Iñigo Olaso y Edurne Sagasta

Buses y Comunicaciones en Sistemas Embebidos

17/04/2019

**PROYECTO BUSES**

* **Breve descripción del proyecto**

Este proyecto realizado en la asignatura de buses tiene como objetivo aplicar los conocimientos adquiridos durante el curso sobre los diferentes buses que se pueden emplear en sistemas embebidos. En Tabla I se recogen qué tipo de buses se han usado, qué componentes se comunican y la aplicación para la que se usa.

Tabla I: **Tipo de bus, qué dispositivos se comunican y la aplicación en la que se ha usado cada tipo de bus**.

|  |  |  |
| --- | --- | --- |
| **Tipo de bus** | **Comunicación entre los dispositivos** | **Aplicación** |
| I2C | Cyclon V- acelerómetro ADXL345 | Acelerómetro |
| Terasic custom display component in Qsys | Cyclon V -ILI 9341 | LCD |
| SPI | Cyclon V - Potenciometro | Potenciometro |
| Inamlámbrico: Radiofrecuencia | Base station-nodo sensor | Sensor de temperatura, humedad, iluminación |
| Serie | Cyclon V-Base station | Sensor de temperatura, humedad, iluminación |

* + Las especificaciones iniciales del proyecto:

En el proyecto que propusimos realizar al inicio integraba:

* + - * Controlador LCD
      * TinyOS
      * Gsensor

La idea del proyecto se basa en mostrar los valores de las variables medidas por la mota y aceleración por el Gsensor en la pantalla LCD.

* + Las especificaciones finales del proyecto: Cambios introducidos durante el desarrollo del proyecto.

Durante el desarrollo del proyecto, decidimos añadir el convertidor analógico digital y la función TAP del Gsensor.

* **Información sobre de cómo compilar y ejecutar el proyecto**

Dentro de la carpeta **Buses\_entrega** se encuentrar las caspetas:

* *Acelerometro*
* *Joystick*
* *LCD\_Tiny*

*Acelerometro* y *Joystick* son módulos Kernel. Dentro de *Acelerometro*, para compilar hay realizar *make*. Para ejecutar el correspondiente módulo, hay que llamar a:

insmod acelerometro.ko.

Dentro de Joystick, para compilar hay que llamar a *make*. Para ejecutar el correspondiente módulo, hay que llamar a:

insmod joystick.ko.

En cuanto al LCD\_Tiny hay que realizar make para compilar y para ejecutar llamar a:

./lcd0 seriallisten /dev/ttyACM0 57600

* **Ejemplos y resultados de las pruebas realizadas para validar el correcto funcionamiento**

Cuando el sistema funciona correctamente el LCD muestra una imagen similar a la que aparece en Figura 1. La pantalla se ha dividido en 4 secciones (A, B, C y D).

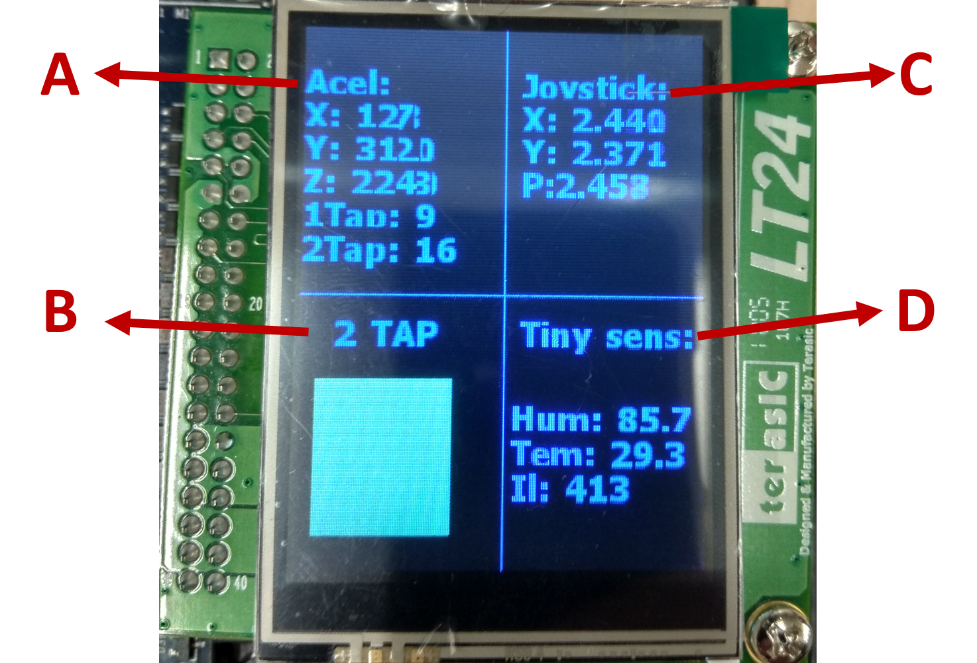


Figura 1: **Información que muestra la pantalla LCD en correcto funcionamiento.** Se muestran las secciones A, B, C y D que se mencionan en el texto.

Sección A muestra la información recogida del acelerómetro. Da los valores leídos en el eje X, Y y Z y muestra el contador de single tap y double tap. Cada vez que se realiza un tap (golpear una sola vez con los dedos el metacrilato), el contador *1Tap* suma 1. Cada vez que se realiza un double tap (golpear dos veces seguidas con los dedos el metacrilato), el contador *2Tap* suma 1. Nótese que al realizar double tap no se suma 2 en la variable 1Tap, sino solamente en 2Tap se hace +1.

Sección B muestra un cuadrado que cambia de color cada vez que detecta un double tap. Por lo tanto, una forma sencilla de confirmar el correcto funcionamiento del sistema, es pulsar dos veces el metacrilato y ver que el cuadrado cambia de color.

Sección C muestra los valores leídos del joystick, en el eje X, Y y el pulsador. Otra forma fácil de ver que el sistema funciona correctamente es pulsar el pulsador y observar que el valor correspondiente va a 0.

Sección D recoge los datos obtenidos del sensor Tiny: humedad (en %), temperatura (ºC) e iluminación en lúmenes. Sólo mostramos las trazas que corresponden a nuestro nodo sensor, cuyo ID es 57.

* **Problemas aparecidos y cómo se han resuelto**
* Problemas con la fecha y hora a la hora de compilar.
  + Al guardar el archivo .c en Notepad, lo guarda en la hora del PC y el compilador no considera que se ha editado el archivo.

Solución hacer make clear.

* Ejecución incorrecta después de la compilación.
  + Varias veces fue por la fecha incorrecta.
* Al inicio tuvimos problemas con el mapeo ya que no elegimos correctamente la base y el SPAN.
* Color de los pixeles en el LCD. La primera vez que mostramos los valores de la humedad, temperatura e iluminación en el LCD, los colores que se mostraban en la pantalla no coincidían con los elegidos. Solución: emplear la variable new\_color.
* Cuando combinamos por primera vez todos los elementos, nos dimos cuenta que el proceso de la extracción de los datos del sensor es muy lenta comparando con la extracción de los datos del acelerómetro y joystick. Solucionamos el problema añadiendo un contador y llamando a la función de Sensor\_ADC(&Display) menos a menudo. De esta forma controlamos el tiempo que transcurre el sistema en cada tarea. La forma de hacerlo mas correctamente sería utilizar forks o threads, pero por el tiempo limitado y la sencillez de la solución que proponemos hemos dejado así el proyecto.

**ANEXO I: Funcionalidad TAP del Gsensor**

El TAP sensing detecta un único tap (first tap en Fig. 2) o tap doble (first tap y second tap en Fig. 2).

Para poder configurar la funcionalidad del tap escribimos los siguientes registros:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Concepto** | **Registro** | **Valor en unidades** | **Unidades-hexad.** | **Valor en hexadec.** |
| Tap threshold | 0x1D | 3 g | 62.5mg/LSB | 0x30 |
| Maximum tap duration | 0x21 | 0.02 sec | 625μs/LSB | 0x20 |
| Tap latency | 0x22 | 0.02 sec | 1.25ms/LSB | 0x10 |
| Double-tap window | 0x23 | 0.3 sec | 1.25ms/LSB | 0xF0 |



Figura 2: Función de único tap y tap doble.

Tap latency es el periodo de espera desde el final del primer tap hasta el comienzo del time window mostrado en la Fig. 2. Durante el time window el segundo tap puede ser detectado.

Por encuesta verificamos si el ESTADO XXXXXX ha cambiado….

