

SEMINARIO DE PROGRAMACION DE SISTEMAS EMBEBIDOS

UNIVERSIDAD DE GUADALAJARA

PRACTICA 1

EQUIPO:

Razo Pérez Joel Alejandro

Ramirez Figueroa Josue

Donovan Odraym Álvarez Sanches

Marco teórico

Es un microcontrolador potente y versátil que ofrece una serie de características, incluyendo conectividad WiFi y Bluetooth, procesamiento de doble núcleo, y una gama de periféricos. El ESP32 es un microcontrolador de bajo costo, alto rendimiento y bajo consumo de energía desarrollado por Espressif Systems. Se caracteriza por integrar conectividad Wi-Fi y Bluetooth en un solo chip, lo que lo convierte en una solución ideal para proyectos de Internet de las Cosas (IoT), dispositivos inteligentes y aplicaciones industriales.

Algunas características generales:

El ESP32 cuenta con una arquitectura avanzada y recursos que lo hacen versátil para múltiples aplicaciones. Sus características principales incluyen:

Procesador: Integra un procesador dual-core Tensilica Xtensa LX6 con frecuencias de hasta 240 MHz, aunque también está disponible en versiones single-core.

Memoria: Memoria RAM: Hasta 520 KB de SRAM.

Flash: Soporte para memoria externa con capacidades de hasta 16 MB.

Conectividad: Wi-Fi: Compatible con los estándares IEEE 802.11 b/g/n, soportando modos AP, STA y Mesh. Bluetooth: Compatible con Bluetooth v4.2 y BLE (Bluetooth Low Energy).

Entradas y Salidas (GPIOs): Hasta 34 pines GPIO configurables como entradas o salidas digitales. ADC (Conversor Analógico-Digital) de 12 bits con hasta 18 canales. DAC (Conversor Digital-Analógico) de 8 bits.

Protocolos: I2C, SPI, UART, I2S, PWM, entre otros.

Consumo de energía: Diseñado para aplicaciones de bajo consumo, con múltiples modos de operación como "Deep Sleep" y "Light Sleep".



Una estructura del ESP32 para hacernos una idea de como utilizar todos los pines.

PINES DE LA PLACA DE DESARROLLO

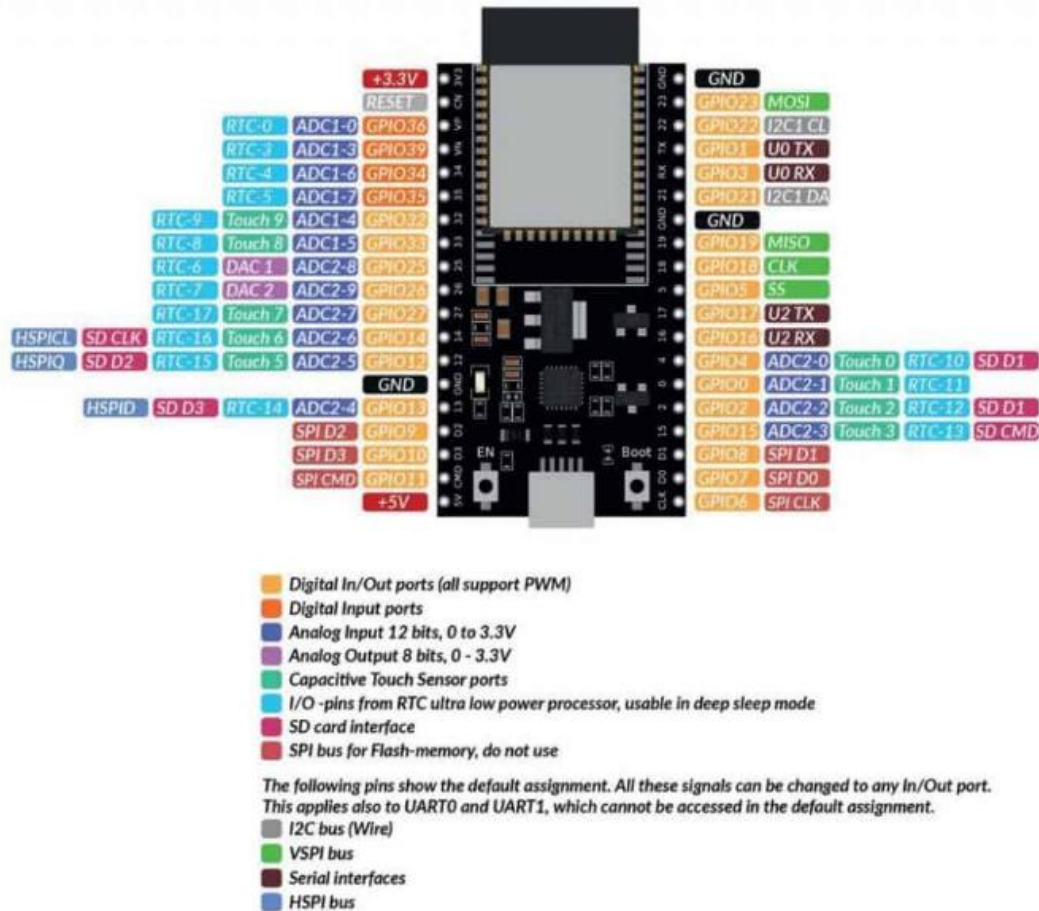


Imagen propiedad de Espressif.



Código:

```
#define LED_PIN 2 // Define el pin donde está conectado el LED (puedes cambiarlo si usas otro pin)
```

```
int delayTime = 1000; // Tiempo inicial de espera (1 segundo)
```

```
bool decreasing = true; // Controla si el tiempo de delay debe decrecer o aumentar
```

```
void setup() {
```

```
    // Configura el pin como salida
```

```
    pinMode(LED_PIN, OUTPUT);
```

```
}
```

```
void loop() {
```

```
    // Enciende el LED
```

```
    digitalWrite(LED_PIN, HIGH);
```

```
    delay(delayTime); // Espera según el tiempo actual
```

```
    // Apaga el LED
```

```
    digitalWrite(LED_PIN, LOW);
```

```
    delay(delayTime); // Espera según el tiempo actual
```

```
    // Cambia el ritmo de parpadeo
```

```
    if (decreasing) {
```

```
        delayTime -= 100; // Reduce el tiempo de delay (parpadeo más rápido)
```

```
        if (delayTime <= 100) {
```

```
            decreasing = false; // Cuando llega a 100ms, empieza a aumentar el tiempo
```

```
        }
```

```
    } else {
```

```
        delayTime += 100; // Aumenta el tiempo de delay (parpadeo más lento)
```



```
if (delayTime >= 1000) {  
    decreasing = true; // Cuando llega a 1000ms, empieza a disminuir el tiempo  
}  
}  
}
```

El LED comienza parpadeando lentamente (1 segundo de encendido y 1 segundo de apagado). Gradualmente, el parpadeo se acelera, disminuyendo el tiempo de espera hasta 100 ms. Luego, el parpadeo desacelera, aumentando el tiempo de espera nuevamente hasta 1 segundo. Este ciclo se repite infinitamente, creando un efecto de aceleración y desaceleración en el parpadeo del LED.

Conclusiones:

El ESP32 es muy versátil cuando se trata de prácticas, al momento de programar nos hacemos una idea de cuán “fácil”, entre comillas puede llegar a ser, te acoplas bien a su método de uso y si sigues las instrucciones de la base de datos es mucho más sencillo.

Es un microcontrolador versátil y poderoso, ideal para aplicaciones modernas que requieren conectividad inalámbrica eficiente. Su integración de Wi-Fi y Bluetooth, junto con una amplia variedad de periféricos y capacidad de procesamiento.

Su bajo costo, alta capacidad de personalización y soporte para plataformas como Arduino y MicroPython facilitan su adopción por principiantes y expertos. Aunque puede ser más demandante en términos de consumo energético y requerimientos de programación, su flexibilidad lo convierte en una herramienta fundamental para proyectos innovadores y de alto impacto tecnológico, destaca como una solución confiable, económica y robusta para enfrentar los desafíos tecnológicos actuales y futuros, en muchos casos solemos hacernos una idea de lo que nos puede deparar el futuro pero también necesitamos adaptarnos conforme este vaya llegando.



EVIDENCNIAS FOTOGRAFICAS:



References

Morales, J. A. (2025). *pasionelectronica*. Retrieved from <https://pasionelectronica.com/esp32-caracteristicas-y-pines/>

