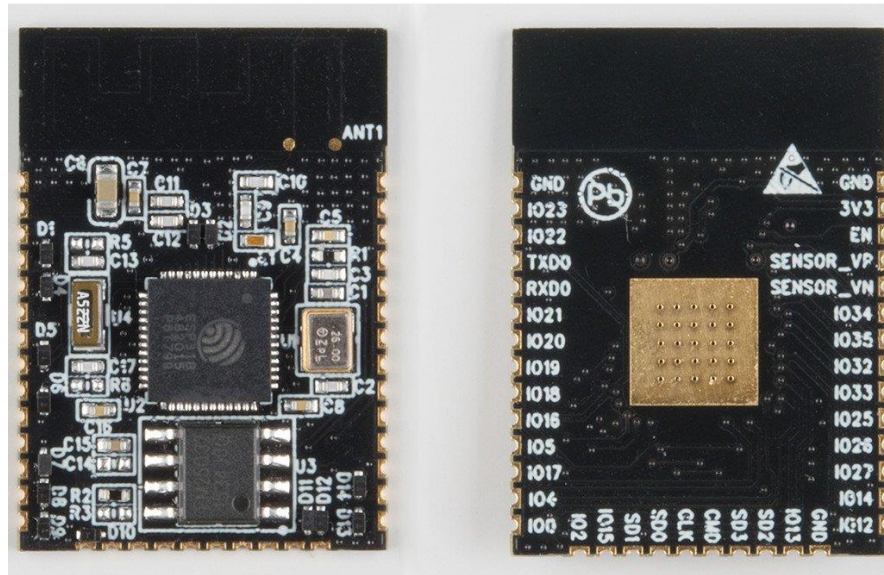


TALLER LOW POWER DESIGN CON ESP32

- **Instructor: Juan David Rosadio Vega**



ESP32-WROOM-32 Specifications

Categories	Items	Specifications
Hardware	Module interfaces	SD card, UART, SPI, SDIO, I ² C, LED PWM, Motor PWM, I ² S, IR, pulse counter, GPIO, capacitive touch sensor, ADC, DAC, Two-Wire Automotive Interface (TWAI [®] , compatible with ISO11898-1)
	On-chip sensor	Hall sensor
	Integrated crystal	40 MHz crystal
	Integrated SPI flash	4 MB
	Operating voltage/Power supply	3.0 V ~ 3.6 V
	Operating current	Average: 80 mA
	Minimum current delivered by power supply	500 mA
	Recommended operating temperature range	-40 °C ~ +85 °C
	Package size	(18.00±0.10) mm × (25.50±0.10) mm × (3.10±0.10) mm
	Moisture sensitivity level (MSL)	Level 3

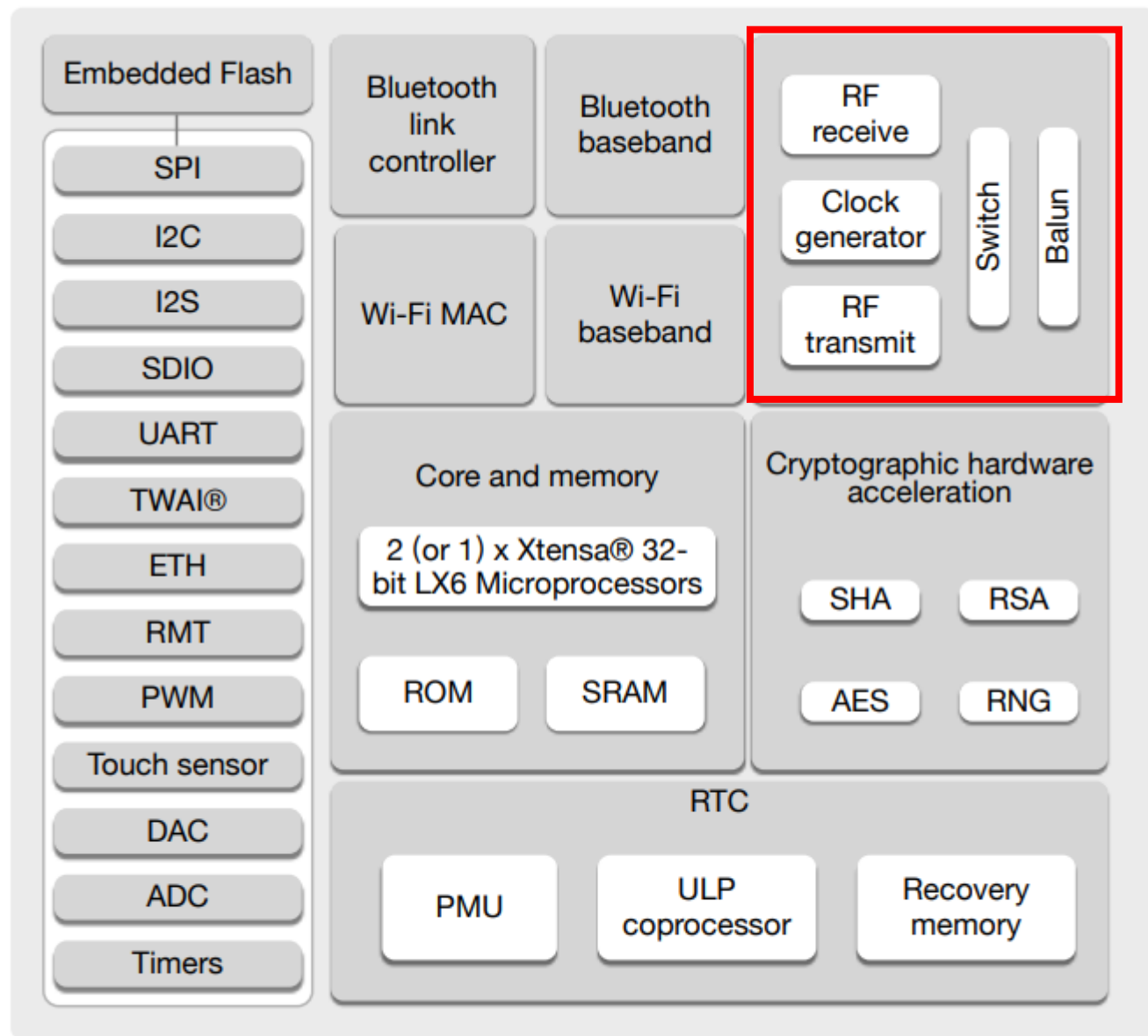


Figure 1: Functional Block Diagram

Las Reglas del Diseño de bajo consumo

USO DE LOS MODOS STANDBY

- **Modem-sleep mode:**

Activo: CPU

Inactivo: Wifi, Bluetooth, Radio.

Funcionamiento: Automatico (gestionado por la CPU)

El consumo típico en este modo es de 15mA.

- **Light-sleep modo:**

Activo: RTC memory, RTC peripherals, Coprocesador ULP

Pausado: CPU

Inactivo: Wifi, Bluetooth, Radio.

El consumo típico pasa a ser de unos 0,5 mA.

- **Deep-sleep mode:**

Activo: RTC memory, RTC peripherals, Coprocesador ULP

Inactivo: CPU, Wifi, Bluetooth, Radio, Peripherals .

El consumo típico pasa a ser de unos 10 uA.

- **Hibernation Mode:**

Activo: RTC peripherals

Inactivo: CPU, Coprocesador ULP, Oscillator internal 8-MHz,
RTC memory, Wifi, Bluetooth, Radio, Peripherals.

El consumo típico pasa a ser de unos 5 uA.

```
//Función en Arduino  
esp_light_sleep_start();
```

```
//Función en Arduino  
esp_deep_sleep_start();
```

```
1
2 #define LED 2
3
4 void setup() {
5     // Set pin mode
6     pinMode(LED, OUTPUT);
7
8 }
9
10 void loop() {
11
12     for(int i=0; i<5;i++){
13         digitalWrite(LED, HIGH);
14         delay(500);
15         digitalWrite(LED, LOW);
16         delay(500);
17     }
18
19     //Despertar por temporizador
20     esp_sleep_enable_timer_wakeup (3*1000000); //En us
21
22     //Inicia el modo Light-Sleep
23     esp_light_sleep_start();
24
25     //Inicia el modo Deep-Sleep
26     esp_deep_sleep_start();
27 }
```

FRECUENCIA DE FUNCIONAMIENTO

La reducción de la frecuencia de reloj del sistema a la mínima disponible reducirá la corriente de funcionamiento del dispositivo.

Power mode	Description			Power consumption
Active (RF working)	Wi-Fi Tx packet			Please refer to Table 15 for details.
	Wi-Fi/BT Tx packet			
	Wi-Fi/BT Rx and listening			
Modem-sleep	The CPU is powered on.	240 MHz [*]	Dual-core chip(s)	30 mA ~ 68 mA
			Single-core chip(s)	N/A
		160 MHz [*]	Dual-core chip(s)	27 mA ~ 44 mA
			Single-core chip(s)	27 mA ~ 34 mA
		Normal speed: 80 MHz	Dual-core chip(s)	20 mA ~ 31 mA
			Single-core chip(s)	20 mA ~ 25 mA
Light-sleep	-			0.8 mA
Deep-sleep	The ULP co-processor is powered on.			150 μ A
	ULP sensor-monitored pattern			100 μ A @1% duty
	RTC timer + RTC memory			10 μ A
Hibernation	RTC timer only			5 μ A
Power off	CHIP_PU is set to low level, the chip is powered off.			1 μ A

OPERACIÓN DE LOS PERIFERICOS

Los periféricos que no se utilicen deben estar apagados para evitar desperdiciar energía. Pero se debe tener en cuenta:

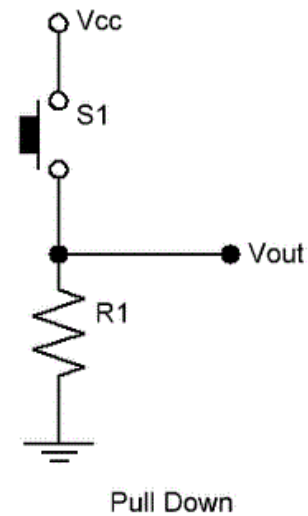
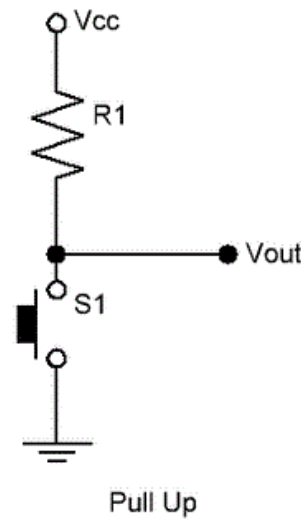
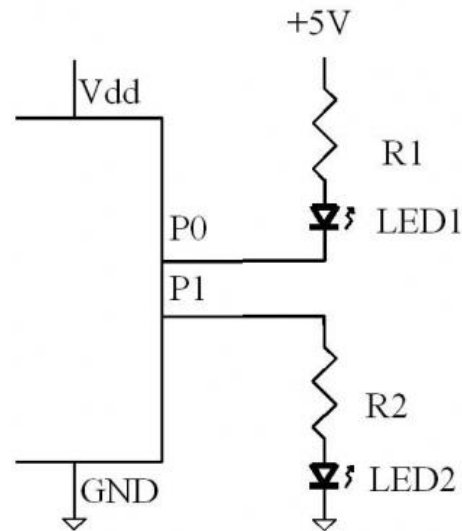
- El apagado debe planificarse con cuidado ya que deshabilitar algunos periféricos puede restablecer sus ajustes de configuración.
- Cualquier función "analógica" consumirá más energía que la mayoría de las funciones digitales, ya que éstas incluyen estática (resistencias, referencias, etc.)

//Función en Arduino

```
esp_sleep_pd_config(ESP_PD_DOMAIN_RTC_PERIPH, ESP_PD_OPTION_ON);
```

CONFIGURACION DE LOS PINES I/O

- Evitar usar salidas digitales en accionamientos de baja impedancia.
- Evitar dejar pines flotantes cuando se configura los pines como entrada digital
- Evitar las resistencias de pull up o pull down (internas o externas), pero si se requiere su uso, usar valores de resistencias altos.
- Cualquier pin de I/O **no utilizado** puede ser configurado como una "salida", ya que esto siempre establecerá un nivel definido ayudando con la inmunidad al ruido del mundo exterior.



CORRECTA ELECCION DE LA FUENTE DE ALIMENTACION

ESP32 Board	Voltage Regulator	Maximum Voltage Dropout	Quiescent Current
ESP32 – DevKitC	AMS1117	1.1V @ 800mA	5mA
Ai-Thinker NodeMCU-32S	AMS1117	1.1V @ 800mA	5mA
Adafruit HUZZAH32	AP2112-3.3	0.4V @ 600mA	80µA
Sparkfun ESP32 Thing	AP2112-3.3	0.4V @ 600mA	80µA
FireBeetle ESP32	RT9080-33GJ5	0.31V @ 600mA	4µA

REFERENCIAS

ESP32 Datasheet

https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

ESP32 WROOM32

https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf

ESPRESSIF Programming Guide – Sleep Modes

https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/sleep_modes.html#

The background is a solid teal color. Overlaid on this are several abstract, semi-transparent shapes. A large, irregular, rounded rectangle is centered, tilted slightly. Within and around this rectangle are several circles of varying sizes, some of which are also tilted. The overall effect is a modern, minimalist design.

APRENDE EN
UMAKER