


```

00001:
00002: /*
00002: ****
00002: ****
00003: *
00004: *
00005: *
00006: *
00007: *
00008: * File : SRC4392.c
00009: * author: CQ
00010: * REV : 00
00011: * Description: .This module is used to process
00012: *
00013: *
00014: ****
00014: ****/
00015:
00016: #include "Includes.h"
00017:
00018:
00019: /*
00019: ****
00019: ****
00020: *
00021: *
00022: ****
00022: ****/
00023:
00024:
00025:
00026:
00027: /*
00027: ****
00027: ****
00028: *
00029: *
00030: ****
00030: ****/
00031:
00032:
00033:
00034:
00035: /*
00035: ****
00035: ****
00036: *
00037: *
00038: ****
00038: ****/
00039:
00040:
00041:
00042:
00043: /*
00043: =====
00043: =====
00044:
00045:
00046:
00047: =====
00047: =====*/
00048: int main (void)
00049: {
00050:     INT8U config_data, target_addr;
00051:
00052:
00053:     _delay_ms(100);
00054:     SystemInit();
00055:
00056:     //config sm5847
00057:     SM5847_CONFIG_DIR_PORT = 0xFF; //port as output;
00058:     SM5847_CONFIG_ENABLE_DIR_PORT |= 0x0C; //bit3,bit3 as output;
00059:     CLRBIT(SM5847_CONFIG_ENABLE_PORT,BIT(SM5847_EN1));
00060:     CLRBIT(SM5847_CONFIG_ENABLE_PORT,BIT(SM5847_EN2));
00061:
00062:     /*config parameter1
00063:     CKSLN = 1 192fs
00064:     1W1N = 0 input bit long:24bit
00065:     1W2N = 0
00066:     OW1N = 0 output bit long: 24bit
00067:     OW2N = 0

```

```

00068:     SYNCN    = 1
00069:     CKDV1    = 0           divider ratio: 1
00070:     CKDV2    = 0
00071:     */
00072:
00073:     SM5847_CONFIG_PORT = BIT(CKSLN) | BIT(SYNCN);
00074:     _delay_us(5);
00075:     SETBIT(SM5847_CONFIG_ENABLE_PORT,BIT(SM5847_EN1));
00076:     _delay_us(100);
00077:     CLRBIT(SM5847_CONFIG_ENABLE_PORT,BIT(SM5847_EN1));
00078:     _delay_us(5);
00079:
00080:     /*config parameter2
00081:     DEMPL    = 0           disable DEMPL
00082:     DEMPR    = 0           disable DEMPR
00083:     FSEL1    = 0
00084:     FSEL2    = 0
00085:     DITHD    = 1
00086:     OMD      = 1           8FS
00087:     */
00088:     SM5847_CONFIG_PORT = BIT(DITHD) | BIT(OMD);
00089:     _delay_us(5);
00090:     SETBIT(SM5847_CONFIG_ENABLE_PORT,BIT(SM5847_EN2));
00091:     _delay_us(100);
00092:     CLRBIT(SM5847_CONFIG_ENABLE_PORT,BIT(SM5847_EN2));
00093:     _delay_us(5);
00094:
00095:
00096:
00097:     //configuration SRC4392 with SPI mode
00098:     //config_data = BIT(PDPB) | BIT(PDPA);
00099:
00100:     config_data = BIT(PDSRC) | BIT(PDRX) | BIT(PDPB) | BIT(PDPA);
00101:     target_addr = POWERDOWN;
00102:
00103:     WriteDataToSrc4392(target_addr, config_data);
00104:     _delay_ms(1);
00105:     //config_data = ReadDataFromSrc4392(target_addr);
00106:     //_delay_ms(1);
00107:
00108:     //GLOBINTSTAS (default)
00109:
00110:     // PortA: 24bit I2S MODE ,Master mode , SRC output source
00111:     //porta ctrl1
00112:     //config_data = BIT(AFMTO) | BIT(AMS) | BIT(AOUTS1) | BIT(AOUTS0);
00113:     config_data = BIT(AMS) | BIT(AOUTS1);
00114:     target_addr = PACTL1;
00115:     WriteDataToSrc4392(target_addr, config_data);
00116:     _delay_ms(1);
00117:     //config_data = ReadDataFromSrc4392(target_addr);
00118:
00119:     //PACTL2: default
00120:
00121:
00122:     //PortB: 24bit I2S MODE ,Slave mode , input mode
00123:     config_data = BIT(BFMTO) | BIT(BOUTS0);
00124:     target_addr = PBCTL1;
00125:     WriteDataToSrc4392(target_addr, config_data);
00126:     _delay_ms(1);
00127:     //config_data = ReadDataFromSrc4392(target_addr);
00128:
00129:     //PBCTL2: default
00130:
00131:     //TXCTL1
00132:     //TXCTL2
00133:     //TXCTL3
00134:     //SDSTAS
00135:     //SDINTMASK
00136:     //SDINTMODE
00137:
00138:     // RX4 is the default channel, MCLK clock
00139:     config_data = BIT(RXMUX1) | BIT(RXMUX0) | BIT(RXCLK);
00140:     target_addr = RXCTL1;
00141:     WriteDataToSrc4392(target_addr, config_data);
00142:     _delay_ms(1);
00143:     //config_data = ReadDataFromSrc4392(target_addr);
00144:
00145:     //RXCTL2

```

```

00146:     config_data = BIT(RXCKOE) | BIT(LOL);
00147:     target_addr = RXCTL2;
00148:     WriteDataToSrc4392(target_addr, config_data);
00149:     _delay_ms(1);
00150:
00151:     //RXPLLCFG1
00152:     config_data = BIT(P1) | BIT(J3);
00153:     target_addr = RXPLLCFG1;
00154:     WriteDataToSrc4392(target_addr, config_data);
00155:     _delay_ms(1);
00156:
00157:     //RXPLLCFG2
00158:     //RXPLLCFG3
00159:     //NPCMAUDDDET
00160:     //RXSTAS1
00161:     //RXSTAS2
00162:     //RXSTAS3
00163:     //RXINTMASK1
00164:     //RXINTMASK2
00165:     //RXINTMODE1
00166:     //RXINTMODE2
00167:     //RXINTMODE3
00168:     //GPO1
00169:     //GPO2
00170:     //GPO3
00171:     //GPO4
00172:     //AUDQSUBCODE1
00173:     //AUDQSUBCODE2
00174:     //AUDQSUBCODE3
00175:     //AUDQSUBCODE4
00176:     //AUDQSUBCODE5
00177:     //AUDQSUBCODE6
00178:     //AUDQSUBCODE7
00179:     //AUDQSUBCODE8
00180:     //AUDQSUBCODE9
00181:     //AUDQSUBCODE10
00182:     //PCBPH
00183:     //PCBPL
00184:     //PDBPH
00185:     //PDBPL
00186:
00187:     //SRC input source: DIR
00188:     config_data = BIT(SRCIS1) | BIT(SRCISO) | BIT(SRCCLK0) | BIT(SRCCLK1);
00189:     //config_data = BIT(SRCISO); //PORTB INPUT
00190:     target_addr = SRCCTL1;
00191:     WriteDataToSrc4392(target_addr, config_data);
00192:     _delay_ms(1);
00193:     //config_data = ReadDataFromSrc4392(target_addr);
00194:
00195:     //SRCCTL2
00196:     //SRCCTL3
00197:     //SRCCTL4
00198:     //SRCCTL5
00199:     //SRCIORATIO1
00200:     //SRCIORATIO2
00201:     //PAGESELN
00202:
00203:     while(1)
00204:     {
00205:         _delay_ms(1);
00206:         _delay_ms(1);
00207:     }
00208:
00209: } ? end main ?
00210:
00211:
00212:
00213:
00214:
00215:
00216: /*
00216: *****
00216: *****
00217: *                               void WriteDataToSrc4392(void)
00218: *
00219: * Description:
00220: *
00221: * Input:      none
00222: *
00223: * Output:     none
00224: *
00225: * Globals used: none

```

```

00226: *
00227: * Return:      none
00228: *****
00228: *****/
00229: void WriteDataToSrc4392(UINT8 address, UINT8 data)
00230: {
00231:     UINT8 write_address;
00232:
00233:     write_address = address;
00234:     SPI_CS_ENABLE();
00235:     _delay_us(10);
00236:     SPIWriteData(write_address);
00237:     SPIWriteData(0);
00238:     SPIWriteData(data);
00239:     _delay_us(10);
00240:     SPI_CS_DISABLE();
00241:
00242: }
00243:
00244: INT8U ReadDataFromSrc4392(UINT8 address)
00245: {
00246:     UINT8 read_address, read_data;
00247:
00248:     read_address = 0x80 | address;
00249:
00250:     SPI_CS_ENABLE();
00251:     _delay_us(10);
00252:     SPIWriteData(read_address);
00253:     SPIWriteData(0);
00254:     read_data = SPIWriteData(0);
00255:     SPI_CS_DISABLE();
00256:
00257:     return read_data;
00258: }
00259:
00260:
00261:
00262:
00263: /*
00263: *****
00263: *****
00264: *
00264: *           void SPIWriteData(void)
00265: *
00266: * Description:
00267: *
00268: * Input:      none
00269: *
00270: *Output:      none
00271: *
00272: *Globals used: none
00273: *
00274: * Return:      none
00275: *****
00275: *****/
00276:
00277: INT8U SPIWriteData(UINT8 data)
00278: {
00279:     SPDR = data;
00280:     while(!CHECKBIT(SPSR,BIT(SPIF)));
00281:
00282:     return SPDR;
00283: }
00284:
00285:
00286:
00287: /*
00287: *****
00287: *****
00288: *
00288: *           void SPIReadData(void)
00289: *
00290: * Description:
00291: *
00292: * Input:      none
00293: *
00294: *Output:      none
00295: *
00296: *Globals used: none
00297: *
00298: * Return:      none
00299: *****
00299: *****/

```

00300:
00301:
00302:
00303: