up_down_counter.hcc

```
//  up_down_counter.hcc

/*  Design a counter that counts up if one pushbutton
 *  is pressed, counts down if the other button is
 *  pressed, and doesn't change if neither button is
 *  pressed.  Display the value of the counter as a
 *  decimal number between zero and 99 in two seven
 *  segment displays.
 */

#ifdef USE_SIM
#define PAL_TARGET_CLOCK_RATE 100000
#endif

#ifdef USE_RC200E
#define PAL_TARGET_CLOCK_RATE 2000000
#endif

#include <pal_master.hch>
#include <stdlib.hch>

unsigned  1 s1=0, s0=0, prev_s1=0, prev_s0=0;
unsigned  7 current_count     = 0;
unsigned  4 msd               = 0;
unsigned  4 lsd               = 0;

//  main()
/*  ----------------------------------------------------
 *
 *      Implement an up/down counter with switch inputs.
 *
 *        Read switches.
 *        Count up or down or neither or both.
 *          Limit range to 0-99
 *        Update display
 */
  void
```

```
  main( void )
  {
    PalVersionRequire( 1, 2 );
    PalSevenSegRequire( 2 );

    PalSevenSegEnable( PalSevenSegCT(0) );
    PalSevenSegEnable( PalSevenSegCT(1) );

#ifdef USE_SIM
    PalSwitchRequire( 10 );
#define right_sw 9
#define left_sw  8
#else
    PalSwitchRequire( 2 );
#define right_sw 0
#define left_sw  1
#endif

    while (1)
    {

      //  Read switches
      par
      {
        prev_s0 = s0;
        prev_s1 = s1;
        PalSwitchRead( PalSwitchCT(right_sw), &s0 );
        PalSwitchRead( PalSwitchCT(left_sw), &s1 );
      }

      //  Count up, down, or not at all.
      if ( prev_s0 & ~s0 )
      {
        if ( current_count < 99 ) current_count++;
      }

      if ( prev_s1 & ~s1 )
      {
        if ( current_count > 0 ) current_count--;
      }

      //  Update Display
      par
```

```
        {
          msd = (current_count / 10) <- 4;
          lsd = (current_count % 10) <- 4;
        }
        par
        {
          PalSevenSegWriteDigit( PalSevenSegCT(0), msd, 0 );
          PalSevenSegWriteDigit( PalSevenSegCT(1), lsd, 0 );
        }
      }
    }
```
