

Intrucciones para manejo del timer

Autor: MSc. Christian Medina

Se proporciona la biblioteca **timeLib.c** en **lenguaje C** con las subrutinas **getCycles** y **getCPS** y el programa **timer.s**. A continuación, se explica el funcionamiento de dichos programas:

```
#include <time.h>

int getCycles(void){
    clock_t cycles;
    cycles = clock();
    return((int) cycles);
}

int getCPS(void){
    return((int)CLOCKS_PER_SEC);
}
```

GetCycles devuelve en **r0** el valor del contador de ciclos del procesador. De modo que la diferencia entre dos llamadas sucesivas a esta subrutina nos permite conocer cuántos ciclos han transcurrido entre cada llamada.

```
b1 getCycles
ldr r1,=start
str r0,[r1]
//codigo a ejecutar
b1 delay
b1 getCycles
ldr r1,=end
str r0,[r1]
```

La diferencia entre los valores de las variables (**end-start**) nos da el número de ciclos transcurridos entre las llamadas sucesivas. En este ejemplo estarían midiendo el número de ciclos que se necesitan para ejecutar la subrutina **delay**.

Para convertir esta medida a tiempo se utiliza la constante **CLOCKS_PER_SEC**. Esta constante nos indica cuantos ciclos de procesador se ejecutan por segundo y se obtiene por la subrutina **getCPS**.

De modo que **(end-start)/CLOCKS_PER_SEC** nos permite convertir los ciclos de ejecución a segundos y **CLOCKS_PER_SEC/(end-start)** nos permite convertir los ciclos de ejecución a Hertz.