

Bitrix

Trabajo de instalaciones industrias

Nombre | Nombre del curso | Fecha

# Introducción

Este proyecto propone crear un Tetris “de un bit”, a diferencia del Tetris común, este no tendrá formas complejas como la L,T, el cubo o el palo; únicamente dispondrá de un único punto (un led) que ira de izquierda a derecha en lo más alto de la pantalla, una vez el jugador apriete un botón, se dejará caer este bit a lo más profundo de la pantalla, el objetivo es que el jugador vaya alineando horizontalmente bit por bit como se muestra en el siguiente gráfico.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| 1 |  |  | 9 |

Una vez la fila es completada el jugador pasara de nivel y deberá proseguir a llenar la siguiente fila, a medida que el desplazamiento del bit superior va aumentando de velocidad.

El desafío del juego se encuentra en que el jugador **NO** debecomenzar a llenar la siguiente fila sin haber completado la anterior (ejemplo en el grafico debajo). Si esto llegara a suceder el jugador perderá una de sus 3 vidas.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | |  | |  | |  | |
|  |  | |  | |  | |  | |
|  |  | |  | |  | |  | |
|  |  | |  | |  | |  | |
|  |  | |  | |  | | El jugador comenzó a completar la 2da | |
|  |  | |  | |  | | Fila sin completar la 1ra | |
|  | |  | |  | |  | |
| 0 | |  | |  | | 8 | |

# Características

## Software

El programa del juego será el encargado de realizar el funcionamiento lógico del juego, en la misma ira tanto la configuración del juego (vidas, niveles, velocidad, etc)

Como el lógico del resto del hardware.

### Configuración del juego:

Vidas: determina la cantidad de vidas que tiene el jugador

Selector de nivel: determina en que nivel se encuentra el jugador

Nivel: es un vector que tiene una cantidad de elementos igual a la cantidad de filas de la matriz, cada elemento tiene un valor especifico que determina la velocidad del juego en base a los ciclos que realiza el oscilador del uC.

### Control lógico del hardware

Display: determina que numero muestran los Display de 7 segmentos tomando el valor de las variables Selector de nivel y Vida.

Multiplexado: es una variable boolean que se utiliza para multiplexar los Display

Filas: posee la altura (cantidad de filas) de la matriz y en base a ella se determina la cantidad de niveles del juego, se usará para el control de la matriz y mostrar animaciones junto con Columnas

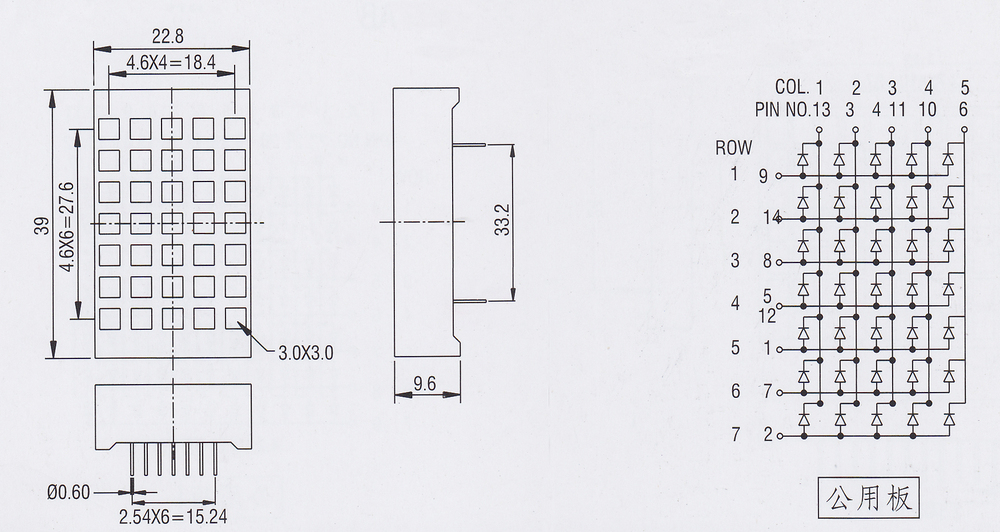
Columnas: posee el ancho (cantidad de columnas) de la matriz y se usará para el desplazamiento del jugador, y determinar en qué posición se encuentra, también se usará para mostrar animaciones junto con Filas

Sonido: se utiliza para realizar las diferentes melodías del juego

## Hardware:

Posee 3 partes principales ara el funcionamiento del juego, obviando la alimentación.

### Matriz LED

Se trata de una matriz led de 6x4 la cual será la pantalla del juego por lo cual se trata de la parte mas importante del juego ya que sin ella el jugador no tendría idea de lo que sucede

### Displays

Son dos displays de 7 segmentos ubicados en los extremos del frente del gabinete, uno mostrará las vidas restantes mientras que el otro se encargará de mostrar en el nivel está el jugador, se controlaran mediante un CD4511 que traduce de binario a 7 segmentos y además mediante dos transistores con el fin de multiplexar los valores que mostrará cada display

### Buzzer

Es un piezoeléctrico que se encargará de darle mas “vida” al juego, emitiendo las melodías dependiendo la situación del juego, necesita amplificador?

# Problemáticas

## Consumo

Al haber de por medio una matriz (la cual contiene 24 leds), 2 displays (cada uno de 7 leds) y un parlante piezoeléctrico que deben ser controlados por un uC se debe tener especial cuidado con la corriente que todo esto llegue a exigirle al integrado ya que este es de baja potencia

### Solución

El principal problema se encuentra en la gran cantidad de led a manejar, ya que idealmente el ATMEGA no puede entregar más de 200 mA. La solución mas viable encontrada es el multiplexado para el control de la matriz.

Utilizando esta técnica, la matriz tendrá el consumo equivalente a un solo led a pesar de que visualmente pareciera que la matriz estuviese completamente prendida si así se deseara.

Por otro lado, el uC pasará a ser solo la entrada de un amplificador que se encargará de amplificar la señal para el piezoeléctrico

El control de los Displays se realizará utilizando un 4511 y multiplexado, no solo para la problemática del consumo si no que también ayuda con otra problemática, la cantidad de pines.

## Cantidad de pines

En principio necesitaríamos:

* 7 pines por display (x2)
* 10 pines para la matriz
* 1 para las melodías
* 1 para el botón del jugador

El problema esta en que el ATMEL solo dispone de 20 pines para uso del usuario y según la lista anterior, se necesitarían unos 26 pines lo cual ya a primera vista puede dejarse ver el conflicto, no solo eso, también que hay ciertas partes que necesitan estar vinculados a un pin especifico, como el piezoeléctrico, debe ir conectado a un pin de PWM y el botón del jugador, para una mejor precisión de lectura debe ir conectado a un pin de interrupción externa

### Solución

Como se mencionó en el apartado de consumo, los displays se controlarán con un CD4511, el cual es un integrado que se encarga de procesar la señal del microcontrolador (en binario) para mostrar el numero enviado en el display. Esto reduciría la cantidad de pines requeridos por los displays de 14 a 8, y gracias a la técnica de multiplexado este número decrece a solo 6 pines; 4 para enviar el numero en binario y 2 para alternar (multiplexar) que numero prender.

Esta problemática también fue la principal razón por la que se decidió utilizar una pantalla matricial ya que si cada led tuviera su propio pin dedicado se necesitarían 24 pines solo para los leds.

## Alimentacion

A la hora de plantear el diseño del esquemático se debió considerar que niveles de tensión manejaría el proyecto.

El principal foco esta en el microcontrolador ya que estos componentes suelen ser los mas delicados y trabajan en angostos rangos de tensión (3,3v a 5,5v).

También se tuvo en cuenta la potencia que se buscaría obtener en el amplificador, esta debería ser suficiente como para ser escuchada en un ambiente de salón de juegos o la muestra escolar.

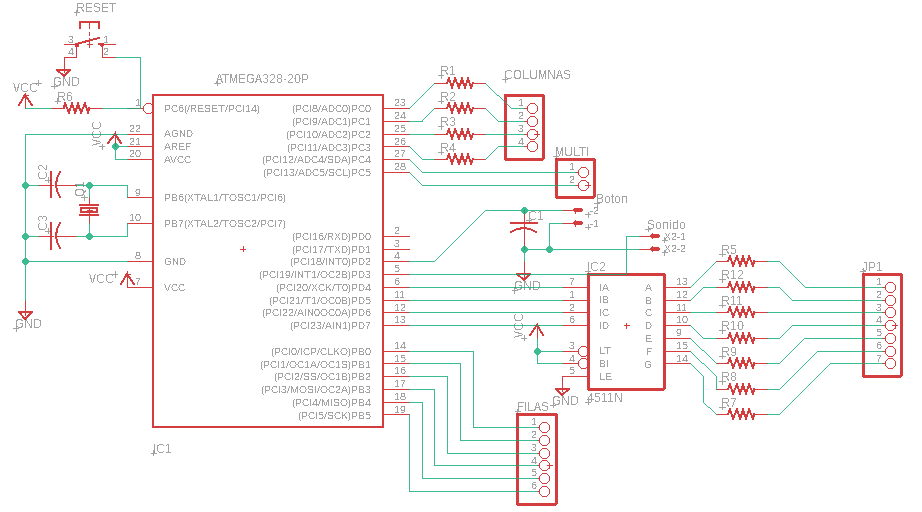
Teniendo en cuenta que el ATMEGA 328p se maneja con 5v, determinamos que esta tensión seria ideal para el control lógico pero no para el circuito de amplificación ya que necesitaríamos relativamente grandes niveles de corriente para que tenga una potencia ideal. Es por esto que decidimos utilizar una tensión principal de 9v y con un regulador de tensión de 5v que derive la tensión correspondiente para el microcontrolador.

El 4511 es despreciado ya que este trabaja en un amplio rango de tensión (3v a 18v).

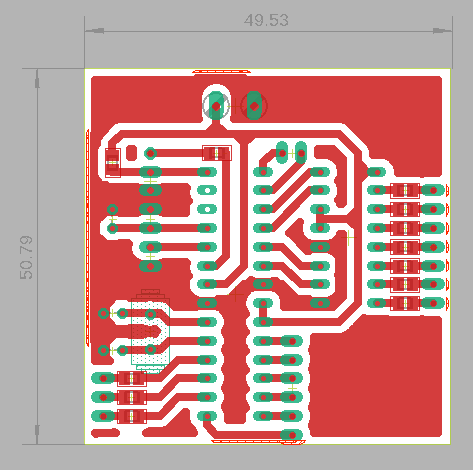
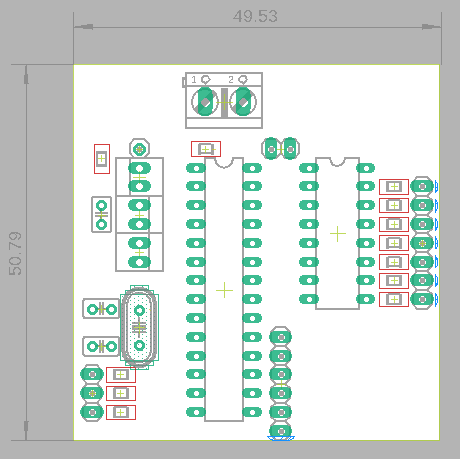
## Diseño del Esquematico y PCB

En base al código y las pruebas realizadas en la placa de desarrollo Arduino UNO se planteó un esquemático guía de como se realizarán las conexiones en la placa principal la cual consiste principalmente del ATMEGA 328p y el 4511 junto con los pines que transmitirán las señales a las diferentes plaquetas y partes del sistema, los displays 7 segmentos y la matriz led como ejemplos.

Teniendo en cuenta lo mencionado se realizó este esquemático:



Posteriormente se procedió a diseñar el PCB, en el cual se tuvo en cuenta los siguientes parámetros:

* Tamaño, si bien no es tan importante en el diseño de un prototipo, si las plaquetas son mandadas a producirse de forma masiva, el tamaño pasa a ser un parámetro crucial ya que mientras mas pequeña sea la el diseño, mas cantidad de placas se podrán realizar por oblea, siendo importante para el precio final del producto.
* Comodidad, como la plaqueta tiene varias conexiones con otras partes del proyecto estas serán realizadas por conexión de pines por lo que estos deberán ser posicionados en un lugar donde sea fácil de encastrar o desencastrar.
* Estética, se trató de mantener los componentes lo más alineados y paralelos posibles para conservar un diseño estético y fácil de entender.

# Seguimiento del proyecto

Todo avance desde el comienzo y conceptualización del proyecto fue siendo agregado a un repositorio de GitHub donde se puede acceder a toda la información de parte por parte del programa, esquemáticos y PCB e incluso este informe

El link al repositorio se encuentra [aquí](https://github.com/ljcmps01/Bitris)

## Contenidos

* .../BitrisPCB: Contiene los archivos relacionados con el diseño de las plaquetas del proyecto (Esquematicos y PCB)
* …/Codigo: Contiene todos los codigos utilizados, incluyendo el principal (BitrisUNO)
* …/Diseño del gabinete: Contiene el diseño en CAD del gabinete del juego
* …/Pruebas del código: Videos del progreso del código
* …/Informe del Bitrix: Este mismo documento
* …/README: Un archivo .txt con un historial de cada actualización del repositorio