EVM2812 Software Instruction

EVM2812 development board testing program includes:

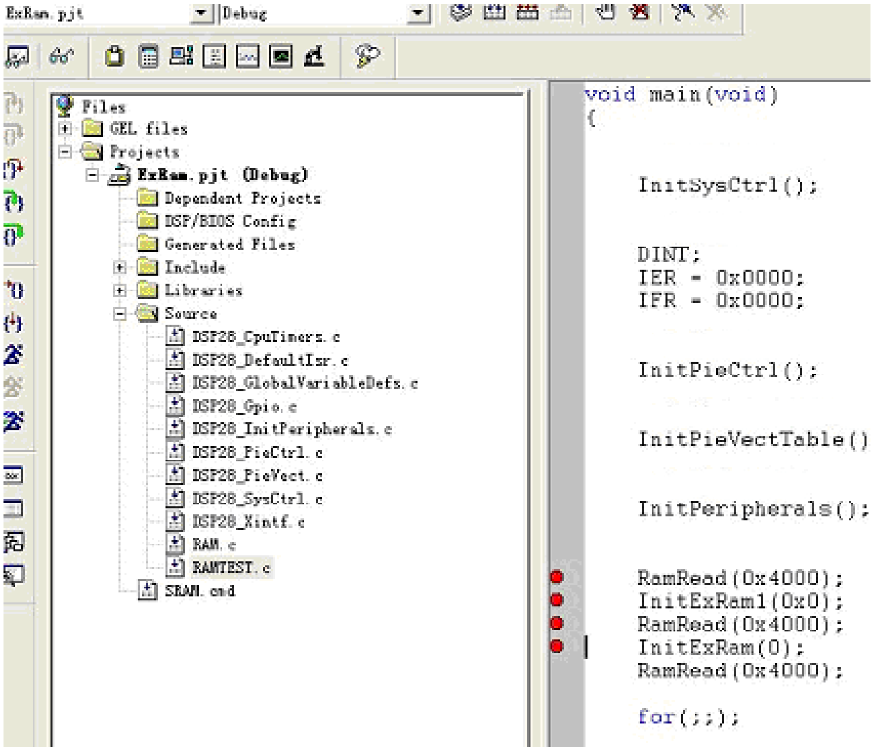
Test SRAM of EVM2812
Test serial RS232 interface of EVM2812
Test DAC of EVM2812

Note: testing tool is through USB emulator. For convenience, we recommend users to use our USB emulator. If you are using other emulators, please refer to their user manuals.

1. ExRam Test

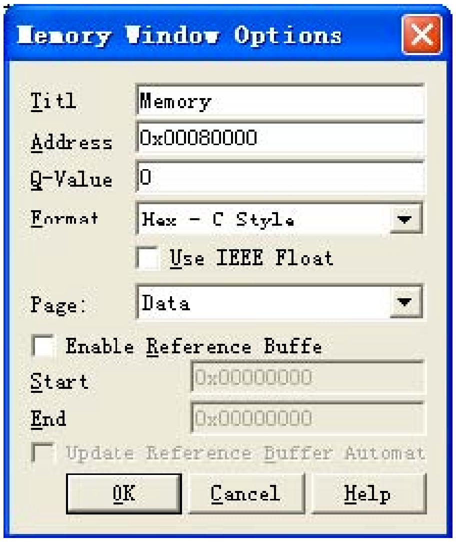
   There is an extended 256K*16 bit SRAM, whose address projects to Zone2. SRAM mainly writes to 0X08000 – 0X084000 of SRAM, meaning that it writes 0 – 0X3FFF to 0X08000 – 0X084000 and then reads out to 0X084000 – 0X087FFF. Users can monitor these two address ranges.

   a. COPY *sy-examples* index to *myproject* index in CCS environment;
   b. In CCS, Project→Open, OPEN *ExRam.pjt* in ..\syexamples\v100\DSP281x_examples\ExRam
   c. Use File→Load Program, LOAD *ExRam.out* in debug index of ExRam folder;
   d. Run main( ) function using Debug→Go Main;
   e. Refer to the following image, and press F5 to run.



```
void main(void)
{

    InitSysCtrl();


    DINT;
    IER = 0x0000;
    IFR = 0x0000;


    InitPieCtrl();


    InitPieVectTable()


    InitPeripherals();


    RamRead(0x4000);
    InitExRam1(0x0);
    RamRead(0x4000);
    InitExRam(0);
    RamRead(0x4000);

    for(;;);
```

   f. Click View→Memory… to input address, as shown below:



Memory Window Options

Titl: Memory

Address: 0x00080000

Q-Value: 0

Format: Hex - C Style

□ Use IEEE Float

Page: Data

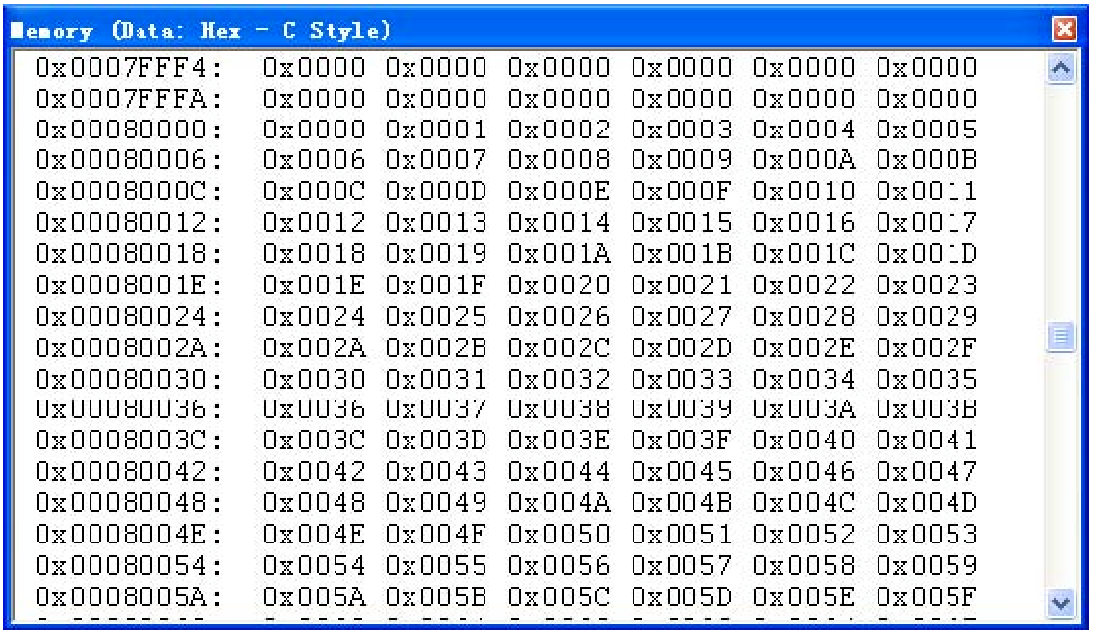
□ Enable Reference Buffe

Start: 0x00000000

End: 0x00000000

□ Update Reference Buffer Automat

After pressing OK, you will see the following screen, if the values increases from 0X0000 to 0X3FFF, it means the external RAM is fine, otherwise please check on the external RAM.



Memory (Data: Hex - C Style)

```
0x0007FFF4:   0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
0x0007FFFA:   0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
0x00080000:   0x0000 0x0001 0x0002 0x0003 0x0004 0x0005
0x00080006:   0x0006 0x0007 0x0008 0x0009 0x000A 0x000B
0x0008000C:   0x000C 0x000D 0x000E 0x000F 0x0010 0x00:1
0x00080012:   0x0012 0x0013 0x0014 0x0015 0x0016 0x00:7
0x00080018:   0x0018 0x0019 0x001A 0x001B 0x001C 0x00:D
0x0008001E:   0x001E 0x001F 0x0020 0x0021 0x0022 0x0023
0x00080024:   0x0024 0x0025 0x0026 0x0027 0x0028 0x0029
0x0008002A:   0x002A 0x002B 0x002C 0x002D 0x002E 0x002F
0x00080030:   0x0030 0x0031 0x0032 0x0033 0x0034 0x0035
UxUUU8UU36:   UxUU36 UxUU37 UxUU38 UxUU39 UxUU3A UxUU3B
0x0008003C:   0x003C 0x003D 0x003E 0x003F 0x0040 0x0041
0x00080042:   0x0042 0x0043 0x0044 0x0045 0x0046 0x0047
0x00080048:   0x0048 0x0049 0x004A 0x004B 0x004C 0x004D
0x0008004E:   0x004E 0x004F 0x0050 0x0051 0x0052 0x0053
0x00080054:   0x0054 0x0055 0x0056 0x0057 0x0058 0x0059
0x0008005A:   0x005A 0x005B 0x005C 0x005D 0x005E 0x005F
```

2. RS232 Interface Testing
   EVM2812 board sends data using TX port of DSP's SCIB. You can also monitor those sending data of serial port.
   a. COPY *sy-examples* index to *myproject* index in CCS environment;
   b. In CCS, Project→Open, OPEN *scitest.pjt* in ..\syexamples\v100\DSP281x_examples\scitest

c. Use File→Load Program, LOAD *ExRam.out* in debug index of scitest folder, then press F5 to run;

d. In the serial port debug assistant, set Baud rate to 19200, 8 data bits, 1 stop bit, and no check digit. Observed received data in the receive window, check if the following data comes out circularly. If it does, it means the port is normal.

```
FF FE FD FC FB FA F9 F8
FE FD FC FB FA F9 F8 F7
FD FC FB FA F9 F8 F7 F6
....
01 00 FF FE FD FC FB FA
00 FF FE FD FC FB FA F9
```

3. DAC Testing

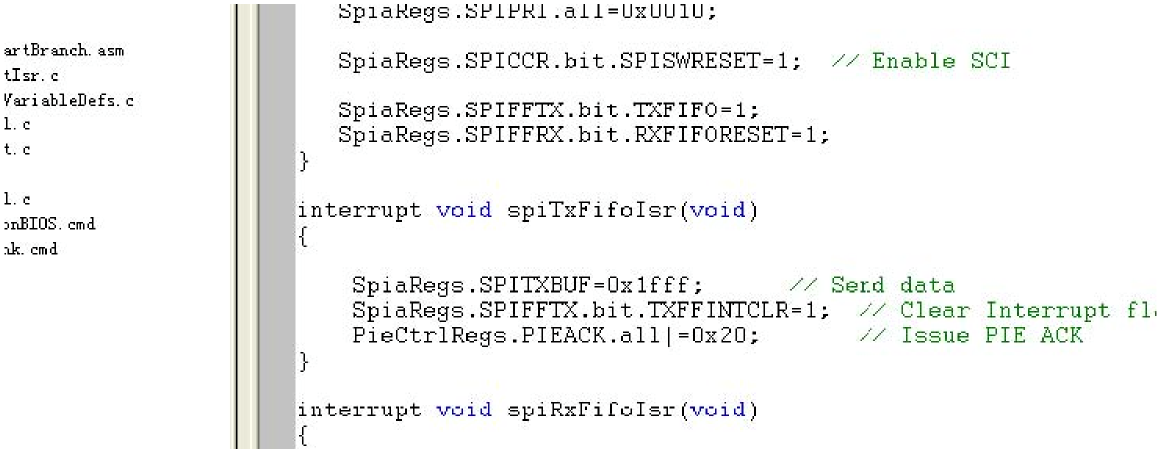
There is DAC interface on board. DSP uses SPI to test 4 DAC channels output of TLV5614.

a. COPY *sy-examples* index to *myproject* index in CCS environment;

b. In CCS, Project→Open, OPEN spitest.*pjt* in ..\syexamples\v100\DSP281x_examples\spitest

c. Use File→Load Program, LOAD *spitest.out* in debug index of scitest folder;

d. As shown below, find SpiaRegs.SPITXBUF=0x1fff in spiTxFifolsr( ) function.



```
   SpiaRegs.SPIPRI.all=0x0010;

   SpiaRegs.SPICCR.bit.SPISWRESET=1;  // Enable SCI

   SpiaRegs.SPIFFTX.bit.TXFIFO=1;
   SpiaRegs.SPIFFRX.bit.RXFIFORESET=1;
}

interrupt void spiTxFifoIsr(void)
{

    SpiaRegs.SPITXBUF=0x1fff;      // Send data
    SpiaRegs.SPIFFTX.bit.TXFFINTCLR=1;  // Clear Interrupt fl
    PieCtrlRegs.PIEACK.all|=0x20;       // Issue PIE ACK
}

interrupt void spiRxFifoIsr(void)
{
```

These values correspond to the following meanings:
0x1fff~0x1000 corresponds to A channel of DAC, maximum output voltage is 3.3V~0V
0x2fff~0x2000 corresponds to B channel of DAC, maximum output voltage is 3.3V~0V
0x3fff~0x3000 corresponds to C channel of DAC, maximum output voltage is 3.3V~0V
0x4fff~0x4000 corresponds to D channel of DAC, maximum output voltage is 3.3V~0V

Users can modify values of SpiaRegs.SPITXBUF according to above definition, also use Project→Rebuild All… in CCS to re-edit Spitest.prj project. Use File→Load Program, LOAD *spitest.out* in debug index of scitest folder, and then press F5 to run. You can now use multi-meter or oscillator to observe voltage change of corresponding DAC pins.