```
//###########################################################################
// FILE:   Main_M3.c
//###########################################################################
//


/* ************************************** */
/*              Définition des includes            */
/* ************************************** */
#include "inc/hw_ipc.h"
#include <string.h>
#include "inc/hw_ints.h"
#include "inc/hw_memmap.h"
#include "inc/hw_gpio.h"
#include "inc/hw_types.h"
#include "inc/hw_sysctl.h"
#include "driverlib/debug.h"
#include "driverlib/flash.h"
#include "driverlib/ipc.h"
#include "driverlib/interrupt.h"
#include "driverlib/sysctl.h"
#include "driverlib/gpio.h"
#include "driverlib/timer.h"

/* ************************************** */
/*             Definition des prototypes              */
/* ************************************** */
```

```
/* ************************************** */
/* Définition des variables et des constantes */
/* ************************************** */
extern unsigned long RamfuncsLoadStart;
extern unsigned long RamfuncsRunStart;
extern unsigned long RamfuncsLoadSize;

unsigned int coefficient_M3[256];
unsigned int i=0;

#pragma DATA_SECTION(Coeff, "CtoM")
unsigned int Coeff;

/* ************************************** */
/*                                MAIN                                */
/* ************************************** */
void main(void)
{
	HWREG(SYSCTL_MWRALLOW)=0xA5A5A5A5; //Disable protection

	SysCtlClockConfigSet(SYSCTL_USE_PLL | (SYSCTL_SPLLIMULT_M & 0xA) |
SYSCTL_SYSDIV_1 | SYSCTL_M3SSDIV_1 | SYSCTL_XCLKDIV_4); //Sets up PLL, M3 running at 100MHz
and C28 running at 100MHz

	/* Copy time critical code and Flash setup code to RAM
	 * This includes the following functions : InitFlash();
	 * The RamFuncsLoadStart, RamfuncsLoadSize and
	 * RamfuncsRunStart symbols are created by the linker.
	 * Refer to the device .cmd file.*/
```

```
        memcpy(&RamfuncsRunStart, &RamfuncsLoadStart, (size_t)&RamfuncsLoadSize);


        FlashInit(); //Call flash initialization to setup flash waitstates (this function
must reside in RAM)


        IPCMtoCBootControlSystem(CBROM_MTOC_BOOTMODE_BOOT_FROM_FLASH); //Send boot
command to allow the C28 application to begin execution

        IPCMtoCBootControlSystem(CBROM_MTOC_BOOTMODE_BOOT_FROM_RAM);


        while(1)

        {

                while(IPCCtoMFlagBusy(IPC_CTOMIPCSTS_IPC1)==0); //Wait for IPC1

                coefficient_M3[i]=Coeff; //Read and store result

                IPCCtoMFlagAcknowledge(IPC_CTOMIPCACK_IPC1); //Clear IPC1-flag

                if(i++>=100)i=0;

        }

}


#endif
```