

Arquitectura de computadoras

Informe Trabajo Práctico Especial - Arduino

Alumno: Juan Gabriel Griggio

Legajo: 59092

Profesores: Horacio Merovich y Santiago Vallés

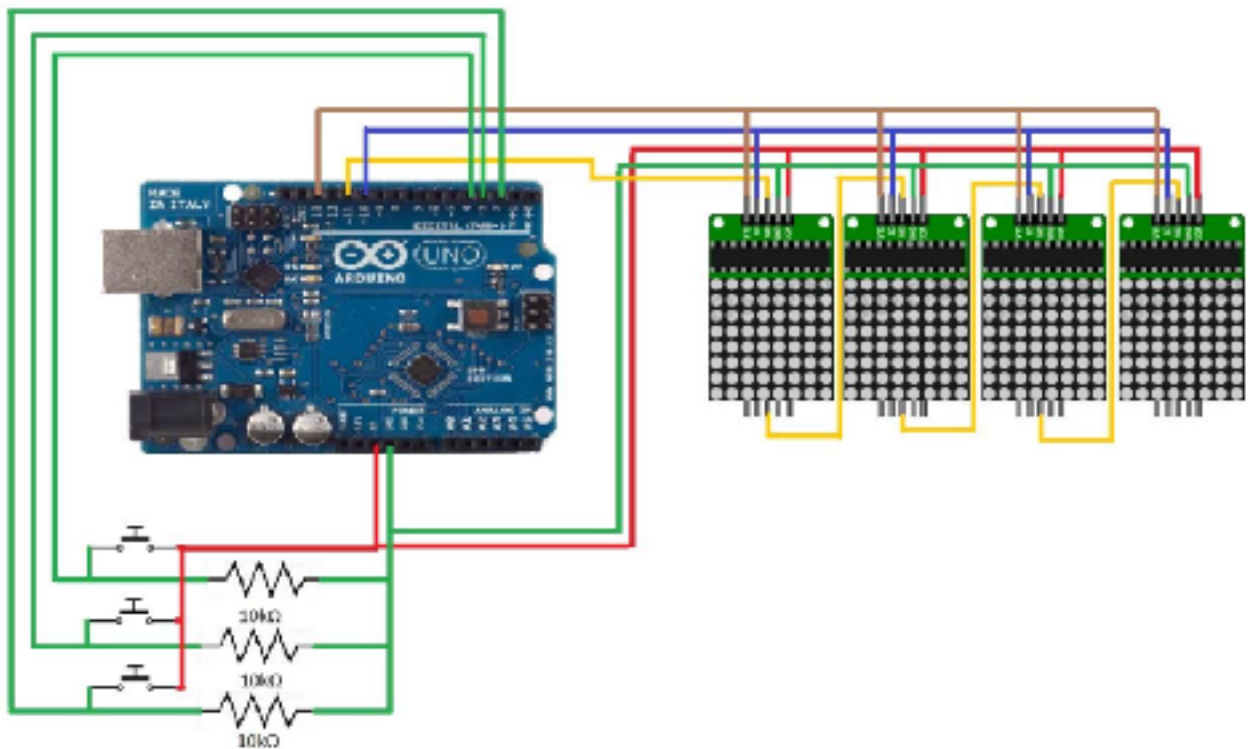
Fecha de entrega: 01/12/2019

Enunciado:

- Implementar el juego Arkanoid en una placa Intel Galileo usando matrices de LED 8x8 de forma tal que conformen una matriz de 16x16 leds, y usando switches para comandar los movimientos de la paleta.
- Cada 15 segundos la velocidad de la pelota deberá aumenta emitiendo un sonido de aviso.
- Al finalizar el juego deberá emitir también un sonido y desplegar el tiempo transcurrido.
- Mediante alguna de los pines de entrada deberán implementar una interrupción que permita suspender el juego y permitir que desde la interface Usb -serie responda a los comandos linux ingresados desde la interface usb serial .
- Implementar un comando que ingresado desde la interface usb serie despliegue la hora y minutos en las matrices de leds.
- Desde la interface usb serie se deberá poder cambiar parámetros o resultado del juego.
- La señal de interrupción deberá también permitir retomar el juego en el estado en el cual fue interrumpido.
- Dado la limitación de la cantidad de filas en las matrices de LEDs, las líneas de bloques se limitarán a 1.

Diseño elegido: hardware

Se armó el siguiente circuito en una protoboard:



Las matrices están conectadas de la siguiente manera:

- Los pines LOAD, CLOCK, VCC Y GND de las matrices están conectados en paralelo a los pines 13, 10, +5V y GND del Galileo respectivamente.
- Los pines de datos de las matrices están conectados en serie: DIN de la primera matriz está conectado al pin 11 del Galileo, su DOUT al DIN de la segunda matriz y así sucesivamente.

Para poder interactuar con la placa usamos tres pulsadores conectados con tres resistencias de 10k ohms en configuración pull-down (conectadas a GND) para que no se registren pulsaciones erróneas. Los pulsadores se conectaron a los pines 2, 3 y 4.

Elegimos el pin 2 para las interrupciones, ya que Galileo trabaja con interrupciones con los pines 2 y 3, que son sus IRQ.

A la vez, la placa debe ir conectada por USB a la computadora, cuyo monitor serie debe estar configurado en 115200 baudios y con la opción “retorno de carro” activada para poder interactuar correctamente a través del puerto serie.

Como no tengo un parlante para poder reproducir los sonidos necesarios, estos se reemplazan por mensajes enviados por el puerto serie, detalle menor que es muy fácil de corregir.

Pros y contras: hardware

La principal ventaja del diseño elegido es que es muy fácil de armar el circuito, requiere pocos componentes y no es costoso. Tampoco requiere soldar, ni grandes conocimientos de electrónica.

Una gran desventaja es que al haber conectado todo en una protoboard no hay forma de que queden las matrices en configuración 16x16, por lo que queda la pantalla “dividida en dos”: cuando, por ejemplo, la pelota sube y llega hasta arriba de una matriz, sigue por la matriz que está al lado. Esto es porque el software está bien programado pero la manera de testearlo no es óptima. Si se hubiera diseñado una estructura, por ejemplo impresa en 3D, que soporte al Galileo y las matrices, entonces se podría visualizar mucho mejor el trabajo en las matrices.

Otra desventaja es la velocidad del Galileo al trabajar con matrices de LED: si bien es mucho más potente que un Arduino tradicional, el manejo de matrices de LED lo hace muchísimo mas lento. Esto hace que el juego parezca menos dinámico, más aburrido se podría decir, pero no tiene que ver con el diseño elegido sino con el hecho de usar Galileo en sí.

Otra desventaja más es que al tratar de implementar las interrupciones, Galileo simplemente las ignoraba. Es muy sencillo el código para trabajar con interrupciones, sin embargo Galileo se negaba a atenderlas. Es por eso que se tuvo que forzosamente trabajar la interrupción no como una interrupción en si sino que se tuvo que chequear constantemente si su pin correspondiente estaba activado para atenderla.

Diseño elegido: software

Dividí el software en distintas partes:

Front-end:

- MatrixHandler: funciones para mostrar los datos / el juego por las matrices.
- Matrix: .h que contiene el vector para poder mostrar textos por las matrices.
- SerialCommunicator: funciones para comunicarse con el puerto serie.

Back-end:

- Arkanoid: toda la funcionalidad del juego en sí.
- Constants: todas las constantes del TPE.
- GlobalVariables: todas las variables globales del TPE.
- ParameterModifier: funciones que modifican los parámetros del juego. Podría estar dentro de Arkanoid pero se hizo separado para que quede más organizado.
- Structs: estructuras que usa la pelota del juego. Podría estar dentro de Arkanoid pero se hizo separado para que quede más organizado.
- TimeHandler: funciones para obtener el tiempo actual.
- TPE: archivo principal, con el setup y el loop.

Pros y contras: software

Una gran ventaja es tener el juego completamente separado de las otras funcionalidades, por lo que se puede adaptar a distintas plataformas muy fácilmente.

La principal desventaja es que hay partes que, si bien se puede separar el front-end del back-end, hacerlo implicaría demasiados archivos y demasiada complejidad al proyecto que no es muy grande. Se decidió en esos casos, como por ejemplo con ParameterModifier, no separar front-end de back-end para evitar complejizar demasiado el proyecto. Tampoco se hizo por cuestiones de tiempo. Si hubiera tenido más tiempo, hubiera separado mucho mejor el front-end del back-end.

Había un punto del enunciado que pedía implementar un comando enviado por el puerto serie que permita mostrar la hora y minutos por las matrices de LED. Si bien no es difícil la comunicación con el puerto serie, el hecho de que el Galileo obtenga la hora actual si es un poco más complejo: se necesitaba tener un Real Time Clock conectado y configurado, o la computadora debería tener un Script abierto que interactúe con el Galileo para poder enviarle la hora. Como era demasiado complejo, por cuestiones de tiempo y además porque pensé que capaz el enunciado no iba por ese lado, implementé una función que muestre minutos y segundos de tiempo de juego al elegirse esa opción en la interrupción y al final del juego.

Otro problema encontrado fue que al imprimir caracteres en las matrices de LED, estos aparecen rotados 90 grados. No era difícil solucionar pero era muy largo y no llegué a hacerlo con el tiempo.

Conclusión

El trabajo fue más difícil de lo que esperaba. El código en sí no es complejo pero me llevó mucho tiempo y por ese motivo hubieron pequeños errores que no pude solucionar que me hubieran gustado hacerlo. Por ejemplo mejorar la separación del front-end y el back-end, la rotación de los números en las matrices o el tema de las interrupciones.