```
//###########################################################################
// FILE:   Main_C28.c
//###########################################################################
//



/* ************************************** */
/*              Définition des includes            */
/* ************************************** */
#include "DSP28X_Project.h"
#include <string.h>
#include <time.h>

/* ************************************** */
/*             Definition des prototypes              */
/* ************************************** */
void spi_init(void);
void spi_fifo_init(void);
void spi_comm(void);
interrupt void spirx_isr(void);
void delay(unsigned int timing);
interrupt void spitx_isr(void);
interrupt void coeff_isr(void);
interrupt void cpu_timer1_isr(void);
interrupt void cpu_timer0_isr(void);

/* ************************************** */
/* Définition des variables et des constantes */
```

```
/* ************************************** */
#pragma DATA_SECTION(LatestCoeff, "CtoM")

#define C28_FREQ 100E6 // ou 100 tout court ?
#define SPI_BAUDRATE 1000000 // unité ?
#define LSPCLK_FREQ (C28_FREQ/4)
#define SPI_BRR ((LSPCLK_FREQ/SPI_BAUDRATE)-1) // SPI baud rate

extern Uint16 RamfuncsLoadStart;
extern Uint16 RamfuncsLoadSize;
extern Uint16 RamfuncsRunStart;

Uint16 LatestCoeff;
Uint16 coefficient_C28[256]={0};
Uint16 data_acq[8];

unsigned int i=0;
unsigned int j=0;
unsigned int transm=0;
unsigned int timing;

void spi_init(void)
{
	SpiaRegs.SPICCR.bit.SPISWRESET=0; //release the SPI from reset
	SpiaRegs.SPICCR.bit.CLKPOLARITY=1; //all data transistions are during
	SpiaRegs.SPICTL.bit.CLK_PHASE=1;   //...the falling edge, non-delayed clock.
                                                    //...Inactive level is
low.
```

```
	SpiaRegs.SPICTL.bit.SPIINTENA=1; //enable SPI interrupts
}

void spi_fifo_init(void)
{
	SpiaRegs.SPIFFTX.bit.SPIRST=0; //Initialize SPI FIFO registers
	SpiaRegs.SPIFFTX.bit.TXFIFO=0; //Reset the FIFO pointer and hold in reset
	SpiaRegs.SPIFFTX.bit.TXFFIL=1; //Set TX FIFO to 1 word.
	SpiaRegs.SPIFFTX.bit.SPIFFENA=1; //enable TX FIFO
	SpiaRegs.SPIFFTX.bit.TXFFIENA=1; //enable TX FIFO interrupts
	SpiaRegs.SPIFFTX.bit.TXFIFO=1; //release FIFO from reset

	SpiaRegs.SPIFFRX.bit.RXFIFORESET=0; //Put RX FIFO in reset
	SpiaRegs.SPIFFRX.bit.RXFFIL=8; //Set RX FIFO to 8 words
	SpiaRegs.SPIFFRX.bit.RXFFIENA=1; //Enable RX FIFO interrupts
	SpiaRegs.SPIFFRX.bit.RXFIFORESET = 1; //Release RX FIFO from reset

	SpiaRegs.SPIFFCT.all=0x0; //transmit delay is zero

	SpiaRegs.SPIFFTX.bit.SPIRST=1; //Realease SPI from reset
}

void spi_comm(void)
{
	SpiaRegs.SPICTL.bit.MASTER_SLAVE=1; //Mode master
	SpiaRegs.SPICTL.bit.TALK=1; //communication is authorized
}
```

```
interrupt void spirx_isr(void)
{
	coefficient_C28[transm]=SpiaRegs.SPIRXBUF; //data received
	LatestCoeff=coefficient_C28[transm];

	CtoMIpcRegs.CTOMIPCSET.bit.IPC1=1; //signal IPC1 to M3

	if(transm<256)
		transm++;
	else
		transm=0;

	SpiaRegs.SPIFFRX.bit.RXFFINTCLR=1; //Clear all RX interrupt flags in SPI
	PieCtrlRegs.PIEACK.all|=M_INT6;
}

interrupt void spitx_isr(void)
{
	for(i=0;i<8;i++)
		SpiaRegs.SPITXBUF=data_acq[i];
	for(i=0;i<8;i++)
		data_acq[i]=data_acq[i]+1;

	SpiaRegs.SPIFFTX.bit.TXFFINTCLR=1;
	PieCtrlRegs.PIEACK.all|=0x20;
}
```

```
void main(void)
{
	/* ************************************** */
	/*                                INIT                            */

	/* ************************************** */
	InitSysCtrl(); // initialize by default

	DINT; //Disable CPU Interrupts

	InitGpio(); // initialize by default

	EALLOW; //disable protection start
	GpioCtrlRegs.GPAMUX2.bit.GPIO20=0; //bit start
	GpioCtrlRegs.GPAMUX2.bit.GPIO21=0; // ST actuator
	GpioCtrlRegs.GPAMUX2.bit.GPIO22=0; // CLK actuator

	GpioG1CtrlRegs.GPADIR.bit.GPIO16=1; //as output
	GpioG1CtrlRegs.GPADIR.bit.GPIO17=0; //as input
	GpioG1CtrlRegs.GPADIR.bit.GPIO18=1;
	GpioG1CtrlRegs.GPADIR.bit.GPIO19=1;
	GpioG1CtrlRegs.GPADIR.bit.GPIO20=1;
	GpioG1CtrlRegs.GPADIR.bit.GPIO21=1;
	GpioG1CtrlRegs.GPADIR.bit.GPIO22=1;

	GpioG1DataRegs.GPACLEAR.bit.GPIO16=1; //low
```

```
    GpioG1DataRegs.GPACLEAR.bit.GPIO17=1;
    GpioG1DataRegs.GPACLEAR.bit.GPIO18=1;
    GpioG1DataRegs.GPASET.bit.GPIO19=1; //high
    GpioG1DataRegs.GPACLEAR.bit.GPIO20=1;
    GpioG1DataRegs.GPASET.bit.GPIO21=1;
    GpioG1DataRegs.GPACLEAR.bit.GPIO22=1;

    EDIS;//disable protection end

    /* 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);
    InitFlash();

    InitPieCtrl(); //Init interrupt vector
    InitPieVectTable();//Init interrupt vector table

    EALLOW;
    PieVectTable.SPIRXINTA=&spirx_isr;
    PieVectTable.SPITXINTA=&spitx_isr;
    EDIS;

    PieCtrlRegs.PIECTRL.bit.ENPIE = 1; //Enable PIE block
    PieCtrlRegs.PIEIER1.bit.INTx1=1; //Enable SPIRXINTA in the PIE: Group 1 interrupt
1
```

```
	PieCtrlRegs.PIEIER1.bit.INTx2=1; //Enable SPITXINTA in the PIE: Group 1 interrupt
2


	data_acq[0]=1;

	data_acq[1]=0;

	data_acq[2]=0;

	data_acq[3]=1;

	data_acq[4]=0;

	data_acq[5]=1;

	data_acq[6]=0;

	data_acq[7]=0;


	InitSpi();

	InitSpiGpio();

	EALLOW;

	void spi_init(void);

	SysCtrlRegs.LOSPCP.all=0;

	SpiaRegs.SPIBRR=SPI_BRR; //16Mbps to have 1MHz

	void spi_fifo_init(void);

	void spi_comm(void);

	EDIS;


	/* ************************************** */

	/*                            GPIO FLASH                        */

	/* ************************************** */

	       GpioG1DataRegs.GPASET.bit.GPIO20=1;


	/* ************************************** */
```

```
/*                          ATTENTE X NS                        */
/* ************************************** */
DELAY_US(0.2); //200ns
/* ************************************** */
/*                   RISING EDGE ON CLK                 */
/* ************************************** */
       GpioG1DataRegs.GPASET.bit.GPIO22=1;

/* ************************************** */
/*                   GPIO FLASH END                     */
/* ************************************** */
       GpioG1DataRegs.GPACLEAR.bit.GPIO20=1;

/* ************************************** */
/*                   FALLING EDGE ON ST                 */
/* ************************************** */
GpioG1DataRegs.GPACLEAR.bit.GPIO21=1;

/* ************************************** */
/*               FALLING EDGE ON CLK                */
/* ************************************** */
GpioG1DataRegs.GPACLEAR.bit.GPIO22=1;

/* ************************************** */
/*                   DELAY ST                      */
/* ************************************** */
DELAY_US(0.4); //1us
```

```
	/* ****************************************** */
	/*                  RISING EDGE ON ST              */
	/* ****************************************** */
	GpioG1DataRegs.GPASET.bit.GPIO21=1;

	/* ****************************************** */
	/*                   BOUCLE FOR 256 TIMES               */
	/* ****************************************** */
	for(j=0;j<256;j++)
	{

		/* ****************************************** */
		/*                    WAIT Y NS                  */
		/* ****************************************** */
			DELAY_US(0.4); //wait optimal time to make spi conversion after A/D converter

		/* ****************************************** */
		/*                 FONCTION SPI CONVERS            */
		/* ****************************************** */
			interrupt void spitx_isr(void);
			if(SpiaRegs.SPIFFRX.bit.RXFFST) //SPI FIFO receive flag rised up
			{
				LatestCoeff=coefficient_C28[transm];

				CtoMIpcRegs.CTOMIPCSET.bit.IPC1=1; //signal IPC1 to M3

				if(transm<256)
```

```
                        transm++;
                    else
                        transm=0;

                    SpiaRegs.SPIFFRX.bit.RXFFINTCLR=1; //Clear all RX interrupt
flags in SPI
                    PieCtrlRegs.PIEACK.all|=M_INT6;
                }

            /* ************************************** */
            /*                 RISING EDGE ON CLK              */
            /* ************************************** */
            GpioG1DataRegs.GPASET.bit.GPIO22=1;
            /* ************************************** */
            /*                           WAIT Z NS              */
            /* ************************************** */
                DELAY_US(0.3); //300ns
            /* ************************************** */
            /*                 FALLING EDGE ON CLK             */
            /* ************************************** */
            GpioG1DataRegs.GPACLEAR.bit.GPIO22=1;
            /* ************************************** */
            /*                           WAIT Z NS              */
            /* ************************************** */
                DELAY_US(0.3); //300ns

        } //end of the boucle of 256times
}
```