```
 1: #include "DAC.h"
 2:
 3: // 9S12DP512 SPI1 interface to Max539
 4: // PS6 (out) SCLK synchronous clock
 5: // PS5 (out) MOSI serial data output
 6: // PS7 (out) CS used to latch data into Max539
 7: // PS4 (in) is associated with SPI1, but not used
 8:
 9: //---------------------DAC_Init----------------------
10: // initializes DAC
11: // Input: none
12: // Output: none
13: void DAC_Init(void) {
14:   DDRS |=  0xE0;  // 1) make PS5, PS6, PS7 outputs, PS4 input
15:   DDRS &= ~0x10;  // DDRS
16:   SPI0CR1 = 0x58; // 2) enable SPI, no interrupts, master, CPOL=1, CPHA=0
17:                   // SPI0CR1 = 0101 1000
18:   SPI0CR2 = 0x00; // 3) set up PS7 as a regular output
19:                   // SSOE=0, MODFEN=0 SPI0CR1, SPI0CR2
20:   SPI0BR = 0x00;  // 4) set the baud rate, SPI0BR
21:   PTS  |= 0x80;   // 5) make PS7=CS high
22: }
23:
24: //---------------------transmitByte---------------------
25: // outputs byte to DAC
26: // Input: none
27: // Output: none
28: void transmitByte(unsigned char data) {
29:   unsigned char dummy;
30:   while(!(SPI0SR&0x20)) {}  // 1) wait for SPTEF to be 1, SPI0SR
31:   SPI0DR = data;            // 2) write 8-bit data to SPI0DR
32:   while(!(SPI0SR&0x80)) {}  // 3) wait for SPIF to be 1, SPI0SR
33:   dummy = SPI0DR;            // 4) clear the SPIF flag by reading the data
34:                             // dummy = SPI0DR;
35: }
36:
37: //---------------------DAC_Out----------------------
38: // outputs 12 bits to DAC
39: // Input: none
40: // Output: none
41: void DAC_Out(unsigned short data) {
42:   PTS &= ~0x80;                    // 1) set PS7=CS low
43:   transmitByte((data&0x3F00) >> 8); // 2) transmit most significant 8-bit data to the DAC
44:   transmitByte(data&0x00FF);        // 3) transmit least significant 8-bit data to the DAC
45:   PTS |=  0x80;                    // 4) set PS7=CS high
46: }
```