

SolidCAM a Mach3

```
; Mach3.gpp
; Postprocesseur pour MACH3
; Pour SOLIDCAM 2006 REV2.1
; Créateur David "DBa" Bartoszek
; Special Porperties : Part Options:
;         1: ONLY_XYZ gives only the koordinates as full_gcode
;         2: TEXT is printing all text it's poisible
;         proc: If blknum > blk_num_max: blk_num will be reset
; 29/10/2006
;-----
;
; 2009 MaxMekker div modifications for my use with Terco CNC45
;
;-----

;
; Mach3
;----
;
@init_post
    global string tool_diameter_f
    global logical first_rapid_move print_gcode
; Non GPPL variables
num_user_procs = 1
line_labels = TRUE ; Jump to N...

; GPPL variables
pre_processor = 'Mach3'
```

```
numeric_def_f = '5.3'  
integer_def_f = '5.0(p)'  
gcode_f      = '2.0(p)'  
mcode_f      = '2.0(p)'  
xpos_f       = '5.3'  
ypos_f       = '5.3'  
zpos_f       = '5.3'  
feed_f       = '4.3(p)'  
tool_diameter_f = '5.3/1'  
blknum_f     = '5.0(p)'  
blknum_gen   = true  
blknum_exist = true  
blknum_letter = 'N'  
end_block_text = ''  
blknum       = 5  
blknum_delta = 5  
blknum_max   = 999995
```

```
; trace "all":1
```

```
; trace "@call_proc":5
```

```
endp
```

```
;-----
```

```
@start_of_file
```

```
{'%'}
```

```
{nl, 'O'program_number, '(', part_name, ')}'
```

```
if rotate_used then
```

```
gcode = 69
call @gen_nb
{'G'gcode}
endif
if mirror_used then
call @gen_nb
{'G50.1 X0 Y0'}
endif
endp
```

```
;-----
```

```
@start_program
```

```
call @gen_nb
{'G21 G40 G54 G80 G91.1 (Initialisation)'}
call @gen_nb
{'(Terco CNC45 and Mach3 MaxMekker.com)'}
call @gen_nb
{'G0 X0 Y0'}

endp
```

```
;-----
```

```
@end_program
```

```
call @gen_nb
{'G0 Z50 M9 M9'}
call @gen_nb
```

```
{'G0 X0 Y0 M5'}
```

```
{nl, 'M30'}
```

```
endp
```

```
;-----
```

```
@end_of_file
```

```
label = first_user_proc
```

```
{nl, '%'}
```

```
endp
```

```
;-----
```

```
@relative_mode
```

```
if only_xyz eq false
```

```
gcode = 91
```

```
call @gen_nb
```

```
{'G'gcode, ' '}
```

```
skipline = FALSE
```

```
endif
```

```
endp
```

```
;-----
```

```
@absolute_mode
```

```
if only_xyz eq false
```

```
gcode = 90
```

```
call @gen_nb
```

```
{'G'gcode, ' '}  
  skipline = FALSE  
endif  
endp
```

```
;-----
```

```
@machine_plane  
if only_xyz eq false  
  if machine_plane eq XY  
    gcode = 17  
  endif  
  if machine_plane eq ZX  
    gcode = 18  
  endif  
; call @gen_nb  
; {'G'gcode}  
endif  
endp
```

```
;-----
```

```
@call_proc  
if only_xyz eq false  
  if active(parm1) then  
    gcode = 65  
    call @gen_nb  
    {'G'gcode, ' P'label}  
    {' A'parm1, [' B'parm2], [' C'parm3]}  
  else
```

```
    call @gen_nb
      {'M98 P'label}
  endif
  if proc_count gt 1 then
    {' L'proc_count}
  endif
endif
```

```
  if text eq true
    {' ('message, ')}
  endif
endif
endp
```

```
;-----
```

```
@proc
  if only_xyz eq false
    {'nl, ':'label}
  endif
endp
```

```
;-----
```

```
@end_proc
  if only_xyz eq false
    call @gen_nb
      {'M99'}
  endif
endp
```

;-----

@loop

if only_xyz eq false

 local integer var_num

 var_num = loop_level + 20

 call @gen_nb

 {'#, var_num, '= 0'}

 call @gen_nb

 {'WHILE ['#', var_num, ' LT ', loop_count, '] DO ', loop_level}

endif

endp

;-----

@end_loop

if only_xyz eq false

 local integer var_num

 var_num = loop_level + 20

 call @gen_nb

 {'#, var_num, '= #', var_num, '+ 1'}

 call @gen_nb

 {'END ', loop_level}

endif

endp

;-----

```
@def_tool
```

```
if only_xyz eq false
```

```
  ; call @gen_nb
```

```
  ; {'(G10 L1 P', (tool_number+1), ' R'tool_offset, ')}
```

```
endif
```

```
endp
```

```
;-----
```

```
@rapid_move
```

```
  gcode = 0
```

```
    call @gen_nb
```

```
    if print_gcode eq true
```

```
      {'G'gcode}
```

```
endif
```

```
  if first_rapid_move eq true
```

```
    call @home_number
```

```
    {' X'xpos, ' Y'ypos}
```

```
    call @gen_nb
```

```
    {'G43 H'tool_number,' Z'zpos}
```

```
    first_rapid_move = false
```

```
  else
```

```
    {' X'xpos], [' Y'ypos], [' Z'zpos]}
```

```
  endif
```

```
endp
```

```
;-----
```



```

@move_4x
    gcode = 0
; if change(gcode) then
    call @gen_nb
    if print_gcode eq true
        {'G'gcode}
; else
;     call @gen_nb
;     {' '}
    endif
; endif
    {' X'xpos], ' Y'ypos , [' Z'zpos], [' A'cpos] }
endp
;-----

```

```

@line
    gcode = 1
; if change(gcode) then
    call @gen_nb
    if print_gcode eq true
        {'G'gcode}
; else
;     call @gen_nb
;     {' '}
    endif
; endif
    {' X'xpos], [' Y'ypos], [' Z'zpos]}
if only_xyz eq false
    {' F'feed}}
endif

```

endp

@line_4x

gcode = 1

; if change(gcode) then

call @gen_nb

if print_gcode eq true

{'G'gcode}

; else

; call @gen_nb

; {' '}

endif

; endif

{[' X'xpos], [' Y'ypos], [' Z'zpos], [' A'cpos]}

if only_xyz eq false

{[' F'feed]}

endif

endp

;-----

@arc

-- arc plane --

if arc_direction eq CCW then

gcode = 3

else ; CW

gcode = 2

```

endif
; if change(gcode) then
    call @gen_nb
    if print_gcode eq true
        {'G'gcode}
; else
;     call @gen_nb
;     {' '}
endif
; endif
    {' X'xpos] [' Y'ypos] [' Z'zpos]}

; if arc_size eq 360 then
    {' I'xcenter_rel:xpos_f, ' J'ycenter_rel:ypos_f}
; else
;     if arc_size >= 180 then
;         radius = -radius
;     endif
;     {' R'radius}
; endif
if only_xyz eq false
    {' F'feed}}
endif
endp

```

```

;-----

```

```

@compensation

```

```

if only_xyz eq false

```

```

    if side eq COMP_LEFT then

```

```
    gcode = 41
endif
if side eq COMP_RIGHT then
    gcode = 42
endif
if side eq COMP_OFF then
    gcode = 40
endif
call @gen_nb
{'G'gcode, ' '}
if side ne COMP_OFF
    {'D'(offset_number), ' '}
endif
skipline = FALSE
endif
endp

;-----
```

```
@delay
if only_xyz eq false
    gcode = 4
    call @gen_nb
    {'G'gcode, ' P'delay_period:integer_def_f}
endif
endp
```

```
;-----
```

```
@change_ref_point
```

```
if only_xyz eq false
  ; Given in absolute mode
  gcode = 10
  call @gen_nb
  {'G91 G'gcode, 'L2 P'home_number ' X'xhome, ' Y'yhome, ' Z'zhome}
  call @gen_nb
  {'G90'}
endif
endp
```

```
;-----
```

```
@home_number
  gcode = 53 + home_number
  if first_rapid_move eq true
    {' G'gcode}
  else
    endif
  endif
endp
```

```
;-----
```

```
@rotate
if only_xyz eq false
  if rotate_cancel then
    gcode = 69
    call @gen_nb
    {'G'gcode}
  else
```

```
gcode = 68
call @gen_nb
{'G'gcode, ' X0 Y0 G91 R'angle}
call @gen_nb
{'G90'}
endif
endif
endp
```

```
;-----
```

```
@fourth_axis
gcode = 0
call @gen_nb
{'G'gcode, ' A'angle}
endp
```

```
;-----
```

```
@change_tool
if only_xyz eq false
local logical save_blknum_gen
if machine_plane eq XY
gcode = 17
endif
if machine_plane eq ZX
gcode = 18
endif
call @gen_nb
{' '
```

```
call @gen_nb
{'G0 Z50 M9'}
call @gen_nb
{'(Tool nr 'tool_number,' - Diametre 'tool_diameter,'mm)'}
call @gen_nb
{' '}
mcode = 6
call @gen_nb
{'M' mcode' T'tool_number' D'tool_number' G43' ' H'tool_number}
call @gen_nb
{' '}
call @start_tool
call @home_number
```

endif

endp

;-----

@message

if only_xyz eq false

; call @gen_nb

; {'(', message, ')'}
endif

endif

endp

;-----

@drill

```
call @rapid_move
```

```
; gcode = 98
```

```
; call @gen_nb
```

```
; {'G'gcode, ' '}
```

```
call @gen_nb
```

```
if drill_type eq G81 then
```

```
    gcode = 81
```

```
endif
```

```
if drill_type eq G82 then
```

```
    gcode = 82
```

```
endif
```

```
if drill_type eq G83 then
```

```
    gcode = 83
```

```
endif
```

```
if drill_type eq G84 then
```

```
    gcode = 84
```

```
endif
```

```
if drill_type eq G85 then
```

```
    gcode = 85
```

```
endif
```

```
if drill_type eq G86 then
```

```
    gcode = 86
```

```
endif
```

```
if drill_type eq G89 then
```

```
    gcode = 89
```

```
endif
```

```
{'G'gcode, ' Z'drill_lower_z, ' R'drill_upper_z}
```

```
if drill_type eq G83 then
```

```
    {' Q'down_step}
```



```

        {' P'verweilzeit:'7.0(*1000P)'}
endif
if drill_type eq G82 then
    {' P'verweilzeit:'7.0(*1000P)'}
endif
if drill_type eq G84 then
    feed = spin * steigung
endif
; {' F'feed ' L0'}
    {' F'feed}

endp

;-----

@drill_point
if not first_drill
    call @gen_nb
    {' ', [' X'xpos], [' Y'ypos], [' Z'zpos]}
endif
endp

;-----

@mirror
if only_xyz eq false
    if mirror_type eq MIRROR_OFF then
        call @gen_nb
        {'G50.1 X0 Y0'}
    else

```

```
call @gen_nb
{'G51.1 '}
if mirror_type eq MIRROR_X then
  {'X1 Y0'}
endif
if mirror_type eq MIRROR_Y then
  {'X0 Y1'}
endif
if mirror_type eq MIRROR_XY then
  {'X1 Y1'}
endif
endif
endif
endp
```

```
;-----
```

```
@end_drill
gcode = 80
call @gen_nb
{'G'gcode}
call @gen_nb
{'G0 Z50 M9'}
endp
```

```
;-----
```

```
@halt_program
if only_xyz eq false
  {' M0'}
```

endif

endp

;-----

@round_comp

; NOP

endp

;-----

@start_of_job

if only_xyz eq false

first_rapid_move = true

; gcode = 43

; call @gen_nb

; {, 'G'gcode, ' H'tool_number, ' Z'znext}

call @gen_nb

{('job_name')}

endif

endp

;-----

@end_of_job

; NOP

endp

```
;-----
```

```
@assign_axis
```

```
    ; TBD
```

```
endp
```

```
; =====
```

```
; USER DEFINED PROCEDURES
```

```
; =====
```

```
@call_simple_proc
```

```
if only_xyz eq false
```

```
    active(message) = FALSE
```

```
    active(parm1) = FALSE
```

```
    active(parm2) = FALSE
```

```
    active(parm3) = FALSE
```

```
    proc_count = 1
```

```
    call @call_proc
```

```
endif
```

```
endp
```

```
;-----
```

```
@start_tool
```

```
if only_xyz eq false
```

```
    if tool_direction eq CW then
```

```
        mcode = 3
```

```
    else ; CCW
```

```
        mcode = 4
```

endif

call @gen_nb

{'S'spin:integer_def_f, ' M'mcode}

call @gen_nb

{'M8'}

call @gen_nb

{' '}

endif

endp

;-----

@stop_tool

if only_xyz eq false

{' M5'}

endif

endp

@gen_nb

if blknum >= blknum_max

blknum = 5

endif

if only_xyz eq false

{nb}

print_gcode = true

else

{nl}

print_gcode = true

```
change(xpos) = true
change(ypos) = true
change(zpos) = true
xpos_f      = '5.3/3(P)'
ypos_f      = '5.3/3(P)'
zpos_f      = '5.3/3(P)'
first_rapid_move = false
endif
endp
```



```
@pre_processor
;Mach3.mac
;Internal parms
machine_type      = MILLING
post_processor    = Mach3

;Machine Initialize
machine_plane     = XY
z_with_xy        = Y
mac_axes         = XYZ
num_axes         = 3
num_simult_axes  = 3
abs_coord        = Y
rotate           = N
mirror           = N
variables        = N
loops           = N

;Program numbers
```

prog_num_min = 1
prog_num_max = 99999999
prog_num_dflt = 1
get_prog_num = Y
proc_num_min = 1
proc_num_max = 99999999
proc_num_dflt = 1
get_proc_num = Y

;Procedures control

full_gcode = N
gen_procs = N
drill_proc = N
turn_proc = N
thread_proc = N
gen_internal_proc = N
turn_common_proc = N
gen_1_line_proc = N
optimize_jobs_loop = N
G_in_2_cols = N

;Home

num_homes = 1
dflt_home = 1
get_job_home = Y
abs_zero_chng = N

;Positioning

dflt_start = 0.0000 200.0000 100.0000, 0.0000 7.8740 3.9370
dflt_end = 0.0000 200.0000 0.0000, 0.0000 7.8740 0.0000

set_z_chng = Y
dflt_tool_chng = 0.0000 0.0000 0.0000, 0.0000 0.0000 0.0000

;Compensation

comp_exist = Y N
comp_arc_arc = Y
comp_arc_line = Y
comp_line_line = Y
next_angle = N
comp_x_start = N
comp_by_arcs = N
chng_tool_table = N
look_forward = 2

;Arc definitions

arc_exist = Y
arc_3d = N
arc_quadrants = N
arc_gt_180 = Y
arc_max_chord = 30.0000, 1.1811
arc_max_angle = 10.0000
arc_max_radius = 2000.0000, 78.7402

;Epsilon values

eps_angle = 0.0020, 0.0001
eps_line = 0.0010, 0.0000
zero_value = 0.0010, 0.0000
min_delt_arc_rad = 0.0100, 0.0004
safety_dist = 2.0000, 0.0787


```
;Feed-Spin
rapid_feed      = 40000.0000, 15748.0312
max_spin        = 42000.0000
max_feed        = 20000.0000, 7874.0156
spin_direction  = CW
```

```
;Timing
time_factor     = 1.0000
block_time      = 0.0000
change_tool_time = 0.0000
```

```
;Part options
options         = ONLY_XYZ LOGICAL
options         = TEXT LOGICAL
```

```
;Job options
```

```
;Drill cycles
drill_type      = G81 Drilling Y
drill_type      = G82 F_Drill Y Verweilzeit
drill_type      = G83 Peck Y Verweilzeit
drill_type      = G84 Tapping Y Steigung
drill_type      = G85 Peck Y
drill_type      = G86 Boring Y
drill_type      = G89 None Y
```

```
;Turning cycles
```

```
;Threading cycles
```

```
;Grooving cycles
```

;Wire Cut cycles

;Turning definitions

turning_cycle = Y N

groove_cycle = Y N

combined_cycles = N

optimize_cycle = N

finish_retreat = N

semi_finish_retreat = N

fanuc_cycle = Y

;Fourth axis

indexial_4th_axis = N

indexial_increment = 0.0000, 0.0000

init_cpos = N

polar_4x = N

cartez_4x = N

set_dir = N

fourth_axis_letter = C

;Wire Cut parameters

lower_guide_level = 0.0000, 0.0000

group_def = N

offset_group_name =

wc_pos_proc = Y

u_max = 100.0000, 3.9370

v_max = 100.0000, 3.9370

xy_abs = Y

uv_abs = N

agie = N

;GCTOOL

int_is_illegal = N

devison_factor = 1000.0000

gc_procs_file_name = NO_FILE

change_tool_at_ref = Y

endp