Using the Animatics Smartmotor:

The Animatics Smartmotors have a built-in controller that is programmed with a Basic-like language somewhat similar to the SELWIN language used by the IAI controllers. The Smartmotors have only seven I/O lines, however. These I/O lines are 5 volt logic, so care must be taken when interfacing the Smartmotor to the PLC or to sensors. Like SELWIN, the SMI language requires your program to explicitly loop from the end to the beginning. Here are equivalent programs:

```
Selwin                                  SMI
SVON
ACC     10                              A=10            ` acceleration in odd units
VEL     100                             V=100           ` velocity in odd units
TAG     1                               C1              ` destination for GOTO or GOSUB

MOVP    1                               P=1000          ` absolute position in encoder counts
                                        G               ` go
                                        TWAIT           ` wait for trajectory to finish

MVPI    2                               D=500           ` relative position in encoder counts
                                        G               ` start motion
                                        TWAIT           ` wait for trajectory to finish

BTON    302                             UA=1            ` turn on signal A

WTON    011                             WHILE UBI==0    ` wait for signal B to go high
                                        LOOP

IN      012     014                     d=UDI
                                        e=UEI*2
                                        f=UFI*4
                                        g=d+e
                                        g=g+f           ' can't use two operators in one expression
IFEQ    99      1                       IF g==1
...
EDIF                                    ENDIF
IFEQ    99      2                       IF g==2
...                                     ...
EDIF                                    ENDIF
IFEQ    99      4                       IF g==4
...                                     ...
EDIF                                    ENDIF

GOTO    1                               GOTO1
END                                     END
```

The scaling factors for acceleration, velocity, and position are:

```
A  = rev/s^2  * 8
V  = rev/s  * 32212
P  = rev * 2000                 ` absolute position
D  = rev * 2000                 ` relative distance move
```

Be careful of absolute moves intermixed with relative moves. If the motor is at absolute position 1000, and you do a relative move 2000, then an absolute move to 2000, you will move backwards a half revolution (1000 counts). This could get very confusing. Rather than relative moves, it might be better to use addition to compute new absolute positions, i.e., P = @P+20000 to move 10 revolutions. Of course, you can always find out your current position at the end of a relative move using the @P variable.

The Smartmotor has a built-in index pulse once per revolution which can be used for homing purposes.

This fragment of program would implement a homing sequence:

```
V=100                   ` some slow velocity
D=2000                  ` one revolution is guaranteed sufficient
G                       ` start motion
WHILE Bt                ` while our trajectory is in motion
        IF Bi           ` index pulse seen?
                S       `stop
                O=0     ` set origin to zero (or some other constant)
        ENDIF
LOOP
```

Example Program smartmotor3.src:

```
'Program smartmotor3.src 2008-06-05  Ralph Stirling
‘I/O:
‘       port A: output, READY signal to PLC
‘       port B: input, TRIGGER from PLC
‘       port C: input, HOME from sensor
‘       port D: input, POS1 from PLC
‘       port E: input, POS2 from PLC
‘       port F: input, POS3 from PLC
‘F E D
‘0 0 0  home
‘0 0 1  abs. move #1
‘0 1 0  rel. move
‘0 1 1  abs. move #2
‘1 1 1  constant torque move, testing for broken wire

UAO   ‘ READY signal to PLC
UBI   ‘ TRIGGER signal from PLC
UCI   ‘ HOME signal from sensor
UDI   ‘ POS1 signal from PLC
UEI   ‘ POS2 signal from PLC
```

```
UFI     ‘ POS3 signal from PLC

ZS      ‘ clear status word, including limit bits

WHILE 1         ‘ loop forever
        UA=1  ‘ let PLC know we’re ready
        WHILE UBI==0 LOOP       ‘ wait for trigger from PLC

        d=UDI                   ‘xx1 or xx0
        e=UEI*2         ‘x1x or x0x
        f=UFI*4         ‘1xx or 0xx
        g=d+e           ‘all bits together
        g=g+f
        ‘ homing move?
        IF g==0
                UA=0            ‘ tell PLC we are moving

                V=32212/4       ‘ 1/4 rev/sec
                A=80            ‘ 10 rev/sec^2

                D=2000                  ‘ 1 rev positive move
                G
                WHILE Bt
                        IF UCI==0       ‘ got home signal?
                                O=0     ‘ set origin
                                S       ‘ stop motion
                                BREAK           ‘ get out of the loop
                        ENDIF
                LOOP

        ‘ position #1 move?
        ELSEIF g==1

                UA=0            ‘ not READY now

                V=1*32212       ‘ 1 rev/sec (60 RPM)
                A=80            ‘ 10 rev/sec^2

                P=500           ‘ 90 degrees
                G
                TWAIT                   ‘ wait for done

        ‘ relative move?
        ELSEIF g==2
                UA=0
                V=1*32212       ‘ 1 rev/sec
                A=80

                D=10*2000       ‘ 10 turns
                G
```

```
            TWAIT

        ‘ constant torque move?
        ELSEIF g==7
            UA=0
            AMPS=500    ‘ 50% max. current as safety
            MT          ‘ torque mode
            T=200       ‘ 20% torque, 200 max vel
            a=@P        ‘ current position
            b=a+20000   ‘ 10 turns
            G
            WHILE Bt
                IF @P>b             ‘ finished all turns
                    X       ‘ slow down and stop
                    BREAK
                ELSEIF @V>190       ‘ almost max velocity
                    S       ‘ stop now, as wire broken
                    BREAK
                ENDIF
            LOOP
            MP      ‘ go back to position mode

        ENDIF

        WHILE UBI==1 LOOP       ‘ be sure PLC saw our UA=0
        UA=1                    ‘ back to READY
LOOP
END
```