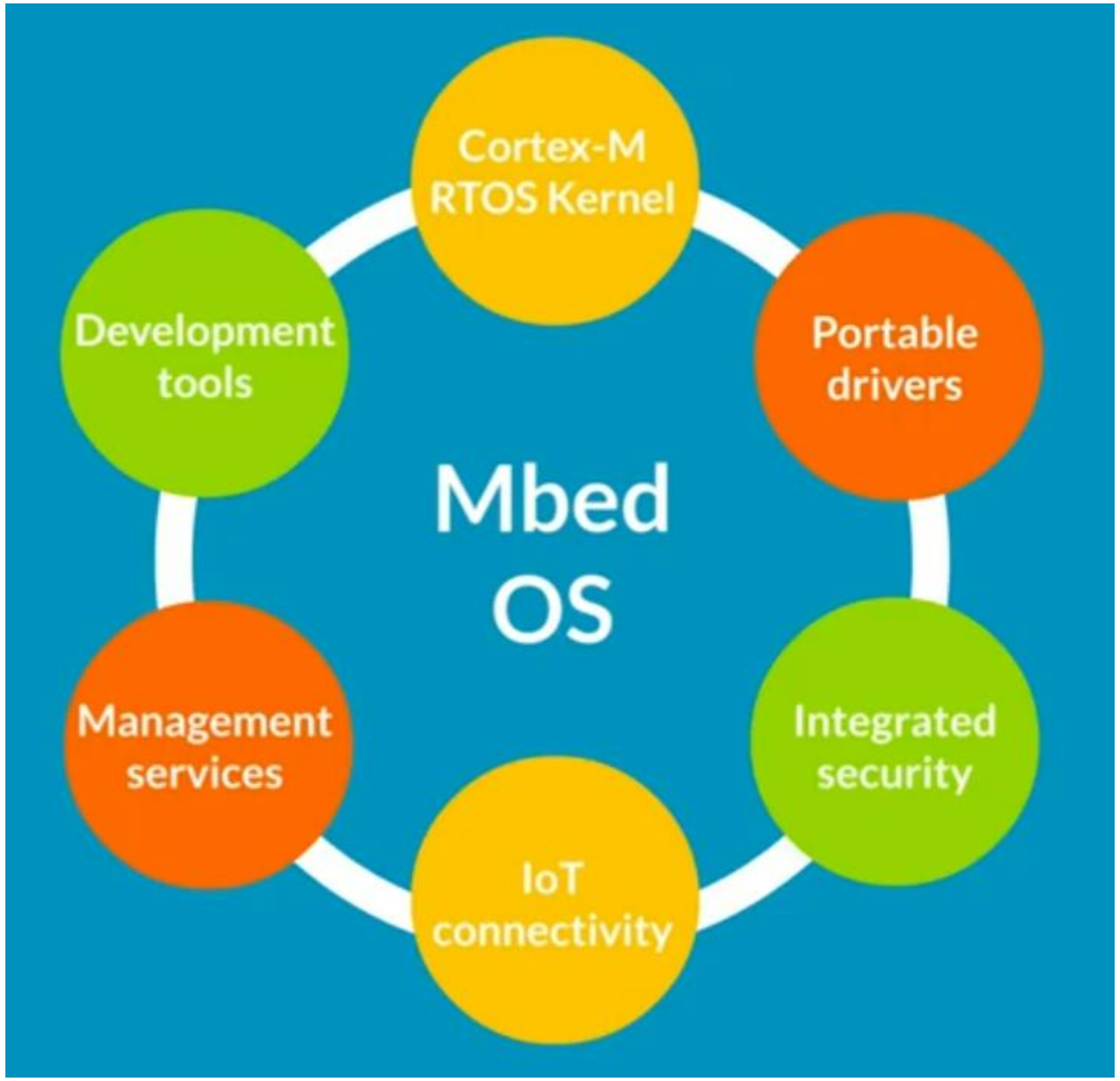
Mbed OS
The Things Network Madrid

Juan Félix Mateos
Octubre 2021
juanfelixmateos@gmail.com

¿Qué es un RTOS?

- Sistema operativo de tiempo real
- Capa de abstracción entre el hardware y el programador
- Determinista: Cada operación tiene un tiempo fijo asignado para ejecutarse (o fallar).
- Conceptos:
  - Semáforos
  - Locks/Mutexes
  - Multi-Treading



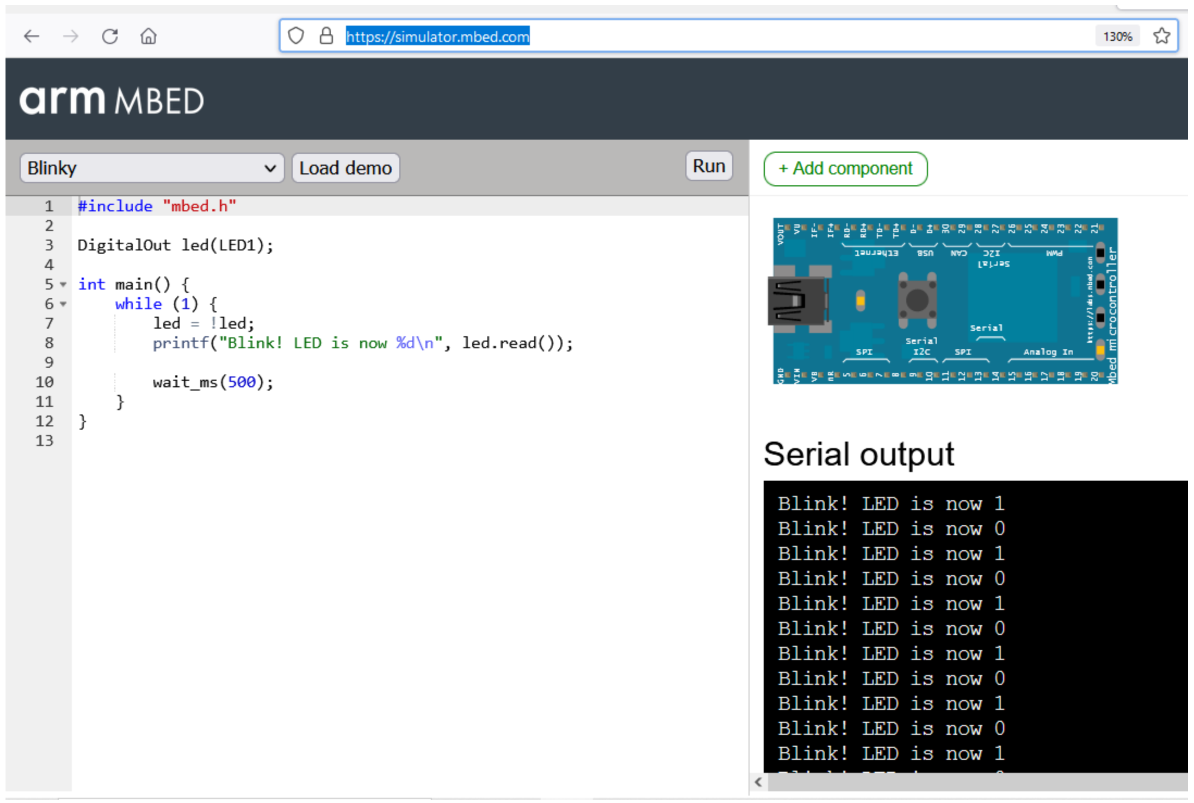
Mbed simulator

https://simulator.mbed.com/

Experimental

Mbed OS 5

Si falla, insistir pulsando nuevamente el botón download que hay a la derecha de Add component



```
#include "mbed.h"

DigitalOut led(LED1);

int main() {
    while (1) {
        led = !led;
        printf("Blink! LED is now %d\n", led.read());

        wait_ms(500);
    }
}

```

```
Blink! LED is now 1
Blink! LED is now 0
Blink! LED is now 1
Blink! LED is now 0
Blink! LED is now 1
Blink! LED is now 0
Blink! LED is now 1
Blink! LED is now 0
Blink! LED is now 1
Blink! LED is now 0
Blink! LED is now 1
```

GPIO

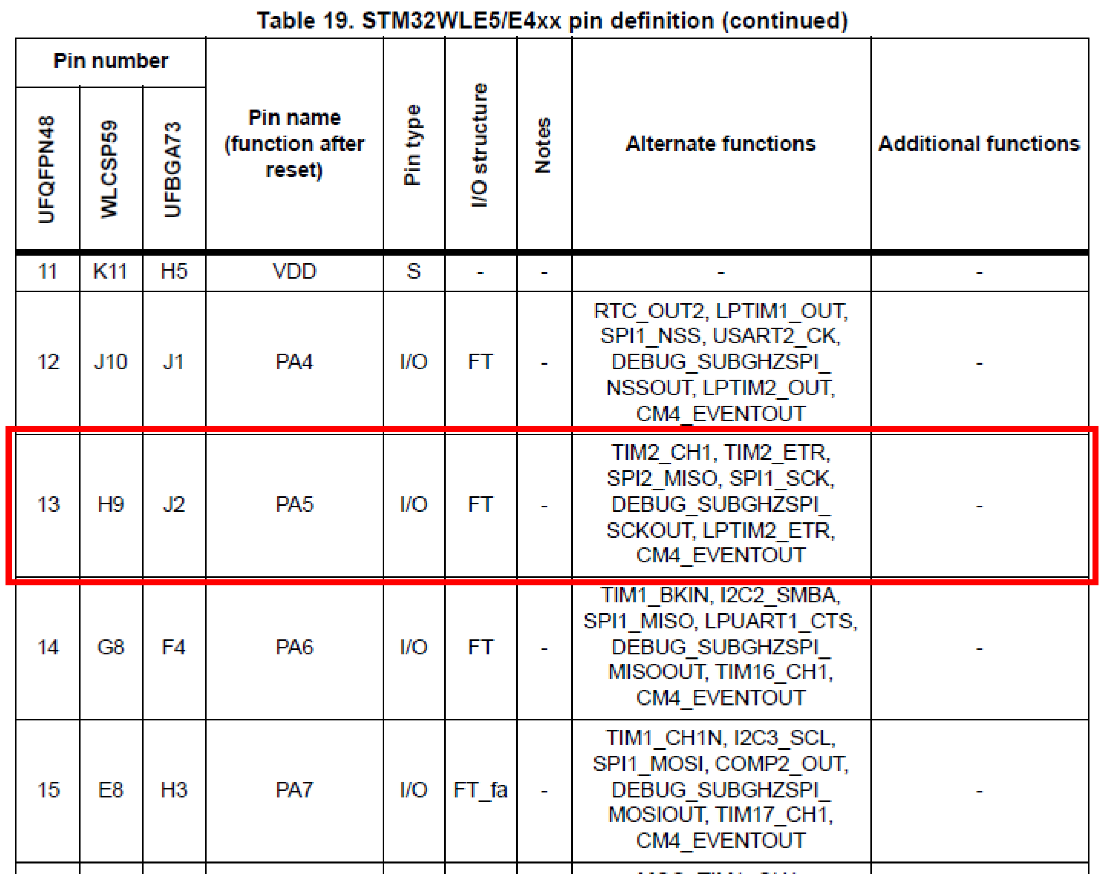
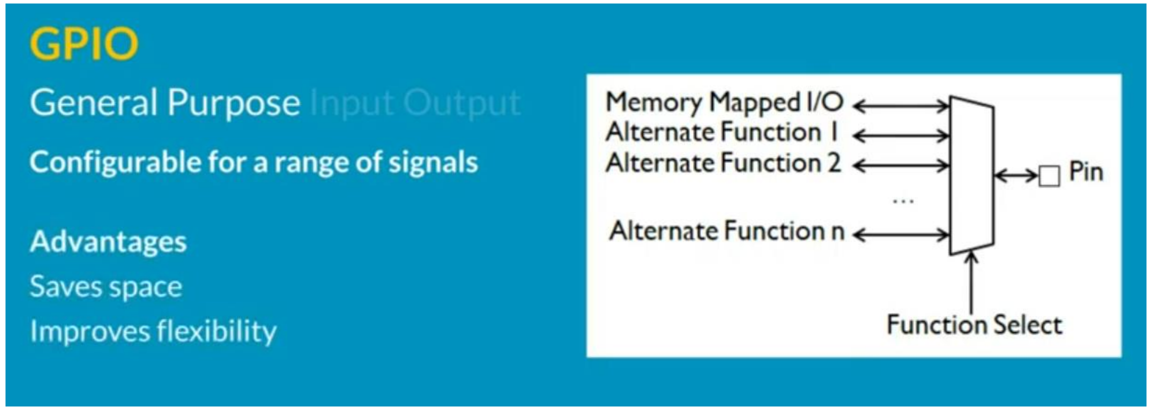


Table 19. STM32WLE5/E4xx pin definition (continued)

Pin number			Pin name (function after reset)	Pin type	I/O structure	Notes	Alternate functions	Additional functions
UFQFPN48	WLCSP59	UFBGA73						
11	K11	H5	VDD	S	-	-	-	-
12	J10	J1	PA4	I/O	FT	-	RTC_OUT2, LPTIM1_OUT, SPI1_NSS, USART2_CK, DEBUG_SUBGHZSPI_ NSSOUT, LPTIM2_OUT, CM4_EVENTOUT	-
13	H9	J2	PA5	I/O	FT	-	TIM2_CH1, TIM2_ETR, SPI2_MISO, SPI1_SCK, DEBUG_SUBGHZSPI_ SCKOUT, LPTIM2_ETR, CM4_EVENTOUT	-
14	G8	F4	PA6	I/O	FT	-	TIM1_BKIN, I2C2_SMBA, SPI1_MISO, LPUART1_CTS, DEBUG_SUBGHZSPI_ MISOOUT, TIM16_CH1, CM4_EVENTOUT	-
15	E8	H3	PA7	I/O	FT_fa	-	TIM1_CH1N, I2C3_SCL, SPI1_MOSI, COMP2_OUT, DEBUG_SUBGHZSPI_ MOSIOUT, TIM17_CH1, CM4_EVENTOUT	-



GPIO

General Purpose Input Output

Configurable for a range of signals

Advantages

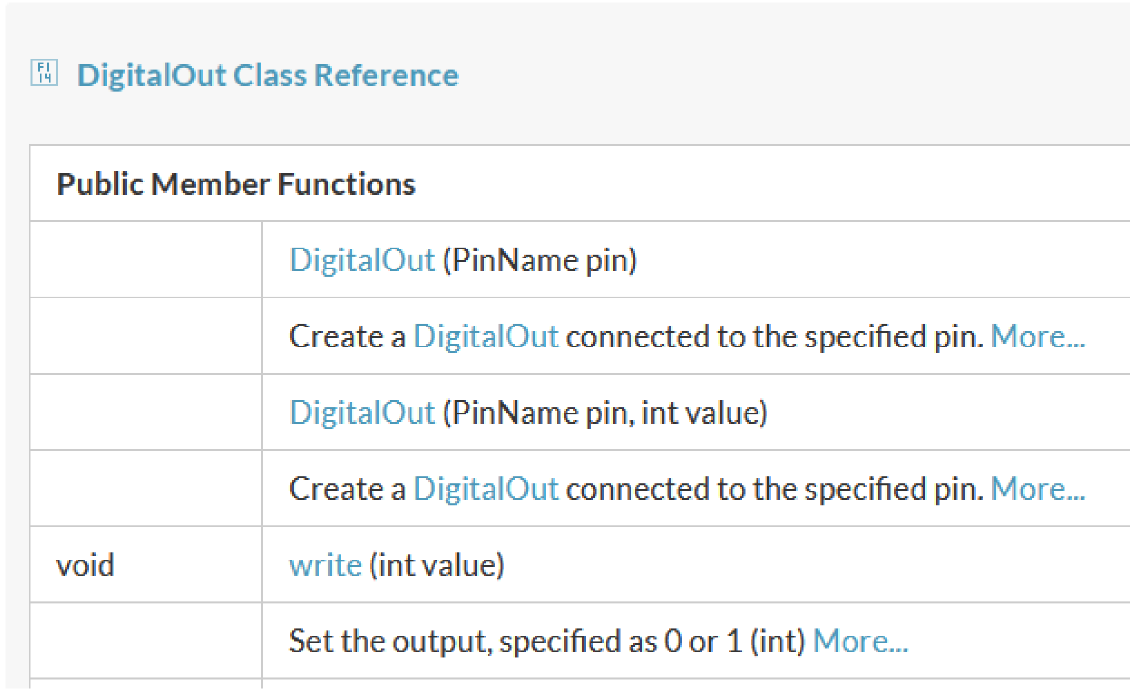
Saves space

Improves flexibility

API: Clases y métodos

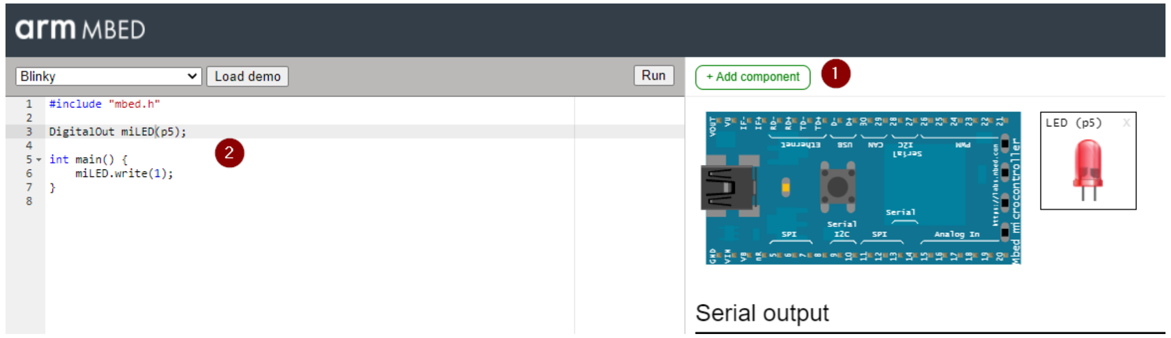
https://os.mbed.com/docs/mbed-os/v6.15/apis/digitalout.html

DigitalOut class reference



DigitalOut Class Reference

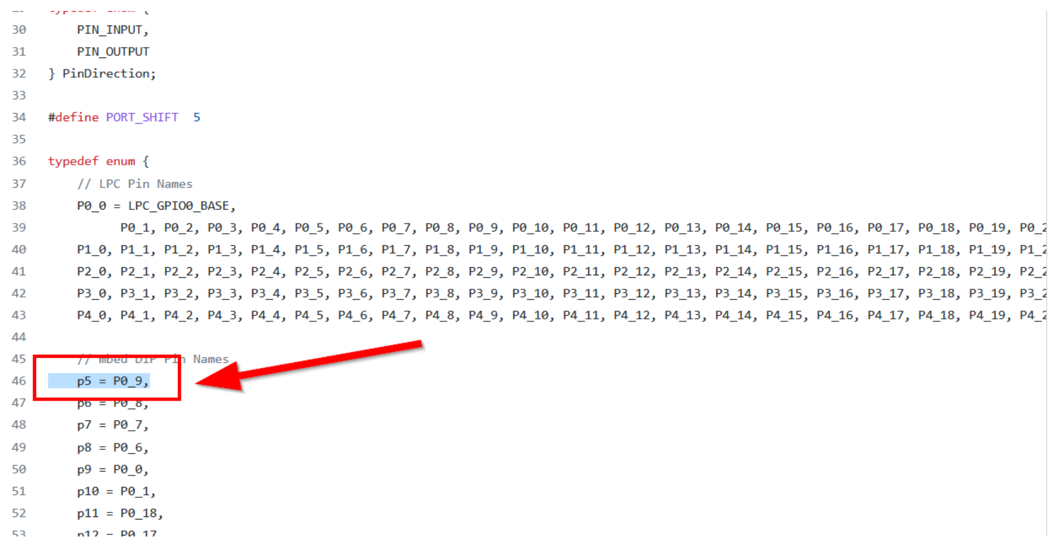
Public Member Functions	
	DigitalOut (PinName pin)
	Create a DigitalOut connected to the specified pin. More...
	DigitalOut (PinName pin, int value)
	Create a DigitalOut connected to the specified pin. More...
void	write (int value)
	Set the output, specified as 0 or 1 (int) More...



```
#include "mbed.h"

DigitalOut miLED(p5);

int main() {
    miLED.write(1);
}
```

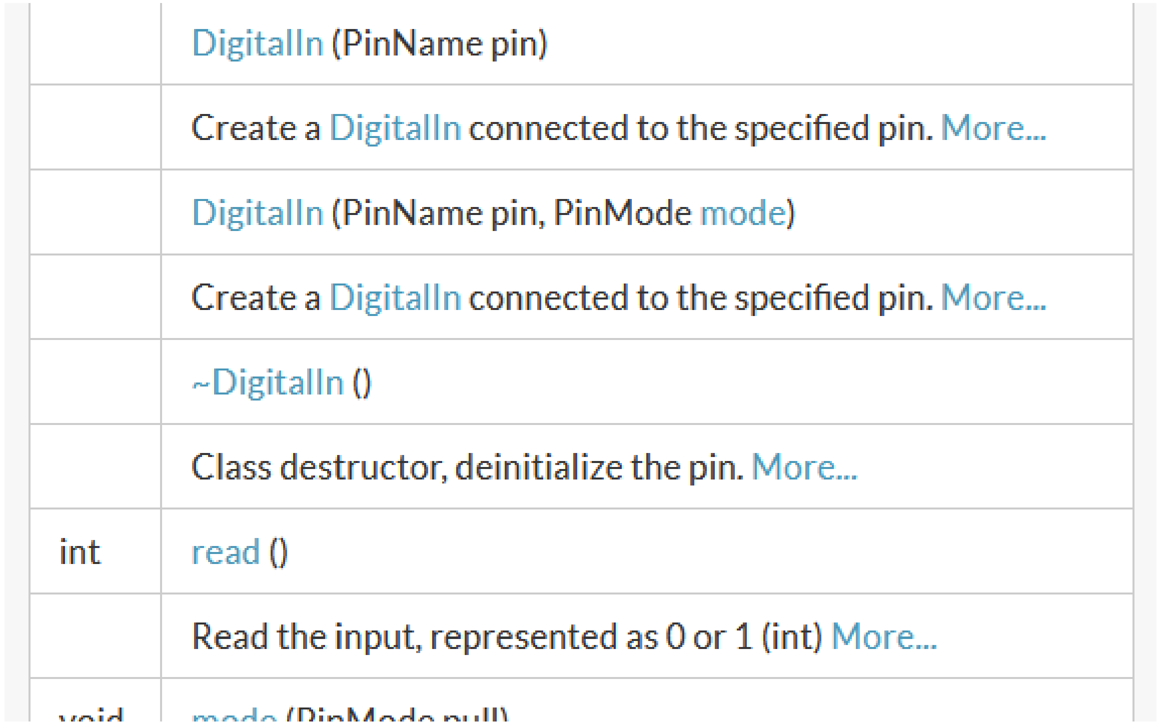
Nomenclatura de los pines

- El emulador está basado en el NXP LPC1768
- https://github.com/ARMmbed/mbed-os/blob/master/targets/TARGET_NXP/TARGET_LPC176X/TARGET_MBED_LPC1768/PinNames.h
- Podríamos cambiar en el código anterior p5 por P0_9 y funcionaría igual

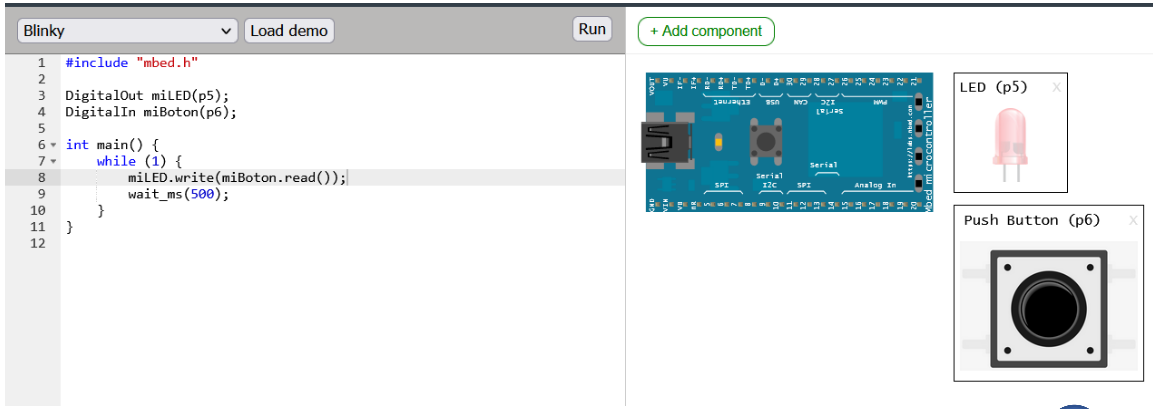


```
    PIN_INPUT,
    PIN_OUTPUT
} PinDirection;

#define PORT_SHIFT  5

typedef enum {
    // LPC Pin Names
    P0_0 = LPC_GPIO0_BASE,
          P0_1, P0_2, P0_3, P0_4, P0_5, P0_6, P0_7, P0_8, P0_9, P0_10, P0_11, P0_12, P0_13, P0_14, P0_15, P0_16, P0_17, P0_18, P0_19, P0_
    P1_0, P1_1, P1_2, P1_3, P1_4, P1_5, P1_6, P1_7, P1_8, P1_9, P1_10, P1_11, P1_12, P1_13, P1_14, P1_15, P1_16, P1_17, P1_18, P1_19, P1_
    P2_0, P2_1, P2_2, P2_3, P2_4, P2_5, P2_6, P2_7, P2_8, P2_9, P2_10, P2_11, P2_12, P2_13, P2_14, P2_15, P2_16, P2_17, P2_18, P2_19, P2_
    P3_0, P3_1, P3_2, P3_3, P3_4, P3_5, P3_6, P3_7, P3_8, P3_9, P3_10, P3_11, P3_12, P3_13, P3_14, P3_15, P3_16, P3_17, P3_18, P3_19, P3_
    P4_0, P4_1, P4_2, P4_3, P4_4, P4_5, P4_6, P4_7, P4_8, P4_9, P4_10, P4_11, P4_12, P4_13, P4_14, P4_15, P4_16, P4_17, P4_18, P4_19, P4_

    // mbed DIP Pin Names
    p5 = P0_9,
    p6 = P0_8,
    p7 = P0_7,
    p8 = P0_6,
    p9 = P0_0,
    p10 = P0_1,
    p11 = P0_18,
    p12 = P0_17
```

DigitalIn



	DigitalIn (PinName pin)
	Create a DigitalIn connected to the specified pin. More...
	DigitalIn (PinName pin, PinMode mode)
	Create a DigitalIn connected to the specified pin. More...
	~DigitalIn ()
	Class destructor, deinitialize the pin. More...
int	read ()
	Read the input, represented as 0 or 1 (int) More...
void	mode (PinMode pull)



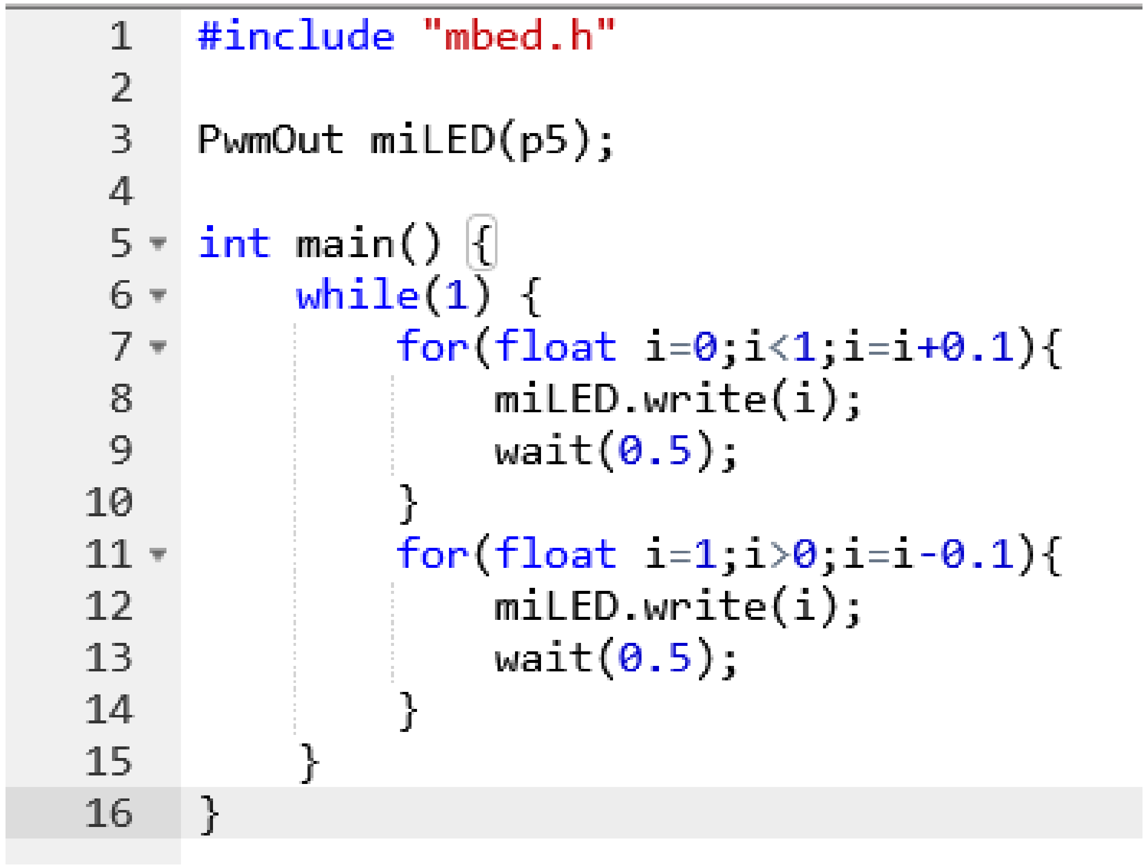
```
#include "mbed.h"

DigitalOut miLED(p5);
DigitalIn miBoton(p6);

int main() {
    while (1) {
        miLED.write(miBoton.read());
        wait_ms(500);
    }
}
```

Los modos pull son:

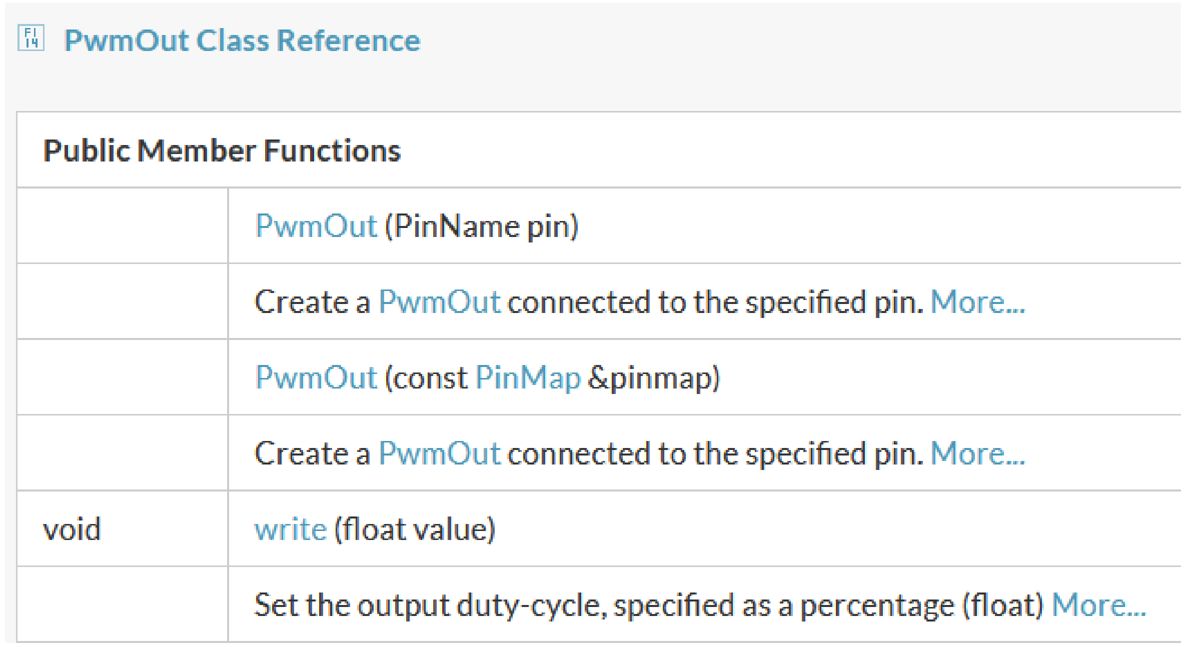
- PullUp, PullDown, PullNone, OpenDrain

Hay que poner siempre wait en los bucles infinitos

PWMOut



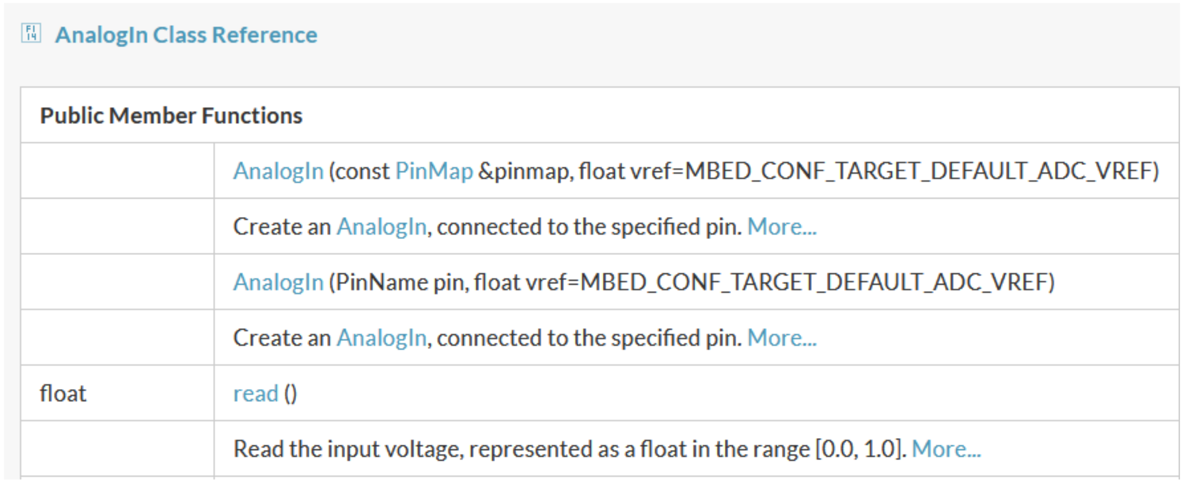
```
#include "mbed.h"

PwmOut miLED(p5);

int main() {
    while(1) {
        for(float i=0;i<1;i=i+0.1){
            miLED.write(i);
            wait(0.5);
        }
        for(float i=1;i>0;i=i-0.1){
            miLED.write(i);
            wait(0.5);
        }
    }
}
```



PwmOut Class Reference

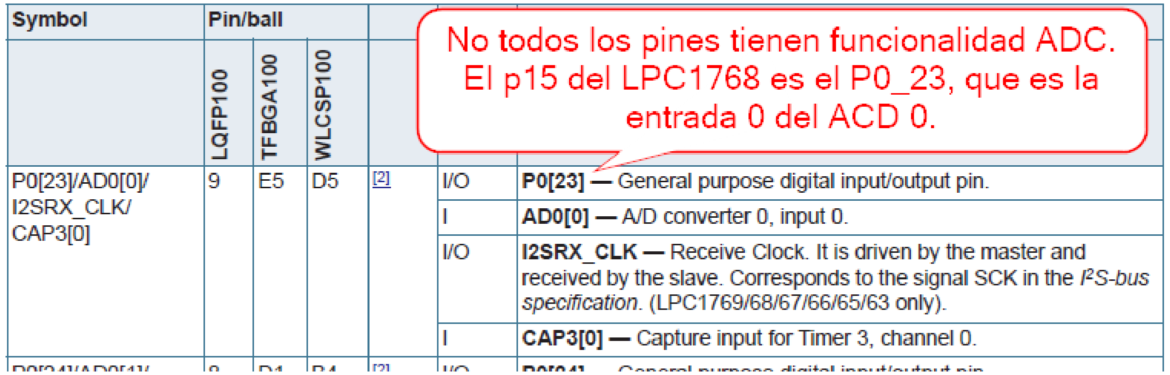
Public Member Functions	
	PwmOut (PinName pin)
	Create a PwmOut connected to the specified pin. More...
	PwmOut (const PinMap &pinmap)
	Create a PwmOut connected to the specified pin. More...
void	write (float value)
	Set the output duty-cycle, specified as a percentage (float) More...

AnalogIn



AnalogIn Class Reference

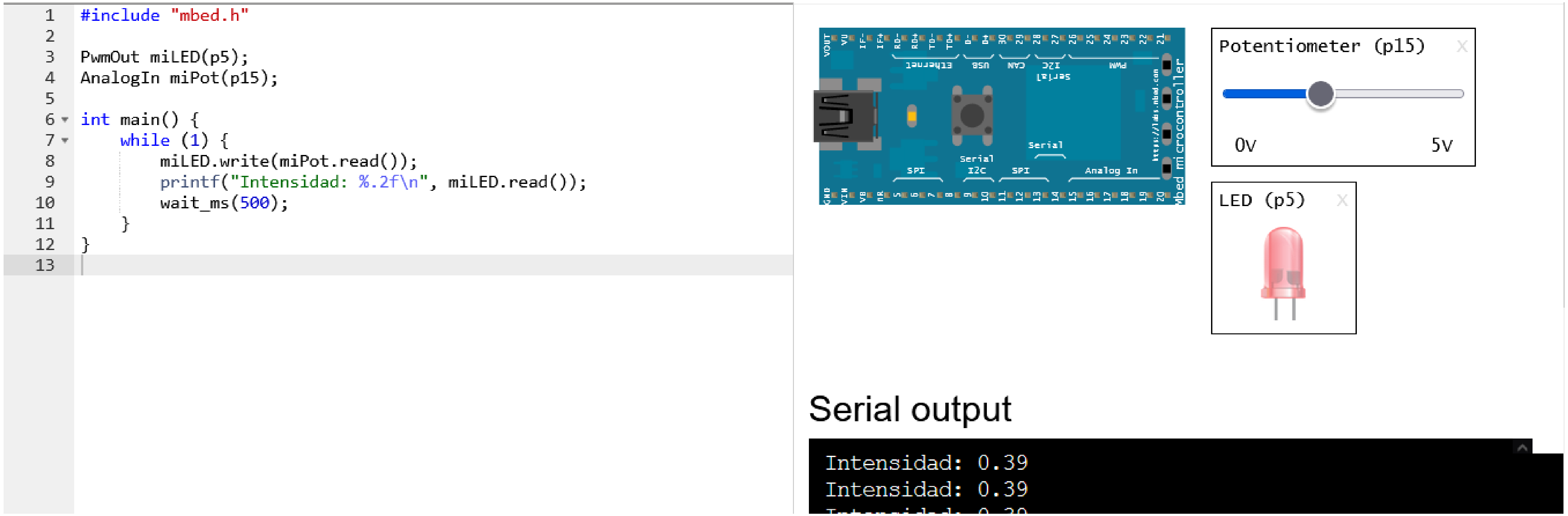
Public Member Functions	
	AnalogIn (const PinMap &pinmap, float vref=MBED_CONF_TARGET_DEFAULT_ADC_VREF)
	Create an AnalogIn, connected to the specified pin. More...
	AnalogIn (PinName pin, float vref=MBED_CONF_TARGET_DEFAULT_ADC_VREF)
	Create an AnalogIn, connected to the specified pin. More...
float	read ()
	Read the input voltage, represented as a float in the range [0.0, 1.0]. More...



No todos los pines tienen funcionalidad ADC. El p15 del LPC1768 es el P0_23, que es la entrada 0 del ACD 0.

Symbol	Pin/ball					
	LQFP100	TFBGA100	WLCSP100			
P0[23]/AD0[0]/ I2SRX_CLK/ CAP3[0]	9	E5	D5	[2]	I/O	**P0[23]** — General purpose digital input/output pin.
					I	**AD0[0]** — A/D converter 0, input 0.
					I/O	**I2SRX_CLK** — Receive Clock. It is driven by the master and received by the slave. Corresponds to the signal SCK in the I^2S*-bus specification*. (LPC1769/68/67/66/65/63 only).
					I	**CAP3[0]** — Capture input for Timer 3, channel 0.
P0[24]/AD0[1]/	8	D1	B4	[2]	I/O	**P0[24]** — General purpose digital input/output pin

AnalogIn y PwmOut

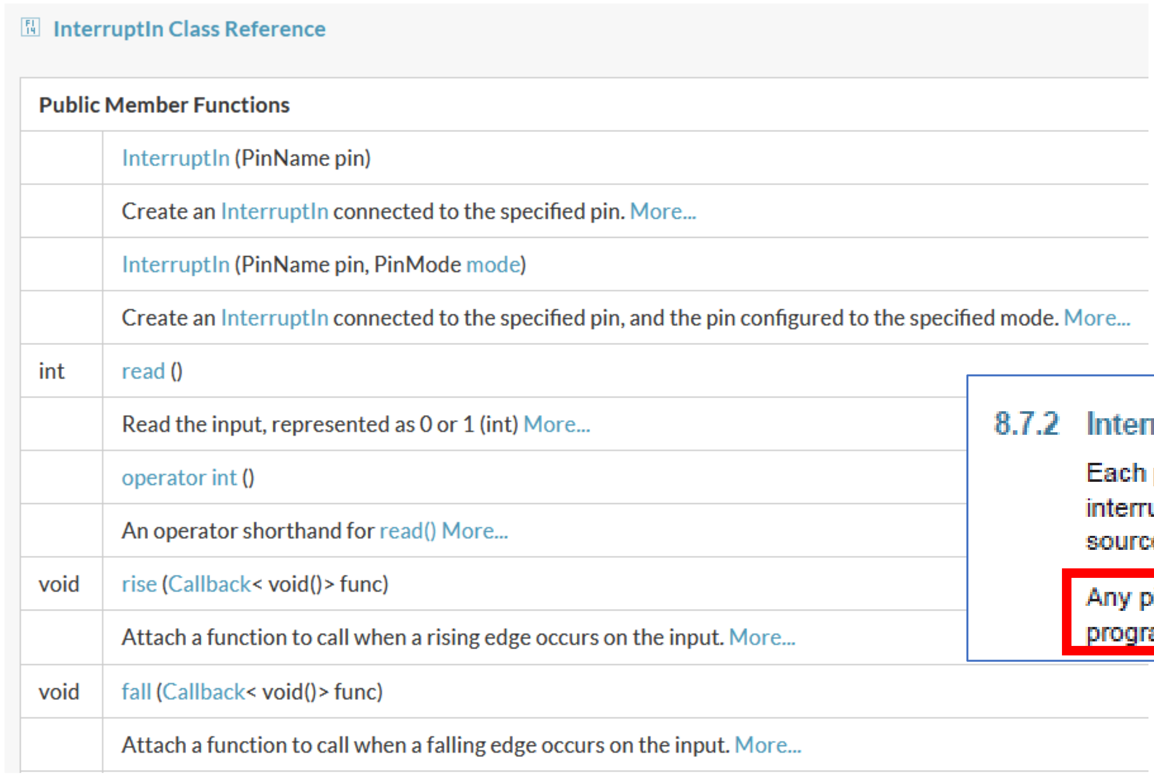


```cpp
#include "mbed.h"

PwmOut miLED(p5);
AnalogIn miPot(p15);

int main() {
    while (1) {
        miLED.write(miPot.read());
        printf("Intensidad: %.2f\n", miLED.read());
        wait_ms(500);
    }
}
```

Serial output

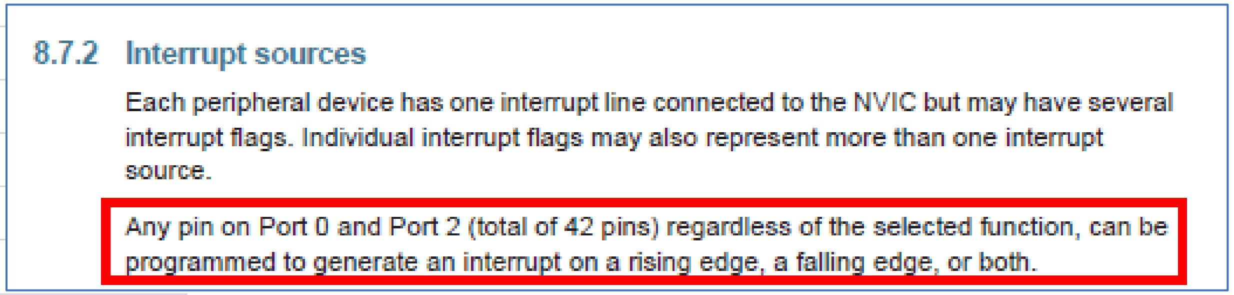
```
Intensidad: 0.39
Intensidad: 0.39
```

Interrupciones externas



InterruptIn Class Reference

Public Member Functions	
	InterruptIn (PinName pin)
	Create an InterruptIn connected to the specified pin. More...
	InterruptIn (PinName pin, PinMode mode)
	Create an InterruptIn connected to the specified pin, and the pin configured to the specified mode. More...
int	read ()
	Read the input, represented as 0 or 1 (int) More...
	operator int ()
	An operator shorthand for read() More...
void	rise (Callback< void()> func)
	Attach a function to call when a rising edge occurs on the input. More...
void	fall (Callback< void()> func)
	Attach a function to call when a falling edge occurs on the input. More...

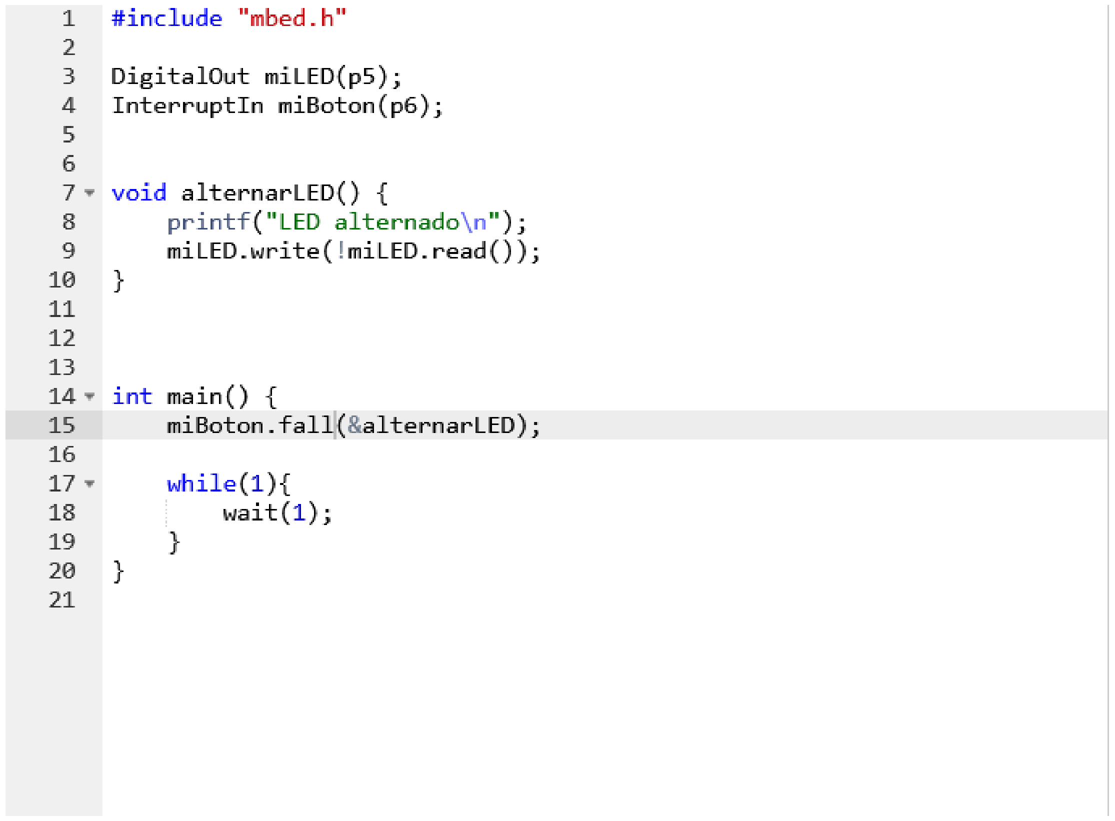


8.7.2 Interrupt sources

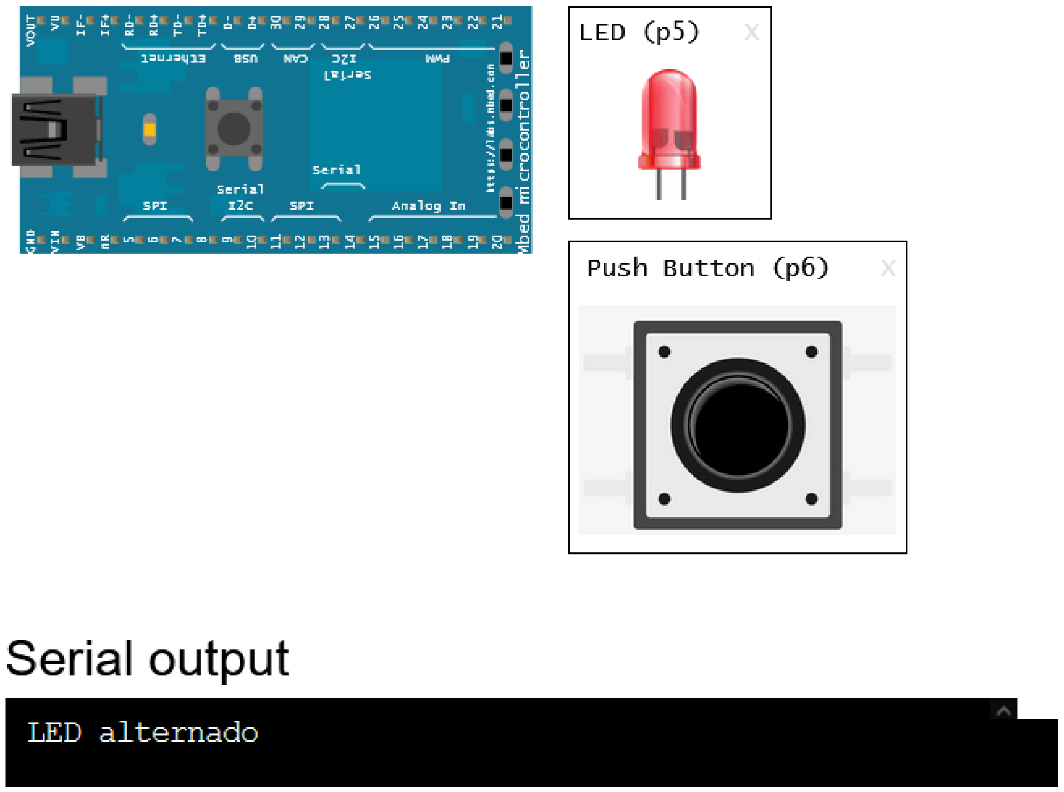
Each peripheral device has one interrupt line connected to the NVIC but may have several interrupt flags. Individual interrupt flags may also represent more than one interrupt source.

Any pin on Port 0 and Port 2 (total of 42 pins) regardless of the selected function, can be programmed to generate an interrupt on a rising edge, a falling edge, or both.

Interrupciones externas

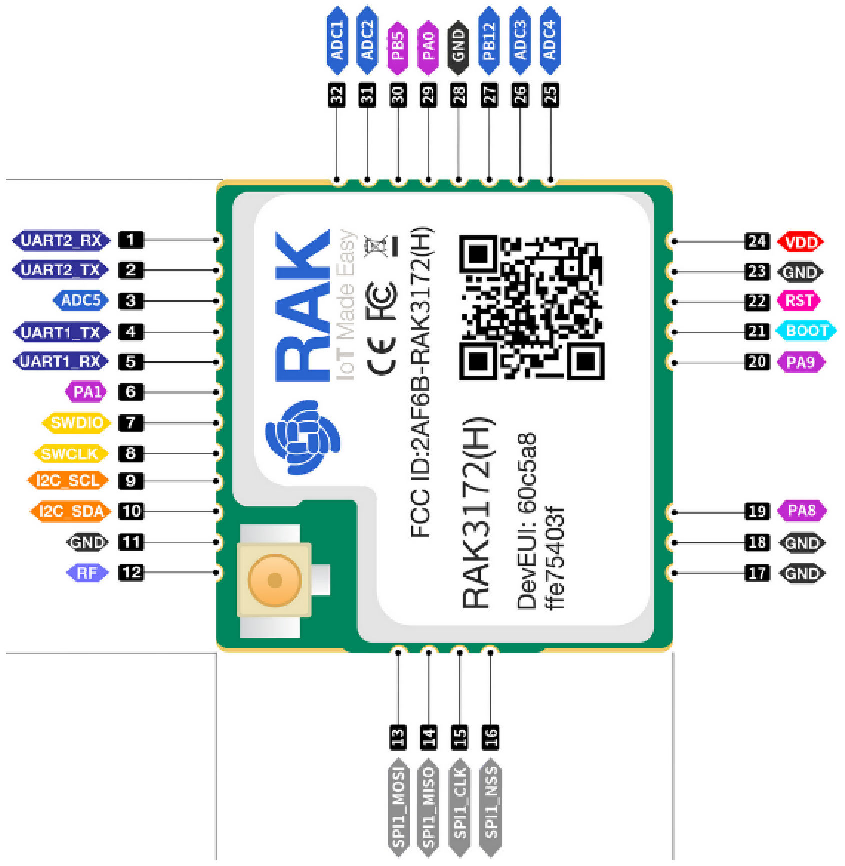


```
#include "mbed.h"

DigitalOut miLED(p5);
InterruptIn miBoton(p6);


void alternarLED() {
    printf("LED alternado\n");
    miLED.write(!miLED.read());
}



int main() {
    miBoton.fall(&alternarLED);

    while(1){
        wait(1);
    }
}

```



```
LED alternado
```

RAK3172



Features

- Based on **STM32WLE5CCU6**
- **LoRaWAN 1.0.3** specification compliant
- **Supported bands:** EU433, CN470, IN865, EU868, AU915, US915, KR920, RU864, and AS923-1/2/3/4
- LoRaWAN Activation by OTAA/ABP
- LoRa Point to Point (P2P) communication
- Easy to use AT Command Set via UART interface
- Long-range - greater than 15 km with optimized antenna
- Arm Cortex-M4 32-bit
- 256 kbytes flash memory with ECC
- 64 kbytes RAM
- Ultra-Low Power Consumption of 1.69 μA in sleep mode
- **Supply Voltage:** 2.0 V ~ 3.6 V
- **Temperature Range:** -40° C ~ 85° C

RAK3272

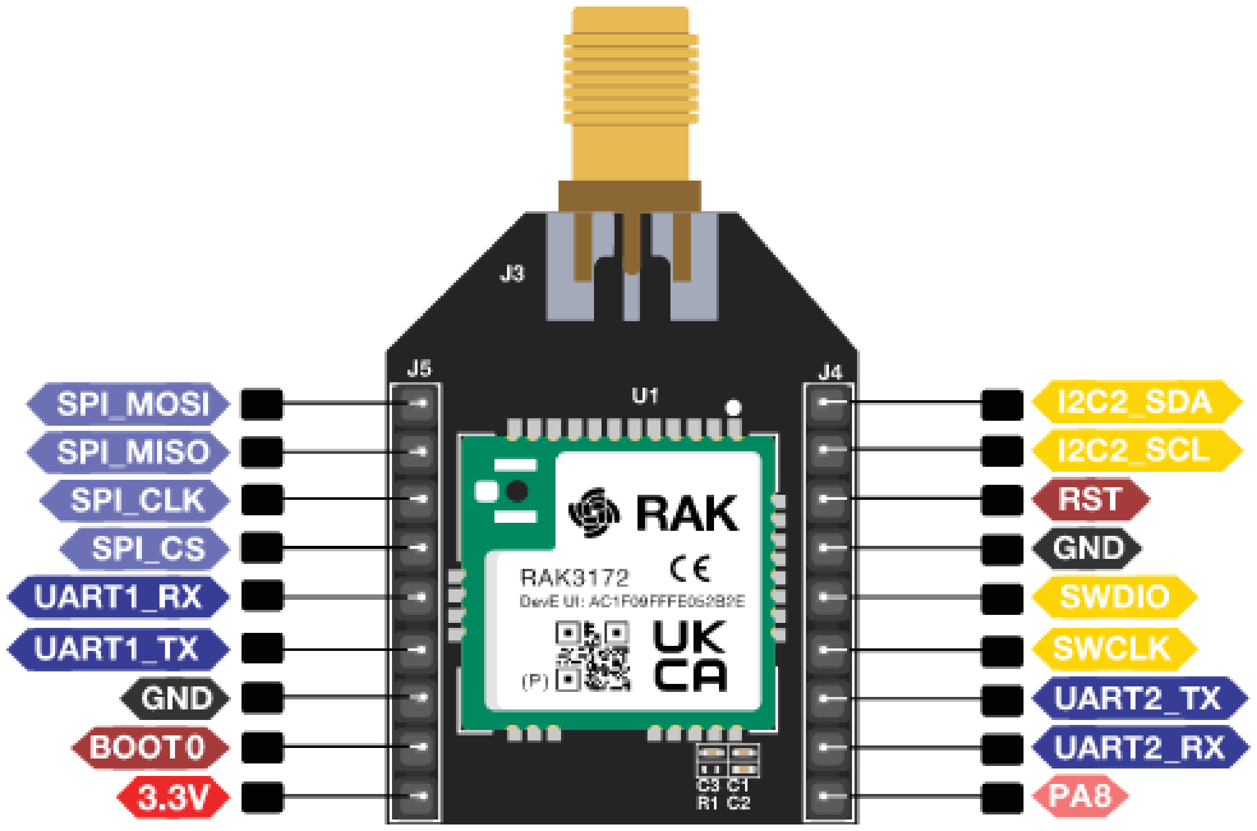
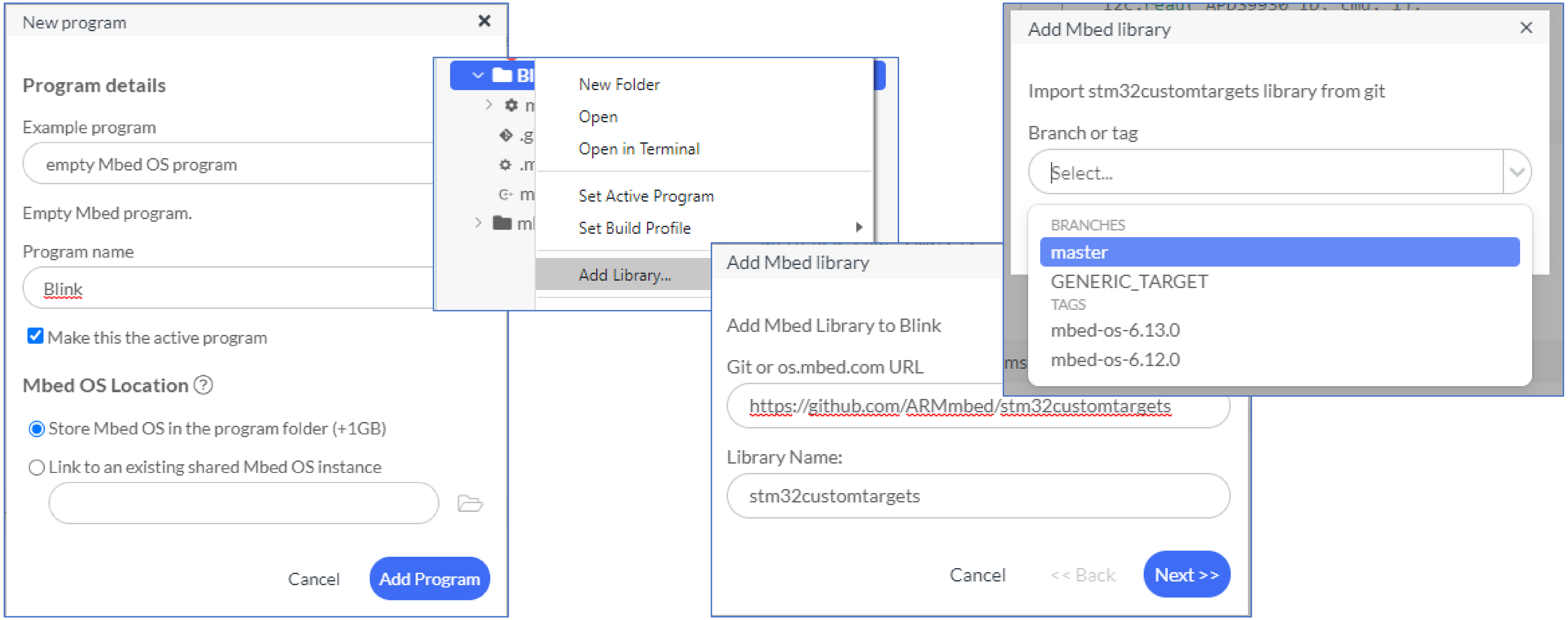


Figure 2: RAK3272S Breakout Board Pinout

RAK3172 en Mbed Studio 1/3

- Mbed Studio no incluye aún el módulo RAK3172
- Afortunadamente Hallard ha incluido una en la librería de "Custom Targets" de Mbed
  - https://github.com/ARMmbed/stm32customtargets
  - Procedimiento:
    1. Crear un programa nuevo
    2. Importar la librería stm32customtargets
    3. Copiar el archivo custom_targets.json de la librería anterior a la carpeta raíz del proyecto
    4. Seleccionar el nuevo target RAK3172_BREAKBOARD

RAK3172 en Mbed Studio 2/3



New program

Program details

Example program: empty Mbed OS program

Empty Mbed program.

Program name: Blink

[✓] Make this the active program

Mbed OS Location

[●] Store Mbed OS in the program folder (+1GB)

[○] Link to an existing shared Mbed OS instance

Cancel | Add Program

Add Mbed library

Add Mbed Library to Blink

Git or os.mbed.com URL: https://github.com/ARMmbed/stm32customtargets

Library Name: stm32customtargets

Cancel | << Back | Next >>

Add Mbed library

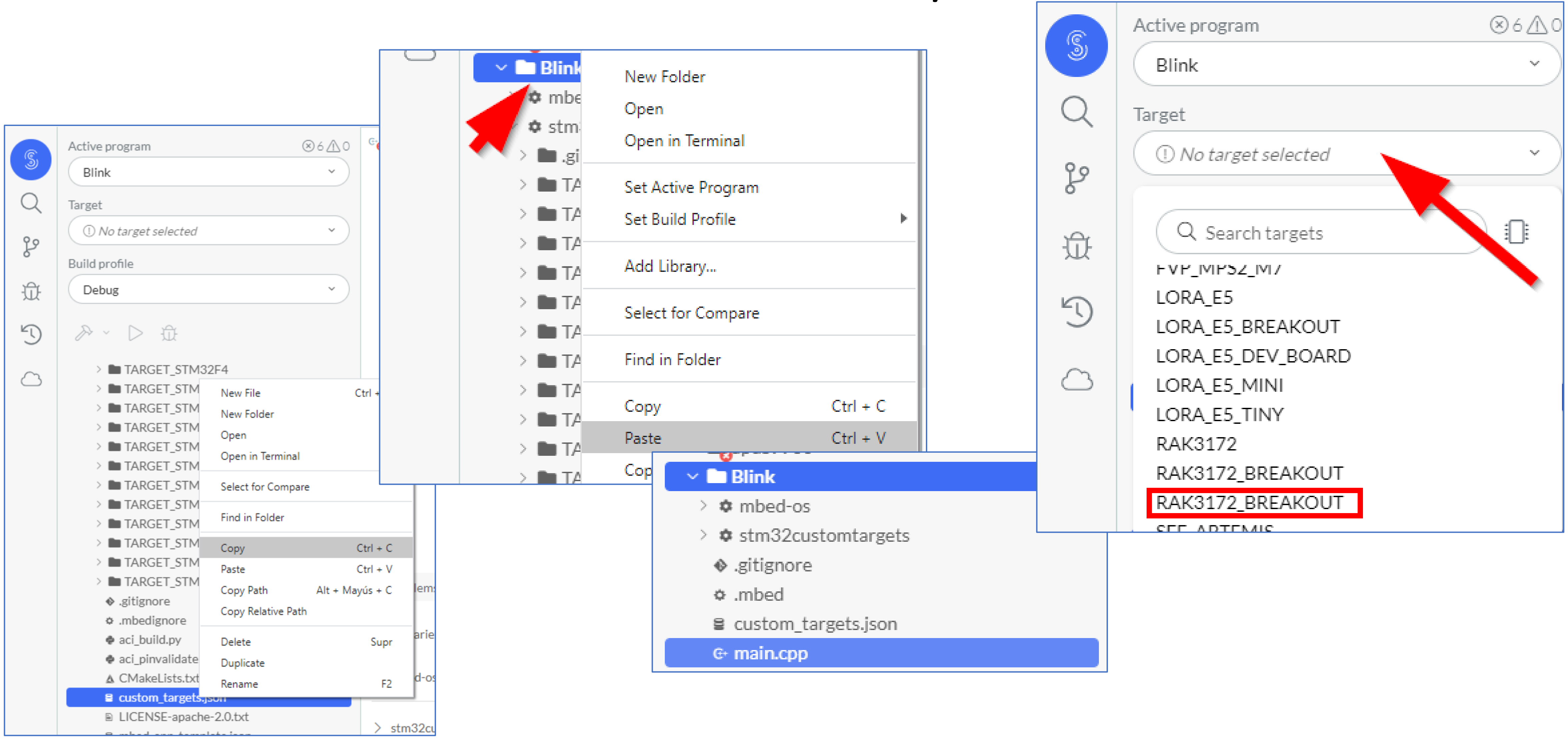
Import stm32customtargets library from git

Branch or tag: Select...

BRANCHES
- master
- GENERIC_TARGET

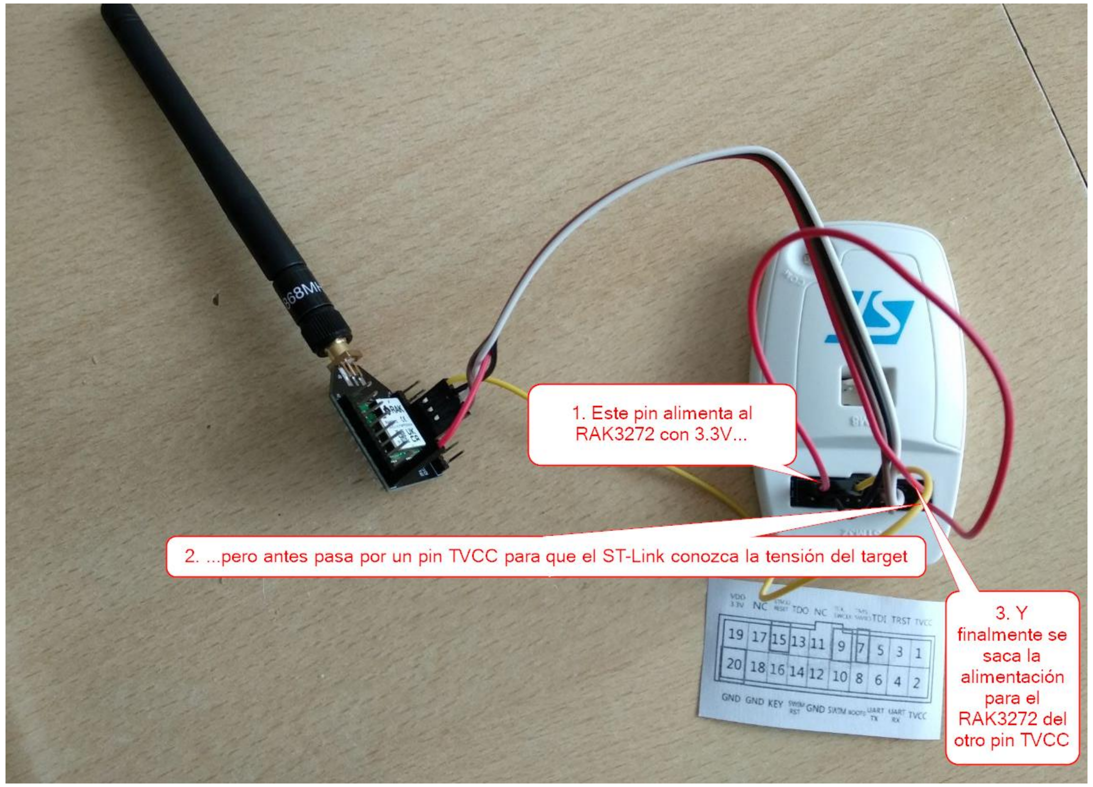
TAGS
- mbed-os-6.13.0
- mbed-os-6.12.0

RAK3172 en Mbed Studio 3/3


Active program: Blink
Target: No target selected
Build profile: Debug

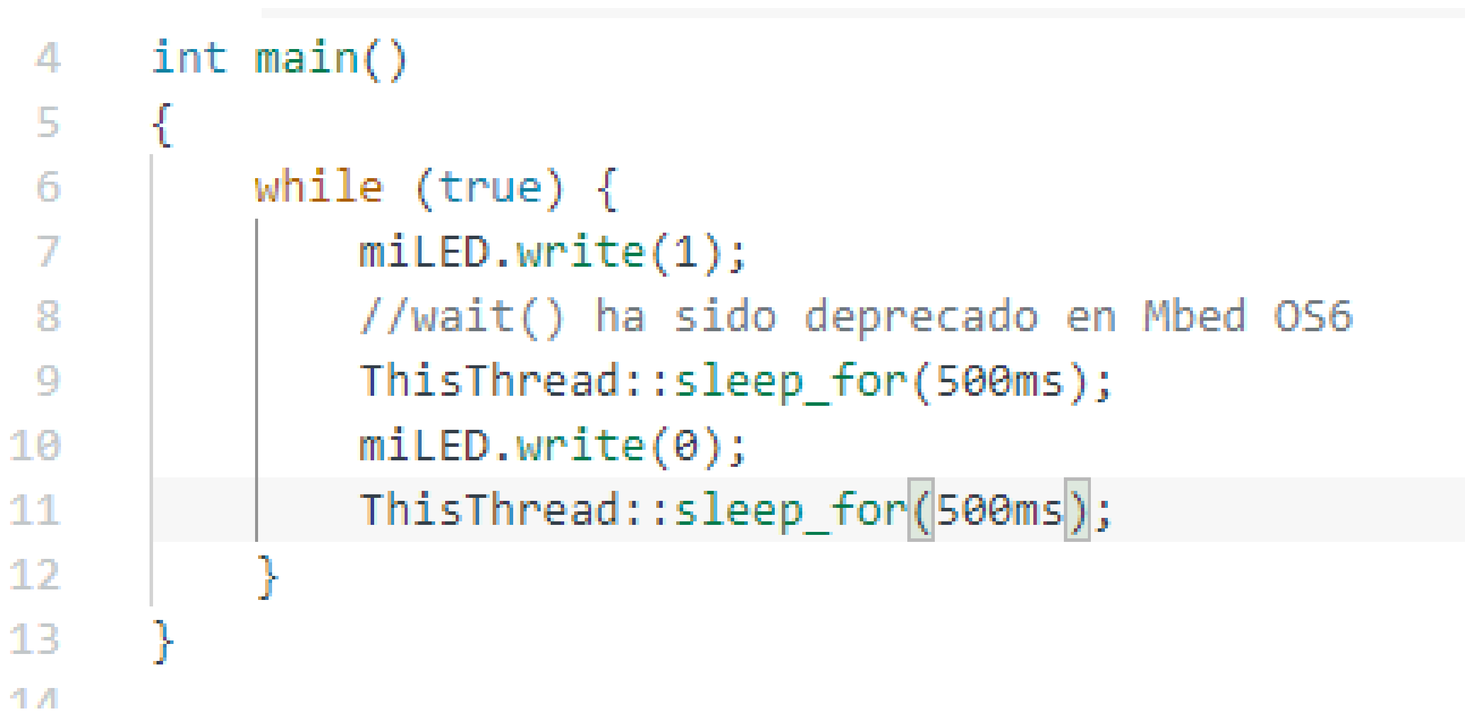
Conexión del ST-Link v2

Se utiliza el pin 3.3V del ST-Link para alimentar el RAK3272, pero además tiene que conectarse a los pines TVCC para que el ST-Link "sepa" cuál es la tensión de alimentación del target.



1. Este pin alimenta al RAK3272 con 3.3V...
2. ...pero antes pasa por un pin TVCC para que el ST-Link conozca la tensión del target
3. Y finalmente se saca la alimentación para el RAK3272 del otro pin TVCC
VDD 3.3V
NC
TDO
NC
TDI
TRST
TVCC
19 17 15 13 11 9 7 5 3 1
20 18 16 14 12 10 8 6 4 2
GND
GND
KEY
SWIM RST
GND
SWIM
UART TX
UART RX
TVCC

Blink en el RAK3272



```
int main()
{
    while (true) {
        miLED.write(1);
        //wait() ha sido deprecado en Mbed OS6
        ThisThread::sleep_for(500ms);
        miLED.write(0);
        ThisThread::sleep_for(500ms);
    }
}
```