

Práctica 4 Parte 2:

Código:

```
#include <Arduino.h>
#define LED 16
unsigned long myTime;
#define interruptPin 0

void Task1(void * parameter);
void anotherTask(void * parameter);

void setup(){
    Serial.begin(115200);

    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 */

    xTaskCreate(
        Task1, /* Task function. */
        "Task 1", /* name of task. */
        10000, /* Stack size of task */
        NULL, /* parameter of the task */
        1, /* priority of the task */
        NULL);
}

void loop(){
    Serial.println("this is ESP32 Task");
    delay(1000);
}

void Task1(void * parameter){
    pinMode(LED,OUTPUT);
    for(;;){
        Serial.println(myTime); // prints time since program started
        delay(500);
        digitalWrite(LED,HIGH);
        Serial.println("ON");
        delay(500);
        digitalWrite(LED,HIGH);
        Serial.println("OFF");
        digitalWrite(LED,LOW);
        delay(500);
    }
}

void anotherTask( void * parameter ){
    /* loop forever */
    for(;;){
        Serial.println("this is another Task");
        delay(1000);
    }
    vTaskDelete( NULL );
}

}
```

Funcionamiento:

En el caso de esta práctica queremos que se nos encienda y apague un led usando las tareas que hemos visto ya en el apartado 1, para ello vemos como el setup de nuestro código va a ser el mismo pero pondremos una tarea mas como vemos aquí:

```
void setup(){
  Serial.begin(115200);

  /* we create a new task here */
  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 */

  xTaskCreate(
    Task1, /* Task function. */
    "Task 1", /* name of task. */
    10000, /* Stack size of task */
    NULL, /* parameter of the task */
    1, /* priority of the task */
    NULL);
}
```

Con esto hecho nos faltara definir el loop y declarar la funcion y exactamente que queremos que esta haga, dicho esto vemos como el loop va a ser sencillo ya que solo queremos que nos escriba por pantalla un mensaje.

```
void loop(){
  Serial.println("this is ESP32 Task");
  delay(1000);
}
```

Finalmente, declararemos las dos tareas que hemos creado anteriormente, primero la "Task1", la cual sera la encargada del LED:

```
void Task1(void * parameter){
  pinMode(LED,OUTPUT);
  for(;;){
    Serial.println(myTime); // prints time since program started
    delay(500);
    digitalWrite(LED,HIGH);
    Serial.println("ON");
    delay(500);
    digitalWrite(LED,HIGH);
    Serial.println("OFF");
    digitalWrite(LED,LOW);
    delay(500);
  }
}
```

Y seguidamente la tarea "anothertask" la cual pondra por pantalla el siguiente mensaje:

```
void anotherTask( void * parameter ){
  /* loop forever */
  for(;;){
    Serial.println("this is another Task");
    delay(1000);
  }
  vTaskDelete( NULL );
}
```