta SD

Programación 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

