Práctica 2
Acceso al Hardware

Lluis Ulzurrun De Asanza Sàez
Victor Grau Moreso
Mark Antony Holland

Contenidos

1 PersonalData

Hemos imprimido todos los valores que nos ofrece el struct de PersonalData. Se puede observar en la **Figura 1** que hemos escogido el color verde para los nombres y el gris para los valores. Hemos observado que en el emulador no$gba, no están definidos algunos de los valores como el *nombre* y el *mensaje personal* pero en el emulador DeSmuMe si que existen valores para todos.

Figura 1: Captura del programa PersonalData.

2 Paso del tiempo

2.1 stopwatch

A continuación están las respuestas de los ejercicios planteados en el boletin.

¿Cual es la definición (prototpipo) de esta función?

```
void timerStart ( int          timer,
                  ClockDivider divider,
                  u16          ticks,
                  VoidFn       callback
                )
```

Figura 2: Prototipo de la función timerStart

Busque también la definición de los argumentos segundo y tercero.

ClockDivider_1024 equivale a dividir el reloj por 1024(32.7284 kHz) y el 0 como parámetro para el número de ticks antes de desbordar hace que cada tick de nuestro timer equivale a un tick de reloj.

2.2 timercallback

¿Qué significa la función que se le pasa como tercer parámetro?

Calcula el valor correcto para el desbordamiento de ticks para una frecuencia dada.

¿Cual es la definición (prototpipo) de la función, el cuarto parámetro, que puede ser llamada por el timer?

```
void timerCallBack()
```

¿Cómo se le pasa/recoge información a/desde esa función?

Con la variable global *play*.

2.3 RealTimeClock

¿Qué dos funciones se utilizan para averiguar la hora y la fecha?

Primero llama a `time` para que devuelve un struct con el unix time, después utiliza gmtime para tratar este `time_t` struct y devolver otro struct con el tiempo en un formato más manejable.

Sus prototipos son las siguientes:

`time_t time(time_t *t)` y `struct tm *gmtime(const time_t *timer)`

¿Cómo inicializa este ejemplo el uso de las dos pantallas de la consola?

Con las lineas #103 y #106:

```
lcdMainOnTop(); //del fichero system.h
videoSetMode(MODE_0_3D); //del fichero video.h
```

3 Gestión de entrada

3.1 keyboard/keyboard_async

Proporciona un teclado con una consola donde aparece lo que se teclea.

Figura 3: keyboard_async

3.2 keyboard/keyboard_stdin

Lee del teclado como entrada estandard con un tipico program de hola $(inserta_nombre).

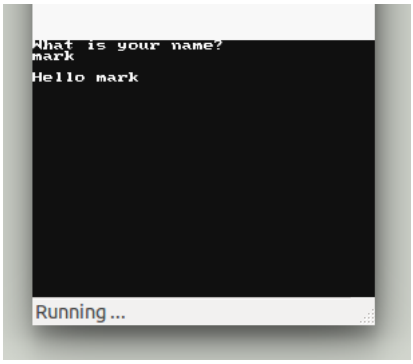

Figura 4: keyboard_stdin

3.3 Touch_Pad/touch_area/

Indice en pantalla que punto de los ejes x/y esta tocando la pantalla el usuario y con que cantidad de fuerza.

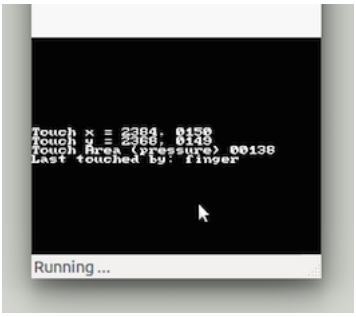

Figura 5: touch_area

3.4 Touch_Pad/touch_look/

Proporciona en la pantalla de arriba un mundo 3D donde podemos "mirar" utilizando la pantalla de abajo deslizando el dedo.

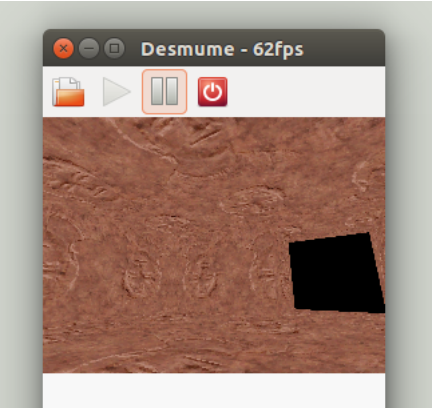

Figura 6: touch_look

3.5 Touch_Pad/touch_test/

Parecido al *test_area* pero proporcionando más información como que botones están siendo pulsados.

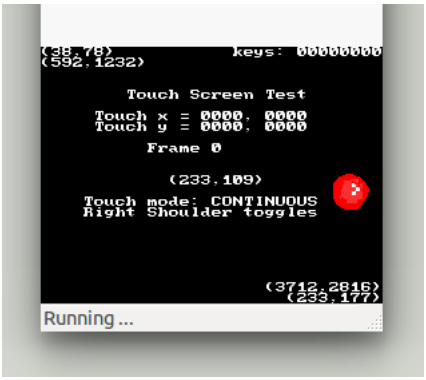

Figura 7: touch_test

4 Trabajo autónomo

Mediante los ejemplos indicados no hubo gran problema para realizar el ejercico, lo único que ha requerido refescar un poco ha sido para mostrar contenido en la pantalla de arriba manteniendo el menu en la de abajo.

A continuación dejamos capturas del programa mostrando el valor de `name` y lo mismo trás darle a la función de `lcdSwap`.

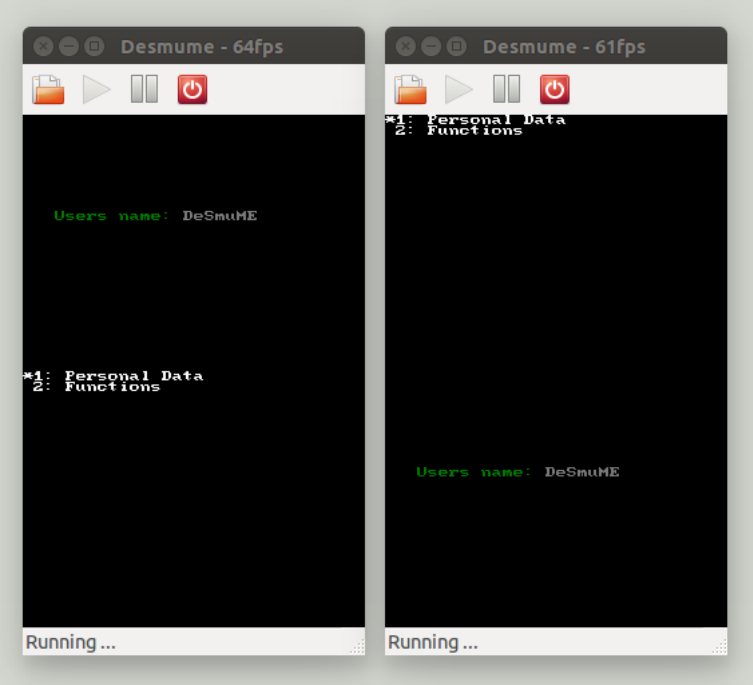

Figura 8: Programa con menu