PIC C CODE FOR DIGITAL POTS

Digital Pots are commonly used in many types of applications, that use a micro-processor such as a PIC to control the Pot setting. Typically, either a switch or feedback from an A/D Converter is used as the controlling device. And the software must communicate with the Digital Pot.

The following code examples are used for writing data to Serial Digital Pots. The examples are for Dallas Semiconductor DS1867 Dual Digital POT and for the Analog Devices AD8400, AD8402 and AD8404 1, 2, and 4 channel Digital Pots. This code can easily be adapted to communicate with many types of serial devices.

EXAMPLE 1

```
/*********************************************************************/
/**                                                                 **/
/**  WRSER17 - Write 17 bits of serial data out Sdata/Sclk          **/
/**                                                                 **/
/**  This routine sends out 17 bits of data to the DS1867 digital   **/
/**  POT.   Data is shifted out on the pin name Sdata and uses a    **/
/**  pin named Sclk to clock data on the rising edge.               **/
/**  The MSbyte of data is stored in SERMSB bits 7..0 and the       **/
/**  LSByte is stored in SERLSB.  Data is sent out MSBit first      **/
/**  (D16...D0).  This data must be in SERMSB and SERLSB before     **/
/**  the routine is called.  The 17th bit D16 is always 0, this is  **/
/**  the stack select bit and is not used.                          **/
/**                                                                 **/
/**  Data is shifted out D16....D0, ie, stack select first, Pot     **/
/**  number 1 data, the pot number 0 data.                          **/
/**                                                                 **/
/** Requires the main source file to have a char variable called    **/
/** SERMSB and a char variable called SERLSB.                       **/
/*********************************************************************/

void WRSER17()
{
  Sclk =  OFF;              /* Clock low for start condition    */
  Potcs = OFF;              /* take pot chip select low         */
  NOP();                    /* Not really sure how long here!!  */

  Sdata = 0;                /* Shift    out - stack select      */
  Sclk = ON;                /* pulse clock high                 */
  NOP();                    /* delay                            */
  Sclk = OFF;               /* take clock low                   */

  for (i=8;i;i--)
    {                       /* send out the 8 msb's -POT 1      */
      Sdata = SERMSB.7;     /* Shift'em out - msb..lsb          */
      Sclk = ON;            /* pulse clock high                 */
      SERMSB  <<= 1;        /* Shift next msb over              */
      Sclk = OFF;           /* take clock low                   */
      NOP();
    }
  for (i=8;i;i--)
    {                       /* send out the 8 lsb's -POT 0      */
      Sdata = SERLSB.7;     /* Shift'em out - msb..lsb          */
      Sclk = ON;            /* pulse clock high                 */
      SERLSB  <<= 1;        /* Shift next msb over              */

      Sclk = OFF;           /* take clock low                   */
      NOP();
    }
  NOP();                  /* write serial register data into  */
  Potcs = ON ;            /* ... the DAC output register      */
  NOP();
  Sclk = OFF;             /* leave clock in low  position     */
}
```

EXAMPLE 2

```
/*********************************************************************/
/**                                                                 **/
/**  WRSER10 - Write 10 bits of serial data out Sdata/Sclk          **/
/**                                                                 **/
/**  This routine sends out 10 bits of data to the ad840X digital   **/
/**  POT.   Data is shifted out on the pin name Sdata and uses a    **/
/**  pin named Sclk to clock data on the rising edge.               **/
/**  The MSbyte of data is stored in SERMSB bits 1..0 and the       **/
/**  LSByte is stored in SERLSB.  Data is sent out MSBit first      **/
/**  (D9...D0).  This data must be in SERMSB and SERLSB before      **/
/**  the routine is called.                                         **/
/**                                                                 **/
/** Requires the main source file to have a char variable called    **/
/** SERMSB and a char variable called SERLSB.                       **/
/*********************************************************************/

void WRSER10()
{
  Sclk =  OFF;              /* Clock low for start condition    */
  Potcs = OFF;              /* take pot chip select low         */
  NOP();                    /* Delay                            */
  for (i=2;i;i--)
    {                       /* send out the 2 msb's-POT address */
      Sdata = SERMSB.1;     /* Shift'em out - msb..lsb          */
      Sclk = ON;            /* pulse clock high                 */
      SERMSB  <<= 1;        /* Shift next msb over              */
      Sclk = OFF;           /* take clock low                   */
      NOP();
    }
  for (i=8;i;i--)
    {                       /* send out the 8 lsb's             */
      Sdata = SERLSB.7;     /* Shift'em out - msb..lsb          */
      Sclk = ON;            /* pulse clock high                 */
      SERLSB  <<= 1;        /* Shift next msb over              */
      Sclk = OFF;           /* take clock low                   */
      NOP();
    }
  NOP();                  /* write serial register data into  */
  Potcs = ON ;            /* ... the DAC output register      */
  NOP();
  Sclk = OFF;             /* leave clock in low  position     */
}
```