Toni Tormo Pla y Abel Vidal Ripoll

1. Introducción

En este trabajo vamos a crear un programa que reproduzca el clásico juego del Tetris, para ello utilizaremos el software de Arduino y el lenguaje C.

Para el montaje y la posterior ejecución del programa utilizaremos una tarjeta Arduino MEGA 2560, junto con diferentes elementos de kit de inicio ELEGOO Mega 2560.

De este kit cabe recalcar que usaremos una Matriz de leds de 32x8 construida a partir de cuatro módulos de 8x8 leds basados en el controlador MAX7219 con bus serie SPI y su recomendada librería MD_MAX72XX. También utilizaremos el joystick proporcionado por este kit.

2. Objetivos

El objetivo de la práctica es la realización de un trabajo final que utilice, desde un punto de vista práctico, todo lo aprendido en la asignatura.

3. Desarrollo

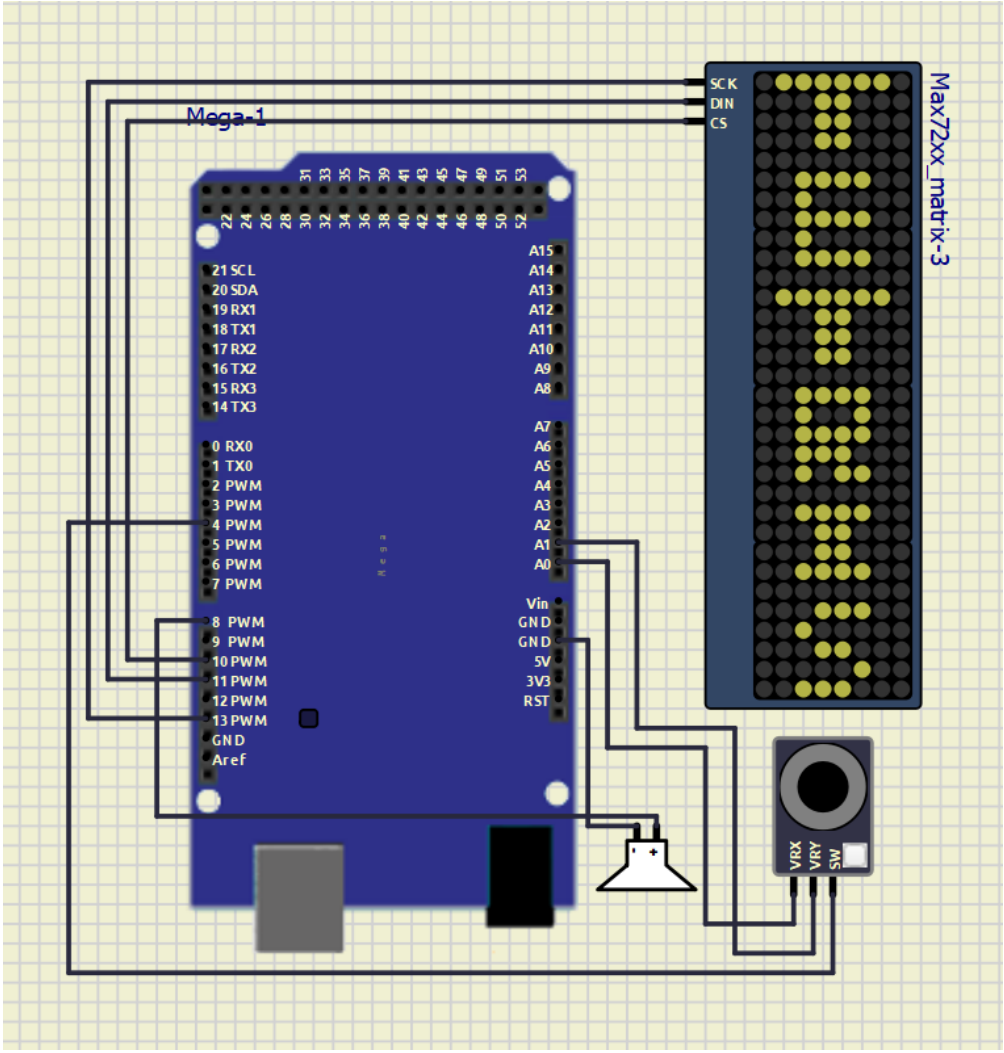
Nuestro Tetris consiste en una Matriz de 32x8 leds donde el primer modulo de 8x8 leds se utiliza para la previsualización de la pieza que tendremos una vez colocada la que esta en juego, y los otros tres módulos restantes se utilizan para la ejecución del juego de Tetris.

La matriz funciona únicamente como pantalla, para poder mover las piezas del juego por ella, nos hacemos valer de un joystick. Este ofrece tres posibilidades en el juego, en su eje x mueve las piezas a izquierda y derecha; en su eje y las mueve hacia abajo por si se busca una mayor velocidad en la caída; y su pulsador que tiene la labor de iniciar el juego, y dentro de este hace que la pieza rote.

A parte, el proyecto también consta de un zumbador, el cual se empieza a reproducir al presionar el pulsador del joystick, y hace que se escuche la clásica melodía del Tetris mientras el juego está en ejecución, una vez el jugador pierde, la música cesa.

3.1. Esquema de montaje de la tarea (realizado con SimulIDE o Wokwi).

Aquí podemos ver como seria el montaje de nuestro programa realizado en SimulIDE:



3.2. Algoritmo de la tarea (realizado con diagrama de flujo o pseudocódigo).

Para el algoritmo de nuestro programa lo vamos a dividir en varias partes, ya que hemos utilizado diferentes “.h” en el mismo “.ino”.

Sprites.h:

```
Incluir la biblioteca Arduino.h

Definir la estructura de los sprites
Función para realizar copias de los sprites

Valores para la Pantalla de inicio

Valores para la Pantalla de juego

Valores para cada una de las piezas en sus 4 rotaciones
```

Musica.h:

```
Incluir la biblioteca Arduino.h

Creamos las estructuras para las notas y los tonos
Asignamos las frecuencias correspondientes para cada nota

Creamos el tema principal (el Tetris theme)
Creamos el acompañamiento (no se ha podido implementar)

Guardamos todas las direcciones de cada melodía
```

Reproductor.h:

```
Incluir la biblioteca Arduino.h

Creamos las estructuras del reproductor y cada pista (solo 1 al final)
Funciones para asignar todos los valores del reproductor

Función para testear el correcto funcionamiento
Función para reproducir una melodía {

      Comprobamos que el tiempo transcurrido es el suficiente
      Comprobamos si hace falta cambiar de nota

      Recogemos la nueva nota de la melodía

      Si esta tiene el símbolo del final, se recoge la primera

      Asignamos al pin la frecuencia correspondiente a la nueva nota
      Indicamos al iterador que hemos avanzado en la melodía
}
Función para parar el reproductor
```

Controladores.h:

```
Incluir la biblioteca Arduino.h
Crear la estructura de los pines
Crear la estructura de las variables del cursor

Función guardar pines del joystick
Funciones para conocer si se interactúa con el joystick

Funciones para transmitir la información a “game.h” y para restablecer las
variables del cursor
```

Pantalla.h:

```
Incluir la biblioteca Arduino.h
Incluir la biblioteca MD_MAX72xx.h

Definimos los valores de la pantalla

Función de borrado de pantalla
Función de inicialización

Función para imprimir la pantalla de inicio
Funciones para leer escribir o borrar en la pantalla

Función para hacer un barrido hacia abajo
```

Game.h:

```
Incluir la biblioteca Arduino.h

Definimos las fases del juego
Definimos la estructura para las piezas
Definimos las variables del juego

Funciones para inicializar el juego o cambiar sus valores

Función para comprobar el movimiento horizontal de los sprites
Función para mover las piezas horizontalmente
Función para comprobar las colisiones entre piezas

Función de nueva pieza {

	Asignamos la próxima pieza a la actual
	Damos un valor aleatorio para la próxima pieza

	Imprimimos la próxima pieza en el cuadro de la pantalla

	Configuramos la pieza para colisiones
	Comprobamos si se puede situar la nueva pieza

	CONDICION: ¿Se puede situar la pieza? {
		Asignamos la pieza de colisiones a la actual
		Imprimimos la pieza en la pantalla

		Cambiamos el estado de juego a MOVIMIENTO
		Retornamos verdadero para que continue el juego
	}
	Si no, fin del juego.
}

Función de movimiento {

	Recogemos los valores de los controles
	Movemos la pieza de colisiones según los controles
	Borramos la pieza actual de la pantalla

	Condición: ¿se puede situar la nueva pieza? {
		Asignamos la pieza de colisiones a la actual
	} NO {
		Cambiamos el estado de juego a COMPROBACION
	}
	Imprimimos la pieza actual en pantalla
}

Función de comprobación {

	Bucle:
		Leemos cada fila de la pantalla
		CONDICION ¿Está completa la fila? {
			Borramos la fila entera
			Cambiamos el estado de juego a CAIDA
			Salimos de la función
		}
	Cambiamos el estado de juego a NUEVA PIEZA
```

```
}
Función de caída {

	Cambiamos el estado de juego a COMPROBACION

	Bucle:
		Leemos cada fila de pantalla
		CONDICION: ¿La fila está vacía? {
			Hacemos un barrido hacia abajo a partir de la fila
			Salimos de la función
		}
}

Función de juego {
	Comprobamos el tiempo transcurrido para ejecutar el frame
	Comprobamos el estado del juego
	Ejecutamos la función correspondiente al estado
}

Función de reset del juego.
```

Tetris.ino:

```
Definimos los valores de los pines
Definimos los FPS y el estado de juego

SETUP {
	Inicializamos el Serial
	Inicializamos el reproductor
	Inicializamos los controles
	Inicializamos el juego
	Inicializamos la pantalla

	Imprimimos la pantalla de inicio
}

LOOP {
	Recogemos información de los controles

	CONDICION: ¿El juego está en marcha? {
		Ejecutamos el reproductor
		Ejecutamos el juego

		CONDICION ¿Se ha acabado el juego? {
			Paramos el reproductor
			Reset del Juego
			Borramos la pantalla
			Imprimimos la pantalla de inicio
		}
	} NO {
		Leemos el botón de inicio

		CONDICION ¿Se ha iniciado el juego? {
			Borramos la pantalla
			Imprimimos el cuadro de juego
		}

	}
}
```

4. Conclusiones

La conclusión a la que llegamos es que como ya hemos visto anteriormente, la matriz de leds es un dispositivo muy versátil y útil para la realización de proyectos creativos, ya que nos permite crear cualquier juego clásico que no requiera una cantidad de pixeles muy grande.

También hemos aprendido a como de útil puede ser un joystick para este tipo de juegos. Y como los diferentes componentes funcionan en armonía para crear algo tan entretenido.

Los retos que hemos encontrado han sido variados:

El mayor de todos ha sido hacer una ejecución “paralela” de la música, la pantalla y con el máximo muestreo posible, la medida del joystick. Todo esto sin ejecutar hilos.

Las piezas trajeron varios problemas. Uno de ellos ha sido hacer que las piezas se desplazaran correctamente hacia los lados, y crear una barrera para que las piezas no saliesen de la pantalla. Otro hacer que se detectase correctamente el espacio que tenía para moverse.

Los controles también nos dieron guerra. Sobre todo, a la hora de conservar y hacer los reset de los valores para cada instancia del juego.