```
//////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////////
/// LABMAIN.C --- INTERRUPT TEMPLATE
//////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////////

#include "shared.h"
#include <math.h> /* Note: We get a warning if math.h is moved up */


interrupt void tx_isr(void); /* prototype the ISR  */

void labmain()
{
  /* Select McBSP1 transmit int for INT15 */
   intr_map(CPU_INT15, ISN_XINT1);

  /* Hook our ISR to INT15                */
   intr_hook(tx_isr, CPU_INT15);

  /* Clear old interrupts                 */
   INTR_CLR_FLAG(CPU_INT15);

  /* Enable interrupts                      */
  /* NMI must be enabled for other ints to occur */
   INTR_ENABLE(CPU_INT_NMI);

  /* Set INT15 bit in IER                 */
   INTR_ENABLE(CPU_INT15);

  /* Turn on enabled ints                 */
   INTR_GLOBAL_ENABLE();

  /* Note: Functions in capital letters are    */
  /* macros defined in intr.h                  */

  /*Write a sample to start transmission  */
   WriteSample(0,0);

   while(1); /* infinite loop */
}



void tx_isr() // interrupt handler
{
      /* You should know what goes here */
      /* (The codec is ready, so just generate and output the next sample */
}
```