PSE. Trabajo práctico ADC y varios periféricos

1- Obtener el código fuente del tp adc y verificarlos.

a. Leer los archivos fuentes del práctico. Entender el archivo cabecera adc.h y observar cómo es utilizado por main.c
b. Agregue un archivo Makefile al proyecto (puede utilizar el de la web modificando los nombres de los archivos del proyecto). Verifique el Makefile con make clean; make

2- Desarrollar un driver (controlador) para el **periférico ADC** del atmega328p, utilizando los archivos propuestos.

a. Comience completando la estructura de datos que hace "overlay" con los registros del hardware del ADC del atmega328. Utilice como tensión de referencia la provista por "vcc con un capacitor externo" (el capacitor externo ya se encuentra en la placa). Una resolución de 8 bits es suficiente.

3- Desarrollar una aplicación que utilice el **Knobs-potenciómetro** para regular el brillo de la pantalla de la notebook.

a. Primero realice un programa de test del knob para enviar por el UART a la PC/notebook el valor obtenido desde el ADC al leer la señal analógica.
b. Utilice el programa brightnessctl en la notebook y amplíe el programa anterior para lograr ajustar el brillo de la notebook a través de este sistema embebido.

4- Desarrollar la misma aplicación anterior pero utilizando la **Fotoresistencia**, de manera tal que al bajar la luz del ambiente el brillo de la pantalla bae automáticamente, y al subir la luz del ambiente el brillo de la pantalla suba.

5- Control del **speaker** para emitir sonidos a distinta frecuencia.

Ampliar el programa para lograr con el knob aumentar la frecuencia del sonido generado por el speaker. Ajuste (sintonice) la frecuencia de lectura del knob (adc) para lograr una percepción del usuario de cambios de frecuencias continuo.

6- Complete el gamepad para controlar alguno de los siguientes juegos arcade Linux (están todos en los repositorios generalmente o en python):

- opentyrian (creo que es el mejor :)
- powermanga
- Chromium-bsu
- arkanoid

- El knob se utilizará para mover la nave izquierda a derecha.
- Un pulsador para disparar.

PSE. Trabajo práctico ADC y varios periféricos

NOTAS adicionales.

Ejercicio 3.b.

```
# by Manuel
sudo cat /dev/ttyUSB0 | while read -n 1 num; do
            brightnessctl s $(($num*100))
      done
#  num*100 porque manda valores entre 0 y 9, y mi monitor maximo acepta 950
```

Una posibilidad es enviar a la notebook valores de 0 a 9 (un byte) que corresponden a diferentes valores del ADC al leer la señal analógica del knob. Son pocos valores (poca resolución). Si desean un ajuste más fino del brillo deberían enviar una mayor cantidad de valores distintos (mayor resolución) y ajustar el script que cambia el brillo. **IMPORTANTE: envíen aproximadamente 1 o 2 valores por segundo unicamente** (no decenas o más), porque el programa que cambia el brillo más el shell leyendo y ejecutando demora aproximadamente un segundo. Si enviamos muchísimos valores por segundo, luego de unos segundos parecerá que funciona mal o que no hace lo que le pedimos. Si eso sucede, en realidad es que tiene valores para procesar durante muchos segundos hasta que consuma todo lo que arribó por el serial.

Es un compromiso balance que deberán sintonizar para todos los ejercicios donde hay comunicación (cuantos datos por segundo envían de un sistema al otro), ya que el que consume y procesa lo que arriba puede tener altas demoras, dependiendo de la acción que tenga que realizar.

Ejercicio 6

Una posibilidad (ustedes pueden implementar otra) es que dividan el rango de valores obtenidos del knob en 3: un rango izquierdo para indicar “movimiento a la izquierda”, un rango central para indicar “sin movimiento”, y un rango derecho para indicar “movimiento a la derecha”.

Del lado de la notebook, si utilizan el “truco” del xdotool, pueden utilizar los argumentos del xdotool keydown y keyup.
Ejemplo:

xdotool keydown s
Inyecta al sistema el evento de que el usuario presionó la tecla ‘s’, pero SIN SOLTAR.

xdotool keyup s
Inyecta al sistema el evento de que el usuario acaba de “liberar” la tecla ‘s’ del teclado.

Entonces, pueden utilizar ese “mecanismo” en reemplazo de simplemente presionar s continuamente (en general, cuando jugamos a un video juego y movemos al personaje, usualmente, no presionamos continuamente la tecla flecha izquierda y soltamos si queremos movernos continuamente a la izquierda. Sino que dejamos la tecla indicada presionada. Eso haría el ejemplo anterior de xdotool).