

## Proyecto Display LED: Control Display LED via GPIO.

Prof. Mauro San Martín

Este documento define requisitos y da instrucciones para la realización del *Taller 2* del Proyecto *Display Led*. Este taller consiste en el diseño e implementación del software controlador del display led en una Raspberry Pi. Al final del taller debe haberse implementado un programa en *C* capaz de mostrar diseños, almacenados en arreglos bidimensionales, en el display led previamente construido.

El *Taller 2* es una actividad grupal con evaluación individual. Es responsabilidad de cada estudiante sistematizar la evidencia de su trabajo y aportes al proyecto grupal. La plataforma recomendada para apoyar la colaboración y dejar un registro del trabajo realizado por cada integrante del grupo es *GitHub*.

**Requisitos.** Diseñar e implementar en una Raspberry Pi el programa controlador del display led de 8 x 8 elementos, construido previamente, conectado a su GPIO. Este programa debe cumplir con los siguientes requisitos:

1. (20 %) El programa debe estar implementado en *C* utilizando la librería *pigpio* para comunicarse con el display led a través de la GPIO de la Raspberry Pi, utilizando los pines indicados en el diagrama entregado en el *Taller 1*. Considere el código de ejemplo adjunto al final de este documento.
2. (20 %) El programa debe ser correcto, legible y estar bien organizado según los principios de la programación estructurada.
3. (30 %) El programa debe incluir módulos para preparar el display para mostrar un diseño dado, así como para asegurarse que todos los pines utilizados queden inactivos antes de finalizar.
4. (30 %) El programa debe almacenar los diseños a mostrar en el display led en matrices de 8 x 8 (arreglos bidimensionales). Para mostrar un diseño en el display led, debe incluir un módulo que reciba un diseño y un tiempo de despliegue en centésimas de segundo, y muestre el diseño en el display led por el tiempo indicado. Si el tiempo es igual a 0, el diseño se debe mostrar hasta que el usuario interrumpa el programa.

IMPORTANTE: el programa **nunca** debe prender más de un led simultáneamente, el diseño se debe visualizar prendiendo momentáneamente y en rápida sucesión cada led que corresponda según el patrón, una y otra vez mientras no se haya agotado el tiempo de despliegue.

5. (30 %) Opcionalmente el programa puede incluir un módulo que, utilizando los módulos descritos previamente, permita visualizar animaciones almacenadas previamente en arreglos de matrices (cada matriz conteniendo un frame de la animación). Los parámetros de este módulo deben ser: la animación almacenada según lo indicado, y un tiempo de presentación de la animación en segundos. Si el tiempo de presentación es 0, la animación se debe presentar hasta que el usuario interrumpa el programa. Este es un requisito opcional que complementará la calificación en caso de cumplirse.

**Instrucciones.** Antes de iniciar la etapa de codificación debe acordar y documentar el diseño del programa y el plan de implementación con su grupo:

1. Defina la organización de su programa e identifique cada uno de los módulos requeridos.
2. Cada módulo debe estar justificado, tener sus entradas y salidas bien definidas, y su funcionamiento especificado en pseudo código.
3. Las tareas de implementación deben estar claramente identificadas, priorizadas y asignadas a los integrantes del grupo.

Cada grupo deberá elaborar y entregar un informe que contenga:

1. Descripción general del programa y de la interacción de sus módulos.
2. Documentación detallada de cada módulo.
3. Código en C.
4. Evaluación crítica del producto construido.

**Materiales y herramientas.** Para completar este taller cada grupo necesitará haber completado su display led, y dispondrá además de:

- Raspberry Pi 4 y su fuente de poder (compartida con un grupo de la sección paralela).
- Tarjeta SD precargada con el ambiente de desarrollo (una por cada grupo).

**Evaluación.**

1. Cada requisito será calificado entre 0 y 6,0.
2. La calificación grupal  $G$  será la promedio de las calificaciones de los requisitos del taller según las ponderaciones indicadas.

3. El aporte individual  $I$  se medirá con un valor entre 0 y 1,0 (o 100 %), que refleje si el esfuerzo y aporte observados al resultado del taller, del o la integrante, es acorde a lo esperado dada la cantidad de integrantes del grupo. Por ejemplo, si el grupo tiene cuatro integrantes se esperará que el esfuerzo se divida uniformemente en cuatro (25 % cada integrante), entonces si un integrante demuestra un esfuerzo  $\geq 25\%$  su aporte individual será de 1,0. En ningún caso un integrante podrá obtener un  $I > 1,0$ . La calificación del taller  $T$  para cada integrante corresponderá entonces a  $G * I$ .

#### Hitos.

- Presentación primera etapa de desarrollo: **miércoles 10 de abril en clases**, 10 mins. por grupo.
- Entrega y demostración proyecto (*Taller 1* y *Taller 2*): **miércoles 17 de abril**.

```

1 //Adaptado de
2 // https://www.woolseyworkshop.com/2018/06/20/
3 // blink-making-an-led-blink-on-a-raspberry-pi/
4 #include <signal.h>
5 #include <stdio.h>
6 #include <pigpio.h>
7
8 const int led = 21; //GPIO 21
9 volatile sig_atomic_t signal_received = 0;
10
11 void sigint_handler(int signal) {
12     signal_received = signal;
13 }
14
15
16 int main() {
17     if (gpioInitialise() == PI_INIT_FAILED) {
18         printf("ERROR: No fue posible inicializar GPIO.\n");
19         return(1);
20     }
21     gpioSetMode(led, PI_OUTPUT);
22
23     signal(SIGINT, sigint_handler);
24     printf("Presionar CTRL-C para salir.\n");
25
26     while (!signal_received) {
27         gpioWrite(led, PI_HIGH);
28         time_sleep(1);
29         gpioWrite(led, PI_LOW);
30         time_sleep(1);
31     }
32     gpioSetMode(led, PI_INPUT);
33     gpioTerminate();
34     printf("\n");
35     return 0;
36 }

```

Puede compilar este ejemplo usando

```
1 gcc -o ejemplo ejemplo.c -lpigpio
```