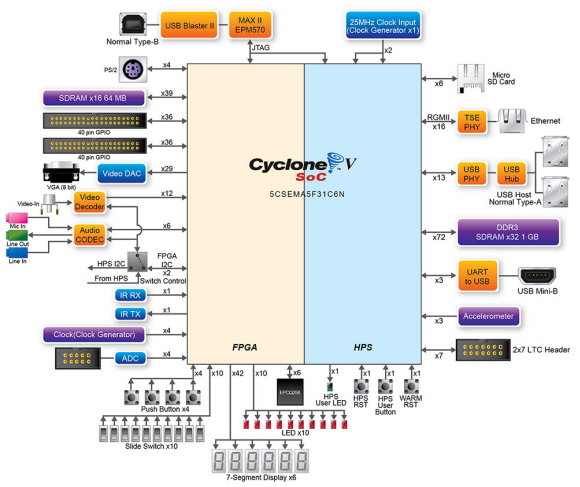
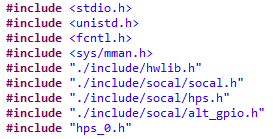
**MANEJO DE PERIFERICOS EN LINUX (HPS)**

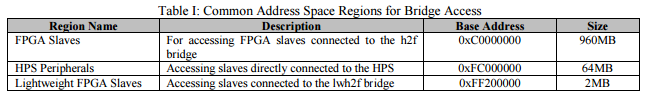
Escribir un código en el software Eclipse DS-5 para que el procesador pueda acceder a periféricos conectados a la FPGA, a través de dispositivos virtuales en memoria (RAM).



Primero se deben incluir las siguientes librerías que hacen parte Altera's Soc Embedded Design Suite (SoC EDS).

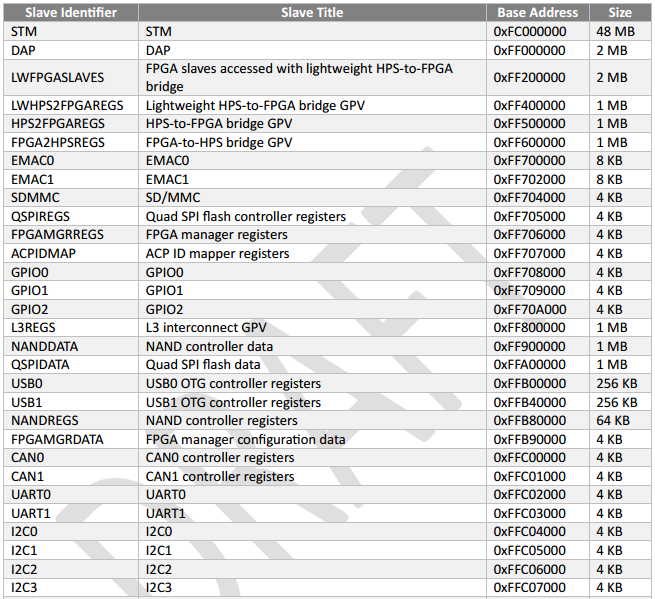


La memoria RAM del procesador ARM de la FPGA es de 1gb que se encuentra distribuido de la siguiente forma:

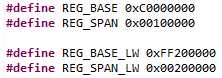


Hay dos posibilidades de conectar el procesador ARM con los periféricos conectados directamente a la FPGA, a través del puente h2f, el cual posee 960MB de espacio disponible y una conectividad que soporta 64bits. Por otra parte, para adquisición o envío de datos de menor densidad, se puede usar una conectividad lwh2f de bajo peso que posee 2MB, la cual soporta 32bits y es ideal para periféricos con bajo flujo de datos.

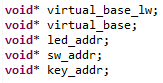
La localidad de memoria de 64MB, esta designada para los periféricos conectados al HPS. Para acceder a ellos se debe tener en cuenta la dirección en memoria de esta tabla, además de habilitarlos en las propiedades del HPS, dentro del QSYS.



Se puede acceder a periféricos conectados a la FPGA a través de los puentes h2f y lwh2f de manera simultánea. Solo se deben tener en cuenta las direcciones de memoria donde inicia cada partición y la cantidad de memoria a la que se requiere acceder. Se crea un registro base que hace referencia a la dirección de inicio de memoria (REG\_BASE) y un registro que indica la cantidad de memoria solicitada (REG\_SPAN), para cada localidad deseada.



Para cada periférico se crea un apuntador y además se debe crear variable de una memoria virtual desde la cual se obtendrán los registros de los demás apuntadores.



Se crean las variables enteras fd, switches y keys. La variable fd almacena el valor que retornará la función open. En lal variables switches y keys se guardaran la posición de los switches y de los botones, respectivamente, de la FPGA.



Se utiliza la función open para acceder al dispositivo de memoria, otorgándole permisos de escritura y lectura, forzando la sincronización de acceso a memoria con los tiempos de trabajo del procesador ARM.



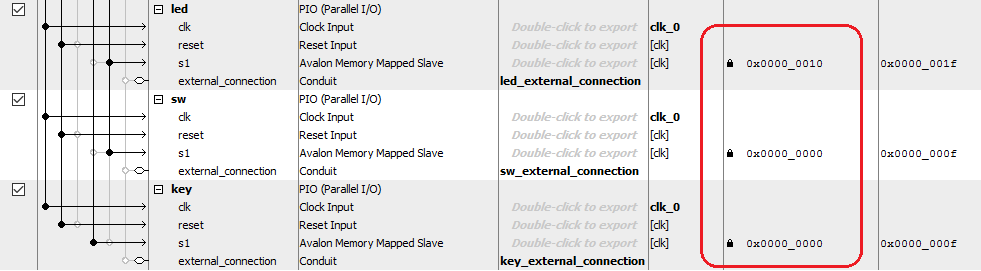
La función mmap permite mapear archivos o dispositivos en memoria. Para ello se indica que el mapeo ocurrirá desde una dirección de memoria específica, y para un ancho de memoria determinado. Además será de tipo compartido y con permiso de lectura y escritura. Una vez mapeado, se obtiene una dirección de memoria utilizable, sobre la cual podemos trabajar.



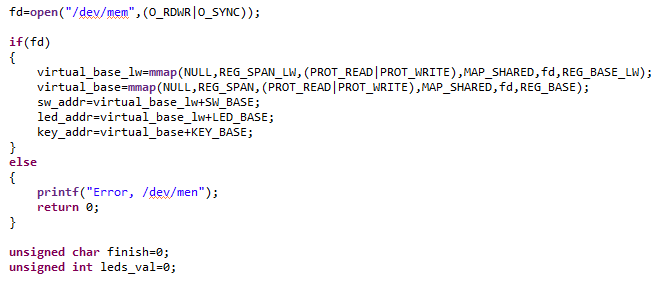
Con el espacio de memoria utilizable, se le dice al programa donde se encuentra cada uno de los componentes a través de los registros base de cada periférico. Como se utiliza el puerto para transmisión de datos de los leds, los switches y los botones, se asignó un espacio en memoria para cada uno de ellos.



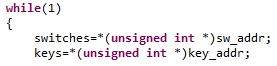
Estos valores se definen en el Qsys, a la hora de diseñar el hardware, donde se asignan direcciones de memoria a cada periférico, dependiendo del número de bits que posea cada uno de ellos.



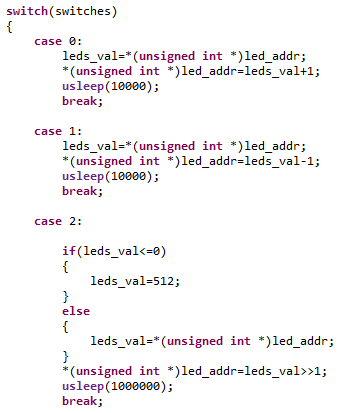
La función open, puede retornar en algunos casos un valor de -1, esto ocurre cuando la localidad de memoria a la que se intentó acceder está restringida, o no se encuentra. Por ello se muestra en pantalla si hay problemas al intentar acceder a determinada localidad.

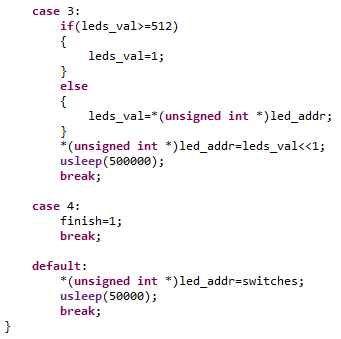


Para obtener la información almacenada en el dispositivo de memoria, se utilizan los apuntadores dentro de un ciclo infinito, y así se extrae la información almacenada en esa localidad de memoria.



Se diseñó un ejemplo a manera de switch case, que permitiera observar en los LEDS de la FPGA, diversos comportamientos, dependiendo de la selección de los switches.





Se debe liberar memoria una vez completada la función realizada. Para ello se utiliza la función munmap y close.

