**Configurando el DMA para que transfiera en un solo paso**

|  |
| --- |
| void DMA\_init(void) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| /\*\* PARAMETERS TO MODIFY\*/ |
|  |

|  |
| --- |
| uint8\_t word\_width\_in\_bytes = 2; |
|  |

|  |
| --- |
| uint32\_t words\_to\_trasmmit = 16 \* word\_width\_in\_bytes; |
|  |

|  |
| --- |
| uint32\_t citer\_steps = 1; |
|  |

|  |
| --- |
| uint32\_t source\_offset = 1; /\*\* write this on words, a word not from k64 archq , a word from DMA transfer\*/ |
|  |

|  |
| --- |
| uint32\_t destination\_offset = 1 /\*\* write this on words, , a word not from k64 archq , a word from DMA transfer \*/; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| DMAMUX->CHCFG[0] = 0; |
|  |

|  |
| --- |
| DMAMUX->CHCFG[0] = DMAMUX\_CHCFG\_ENBL\_MASK | /\*enables DMA MUX channel\*/ |
|  |

|  |
| --- |
| DMAMUX\_CHCFG\_SOURCE(DMA\_SOURCE\_GPIO);/\*source is GPIO PIN corresponding to sw2\*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| DMA0->ERQ = 0x01;//enables DMA0 request |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| DMA0->TCD[0].SADDR = (uint32\_t)(&g\_data\_source[0]);/\*defines source data address\*/ |
|  |

|  |
| --- |
| DMA0->TCD[0].SOFF = source\_offset \*word\_width\_in\_bytes;/\*Source address signed offset;it is expressed in number of bytes\*/ |
|  |

|  |
| --- |
| DMA0->TCD[0].DADDR = (uint32\_t)(&g\_data\_desti[0]);/\*defines destination data address\*/ |
|  |

|  |
| --- |
| DMA0->TCD[0].DOFF = destination\_offset;/\*destination address signed offset;it is expressed in number of bytes\*/ |
|  |

|  |
| --- |
| /\*CITER represents the number of minor loops that compose a major loop, every time a minor loop is completed CITTER is decremented by one. |
|  |

|  |
| --- |
| \* Once CITTER is 0 the DMA copy BITTER into CITTER and adjust SADDR and DADDR with the values specified in SLAST and DLAST\_SGA respectively |
|  |

|  |
| --- |
| \* SADDR = SADDR + SLAST and DADDR + DLAST\_SGA\*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| DMA0->TCD[0].CITER\_ELINKNO = DMA\_CITER\_ELINKNO\_CITER(citer\_steps); |
|  |

|  |
| --- |
| /\* Once a major loop is completed, BITTER is copy to CITTER\*/ |
|  |

|  |
| --- |
| DMA0->TCD[0].BITER\_ELINKNO = DMA\_BITER\_ELINKNO\_BITER( citer\_steps ); |
|  |

|  |
| --- |
| DMA0->TCD[0].NBYTES\_MLNO = (words\_to\_trasmmit\*word\_width\_in\_bytes ) / citer\_steps ;/\*byte number for transference\*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| DMA0->TCD[0].ATTR = DMA\_ATTR\_SSIZE(word\_width\_in\_bytes-1);/\*16 bit transfer size, in order to transfer see Kinetis user manual\*/ |
|  |

|  |
| --- |
| DMA0->TCD[0].SLAST = -words\_to\_trasmmit\*word\_width\_in\_bytes;//restores the source address to the initial value, which is expressed in the amount of bytes to restore\*/ |
|  |

|  |
| --- |
| DMA0->TCD[0].DLAST\_SGA = - (words\_to\_trasmmit \* word\_width\_in\_bytes \* destination\_offset );/\*restores the destination address to the initial value, which is expressed in the amount of bytes to restore\*/ |
|  |

|  |
| --- |
| DMA0->TCD[0].CSR = DMA\_CSR\_INTMAJOR\_MASK;/\*The end-of-major loop interrupt is enabled\*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

}

**Configurando el DMA para que transfiera en dos pasos , (un paso por cada vez que se presione el SW2)**

|  |
| --- |
| void DMA\_init(void) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| uint8\_t word\_width\_in\_bytes = 2; |
|  |

|  |
| --- |
| uint32\_t words\_to\_trasmmit =16 \* word\_width\_in\_bytes; |
|  |

|  |
| --- |
| DMAMUX->CHCFG[0] = 0; |
|  |

|  |
| --- |
| DMAMUX->CHCFG[0] = DMAMUX\_CHCFG\_ENBL\_MASK | /\*enables DMA MUX channel\*/ |
|  |

|  |
| --- |
| DMAMUX\_CHCFG\_SOURCE(DMA\_SOURCE\_GPIO);/\*source is FTM0 channel 0\*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| DMA0->ERQ = 0x01;//enables DMA0 request |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| DMA0->TCD[0].SADDR = (uint32\_t)(&g\_data\_source[0]);/\*defines source data address\*/ |
|  |

|  |
| --- |
| DMA0->TCD[0].SOFF = 1\*word\_width\_in\_bytes;/\*Source address signed offset;it is expressed in number of bytes\*/ |
|  |

|  |
| --- |
| DMA0->TCD[0].DADDR = (uint32\_t)(&g\_data\_desti[0]);/\*defines destination data address\*/ |
|  |

|  |
| --- |
| DMA0->TCD[0].DOFF = 1;/\*destination address signed offset;it is expressed in number of bytes\*/ |
|  |

|  |
| --- |
| /\*CITER represents the number of minor loops that compose a major loop, every time a minor loop is completed CITTER is decremented by one. |
|  |

|  |
| --- |
| \* Once CITTER is 0 the DMA copy BITTER into CITTER and adjust SADDR and DADDR with the values specified in SLAST and DLAST\_SGA respectively |
|  |

|  |
| --- |
| \* SADDR = SADDR + SLAST and DADDR + DLAST\_SGA\*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| uint32\_t source\_address = DMA0->TCD[0].SADDR; |
|  |

|  |
| --- |
| uint32\_t desti\_address = DMA0->TCD[0].DADDR; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| DMA0->TCD[0].CITER\_ELINKNO = DMA\_CITER\_ELINKNO\_CITER(NUM\_STEPS); |
|  |

|  |
| --- |
| /\* Once a major loop is completed, BITTER is copy to CITTER\*/ |
|  |

|  |
| --- |
| DMA0->TCD[0].BITER\_ELINKNO = DMA\_BITER\_ELINKNO\_BITER(NUM\_STEPS); |
|  |

|  |
| --- |
| DMA0->TCD[0].NBYTES\_MLNO = (words\_to\_trasmmit\*word\_width\_in\_bytes )/NUM\_STEPS;/\*byte number for transference\*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| DMA0->TCD[0].ATTR = DMA\_ATTR\_SSIZE(word\_width\_in\_bytes-1);/\*16 bit transfer size, in order to transfer see Kinetis user manual\*/ |
|  |

|  |
| --- |
| DMA0->TCD[0].SLAST = -words\_to\_trasmmit\*word\_width\_in\_bytes;//restores the source address to the initial value, which is expressed in the amount of bytes to restore\*/ |
|  |

|  |
| --- |
| DMA0->TCD[0].DLAST\_SGA = -words\_to\_trasmmit\*word\_width\_in\_bytes;/\*restores the destination address to the initial value, which is expressed in the amount of bytes to restore\*/ |
|  |

|  |
| --- |
| DMA0->TCD[0].CSR = DMA\_CSR\_INTMAJOR\_MASK;/\*The end-of-major loop interrupt is enabled\*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

}

**Configurando el DMA para que genere la siguiente secuencia: 5,7,9,1,3,5**

|  |
| --- |
| void DMA\_init(void) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| /\*\* PARAMETERS TO MODIFY\*/ |
|  |

|  |
| --- |
| uint8\_t word\_width\_in\_bytes = 2; |
|  |

|  |
| --- |
| uint32\_t words\_to\_trasmmit = 3 \* word\_width\_in\_bytes; |
|  |

|  |
| --- |
| uint32\_t citer\_steps = 1; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| uint32\_t source\_offset = 2; /\*\* write this on words, a word not from k64 archq , a word from DMA transfer\*/ |
|  |

|  |
| --- |
| uint32\_t destination\_offset = 1 /\*\* write this on words, , a word not from k64 archq , a word from DMA transfer \*/; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| DMAMUX->CHCFG[0] = 0; |
|  |

|  |
| --- |
| DMAMUX->CHCFG[0] = DMAMUX\_CHCFG\_ENBL\_MASK | /\*enables DMA MUX channel\*/ |
|  |

|  |
| --- |
| DMAMUX\_CHCFG\_SOURCE(DMA\_SOURCE\_GPIO);/\*source is GPIO PIN corresponding to sw2\*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| DMA0->ERQ = 0x01;//enables DMA0 request |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| DMA0->TCD[0].SADDR = (uint32\_t)(&g\_data\_source[4]);/\*defines source data address\*/ |
|  |

|  |
| --- |
| DMA0->TCD[0].SOFF = source\_offset \*word\_width\_in\_bytes;/\*Source address signed offset;it is expressed in number of bytes\*/ |
|  |

|  |
| --- |
| DMA0->TCD[0].DADDR = (uint32\_t)(&g\_data\_desti[0]);/\*defines destination data address\*/ |
|  |

|  |
| --- |
| DMA0->TCD[0].DOFF = destination\_offset;/\*destination address signed offset;it is expressed in number of bytes\*/ |
|  |

|  |
| --- |
| /\*CITER represents the number of minor loops that compose a major loop, every time a minor loop is completed CITTER is decremented by one. |
|  |

|  |
| --- |
| \* Once CITTER is 0 the DMA copy BITTER into CITTER and adjust SADDR and DADDR with the values specified in SLAST and DLAST\_SGA respectively |
|  |

|  |
| --- |
| \* SADDR = SADDR + SLAST and DADDR + DLAST\_SGA\*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| DMA0->TCD[0].CITER\_ELINKNO = DMA\_CITER\_ELINKNO\_CITER(citer\_steps); |
|  |

|  |
| --- |
| /\* Once a major loop is completed, BITTER is copy to CITTER\*/ |
|  |

|  |
| --- |
| DMA0->TCD[0].BITER\_ELINKNO = DMA\_BITER\_ELINKNO\_BITER( citer\_steps ); |
|  |

|  |
| --- |
| DMA0->TCD[0].NBYTES\_MLNO = (words\_to\_trasmmit\*word\_width\_in\_bytes ) / 2 ;/\*byte number for transference\*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| DMA0->TCD[0].ATTR = DMA\_ATTR\_SSIZE(word\_width\_in\_bytes-1);/\*16 bit transfer size, in order to transfer see Kinetis user manual\*/ |
|  |

|  |
| --- |
| DMA0->TCD[0].SLAST = -10\*word\_width\_in\_bytes;//restores the source address to the initial value, which is expressed in the amount of bytes to restore\*/ |
|  |

|  |
| --- |
| DMA0->TCD[0].DLAST\_SGA = - 0;/\*restores the destination address to the initial value, which is expressed in the amount of bytes to restore\*/ |
|  |

|  |
| --- |
| DMA0->TCD[0].CSR = DMA\_CSR\_INTMAJOR\_MASK;/\*The end-of-major loop interrupt is enabled\*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| } |
|  |
|  |