Ball Turning with Manual Numerical Control, version 2.1.0

By R. G. Sparber



I suspect that just about everyone is familiar with Computer Numerical Control (CNC). It has changed the look of most products we own. Before CNC, products were rather boxy with lots of straight lines and planes. CNC frees designers from this simplicity and lets us have complex, curved surfaces in our toasters, cars, and everything in between.

Many hobbyists have taken the plunge and run CNC in their shop. Others, like me, have resisted the pull because we enjoy doing things manually. Yet, there is a middle ground: Manual Numerical Control (MNC). This method of machining can create amazing shapes but the work can also be mind numbing and demanding at the same time.

[1] You are free to copy and distribute this document but not change it.

One way to combat the demands of MNC, is to have a program supervise your machining. It can tell you what to do next and also hold your place if you want to take a break. This article presents such a program that runs on a TI-83 Plus calculator and lets you turn a ball, crown, or full radius using only a parting tool for a cutter.

The surfaces created with this procedure can be made to look very nice. But do not expect the process to turn out anything precision. You would lose your mind trying to take that many cuts.

I expect there are two kinds of people that will be interested in this program. The first group would like to understand how CNC works. Using the program will give them firsthand knowledge of how a series of cutting coordinates creates a surface. The second group has no interest in CNC but wants to cuts balls, crowns, or full radii on their lathe without any special tooling. Hopefully this program will satisfy both groups.

If you own a TI-83 Plus or compatible programmable calculator and wish a copy of the program file, please contact me directly at rgsparber@aol.com. I will send you the file to download directly into your calculator.

One bit of jargon: DOC means Depth Of Cut.

Background

Over the last few decades, I have played around with MNC a few times. My first exposure to this concept was in Guy Lautard's first Bedside Reader, page 72. It inspired me to write a program that ran on an HP pocket calculator. This calculator has a very limited alpha numeric screen making it easy to get confused.

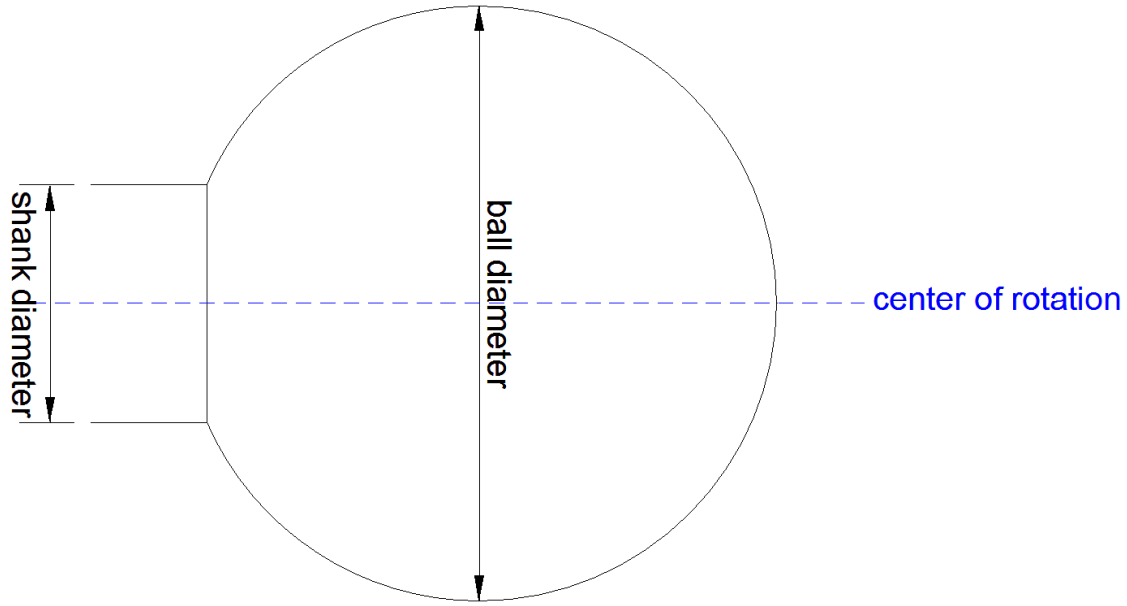
The next program ran in Excel. Generate and print out the spreadsheet and bring into the shop. This was a big improvement and essentially free. Since it is just a piece of paper, I could easily move forward and backward through the data if I got confused. Yet, it was still possible to lose my place which typically then caused a defect in the surface that could not be removed.

Recently I have started to play with a TI-83 Plus calculator which has an excellent alphanumeric screen and plenty of memory. So, just for fun, I decided to take a crack at this old chestnut again.

The program feeds me data similar to the spreadsheet but with a few important improvements. I define the size of the ball, crown, or full radius and shank in my shop and don't need to run to my PC and print something out. It also keeps track of where I am at all times. And lastly, I can move forward and back in the data if I get confused. Just like with the spreadsheet, I can walk away at any time and the data is saved.

When using the spreadsheet, I tended to make one error on every surface. So far, I have made no mistakes using the calculator.

Overview



This software tool will let you turn a ball that is attached to a shank. The user specifies the workpiece's initial diameter, finished ball diameter, finished shank diameter and the program guides you in the machining operations.

This is done in three passes: outline turning, a roughing pass, and a finish pass.

The outline turning reduces the workpiece's diameter to equal the roughing diameter of the ball and the roughing diameter of the shank.

The roughing pass cuts steps almost as wide as your parting tool and about .01" away from the finished surface[2]. The work goes fast.

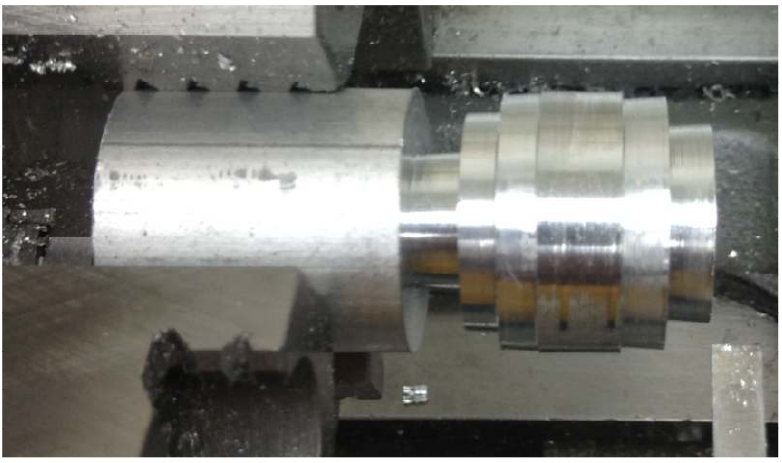
The finish pass consists of many light cuts that land close to or on the finished surface. As Guy Lautard explains, a file is used to get to the final shape. The result is a nice looking ball.

[2] This value is user specified.

Shop Experience

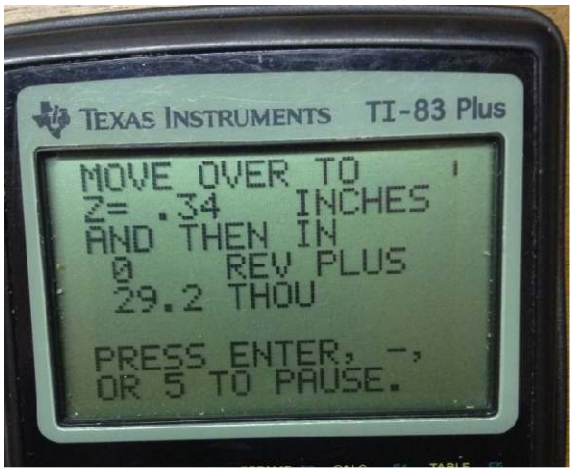
All steps are explained on the TI-83 Plus's screen.

The outline of the ball and shank are roughed out.

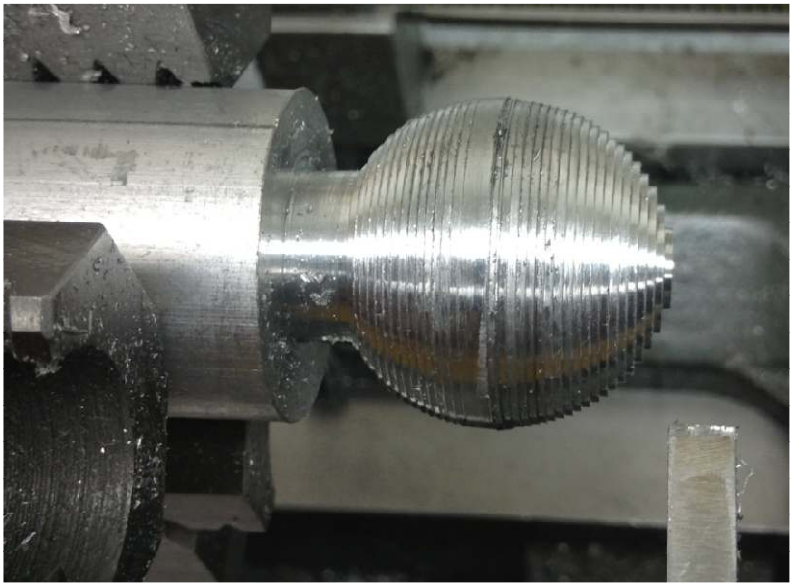


Using a parting tool, the roughing pass is performed. The step size is set to be of equal width and just a little narrower than the parting tool's width. This minimizes the time it takes to rough out the ball and associated shank. Here you see four plunge cuts with the parting tool to roughly define the ball.

If you have a form tool or ball turning attachment, this may be as far as you need to go.

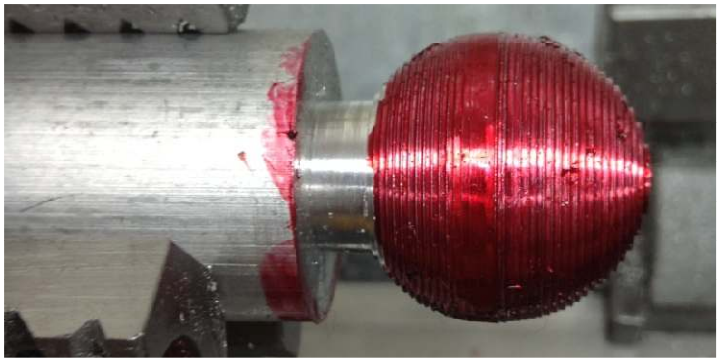


Here is a typical screen during the shaping of the ball. If you presses ENTER, the next longitudinal (Z axis) and cross feed (X axis) values appear. If you press "-", the previous set of numbers are displayed. You can also take a break by pressing "5". After a few minutes, the calculator powers down. Just press the ON button and then the ENTER button and you will be back to the same place in the program.



This finish pass was done with a 0.025" wide step size. There is a tradeoff between the coarseness of the steps and the ability of the user to retain sanity.

It is very tedious when you are taking a large number of cuts. Fortunately, the program lets you stop at any time, pick up at that point later, move forward, and backward, in the position of the cutter. The experience certainly gives you an appreciation for how CNC works.



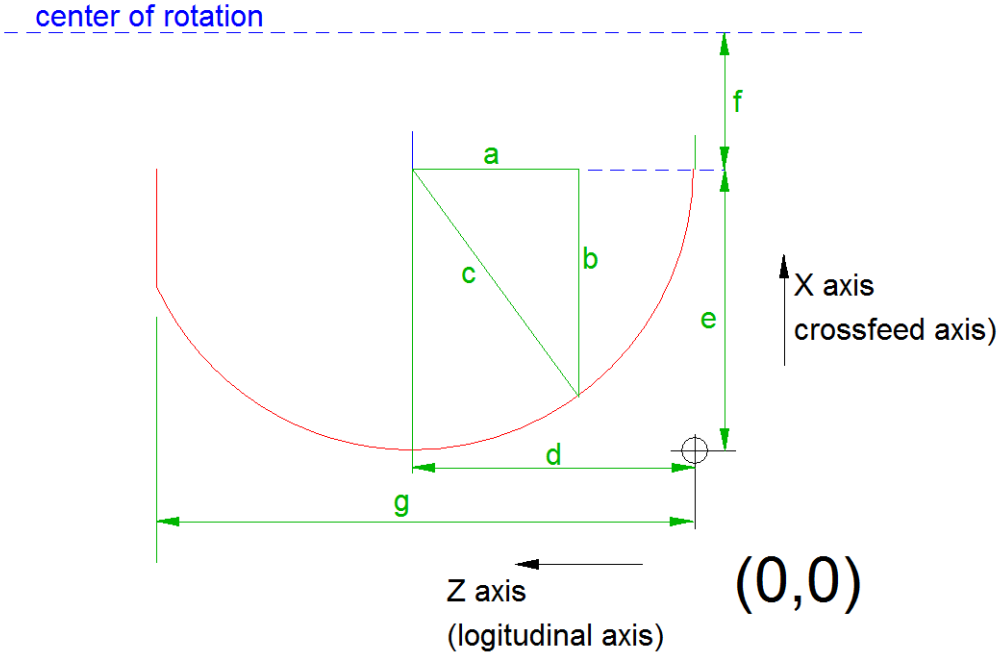
Using the method describe by Guy Lautard, I coated the surface with dye.

A file was then used to remove the steps. As you get close to the final shape, the dye will show as fine lines. When the lines are gone, it is time to switch to emery cloth for the final finishing.

Since I can cut a ball, I can cut the end of a piece of round stock so it has a crown. In this case the stock is 1" in diameter and the crown has a 4" diameter.

Unlike a ball cutting attachment, the program can handle a ball or segment of a ball of any diameter. So if you wanted a crown with a diameter of 40", it is just a matter of entering the number.

Theory

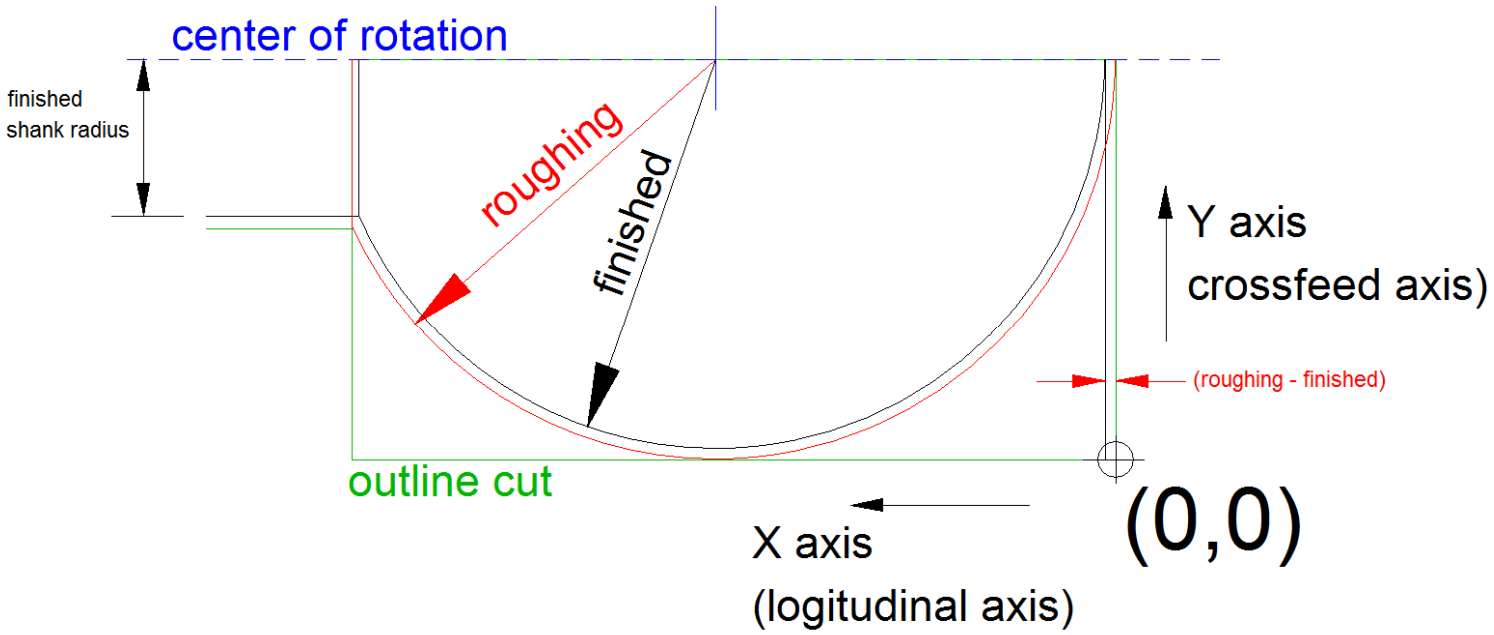


This is a top view of the workpiece showing only from the center of rotation to the radius. The shank begins at a distance "g" from the origin.

Notice that I have slipped in a bit of flexibility here in the form of distance "f". If turning a ball, the distance from the center of the ball to the center of rotation is zero. But if "f" is not zero, I am turning a disk with the given radius at its edges. A "full radius" has this shape which was cut with the program in control.

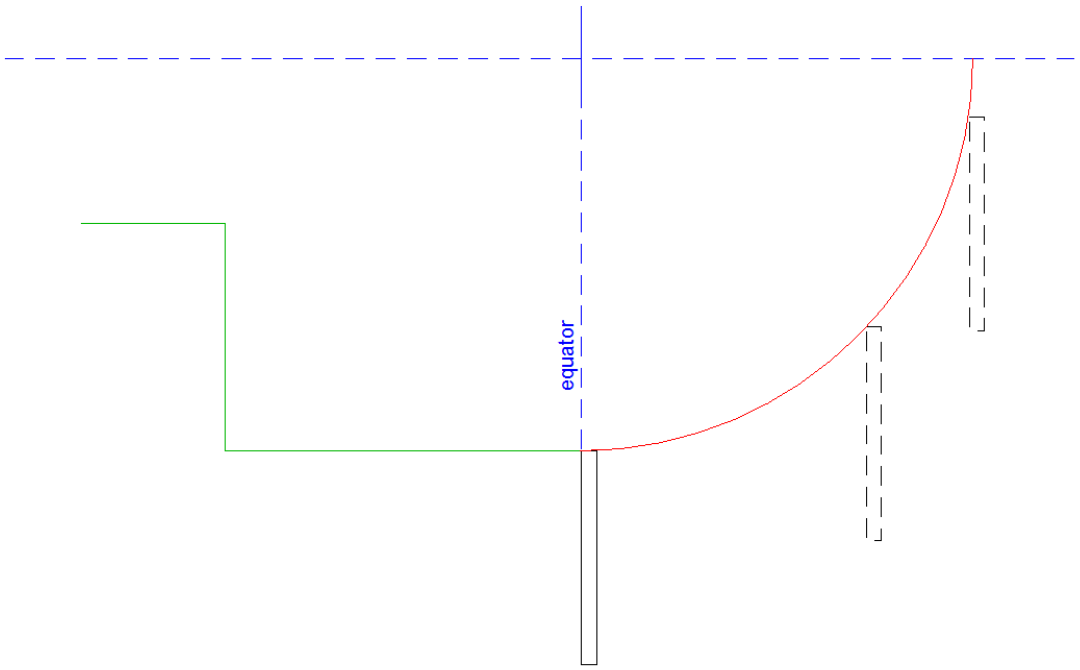
This program is rather large in order to make it as easy as possible for the user. Buried deep in the code within subroutine BTU12 is a single equation. That is all it takes to define a ball, crown, or full radius. This equation translates a longitudinal value to a crossfeed value.

The rest of this theory will only discuss forming a ball.

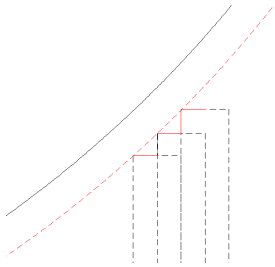


The first step in making the ball is to cut the outline, shown in green. This is done by first facing the right end using a left hand cutter. Then the roughing radius is turned. And finally, the roughing radius of the shank is turned using both a left and right hand cutter or a parting tool depending on how much room you have.

The right end of the workpiece is my Z=0 reference plane. The outside diameter is my X=0 surface.



Next, a series of roughing cuts are made using a parting tool to form a stair step that follows the roughing surface.



The front left corner of the tool cuts the surface of the ball from the equator to right end the of the ball. By moving in this direction, the operator is always feeding in so avoids cross feed backlash.

After the right half of the ball is formed, the front right corner of the parting tool is used to cut from the equator to the start of the shank. This, again, avoids crossfeed backlash.

All of this cutter movement is handled by the program and the user is simply given longitudinal and crossfeed settings.

The finish pass follows the same logic as the roughing pass. The only difference is how much you longitudinally advance the parting tool for each cut. This distance is mostly determined by how much of this repetitive cutting you can stand before going nuts. One instant of distraction and the cutter can go too far. The error will then stick out like a sore thumb.

Frequent breaks are a good idea. Just walk away from the lathe and calculator. The calculator will automatically power down after a few minutes. Press the ON button and then ENTER to return to where you were before your break.

Thanks to Corey for telling me the proper term for "full radius".

I welcome your comments and questions.

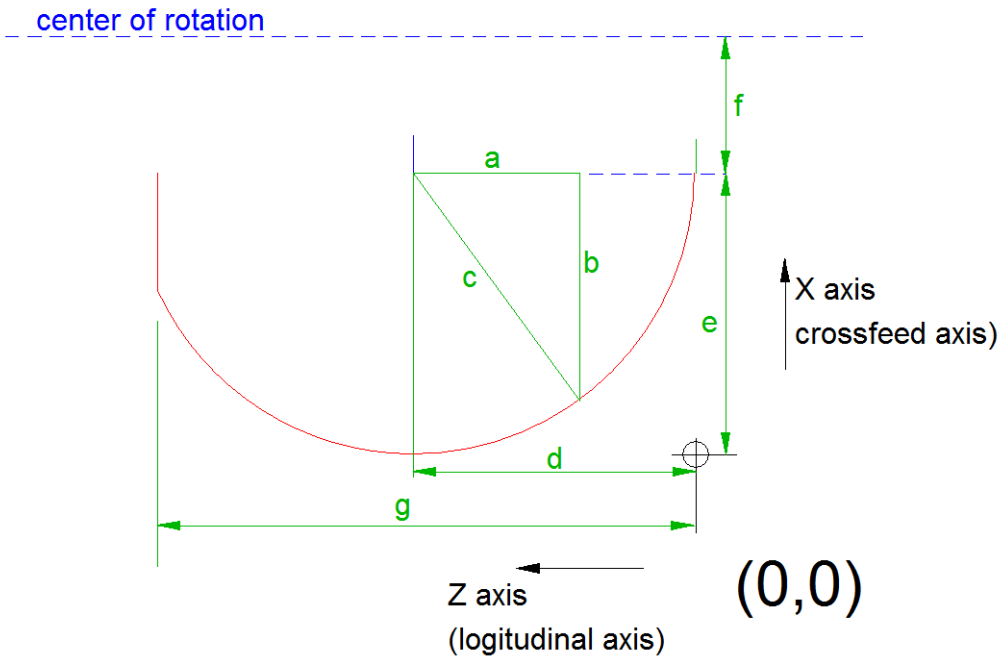
Rick Sparber
Rgsparber@aol.com
Rick.Sparber.org

Appendix: The Program

Subroutine Listing

Variables Used

A - finished ball diameter in inches
B - finished shank diameter in inches
C - parting tool width in thou
D - finish Depth Of Cut (DOC) in thou
E - finish step size in thou
F - roughing ball diameter in inches
G - roughing shank diameter in inches
H - OAL of cylinder containing ball in inches
I - number of Z steps (used to right and then left of equator and for rough and finish cuts)
J - width of each cut step (used to right and then left of equator and for rough and finish cuts) in inches
K - cut counter steps (used to right and then left of equator and for rough and finish cuts)
L - size of crossfeed dial; if >0, then it is a radius dial; if <0 it is a diameter dial in thou
M - current X axis internal radius dial position: integer part is full turns, fractional part is thou
N - distance from origin to shank in inches
O - number of full crossfeed dial revolutions for the given X value
P - dial value (may be for a radius or diameter dial)
Q - temp
R - profile control (0 = ball, 1= crown, 2 = full radius)
S - used only by BTU6 and BTU8
T
U
V - if crossfeed uses just a dial, V not equal to 1; if crossfeed has a DRO, V = 1
W - temp
X - current X axis value in thou
Z - current Z axis value in inches



[A] - a 1 x 9 matrix used to pass the surface defining equation its parameters

1. a - not used at the present time
2. b - not used at the present time
3. c
4. d
5. e
6. f - not used at the present time
7. g- not used at the present time
8. h - temp
9. i - temp

BTR - Ball Turning Tool

```
{1,9} → dim([A])          define shape vector and zero all elements
LBL D
MENU ("BALL TURN V1.2", "RUN", A, "SET UP", B, "INITILIZE", E,
"EXIT",C)
LBL A
Prgm BTU
GO TO D
LBL B
Prgm BTS
GO TO D
LBL E
PrgmBTR1
GO TO D
LBL C
ClrHome
PrgmA       program A is main calling menu
END
```

BTR1 initialization of variables

-1 -> V	dial (if V=1 it would be DRO)
187.5 → C	parting tool width
10 → D	finish DOC
50 → E	finish step size
100 → L	dial has 100 thou/rev and since >0, is radius
0→R	profile set to ball
RETURN	
END	

BTS - Setup

Store parting tool width, roughing - finish cut distance, finish pass step size, crossfeed dial size, radius vs diameter crossfeed dial

```
LBL A
MENU("SET UP", "PARTING TOOL", B, "FINISH DOC",C, "FINISH STEP",
D, "CROSSFEED", G, "PROFILE", H, "EXIT",E)

LBL B
Prgm BTS1          C stores parting tool width
GO TO A

LBL C
Prgm BTS2          D stores finish DOC
GO TO A

LBL D
PrgmBTS3           E stores finish step size
GO TO A

LBL G
PrgmBTS7           DRO/DIAL, radius/diameter, dial size
GO TO A

LBL H
Prgm BTS6          select profile R - (0 = ball, 1=  crown, 2 = full radius, 3 =
                   concave)
GO TO A

LBL E
RETURN
END
```

BTS1 - prompt for parting tool width and store in C

```
"PARTING TOOL ", "→Str1
2→T
C→W
PrgmWP1
W→C
RETURN
END
```

BTS2 - prompt for finish DOC and store in D

```
"FINISH DOC" ->Str1
2->T
D->W
PrgmWP1
W->D
RETURN
END
```

BTS3 - prompt for finish step size and store in E

```
"FINISH STEP SIZE"->Str1
2->T
E->W
PrgmWP1
W->E
RETURN
END
```

BTS4 - prompt for crossfeed dial size and store in L

```
LBL C
        "DIAL SIZE "-> Str1
        2->T                tell WP1 units are THOU

        IF L>0              if L positive, this is a radius dial so use as is
        THEN
        L->W
        ELSE                otherwise, this is a diameter dial so double and invert
                            sign before showing user
        -L*2->W
        END
        PrgmWP1                 display value and prompt for change
        Abs(W) → W          force L to be positive so it does not interfere with
                            radius/dia. flag
        IF L>0              if L positive, this is a radius dial so use as is
        THEN
        W -> L
        ELSE                otherwise, this is a diameter dial so cut in half and make
                            negative before storing internally
        -W/2 -> L
        END
        RETURN
        END
```

BTS5 - sign of L flags crossfeed dial as radius or diameter

```
LBL C
        IF L>0
        THEN
        MENU("SETS RADIUS ", "KEEP", A, "MAKE DIAMETER",B)
        ELSE
        MENU("SETS DIAMETER ","KEEP",A,"MAKE RADIUS",B)
LBLB
        -L →L
        GO TO C
LBL A
        RETURN
        END
```

BTS6 - select the profile

R - profile control (0 = ball, 1= crown, 2 = full radius)

```
LBL D
        ClrHome
        OUTPUT(1,1,"SELECTED PROFILE ")
                IF R = 0
                THEN
        OUTPUT(2,1," IS BALL.")
                GO TO E
                END

                IF R = 1
                THEN
        OUTPUT(2,1," IS CROWN.")
                GO TO E
                END

                IF R = 2
                THEN
        OUTPUT (2,1," IS FULL RADIUS.")
                GO TO E
                END

        OUTPUT(2,1," IS UNKNOWN.")              otherwise, unknown
LBL E
        PrgmBTU0          prompt to press ENTER and pause
```

```
        ClrHome
        MENU("PROFILES","BALL", A, "CROWN", B, "FULL RADIUS",C,
"EXIT", F)
LBL A
        0→R
        GO TO D

LBL B
        1→R
        GO TO D

LBL C
        2→R
        GO TO D

LBL F
        RETURN
        END
```

BTS8 - DRO or DIAL with V as flag

```
LBL A

IF V=1              if V=1 it means DRO
THEN
Menu("HAVE DRO","KEEP",B,"MAKE IT DIAL",C)
ELSE
Menu("HAVE DIAL", "KEEP",B, "MAKE IT DRO",D)
END

LBL B
RETURN

LBL C
-1 -> V             any V not equal to 1 means DIAL
GO TO A             review change

LBL D
1 -> V              set V to 1 to flag DRO
GO TO A             review change
END
```

BTS7 - CROSSFEED parameters

```
PrgmBTS8          set flag for DRO versus DIAL
PrgmBTS5          set flag for radius versus diameter

IF V =1           if using a DRO, return
THEN
RETURN
END

PrgmBTS4          if not DRO, prompt for dial size
RETURN
END
```

BTU - Main

Prompt for finished ball diameter or radius, finished shank diameter, and initial workpiece OD

Calculate outline dimensions.

Tell operator to face end of bar and turn to roughing cut diameters using left and right hand cutters.

Tell operator to install parting tool with width specified in setup. Set origin. Guide operator over roughing pass.

Guide operator over finish pass.

Design interface so operator can stop at any time and restart later without confusion.

Prgm BTU1	get finished ball diameter
Prgm BTU2	get finished shank diameter
Prgm BTU3	calculate outline cuts
Prgm BTU4	tell operator to face end and turn ball rough diameter plus shank rough diameter
PrgmBTU5	tell operator to verify parting tool width and then zero parting Tool
PrgmBTU6	make equator to origin roughing cuts
PrgmBTU7	make equator to ball/shank interface roughing cuts
PrgmBTU8	make equator to origin finish cuts
PrgmBTU9	make equator to shank finish cuts
PrgmBTU10	using a left hand cutter, cut finish pass on shank
RETURN	
END	

WP1 - prompt with titles

```
IF T=1                          T indicates inches, thou, or blank after value
THEN
"INCHES."→Str2
END

IF T=2
THEN
"THOU."→Str2
END

IF T=3
THEN
""→Str2
END

LBL C
ClrHome
1 + iPart((16-length(Str1))/2)→Q      center title held in string 1
OUTPUT(1,Q,Str1)
DISP ""
DISP "PRESENT VALUE IS"
OUTPUT(3,1,W)                         value is passed to subroutine in W
OUTPUT(3,7,Str2)
DISP ""                               blank DISP commands move line pointer
                                      Along because OUTPUT does not do it
DISP ""
DISP "1)KEEP  2)CHANGE"
WHILE 1
GETKEY → Q                            scan for 1 or 2 being pressed. Ignore all else
WHILE Q=0
GETKEY→Q
END

IF Q=92
THEN
RETURN
END
```

```
IF Q=93
THEN
GO TO B
END

END
LBL B
DISP ""
DISP "NEW VALUE="
INPUT W
GO TO C              after getting new value, restart dialog
END
```

BTU0 - prompt and pause utility

```
OUTPUT(8,1,"PUSH  ENTER")
PAUSE
RETURN
END
```

BTU1 - finished ball diameter prompt

```
ClrHome
IF R=0
THEN
DISP "  BALL PROFILE"
END

IF R=1
THEN
DISP " CROWN PROFILE"
END

IF R=2                  if cutting full radius, change prompt text
THEN
DISP "  FULL RADIUS"
DISP "    PROFILE"
OUTPUT(4,1," RADIUS")
OUTPUT(5,1"IN INCHES")
DISP ""
DISP ""
DISP ""
DISP ""                 four DISP "" put INPUT on line 5
INPUT A
2*A→A                   convert to diameter in inches
RETURN
ELSE
OUTPUT (3,1,"FINISHED BALL")
OUTPUT (4,1," DIAMETER ")
OUTPUT(5,1,"IN INCHES  ")
DISP ""
DISP ""
DISP ""
DISP ""
DISP ""
INPUT A
RETURN
END
```

BTU2 - finished shank diameter prompt

```
ClrHome
OUTPUT (1,1,"FINISHED SHANK")
OUTPUT (2,1," DIAMETER ")
OUTPUT(3,1,"IN INCHES  ")
DISP ""
DISP ""
DISP ""
DISP ""
INPUT B
RETURN
END
```

BTU3 - calculate outline cuts

```
A + (.002*D) → F                ball roughing/outline diameter in
                                inches
B + (.002*D) → G                shank roughing/outline
                                Diameter in inches

IF R=1                          this is a crown cut
THEN
GO TO A
END

IF R=2                          this is a full radius
THEN
GO TO B
END

 (F/2) + SQRT((F/2)^2 - (G/2)^2) → H     OAL of cylinder containing ball in
                                         inches
RETURN

LBL A
(F/2) - SQRT((F/2)^2 - (G/2)^2) → H      OAL of cylinder containing crown in
                                         inches
RETURN

LBL B
F → H                   OAL of cylinder is roughing diameter of ball
RETURN
END
```

BTU4 - tell operator to face end and turn ball rough outline plus turn shank to roughing diameter

```
ClrHome
OUTPUT(1,1,"FACE END")
OUTPUT(2,1, "OF WORKPIECE.")
PrgmBTU0

IF R=1                    we are cutting a crown so skip outline of ball
THEN
GO TO A
END                       for ball and full radius, use ball equations

ClrHome
OUTPUT(1,1,"TURN DIAMETER")
OUTPUT(2,1, "OF")

IF R=2                    if cutting a full radius
THEN                      then use alternate text
OUTPUT(2, 4"DISK ")
2*D → W
ROUND(W,0)→W
OUTPUT(3,1,"PLUS")
OUTPUT(3,6, W)
OUTPUT(3,9,"THOU.")
GO TO B
END

ROUND(F,3) → W
OUTPUT(2,4, W)
OUTPUT(3,1, "INCHES")
LBL B
OUTPUT(4,1,"FOR OAL OF")
ROUND(H,3)→W
OUTPUT(4,12,W)
OUTPUT(5,1,"INCHES")
PrgmBTU0
LBL A
```

```
IF B = 0              if the shank finish diameter is zero, skip outline of it
THEN
RETURN
END

ClrHome
OUTPUT(1,1,"TURN DIAMETER")
OUTPUT(2,1, "OF")
OUTPUT(2,4, G)
OUTJPUT(2,11, "INCHES")
OUTPUT(3,1,"FOR LENGTH OF")
OUTPUT(4,1,"SHANK.")
PrgmBTU0
RETURN
END
```

BTU5 - zero parting tool

```
ClrHome
OUTPUT(1,1," LEFT SIDE OF")
OUTPUT(1,14,C)
OUTPUT(2,1,"THOU PARTING "
OUTPUT(3,1" TOOL ON RIGHT ")
OUTPUT(4,1, "END OF WORKPIECE ")
OUTPUT(5,1," AND ZERO Z AXIS ")
PrgmBTU0                    prompt and pause
ClrHome
OUTPUT(1,1,"PLACE FRONT OF")
OUTPUT(2,1," PARTING TOOL")
OUTPUT(3,1, "ON OD OF")
OUTPUT(4,1,"WORKPIECE AND")
OUTPUT(5,1"ZERO X AXIS")
PrgmBTU0                    prompt and pause
RETURN
END
```

no backlash problem with Z because of DRO

BTU6- make equator to origin roughing cuts if this is a ball or crown roughing cuts

```
        F/2 → [A](1,3)              initialize form vector for roughing cut
        F/2 → [A](1,4)
        F/2 → [A](1,5)

        PrgmBTU11                   tell user to back cutter away from max OD

        IF R=1              if this is a crown cut, Z distance is H
        THEN
        H → S
        ELSE
        F/2 →S              if this is a ball or full radius cut, distance is roughing
                            radius of ball
        END

        iPart(S/(C*0.001)) + 1→ I   calc # of Z steps to go from equator to right
                                    end of ball or for crown on the roughing cut
                                    by rounding up
        S /I → J            calc width of each Z step in inches
0→K                         start of loop similar to FOR(K,0,I) except I can modify K
LBL B

        IF K>I
        THEN
        RETURN
        END

LBL A

            S - (K*J) → Z           in inches
            PrgmBTU12       take Z and generate X in thou
            PrgmBTU61       output next (X,Z) pair saying to move Z first and
                            then X; take into consideration radius/diameter dial
            PrgmBTU71       go to next step or go back one if possible (W flag)

        IF W>0              flag = 1 means return to the top of the loop with
                            decremented or unchanged count value
```

```
THEN
GO TO A          give last location again
END

K+1 → K          for normal loop cycle, increment K
GO TO B          end of loop rough cutting from equator to right end
END
```

BTU61 - translates from internal radius distance to crossfeed dial setting and display to user

X is total distance from origin. It must be broken down into the number of full turns of the cross slide dial plus number on dial. X is radius feed. If the dial is marked for diameter, I must double X before displaying it to user. Internally, radius is used. If using a DRO, V=1 and I don't show full revolutions of dial.

```
LBL A
        IF R=1              if cutting a crown, set X=0 at roughing OD of shank and
                            not to roughing diameter of ball
        THEN
        X - (F-G)/.002 → Q      reduce X by the difference between ball roughing
                                radius and shank roughing radius
        ELSE
        X → Q               if not cutting a crown, store X in Q
        END

        IF V=1              if V=1, we have a DRO so skip modulo dial calculations
        THEN
        Q -> P              P is dial value and also DRO value
        GO TO B
        END

        Q/ L  → W   normalize X to the radius dial size

        Abs(iPart(W)) → O       integer part is the number of full revolutions of the
radius
                            dial

        fPart(W) * L→P      then multiply by radius dial size L to get radius dial
                            setting
```

```
LBL B
IF L< or = 0          if this is a diameter dial,
THEN
2*P -> P              then double dial position
END

ROUND(Z,3) → W        reusing W for rounded Z (xx.xxx)
ClrHome
OUTPUT(1,1,"MOVE OVER TO")
OUTPUT(2,1," Z= ")
OUTPUT(2,4,W)
OUTPUT(2,10,"INCHES")
OUTPUT(3,1,"AND THEN IN")


IF V=1                if we have a DRO, just show value in inches
THEN
P/1000 -> W
ROUND(W,4) -> W
OUTPUT(3,13,"TO")
OUTPUT(4,1,W)
GO TO C
END
OUTPUT(4,2,O)
OUTPUT(4,7,"REV PLUS")
ROUND(P,1) → W        reusing W for rounded dial (xxx.x)
OUTPUT(5,2, W)
OUTPUT(5,7,"THOU")
LBL C
OUTPUT(7,1,"PRESS ENTER, -")
OUTPUT(8,1,"OR 5 TO PAUSE.")

WHILE 1
GetKEY → W
While W=0
GetKEY→ W
END
ClrHome
DISP W
PAUSE
```

```
IF W=105                "ENTER" detected so clear reverse flag and return
THEN
0 → W                   W=0 means continue on to next cut
RETURN
`  END

IF W=85                 "-" detected so set reverse step flag and return
THEN
1 →W                    W=1 means back up on step
RETURN
`  END

IF W=83                 "5" detected so PAUSE. In this state, the calculator
                        will automatically power down in a few minutes
                        and when powered up again, return to this point in
                        the program.
THEN
ClrHome
OUTPUT(1,1," PAUSE MODE:")
OUTPUT(3,2,"LEAVE POWER ON")
OUTPUT(4.1," AND LET CALC")
OUTPUT(5,1," POWER DOWN.")
OUTPUT(7,1,"PRESS -ENTER- TO")
OUTPUT(8,1,"RETURN TO PASS.")
PAUSE
GO TO A
END

END
```

BTU7 make equator to shank interface roughing cuts

The Z equation starts at the equator and adds steps to get cutter over to shank. It also includes the parting tool's width so we are cutting on the front right corner. X equation must subtract the parting tool's width from Z before using it to determine in-feed.

```
IF B=A OR R=1                    if shank finish diameter equals ball finish
                                 diameter or if crown,

THEN                             then skip this phase
RETURN
END

F/2 → [A](1,3)                   initialize form vector for roughing cut
F/2 → [A](1,4)
F/2 → [A](1,5)

PrgmBTU11                        tell user to back cutter away from max OD
iPart((H - (F/2))/(0.001*C)) +1→ I   calc # of Z steps to go from equator to left
                                 end of ball on the roughing cut by rounding
                                 down and adding 1
((H - (F/2))/ I → J  calc width, in inches, of each Z step
      0→K            start of loop similar to FOR(K,0,I) except I can modify K
LBL B


      IF K>I
      THEN
      RETURN
      END

LBL A
      K*J → W     in inches, Z axis distance from center of ball to cut point
      (F/2) + W + (0.001*C)  → Z          in inches
      PrgmBTU12         take Z and generate X in thou
      PrgmBTU61         output next (X,Z) pair saying to move Z first and then X;
```

```
                    take into consideration radius/diameter dial. W=1 flags
                    request to back up to previous location
PrgmBTU71           go to next step or go back one if possible


IF W>0              flag = 1 means return to the top of the loop with
                    decremented or unchanged count value
THEN
GO TO A             give last location again
END

K+1 → K             for normal loop cycle, increment K
GO TO B             end of loop rough cutting from equator to right end
RETURN
END
```

BTU71 back up on step if possible

```
ClrHome
DISP W
DISP K
PAUSE

IF W>0 AND K> OR = 1            if back up flag set and loop count,
                                  K>0, decrement K
THEN
K-1→K           decrement K so user will be at previous step
1 → W           flag that we must return to top of FOR loop
RETURN
END

IF W>0 AND K<1          if back up flag set but K=0, can't back up any more
THEN
ClrHome
OUTPUT(3,1,"ERROR, AT START  "
PrgmBTU0                prompt and pause
1 → W           flag that we must return to top of FOR loop without
                decrementing K
RETURN
END

-1 → W          else, flag to continue in FOR loop
RETURN          and go to first line of FOR loop
END
```

BTU8 make equator to origin finish cuts

```
A/2 → [A](1,3)                initialize form vector for finish cut
F/2 → [A](1,4)
F/2 → [A](1,5)
PrgmBTU11                     tell user to back cutter away from max OD

      IF R=1                  if this is a crown cut, distance is H-D
      THEN
      H- (0.001*D)  → S             inches
      ELSE
      A/2 →S                  inches; if this is a ball cut, distance is finish radius of ball
      END

iPart(S/(.0001*E)) → I        calc # of Z steps to go from equator to right end

0→K           start of loop similar to FOR(K,0,I) except I can modify K
LBL B

      IF K>I
      THEN
      RETURN
      END
LBL A
       0.001*K*E → W in inches
      S- W + (.001*D) → Z             in inches
      PrgmBTU12               take Z and generate X in thou
      PrgmBTU61               output next (X,Z) pair saying to move Z first and then X;
                              take into consideration radius/diameter dial
                              end of loop rough cutting from equator to right end
      PrgmBTU71               go to next step or go back one if possible

      IF W>0                  flag = 1 means return to the top of the loop with
                              decremented or unchanged count value
      THEN
      GO TO A                 give last location again
      END

      K+1 → K                 for normal loop cycle, increment K
```

```
GO TO B
END
```

BTU9 make equator to shank finish cuts

The Z equation starts at the equator and adds steps to get cutter over to shank. It also includes the parting tool's width so we are cutting on the front right corner. X equation must subtract the parting tool's width from Z before using it to determine in-feed.

```
IF B=A OR R=1                  if shank finish diameter equals ball finish
                               diameter or if crown,
THEN                           skip equator to shank cuts
RETURN
END

A/2 → [A](1,3)                 initialize form vector for roughing cut
F/2 → [A](1,4)
F/2 → [A](1,5)
PrgmBTU11                      tell user to back cutter away from max OD

IF R=2                         if cutting full radius,
THEN
A/2 → W                        then take radius of finished ball
 ELSE
 SQRT((A/2)^2 - (B/2)^2) → W                    distance in inches from
                                                ball center to shank
 END

iPart(W/(0.001*E))  → I        calc # of Z steps to go from equator to left
                               end of ball on the finish cut by rounding down
0→K          start of loop similar to FOR(K,0,I) except I can modify K
LBL B

      IF K>I
      THEN
      RETURN
      END

LBL A
      (A/2) + (.001*D) + (.001*C) + 0.001*K*E  → Z          in inches; includes
            offset from roughing origin to finish origin and width of parting tool
```

```
PrgmBTU12       take Z and generate X in thou
PrgmBTU61       output next (X,Z) pair saying to move Z first and then X;
                take into consideration radius/diameter dial
                end of loop finish cutting from equator to right end
PrgmBTU71       go to next step or go back one if possible

IF W>0          flag = 1 means return to the top of the loop with
                decremented or unchanged count value
THEN
GO TO A         give last location again
END

K+1 → K         for normal loop cycle, increment K
GO TO B
RETURN
END
```

BTU10 using a left hand cutter, cut finish pass on shank

```
IF B = 0            if the shank finish diameter is zero, skip finish pass
THEN
RETURN
END

ClrHome
OUTPUT(1,1,"CHANGE TO")
OUTPUT(2,1,"LEFT HAND"
OUTPUT(3,1,"CUTTER")
PrgmBTU0            pause and wait for ENTER
OUTPUT(1,1,"SET X = 0 AT")
OUTPUT(2,1,"RIGHT END")
OUTPUT(3,1,"OF SHANK")
OUTPUT(4,1,"AND CUT FINAL")
OUTPUT(5,1,"OD OF SHANK.")
PrgmBTU0            pause and wait for ENTER
RETURN
END
```

BTU11 tell user to back parting tool away from max OD

```
ClrHome
OUTPUT(1,1,"BACK PARTING")
OUTPUT(2,1,"TOOL AWAY")
OUTPUT(3,1,"FROM MAX OD")
PrgmBTU0
RETURN
END
```

BTU12 - translate any Z in inches less than the diameter of the ball and return X in thou; uses vector [A] to set size and location from origin

Calling program has already offset Z to compensate for parting tool width when at values of Z greater than the equator. Here I remove that compensation in order to get X.

```
IF Z < or = [A](1,4)                    if Z is to right of equator
THEN
Z → [A](1,8)                    then use it directly
ELSE                            if Z is to left of equator

Z - 0.001*C → [A](1,8)          then subtract width of parting tool
END
```

1000*([A](1,5) - SQRT([A](1,3)^2 - ([A](1,4) - [A](1,8))^2)) → X this single line defines a ball for both roughing and finish passes

```
RETURN
END
```

Test Cases

Ball - radius dial

Parting tool 100 thou
Finish DOC 10 thou
Finish step size 50 thou
Crossfeed: have dial, sets radius, dial size 100 thou
Profile: ball

Finish diameter .5
Finish shank diameter .2

Turn .52 diameter for .496 inches
Turn shank .22" diameter

Z, inches	X (revolutions/thou)	
.26	0/0	equator to right end roughing
.173	0/14.9	
.087	0/66.2	
0	2/60	
.36	0/0	equator to shank end roughing
.439	0/12.1	
.517	0/52.8	
.596	1/50	
.26	0/10	equator to right end finishing
.21	0/15.1	
.16	0/30.9	
.11	0/60	
.06	1/10	
.01	2/60	
.36	0/10	equator to shank end finishing
.41	0/15.1	
.46	0/30.9	
.51	0/60	
.56	1/10	

Ball - diameter dial

Parting tool 100 thou
Finish DOC 10 thou
Finish step size 50 thou
Crossfeed: have dial, sets diameter, dial size 200 thou
Profile: ball

Finish diameter .5
Finish shank diameter .2

Turn .52 diameter for .496 inches
Turn shank .22" diameter

Z, inches	X (revolutions/thou)	
.26	0/0	equator to right end roughing
.173	0/29.7	
.087	0/132.4	
0	2/120	

Abort the rest.

Ball - diameter DRO

Parting tool 100 thou
Finish DOC 10 thou
Finish step size 50 thou
Crossfeed: have DRO, sets diameter
Profile: ball

Finish diameter .5
Finish shank diameter .2

Turn .52 diameter for .496 inches
Turn shank .22" diameter

Z, inches	X (inches)	
.26	0	equator to right end roughing
.173	.0297	
.087	.1324	
0	.52	

Abort the rest.

Crown

Parting tool 100 thou
Finish DOC 10 thou
Finish step size 10 thou
Crossfeed: have dial, sets radius, dial size 100 thou
Profile: crown
Finish diameter 5
Finish shank diameter 1

Turn 1.02 diameter for length of shank

Z, inches	X (revolutions/thou)	
.052	0/0	roughing pass to right end but no cut occurred
0	5/10	finish pass to right
0.052	0/51.7	
.042	1/9.1	
.032	1/76.4	
.022	2/61.7	
.012	4/1.4	

Cut final OD of shank

Full radius

Parting tool 100 thou
Finish DOC 10 thou
Finish step size 50 thou
Crossfeed: have dial, sets radius, dial size 100 thou
Profile: full radius

Radius in inches 0.125
Finish shank 1
TURN DIAMETER OF DISK PLUS 20 THOU FOR OAL OF .27 INCHES

Turn shank to 1.02" for length

BTU6

Z, inches	X(revolutions/thou)
.135	0/0
.068	0/18.1
0	1/35
.235	0/0
.303	18.1
.37	1/35
.135	0/10
.085	0/20.4
.035	0/60
.235	0/10
.285	0/20.4
.335	0/60

TURN SHANK TO FINISH DIA.