

## #5: Cut-to-Length Operation Software Example

When developing a “flying cut-off” control, it is critical that the designer be certain of :

- The control’s ability to initiate a tracking move at a prescribed material length (i.e. to meet or exceed the desired length plus or minus a tolerance requirement).
- The control’s ability to match and maintain velocities of the tracking unit with the material during the cutting cycle (real-time capability).
- In the event of a catastrophic failure, control’s ability to provide appropriate fault recovery, protecting operators, product, tools, and machines.

- The velocity of the cutting edge: that it is within the operating guidelines of the cutter so as to prevent premature wear, or damage.

The bottom line here is that the control must be programmed to handle system real-time reaction needs. Depending upon the system’s velocity and acceleration requirements, the mechanics will be the final controlling limitation of the system.

The following is a sample of an assembly code routine taken from an existing registration/cut-to-length program. The master/slave operation is set up using an 8051 processor registering to line speeds of 400 RPM and tolerance to within plus or minus 0.015 inch. The system resolution is 0.001 inch.

```
;Register on measured lengths indicated
REG_FLOW:      JNB      TR0,RF1          ;Tracking unit must be stopped
                                           ;to begin a new operation
                                           ;Tracking unit not ready
                                           ;Load the req. length only
                SETB     LneSpdToFast
                JMP      LD_PREST

RF1:MOV        DPTR,#CMD24                ;Counter Command Port
                MOV      A,#Rst1         ;Reset Counter & Compare Flgs
                @DPTR,A

                MOV      LstCNT1,#0     ;Zero the LastCount Holder
                MOV      LstCNT2,#0     ;(24 bits)
                MOV      LstCNT3,#0

                MOV      TLO,#000H      ;Reload the 1.024 msec Timer
                MOV      TH0,#0FCH

                SETB     LODNXT          ;Prepare to calc main velocity
                SETB     TR0            ;Timer ON

RF2:JNB        LODNXT,SRT_MVE            ;In ready - Start Tracking

                MOV      DPTR,#IOPort    ;Computer I/O Port
                MOVX     A,@DPTR         ;Get the Port Data
                JB       ACC.0,RF2A      ;Switch to Run-Cycle ?

RF2_EXT:       JMP      REG_EXT         ;End Routine

RF2A:          JB       FLTBIT,RF2_EXT   ;Catastrophic Failure ?

                MOV      DPTR,#RecPort   ;Communications ?
                MOVX     A,@DPTR         ;Test the Serial Port
                JNB     ACC.0,RF2        ;Any Serial Data Rec'd ?

                CALL     GET_SERIAL
                JMP      RF2

;Start the Registration cycle
STR_MVE::     MOV      DPTR,#CMD28      ; Motion Command port

SC1:MOVX      A, @DPTR                  ; Get the data
                JB       ACC.0,SC1       ; LM628 BUSY ?

                MOV      A,#001H        ;Stars the Registration Motion
                MOVX     @DPTR,A        ;Send the Command

                MOV      TLO,#000H      ;Reload the 1.024 msec Timer
                MOV      TH0,#0FCH

                SETB     TR0            ;Start the Velocity Update
```

# TECH TIPS

```

LD_PREST:      MOV     DPTR,#PRESET           ;Counter Data Port
               MOV     A,Low_Length      ;Set the Length Breakpoint
               MOVX    @DPTR,A
               MOV     A,Mid_Length
               MOVX    @DPTR,A
               MOV     A,Hi_Length
               MOVX    @DPTR,A

               MOV     DPTR, #CMD24       ;Counter Command
               MOV     A, #Rst2           ;Reset the Compare Flags
               MOVX    @DPTR, A

;At regulated Length or Clr of HmSw ?
RF3:MOV        DPTR,#CMD24               ;Counter Command Port
               MOVX    A,@DPTR           ;Get the Counter Status
               JB      ACC.2, RF4         ;At the Requested Length ?

               MOV     DPTR,#IOPort      ;Computer I/O Port
               MOVX    A,@DPTR           ;Get the Port Data
               JB      ACC.0,RF3A        ;Switch to Run Cycle ?
               JNB     ACC.1,RF5        ;ReadyToCut (OFF Hm Sw) ?

RF3_EXT:      JMP     REG_EXT             ;End Routine

RF3A:         JB      FLTBIT,RF3_EXT     ;Catastrophic Failure ?

               MOV     DPTR,#RecPort     ;Communications ?
               MOVX    A,@DPTR           ;Test the serial Port
               JNB     ACC.0,RF3         ;Any Serial Data Rec'd ?
               CALL   GET_SERIAL
               JMP     RF3

;At the Length to Cut but Axis didn't clear Home Sw
RF4:          SETB   CutterNotClear     ;Motor/Sw Fauly ?
               JMP     LD_PREST          ;Not OK to start new move

;Off Home Sw .... Start the Cutting Operation
RF5:          CALL   CutPart

;Monitor the Cutting operation .... Done ?
RF6:          MOV     DPTR,#CMD24       ;Counter Command Port
               MOVX    A,@DPTR           ;Get the Counter Status
               JB      ACC.2,RF4         ;At the Requested Length ?

               MOV     DPTR,#IOPort      ;Computer I/O Port
               MOVX    A,@DPTR           ;Get the Port Data
               JB      ACC.0,RF6A        ;Switch to Run-Cycle ?
               JB      ACC.2,RF7        ;Cutting Complete ?

RF6_EXT:      JB      REG_EXT             ;End Routine

RF6A:         JB      FLTBIT, RFG_EXT    ;Catastrophic Failure ?

               MOV     DPTR,#RecPort     ;Communications ?
               MOVX    A,@DPTR           ;Test the Serial Port
               JNB     ACC.0,RF6         ;Any Serial Data Rec'd ?
               CALL   GET_SERIAL
               JMP     RF6

;Cut Complete .... Send the Cutter Home
RF7:          SETB   RdyToHme           ;On next interrupt ... Hm cutter

;Cutter at Home ?
RF8:          MOV     DPTR,#CMD24       ;Counter Command Port
               MOVX    A,@DPTR           ;Get the Counter Status
               JB      ACC.2,RF4         ;At the Requested Length ?

               MOV     DPTR,#IOPort      ;Computer I/O Port
               MOVX    A,@DPTR           ;Get the Port Data
               JB      ACC.0,RF8A        ;Switch to Run-Cycle ?
               JB      ACC.1,RF9        ;Cutter at Home ?

RF8_EXT:      JMP     REG_EXT             ;End Routine

RF8A:         JB      FLTBIT,RF8_EXT     ;Catastrophic Failure ?

               MOV     DPTR,#RecPort     ;Communications ?
               MOVX    A,@DPTR           ;Test the Serial Port

```

## #5: Cut-to-Length Operation Software Example

```

JNB ACC.0,RF8 ;Any Serial Data Rec'd ?
CALL GET_SERIAL
JMP RF8

;Wait for next Point to Cut
RF9:MOV DPTR,#CMD24 ;Counter Command Port
MOVX A,@DPTR ;Get the Counter Status
JB ACC.2,RF9B ;At the Requested Length ?

MOV DPTR,#IOPort ;Computer I/O Port
MOVX A,@DPTR ;Get the Port Data
JB ACC.0,RF9A ;Switch to Run-Cycle ?

RF9_EXT: JMP REG_EXT ;End Routine

RF9A: JB FLTBIT,RF9_EXT ;Catastrophic Failure ?

MOV DPTR,#RecPort ;Communications ?
MOVX A,@DPTR ;Test the Serial Port
JNB ACC.0,RF9 ;Any Serial Data Rec'd ?
CALL GET_SERIAL
JMP RF9

RF9B: JMP REG_FLOW ;Do the next cycle

;Exit the Regiartation Routine
REG_EXT: Return to calling routine

;TMR0 Interrupt Jump .... Setup for 1.024 msec (4 LM628 Updates)
REG_VECT: MOV TLO,#OOOH ;Reload the 1.024 msec Timer
MOV TH0,#OFCH

PUSH PSW
PUSH ACC ;Save the required variables
PUSH DPL
PUSH DPH

MOV CNT1,LstCNT1 ;Put the Last VelCnt into the
MOV CNT2,LstCNT2 ;new count registers
MOV CNT3,LstCNT3 ;(24 bits)
MOV DPTR,#CMD24 ;Counter Command Port
MOV A,#GetCnt ;Capture the Current Count
MOVX @DPTR,A

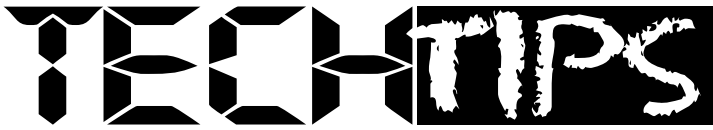
MOV PSW,#0 ;Get Count moved in 1.024 msec
MOV DPTR,#Count ;Counter Data Port
MOVX A,@DPTR ;Get the Lo_Count
MOV LstCNT1,A
SUBB A,CNT1
MOV CNT1,A
MOVX A,@DPTR ;Get the Mid_Count
MOV LstCNT2,A
SUBB A,CNT2
MOV CNT2,A
MOVX A,@DPTR ;Get the Hi_Count
MOV LstCNT3,A
SUBB A,CNT3
MOV CNT3,A

;Slave Velocity = Master Velocity * Velocity Ratio
;
; RTO3.RTO2 RTO1 RATIO = #.##
; X CNT1 X COUNT DIFFERENCE = #
;
; VEL4 VEL3.VEL2 VEL1 CNTS/SAMPLETIME = ##.##
;

MOV A,RATO1
MOV B,CNT1
MUL AB
MOV VEL1,A
MOV VEL2,B

MOV A,RATO2
MOV B,CNT1
MUL AB
ADD A,VEL2
MOV VEL2,A

```



```

MOV      VEL3,B

MOV      A,RATO2
MOV      B,CNT1
MUL      AB
ADD      A,VEL3
MOV      VEL3,A
MOV      VEL4,B

MOV      DPTR,#CMD28                ;Motion Command Port
MOVX     A,@DPTR                    ;Get the Status
JB       ACC.5,MVERR                ;Excessive Folloeing Error ?
JNB      ACC.2,LD_VEL               ;Move Complete ?
JB       RdyToHme,GOHME             ;Done ?
JMP      LoadNxt                    ;Yes !

GOHME:   JMP HmeCycle                ;Start the unit Home

MVERR:   SETB  FolErr                ;Set the Reason for the stop
MOV      DPTR,#IOPort              ;TURN THE MOTOR OFF
MOV      A,OUTPORT
CLR      MotorON
MOV      OUTPORT,A
MOVX     @DPTR,A

MVE1:    MOV      DPTR,#CMD28        ;Motion Command Port
MOVX     A,@DPTR
JB       ACC.0,MVE1                ;LM628 Busy ?
MOV      A,#01FH                    ;Set the Stop Command
MOVX     @DPTR,A                  ;Send the Command

MVE2:    MOVX     A,@DPTR
JB       ACC.0,MVE2                ;LM628 Busy ?

MOV      DPTR,#DAT28                ;Motion Data Port
MOV      A,#STOP                    ;ABRUPT STOP Data Command
MOVX     @DPTR,A                  ;Send the Command
MOV      A,#0
MOVX     @DPTR,A

SETB     FLTBIT                      ;Set the Fault indicator
POP      DPH                        ;Recover the saved variables
POP      DPL
POP      ACC
POP      PSW

RETI                                     ;Return from the Interrupt

;Load the new Velocity if in a Cycle
LD_VEL:  MOV      A,#01FH            ;LM628 Trajectory Command
MOVX     @DPTR,A                  ;Send the Command
MOV      DPTR,#DAT28                ;Motion Data Port
MOV      A,#0
MOVX     @DPTR,A
MOV      A,#008H                    ;Set absolute velocity
MOVX     @DPTR,A

LDV1:    MOV      DPTR,#CMD28        ;Motion Command Port
MOVX     A,@DPTR
JB       ACC.0,LDV1                ;LM628 Busy ?

MOV      DPTR,#DAT28                ;Motion Data Port
MOV      A,VEL4                      ;Load the Velocity
MOVX     @DPTR,A                  ;Send Byte 4
MOV      A,VEL3                      ;Send Byte 3
MOVX     @DPTR,A

LDV2:    MOV      DPTR,#CMD28        ;Motion Command Port
MOVX     A,@DPTR
JB       ACC.0,LDV2                ;LM628 Busy ?

MOV      DPTR,#DAT28                ;Motion Data Port
MOV      A,VEL2                      ;Send Byte 2
MOVX     @DPTR,A
MOV      A,VEL1                      ;Send Byte 1
MOVX     @DPTR,A

MOV      DPTR,#CMD28                ;Motion Command Port

```

## #5: Cut-to-Length Operation Software Example

```

LVD3:      MOVX   A,@DPTR
           JB     ACC.0,LDV3                ;LM628 Busy ?

           MOV   A,#001H                    ;Initiate the new velocity
           MOVX  @DPTR,A                    ;Send the Command

           POP   DPH                         ;Recover the saved variables
           POP   DPL
           POP   ACC
           POP   PSW
           RETI                               ;Return from the Interrupt

;When Commanded ... Load the Upcoming Velocity
LoadNxt:   JB     LODNXT,LDCMD
           JMP    RVEXT

LDCMD:     MOV   A,#01FH                    ;LM628 Trajectory Command
           MOVX  @DPTR,A                    ;Send the Command

           MOV   DPTR,#DAT28                ;Motion Data Port
           MOV   A,#0
           MOVX  @DPTR,A
           MOV   A,#008H                    ;Set absolute Velocity
           MOVX  @DPTR,A

LDC1:      MOV   DPTR,#CMD28                ;Motion Command Port
           MOVX  A,@DPTR
           JB     ACC.0,LDC1                ;LM628 Busy ?

           MOV   DPTR,#DAT28                ;Motion Data Port
           MOV   A,VEL4                     ;Load the Velocity
           MOVX  @DPTR,A                    ;Send Byte 4
           MOV   A,VEL3
           MOVX  @DPTR,A                    ;Send Byte 3

LDC2:      MOV   DPTR,#CMD28                ;Motion Command Port
           MOVX  A,@DPTR
           JB     ACC.0,LDC2                ;LM628 Busy ?

           MOV   DPTR,#DAT28                ;Motion Data Port
           MOV   A,VEL2                     ;Load the Velocity
           MOVX  @DPTR,A                    ;Send Byte 2
           MOV   A,VEL1
           MOVX  @DPTR,A                    ;Send Byte 1

RVEXT:     POP   DPH
           POP   DPL
           POP   ACC
           POP   PSW

           CLR   LODNXT                     ;Clear the prep load flag
           CLR   TR0                        ;Only once through
           RETI                               ;Return from the Interrupt

RtnToHme:  MOV   A,#01FH                    ;LM628 Trajectory Command
           MOVX  @DPTR,A                    ;Send the Command

           MOV   DPTR,#DAT28                ;Motion Data Port
           MOV   A,#0
           MOVX  @DPTR,A
           MOV   A,#00CH                    ;Set Abs Velocity & Pos
           MOVX  @DPTR,A

RTH1:      MOV   DPTR,#CMD28                ;Motion Command Por
           MOVX  A,@DPTR
           JB     ACC.0,RTH1                ;LM628 Busy ~

           MOV   DPTR,#DAT28                ;Motion Data Port
           MOV   A,VEL4                     ;Load the Velocity
           MOVX  @DPTR,A                    ;Send byte
           MOV   A,VEL3
           MOVX  @DPTR,A                    ;Send Byte

RTH2:      MOV   DPTR,#CMD28                ;Motion Command Poz
           MOVX  A,@DPTR
           JB     ACC.0,RTH2                ;LM628 Busy

```

```

MOV     DPTR, #DAT28                ;Motion Data Port
MOV     A, VEL2
MOVX    @DPTR, A                    ;Send Byte 2
MOV     A, VEL1
MOVX    @DPTR, A                    ;Send Byte 1

RTH3:  MOV     DPTR, #CMD28          ;Motion Command Port
        MOVX    A, @DPTR
        JB     ACC.0, RTH3          ;LM628 Busy ?

        MOV     DPTR, #DAT28        ;Motion Data Port
        MOV     A, Hme4
        MOVX    @DPTR, A            ;Load the Home Position
        MOV     A, Hme3
        MOVX    @DPTR, A            ;Send byte 4

RTH4:  MOV     DPTR, #CMD28          ;Motion Command Port
        MOVX    A, @DPTR
        JB     ACC.0, RTH4          ;LM628 Busy ?

        MOV     DPTR, #DAT28        ;Motion Data Port
        MOV     A, Hme2
        MOVX    @DPTR, A            ;Send Byte 2
        MOV     A, Hme1
        MOVX    @DPTR, A            ;Send Byte 1

RTH5:  MOV     DPTR, #CMD28          ;Motion Command Port
        MOVX    A, @DPTR
        JB     ACC.0, RTH5          ;LM628 Busy ?

        MOV     A, #001H            ;Initiate the new velocity
        MOVX    @DPTR, A            ;Send the Command

        SETB    HomeCycle           ;Unit on the way Home
        POP     DPH                  ;Recover the saved variables
        POP     DPL
        POP     ACC
        POP     PSW

        RETI                         ;Return from the Interrupt
    
```



### About the Author:

In his more than two decades in the industry, **Chuck Raskin, PE, CMCS**, has contributed to many industry publications, including *Motion, Motion Control, & PCIM*, and is currently working on the fourth edition of the *Designing with Motion Handbook*. Chuck is currently the manager of technical services for Technology 80 and a board member of the American Institute of Motion Engineers (AIME).



### Liked what you've read? Now you can buy the book!

Now in its fifth edition, Chuck Raskin's *Designing With Motion Handbook* will be a valuable resource for you regardless of your level of expertise. Considered to be the single best text in the motion control industry, it is used as a core text in the American Institute of Motion Engineers' Certification Program.

With more than three decades of industry experience, Chuck's 450+ page book contains valuable insights and examples that will help you balance all of the mechanical, electrical and software considerations of virtually any motion control application.

**As a special offer, you can receive a FREE copy of The PC Handbook, a 96-page pocket-size reference that's a valuable resource when designing PC-based applications (a \$9.95 value). To order the handbook for only \$50, call 800-545-2980 and ask for the "TECH TIPS HANDBOOK OFFER."**

Please include \$5 for shipping & handling. CA and MN residents also include appropriate sales tax.