

# Práctica 2

## Acceso al Hardware

Lluís Ulzurrun De Asanza Sàez  
Victor Grau Moreso  
Mark Antony Holland

### Contenidos

<b>1</b>	<b>PersonalData</b>	<b>2</b>
<b>2</b>	<b>Paso del tiempo</b>	<b>3</b>
2.1	stopwatch . . . . .	3
2.2	timercallback . . . . .	3
2.3	RealTimeClock . . . . .	4
<b>3</b>	<b>Gestión de entrada</b>	<b>4</b>
3.1	keyboard/keyboard_async . . . . .	4
3.2	keyboard/keyboard_stdin . . . . .	5
3.3	Touch_Pad/touch_area/ . . . . .	5
3.4	Touch_Pad/touch_look/ . . . . .	6
3.5	Touch_Pad/touch_test/ . . . . .	6
<b>4</b>	<b>Trabajo autónomo</b>	<b>7</b>



## 2 Paso del tiempo

### 2.1 stopwatch

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

¿Cual es la definición (prototipo) 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 (prototipo) 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 devuelva 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 líneas `#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

Leer del teclado como entrada estandar.

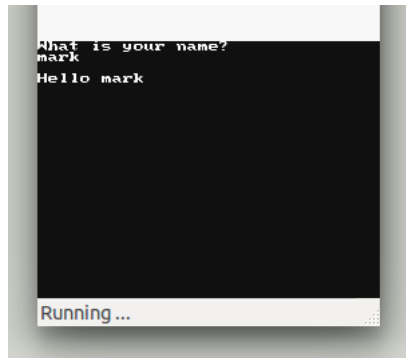


Figura 4: keyboard\_stdin

## 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.

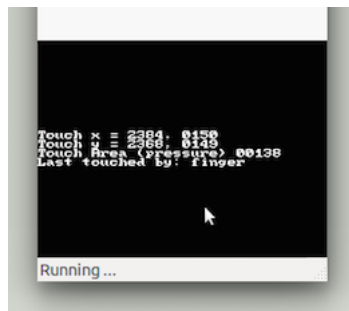


Figura 5: touch\_area

### 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.

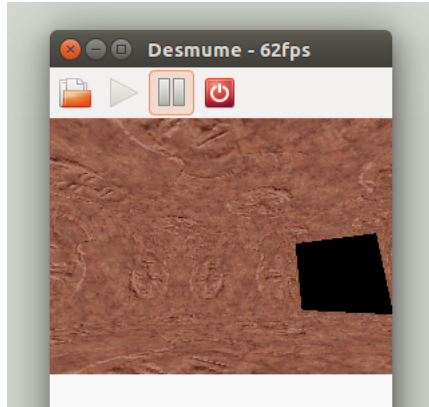


Figura 6: touch\_look

### 3.5 Touch\_Pad/touch\_test/

Parecido al *test\_area* pero proporcionando más información como que botones están siendo pulsados.

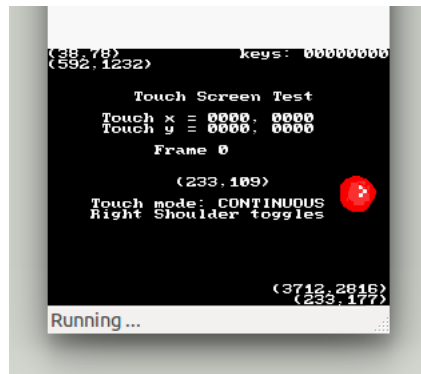


Figura 7: touch\_test

## 4 Trabajo autónomo

Mediante los ejemplos indicados no hubo gran problemas para realizar el ejercicio, lo único que ha requerido refrescar 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 tras darle a la función de `lcdSwap`.

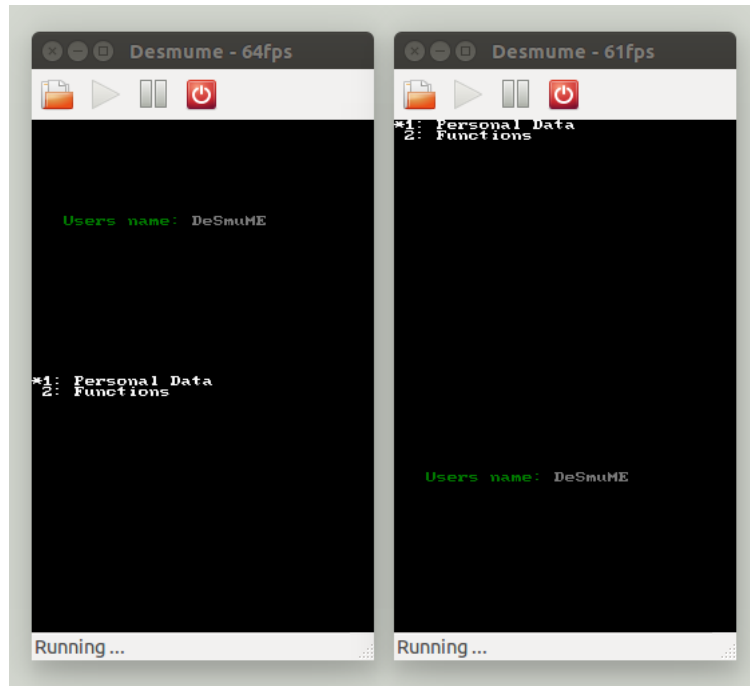


Figura 8: Programa con menu