

# Ejemplos

---

Los ejemplos incluidos son:

## Asembler de ARM Cortex M4

- `asm_lpc_open`: assembler con biblioteca LPC Open.
- `asm_sapi`: assembler con biblioteca sAPI.

## Lenguaje C

- **sAPI library**
  - Bare metal:
    - `adc_dac`: ADC y DAC.
    - `bluetooth`
      - `hm10_uart_bridge`: Transmisión y recepción con módulo Bluetooth HM10 por comandos AT.
    - `cycles_counter`: Funciones para conteo de ciclos de clocks, solo funcionan en modo *debug*. Permite realizar trazabilidad sobre el tiempo de ejecución del programa.
    - `dht11_temp_humidity`: Sensor de humedad y temperatura DHT11.
    - `gpio`
      - `gpio_01_switches_leds`: cada botón activa el led sobre el mismo.
      - `gpio_02_blinky`: led destellando con retardo bloqueante.
      - `gpio_03_blinky_switch`: led destellando con retardo no bloqueante que permite controlar otro led con un switch al mismo tiempo.
      - `gpio_04_led_sequences`: secuencia de leds usando retardo no bloqueante.
    - `hcsr04_ultrasonic_sensor`: Sensor ultrasónico de distancia HC-SR04.
    - `i2c`:
      - IMUs
        - `mpu9250`: MPU9250 IMU 9 grados de libertad (Giróscopo, Acelerómetro y Magnetómetro) + Temperatura.
      - `magnetometers`. El módulo Chino GY-273 puede estar soldado uno de estos 2 magnetómetros, que tiene el mismo pinout pero diferente mapa de registros. Para diferenciarlos se debe ignorar la serigrafía de la placa y revisar el código del chip.
        - `hmc5883l`: Magnetómetro HMC5883L.
        - `qmc5883l`: Magnetómetro QMC5883L.
    - `keypad_7segment`: Teclado matricial 4x4 y display 7 segmentos.
    - `lcd`: Display LCD (por GPIO o I2C).
    - `pwm`
      - `pwm_01`: PWM aplicado a dimerizado de LEDs.
      - `pwm_02_rgb_controller_uart`: Led RGB.
    - `rtc`: Periférico RTC para tener fecha hora.

- `sapi_basic_example`: ejemplo básico con biblioteca sAPI.
  - `servo`: Control de servomotor angular con PWM.
  - `spi_sd_card`
    - `spi_sd_card_01_wite_file`: Escribir archivo de texto en una tarjeta SD/MicroSD (conectada por SPI) utilizando un sistema de archivos FAT (biblioteca Elm-Chan FAT FS).
    - `spi_sd_card_02_log_time_stamp`: Escribir archivo de texto en una tarjeta SD/MicroSD (conectada por SPI) utilizando un sistema de archivos FAT (biblioteca Elm-Chan FAT FS) con estampa de tiempo (*time-stamp*) en las muestras.
    - `spi_sd_card_03_list_files`: Listar los archivos y carpetas coontenidos en una tarjeta SD/MicroSD (conectada por SPI) utilizando un sistema de archivos FAT (biblioteca Elm-Chan FAT FS). Muestra el resultado por UART.
  - `tick_hook`: Función de ejecución periódica (basado en interrupciones) con un callback periódico.
  - `uart`
    - `uart_01_echo`: eco por UART, es decir, la placa responde lo mismo que le envió desde la PC.
    - `uart_02_receive_string_blocking`: Espera hasta recibir cierto String patrón en una UART o expire por *timeout* (código bloqueante). No graba el String recibido.
    - `uart_03_receive_string`: Espera hasta recibir cierto String patrón en una UART o expire por *timeout* (código bloqueante). No grava el String recibido.
  - `wifi_esp8266`
    - `wifi_esp8266_01_uart_bridge`: use thais to send AT commands directly to ESP01 module.
    - `wifi_esp8266_02_thingspeak`: send data to thingspeak dashboards.
  - `usb`
    - `usb_device_01_hid_keyboard`: USB lado Device. Placa como Teclado USB.
    - `usb_device_02_cdc_virtualSerialPort`: USB Device. Utilización de USB OTG como puerto serie virtual.
  - Operating Systems:
    - `rtos_cooperative` (see M.J. Pont's book at [https://www.safetty.net/publications/ptte\\_s](https://www.safetty.net/publications/ptte_s))
      - `scheduler_01_seos`: primer sistema operativo cooperativo.
      - `scheduler_02_seos_background_foreground`: segundo sistema operativo cooperativo.
      - `seos_pont_2014_01_first_app`: tercer sistema operativo cooperativo.
    - `rtos_freertos`
      - `dynamic_mem`
        - `freRTOS_01_blinky`
        - `freRTOS_03_ChanFatFS_SPI_SdCard_ADC_log`: logueo de ADC en una SD/MicroSD Card (SPI an using a FAT File System (ChanFS), freeRTOS and sAPI).
      - `static_mem`
        - `freRTOS_01_blinky`
- **LPC Open library**

- bare\_metal
  - blinky
  - blinky\_input
  - blinky\_ram
  - blinky\_rit
  - boot
  - i2c
  - lpc\_open\_basic\_example
  - mpu
  - pwm
- operating\_systems
  - freertos\_blinky
- statecharts
  - statechart
- newlib\_printf\_scanf
- without\_libs

## Lenguaje C++

- cpp\_sapi\_basic\_example: ejemplo básico de C++ con biblioteca SAPI.

## Más información

---

- [Volver al README.](#)