



LS7266R1 Application Note



LSI Computer Systems, Inc. 1235 Walt Whitman Road, Melville, NY 11747 (631) 271-0400 FAX (631) 271-0405

C Sample Routines for Interfacing with LS7266R1

May 2004

```

#include<stdlib.h>
#include <stdio.h>
#include <conio.h>

#define XDATA(arg) (arg +0)
#define XCMD (arg) (arg + 1)
#define YDATA (arg) (arg +2)
#define YCMD (arg) (arg +3)

// RLD Reg.
#define RLD (arg) (arg | 0x80)
#define XRLD (arg) (arg | 0)
#define YRLD (arg) XRLD(arg)
#define Rst_BP 0x01
#define Rst_CNTR 0x02
#define Rst_FLAGS 0x04
#define Rst_E 0x06
#define Trf_PR_CNTR 0x08
#define Trf_CNTR_OL 0x10
#define Trf_PS0_PSC 0x18

//CMR Reg.
#define CMR(arg) (arg | 0xA0)
#define XCMR(arg) (arg | 0x20)
#define YCMR(arg) XCMR(arg)
#define BINCnt 0x00
#define BCDCnt 0x01
#define NrmCnt 0x00
#define RngLmt 0x02
#define NRcyc 0x04
#define ModN 0x06
#define NQDX 0x00
#define QDX1 0x08
#define QDX2 0x10
#define QDX4 0x18

//IOR Reg.
#define IOR(arg) (arg | 0xC0)
#define XIOR(arg) (arg | 0x40)
#define YIOR(arg) XIOR(arg)
#define DisAB 0x00
#define EnAB 0x01

#define LCNTR 0x00
#define LOL 0x02
#define RCNTR 0x00
#define ABGate 0x04
#define CYBW 0x00
#define CPBW 0x08
#define CB_UPDN 0x10
#define IDX_ERR 0x18

// IDR
#define IDR(arg) (arg | 0xE0)
#define XIDR(arg) (arg | 0x60)
#define YIDR(arg) XIDR(arg)
#define DisIDX 0x00
#define EnIDX 0x01
#define NIDX 0x00
#define PIDX 0x02
#define LIDX 0x00
#define RIDX 0x04

void Init_7266(int Addr);
/* Initialize 7266 as follows (X + Y CNTR)
Modulo N count mode for N = 0x123456
Binary Counting
Index on LCNTR/LOL Input
CY and BW outputs
RCNTR/ABG controls Counters
A and B Enabled
*/
void Init_7266(int Addr)
{
/Setup IOR Reg.
outp(XCMD(Addr),IOR(DisAB + LOL + ABGate + CYBW)); //Disable Counters and Set CY BW Mode

//Setup RLD Reg.
outp(XCMD(Addr),RLD(Rst_BP + Rst_FLAGS)); //Reset Byte Pointer(BP) And Flags
outp(XDATA(Addr),0x06); //Load 6 to PR0 to setup Transfer to PS0
outp(YDATA(Addr),0x06); //Load 6 to PR0 to setup Transfer to PS0
outp(XCMD(Addr),RLD(Rst_E + Trf_PS0_PSC)); //Reset E Flag and Transfer PR0 to PSC
outp(XCMD(Addr),RLD(Rst_BP + Rst_CNTR)); //Reset BP and Reset Counter

//Setup IDR Reg.
outp(XCMD(Addr),IDR(EnIDX + NIDX + LIDX)); //Enable Negative Index on LCNTR/LOL Input

//Setup CMR Reg.
outp(XCMD(Addr),CMR(BINCnt + ModN + QDX4)); //Set Binary Mondulo N Quadrature x4

```

```

//Setup PR Reg. for Modulo N Counter to 0x123456
outp(XDATA(Addr),0x56); //Least significant Byte first
outp(XDATA(Addr),0x34); //then middle byte
outp(XDATA(Addr),0x12); //then most significant byte
//Setup PR Reg. for Modulo N Counter to 0x123456
outp(YDATA(Addr),0x56); //Least significant Byte first
outp(YDATA(Addr),0x34); //then middle byte
outp(YDATA(Addr),0x12); //then most significant byte

//Enable Counters
outp(XCMD(Addr),IOR(EnAB));

}

/* Write_7266_PR
Input: Addr has Address of 7266 counter.
Data: has 24 bit data to be written to PR register
*/
void Write_7266_PR(int Addr,unsigned long Data);
void Write_7266_PR(int Addr,unsigned long Data)
{
    outp(XCMD(Addr),RLD(Rst_BP)); //Reset Byte Pointer to Synchronize Byte Writing
    outp(XDATA(Addr),(unsigned char)Data);
    Data >>= 8;
    outp(XDATA(Addr),(unsigned char)Data);
    Data >>= 8;
    outp(XDATA(Addr),(unsigned char)Data);
}

/* Read_7266_OL
Input: Addr has Address of 7266 counter.
Output: Data returns 24 bit OL register value.
*/
unsigned long Read_7266_OL(int Addr);
unsigned long Read_7266_OL(int Addr)
{
    unsigned long Data=0;
    outp(XCMD(Addr),(RLD(Rst_BP + Trf_Cntr_OL)); //Reset Byte Pointer to Synchronize Byte reading and
                                                Transferring of data from counters to OL.

    Data |= (unsigned long)inp(XDATA(Addr)); //read byte 0 from OL
    lrotr(Data,8); //Rotate for next Byte
    Data |= (unsigned long)inp(XDATA(Addr)); //read byte 1 from OL
    lrotr(Data,8); //Rotate for next Byte
    Data |= (unsigned long)inp(XDATA(Addr)); //read byte 2 from OL
    lrotr(Data,16); //Rotate for last Byte
    return(Data);
}

/* Get_7266_Flags
Input: Addr has Address of 7266 counter.
returns Flags of counter
*/
unsigned char Get_7266_Flags(int Addr);
unsigned char Get_7266_Flags(int Addr)
{
    return(inp(CMD(Addr)));
}

```