Documentación Técnica

Aldo Reyes – 21003676

Funciones en el archivo os.c:

UART:

- **uart_putc:** Envia un solo carácter, se queda esperando hasta que el hardware este listo para transmitir.
- **uart_getc:** Recibe un solo carácter, se queda escuchando hasta que se presiona una tecla en la terminal.
- **os_write:** Es como un print básico, recibe un string y usa uart_putc para enviar la palabra completa.
- **os_read:** Lee una frase escrita por el usuario, lo que se escribe se guarda en un buffer hasta que se presiona enter.
- **Uart_putnum:** Es una función que convierte números en texto para que pueda verse en la terminal al momento de imprimir los resultados.

Timer:

- **timer_init:** enciende el reloj del timer, le dice al controlador de interrupciones que deje pasar la señal del Timer2.
- **timer_irq_handler:** Se ejecuta automáticamente cada 2 segundos, sin importar que este haciendo el programa principal, también es importante mencionar que también limpia las banderas que dan el aviso de que el evento fue atendido y que se puede recibir “Tick”

unsigned int rand(void): es un algoritmo matematico simple que genera números “aleatorios” que se despliegan en pantalla.

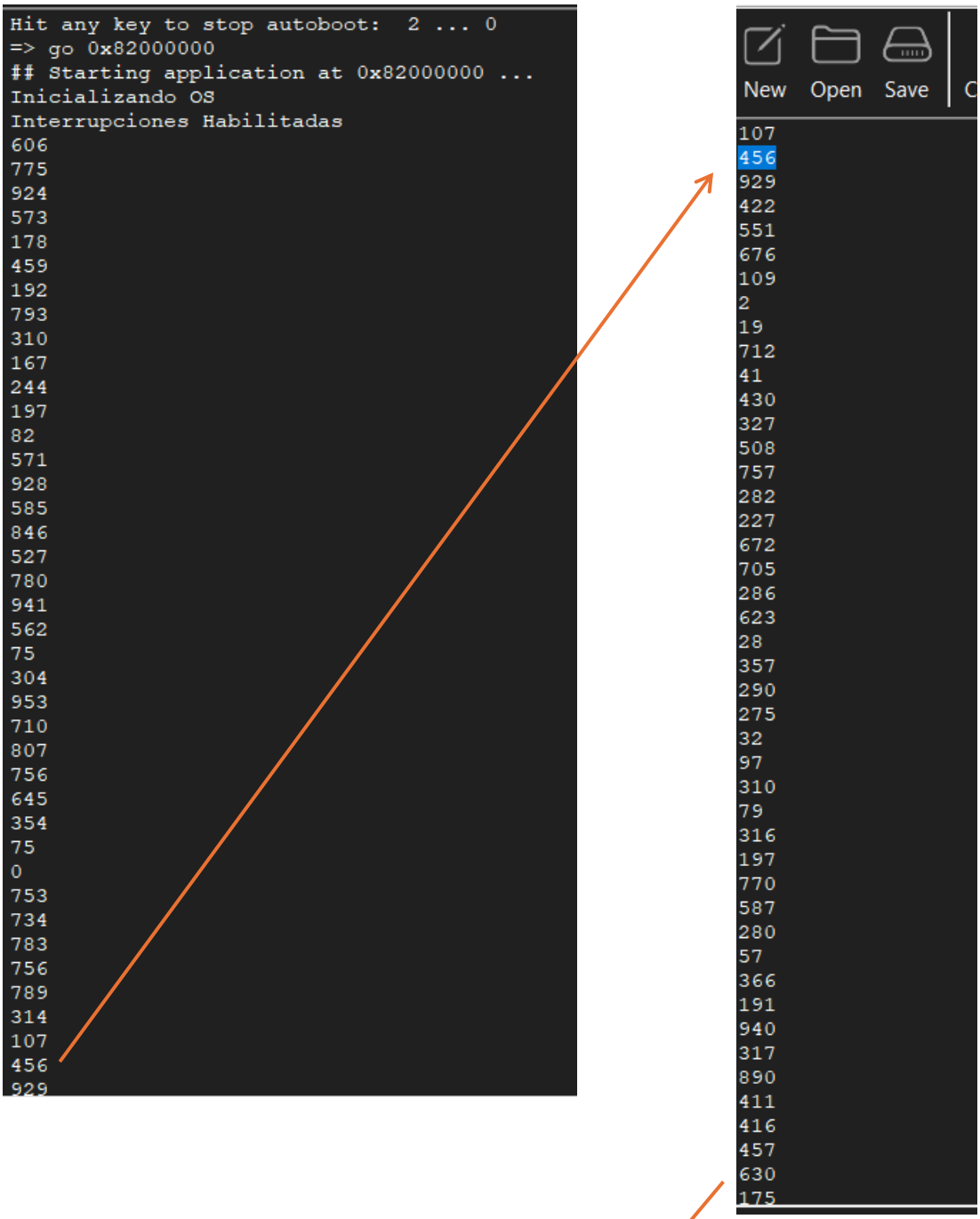
main: manda un mensaje de bienvenida, prepara el timer, activa las interrupciones globales y entra en un bluce infinito.

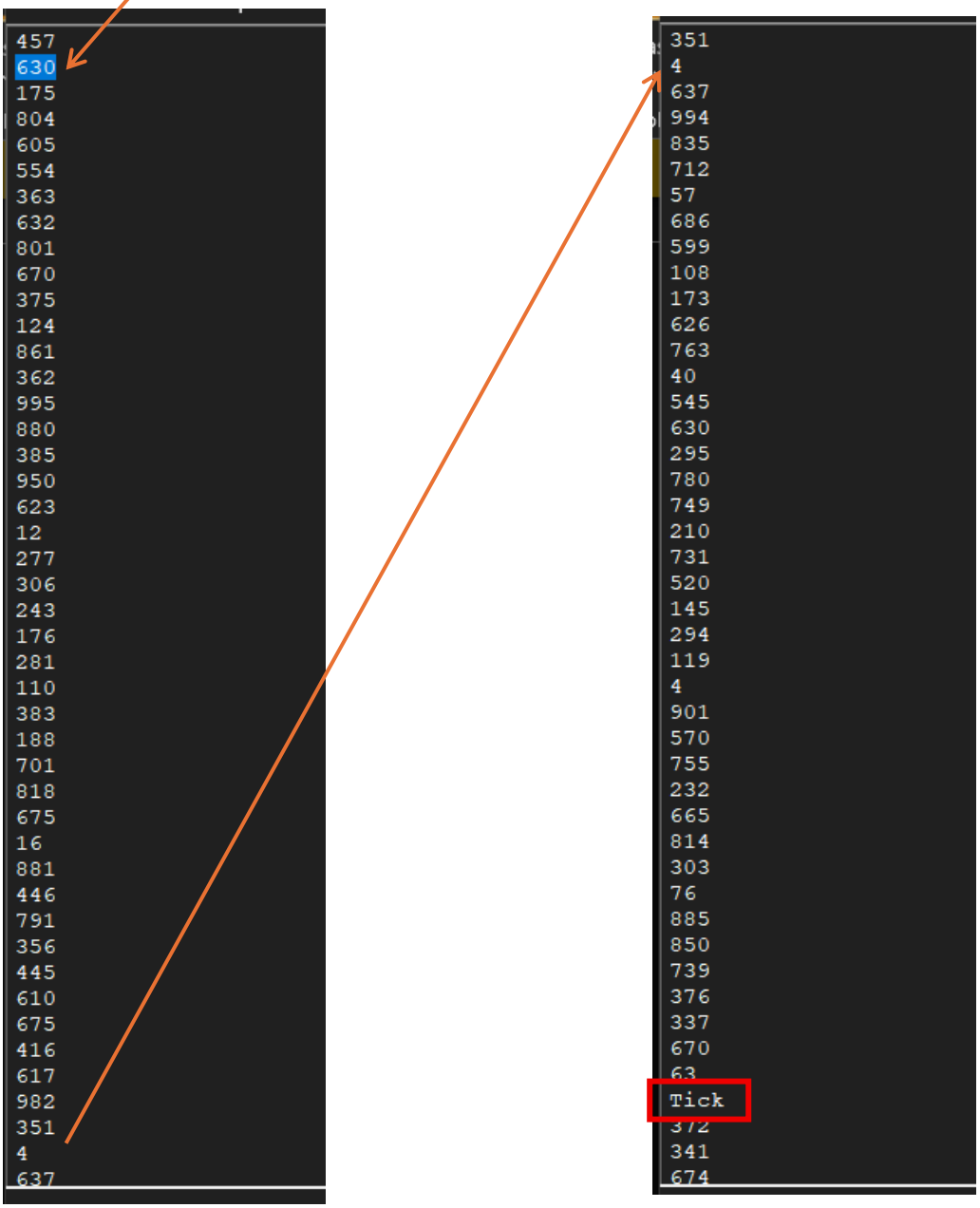
Funciones en el archivo root.s

- **reset_handler:** Configura el stack para los modos IRQ y System y registra la ubicación de la vector_table en el procesador.
- **irq_handler:** Gestiona el cambio de contexto, guarda los registros, llama a la lógica en C y restaura los registros.
- **PUT32(addr, val):** Escribe un valor de 32bits en una dirección de memoria especifica.
- **GET32(addr):** Lee el contenido de una dirección de memoria.
- **enable_irq:** Modifica el registro de estado del procesador (CPSR), limpia el bit 7 para permitir que las señales de interrupción lleguen al núcleo.

Arquitectura:

- **Nivel de periféricos (DMTIMER2):** El contador TCRR se incrementa con el reloj del sistema, al llegar al máximo, ocurre un overflow.
- **Nivel de controlador (INTC):** la señal del Timer llega al controlador de interrupciones de la BeagleBone, como el bit en INC_MIR_CLEAR2 esta habilitado, el INTC determina que esta interrupción debe notificarse al procesador ARM
- **Nivel de Procesador:** si CPSR esta en 0, el procesador detiene la ejecución del main, cambia al modo IRQ y salta al Vector de Interrupciones.
- **Nivel de Software:** El código de bajo nivel identifica que la fuente es el temporizador y llama a la funcion C.
- **Hardware:** El timer llega a cero y en la BeagleBone se envia una señal eléctrica al controlador de interrupciones (INTC).
- **Procesador:** Salta a la dirección de la Vector Table
- **Context Switch:** Se ejecuta el código en irq_handler que hace un Push a los registros.


Hit any key to stop autoboot: 2 ... 0
=> go 0x82000000
Starting application at 0x82000000 ...
Inicializando OS
Interrupciones Habilitadas
606
775
924
573
178
459
192
793
310
167
244
197
82
571
928
585
846
527
780
941
562
75
304
953
710
807
756
645
354
75
0
753
734
783
756
789
314
107
456
929
New
Open
Save
107
456
929
422
551
676
109
2
19
712
41
430
327
508
757
282
227
672
705
286
623
28
357
290
275
32
97
310
79
316
197
770
587
280
57
366
191
940
317
890
411
416
457
630
175


457
630
175
804
605
554
363
632
801
670
375
124
861
362
995
880
385
950
623
12
277
306
243
176
281
110
383
188
701
818
675
16
881
446
791
356
445
610
675
416
617
982
351
4
637
351
4
637
994
835
712
57
686
599
108
173
626
763
40
545
630
295
780
749
210
731
520
145
294
119
4
901
570
755
232
665
814
303
76
885
850
739
376
337
670
63
Tick
372
341
674