

```

/* *****
* Filename: demo.c
* Author: Don Luebbe
* Last Modified: 08/24/02
*
* Description: This program illustrates basic initialization and
* operation of the LF2407 DSP. The following peripherals are
* exercised:
* 1) Controlling the duty cycle of a PWM with and analog input.
* 2) Controlling a series of digital outputs with either a timer or
* an encoder. The input will be selected with a digital input.
*
*
***** */

```

comment

```

/** Header files */
#include "f2407_c.h" /* Defines register names and addresses */
#include <math.h> /* Enables the DSP to do certain math
functions */

```

```

/** Function Prototypes */
void initialize(void);

```

```

***** MAIN ROUTINE
*****/
void main(void)
{

```

```

/* local variables */
int index = 0;
int period = 30000;

```

```

/* initialize basic registers and functions of DSP */
initialize();

```

```

/** Setup shared I/O pins */
*MCRA = 0x0000; /* group A pins */
/* Bit 15 is I/O PB7 control bit, 0 means I/O, 1 other function (clock)
bit 15 0: 0=IOPB7, 1=TCLKINA
bit 14 0: 0=IOPB6, 1=TDIRA
bit 13 0: 0=IOPB5, 1=T2PWM/T2CMP
bit 12 0: 0=IOPB4, 1=T1PWM/T1CMP
bit 11 0: 0=IOPB3, 1=PWM6
bit 10 0: 0=IOPB2, 1=PWM5
bit 9 0: 0=IOPB1, 1=PWM4
bit 8 0: 0=IOPB0, 1=PWM3

```

```

bit 7    0:    0=IOPA7,   1=PWM2
bit 6    0:    0=IOPA6,   1=PWM1
bit 5    0:    0=IOPA5,   1=CAP3
bit 4    0:    0=IOPA4,   1=CAP2/QEP2
bit 3    0:    0=IOPA3,   1=CAP1/QEP1
bit 2    0:    0=IOPA2,   1=XINT1
bit 1    0:    0=IOPA1,   1=SCIRXD
bit 0    0:    0=IOPA0,   1=SCITXD
*/

```

```

    *MCRB = 0xFE00;          /* group B pins */
/*
bit 15   1:    0=reserved, 1=TMS2 (always write as 1)
bit 14   1:    0=reserved, 1=TMS  (always write as 1)
bit 13   1:    0=reserved, 1=TD0  (always write as 1)
bit 12   1:    0=reserved, 1=TDI  (always write as 1)
bit 11   1:    0=reserved, 1=TCK  (always write as 1)
bit 10   1:    0=reserved, 1=EMU1 (always write as 1)
bit 9    1:    0=reserved, 1=EMU0 (always write as 1)
bit 8    0:    0=IOPD0,   1=XINT2/ADCSOC
bit 7    0:    0=IOPC7,   1=CANRX
bit 6    0:    0=IOPC6,   1=CANTX
bit 5    0:    0=IOPC5,   1=SPISTE
bit 4    0:    0=IOPC4,   1=SPICLK
bit 3    0:    0=IOPC3,   1=SPISOMI
bit 2    0:    0=IOPC2,   1=SPISIMO
bit 1    0:    0=IOPC1,   1=BIO*
bit 0    0:    0=IOPC0,   1=W/R*
*/

```

```

    *MCRC = 0x0000;          /* group C pins */
/*
bit 15   0:    reserved
bit 14   0:    0=IOPF6,   1=IOPF6
bit 13   0:    0=IOPF5,   1=TCLKINB
bit 12   0:    0=IOPF4,   1=TDIRB
bit 11   0:    0=IOPF3,   1=T4PWM/T4CMP
bit 10   0:    0=IOPF2,   1=T3PWM/T3CMP
bit 9    0:    0=IOPF1,   1=CAP6
bit 8    0:    0=IOPF0,   1=CAP5/QEP4
bit 7    0:    0=IOPE7,   1=CAP4/QEP3
bit 6    0:    0=IOPE6,   1=PWM12
bit 5    0:    0=IOPE5,   1=PWM11
bit 4    0:    0=IOPE4,   1=PWM10
bit 3    0:    0=IOPE3,   1=PWM9
bit 2    0:    0=IOPE2,   1=PWM8

```

```

bit 1    0:    0=IOPE1,   1=PWM7
bit 0    0:    0=IOPE0,   1=CLKOUT
*/
    assigns 1111,1100,0000,0000 to address 0X709C
    this address stores PCDATDIR register for I/O port C data (0 or 1) and
    direction (input or output)
/***/ Configure port C /***/
    *PCDATDIR = 0xFC00; /* set c0,1 as inputs and c2-7 as outputs/low*/
*/
bits 15-8 0=INPUT,   1=OUTPUT
bits 7-0  0=LOW,     1=HIGH
*/
    Timer 1 controller
/***/ Timer 1 setup /***/
    *T1CON = 0x0000; /* disable timer 1 */
    General purpose timer control register
    *GPTCONA = 0x0000; /* configure GPTCONA */
*/
bit 15    0:    reserved
bit 14    0:    T2STAT, read-only
bit 13    0:    T1STAT, read-only
bit 12-11 00:   reserved
bit 10-9  00:   T2TOADC, 00 = no timer2 event starts ADC
bit 8-7   00:   T1TOADC, 00 = no timer1 event starts ADC
bit 6     0:    TCOMPOE, 0 = Hi-z all timer compare outputs
bit 5-4   00:   reserved
bit 3-2   00:   T2PIN, 00 = forced low
bit 1-0   00:   T1PIN, 00 = forced low
*/
    *T1CNT = 0x0000; /* clear timer counter */
    *T1PR = period; /* set timer period */
    *DBTCONA = 0x0000; /* deadband units off */
    *T1CON = 0x0F40; /* configure T1CON register */
*/
bit 15-14 00:   stop immediately on emulator suspend
bit 13     0:   reserved
bit 12-11 01:   01 = continous-up/down count mode
bit 10-8   111: 000 = x/1 prescaler
bit 7      0:   reserved in T1CON
bit 6      1:   TENABLE, 1 = enable timer
bit 5-4    00:  00 = CPUCLK is clock source
bit 3-2    00:  00 = reload compare reg on underflow
bit 1      0:   0 = disable timer compare
bit 0      0:   reserved in T1CON
*/

```

```

/** Proceed with main routine */
while(1)
{
    index = *PCDATDIR & 0x0003;
    switch(index)
    {
        case 0:
            if (*T1CNT >= (period/2)) /* blink LEDs */
            {
                *PCDATDIR = *PCDATDIR | 0x000C;
            }
            else
            {
                *PCDATDIR = *PCDATDIR & 0xFFF3;
            }
            break;
        case 1:
            *PCDATDIR = *PCDATDIR | 0x0004; /* turn off LED A */
            *PCDATDIR = *PCDATDIR & 0xFFF7;
            /* turn on LED B */
            break;
        case 2:
            *PCDATDIR = *PCDATDIR | 0x0008; /* turn off LED B */
            *PCDATDIR = *PCDATDIR & 0xFFFB; /* turn on LED A */
            break;
        case 3:
            *PCDATDIR = *PCDATDIR | 0x000C; /* turn off LEDs */
            break;
        default:
            *PCDATDIR = *PCDATDIR | 0x000C; /* turn off LEDs */
    }
}
}

```

ends main program

```

void initialize(void)
{
    /** Configure the System Control and Status registers */
    *SCSR1 = 0x00FD;
    /*
    bit 15    0:    reserved
    bit 14    0:    CLKOUT = CPUCLK
    bit 13-12 00:   IDLE1 selected for low-power mode
    bit 11-9  000:  PLL x4 mode
    bit 8     0:    reserved
    bit 7     1:    1 = enable ADC module clock
    bit 6     1:    1 = enable SCI module clock
    */
}

```

bit 5 1: 1 = enable SPI module clock
bit 4 1: 1 = enable CAN module clock
bit 3 1: 1 = enable EVB module clock
bit 2 1: 1 = enable EVA module clock
bit 1 0: reserved
bit 0 1: clear the ILLADR bit
*/

```
*SCSR2 = (*SCSR2 | 0x000B) & 0x000F;  
/*  
bit 15-6 0's: reserved  
bit 5 0: do NOT clear the WD OVERRIDE bit  
bit 4 0: XMIF_HI-Z, 0=normal mode, 1=Hi-Z'd  
bit 3 1: disable the boot ROM, enable the FLASH  
bit 2 no change MP/MC* bit reflects state of MP/MC* pin  
bit 1-0 11: 11 = SARAM mapped to prog and data  
*/
```

```
/**/ Disable the watchdog timer /**/  
*WDCR = 0x00E8;  
/*
```

```
bits 15-8 0's: reserved  
bit 7 1: clear WD flag  
bit 6 1: disable the dog  
bit 5-3 101: must be written as 101  
bit 2-0 000: WDCLK divider = 1  
*/
```

```
/**/ Setup external memory interface for LF2407 EVM /**/  
WSGR = 0x0040;  
/*
```

```
bit 15-11 0's: reserved  
bit 10-9 00: bus visibility off  
bit 8-6 001: 1 wait-state for I/O space  
bit 5-3 000: 0 wait-state for data space  
bit 2-0 000: 0 wait state for program space  
*/  
}
```