

```
#include "defs.h"
#include "music.h"

unsigned const char explode[EXPLODE];

unsigned const char whistle[WHISTLE];

int soundEffect;

// 9S12DP512 SPI1 interface to Max539
// PS6 (out) SCLK synchronous clock
// PS5 (out) MOSI serial data output
// PS7 (out) CS used to latch data into Max539
// PS4 (in) is associated with SPI1, but not used

//-----DAC_Init-----
// initializes DAC
// Input: none
// Output: none
void DAC_Init(void) {
    SS_DDR = 1; // 1) make PS5, PS6, PS7 outputs, PS4 input
    MOSI_DDR = 1;
    SCK_DDR = 1; // DDRS

    SPICR1 = 0x58; // 2) enable SPI, no interrupts, master, CPOL=1, CPHA=0
                // SPI0CR1 = 0101 1000
    SPICR2 = 0x00; // 3) set up PS7 as a regular output
                // SSOE=0, MODFEN=0 SPI0CR1, SPI0CR2
    SPIBR = 0x00; // 4) set the baud rate, SPI0BR
    SS = 1; // 5) make PS7=CS high
}

//-----transmitByte-----
// outputs byte to DAC
// Input: none
// Output: none
void transmitByte(unsigned char data) {
    unsigned char dummy;
    while(!(SPISR&0x20)) {} // 1) wait for SPTEF to be 1, SPI0SR
    SPIDR = data; // 2) write 8-bit data to SPI0DR
    while(!(SPISR&0x80)) {} // 3) wait for SPIF to be 1, SPI0SR
    dummy = SPIDR; // 4) clear the SPIF flag by reading the data
                // dummy = SPI0DR;
}

//-----DAC_Out-----
// outputs 12 bits to DAC
// Input: none
// Output: none
void DAC_Out(unsigned char data) {
    SS = 0; // 1) set PS7=CS low
    //transmitByte((data&0x3F00) >> 8); // 2) transmit most significant 8-bit data to the DAC
    transmitByte(0);
    transmitByte(data); // 3) transmit least significant 8-bit data to the DAC
    SS = 1; // 4) set PS7=CS high
}

//-----Music_InitOC0-----
// arm output compare 0 for melody
// also enables timer to 43 ns period
// Input: none
// Output: none
void Music_InitOC7(void) {
    TIOS |= 0x80; // activate TC0 as output compare
}

void Music_EnableOC7(int sound) {
    while(TIE&0x80);
    soundEffect = sound;
    TIE |= 0x80;
    TC7 = TCNT+50; // first interrupt right away
}

// OC handler for melody
interrupt 15 void TC7Handler() {
    unsigned static long i = 0;
    TFLG1 = 0x80;
    if(i >= soundEffect) {
```

```
    i = 0;  
    TIE &= ~0x80;  
}  
else {  
    if(soundEffect == EXPLODE) {  
        DAC_Out(explode[i]);  
    }  
    else {  
        DAC_Out(whistle[i]);  
    }  
    i++;  
}  
  
TC7 += 187;  
}
```