# **PROYECTO 3 - VIDEOJUEGO PORTABLE**

## **Introducción**:

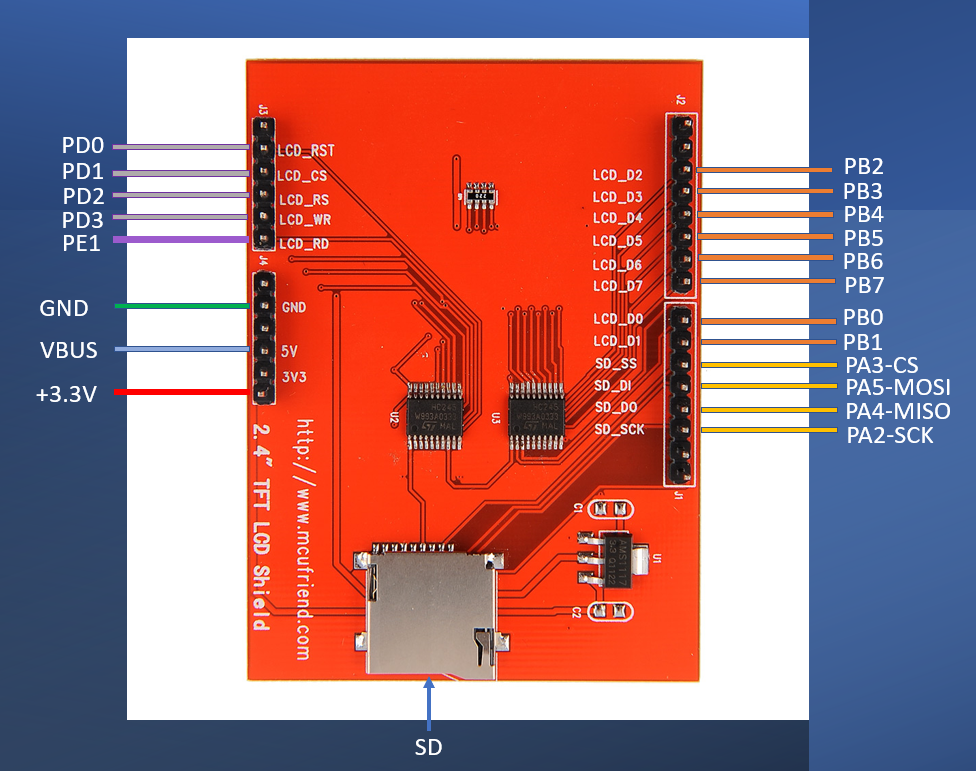
En el presente trabajo escrito se detalla y se copila toda la información necesaria para llevar a cabo el proyecto 3 de la clase de electrónica digital 2. El cual se basa en la elaboración de un videojuego implementado en la pantalla LCD ILI9341 en conjunto con el microcontrolador TIVA-C elaborado en el programa de energia. Como controles para el videojuego se usaron los dos botones que tiene le Tiva -C. Se uso una segunda Tiva para la implementación de dos melodías en 8 bits las cuales fueron Gourmet Race – Kirby y Take On me – a-ha para la melodía del jugador ganador.

## **Sobre el videojuego:**

El videojuego realizado fue Flappy Bird con dos jugadores. El cual tiene como objetivo alcanzar la puntuación máxima de 9 puntos sin chocar con los pilares que se van generando. Cada vez que un jugador choca con un pilar se le irá descontando un punto del Score. El primero en lograr el Score máximo será el ganador del videojuego.

## **Circuitos empleados:**

La siguiente imagen muestra el circuito principal del proyecto. Se puede observar el uso de la Pantalla LCD ILI9341 conectada a la Tiva uno. La cual se interconectada a la Tiva dos y el buzzer.

En esta imagen se observa la conexión de todos los pines que se usaron de la pantalla LCD ILI9341 a la Tiva-C uno. Cada uno con su respetivo pin de entrada en el microcontrolador.

## **Gráficos utilizados:**

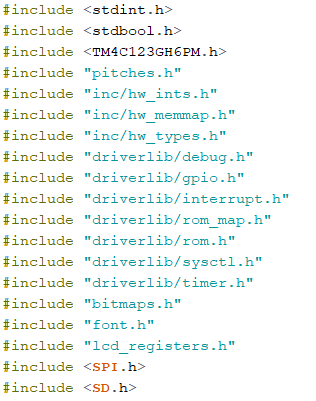
* Player1
* Player2
* Imagen que contiene dibujo

  Descripción generada automáticamenteGráfico Portada1
* Un dibujo de una caricatura

  Descripción generada automáticamente con confianza bajaGráfico Portada2

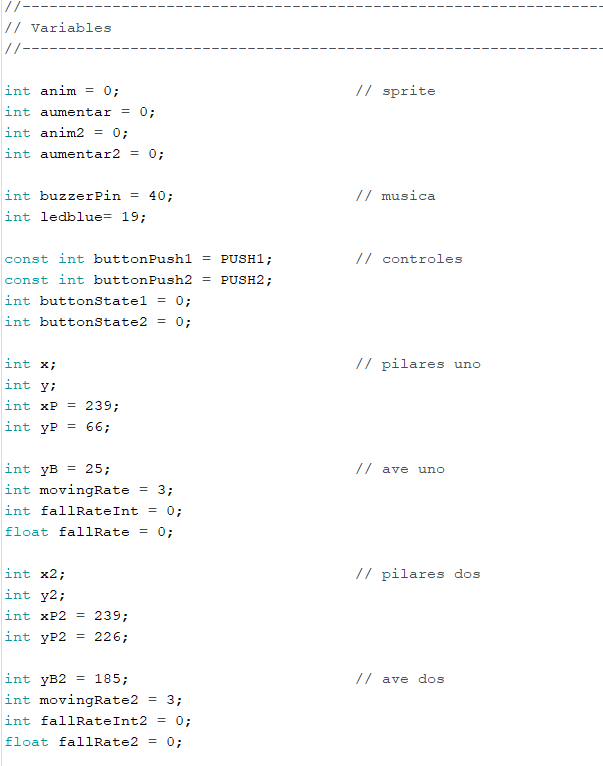
## **Análisis realizado:**

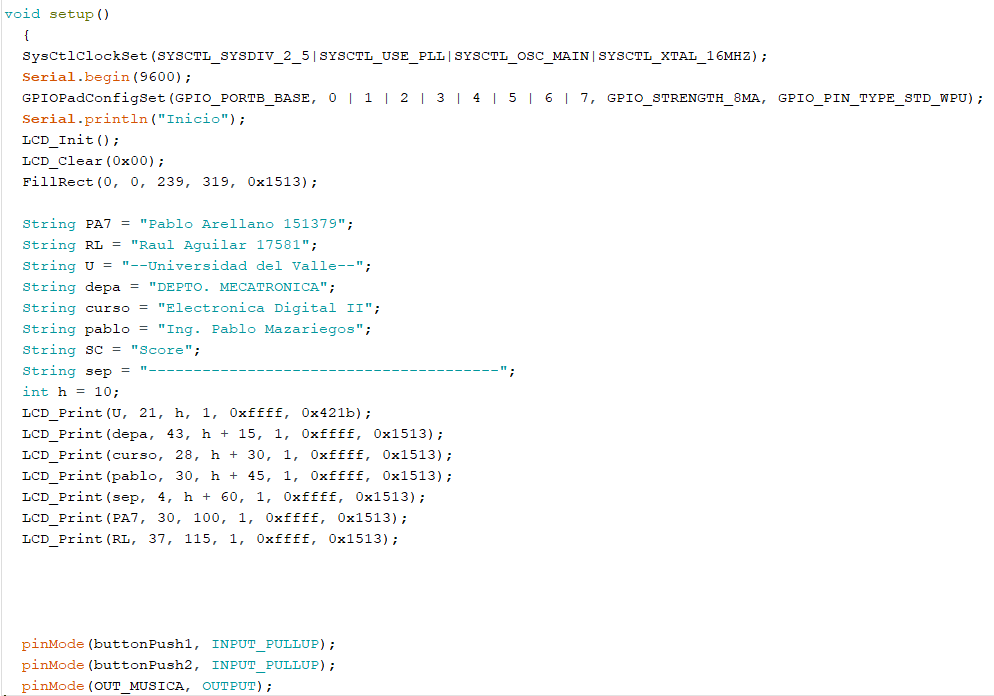
Se eligió flappyBird ya que se llegó a la conclusión que era completamente implementable con las condiciones dadas. De Igual forma se observó que se podía cubrir con todos los requerimientos estipulados en la guía del proyecto, además de ser un juego divertido y hasta cierto punto retador para los usuarios.

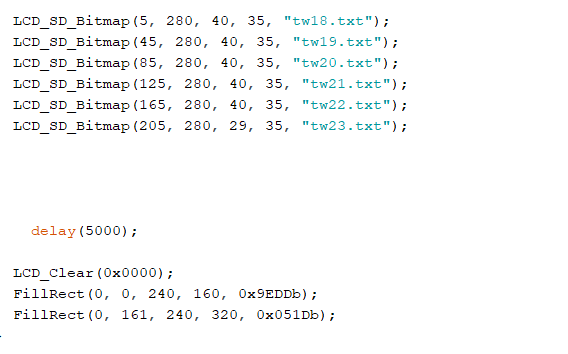
Para poder llevar a cabo el Flappybrid se utilizó la plataforma energía en la Tiva-C, se utilizaron las respectivas librerías para la correcta inicialización de la pantalla LCD ILI9341 en energía.

## **Estructura de la programación TIVA-C Uno**

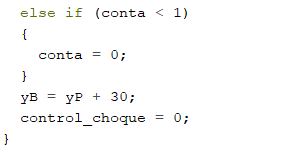
1. **Inicio**- Bienvenida al jugador:
   * 1. Configuración de reloj.
     2. Inicialización de la Tiva-C.
     3. Títulos de los diseñaros del juego.
     4. Títulos generales (Universidad, curso, departamento)
     5. Se llama a las imágenes de portada en la SD.
2. **Loop principal**:
   * 1. Se pone en 1 el pin OUT\_MUSICA.
     2. Se define la condición de el valor máximo de Score.
     3. Si se cumple se manda a llamar a gameover.
     4. Se define las coordenadas de los pilares.
     5. Se dibujan los pilares.
     6. Se define la caída y la aceración de caída del ave.
     7. Se define la condición de generación de los pilares.
     8. Se define la altura random de los pilares superiores e inferiores.
     9. Se lee los botones.
3. **Drawbird:**
   * 1. Se define la función de aleteo.
     2. Se define en las coordenadas que se mostrara el Sprite del bird.
     3. Se definen condiciones de vuelo del ave.
     4. Se define cuando el bird pasa entre los pilares.
4. **DrawPilares:**
   * 1. Se define el Score de los jugadores.
     2. Se muestra un cuadro celeste en donde se visualizarán.
     3. Se definen pilares arriba y debajo de la pantalla.
     4. Se definen pilares del mismo color que el fondo.
     5. Se define la generación de pilares de colores verde y atrás color del fondo.
5. **Gameover:**
   * 1. Se vuelve a definir la condición de el puntaje máximo en los dos jugadores.
     2. Si uno de los dos jugadores gana, el pin OUT\_MUSICA se pone en 0.
     3. Se muestra texto definiendo el jugador que ganó.

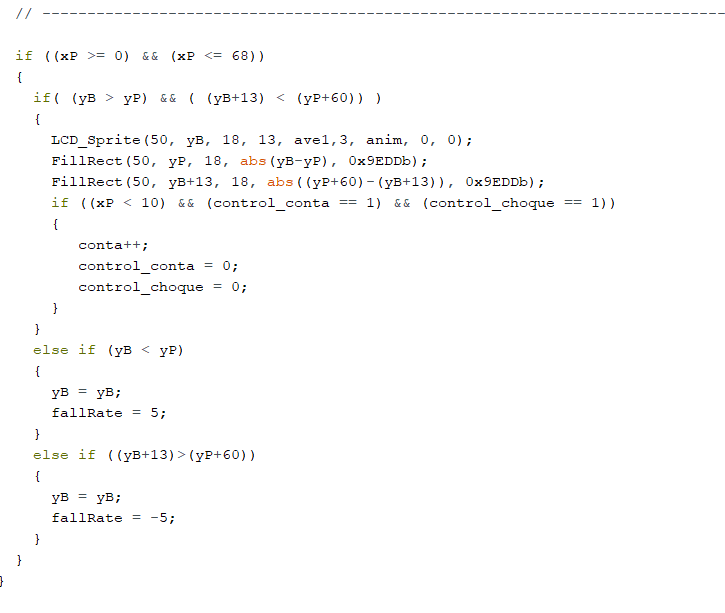
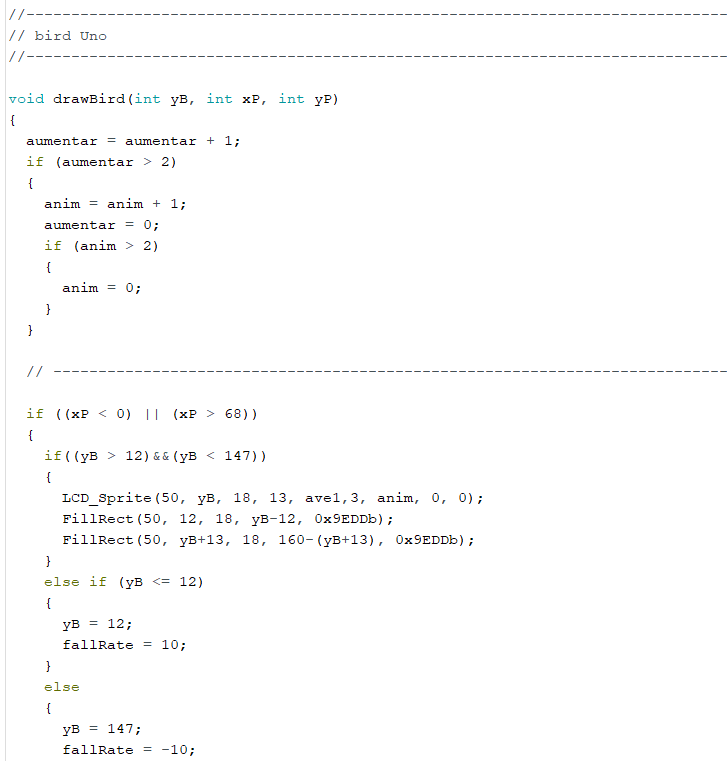
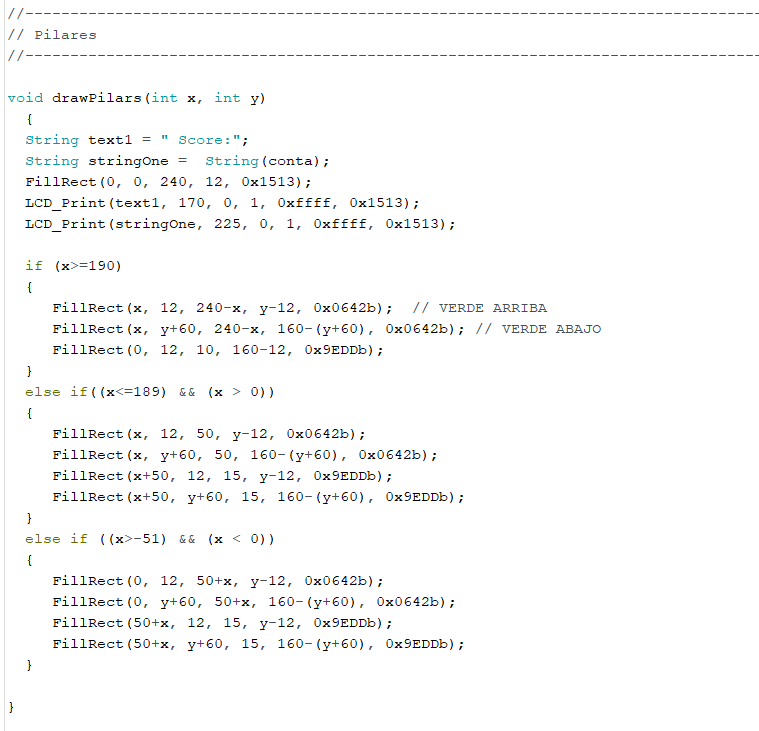
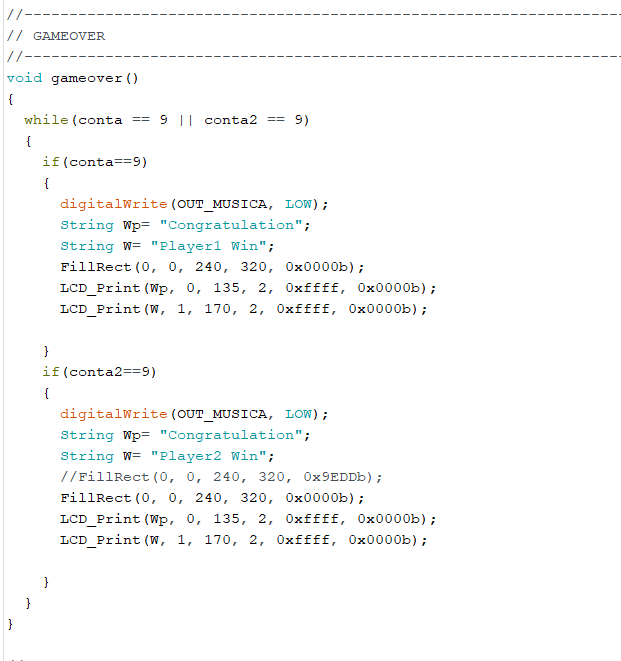
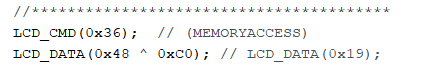


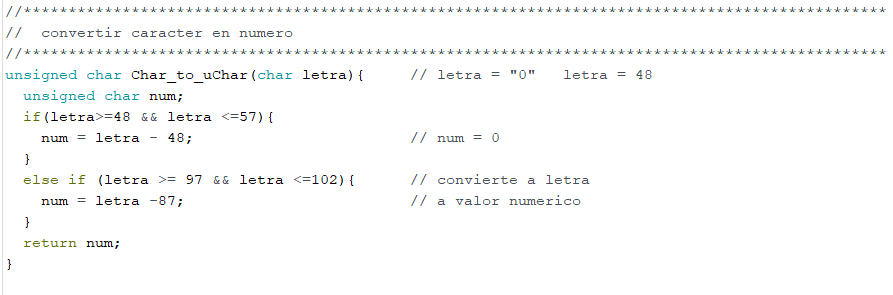
* + 1. **Setup.** Se inicializa la pantalla, se muestra la portada con datos generales del proyecto. Se le incluyen imágenes de bienvenida llamadas desde la SD. Se configuran los botones en pull-up.

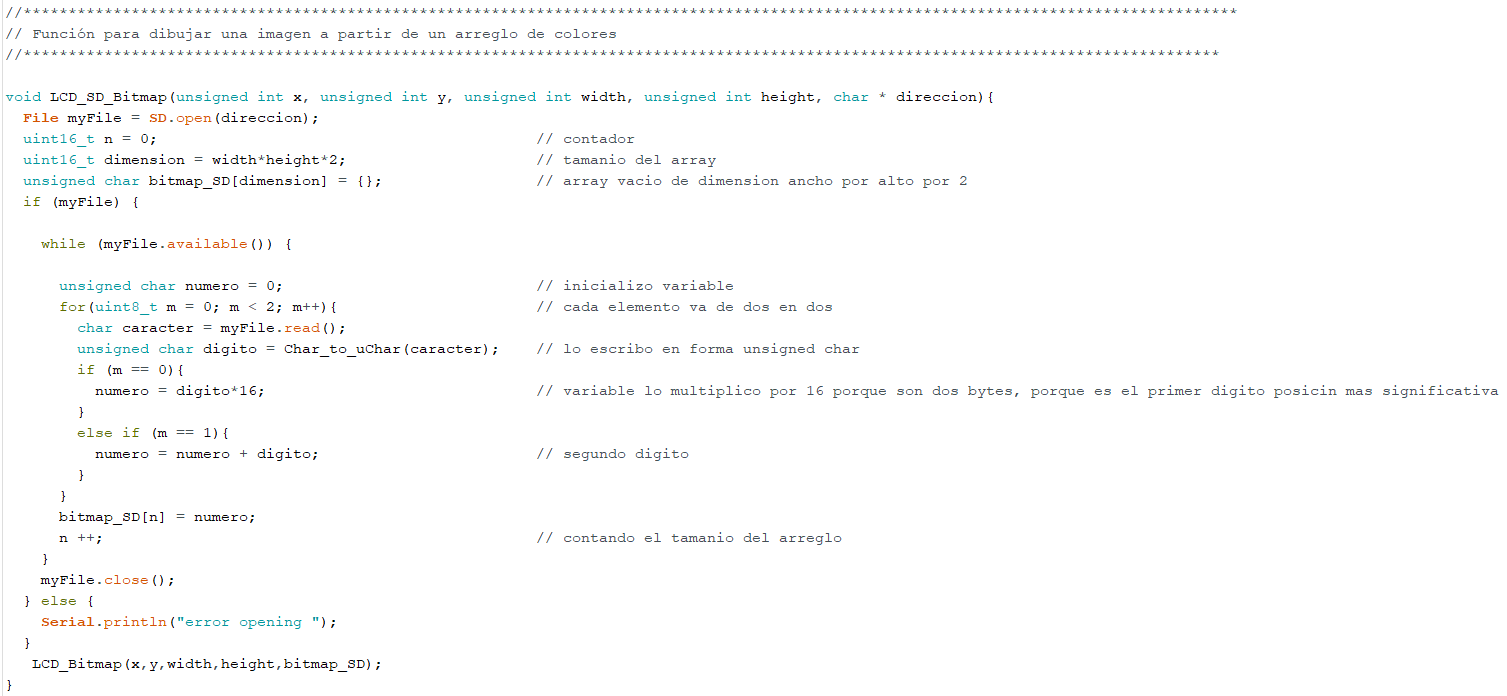
Se detalla a las coordenadas que se quieren las imágenes recortadas del Bird de bienvenida.

* + 1. **Loop.** Se pone en 1 el pin OUT\_MUSICA. El cual da inicio a la melodía de Gourmet Race. Se define la condición del valor máximo de Score. Si se cumple se manda a llamar a gameover. Se define las coordenadas de los pilares. Se dibujan los pilares. Se define la caída y la aceración de caída del ave. Posteriormente se define las condiciones de generación de los pilares. Se define la altura random de los pilares superiores e inferiores y se lee los botones de la Tiva-c. De ultimo se detalla el choque de bird con los pilares y su decremento al score.



* + 1. **drawBird.** En general esta función define los parámetros en el cual se dibujará el ave. Se estipula que el ave estará en una coordenada (50, yB). yB ira variando. De igual forma se restringe su plano de movimiento y se detalla su moviente entre los pilares y el aumento al Score.
    2. **drawPilar.** Aquí se programa el Score acendente de los jugadores y su vizulisación en la pantalla. Posteriormente se detalla el dibujado de pilares superiores e inferirores en diferntes coordenadas y se automatico dibujado del mismo color del fono para dar el efecto de movimiento.
    3. **gameover.** Función que permite la correcta finalización del videojuego.
    4. **Vertical Screen:** Cabe resaltar el siguiente registro que se uso para poder visualizar la pantalla en vertical con el comando 0x36 y luego se envia el dato 0x48 ^ 0xC0.
    5. **Carácter a Numero:** Otra función importante es la siguiente. En la que se detalla la conversión de un carácter en número para poder visualizar las imágenes en la SD.



* + 1. **Dibujar de SD:** El siguiente código permitió dibujar las imágenes guardadas en la SD por medio de un arreglo de colores.

## **Estructura de la programación TIVA-C 2 (Música)**

Se agrega la librería <pitches.h>

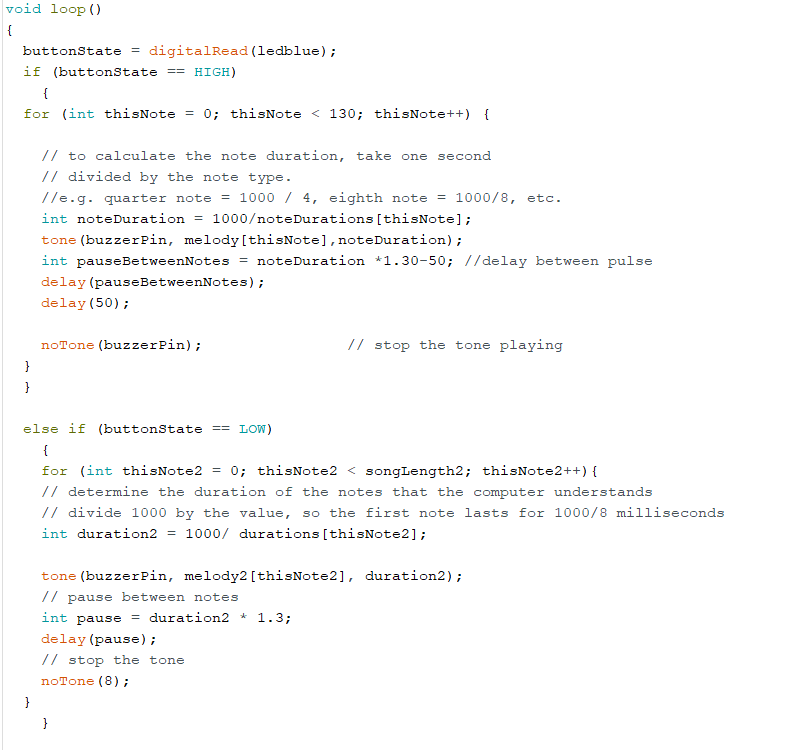
Se definen las notas de la melodía.

Se define la duración e la notas.

Se define el largo de la canción.

Loop.

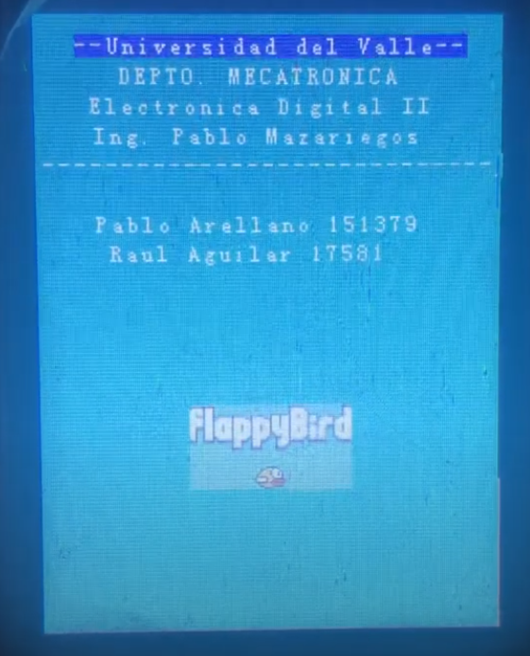
1. Se define la duración de las notas.
2. Se define las pausas entre las notas.



## **Proyecto Terminado:**

Imagen que contiene computadora

Descripción generada automáticamente



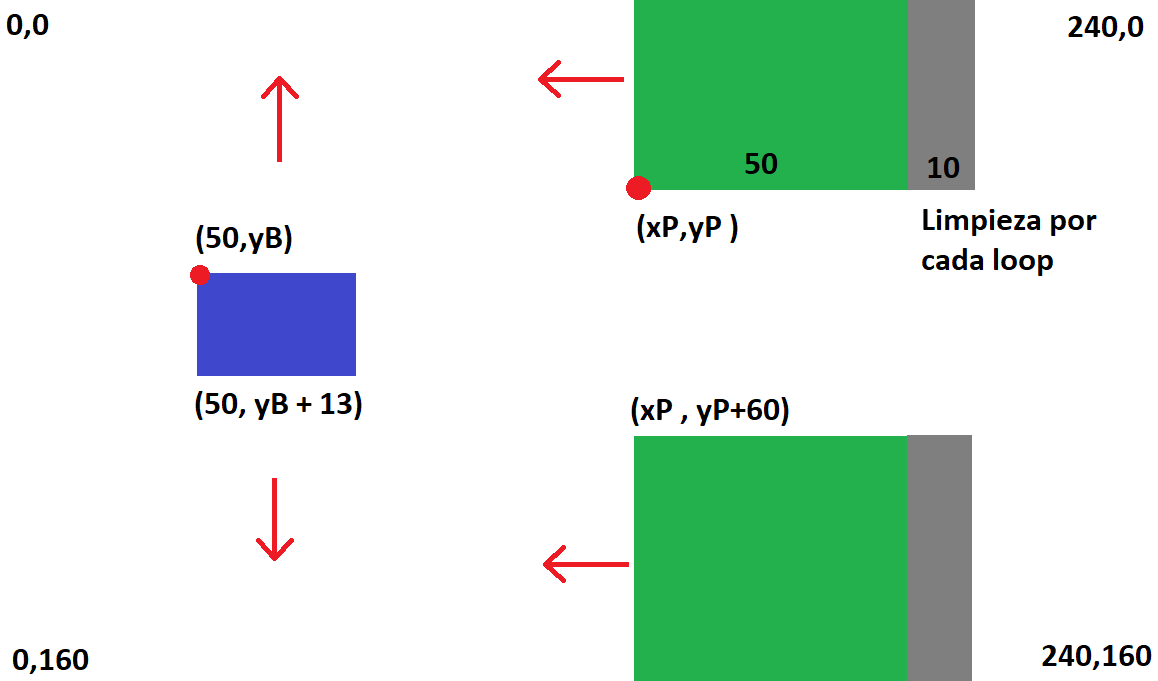
Pantalla de computadora con fondo azul

Descripción generada automáticamente con confianza mediaPantalla de computadora con fondo azul

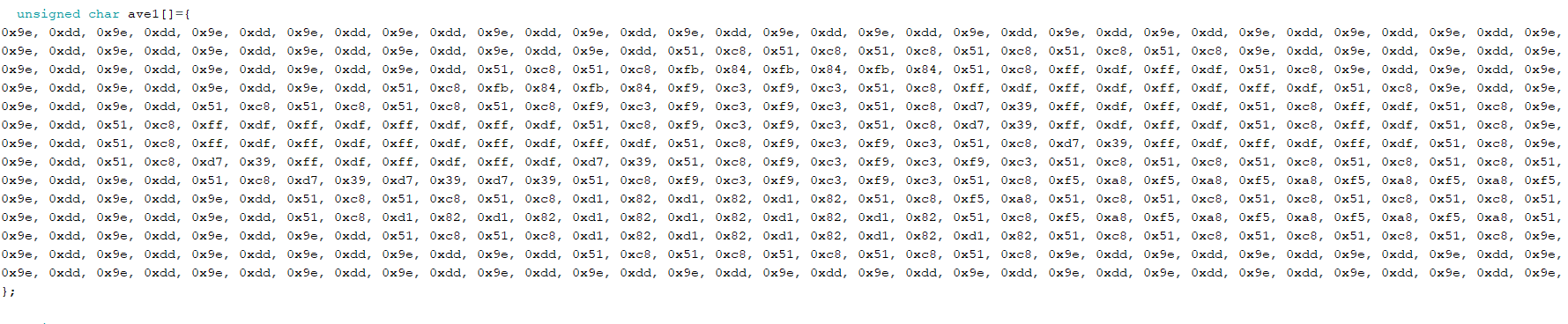
Descripción generada automáticamente con confianza media

## **Anexos:**

**Diagrama para Posicionamiento de Elementos**



**Bitmaps en Tiva**



**Archivos en la SD**

