

Práctica 4 Parte 1:

Código:

```
#include <Arduino.h>

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

  /* delete a task when finish,
  this will never happen because this is infinity loop */
  vTaskDelete( NULL );
}

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 */
}

/* the forever loop() function is invoked by Arduino ESP32 loopTask */
void loop()
{
  Serial.println("this is ESP32 Task");
  delay(1000);
}
```

Pantalla:

Cuando subimos el código a la ESP32, este hace que salgan por pantalla dos mensajes a la vez. Uno se muestra desde el loop y el otro se muestra con el uso de una tarea. Los dos se muestran a la vez ya que los dos usan loops infinitos, de manera que como no hay ningún tipo de "stop" estos mensajes aparecerán hasta que se termine el programa.

□

Funcionamiento:

En este código hemos usado las tareas. Para ello primero es necesario crear una tarea, eso lo haremos dentro del void setup():

```
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 */
}
```

Como podemos ver cada linea de codigo tiene un comentario al lado que nos permite saber que es exactamente el parametro que estamos poniendo.

Una vez creada la tasca, en el void loop(), ponemos el tiempo que va a tardar en escribir por pantalla la frase y tambien la frase que queremos que escriba claro.

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

Una vez hecho el loop hay que hacer el subprograma, en este un loop, que esta se va encargar de escribir por pantalla una frase y lo va a hacer cada vez que se cumpla el delay. En nuestro caso como hemos dicho antes el bucle no tiene fin por lo tanto va a seguir indefinidamente.

Para que este funciona hay que declarar como vemos en la primera linea.

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

  /* delete a task when finish,
  this will never happen because this is infinity loop */
  vTaskDelete( NULL );
}
```