

INFORME PRÀCTICA 4: SISTEMAS OPERATIVOS EN TIEMPO REAL

Multitarea en ESP32

Código 1

```
#include <Arduino.h>

void anotherTask( void * parameter );

void setup()
{
  Serial.begin(115200);
  /* we create a new task here */
  pinMode(LED_BUILTIN, OUTPUT);
  xTaskCreate(
    anotherTask, /* Task function. */
    "another Task", /* name of task. */
    10000, /* Stack size of task */
    NULL, /* parameter of the task */
    1, /* priority of the task */
    NULL); /* Task handle to keep track of created task */
}
/* the forever loop() function is invoked by Arduino ESP32 loopTask */
void loop()
{
  Serial.println("this is ESP32 Task");
  vTaskDelay(300);
}
/* this function will be invoked when additionalTask was created */
void anotherTask( void * parameter )
{
  /* loop forever */
  for(;;)
  {
    Serial.println("this is another Task");
    vTaskDelay(300);
    digitalWrite(LED_BUILTIN, HIGH);
    vTaskDelay(300);
    digitalWrite(LED_BUILTIN, LOW);
    vTaskDelay(300);
  }
}
/* delete a task when finish, this will never happen because this is infinity loop */
vTaskDelete( NULL );
}
```

Este código enciende y apaga el led *LED_BUILTIN*, el cual corresponde al LED integrado del ESP32 y emite luz blanca, a través de una tarea (*Task*). Muestra por pantalla del terminal los mensajes asignados a cada tarea para ver que son independientes.

Inclusión de librerías

- + `#include <Arduino.h>`
 - Incluye la librería propia de Arduino para utilizar sus funciones y definiciones.

Declaración de la función *anotherTask*

- + `void anotherTask(void * parameter)`
 - Es un prototipo que se escribe antes de empezar el código y se trata de una tarea secundaria ejecutada por el sistema operativo en tiempo real, un RTOS, del microprocesador.

Configuración de la función *setup()*

- + `Serial.begin(115200)`
 - Inicializa la comunicación serial a una velocidad de 115200 baudios
- + `pinMode(LED_BUILTIN, OUTPUT)`
 - Configura como pin de salida que representa el LED integrado en la placa
- + `xTaskCreate()`
 - Se crea una nueva tarea con la función `xTaskCreate()`, la cual ejecuta la función `anotherTask`. Esta tarea se crea con prioridad de 1 y sin parámetros adicionales.

Bucle principal *loop()*

- + `Serial.println("this is ESP32 Task"); vTaskDelay(300);`
 - Se imprime un mensaje cada 300 ms para que se muestre como funcionan las tareas.

Función *anotherTask*

- + `Serial.println("this is another Task"); vTaskDelay(300);`
 - Dentro de esta función se ejecuta una tarea separada.
- + Dentro del `for`
 - Se trata de un bucle infinito en el cual se imprime un mensaje cada 300 ms. Dentro del bucle también se ejecuta una función para encender y apagar el led integrado en el microcontrolador cada 300 ms.
- + `TaskDelete(NULL)`
 - Esta línea está destinada a eliminar la tarea cuando termina pero es una función que nunca se ejecutará debido al bucle anterior, ya que la tarea es infinita.

Ejercicio práctico

```
#include <Arduino.h>

void anotherTask( void * parameter );
void ThirdTask( void * parameter );
```

```

void setup()
{
    Serial.begin(115200);
    /* we create a new task here */
    pinMode(LED_BUILTIN, OUTPUT);
    pinMode(19, OUTPUT);
    xTaskCreate(
        anotherTask, /* Task function. */
        "another Task", /* name of task. */
        1000, /* Stack size of task */
        NULL, /* parameter of the task */
        1, /* priority of the task */
        NULL); /* Task handle to keep track of created task */

    xTaskCreate(
        ThirdTask,
        "Third Task",
        NULL,
        1,
        NULL);
}
/* the forever loop() function is invoked by Arduino ESP32 loopTask */
void loop()
{
    Serial.println("this is ESP32 Task");
    vTaskDelay(300);
    digitalWrite(LED_BUILTIN, HIGH);
    vTaskDelay(300);
    digitalWrite(LED_BUILTIN, LOW);
    vTaskDelay(300);
}

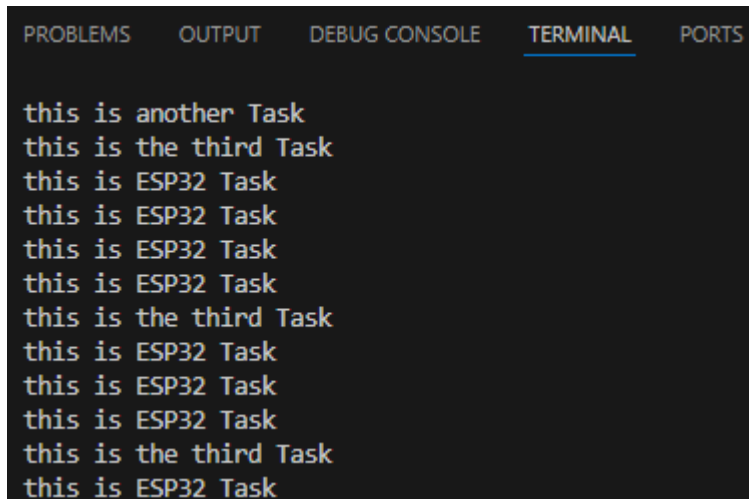
/* this function will be invoked when additionalTask was created */
void anotherTask( void * parameter )
{
    /* loop forever */
    for(;;)
    {
        Serial.println("this is another Task");
        vTaskDelay(10000);
    }
}

void ThirdTask( void * parameter )
{
    /* loop forever */
    for(;;)
    {
        Serial.println("this is the third task");
        vTaskDelay(1000);
        digitalWrite(19, HIGH);
        vTaskDelay(1000);
        digitalWrite(19, LOW);
    }
}

```

```
vTaskDelay(1000);  
}  
  
vTaskDelay( NULL );  
}
```

- * En este segundo código, he añadido una tercera tarea (*ThirdTask*) la cual enciende un led conectado al pin 19 de la placa y en el terminal se escribe el mensaje *this is the third task* cuando el led se enciende. Como conjunto, en la placa se enciende el *LED_BUILTIN* integrado y el LED externo conectado al pin 19. En el terminal, cada cierto tiempo definido en el código, se ve lo siguiente:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  
  
this is another Task  
this is the third Task  
this is ESP32 Task  
this is ESP32 Task  
this is ESP32 Task  
this is ESP32 Task  
this is the third Task  
this is ESP32 Task  
this is ESP32 Task  
this is ESP32 Task  
this is the third Task  
this is ESP32 Task
```