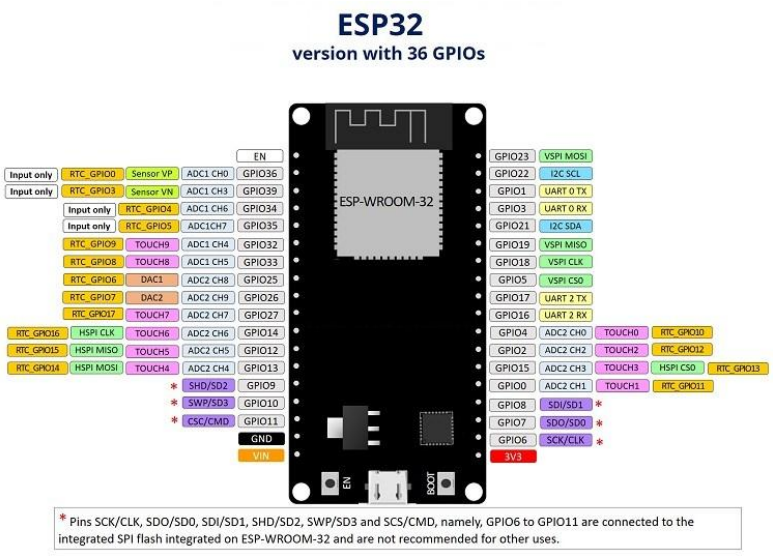
IoT (Internet of Things)

ESP32 y sueño profundo

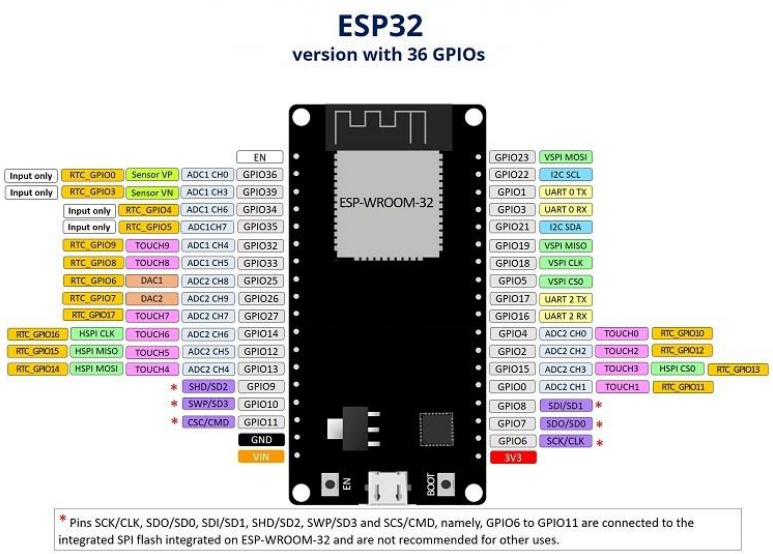
Departamento de Ingeniería en Sistemas y Computación
Universidad Católica del Norte, Antofagasta.

Procesamiento ESP32



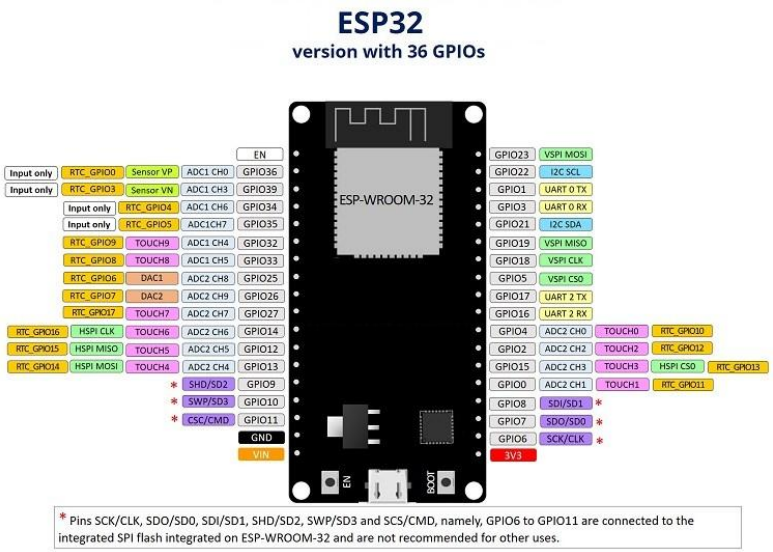
- sucesor de ESP8266
- procesador dual core de 32 bits
- reloj entre 160 y 240 mhz
- 520kb de RAM

Comunicación ESP32



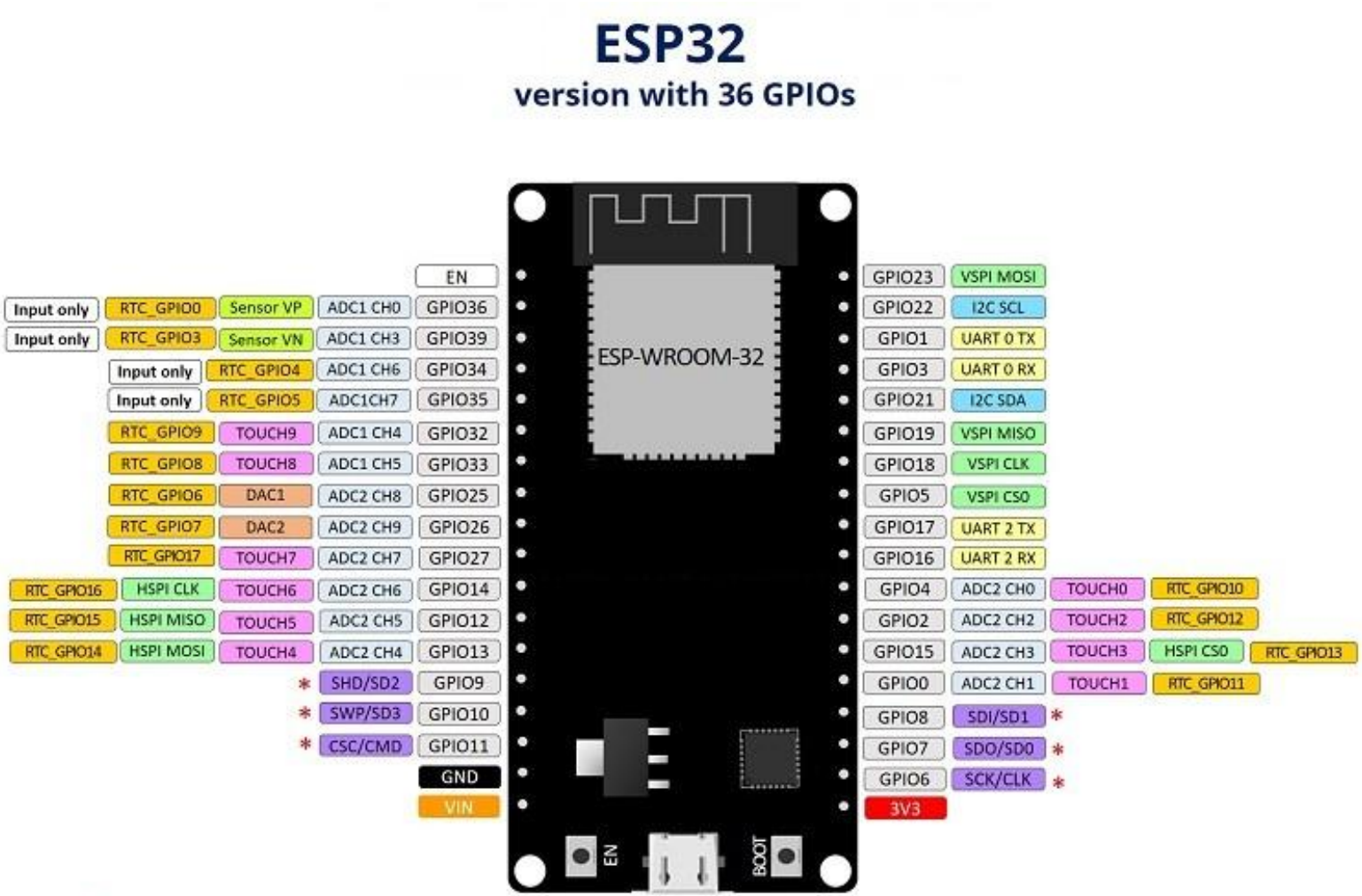
- WiFi integrado (Access point y Estación)
- Bluetooth 4.2, BT 2.0 y BLE 4.0
- 2 puertos seriales UART
- 2 puertos I^2C
- 4 puertos SPI

Pines ESP32



- 36 GPIO pins
- 16 ADC con 12 bits de precisión
- 2 DAC de 8 bits

Pines ESP32



* Pins SCK/CLK, SDO/SD0, SDI/SD1, SHD/SD2, SWP/SD3 and SCS/CMD, namely, GPIO6 to GPIO11 are connected to the integrated SPI flash integrated on ESP-WROOM-32 and are not recommended for other uses.

ESP32 en Arduino IDE

con Administrador de placas:

- en `File→Preferences`, incluir en Additional Boards Manager URL:
  `https://dl.espressif.com/dl/package_esp32_index.json`

- luego con el Administrador de placas (en `Sketch→Include Library→Manage Libraries`, buscar **ESP32 by Espressif Systems**

ESP32 en Arduino IDE

Instalación manual:

- crear carpeta `hardware/espressif` en la ruta de los sketchbooks (revisar `Sketchbook location` en `File→Preferences`)
- en la carpeta recién creada, clonar el repositorio de github: `git clone https://github.com/espressif/arduino-esp32.git esp32`
- en subcarpeta `esp32`, actualizar con `git submodule update -init -recursive`
- abrir carpeta `tools` y ejecutar `get`
  - Windows: `get.exe`
  - Linux y Mac OS: `python get.py`

Sensor de tacto

- la ESP32 cuenta con 10 sensores de tacto (T0, T1, ..., T9)
- `touchRead(pin)` toma como entrada el pin de tacto (p.ej. T0 en lugar de GPIO4)

Actividad

- Escoja 3 pines disponibles como sensor de tacto, conecte cada uno de estos a una superficie conductora distinta, y asocie cada una de estas superficies con un color distinto que deberá encender en un led RGB.

Sueño profundo

Extremadamente útil para un consumo energético eficiente:

- consumo promedio en operación normal: **75 mA**
- consumo promedio en transmisión por WiFi: **240 mA**
- consumo promedio en sueño profundo: **10μA**

Sueño profundo

En sueño profundo:

- cpu y memoria principal quedan inactivas
- se limpian todos los datos, excepto aquellos guardados en el módulo RTC (Real Time Clock)
  p.ej. `RTC_DATA_ATTR int contador = 0;`
- inicio de sueño profundo: `esp_deep_sleep_start();`

Sueño profundo

Es posible despertar desde:

- Temporizador
- Sensores de tacto
- Eventos externos (ext0 y ext1)

Despertar

Identificación de evento:

```
esp_sleep_wakeup_cause_t razon;
razon = esp_sleep_get_wakeup_cause();
```

- 1: señal externa usando RTC_IO
- 2: señal externa usando RTC_CNTL
- 3: temporizador
- 4: tacto
- 5: programa ULP (Ultra Low Power)

Despertar con temporizador

Antes de dormir:

```
esp_sleep_enable_timer_wakeup(tiempo);
```

- `tiempo` debe estar especificado en microsegundos
- luego se debe ejecutar la instrucción para dormir (`esp_deep_sleep_start();`)

Despertar con sensor de tacto

Antes de dormir:

```
touchAttachInterrupt(T3, funcion, umbral);
esp_sleep_enable_touchpad_wakeup();
```

- `funcion` es la rutina que se ejecuta al despertar cuando el sensor detecta un valor bajo `umbral`
- luego se debe ejecutar la instrucción para dormir (`esp_deep_sleep_start();`)
- `esp_sleep_get_touchpad_wakeup_status();` devuelve el número del pin activado

Eventos externos

Existen dos tipos de eventos gatillantes:

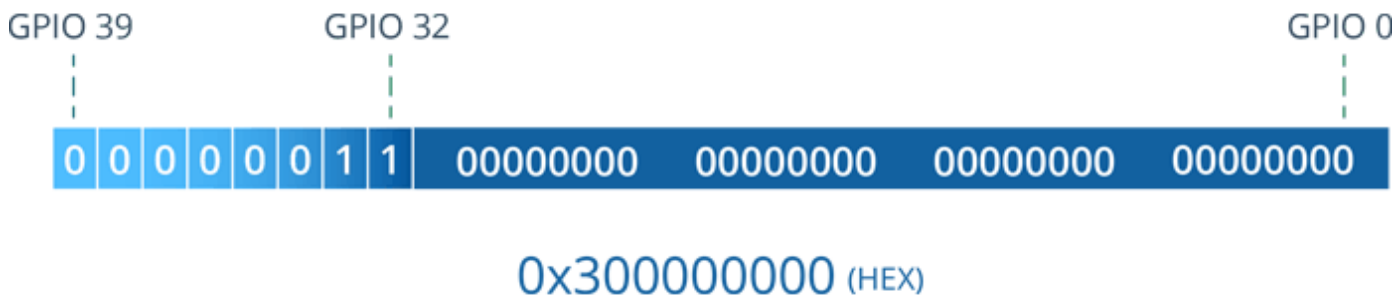
- **ext0**: usado cuando se quiere despertar solo desde **un** pin particular. Se puede usar cualquier pin RTC en GPIO (0-2, 4, 12-15, 25-27, 32-39)
- **ext1**: usado cuando se habilitan múltiples pines para despertar. Se pueden usar solo los pines GPIO32 a GPIO39

ext0

- se debe configurar pin: `esp_sleep_enable_ext0_wakeup(GPIO_NUM_33,1);`, donde el último argumento indica si despertará con valor lógico alto (1) o bajo (0).

ext1

- pines se configuran según máscara (en hexadecimal):
  `esp_sleep_enable_ext1_wakeup(0x300000000, ESP_EXT1_WAKEUP_ANY_HIGH);`
  (también es posible especificar `ESP_EXT1_WAKEUP_ALL_LOW`)



Actividad

Implemente un programa que mida la intensidad de luz del entorno, **promediando los valores obtenidos en 1 segundo**, y que al identificar 5 segundos con el mismo valor promedio se ponga automáticamente a dormir. El programa despertará nuevamente al activarse un sensor de tacto, un pin específico o al pasar 10 segundos.

Se debe mostrar por el monitor serial cuantas veces se ha dormido/despertado, indicando cada vez la causa (temporizador, tacto o un pin especifico).