```
#include <vxWorks.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <logLib.h>
#include <math.h>

#include "dbAccess.h"

#include "pscStruct.h"

/* #define DEBUG */
#ifdef DEBUG
extern void pscRampStructPrint(PSC_RAMP *);

BOOL        PSCRamperNotice = FALSE;
int                 PSCRamperMaxMsgs = 3;
int                 PSCRamperMsgs = 0;
#endif /* DEBUG */

void
PSCRamper(PSC_RAMP *pStruct)
{
        BOOL        increasing;
        double      step, RT, requested, prev_req, prev_out;
        double      filter_out, diff;

        if (pStruct->alive) {

/* Read in previous output and previous request */
                    prev_out = pStruct->filter_output;
                    prev_req = pStruct->limited_setpt;

/* If new values entered... */
                    if (pStruct->changed) {
                            pStruct->changed = FALSE;

/*          ... copy new val's into struct. */
                            bcopy((char *)pStruct->pnewvals, (char *)pStruct->pcurvals,
                                                NUM_NEWVALS*sizeof(double));
                     /*
                    * Flag send_zeroes is set if power supply has just been switched
                    *       on.  Set requested current and DAC output to zero.
                    */
                            if (pStruct->send_zeroes) {
#ifdef DEBUG
                                    printf("PSCRamper: ON/OFF transition.\n");
                                    PSCRamperMsgs = 0;
#endif
                                    prev_out = 0.0;
                                    (void)dbPutField(pStruct->ppsc_pv, DBR_DOUBLE, &prev_out, 1);
                                    pStruct->filter_output = 0.0;
                                    prev_req = 0.0;
                                    (void)dbPutField(pStruct->preq_pv, DBR_DOUBLE, &prev_req, 1);
                                    pStruct->limited_setpt = 0.0;
                                    pStruct->send_zeroes = FALSE;
                            }
                    }
/*
#ifdef DEBUG
                    if (PSCRamperMsgs < PSCRamperMaxMsgs) {
                            PSCRamperMsgs++;
                            printf("\n");
                            pscRampStructPrint(pStruct);
```

PSCRamper

```
                                        printf("\n");
                                }
#endif
* /
                                requested = pStruct->limited_setpt;

 /*                Continue only if new setting is different from last output. * /
                                if (requested != prev_out) {
#ifdef DEBUG
                                        PSCRamperNotice = FALSE;
#endif  /* DEBUG * /

 /* If we're within least difference, we're there. * /
                                        diff = requested - prev_out;
                                        if (fabs(diff) < pStruct->smallest_change) {
                                                filter_out = requested;
#ifdef DEBUG
                                                printf("PSCRamper: setting filter_out = requested.\n");
#endif
                                        }

 /*                First-order filter output y2 = A*prev_y2 + B*y1 * /
 /*                        (where y1 is requested setting). * /
                                        else {
                                                filter_out = (pStruct->filt_coef_a * prev_out) +
                                                                                        (pStruct->filt_coef_b * requested);

                /*                        See whether filtered setting would exceed max ramp rate... * /
                                                step = filter_out - prev_out;
                                                RT = pStruct->max_step_size;
                                                if (fabs(step) > RT) {

                /* ... if so, adjust increment or decrement to maximum step size. * /
                                                                        if (step > RT) step = RT;
                                                                        else if (-step > RT) step = -RT;
                                                                        filter_out = prev_out + step;
                                                }
                /*                        Check filtered setting against limits. * /
                                                if (filter_out > pStruct->output_max) {
                                                        filter_out = pStruct->output_max;
                                                }
                                                if (filter_out < pStruct->output_min) {
                                                        filter_out = pStruct->output_min;
                                                }
                /* Make sure we don't overshoot or undershoot the requested setting. * /
                                                if (diff > 0) increasing = TRUE;
                                                else increasing = FALSE;

                                                /* Here !increasing is same as decreasing because we
                                                *               wouldn't be here if diff were zero.
                                                * /
                                                if (increasing && (filter_out > requested)) {
                                                        filter_out = requested;
#ifdef DEBUG
                                                        printf("PSCRamper: preventing overshoot.\n");
#endif
                                                }
                                                else if (!increasing && (filter_out < requested)) {
                                                        filter_out = requested;
#ifdef DEBUG
                                                        printf("PSCRamper: preventing undershoot.\n");
#endif
                                                }
                                        }
```

```
/*          Send filtered setting to the hardware. */

                        /* Write new filter_output to the structure for next instance. */
                        pStruct->filter_output = filter_out;
#ifdef DEBUG
                        pStruct->prev_flt_out = prev_out;
                        pStruct->prev_lim_setpt = prev_req;
#endif
                        /* don't report errors because this routine is called so often */
                        (void)dbPutField(pStruct->ppsc_pv, DBR_DOUBLE, &filter_out, 1);
                }
#ifdef DEBUG
                else {
                        if (!PSCRamperNotice) {
                                printf("PSCRamper: requested same as prev_out.\n");
                                PSCRamperNotice = TRUE;
                        }
                }
#endif
        }
        return;
}
```