MOTUSCNC

MotusCNC G-Code Interpreter Manual revision 1.0.37 – February 2018

MotusCNC G-Code Interpreter Manual

Table of Contents

Change History

2017 May 1 : SJH : Initial, extracted from user manual.

2017 Sep. 10 : SJH : Clarified coordinate transforms in G38 probing; documented handling of repeated comments and odd usage of parameters; added warning about using canned cycles with TWP.

2017 Nov. 6 : SJH : Documented M105 commands, documented conditional block execution.

2018 Feb. 9 : SJH : First publish.

Numeric Control (G-Code)

MotusCNC uses an NC interpreter based on the public domain RS274NGC work by NIST. Some proprietary extensions have been added, and the meaning of some functions have been changed to conform with common practice in the industry. The documentation in this section is largely taken from the RS274NGC version 3 manual. Refer to that manual for low-level details.

Files and File Naming Conventions

The following file name extensions are assumed to denote NC files:

.nc

.ngc

.gcode

.rs274

.tap

If a file with a different extension than the above is presented to MotusCNC, then the initial content of the file is scanned to determine whether it is a valid NC file. Informally, if the first 4 non-blank lines appear to be a sequence of single letters separated by numbers, the file is assumed to be an NC file.

The detailed rule is:

Examine lines starting with the first, apply the following tests:

- after removal of spaces, '#', '.','-' and anything at and after '(', and ignoring any blank lines or lines starting with any of 'Oo%', and skipping over an initial / or \, the remainder of the data is single letters separated by 'numbers'

- a 'number' is a decimal number or anything inside matching '[' ']'. For simplicity, the part of the line after the first '[' is replaced with '0'.

If the first 4 (or all, if less than 4) non-blank lines follow this rule, then the file is assumed to be an NC file.

Format of a Line

A permissible line of input NC code consists of the following, in order, with the restriction that there is a maximum (currently 256) to the number of characters allowed on a line.

1. an optional block delete character, which is a slash "/" , or an inverted block delete character '\'. Either of these may be optionally followed by a conditional expression between '[' and ']'.
2. an optional line number or subroutine label (N- or O-word).
3. any number of words, parameter settings, and comments.
4. an end of line marker (carriage return or line feed or both).

Any input not explicitly allowed is illegal and will cause the Interpreter to signal an error.

Spaces and tabs are allowed anywhere on a line of code and do not change the meaning of the line, except inside comments. This makes some strange-looking input legal. For example, the line

```
g0x +0. 12 34y 7
```

is equivalent to

```
g0 x+0.1234 y7
```

Blank lines are allowed in the input. They are to be ignored.

Input is case insensitive, except in comments, i.e., any letter outside a comment may be in upper or lower case without changing the meaning of a line.

Block Delete

Any line may be prefixed with one of the following constructs:

- /
  Execute the line only if the block delete switch is OFF.
- \
  Execute the line only if the block delete switch is ON.
- /[expression]
  Execute the line only if the value of the expression is non-zero.
- \[expression]
  Execute the line only if the value of the expression is zero.

NOTE: testing the expression for zero actually allows for a small rounding error (0.0001).

Use of the conditional expression can simplify and speed up program execution. For example, here are two ways of executing a block of code if the parameter #1 is (nearly) zero[1]:

```
\[#1] G0 X0 Y0  (only do this line if #1 is zero)
```

```
M98 L#1 R123 (goto line O123, skipping next, if #1 is not zero)
G0 X0 Y0
O123
```

The first way is faster and (arguably) easier to read and understand. It also does not require a label number (123 in this case) to be made up. The advantage becomes even clearer when multiple options are required, and each option can be written as a single block of code:

```
/[#1=0] G0 X0 Y0
/[#1=1] M6 T1
/[#1=2] M98 P101
```

Only one (or none) of the above lines will be executed, depending on whether #1 is close to 0, 1 or 2.

NOTE: the square brackets are required. If accidentally omitted, then usually an error will occur, but sometimes the meaning may be changed, introducing a bug in the code:

1 It is assumed that #1 is very nearly an integer. If not, then the two examples are not equivalent.

```
/#1=1 M6 T1
```
vs.
```
/[#1=1] M6 T1
```
The first case is executes as “if the block delete switch is OFF, then set parameter 1 to 1 then do a tool change”. The second is “if the value of parameter 1 is 1, then do a tool change”.

Line Numbers and Subroutine Labels

A line number is the letter N or O followed by an integer (with no sign) greater than or equal to 0. 'N' line numbers may be repeated or used out of order, although normal practice is to avoid such usage. Line numbers may also be skipped, and that is normal practice. A line number is not required to be used, but must be in the proper place if used.

'O' line numbers, which are usually used as subroutine labels, should be unique in the program. The interpreter does not check this, however the program may have an undefined order of execution.

The only command which references an N line number is the M97 command (subroutine call or GOTO with N word target).

The only command which references an O line number is the M98 command (subroutine call or GOTO with O word target).

Word

A word is a letter other than N or O followed by a real value.

Words may begin with any of the letters shown in Table 3. The table includes N and O for completeness, even though, as defined above, line numbers and subroutine labels are not words. Most letters have different meanings in different contexts.

Letter	Meaning
A	Rotary axis, probe parameter
B	Rotary axis, probe parameter
C	Rotary axis, probe parameter
D	Tool radius compensation index
F	Feed rate
G	General function
H	Tool length offset index
I	X offset (for arcs), orientation vector X component, general index
J	Y offset (for arcs), orientation vector Y component, general index
K	Z offset (for arcs), orientation vector Z component, general index
L	Repetition number, buffer index
M	Miscellaneous function
N	Line number (must be first word on line)

Letter	Meaning
O	Subroutine label (must be first word on line)
P	General parameter, dwell time, function code
Q	General parameter
R	Arc radius, general parameter
S	Spindle speed
T	Tool slot/index
X	Linear axis
Y	Linear axis
Z	Linear axis

A real value is some collection of characters that can be processed to come up with a number. A real value may be an explicit number (such as 341 or -0.8807), a parameter value, an expression, or a unary operation value. Definitions of these follow immediately. Processing characters to come up with a number is called “evaluating”. An explicit number evaluates to itself.

Number

The following rules are used for (explicit) numbers. In these rules a digit is a single character between 0 and 9.

- A number consists of (1) an optional plus or minus sign, followed by (2) zero to many digits, followed, possibly, by (3) one decimal point, followed by (4) zero to many digits — provided that there is at least one digit somewhere in the number.
- There are two kinds of numbers: integers and decimals. An integer does not have a decimal point in it; a decimal does.
- Numbers may have any number of digits, subject to the limitation on line length. Only about seventeen significant figures will be retained, however (enough for all known applications).
- A non-zero number with no sign as the first character is assumed to be positive.

Notice that initial (before the decimal point and the first non-zero digit) and trailing (after the decimal point and the last non-zero digit) zeros are allowed but not required. A number written with initial or trailing zeros will have the same value when it is read as if the extra zeros were not there.

Numbers used for specific purposes in RS274/NGC are often restricted to some finite set of values or some to some range of values. In many uses, decimal numbers must be close to integers; this includes the values of indexes (for parameters and carousel slot numbers, for example), M codes, and G codes multiplied by ten. A decimal number which is supposed be close to an integer is considered close enough if it is within 0.0001 of an integer.

Parameter Value (aka “hashvars”)

A parameter value is the pound (or hash) character '#' followed by a real value. The real value must evaluate to an integer between 1 and 5399. The integer is a parameter number, and the value of the parameter value is whatever number is stored in the numbered parameter.

The # character takes precedence over other operations, so that, for example, “#1+2” means the number found by adding 2 to the value of parameter 1, not the value found in parameter 3. Of course, #[1+2] does mean the value found in parameter 3. The # character may be repeated; for example ##2 means the value of the parameter whose index is the (integer) value of parameter 2.

NOTE: MotusCNC uses parameters 4000-5399 for special purposes, so these should not be used for general program parameters, unless documented as such.

Expressions and Binary Operations

An expression is a set of characters starting with a left bracket [and ending with a balancing right bracket]. In between the brackets are numbers, parameter values, mathematical operations, and other expressions. An expression may be evaluated to produce a number. The expressions on a line are evaluated when the line is read, before anything on the line is executed. An example of an expression is

```
 [ 1 + acos[0] - [#3 ** [4.0/2]]]
```

Binary operations appear only inside expressions. Fifteen binary operations are defined. There are four basic mathematical operations: addition (+), subtraction (-), multiplication (*), and division (/). There are three logical operations: non-exclusive or (OR), exclusive or (XOR), and logical and (AND). The eighth operation is the modulus operation (MOD). The ninth operation is the “power” operation (**) of raising the number on the left of the operation to the power on the right.

There are also the 6 arithmetic comparison operators: less-than (<), less-than-or-equal (<=), equal (=), greater-than-or-equal (>=), greater-than (>) and not-equal (<>).

The binary operations are divided into three groups:

**	Highest precedence
* / MOD	
+ - OR XOR AND < <= = >= > <>	Lowest precedence

The first group is: power. The second group is: multiplication, division, and modulus. The third group is: addition, subtraction, logical non- exclusive or, logical exclusive or, logical and, and all the comparison operators. If operations are strung together (for example in the expression [2.0 / 3 * 1.5 - 5.5 / 11.0]), operations in the first group are to be performed before operations in the second group and operations in the second group before operations in the third group. If an expression contains more than one operation from the same group (such as the first / and * in the example), the operation on the left is performed first. Thus, the example is equivalent to: [((2.0 / 3) * 1.5) - (5.5 / 11.0)] , which simplifies to [1.0 - 0.5] , which is 0.5.

The logical operations and modulus are to be performed on any real numbers, not just on integers.

The result of the logical and comparison operations is either 0.0 (false) or 1.0 (true).

As an operand of a logical operator, the number zero is equivalent to logical false, and any non-zero number is equivalent to logical true.

Unary Operation Value

A unary operation value is either "ATAN" followed by one expression divided by another expression (for example "ATAN[2]/[1+3]") or any other unary operation name followed by an expression (for example "SIN[90]"). The unary operations are: ABS (absolute value), ACOS (arc cosine), ASIN (arc sine), ATAN (arc tangent), COS (cosine), EXP (e raised to the given power), FIX (round down), FUP (round up), LN (natural logarithm), ROUND (round to the nearest whole number), SIN (sine), SQRT (square root), and TAN (tangent). Arguments to unary operations which take angle measures (COS, SIN, and TAN) are in degrees. Values returned by unary operations which return angle measures (ACOS, ASIN, and ATAN) are also in degrees.

Group	Unary operator	Notes
Inverse trig	ATAN	2-argument arctan(y/x)
	ASIN	arcsine
	ACOS	arccos
Trigonometric	SIN	sine
	COS	cosine
	TAN	tangent
Eulerian	EXP	e^x
	LN	natural logarithm
Truncation	FIX	floor
	FUP	ceil
	ROUND	nearest integer
Other	SQRT	square root
	ABS	absolute value

The FIX operation rounds towards the left (less positive or more negative) on a number line, so that FIX[2.8] =2 and FIX[-2.8] = -3, for example. The FUP operation rounds towards the right (more positive or less negative) on a number line; FUP[2.8] = 3 and FUP[-2.8] = -2, for example.

FIX is commonly known as the "floor" function, while FUP is the "ceiling" function.

Parameter Setting

A parameter setting is the following four items one after the other: (1) a pound character # , (2) a real value which evaluates to an integer between 1 and 5399, (3) an equal sign = , and (4) a real value. For example "#3 = 15" is a parameter setting meaning "set parameter 3 to 15."

A parameter setting does not take effect until after all parameter values on the same line have been found. For example, if parameter 3 has been previously set to 15 and the line "#3=6 G1 x#3" is interpreted, a straight move to a point where x equals 15 will occur and the value of parameter 3 will be 6.

Comments and Messages

Printable characters and white space inside parentheses is a comment. A left parenthesis always starts a comment. The comment ends at the first right parenthesis found thereafter. Once a left parenthesis is placed on a line, a matching right parenthesis must appear before the end of the line.

Comments may not be nested; it is an error if a left parenthesis is found after the start of a comment and before the end of the comment. Here is an example of a line containing a comment:

```
G80 M5 (stop motion)
```

Comments do not cause a machining center to do anything.

A comment contains a message if “MSG,” appears after the left parenthesis and before any other printing characters. Variants of “MSG,” which include white space and lower case characters are allowed. The rest of the characters before the right parenthesis are considered to be a message.

Messages are displayed in the MotusCNC message area, and cause the interpreter to halt, like an M0 command. For example:

```
(msg, Please check for tool clearance)
```

will halt the program and display that message. The operator continues the program by pressing the Cycle Start button.

MotusCNC supports some extensions to the standard messaging convention as described below.

Markup in Messages

If the first non-blank character of a MSG comment is a commercial 'at' sign (@), then the message text may include a limited form of markup, which allows messages to be in bold, italic or different colors.

Example:

```
(msg,@<b>This message is in boldface</b>)
(msg,@And the last word of this line is <i>italic</i>)
(msg,@<span fgcolor="red">Danger Will Robinson!</span>)
```

Acceptable markup is described in the following table:

Start tag	End tag	Meaning
<i>	</i>	Italicize content between tags
		Boldface content between tags
		Set color of content between tags. *color* should be a simple color name, like red or blue.
&		Ampersand character (&)
<		Less-than character (<)
(		Open parenthesis
)		Close parenthesis
&#*ddd*;		Any character, specified by its ASCII decimal number as *ddd*.

The ability to specify character entities ((etc.) allows working around the inability to directly use characters like parentheses which otherwise would cause a syntax error.

Note that only ampersand and less-than may be specified by name rather than ASCII code.

Multi-line Messages

In some cases, the message is too long to reasonably fit on one line in the message area. MotusCNC provides an extension so that multi-line message may be generated. Markup may also be used.

A new type of “magic comment” is defined. instead of starting with MSG, these comments start with USR. The first character of message lines other than the first message starts with plus (+). Example:

```
(usr,First line)
(usr,+Second line)
(usr,+Third line)
M0
```

The final M0 (program pause) command is required to actually display the message. As the NC interpreter reads the successive USR comments, it stores them internally. Since there is no explicit indication of the “last” message, the M0 (or M1) is required to finally output the accumulated message.

Multi-line messages can include markup if the first USR message starts with '@':

```
(usr,@Ready to machine next part)
M0 (usr,+<span fgcolor=”green”>Press cycle start</span>)
```

The M0 may be specified on the last USR message line, since stop commands are executed after comments.

Interpreter Directive Messages

These messages are comments which contain instructions for the NC interpreter, which are specific to MotusCNC.

Any comment starting directly with the back-quote character (`) is an interpreter directive.

At present, the only meaningful directive is

```
(`nosim)
```

This indicates that the remainder of the NC program is not to be simulated. This is usually necessary for programs that include probing and other events that provide information that cannot be predicted by the simulator. Attempting to simulate such programs may cause the simulator to think there are errors, when none would occur when the program actually runs.

Avoid coding any comment which starts with the back-quote character. Future versions of MotusCNC may implement new directives, so it is important to avoid any possible future conflicts.

Item Repeats

A line may have any number of G words, but two G words from the same modal group (see Section 3.4) may not appear on the same line.

A line may have zero to four M words. Two M words from the same modal group may not appear on the same line.

For all other legal letters, a line may have only one word beginning with that letter.

If a parameter setting of the same parameter is repeated on a line, "#3=15 #3=6", for example, only the last setting will take effect. It is silly, but not illegal, to set the same parameter twice on the same line.

If more than one comment appears on a line, only the last one will be used; each of the other comments will be read and its format will be checked, but it will be ignored thereafter. It is expected that putting more than one comment on a line will be very rare.

Item order

The three types of item whose order may vary on a line (as given at the beginning of this section) are word, parameter setting, and comment. Imagine that these three types of item are divided into three groups by type.

Word order

The first group (the words) may be reordered in any way without changing the meaning of the line.

Parameter order

If the second group (the parameter settings) is reordered, there will be no change in the meaning of the line unless the same parameter is set more than once. In this case, only the last setting of the parameter will take effect. For example, after the line "#3=15 #3=6" has been interpreted, the value of parameter 3 will be 6. If the order is reversed to "#3=6 #3=15" and the line is interpreted, the value of parameter 3 will be 15.

The result of using and referencing a parameter on the same line is undefined. For example,

```
#1=2 #2=#1
```

does not necessarily set #2 to 2, as might be expected. This is because #1 is both set (in the first expression) and referenced (in the second).

Comment order

If the third group (the comments) contains more than one comment and is reordered, only the last comment will be used. For example,

```
(msg,Hello) (msg,World)
```

will only show the message 'World'. The first comment is ignored.

If each group is kept in order or reordered without changing the meaning of the line, then the three groups may be interleaved in any way without changing the meaning of the line. For example, the line

```
g40 g1 #3=15 (foo) #4=-7.0
```

has five items and means exactly the same thing in any of the 120 possible orders such as

```
#4=-7.0 g1 #3=15 g40 (foo)
```

for the five items.

Commands and Machine Modes

In RS274/NGC, many commands cause a machining center to change from one mode to another, and the mode stays active until some other command changes it implicitly or explicitly. Such commands are called "modal". For example, if coolant is turned on, it stays on until it is explicitly turned off. The G codes for motion are also modal. If a G1 (straight move) command is given on one line, for example, it will be executed again on the next line if one or more axis words is available on the line, unless an explicit command is given on that next line using the axis words or cancelling motion.

"Non-modal" codes have effect only on the lines on which they occur. For example, G4 (dwell) is non-modal.

Modal Groups

Modal commands are arranged in sets called "modal groups", and only one member of a modal group may be in force at any given time. In general, a modal group contains commands for which it is logically impossible for two members to be in effect at the same time — like measure in inches vs. measure in millimeters. A machining center may be in many modes at the same time, with one mode from each modal group being in effect. The modal groups are shown in Table 4.

The modal groups for G codes are:

group 1 = {G0, G1, G2, G3, G38, G38.2, G53.1, G80, G81, G82, G83, G84, G85, G86, G87, G88, G89} motion

group 2 = {G17, G18, G19} plane selection

group 3 = {G90, G91} distance mode

group 5 = {G93, G94} feed rate mode

group 6 = {G20, G21} units

group 7 = {G40, G41, G42} cutter radius compensation

group 8 = {G43, G43.4, G43.5, G49} tool length offset

group 10 = {G98, G99} return mode in canned cycles group 12 = {G54, G55, G56, G57, G58, G59, G59.1, G59.2, G59.3, G54.4} coordinate system selection

group 13 = {G61, G61.1, G64} path control mode

group 14 = {G96, G97} constant spindle speed mode

The modal groups for M codes are:

group 4 = {M0, M1, M2, M30, M60, M97, M98, M99} stopping and program flow (subroutines)

group 6 = {M6} tool change

group 7 = {M3, M4, M5} spindle turning

group 8 = {M7, M8, M9} coolant (special case: M7 and M8 may be active at the same time)

group 10 = {M100 through M119} special purpose

In addition to the above modal groups, there is a group for non-modal G codes:

group 0 = {G4, G10, G10.2, G10.4, G10.7, G10.8, G28, G30, G52, G53, G68.2, G68.4, G69, G92, G92.1, G92.2, G92.3}

For several modal groups, when a machining center is ready to accept commands, one member of the group must be in effect. There are default settings for these modal groups. When the machining center is turned on or otherwise re-initialized, the default values are automatically in effect.

Group 1, the first group on the table, is a group of G codes for motion. One of these is always in effect. That one is called the current motion mode.

It is an error to put a G-code from group 1 and a G-code from group 0 on the same line if both of them use axis words. If an axis word-using G-code from group 1 is implicitly in effect on a line (by having been activated on an earlier line), and a group 0 G-code that uses axis words appears on the line, the activity of the group 1 G-code is suspended for that line. The axis word-using G- codes from group 0 are G10, G28, G30, and G92.

G Codes

G codes of the RS274/NGC language are shown in following table and described after that.

The descriptions contain command prototypes.

In the command prototypes, three dots (...) or an underscore (_) stand for a real value. As described earlier, a real value may be (1) an explicit number, 4, for example, (2) an expression, [2+2], for example, (3) a parameter value, #88, for example, or (4) a unary function value, acos[0], for example.

In most cases, if axis words (any or all of X..., Y..., Z..., A..., B..., C...) are given, they specify a destination point. Axis numbers are in the currently active coordinate system, unless explicitly described as being in the absolute coordinate system. Where axis words are optional, any omitted axes will have their current value. Any items in the command prototypes not explicitly described as optional are required. It is an error if a required item is omitted.

In the prototypes, the values following letters are often given as explicit numbers. Unless stated otherwise, the explicit numbers can be real values. For example, G10 L2 could equally well be written

G[2*5] L[1+1]. If the value of parameter 100 were 2, G10 L#100 would also mean the same. Using real values which are not explicit numbers as just shown in the examples is rarely useful.

If L... or L_ is written in a prototype the "..." or "_" will often be referred to as the "L number". Similarly the "..." in H... may be called the "H number", and so on for any other letter.

G-code	Meaning	Notes
G0	Rapid positioning	
G1	Linear interpolation	
G2	Clockwise arc or helix	
G3	Counterclockwise arc or helix	
G4	Dwell	
G10	Coordinate system origin setting	
G10.2	Set WSEC	See the WSEC chapter
G10.4	Compose WSEC	See the WSEC chapter
G10.7	Probing options	See the G38 probing chapter
G10.8	Probe result buffer	See the G10.8 Probe Data Buffer chapter
G17	XY plane	
G18	ZX plane	
G19	YX plane	
G20	Imperial (inch) units	
G21	Metric (mm) units	
G28	Home position 1 (near axis homing positions)	
G30	Home position 2 (above rotary table center)	
G38	General probing	See the G38 probing chapter
G38.2	Simple linear probing	See the G38 probing chapter
G40	Cancel tool radius compensation	
G41	Start tool radius compensation on left	
G42	Start tool radius compensation on right	
G43	Start tool length compensation	
G43.4	Start TCPC type 1	See the 5-axis kinematics chapter
G43.5	Start TCPC type 2	See the 5-axis kinematics chapter
G43.9	Change tool length compensation	
G49	Cancel tool length compensation	

G-code	Meaning	Notes
G52	Directly set global offset	
G53	Move in absolute coordinates	
G53.1	Orient tool to local Z axis	
G54,55,56,57, 58,59, 59.1,59.2,59.3	Select fixture offset	
G54.4	Select WSEC offset	See the WSEC chapter
G68.2	Set TWP	See the TWP chapter
G68.4	Compose TWP	See the TWP chapter
G69	Cancel TWP	See the TWP chapter
G80	Cancel motion	
G81	Canned cycle: drill	
G82	Canned cycle: drill with dwell	
G83	Canned cycle: peck drill	
G84	Canned cycle: right-hand tap	Not useful with available spindles
G85	Canned cycle: boring/reaming	
G86	Canned cycle: boring with stopped retract	
G87	Canned cycle: back boring	Not useful with available spindles
G88	Canned cycle: boring with manual retract	
G89	Canned cycle: boring with dwell	
G90	Absolute distance mode	
G90.1	Absolute distance mode for arc I,J,K	
G91	Incremental distance mode	
G91.1	Incremental distance mode for arc I,J,K	
G92	Indirectly set global offset	
G92.1	Cancel global offset and set parameters zero	
G92.2	Cancel global offset	
G92.3	Set global offset from parameters	
G93	Inverse time feed rate	
G94	Units per minute feed rate	
G96	Constant spindle speed mode – lathe only	
G97	Normal spindle mode – lathe only	

G-code	Meaning	Notes
G98	Initial level return for canned cycles	
G99	R level return for canned cycles.	

Rapid Linear Motion — G0

For rapid linear motion, program

```
G0 X_ Y_ Z_ A_ B_ C_ I_ J_ K_
```

where all the axis words are optional, except that at least one must be used. The G0 is optional if the current motion mode is G0. This will produce coordinated linear motion to the destination point at the current traverse rate (possible slowed down by the rapid override setting). It is expected that cutting will not take place when a G0 command is executing.

I,J,K are only permissible when running in TCPC type 2 mode, in which case they specify the end point tool orientation as a vector. If at least one of I,J,K is specified, then any which are omitted are assumed to be 0. If all are omitted, there is no orientation change.

A,B,C are only permissible in 3+2 axis, or TCPC type 1 mode, and the actual words used must correspond to the kinematics mode: A and B are used for 3+2, and A and C are used for TCPC type 1.

It is an error if:

- all axis words are omitted.
- an inappropriate axis word is used for the current kinematics mode.

If cutter radius compensation is active, the motion will differ from the above. If G53 is programmed on the same line, the motion will also differ.

TCPC is suppressed if G53 is used.

Linear Motion at Feed Rate — G1

For linear motion at feed rate (for cutting or not), program

```
G1 X_ Y_ Z_ A_ B_ C_ I_ J_ K_
```

where all the axis words are optional, except that at least one must be used. The G1 is optional if the current motion mode is G1. This will produce coordinated linear motion to the destination point at the current feed rate (or slower if the machine will not go that fast).

In terms of axis words, usage of G1 is identical to G0. The only differences are:

- Motion honors the current feed (F) rate with G1, including feed or combined overrides.
- In TCPC modes, TCPC control in G1 is maintained even outside the defined workpiece envelope. With G0, TCPC mode is suppressed if either end of the motion is outside the workpiece envelope.

Arc at Feed Rate — G2 and G3

A circular or helical arc is specified using either G2 (clockwise arc) or G3 (counterclockwise arc). The axis of the circle or helix must be parallel to the X, Y, or Z-axis of the machine coordinate system. The axis (or, equivalently, the plane perpendicular to the axis) is selected with G17 (Z- axis, XY-plane), G18 (Y-axis, XZ-plane), or G19 (X-axis, YZ-plane). If the arc is circular, it lies in a plane parallel to the selected plane.

If a line of RS274/NGC code makes an arc and includes rotational axis motion, the rotational axes turn at a constant rate so that the rotational motion starts and finishes when the XYZ motion starts and finishes. Lines of this sort are hardly ever programmed.

If cutter radius compensation is active, the motion will differ from what is described here.

Two formats are allowed for specifying an arc, except in TCPC type 2 mode, where only the radius format is allowed. We will call these the center format and the radius format. In both formats the G2 or G3 is optional if it is the current motion mode.

MotusCNC supports TWP and WSEC which may rotate the local coordinate system so that the arc axis is not necessarily parallel to the X, Y or Z axes. In other words, selecting G17 (XY plane) is only a local specification and does not necessarily correspond to the absolute machine XY plane.

In the case of tilted work planes, arcs are still supported. In addition, MotusCNC relaxes many of the restrictions on arcs that were originally imposed by RS274NGC. For example, the post processor settings may be used to:

- Allow center format arcs to specify the center directly in program coordinates (not just as an offset);
- Allow the starting and ending radii to differ substantially, so that spirals are generated.

In addition, TCPC modes allow the tool orientation to be specified at the end of the arc, and helical interpolation follows the Fanuc specification for this.

Radius Format Arc

In the radius format, the coordinates of the end point of the arc in the selected plane are specified along with the radius of the arc. Program

```
G2 X_ Y_ Z_ A_ B_ C_ R_ I_ J_ K_
```

(or use G3 instead of G2). R is the radius. The axis words are all optional except that at least one of the two words for the axes in the selected plane must be used. The R number is the radius. A positive radius indicates that the arc turns through 180 degrees or less, while a negative radius indicates a turn of 180 degrees to 359.999 degrees. If the arc is helical, the value of the end point of the arc on the coordinate axis parallel to the axis of the helix is also specified.

I,J and K specify the ending tool orientation vector in TCPC type 2 mode.

A,B specify direct rotary axis end points in 3+2 mode, or A,C specify ending tool orientation in TCPC type 1 mode.

It is an error if:

- both of the axis words for the axes of the selected plane are omitted,

- the end point of the arc is the same as the current point.
- I,J or K are used other than in TCPC type 2 kinematics mode.
- A,B or C are used in TCPC type 2 mode.

It is not good practice to program radius format arcs that are nearly full circles or are semicircles (or nearly semicircles) because a small change in the location of the end point will produce a much larger change in the location of the center of the circle (and, hence, the middle of the arc).

The magnification effect is large enough that rounding error in a number can produce out-of- tolerance cuts. Nearly full circles are outrageously bad, semicircles (and nearly so) are only very bad. Other size arcs (in the range tiny to 165 degrees or 195 to 345 degrees) are OK.

Here is an example of a radius format command to mill an arc:

```
G17 G2 x 10 y 15 r 20 z 5
```

That means to make a clockwise (as viewed from the positive Z-axis) circular or helical arc whose axis is parallel to the Z-axis, ending where X=10, Y=15, and Z=5, with a radius of 20. If the starting value of Z is 5, this is an arc of a circle parallel to the XY-plane; otherwise it is a helical arc.

Center Format Arc

In the center format, the coordinates of the end point of the arc in the selected plane are specified along with either:

- the offsets of the center of the arc from the current location.
- the program coordinates of the center of the arc.

The convention to use is specified in the post processor settings in the MotusCNC settings window. By default, program coordinates are used (since this is what most post processors are set to), however the setting may be changed to the arc center "offset" mode if that is the local shop practice.

In addition, it is possible to specify the arc center convention on an individual program basis using the G90.1 or G91.1 commands. G90.1 uses absolute mod, G91.1 uses incremental mode.

In this format, it is OK if the end point of the arc is the same as the current point. It is an error if:

- when the arc is projected on the selected plane, the distance from the current point to the center differs from the distance from the end point to the center by more than 0.0002 inch (if inches are being used) or 0.002 millimeter (if millimeters are being used). This restriction may be relaxed by specifying the appropriate post processor setting in the MotusCNC settings window. MotusCNC allows separate tolerances for metric and imperial, and the tolerances may be very large if desired to allow spirals.

When the XY-plane is selected, program

```
G2 X_ Y_ Z_ A_ B_ C_ I_ J_
```

(or use G3 instead of G2). The axis words are all optional except that at least one of X and Y must be used. I and J are the offsets from the current location, or direct coordinates, (in the X and Y directions, respectively) of the center of the circle, depending on the post processor setting. I and J are optional except that at least one of the two must be used. It is an error if:

- X and Y are both omitted,

- I and J are both omitted.

When the ZX-plane is selected, the axis words are analogous, and I and K are used.

When the YZ-plane is selected, the axis words are analogous, and J and K are used.

Here is an example of a center format command to mill an arc (assuming center offsets):

```
G17 G2 x 10 y 16 i 3 j 4 z 9
```

That means to make a clockwise (as viewed from the positive z-axis) circular or helical arc whose axis is parallel to the Z-axis, ending where X=10, Y=16, and Z=9, with its center offset in the X direction by 3 units from the current X location and offset in the Y direction by 4 units from the current Y location. If the current location has X=7, Y=7 at the outset, the center will be at X=10, Y=11. If the starting value of Z is 9, this is a circular arc; otherwise it is a helical arc. The radius of this arc would be 5.

In the center format, the radius of the arc is not specified, but it may be found easily as the distance from the center of the circle to either the current point or the end point of the arc.

Dwell — G4

For a dwell, program

```
G4 P_
```

This will keep the axes unmoving for the period of time:

- in seconds if the P value is less than 5;
- in milliseconds if the P value is >= 5.

Thus, a 6 second dwell must be coded as

```
G4 P6000
```

It is an error if:

- the P number is negative.

Set Coordinate System Data — G10

To set the coordinate values for the origin of a coordinate system, program

```
G10 L2 P_ X_ Y_ Z_ A_ B_ C_
```

where the P number must evaluate to an integer in the range 1 to 9 (corresponding to G54 to G59.3) and all axis words are optional. The coordinates of the origin of the coordinate system specified by the P number are reset to the coordinate values given (in terms of the absolute coordinate system). Only those coordinates for which an axis word is included on the line will be reset.

It is an error if:

- the P number does not evaluate to an integer in the range 1 to 9.

If origin offsets (made by G92 or G92.3) were in effect before G10 is used, they will continue to be in effect afterwards.

The coordinate system whose origin is set by a G10 command may be active or inactive at the time the

G10 is executed.

Example:

```
G10 L2 P1 x 3.5 y 17.2
```

sets the origin of the first coordinate system (the one selected by G54) to a point where X is 3.5 and Y is 17.2 (in absolute coordinates). The Z coordinate of the origin (and the coordinates for any rotational axes) are whatever those coordinates of the origin were before the line was executed.

Plane Selection — G17, G18, and G19

Program G17 to select the XY-plane, G18 to select the ZX-plane, or G19 to select the YZ-plane.

Note that if TWP or WSEC is in effect, the program coordinate system may be rotated with respect to the machine axes. These plane selection commands refer to the local (program) coordinate system. The "working plane" is used by G2 and G3 arc commands, as well as some canned cycles.

Length Units — G20 and G21

Program G20 to use inches for length units. Program G21 to use millimeters.

It is usually a good idea to program either G20 or G21 near the beginning of a program before any motion occurs, and not to use either one anywhere else in the program. It is the responsibility of the user to be sure all numbers are appropriate for use with the current length units.

Return to Home — G28 and G30

Two home positions are defined by MotusCNC:

G28 is the normal home position for each axis (generally, near the limit switch).

G30 is directly above the trunnion table rotary center.

The action of G28 and G30 is considerably changed from the original RS274NGC, and complies more closely with industry standard.

```
G91 G28 X_ Y_ Z_ A_ B_   (G30 is similar)
G90 (back to absolute distance mode if required)
```

Although G91 and G90 are not part of this command, it is almost mandatory to use incremental motion (G91) mode when using G28/G30.

G28/G30 move one or more axes to their home positions, via the specified way point. The only axes moved are those mentioned on the same block. The axis word values (taken together) specify an intermediate point, to which an initial traverse move is made. Then, each axis moves at traverse rate to its specified home position.

Incremental mode (G91) is highly recommended, with an axis value of 0 so that there is no initial motion, and the axes mentioned in the command go directly to their home position.

Examples:

```
G91 G30 Z0
```

Move the Z axis only to its home position.

```
G91 G30 Z0
```

```
G30 X0 Y0 A0
```

Move the Z axis home first. Then, move X, Y and A simultaneously to their home position directly over the rotational center of the trunnion table (and the A axis in the normal “horizontal” position).

```
G90 G28 Z10 X30 Y-15
```

Probably a bad idea: G90 is not incremental, so it is not necessarily clear where the intermediate point (30,-15,10) is – it would depend on current offsets etc.. And then, it is going to move X,Y and Z to the home position together, which might crash into something. It's usually best to get Z to a safe level before moving X and Y.

Probing — G38 and G38.2

See chapter on probing for a complete description.

Cutter Radius Compensation — G40, G41, and G42

To turn cutter radius compensation off, program G40. It is OK to turn compensation off when it is already off.

Cutter radius compensation may be performed only if the XY-plane is active.

To turn cutter radius compensation on left (i.e., the cutter stays to the left of the programmed path when the tool radius is positive), program

```
G41 D_
```

To turn cutter radius compensation on right (i.e., the cutter stays to the right of the programmed path when the tool radius is positive), program G42 D... .

The D word is optional; if there is no D word, the radius of the tool currently in the spindle will be used. If used, the D number should normally be the slot number of the tool in the spindle, although this is not required. It is OK for the D number to be zero; a radius value of zero will be used.

It is an error if:

- the D number is not an integer, is negative or is larger than the number of carousel slots,
- the XY-plane is not active,
- cutter radius compensation is commanded to turn on when it is already on.

Tool Length Offsets and TCPC — G43, G43.4, G43.5, G43.9 and G49

To use a tool length offset in 3-axis or 3+2 kinematics mode, program

```
G43 H_
```

where the H number is the desired index in the tool table. It is expected that all entries in this table will be positive. The H number should be, but does not have to be, the same as the slot number of the tool currently in the spindle. It is OK for the H number to be zero; an offset value of zero will be used (although it is preferable to use G49 for this).

It is an error if:

- the H number is not an integer, is negative, or is larger than the number of carousel slots.

To use no tool length offset, program G49.

It is OK to program using the same offset already in use. It is also OK to program using no tool length offset if none is currently being used.

Use G43.4 H_ to start TCPC type 1 mode.

Use G43.5 H_ P_ to start TCPC type 2 mode.

The TCPC modes are described in great detail in the chapter on 5-axis operation.

G43.9 H_ is similar to G43, except the current TCPC mode is retained. This is intended for manual intervention (MDI) to avoid inadvertent change of 5-axis mode.

Move in Absolute Coordinates — G53

For linear motion to a point expressed in absolute coordinates, program

```
G1 G53 X_ Y_ Z_ A_ B_  (or use G0 instead of G1)
```

where all the axis words are optional, except that at least one must be used. The G0 or G1 is optional if it is the current motion mode. G53 is not modal and must be programmed on each line on which it is intended to be active. This will produce coordinated linear motion to the programmed point. If G1 is active, the speed of motion is the current feed rate (or slower if the machine will not go that fast). If G0 is active, the speed of motion is the current traverse rate (or slower if the machine will not go that fast).

It is an error if:

- G53 is used without G0 or G1 being active,
- G53 is used while cutter radius compensation is on.

G53 motion temporarily cancels TCPC modes.

Align Tool Axis Perpendicular – G53.1

This command causes the machine to move at least its rotary axes, so that the tool axis is aligned along the current program Z axis. In TCPC modes, the controlled point will not move with respect to the workpiece/table, however there may be a rotary movement that will occur at traverse rate.

It is very common to issue this command immediately after a TWP or WSEC command, since the program code can then use 2-D programming (in X and Y) including canned cycles.

Select Coordinate System (Conventional Fixture Offset, CFO) — G54 to G59.3

To select coordinate system 1, program G54, and similarly for other coordinate systems. The system-number—G-code pairs are: (1—G54), (2—G55), (3—G56), (4—G57), (5—G58), (6— G59), (7—G59.1), (8—G59.2), and (9—G59.3).

Use of CFO is mutually exclusive with WSEC. Thus, there are effectively 16 different coordinate systems that may be selected: 9 conventional and 7 WSEC.

The primary difference between CFO and WSEC is that CFOs do not truly allow compensation for workpiece rotation: the angular offsets in the CFO are merely added to the rotary axis coordinates, which is not the same as the true rotation compensation allowed by WSEC.

It is an error if:

- one of these G-codes is used while cutter radius compensation is on.

Select Work Setting Error Compensation (WSEC) — G54.4

See the chapter on WSEC for further details.

G54.4 P_ selects one of 7 (P1-P7) WSEC offsets.

G54.4 P0 clears the WSEC offset.

NOTE: Use of WSEC and the conventional fixture offset (G54 etc.) is **mutually exclusive**. It is recommended that after canceling WSEC, the conventional fixture offset is explicitly selected in the immediately following code. For example:

```
G55 (Conventional offset)
...
G54.4 P1 (Use WSEC 1)
...
G54.4 P0 (Turn off WSEC)
G55 (Back to conventional offset)
...
```

Cancel Modal Motion — G80

Program G80 to ensure no axis motion will occur. It is an error if:

- Axis words are programmed when G80 is active, unless a modal group 0 G code is programmed which uses axis words.

Note that G80 will cancel any motion mode in effect, including G0/1.

Canned Cycles — G81 to G89

The canned cycles G81 through G89 have been implemented as described in this section. Two examples are given with the description of G81 below.

All canned cycles are performed with respect to the currently selected plane. Any of the three planes (XY, YZ, ZX) may be selected. Throughout this section, most of the descriptions assume the XY-plane has been selected. The behavior is always analogous if the YZ or ZX-plane is selected.

Rotational axis (or tool orientation) words are allowed in canned cycles, but it is better to omit them. If rotational axis words are used, the numbers must be the same as the current position numbers so that the rotational axes do not move.

All canned cycles use X, Y, R, and Z numbers in the NC code. These numbers are used to determine X, Y, R, and Z positions. The R (usually meaning retract) position is along the axis perpendicular to the currently selected plane (Z-axis for XY-plane, X-axis for YZ-plane, Y-axis for ZX-plane). Some canned cycles use additional arguments.

For canned cycles, we will call a number "sticky" if, when the same cycle is used on several lines of code in a row, the number must be used the first time, but is optional on the rest of the lines.

Sticky numbers keep their value on the rest of the lines if they are not explicitly programmed to be different. The R number is always sticky.

In incremental distance mode: when the XY-plane is selected, X, Y, and R numbers are treated as increments to the current position and Z as an increment from the Z-axis position before the move involving Z takes place; when the YZ or ZX-plane is selected, treatment of the axis words is analogous. In absolute distance mode, the X, Y, R, and Z numbers are absolute positions in the current coordinate system.

The L number is optional and represents the number of repeats. L=0 is not allowed. If the repeat feature is used, it is normally used in incremental distance mode, so that the same sequence of motions is repeated in several equally spaced places along a straight line. In absolute distance mode, L > 1 means "do the same cycle in the same place several times," Omitting the L word is equivalent to specifying L=1. The L number is not sticky.

When L>1 in incremental mode with the XY-plane selected, the X and Y positions are determined by adding the given X and Y numbers either to the current X and Y positions (on the first go- around) or to the X and Y positions at the end of the previous go-around (on the repetitions). The R and Z positions do not change during the repeats.

The height of the retract move at the end of each repeat (called "clear Z" in the descriptions below) is determined by the setting of the retract mode: either to the original Z position (if that is above the R position and the retract mode is G98, OLD_Z), or otherwise to the R position.

It is an error if:

- X, Y, and Z words are all missing during a canned cycle,
- a P number is required and a negative P number is used,
- an L number is used that does not evaluate to a positive integer,
- rotational axis motion is used during a canned cycle,
- inverse time feed rate is active during a canned cycle,
- cutter radius compensation is active during a canned cycle.

When the XY plane is active, the Z number is sticky, and it is an error if:

- the Z number is missing and the same canned cycle was not already active,
- the R number is less than the Z number.

When the ZX plane is active, the Y number is sticky, and it is an error if:

- the Y number is missing and the same canned cycle was not already active,
- the R number is less than the Y number.

When the YZ plane is active, the X number is sticky, and it is an error if:

- the X number is missing and the same canned cycle was not already active,
- the R number is less than the X number.

Preliminary and In-Between Motion

At the very beginning of the execution of any of the canned cycles, with the XY-plane selected, if the current Z position is below the R position, the Z-axis is traversed to the R position. This happens only once, regardless of the value of L.

In addition, at the beginning of the first cycle and each repeat, the following one or two moves are made:

1. a straight traverse parallel to the XY-plane to the given XY-position,
2. a straight traverse of the Z-axis only to the R position, if it is not already at the R position.

If the ZX or YZ plane is active, the preliminary and in-between motions are analogous.

G81 Cycle

The G81 cycle is intended for drilling. Program

```
G81 X... Y... Z... A... B... C... R... L...
```

1. Preliminary motion, as described above.
2. Move the Z-axis only at the current feed rate to the Z position.
3. Retract the Z-axis at traverse rate to clear Z.

Example 1. Suppose the current position is (1, 2, 3) and the XY-plane has been selected, and the following line of NC code is interpreted.

```
G90 G81 G98 X4 Y5 Z1.5 R2.8
```

This calls for absolute distance mode (G90) and OLD_Z retract mode (G98) and calls for the G81 drilling cycle to be performed once. The X number and X position are 4. The Y number and Y position are 5. The Z number and Z position are 1.5. The R number and clear Z are 2.8. Old Z is 3.

The following moves take place.

1. a traverse parallel to the XY-plane to (4,5,3)
2. a traverse parallel to the Z-axis to (4,5,2.8)
3. a feed parallel to the Z-axis to (4,5,1.5)
4. a traverse parallel to the Z-axis to (4,5,3)

Example 2. Suppose the current position is (1, 2, 3) and the XY-plane has been selected, and the following line of NC code is interpreted.

```
G91 G81 G98 X4 Y5 Z-0.6 R1.8 L3
```

This calls for incremental distance mode (G91) and OLD_Z retract mode (G98) and calls for the G81 drilling cycle to be repeated three times. The X number is 4, the Y number is 5, the Z number is -0.6 and the R number is 1.8. The initial X position is 5 (=1+4), the initial Y position is 7 (=2+5), the clear Z position is 4.8 (=1.8+3), and the Z position is 4.2 (=4.8-0.6). Old Z is 3.

The first move is a traverse along the Z-axis to (1,2,4.8), since old Z < clear Z.

The first repeat consists of 3 moves.

1. a traverse parallel to the XY-plane to (5,7,4.8)

2. a feed parallel to the Z-axis to (5,7, 4.2)
3. a traverse parallel to the Z-axis to (5,7,4.8)

The second repeat consists of 3 moves. The X position is reset to 9 (=5+4) and the Y position to 12 (=7+5).

1. a traverse parallel to the XY-plane to (9,12,4.8)
2. a feed parallel to the Z-axis to (9,12, 4.2)
3. a traverse parallel to the Z-axis to (9,12,4.8)

The third repeat consists of 3 moves. The X position is reset to 13 (=9+4) and the Y position to 17 (=12+5).

1. a traverse parallel to the XY-plane to (13,17,4.8)
2. a feed parallel to the Z-axis to (13,17, 4.2)
3. a traverse parallel to the Z-axis to (13,17,4.8)

G82 Cycle

The G82 cycle is intended for drilling. Program

```
G82 X... Y... Z... A... B... C... R... L... P...
```

1. Preliminary motion, as described above.
2. Move the Z-axis only at the current feed rate to the Z position.
3. Dwell for the P number of seconds.
4. Retract the Z-axis at traverse rate to clear Z.

G83 Cycle

The G83 cycle (often called peck drilling) is intended for deep drilling or milling with chip breaking. The retracts in this cycle clear the hole of chips and cut off any long stringers (which are common when drilling in aluminum). This cycle takes a Q number which represents a “delta” increment along the Z-axis. Program

```
G83 X... Y... Z... A... B... C... R... L... Q...
```

1. Preliminary motion, as described above.
2. Move the Z-axis only at the current feed rate downward by delta or to the Z position, whichever is less deep.
3. Rapid back out to the clear_z.
4. Rapid back down to the current hole bottom, backed off a bit.
5. Repeat steps 1, 2, and 3 until the Z position is reached at step 1.
6. Retract the Z-axis at traverse rate to clear Z.

It is an error if:

- the Q number is negative or zero.

G85 Cycle

The G85 cycle is intended for boring or reaming, but could be used for drilling or milling. Program

```
G85 X... Y... Z... A... B... C... R... L...
```

1. Preliminary motion, as described above.
2. Move the Z-axis only at the current feed rate to the Z position.
3. Retract the Z-axis at the current feed rate to clear Z.

G86 Cycle

The G86 cycle is intended for boring. This cycle uses a P number for the number of seconds to dwell. Program

```
G86 X... Y... Z... A... B... C... R... L... P...
```

1. Preliminary motion, as described above.
2. Move the Z-axis only at the current feed rate to the Z position.
3. Dwell for the P number of seconds.
4. Stop the spindle turning.
5. Retract the Z-axis at traverse rate to clear Z.
6. Restart the spindle in the direction it was going.

The spindle must be turning before this cycle is used. It is an error if:

- the spindle is not turning before this cycle is executed.

G88 Cycle

The G88 cycle is intended for boring. This cycle uses a P word, where P specifies the number of seconds to dwell. Program

```
G88 X... Y... Z... A... B... C... R... L... P...
```

1. Preliminary motion, as described above.
2. Move the Z-axis only at the current feed rate to the Z position.
3. Dwell for the P number of seconds.
4. Stop the spindle turning.
5. Stop the program so the operator can retract the spindle manually.
6. Restart the spindle in the direction it was going.

G89 Cycle

The G89 cycle is intended for boring. This cycle uses a P number, where P specifies the number of seconds to dwell. Program

```
G89 X... Y... Z... A... B... C... R... L... P...
```

1. Preliminary motion, as described above.
2. Move the Z-axis only at the current feed rate to the Z position.
3. Dwell for the P number of seconds.
4. Retract the Z-axis at the current feed rate to clear Z.

Set Distance Mode — G90 and G91

Interpretation of NC code can be in one of two distance modes: absolute or incremental.

To go into absolute distance mode, program G90. In absolute distance mode, axis numbers (X, Y, Z, A, B, C) usually represent positions in terms of the currently active coordinate system. Any exceptions to that rule are described explicitly in this section.

To go into incremental distance mode, program G91. In incremental distance mode, axis numbers (X, Y, Z, A, B, C) usually represent increments from the current values of the numbers.

The meanings of I,J,K do not depend on the distance mode. The following commands (G90.1, G91.1) may affect the interpretation of I,J and K.

Set Arc Center Mode — G90.1 and G91.1

G2 and G3 arcs specified with center format (I,J,K words) can specify the center offset in one of two distance modes: absolute or incremental. The default mode is specified using the post processor settings. Individual programs may override the default using either of these commands:

G90.1 sets the center format to program position.

G91.1 sets the center format to incremental (i.e. distance from current program coordinate).

This command would be used rarely (it is not standard). It is intended for the following situation:

Most shops will use output from a single post processor, which will generate arc centers in a consistent format, so that the default format set in the MotusCNC post processor setting will match. On occasion, some g-code from a different post processor will need to be run. If that code happens to use the opposite arc center format, then it will not run correctly. Rather than change the MotusCNC setting just for this job, it would be easier to edit the g-code directly and insert the appropriate G90.1 or G91.1 at the top of the program.

Coordinate System Offsets — G92, G92.1, G92.2, G92.3

To make the current point have the coordinates you want (without motion), program

```
G92 X... Y... Z... A... B... C...
```

where the axis words contain the axis numbers you want. All axis words are optional, except that at least one must be used. If an axis word is not used for a given axis, the coordinate on that axis of the current point is not changed. It is an error if:

- all axis words are omitted.

When G92 is executed, the origin of the currently active coordinate system moves. To do this, origin offsets are calculated so that the coordinates of the current point with respect to the moved origin are as specified on the line containing the G92. In addition, parameters 5211 to 5216 are set to the X, Y, Z, A, B, and C-axis offsets. The offset for an axis is the amount the origin must be moved so that the coordinate of the controlled point on the axis has the specified value.

Here is an example. Suppose the current point is at X=4 in the currently specified coordinate system and the current X-axis offset is zero, then G92 x7 sets the X-axis offset to -3, sets parameter 5211 to -3, and causes the X-coordinate of the current point to be 7.

The axis offsets are always used when motion is specified in absolute distance mode using any of the nine coordinate systems (those designated by G54 - G59.3). Thus all nine coordinate systems are affected by G92.

Being in incremental distance mode has no effect on the action of G92. Non-zero offsets may be already be in effect when the G92 is called. If this is the case, the new value of each offset is A+B, where A is what the offset would be if the old offset were zero, and B is the old offset. For example, after the previous example, the X-value of the current point is 7. If G92 x9 is then programmed, the new X-axis offset is -5, which is calculated by [[7-9] + -3].

To reset axis offsets to zero, program G92.1 or G92.2. G92.1 sets parameters 5211 to 5216 to zero, whereas G92.2 leaves their current values alone.

To set the axis offset values to the values given in parameters 5211 to 5216, program G92.3.

You can set axis offsets in one program and use the same offsets in another program. Program G92 in the first program. This will set parameters 5211 to 5216. Do not use G92.1 in the remainder of the first program. The parameter values will be saved when the first program exits and restored when the second one starts up. Use G92.3 near the beginning of the second program. That will restore the offsets saved in the first program. If other programs are to run between the the program that sets the offsets and the one that restores them, make a copy of the parameter file written by the first program and use it as the parameter file for the second program.

Set Feed Rate Mode — G93 and G94

Two feed rate modes are recognized: units per minute and inverse time. Program G94 to start the units per minute mode. Program G93 to start the inverse time mode.

Generally, inverse time feed rates are used in 3+2 axis programs where the post processor is computing all kinematics. This is used in many legacy programs. With TCPC, the feed rate is directly specified since the controller computes all the kinematics dynamically, so that units per minute feed rate is used, even for 5-axis machining.

In units per minute feed rate mode, an F word (no, not that F word; we mean feedrate) is interpreted to mean the controlled point should move at a certain number of inches per minute, millimeters per minute, or degrees per minute, depending upon what length units are being used and which axis or axes are moving.

In inverse time feed rate mode, an F word means the move should be completed in [one divided by the F number] minutes. For example, if the F number is 2.0, the move should be completed in half a minute.

When the inverse time feed rate mode is active, an F word must appear on every line which has a G1, G2, or G3 motion, and an F word on a line that does not have G1, G2, or G3 is ignored. Being in inverse time feed rate mode does not affect G0 (rapid traverse) motions. It is an error if:

- inverse time feed rate mode is active and a line with G1, G2, or G3 (explicitly or implicitly) does not have an F word.

Set Canned Cycle Return Level — G98 and G99

When the spindle retracts during canned cycles, there is a choice of how far it retracts: (1) retract perpendicular to the selected plane to the position indicated by the R word, or (2) retract perpendicular to the selected plane to the position that axis was in just before the canned cycle started (unless that position is lower than the position indicated by the R word, in which case use the R word position).

To use option (1), program G99. To use option (2), program G98. Remember that the R word has different meanings in absolute distance mode and incremental distance mode.

Input M Codes

M codes accepted by the NC interpreter are shown in the following table.

M Code	Meaning
M0	Program stop
M1	Optional program stop
M2	Program end
M3	Spindle on clockwise
M4	Spindle on counterclockwise
M5	Stop spindle
M6	Tool change
M7	Mist coolant on
M8	Flood coolant on
M9	Coolant off
M30	Program end, rewind, pallet shuttle
M60	Pallet shuttle, program stop
M97	Subroutine call (N word target) or GOTO
M98	Subroutine call (O word target) or GOTO
M99	Subroutine return
M100	Probing (old style)
M101	Point cloud probing
M102	Point cloud buffer control

M Code	Meaning
M105	Miscellaneous interpreter functions
M118	Machine settings control
M119	Miscellaneous machine controller functions

Program Stopping and Ending — M0, M1, M2, M30, M60

To stop a running program temporarily (regardless of the setting of the optional stop switch), program M0.

To stop a running program temporarily (but only if the optional stop switch is on), program M1.

It is OK to program M0 and M1 in MDI mode, but the effect will probably not be noticeable, because normal behavior in MDI mode is to stop after each line of input, anyway.

To exchange pallet shuttles and then stop a running program temporarily (regardless of the setting of the optional stop switch), program M60. Note that “pallet shuttle” does not do anything on current machines. The terminology is retained for compatibility with other systems. It is preferable to use M0 instead of M60.

If a program is stopped by an M0, M1, or M60, pressing the cycle start button will restart the program at the following line.

To end a program, program M2.

To end a program, exchange pallet shuttles, and rewind the program, program M30.

Both M2 and M30 have the following effects.

1. Selected plane is set to XY (like G17).
2. Distance mode is set to absolute (like G90).
3. Feed rate mode is set to units per minute (like G94).
4. Cutter compensation is turned off (like G40).
5. The spindle is stopped (like M5).
6. The current motion mode is set to straight feed (like G1).
7. Coolant is turned off (like M9).
8. Statistics for the completed job are generated.

No more lines of code in an RS274/NGC file will normally be executed after the M2 or M30 command is executed. For M2, MotusCNC requires the operator to press Rewind before the program can be started again, otherwise the program will continue from the block after the M2, which is probably undesirable. M30 is like M2, except that the program is also rewound (exactly like pressing the Rewind button). Thus, to avoid potential operator error, it is preferable to use M30 to terminate the program.

Spindle Control — M3, M4, M5

To start the spindle turning clockwise at the currently programmed speed, program M3.

To start the spindle turning counterclockwise at the currently programmed speed, program M4.

To stop the spindle from turning, program M5.

It is OK to use M3 or M4 if the spindle speed is set to zero. If this is done (or if the speed override switch is enabled and set to zero), the spindle will not start turning. If, later, the spindle speed is set above zero (or the override switch is turned up), the spindle will start turning. It is OK to use M3 or M4 when the spindle is already turning or to use M5 when the spindle is already stopped.

Tool Change — M6

To change a tool in the spindle from the tool currently in the spindle to the tool most recently selected (using a T word), program M6.

The details of tool change depend on the machine hardware. The M6 command is generalized to support the concept of conventional spindle-mounted tools, and non-spindle mounted tools such as independent touch probes or cameras.

The machine keeps track of two tool states:

- Which tool is mounted in the spindle (if any);
- Which independent (virtual or non-spindle mount) tool is in use (if any).

"Tool numbers" (which are specified for T, H or D words) are classified as follows:

- Null – T0 represents the 'null' tool i.e. no tool at all, or empty spindle.
- Slot number – numbers 1..89 represent physical tool slot numbers (usually only the first 8 or 10 are valid).
- Virtual slot number – numbers 90..99 represent "virtual" tool slots, which basically cause the machine to use an alternative procedure for mounting the tool, or specify non spindle-mount tools.
- Tool ID – numbers 100..9999 represent a tool ID i.e. a tool defined by its characteristics rather than a physical slot position. The software will map this ID to a physical slot (but not a virtual slot) when the program is run and the M6 command is encountered. The mapping is defined by the user maintaining a tool table via MotusCNC. The mapping is shared over all projects.

Use of tool IDs allows improved and simplified programming, because the G-code does not need to be modified (or re-posted) if tool slot loading is changed, and tool life management and "tool pooling" are simplified.

Tool life management allows automatic tool replacement during a long-running program. Each tool slot retains an associated lifetime counter. This counter is incremented with special M105 commands. When the defined tool lifetime is exceeded, the next tool with the same ID will be used automatically at the next tool change for that ID. Multiple tools with the same ID is an instance of "tool pooling".

For example, if a job drills many small holes in hard material, it is quite likely that a single drill would wear out and break before the job was completed. Thus, it would be preferable to load the same drill type into multiple tool changer slots, then make sure any one drill was not used more than a fixed number of times.

As mentioned above, the machine tracks the current spindle tool, and the current independent (i.e. non spindle-mount) tool.

An M6 command specifying an independent tool (which must be done by specifying a virtual slot number) will temporarily override any spindle tool, even though the spindle tool is still physically mounted and possibly turning. To unmount the independent tool and return to the one which was mounted in the spindle, it is necessary to provide the same slot or ID number as the original spindle tool. The following examples illustrate.

```
m6 t1 g43 h1 (mount spindle tool 1, use offset)
m3 s10000 (spindle on)
... (do machining stuff)
m6 t96 g43 h96 (activate independent non-contact probe etc.)
... (do probing)
m6 t1 g43 h1 (go back to tool 1, de-activate probe)
... (more machining)
m6 t0 g49 (unmount tool, no offsets)
```

In the above, slot 1 contains a conventional tool, and virtual slot 96 is used to define an independent tool like a non-contact probe. The section labeled “do probing” would be using the independent tool. The spindle still retains T1 in this case, and it may even be turning if no M5 code was used. The advantage of this is that the tool changes will be very fast because no physical movement is required.

If retaining the spindle tool was undesirable, because of the danger of the spindle tool striking something, then it would be preferable to use something like the following:

```
m6 t1 g43 h1 (mount spindle tool 1, use offset)
m3 s10000 (spindle on)
... (do machining stuff)
m6 t0 (unmount spindle tool, spindle will turn off)
m6 t96 g43 h96 (activate independent non-contact probe etc.)
... (do probing)
m6 t1 g43 h1 (go back to tool 1, de-activate probe)
... (more machining)
m6 t0 g49 (unmount tool, no offsets)
```

The only difference is the explicit tool unmount on the 4th line.

Note that it is OK to select a different tool after the virtual slot instead of the spindle tool that was there previously. In this case, the controller remembers that T1 was mounted, so after using T96, if the next tool was T2, then the current spindle tool (T1) would be unmounted in the normal manner before mounting T2.

When a conventional spindle tool change (which actually changes to a new spindle tool) is complete:

- The spindle will be stopped.
- The tool that was selected (by a T word on the same line or on any line after the previous tool change) will be in the spindle. The T number is an integer giving the changer slot of the tool (or

its id).

- If the selected tool was not in the spindle before the tool change, the tool that was in the spindle (if there was one) will be in its changer slot.
- The linear (X, Y, and Z) axes will be positioned near the physical tool changer. Some machines may also change the rotary axis positions.
- No other changes will be made. For example, coolant will continue to flow during the tool change unless it has been turned off by an M9.

It is recommended to use the following code to reposition the tool roughly over the work, after completing a tool change:

```
m3 (turn on spindle, since m6 stops it)
g91 g30 z0 (safe Z)
g30 x0 y0 (move XY over rotary table center)
g90 (absolute mode)
```

It is OK to program a change to the tool already in the spindle. In fact, this is useful when using tool life management and/or tool pooling. The program can also use this to force the current tool to be checked for breakage.

If slot zero was last selected, the spindle will be empty (no tool mounted).

Coolant Control — M7, M8, M9

To turn mist coolant on, program M7.

To turn flood coolant on, program M8.

To turn all coolant off, program M9.

It is always OK to use any of these commands, regardless of what coolant is on or off.

Subroutine Call and Return – M97, M98, M99

The following control flow commands are implemented:

- M97 Pxxx
  makes the call to line with Nxxx value
- M97 Pxxx Lrrr
  makes the call to line with Nxxx value, rrr times (can use Q instead of L)
- M97 Rxxx
  goto line with Nxxx value
- M97 Rxxx Lrrr
  goto line with Nxxx value if rrr non-zero
- M98 Pxxx
  makes the call to line with Oxxx value
- M98 Pxxx Lrrr
  makes the call to line with Oxxx value, rrr times (can use Q instead of L)

- M98 Rxxx
  goto line with Oxxx value
- M98 Rxxx
  goto line with Oxxx value if rrr non-zero
- M99
  returns from subroutine call
- M99 Lrrr
  returns if rrr is non-zero.

M97 specifies an N word target, whereas M98 specifies an O word target. Otherwise, these codes work identically.

Using a P word for the target specifies a subroutine call, whereas an R word specifies a GOTO. Only one of these should be used. If both P and R are used, R takes precedence (i.e. the command will be a GOTO).

See the usage of the '/' (conditional line execution) prefix for an alternative means of conditionally executing code.

Most programs will use the traditional M98 P_ and M99 codes, which is recommended unless there is good reason to use any of the alternative constructs listed above. To call a g-code subroutine, program

M98 P_ L_ Q_

Where the P number refers to a line of code in the same program labeled with a matching 'O' (capital O) number.

The optional L or Q number is a repeat count. If omitted, it is equivalent to L1 i.e. a single call. Otherwise, the subroutine is called the specified number of times before proceeding with the program after the M98 line.

The L number may be 0, in which case the subroutine is not called at all.

Note: L and Q have the same meaning. It is recommended to use L. Q is for compatibility with some post processors. If both L and Q are specified, Q is ignored.

To code a subroutine, label the first line of the subroutine with an O word, and place an M99 at the end of the subroutine to return to the caller.

It is an error if:

- An M99 is encountered when there has been no call.
- An excessive call depth is encountered (too many M98 without a corresponding M99). This may happen if a subroutine calls itself, directly or indirectly.
- There is no line in the current program with an O number matching the P number in the M98 call.

Example:

```
M3 S1000 F500
M98 P101       (call the line with O101)
M98 P102 L5    (call the line with O102, 5 times)
```

```
M2

O101          (start of subroutine 101)
G53 G0 Z0
M99           (return from subroutine 101)

O102          (start of subroutine 102)
G1 X0 Y0 Z0
X10 Y10
X10 Y0
M99           (return from subroutine 102)
```

It is often useful to use a boolean expression to create a conditional subroutine call. The following example shows how to call an error handling subroutine if a probing error is detected:

```
(probe setup omitted)
m98 p1        (probe)
m98 p100  (check errors)
(msg,Probing OK)
m30

o1             (probe sphere)
... (probing code, leaves status code in #5060)
m99

o100      (check for probe error, end if so)
m98 p101 L[#5060<>0]  (check if probe result non-zero [error])
m99

o101
(msg,Probing error encountered)
m30
```

Note the use of conditional calling inside the O100 routine. It calls O101 only if there was an error. This works because the result of the "not equals" operator (<>) is 1 if true, or 0 if false. Thus, O101 is only called if the probe result in #5060 was not equal to zero.

In this example, subroutine O101 does not return, since it uses M30 to end the entire program.

Special Functions – M100 through M119

These codes are described in the chapter "Proprietary M-codes".

Other Input Codes

Set Feed Rate — F

To set the feed rate, program

```
F_
```
This provides a base feed rate. Use of the feed rate override control changes the actual feed rate as a proportion of this.

Set Spindle Speed — S

To set the speed in revolutions per minute (rpm) of the spindle, program

```
S_
```
The spindle will turn at that speed when it has been programmed to start turning. It is OK to program an S word whether the spindle is turning or not. If the speed override switch is enabled and not set at 100%, the speed will be different from what is programmed. It is OK to program S0; the spindle will not turn if that is done. It is an error if:

- the S number is negative.

Select Tool — T

To select a tool, program

```
T_
```
where the T number is the tool pod or carousel slot for the tool. The tool is not changed until an M6 is programmed (see Section 3.6.3). The T word may appear on the same line as the M6 or on a previous line. It is OK, but not normally useful, if T words appear on two or more lines with no tool change: only the most recent T word will take effect at the next tool change. It is OK to program T0; no tool will be selected. This is useful if you want the spindle to be empty after a tool change. It is an error if:

- a negative T number is used,
- a T number larger than the number of slots in the carousel is used, except that 96, 97, 98 and 99 have special meaning for MotusCNC.

T96 is used for manually mounting the secondary touch probe in the spindle; T97 is for the primary touch probe; T98 is for a miscellaneous manually mounted tool with no THS (Tool Height Setter) cycle; and T99 if for a manual tool with THS cycle.

Order of Execution

The order of execution of items on a line is critical to safe and effective machine operation. Items are executed in the order shown in the following table if they occur on the same line.

1. comment (includes message).
2. set feed rate mode (G93, G94 — inverse time or per minute).
3. set feed rate (F).
4. set spindle speed (S).
5. select tool (T).
6. change tool (M6).
7. spindle on or off (M3, M4, M5).

8. coolant on or off (M7, M8, M9).
9. enable or disable overrides (M48, M49).
10. dwell (G4).
11. set active plane (G17, G18, G19).
12. set length units (G20, G21).
13. cutter radius compensation on or off (G40, G41, G42)
14. cutter length compensation on or off (G43, G43.x, G49)
15. coordinate system selection (G54, G55, etc.; G54.4).
16. set distance mode (G90, G91).
17. set retract mode (G98, G99).
18. home (G28, G30) or change coordinate system data (G10) or set axis offsets (G52, G92, G92.1, G92.2, G94).
19. perform motion (G0 to G3, G80 to G89), as modified (possibly) by G53.
20. stop (M0, M1, M2, M30, M60) and subroutine call (M97, M98, M99).

Proprietary M-codes

This chapter describes proprietary M-codes used by MotusCNC.

M100 – Probing

NOTE: M100 is a legacy command. New applications should use G38 (c.f.). G38 supports the same set of operations as M100, but is more efficient and works better with 5-axis modes. The material in this section is still relevant to G38 probing.

M100 with P,Q,R words is used to start a probing operation. The controller supports a primary and secondary probe. Either probe may be configured to be either spindle mounted (like a normal tool) or independently mounted.

The primary probe uses tool slot 97, and the secondary probe uses tool slot 96. There must be a tool slot entry for these in the MotusCNC tool table.

Probe Usage

There is a fundamental distinction between spindle-mounted and independent probes. Spindle-mounted probes are treated as normal tools and are activated and deactivated using normal tool change (M6) commands. They behave like normal tools in most ways, except that the spindle is usually inhibited from turning. Independent probes are activated and deactivated using special M119 commands, and are independent of normal spindle tool changes. For example, an M6 may be used to change tools while an independent probe is activated.

For a spindle-mounted probe, the recommended order of operations in g-code is:

- Tool change M6 T97 (primary) or M6 T96 (secondary);
- G43 H97/96 to enable tool length offset;
- M100 (or G38) as many times as required;
- Tool change to next tool (M6 T*n* G43 H*n*).

If independently mounted:

- M119 P20 Q1 (primary) or M119 P20 Q2 (secondary) is used to activate the probe (e.g. to extend it into the active position);
- M100 (or G38) as many times as required;
- M119 P21 is used to deactivate the independent probe (e.g. retract it to a safe position).

Independent Probe Considerations

When an independent probe is activated, it takes priority over a spindle-mounted probe (if any). This means that the independent probe will be used for any M100 operations until such time as the independent probe is deactivated.

Since independent probes do not mount in the spindle, they must be defined as a virtual slot number. By convention, virtual slot number 96 is used for a side-mounted probe.

Typically, independent probes are offset from the spindle axis. This requires that X and Y displacements from the spindle axis must be measured and entered into the MotusCNC tool table. The following procedure outlines the steps that need to be taken to initially set up an independent probe.

- Ensure the virtual slot is loaded with the appropriate tool ID. By convention, the tool is defined with ID set to the virtual slot number, plus 1000. So the independent probe would be defined with ID 1096. In MotusCNC, drag the tool table entry over to virtual slot 96.

- Ensure the independent probe is powered up and that it can signal the appropriate controller input. Inputs can be monitored using the Diagnostic panel in MotusCNC. Jogging the Z axis up and down should be able to locate the trigger point by observing the relevant input state (usually input 3 or 4).

- Mount a normal spindle tool with a reasonably sharp point to use as a guide, and enable its offset e.g. enter M6 T1 G43 H1 in the MDI.

- Jog the spindle axis over a known point (e.g. a target mark on a horizontal plane).

- Zero the X, Y and Z DROs at this point, and ensure display units are set to mm (G21). THe precision with which this is done determines the resulting precision of the probe offsets.

- Jog X and Y so that the independent probe axis is over the same point. Jog the Z height to the trigger point for the probe.

- Note the X and Y DRO values. These values should be directly entered in the MotusCNC tool table X and Y offsets for the appropriate tool ID. You will need to press the edit button, then apply the changes after entering the X and Y offsets.

- It is recommended to also measure the probe TLO (Tool Length Offset) if this is reasonably repeatable. If this is set correctly (see next section) then it will not be necessary to perform a THS cycle every time the probe is activated.

Since an independent probe may be used in conjunction with a cutting tool, it is often required to intersperse probing and machining operations. The following example demonstrates one way of accomplishing this:

```
m6 t1 g43 h1        (mount cutting tool 1, use offsets)
... (machining)
(optional safer but slower: use m6 t0 here to unmount T1 while probe)
m6 t96 g43 h96
... (probing)
m6 t1 h43 h1
... (machining)
(etc.)
```

Probe Tool Length Offsetting

For any probe, the g-code should **always** use tool length compensation after mounting the probe: G43 H97 or G43 H96 as appropriate. Without G43, the Z height measurements will be referenced to an arbitrary level, and it will be difficult to relate future tool mounts to the probed Z levels.

When user coordinates are returned as probing results, the values will be transformed to be correct for the TLO of the tool whose offset is specified in the G43 H*n* which is in effect at the time. The result of this is that if, say, the M100 operation determines that the Z height of a plane is +5.5mm, then a g-code block with 'G1 Z5.5' will move the current tool controlled point to the actual level of the measured Z height, provided that the current tool is correctly offset using G43.

The TLO of the probe itself is such as to specify the offset to the center of the spherical probe tip, not the point of the tip which actually contacts the THS. Thus, the lowest part of the probe is actually one tip Z radius lower than the controlled point.

Probes may have X and Y offsets. These are manually added to the MotusCNC tool table and work analogously to the normal Z TLO. If, for example, the probe is 100mm to the right (+X) from the spindle axis, then the X offset is specified as -100.

Spindle-mounted probes use the normal THS cycle on tool change (M6 T97) in order to determine their TLO dynamically. The controller, when performing the THS cycle for a probe, looks for triggering of the probe itself, as well as the tool height setter – this handles the case where the probe is more sensitive than the THS, such as for non-contact type probes.

NOTE: it is recommended to have a touch probe which is stiff enough that the THS triggers, not the probe, when it is touched off. Then, the probe offset will be consistent with normal tool offsets in the case that the THS platform moves down slightly before triggering.

Independent probes are marked as such in the tool table, and there is no automatic THS cycle when activating an independent probe. Thus, the length offset must be calculated and entered manually for any non spindle-mount tool.

Starting a Probing Operation

M100 P,Q,R starts a probing operation. Since probing requires quite a large number of parameters, and can return a large amount of data back to MotusCNC, the majority of the data communication is passed via "hash variables" in the G-code program. In addition, information from the machine settings configuration (blocks 17 and 18) are used to provide default probing parameters.

Owing to the inability of G-code to deftly handle conditional execution (other than conditional subroutine execution) it is difficult to write probing routines. Nevertheless, it is possible to write G-code programs for simple probing. The following is an example which, for simplicity, omits error handling:

```
g21
m6 t97
g43 h97
(Probe centre of expected 18mm sphere)
(Jog to about 1mm above top pole of sphere)
m0

#5000=2   (Options : 2=halt g-code if error)
#5001=0   (0 = use configured settings)
#5005=60  (seconds overall timeout)
#5006=14  (type sphere=14)
#5007=5   (zhop)
#5008=1   (clear distance)
#5009=4   (stepover distance)
#5013=1800     (approach mm/min)
#5014=180 (backoff mm/min)
m100 p0 q-13 r9
(R=expected radius, P,Q = initial move out for equatorial probe)
g0z[#5023+13]
g0x[#5021]y[#5022]  (move 13mm above exact centre)
m2 (That's it, folks)
```

On completion of the M100 operation above, results are returned in hashvars 5021 etc. which, in the above example, are used to move the axes above the probe result position.

Probe Control Variables

Variable number	Description
5000	Options. This is a bitfield (composed by adding the following numbers, for each desired option): 1: Quiet. No user interaction if error. Program needs to check for errors. 2: Halt on error. G-code is halted if any error encountered. 4: Set up for point-cloud probing. This option suppresses probing for this M100 command, but saves the parameters for the following M101 (high-speed probe) command(s). See the M101/M102 section.
5001	Probe tip radius dimension units: 1 if inch, 2 if mm, 0 to use the default probe tip radius defined in the M118 settings (block 17/18). It is recommended to use 0

Variable number	Description
	here, since it works better with G43 and supports different radii in the polar and equatorial directions.
5002	Reserved
5003	Reserved
5004	Probe radius, defined in terms of units specified in #5001; ignored if #5001=0
5005	Overall timeout in seconds. Error flagged if this is exceeded.
5006	Probe operation type. See “Probe Operations” section.
5007	Z hop. Displacement in +Z when hopping over boss, ridge etc.
5008	Clearance in X,Y,Z - distance to move probe away from contacted surface when rapid repositioning
5009	stepover distance in X,Y (max amount moved from current X,Y to move into empty space, or max extra radial dist for finding boss/ridge)
5010,5011,5012	Probe indices for indexed computations. Non-negative values index from the oldest result in the PRS (Probe Result Stack), negative indices count back from the most recent.
5013	units/min probe approach rate (when no contact). If zero, uses settings value configured for this probe.
5014	units/min probe backoff rate (when contact). If zero, uses settings value configured for this probe.

Probe Result Variables

Variable number	Description
5020	Probe result status. See “Probe Result Status” table below
5021, 5022, 5023	Probed position (may be position at which probe hit, or corrected for radius, depending on op). These are in current units, including any offsets. Note that these coordinates are only valid for the current (probe) tool. If the probe has an X,Y offset, then the coordinates will need to be adjusted if used for a different tool, since most tools will have zero X,Y offsets. This is because of limitations with G43, since it can only specify a relative Z offset. To move a zero-offset tool to the correct X,Y location, the program would need to **add** the original probe X,Y offsets. Using G43 correctly will adjust Z automatically. When measuring a Z level, and using G43 H97 etc. (that is, using the tool length offset for the probe), the controlled point of the probe is the centre of the spherical tip, not the bottom of the probe tip which touches the THS. So, for a “raw” probe reading of a Z level, the actual surface is one tip radius less than the

Variable number	Description
	Z position stored here. If a normal cutting tool is mounted, and its TLO applied, then if moved to the Z coordinate stored here, that tool will end up one tip radius too high. For measurements other than raw readings, such as planar or spherical probe operations, the tip radius is already subtracted out so that the value stored here does not have to be further compensated.
5024, 5025, 5026	As above, in Tool Coordinate System (TCS) frame, in current units. The TCS frame is designed to be fixed relative to the machine table, regardless of probe offsets. This frame is defined so that at Z=0 the measured point is level with the THS (Tool Height Setter). X and Y correspond to the machine axis positions, with any probe X or Y offsets compensated out. The machine spindle axis will pass through this point. TCS is useful because it is independent of the current tool offset, provided that the tool offsets are always known by reference to the THS. The values are also consistent regardless of whether G43 is in effect.
5027	Computed radius of hole/boss/circle, or half slot/ridge width, or distance point to plane etc., in current units.
5028 thru 5039	Computed data specific to operation. If applicable, this will be in the TCS frame.

Probe Result Status

Hashvar #5020 returns a bitfield indicating any error(s) that occurred. If the value is zero after the M100, then no errors occurred.

Bit value	Description
0x0001	Probe did not contact one or more tested points
0x0002	Operation timed out
0x0004	Control input error (pad parameters)
0x0008	Communication error (could not get parameters etc.)
0x0010	Tool is not a probe (probably forgot to select the appropriate option in the tool table entry for this probe, or forgot to activate using e.g. M6T97)
0x0020	Aborted because of EStop
0x0040	Aborted because axis disabled
0x0080	Aborted because probe in contact at start
0x0100	Probe jammed in contacted position (failed to go off on backoff)
0x0200	Position not captured accurately (may be ok, depending on backoff speed)

Bit value	Description
0x0400	Approach or backoff speed unreasonably slow (probably forgot to set those vars)
0x0800	Probe encountered unexpected obstacle
0x1000	Initial probe direction ambiguous (zero or almost zero distance specified)
0x2000	Would exceed soft limit on X axis
0x4000	Would exceed soft limit on Y axis
0x8000	Would exceed soft limit on Z axis
0x10000	Missing data required for this probe type
0x20000	Computation cannot be performed because planes are parallel etc.
0x40000	Cannot perform this operation with Z-only probe
0x80000	Point cloud buffer filled
0x100000	Point cloud not initialized. Requires M102 to reset buffer, and a previous M100 probe to set the parameters for each subsequent M101 probe.
0x200000	"Probe valid" input indicated problem.
0x400000	Bad index used for indexed operation
0x800000	Bad result type probe for indexed operation

Probe Operations

Hashvar #5006 specifies the type of probing operation to perform, according to the following table. Parameters are passed via the P/Q/R words supplied on the M100 line, plus certain hashvars. Only those hashvar parameters that have different meanings for different probe operations are documented. Only hashvars parameters #5007 (Z-hop), #5008 (Clearance), #5009 (Stepover) are described.

Results are supplied in certain hashvars. Results in #5024-6 are not described here, since they are always the machine absolute coordinates corresponding to the result in #5021-3. Also, #5021 is described in the previous section.

FPP indicates the Final Probe Position after the M100 operation completes. All probes start from the current probe position, unless otherwise noted, so in many cases the operator is expected to have roughly positioned the probe to the appropriate location using jogging etc.

<table>
<tr><th>Opcode</th><th>Description</th></tr>
<tr><td>0</td><td>DEST: Linear probe to specified destination.
<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>P</td><td>End point of probe, X coordinate</td></tr>
<tr><td>Q</td><td>End point of probe, Y coordinate</td></tr>
<tr><td>R</td><td>End point of probe, Z coordinate</td></tr>
<tr><th>Result</th><th></th></tr>
<tr><td>#5021-3</td><td>Position where probe tripped, not corrected for tip radius.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Opcode</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Result</th><th></th></tr>
<tr><td>FPP</td><td>Backed off to just before the hit position.</td></tr>
<tr><td></td><td></td></tr>
</table>
Example:
<code>#5006=0</code>
<code>M100 P20 Q-5 R0</code>
Probe towards X=20, Y=-5, Z=0
</td></tr>
<tr><td>1</td><td>
DIR: Linear probe in specified direction.
<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>P</td><td>Distance to probe in X direction</td></tr>
<tr><td>Q</td><td>Distance to probe in Y direction</td></tr>
<tr><td>R</td><td>Distance to probe in Z direction</td></tr>
<tr><th>Result</th><th></th></tr>
<tr><td>#5021-3</td><td>Position where probe tripped, not corrected for tip radius.</td></tr>
<tr><td>FPP</td><td>Backed off to just before the hit position.</td></tr>
<tr><td></td><td></td></tr>
</table>
Example:
<code>#5006=1</code>
<code>M100 P-10 Q0 R0</code>
Probe in the negative X direction, for 10 units.
</td></tr>
<tr><td>2</td><td>
HOLE: Probe centre of vertical round hole. Current point should be inside the hole, preferably near the side of the hole at (P,Q) direction from the centre (for fastest results), although near the centre is not much slower. This works by finding perpendicular bisector of the initial trial chord, finding the diameter along this bisector, then again along the perp. bisector of that if necessary for accuracy. The radii in both directions are averaged to find the result.
<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>P</td><td>Initial direction X component</td></tr>
<tr><td>Q</td><td>Initial direction Y component</td></tr>
<tr><td>R</td><td>Expected hole radius. The maximum probing distance will be 2 times this (a full diameter).</td></tr>
<tr><th>Result</th><th></th></tr>
<tr><td>#5021-3</td><td>Hole center coordinates (Z will be starting Z)</td></tr>
<tr><td>FPP</td><td>An arbitrary point inside the hole (near the side).</td></tr>
<tr><td>#5027</td><td>Computed hole radius, accounting for tip radius.</td></tr>
</table>
Example:
</td></tr>
</table>

<table>
<tr><th>Opcode</th><th>Description</th></tr>
<tr><td></td><td><pre>#5006=2
M100 P-1 Q0 R50</pre>
Probe in the negative X direction, for at most 100 units, then find at least 3 other contact points in the hole.</td></tr>
<tr><td>3</td><td>BOSS: Probe centre of vertical round boss. Current point should be near the side of the boss at a Z level to hit it, in the (-P,-Q) direction from the centre. This works analogously to the above hole probe, except that the probe is hopped over the boss when crossing to the other side.
<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>P</td><td>Initial direction X component</td></tr>
<tr><td>Q</td><td>Initial direction Y component</td></tr>
<tr><td>R</td><td>Expected boss radius. The maximum probing distance will be 2 times this (a full diameter).</td></tr>
<tr><td>#5007</td><td>Z-hop: amount to raise Z so that the probe can cross over the boss without contacting it.</td></tr>
<tr><td>#5008</td><td>Clearance: amount to back off from side after contact before raising the probe for cross-over.</td></tr>
<tr><td>#5009</td><td>Stepover: this is added to the expected radius (R) and the tip radius, and the result multiplied by 2, to get the move distance when hopping over the boss.</td></tr>
<tr><th>Result</th><th></th></tr>
<tr><td>#5021-3</td><td>Boss center coordinates (Z will be starting Z)</td></tr>
<tr><td>FPP</td><td>An arbitrary point outside the boss (near the side).</td></tr>
<tr><td>#5027</td><td>Computed boss radius, accounting for tip radius.</td></tr>
</table>
Example:
<pre>#5006=3
#5007=10 (z hop)
#5009=1 (stepover)
#5004=2 (probe tip radius)
M100 P1 Q0 R9</pre>
Probe in the positive X direction, for at most (1+2+9)*2 = 24 units, then find at least 3 other contact points around the boss, hopping Z by 10 units when crossing over. The crossover distance is also 24 units.</td></tr>
<tr><td>4</td><td>RECT_XY: Probe rectangular extent of XY (top) plane. For planar probing, 3 points are probed to determine a plane (if both extents are non-zero), or 2 points (if only one extent is non-zero), or 1 point (if both extents are zero). In the latter cases, the plane is assumed to be perfectly axis-aligned in the non-probed direction(s).

The probe results for the plane are stored in #5028-31, accounting for the probe tip</td></tr>
</table>

<table>
<tr><th>Opcode</th><th>Description</th></tr>
<tr><td></td><td>
radius. #5028-30 is normalized normal vector, #5031 is displacement of plane from origin (machine units). Note that results are stored separately for RECT_XY, YZ, ZX so that later the 3 results can be combined to compute an intersection point. These results are not in inch or mm units, since they are expected to be used in subsequent operations. In fact, the plane results are stored in Hessian normal form.

In addition to the plane equation, the first contact point is stored in the normal 3-D result, however this result is not used by subsequent operations.

Initially, the probe tip should be positioned above some point on the plane, less than R units above the plane. The probe will find the Z touch at that position, then make a similar measurement at X displaced by P, then Y displaced by Q.
<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>P</td><td>X extent of rectangle</td></tr>
<tr><td>Q</td><td>Y extent of rectangle</td></tr>
<tr><td>R</td><td>Z probing extent (negative, to probe downwards)</td></tr>
<tr><td>#5008</td><td>Clearance: amount to back off from surface after contact before moving to next position during multiple point probe.</td></tr>
<tr><th>Result</th><th></th></tr>
<tr><td>#5021-3</td><td>Coordinate (in plane) of first contact point.</td></tr>
<tr><td>FPP</td><td>Above the last contact point.</td></tr>
<tr><td>#5028-31</td><td>Computed plane equation (Hessian normal form).</td></tr>
</table>
Example:
<pre>
#5006=4
M100 P50 Q20 R-5
</pre>
Probe 3 corners of a 50x20 rectangle, at most 5 units down from the current probe position.
</td></tr>
<tr><td>5</td><td>
RECT_YZ: Probe rectangular extent of YZ (side) plane. This works analogously to RECT_XY, except that probing is along the X axis and results are stored in different hashvars.
<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>P</td><td>X probing extent (negative to probe left, positive to probe right)</td></tr>
<tr><td>Q</td><td>Y extent of rectangle</td></tr>
<tr><td>R</td><td>Z extent of rectangle</td></tr>
<tr><td>#5008</td><td>Clearance: amount to back off from surface after contact before moving to next position during multiple point probe.</td></tr>
<tr><th>Result</th><th></th></tr>
<tr><td>#5021-3</td><td>Coordinate (in plane) of first contact point.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Opcode</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Result</th><th></th></tr>
<tr><td>FPP</td><td>Above the last contact point.</td></tr>
<tr><td>#5032-35</td><td>Computed plane equation (Hessian normal form).</td></tr>
</table>
Example:
<pre>#5006=5
M100 P5 Q20 R15</pre>
Probe 3 corners of a 20x15 rectangle, at most 5 units to the right of the current probe position.
</td></tr>
<tr><td>6</td><td>
RECT_ZX: Probe rectangular extent of ZX (front/rear) plane. This works analogously to RECT_XY, except that probing is along the Y axis and results are stored in different hashvars.
<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>P</td><td>X extent of rectangle</td></tr>
<tr><td>Q</td><td>Y probing extent (negative to probe to front, positive to probe to rear)</td></tr>
<tr><td>R</td><td>Z extent of rectangle</td></tr>
<tr><td>#5008</td><td>Clearance: amount to back off from surface after contact before moving to next position during multiple point probe.</td></tr>
<tr><th>Result</th><th></th></tr>
<tr><td>#5021-3</td><td>Coordinate (in plane) of first contact point.</td></tr>
<tr><td>FPP</td><td>Above the last contact point.</td></tr>
<tr><td>#5036-39</td><td>Computed plane equation (Hessian normal form).</td></tr>
</table>
Example:
<pre>#5006=6
M100 P50 Q5 R15</pre>
Probe 3 corners of a 25x15 rectangle, at most 5 units to the rear of the current probe position.
</td></tr>
<tr><td>7</td><td>
SLOT: Probe centre of a slot. Current point should be inside the slot. This is analogous to hole probing, except the probe is only performed once across the slot in the specified direction.

The result will only be accurate if the slot is oriented perpendicular to the specified probing direction.
<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>P</td><td>Initial direction X component</td></tr>
<tr><td>Q</td><td>Initial direction Y component</td></tr>
<tr><td>R</td><td>Expected slot half-width. The maximum probing distance will be 2 times this (a full slot width).</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Opcode</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Result</th><th></th></tr>
<tr><td>#5021-3</td><td>Slot center coordinates (Z will be starting Z)</td></tr>
<tr><td>FPP</td><td>An arbitrary point inside the slot along the specified direction (near the side).</td></tr>
<tr><td>#5027</td><td>Computed slot half-width, accounting for tip radius.</td></tr>
</table>
</td></tr>
<tr><td>8</td><td>RIDGE: Probe centre of a ridge. Current point should be near the side of the ridge at a Z level to hit it, in the (-P,-Q) direction from the centre. This works analogously to the boss probe, except that only one direction is probed.

The result will only be accurate if the ridge is oriented perpendicular to the specified probing direction.
<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>P</td><td>Initial direction X component</td></tr>
<tr><td>Q</td><td>Initial direction Y component</td></tr>
<tr><td>R</td><td>Expected ridge half-width.</td></tr>
<tr><td>#5007</td><td>Z-hop: amount to raise Z so that the probe can cross over the ridge without contacting it.</td></tr>
<tr><td>#5008</td><td>Clearance: amount to back off from side after contact before raising the probe for cross-over.</td></tr>
<tr><td>#5009</td><td>Stepover: this is added to the expected radius (R) and the tip radius, and the result multiplied by 2, to get the move distance when hopping over the boss.</td></tr>
<tr><th>Result</th><th></th></tr>
<tr><td>#5021-3</td><td>Ridge center coordinates (Z will be starting Z)</td></tr>
<tr><td>FPP</td><td>An arbitrary point outside the ridge (near the side).</td></tr>
<tr><td>#5027</td><td>Computed ridge half-width, accounting for tip radius.</td></tr>
</table>
</td></tr>
<tr><td>9</td><td>3PLANE: compute intersection of 3 most recent plane probes (RECT_XY,YZ,ZX). This does no actual probing, but merely sets the theoretical intersection point of the most recent planar probes. Must be one of each in the XY, YZ and ZX.
<table>
<tr><th>Result</th><th></th></tr>
<tr><td>#5021-3</td><td>Planar intersection point</td></tr>
<tr><td>FPP</td><td>No change from initial point</td></tr>
</table>
</td></tr>
<tr><td>10</td><td>Reserved</td></tr>
<tr><td>11</td><td>Reserved</td></tr>
<tr><td>12</td><td>Reserved</td></tr>
</table>

<table>
<tr><th>Opcode</th><th>Description</th></tr>
<tr><td>13</td><td>
CORNER: Combines RECT_XY/YZ/ZX and 3PLANE to give the coordinates of a parallelepiped corner. Start position is just inside top side of corner, above it (+z). PQR define probing extents in x,y,z directions. Starts as for rect_xy.

Initially, the probe tip should be positioned just inside corner, less than R units above the corner (over the material). The probe will initially work as for RECT_XY, then moves back to start, steps over in x direction, then down to 1 probe tip diameter below the previous probed Z position. Performs RECT_YZ, then moves back to start (top), steps over in Y, performs RECT_ZX. Finally, computes intersection point as per 3PLANE. Note: R should be -ve. Sign of P,Q determine whether to probe the SW, SE, NW or NE corner.
<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>P</td><td>X probing extent (negative to probe left, positive to probe right)</td></tr>
<tr><td>Q</td><td>Y probing extent (negative to probe to front, positive to probe to rear)</td></tr>
<tr><td>R</td><td>Z probing extent (negative, to probe downwards)</td></tr>
<tr><td>#5008</td><td>Clearance: amount to raise Z above the XY plane so that the probe can cross over without contacting it, and also clearance from sides.</td></tr>
<tr><td>#5009</td><td>Stepover: this is the movement in X or Y when the probe is being positioned to measure the YZ and ZX planes. Thus, the probe should be initially positioned close enough to the corner that the stepover distance is sufficient to move the probe so it is clear to move down in Z without contacting material.</td></tr>
<tr><th>Result</th><th></th></tr>
<tr><td>#5021-3</td><td>Result coordinate.</td></tr>
<tr><td>FPP</td><td>At the starting position (X,Y) and the clearance distance above the top surface (Z).</td></tr>
</table>
Example:
<pre>
#5006=13
#5008=1
#5009=10
M100 P-50 Q-20 R-10
</pre>
Probe 3 sides of a North-East workpiece corner (i.e. corner is at positive X and Y).
</td></tr>
<tr><td>14</td><td>
SPHERE: Probe from current position in -Z direction which should be approximately over the sphere's pole (defined here as the point with maximum Z coordinate). Probes the pole and 3 points around equator. P,Q specify the initial move out from the pole before moving down by R from the top, then probing to the equator in direction (-P,-Q). PQ is rotated by 120 deg for each equator probe. When moving to the next equatorial probe point, the Z axis is raised to the level found with the pole probe, before moving above and then down to the next equatorial start point.

Normally, PQ should be R+probe radius+allowance for a full sphere probe. Partial (top
</td></tr>
</table>

<table>
<tr><th>Opcode</th><th>Description</th></tr>
<tr><td></td><td>segment) probing may be performed by specifying R < true radius. The probe moves out from the pole touch, then down by R+tip radius, then back towards the centre. Provided the sphere is not hit in the outward or downward movement, the true sphere radius can be determined, albeit with diminished accuracy compared with a full equatorial probe.

The method of determining the centre point is to find the unique intersection point of the 3 bisector planes of each chordal segment probed.

This mode only works for probing a solid sphere, not the inside of a concave spherical surface.
<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>P</td><td>Initial outward move X component</td></tr>
<tr><td>Q</td><td>Initial outward move Y component</td></tr>
<tr><td>R</td><td>Expected sphere radius.</td></tr>
<tr><td>#5008</td><td>Clearance: amount to back off from surface before repositioning.</td></tr>
<tr><th>Result</th><th></th></tr>
<tr><td>#5021-3</td><td>Sphere center coordinates</td></tr>
<tr><td>FPP</td><td>An arbitrary point at the equatorial level, close to the sphere.</td></tr>
<tr><td>#5027</td><td>Computed sphere radius, accounting for tip radius.</td></tr>
</table>
Example:
#5006=14
M100 P0 Q-12 R9
Probe the top (pole) of the sphere, for a maximum 9 units down. When found, move out by 12 units, down by the expected radius, then probe back inwards until the equator is touched. Probe twice more at 120 degree intervals around the equator. Finally, do the math to compute the centre and radius.</td></tr>
<tr><td>15</td><td>CIRCLE: Compute centre, radius and normalized normal vector of a circle defined by 3 most recent point probes. Does no actual probing, but sets result parameters only. Probes used to define points must be either SPHERE, CORNER, 3PLANE or a previous CIRCLE, since these are the only probe types to compute a definite 3-D point. The normal direction is defined by right hand rule (p1-p0) x (p2-p1) where p0 is oldest point and p2 is most recent.
<table>
<tr><th>Result</th><th></th></tr>
<tr><td>#5021-3</td><td>Centre of circle</td></tr>
<tr><td>#5028-30</td><td>Normal vector of circle plane (normalized).</td></tr>
<tr><td>#5027</td><td>Radius of circle</td></tr>
<tr><td>FPP</td><td>No change from initial point</td></tr>
</table></td></tr>
</table>

Opcode	Description
18	SPHERE_HA: High-accuracy sphere. This is similar to SPHERE except it makes multiple redundant probes then uses a least-squares fit to provide improved statistical accuracy. This is preferred for critical calibration such as trunnion table alignment.
20	3PLANE_INDEXED: Similar to 3PLANE, except uses indexed entries from the Probe Result Stack instead of the most recent 3 plane probe results. The PRS contains the most recent 64 probe results, however it is recommended to reset the PRS (using M102 P3) before allowing this buffer to wrap around, otherwise indexing becomes difficult to comprehend. The indices to use are set in hashvars 5010, 5011 and 5012. These must all index the result of planar probes, otherwise an error is flagged. The index number is non-negative (0, 1, 2...) to index a probe result starting at the point where the PRS was last reset (M102 P3), or a negative number (-1, -2, ...) to index the most recent probe result going backward. The advantage of using indexed probe results is that it is more flexible and efficient, since the same probe need not be repeated in order to get the required results in the correct order.
21	CIRCLE_INDEXED: Similar to CIRCLE, except uses indexed probe results as described above. Since the circle is computed from three points, each indexed probe result must be a point result: any combination of CIRCLE, SPHERE, SPHERE_HA, CORNER or 3PLANE (or any indexed version of these) is allowed.

M101 – Point Cloud Probing

This command is similar to M100, except that it is optimized for rapid probing of multiple points using the same probe setup commands. The differences between M101 and M100 are:

- M101 does not read or write setup or result hashvars.
- M101 re-uses setup parameters from the most recent M100.
- M101 stores results in the “point cloud buffer” (PCB).
- M101 stores X,Y,Z results (in the tool coordinate system) for each probe operation which results in a point, and additionally stores the current rotary axis positions (A and B). Thus, some probing results (such as plane equations, circle/sphere radii, and probe types which depend on mathematical combination of previous probes) will not be meaningful.
- M101 stores the status code (which would be #5020 if normal M100 used) along with the above coordinates. Thus, G-code which uses the results can determine whether the result was valid.

The PCB is a large data storage area in the machine controller, which is manipulated (reset or read back) using M102. At least 100,000 results may be stored. If this is exceeded, an error is raised.

Although M101 does not allow changing the probing parameter hashvars, it still accepts values for the P,Q and R parameters, thus allowing a certain amount of control point-to-point.

Example Point Cloud Probing

```
m102 p2   (reset point cloud buffer)

#5000=6   (Options : 2=halt g-code if error + )
                 (4=dummy setup for cloud probing)
#5001=0   (0 = use configured settings)
#5005=10  (seconds overall timeout)
#5006=1   (type directional=1)
#5013=3000     (approach mm/min)
#5014=3000     (backoff mm/min)
m100 p0 q0 r1  (just setup parms, no actual probe here)

g0 x0 y0 z5
m101 p0 q0 r-10
x1 z5
m101 p0 q0 r-10
x2 z5
m101 p0 q0 r-10
x3 z5
m101 p0 q0 r-10
x0 y1 z5
m101 p0 q0 r-10
x1 z5
m101 p0 q0 r-10
x2 z5
m101 p0 q0 r-10
x3 z5
m101 p0 q0 r-10

(Done, now have 8 results in PCB)
m102 p1   (reset PCB for readback of results in order)

m102 p0 q123   (get first result into #123..#128)
...  (do something with it)
m102 p0 q222   (get 2nd result into #222-#227)
...  (etc.)

m2
```

M102 – Point Cloud Buffer Control

This command is used to manipulate the PCB and probe result stack.

P0 – Retrieve next result

M102 P0 Q*n*

Read the next result from the PCB and store in local hashvars.

n specifies the index of the first hashvar in which to store results. A total of 6 consecutive hashvars are updated. The hashvar indexed by *n* is set to the X coordinate, that indexed by *n*+1 is set to Y, and so on through Z, A, B. The hashvar indexed by *n*+5 is set to the status code of the original probing operation. If the status code is zero, then the original probe operation was successful. Otherwise, the program will need to handle the error.

Coordinates retrieved from the PCB are transformed using the offsets and units in effect at the most recent M102 P1 (read pointer reset). Points are actually stored in the PCB in tool coordinates, which are independent of the current g-code offsets. Thus, if the PCB is reset (M102 P1) and read again, but with (say) a different fixture offset in effect before the M102 P1 was issued, then the results stored in the hashvars will appear to be different, however they will correspond to the same machine control point location after accounting for the different fixture offset in effect.

If, on the other hand, fixture offsets are modified **after** the M102 P1, then the points retrieved will no longer correspond to the actual digitized positions. This may be useful if it is desired to, say, machine a copy of a digitized part at a different machine location.

P1 – Reset the Read Pointer

M102 P1 R*m*

After using successive M101 operations, when it is required to read back the results into the g-code program, then M102 P1 will reset the PCB read pointer so that it references the first result in the PCB.

m specifies how many points to skip from the initial data in the PCB. If zero or negative, start retrieving from the first available point.

This command also "locks in" the current units and program offsets, as described for M102 P0, so that retrieval of points from the PCB will result in coordinates at which the controlled point will coincide with the physical location of the original point location.

P2 – Reset and Clear the PCB

This command clears the PCB and gets it ready to start probing a new set of results. It is required to use this command before the first M101 executed by the controller after power-up. For predictable results, any digitizing g-code program should issue this command before any M101.

P3 – Reset the Probe Result Stack

The PRS is the small buffer which retains all probe results; this is the buffer which is accessed by some of the probe calculations such as computing the circle center of the last 3 probes. M102 P3 resets this buffer to empty. This is useful when used in conjunction with PRS indexing options.

M105 – Application Control

The M105 command permits G-code programs to control and query some application functionality. In general, M105 commands do not directly interact with the machine controller. M105 is used with a P_ word to specify the function, plus optional Q_ and R_ words and hashvars to specify parameters for the function.

NOTE: do not code any P_ word numbers other than those documented below, since there are some undocumented functions used for factory diagnostics and calibration. Some of these functions may be hazardous in that they bypass some controller safety checks.

Unless otherwise noted, M105 commands may operate asynchronously with respect to any machine motion. In particular, they may act ahead of time from their location in the program relative to surrounding motion commands. This is because the G-code interpreter looks several seconds ahead of the machine position for motion commands (G0, G1 etc.) but will execute the M105 as soon as it is seen. If it is desired to synchronize the M105 to motion commands, then place the M105 immediately after another M code, or after an M105 P0.

P0 – synchronize

This command may be used to synchronize any of the following M105 commands to machine position. See the above paragraph for an explanation.

Example:

```
(assume X is at 0 and machine stopped)
G1 X100  (start moving to X=100)
M105 P7 Q0 R500  (this command will run when the machine)
                 (is still near X=0, not when X=100)
M105 P0  (synchronize, i.e. wait for the above G1 move to)
         (complete and machine stopped)
M105 P7 Q0 R500  (this is guaranteed to run when X=100)
```

In the above example, it doesn't really matter exactly where the M105 P7 command is run, since this command retrieves tool data which is not changed by machine movement. Some other (future) M105 commands may, however, depend on actual machine position, so it will be necessary to consider synchronization with these commands.

P3 Q_ R_ - set tool X offset

P4 Q_ R_ - set tool Y offset

P5 Q_ R_ - set tool Z (length) offset

P6 Q_ R_ - set tool diameter

These commands set the specified dimension offset of tools slot Q, to the value specified by R. R is specified in current interpreter units (inch for G20, mm for G21).

Normally, tool length offsets are automatically set if a tool height setter is available. In special cases where this cannot be done (e.g. if the tool is a probe or unconventional in other ways) but a g-code program can be written to somehow measure these values, then this command is useful for fully automating tool offsetting.

Note that tools are always specified by slot number, not tool table ID. Thus, Q must refer to a valid real (or virtual) slot number. If the slot contains a valid tool table entry, that entry is updated.

For example, if tool slot 2 contains tool ID 1322, and the program contains

```
G21 (mm)
M105 P5 Q2 R12.7
```

then the tool in slot 2 will have its length offset set to 12.7mm. Furthermore, the tool table entry for tool ID 1322 will also be set to that length offset.

This will NOT modify the current program tool length offset if that tool slot happens to be in use via G43 H2. The new offset will only be used if there is a subsequent G43 H2.

P7 Q_ R_ - get tool information

This command reads the tool table and places the resulting information in a contiguous range of ten hashvars specified by the R word. Where applicable, the dimensional values are converted to current units (inch for G20, mm for G21).

Q specifies the tool slot number. If Q0, then assume the currently active tool from the last M6. If there is no tool defined for the specified slot, then all information will be set to zeros.

The hashvars are set as follows:

Hashvar	Description
r	X offset
r+1	Y offset
r+2	Z (length) offset
r+3	Diameter
r+4	Tip radius
r+5	Active location (see below)
r+6	Device type (see below)
r+7	1.0 if control point at center of tip (radius), else 0.0.
r+8	Reserved, set to 0.0
r+9	Reserved, set to 0.0

The active location code is one of:

Code	Description
0	Spindle mount: normal spindle tool on axis.
1	Head mount: not on spindle axis, but fixed on spindle head.
2	Table: on main (XY) table.
3	Work: fixed with respect to workpiece, including any rotary axes.
4	Other.

The device type code is one of:

Code	Description
0	Cutter: normal rotating cutter tool.
1	Probe: touch probe or other measurement device.
2	Camera: video camera or other optical inspection device.

Example:

```
M105 P7 Q2 R500
```

will retrieve the information associated with the tool in slot 2, writing the results to #500, #501, ... #509.

P8 Q_ - get current program position

This command reads the current program position and stores it to the 6 consecutive hashvars starting at the Q word. This is useful for coding subroutines which return the controlled point to the starting point before returning, but there are many other possible uses, particularly in conjunction with probing.

The six consecutive hashvars are set to X, Y, Z, then the rotary axis positions, which depend on the current 4- or 5-axis mode. In 3+2 axis mode, the 4th through 6th hashvars are set to (alpha, beta, 0). In G43.5 (TCPC type 2) mode they are set to (I,J,K). In G43.4 (TCPC type 1) mode they are set to (A,0,C).

Example:

```
M105 P8 Q600
M6 T3 G43 H3
G0 X#600 Y#601
Z#602
```

This reads the current position into #600, #601...#605. It then changes to tool T3. The G0 moves the tool back exactly to where it was before the tool change.

P10 Q_ R_ - set tool lifetime

P11 Q_ R_ - increment tool lifetime

These commands are used with tool life management. Each tool slot maintains a lifetime counter, as a percentage of total tool life from 0 to 100. It is up to the g-code programmer to insert M105 P10/11 commands to increase the counter when a tool has been “used” in a job. Any tool which reaches 100 (or more) is considered “retired”. Retired tools will not be used if there is a non-retired tool of the same tool ID available in a slot.

The Q word specifies the slot number of the tool to modify, with Q0 meaning the currently mounted tool. The R word specifies the lifetime or lifetime increment.

In MotusCNC, when the operator drags a tool from the tool table to a specific slot (specifying the tool changer slot loading), the tool slot is set to lifetime 0, i.e. brand new. The lifetime can also be reset by dragging the 'OK' icon over the tool slot.

M105 P10 sets the tool lifetime counter directly to the value specified by the R word, which would normally be in the range of 0 to 100.

M105 P11 increments the tool lifetime by the amount specified by the R word.

Example:

```
o100 (my drilling routine with lifetime management)
g83 X... Y...  R...  (canned drill cycle)
m105 p11 q0 r0.5 (total lifetime is 200 hits, hence increment by)
                 (100/200 = 0.5 each hit)
m99 (return)
```

P12 – save tool table

This command saves the tool table to permanent storage. This is mainly used in tool setting and calibration routines, to save the results.

P20 R_ - set X via current fixture offset

P21 R_ - set Y via current fixture offset

P22 R_ - set Z via current fixture offset

These commands have the same effect as the operator manually setting the DROs, thereby setting the current fixture offset (and not moving the machine). The current fixture offset which is in effect (G54, G55, ... G59.3, or G54.4 WSEC) will be modified such that the X, Y or Z program position will be equal to the given R word.

Example:

```
G54
M0 (usr,Jog machine to reference point)
M105 P20 R0
M105 P21 R0
M105 P22 R0
```

This prompts the operator to manually jog the tool tip to the "reference point", which is presumably the (0,0,0) location at which the part was generated by the CAD program. After the operator presses cycle start, the program location is forced to be (0,0,0) by the M105 commands, thus matching the program to the actual part offset.

This has the advantage that the operator does not need to remember to zero the DROs manually.

M118 – Machine Settings

Normally, machine settings are changed using the Interpreter/Machine Settings window (see that section of this manual). If necessary, the M118 command allows settings to be changed from within an NC program.

If machine settings are changed using M118, it is the user's responsibility to synchronize with

MotusCNC by selecting the "revert settings from controller" button in the machine settings window, Command/Options tab.

Up to 3 parameters (P, Q, R), plus various hashvars, are provided with M118 code to set various machine options. All of these options have "factory defaults", however it is possible to modify these and save to non-volatile memory.

The P word specifies the function code; Q and R, with specified hashvars, provide parameters.

Hashvars for Machine Settings

The hashvars used to store machine settings is the block starting at #4060 with several hundred entries (not all of which are used). These settings are documented in the following table.

NOTE: most of the information in the table below is provided in the Machine Settings window in MotusCNC. Selecting a row of the settings table will show the descriptive information. The information in the Machine Settings window is more likely to be up-to-date than the table below.

'cal' indicates that this setting is a calibration value, and is protected against overwrites when settings blocks are loaded from MotusCNC (unless the protection check box is unchecked).

'init' indicates the setting is only examined when the controller is initialized (boot up or reset).

'home' indicates the setting is only examined when homing axes.

Hash var	Block	Description	Default
Block 0 is for rotary axis kinematics parameters:			
4060-4062	0	cal X,Y,Z position of A axis reference point. Absolute machine steps, with Z relative to THS datum. This is an arbitrary point on the A axis, however by convention the point closest to the C axis is chosen.	-318140, -46660, -5756
4063-4065		cal A axis direction vector, normalized. Note that the default direction is opposite the X axis because the A axis rotation is opposite that for a normal right-hand-rule coordinate system.	-1,0,0
4066-4068		cal X,Y,Z position of C axis reference point, when A=0. Absolute machine steps, with Z relative to THS datum. This is an arbitrary point on the C axis, however by convention the point closest to the A axis is chosen.	-318140, -46691, -5756
4069-4071		cal C axis direction vector, normalized, when A=0. The zero point of A is chosen so that the C axis direction has no Y component when A=0.	0,0,1
4072-4074		cal Workpiece origin, when A=C=0. Absolute machine steps, with Z relative to THS datum. Default is the C axis reference point, which is normally selected to be near the intersection of the A and C axes. On the trunnion table, this is a physically accessible point about 27mm above the rotating table.	-318140, -46691, -5756
4075-4083		cal Workpiece alignment (rotation) matrix.	1,0,0, 0,1,0,

Hash var	Block	Description	Default
			0,0,1
4084		cal A axis offset. Kinematics uses 'alpha' axis. This is added to alpha to get the actual machine A axis value. Relative machine steps (millidegrees). This value allows for correction of slight C axis tilt if the A axis does not zero in a perfectly horizontal position.	0
4085		cal C axis offset. Kinematics uses 'beta' axis. This is added to beta to get the actual machine B axis value (which in kinematics terms is actually the C axis, since the axis points along Z when A=0). Relative machine steps (millidegrees).	0
Block 1 below groups the axis reffing settings:			
4100-4102	1	cal X,Y,Z index mark offset (machine steps of 1um) from limit switch. Setting this value correctly (to the value reported when reffing the axes) will speed up the homing cycle. Sign of value determines direction of offset from limit switch position. For example, if the axis has a limit switch at the negative end, then an index mark offset of -5000 indicates that the index mark is to be found further negative by 5mm from the limit switch position. In this case, the buffer space for this axis must be at least 6000.	-35200, -9700, -12500
4103-4015	1	X,Y,Z buffer space (1um steps). Specify distance from limit switch activation position to hard stop. Specify 0 or a positive value; negative values reserved. If non-zero, then axis can travel beyond the limit switch point by this amount (still leaving a safety margin). In this case, the limit switch does not automatically halt the axis if hit during operation, and the machine relies on soft limits. This also allows the index mark offset to be specified within the buffer zone.	0,0,0
4106	1	Index search bracket (1um steps) – default 1mm	1000
4107	1	Index spacing	50000
4108	1	Index search velocity (um/sec) – default 1mm/sec	1000
4109	1	Axis reffing options: 0x0001 – X axis soft limit in buffer space 0x0002 – Y axis soft limit in buffer space 0x0004 – Z axis soft limit in buffer space - above flags set ON if the soft limit is to be set to the buffer space limit; set OFF (default) if the soft limit is set to the limit switch position. If ON, then normal axis motion will include positions within the buffer space and the limit switch will be activated in those positions. Thus, the limit switch may be activated during normal operation, and does not automatically halt the axis. Otherwise, the default setting (OFF) will not activate the limit switch during normal operation, but there will be a reduction in the range of motion by the value in buffer_space.	0x0000

Hash var	Block	Description	Default
In the following, blocks 2-7, 'a' is the axis digit (1=X, 2=Y ... 6=C). There is a separate block for each axis. Defaults depend on axis; they are generally conservative settings that work with the standard mill/trunnion table, and assume servo motors. In general, these settings are read at init time, however they may also be accessed after homing, during tool change, and other times (unless otherwise noted).			
41a0	1+a	Rapid velocity in open loop mode, steps/sec	
41a1	1+a	Rapid velocity in closed loop mode, steps/sec	
41a2	1+a	Homing velocity, steps/sec. This should be a positive value regardless of whether positive or negative limit switch is used.	
41a3	1+a	Post-move after reffing to limit switch, steps. This should be a positive value regardless of whether positive or negative limit switch is used.	
41a4	1+a	Rapid acceleration in open loop mode, steps/sec/sec	
41a5	1+a	Rapid acceleration in closed loop mode, steps/sec/sec	
41a6	1+a	Rapid jerk, steps/sec/sec/sec	
41a7	1+a	Output gain.	
41a8	1+a	Axis motion range, um steps. Default set up for standard mill. Note that this range is used to set the soft limit of motion at the axis position opposite the limit switch end. It is defined relative to the point at which the limit switch de-activates. It does not include the axis buffer space (defined in the axis reffing block) if any.	
41a9	1+a	Axis flags: **WARNING: non-zero settings here may be dangerous**. They are intended for special machine setups with (e.g.) particular axes not installed. These flags are only examined when the controller is homed or init. 0x0001 – home This axis is homed "in-place" with no movement. This is for testing only. 0x0002 – home This axis is not physically present. It is left disabled and there is no attempt to home it. 0x0004 – home Use this axis in open-loop mode even if an encoder is detected. **This option will change the machine absolute zero position for this axis**, since the limit switch becomes the reference point rather than the encoder index pulse. Thus, settings which depend on absolute positions (such as the tool changer) may become invalidated, or will need re-calibration. 0x0008 – home Encoder has no index pulse. Homing does not attempt to zero the axis on the encoder index position. **This option will change the machine absolute zero position for this axis;** the zero will be at the limit switch position. 0x0010 – home This axis homes in the negative direction i.e. limit switch is at negative end of travel. If set, then the machine coordinates for this axis will be mostly positive values, since the machine zero is set at the limit	0

Hash var	Block	Description	Default
		switch point (or the nearest encoder index, if available). 0x0020 – home Limit switch polarity is inverted. Using this is not recommended since it voids the fail-safe feature (i.e. if limit switch has faulty connection to controller, it appears “off”). Currently, only the trunnion table A axis uses this setting. 0x0040 – home Zero after post-move. When homing an axis, the machine zero is normally set at the limit switch point (unless there is an encoder index pulse). If this flag is set, then the axis is zeroed at the post-move point. Currently, only the trunnion table A,B axes use this flag. 0x0080 – home Do not use hard limits. After using the limit switch for homing, the limit switch is ignored. By default, the A,B axes set this, since their limits are fairly forgiving (or allow infinite rotation for B) so only the soft limits are required.	
Block 8 is for tool changer settings:			
4170	8	cal Z position at which to pick up tool, steps absolute (Z_MIN)	-127700
4171	8	cal X position of tool slot #1, steps absolute	-500
4172	8	cal Y position of tool slot #1, steps absolute	-46700
4173	8	THS approach speed, steps/sec	25000
4174	8	THS backoff speed, steps/sec	500
4175	8	Tool changer pickup speed, steps/sec	12000
4176	8	Tool changer drop-off speed, steps/sec	16666
4177	8	THS reference tool length offset, relative steps. Note that the negative of this (-135mm for the default) will be the Z DRO reading when TLO is in effect and there are no other offsets, and the tool tip is at the THS touch-off height.	135000
4178	8	Air blast time for tool cleaning, seconds	0.75
4179	8	X position of tool height setter center, steps, as an offset from tool slot 1 X position.	17000
4180	8	Y position of tool height setter center, steps, as an offset from tool slot 1 Y position.	-47000
4181	8	Addition to Z_MIN where spindle is clear of tool shanks (Z_CLEAR)	25000
4182	8	Addition to Z_MIN where spindle with longest tool is clear of obstructions in the vicinity of the THS. (Z_SAFE_ENV)	65000
4183	8	Addition to Z_MIN to define minimum Z when testing over THS (THS_MIN)	10000
4184	8	Z position for start/end of tool change, as offset from positive Z axis limit.	-500

Hash var	Block	Description	Default
		(Z_SAFE). Default of -0.5mm leaves a bit of room to avoid triggering limit switch if slight overshoot by rapid up.	
Block 9 is for MPG settings:			
4185	9	Mask of axes to disable in manual mode	0x00
4186	9	Mask of axes to enable for jogging in manual mode	0x1F
4187	9	Default jogging mode at controller INIT: 0: Machine coordinate system 1: Workpiece coordinate system 2: Machine coordinates for XYZ, workpiece coordinates for rotary. These jogging modes may also be dynamically set using M118 P7.	0
4188	9	Jog shuttle smoothness in range 0.1..10. Smaller values give more accurate tracking of motion path w.r.t. MPG rotation; higher values try to even out velocity variations and produce smoother feed rates.	1.0
4189	9	Jog shuttle maximum relative feed rate. If less than 1, then jog shuttle mode feed rates will be limited to less than the feed rate commanded in the g-code. If more than 1, then it is possible to “fast forward” through the motion path. Use caution with fast-forward settings because accelerations are magnified and may cause following errors.	1.0
Block 10 is for general machine settings:			
4190	10	Bitfield: 0x0001 – Ignore air pressure check for IMT spindle 0x0002 – Ignore spindle ready error signal for IMT spindle 0x0004 – Ignore spindle overload error signal for IMT spindle 0x0008 – Ignore spindle over-temperature error signal for IMT spindle 0x0100 – Assume stepper trunnion table connected (normally auto detects via limit switch polarity). 0x1000 – More verbose console settings for MPG status 0x2000 – On Estop, rewind host g-code. 0x4000 – Jog shuttle mode: hold at end of motion buffer (if set, requires turning off shuttle mode in order to return control to the interpreter). 0x8000 – Jog shuttle mode: motion halted as rapidly as possible if axis selection button released. Otherwise, motion may continue for a short while to catch up to the position commanded by the MPG rotation. 0x10000 – Dummy machine. This flag used for testing controller board with no machine attached.	0
4191	10	Tool changer options: 0x0001: Check On Unmount – Check tool via THS before dropping off. 0x0004: Run tool changer in step-by-step mode (for calibration and testing) 0x0008: Tool changer is NOT installed. 0x0010: Do NOT issue G43Hn after M6 tool change. By default, the G43	0x0001

Hash var	Block	Description	Default
		is issued on the PC so that TLO is always maintained. This option prevents that and puts the onus on the operator to set the correct TLO compensation. 0x0020: Allow manual taper/collet open even when tool changer available. Normally, manual collet open is only allowed as part of M6T98/99 command. This option allows a dedicated button to open the collet any time the spindle is not turning. There is a software interlock to ensure that the spindle cannot be started when the collet is open. This mode is allowed when the controller is in test mode, regardless of this configuration option.	
4195	10	Tool changer bilateral tolerance for allowable TLO when re-checking. Relative steps. Default is +/- 0.1mm. This is relevant if the Check On Unmount option is active.	100
4196	10	Solenoid output number to use for when mist coolant is selected.	3
4197	10	Solenoid output number to use for spindle collet opening.	2
4198	10	Solenoid output number to use for tool cleaner air blast.	1
4199	10	GP input number for manual taper/collet open button (0 if not available).	0
4200	10	cal Controller serial number. The meaning of this number is vendor-specific, but it should be globally unique for each KFlop-based controller. Note that any files written on the PC by the controller will include this serial number in order to avoid problems if multiple controllers are attached. The default serial number (0) means unassigned. From firmware version 433q: A serial number is assigned in the factory using the undocumented M118 P99 command. Attempting to overwrite this setting using any other M118 command will not have any effect, since the serial number is stored in a special flash location.	0
4201	10	GP input number for tool height setter main contact. Default (0) indicates to use INPUT1 (for DM6-SCL) or use the dynamic I/O channel defined by the tool changer (for the MA360). This may be configured to a non-zero setting (1..2) if using a THS without a tool changer (manual machine). A non-zero value also overrides the dynamic selection for the MA360.	0
4202	10		
4203	10		
4204	10		
4205	10		
Blocks 11-16 are reserved for axis tuning parameters (20 vars each)			
42a0	10+a	Integral term coefficient for closed loop axes	0.007
42a1	10+a	Velocity feed-forward	

Hash var	Block	Description	Default
42a2	10+a	Acceleration feed-forward	
42a3.. 42a7	10+a	3rd stage IIR filter coefficients	
42a8	10+a	Max integrator windup	
42a9	10+a	Max error term	
42b0	10+a	Max following error	
42b1	10+a	Max control loop output	
42b2	10+a	Max velocity, low force mode	
42b3	10+a	Reduction factor for accel, jerk in low force mode	0.1
42b4	10+a	Reduction factor for windup, error term in LF mode	0.2
42b5	10+a	Max following error in LF mode	
42b6	10+a	Max control loop output in LF mode	
42b7	10+a	Deadband factor. If 1.0, then the deadband is set to +/- 1/2 of the linear scale resolution. This is optimum (to prevent hunting) provided the machine is in good condition. If there is some wear, hunting can occur because of increasing backlash. To prevent that, this value can be increased to suppress hunting.	1.0
42b8	10+a		
42b9	10+a		
Blocks 17,18 are for primary and secondary probe settings. If either probe is spindle-mounted, then it must be mounted using M6 T97 (primary) or M6 T96 (secondary) before using M100. Otherwise, if the probe is independently mounted, then M119 P20 Q1 (primary) or M119 P20 Q2 (secondary) must be used before M100. Default input is 3 for both primary and secondary, since it is unlikely that both would be used simultaneously.			
4330	17	Probe tip Z effective radius, machine steps. The Z radius is used to specify the position of the centre of the probe when it is touched off against the THS. The effective TLO of the probe is reduced by this amount, so that the “controlled point” of the probe is at the centre of its spherical tip, not the bottom of the tip which touches the THS. Normally, this value will be the same as the following, although it can also be used to compensate for different probe stiffness in the vertical compared with the horizontal direction.	0
4331	17	Probe tip XY effective radius, machine steps. The XY radius is what is generally assumed to be the spherical tip radius when making measurements, except when measurements are straight vertical.	0

Hash var	Block	Description	Default
4332	17	cal Probe axis X offset from spindle axis, machine steps. If probe axis is to the right of the spindle axis (i.e. in the +X direction) then this value is positive.	0
4333	17	cal Probe axis Y offset from spindle axis, machine steps. If probe axis is to the rear of the spindle axis (i.e. in the +Y direction) then this value is positive. Currently, X and Y offsets need to be separately programmed into this setting and the CNC application tool table. The signs of the values are reversed in the tool table. For example, if the offset in this setting is +52000, then the corresponding tool table offset (for slots 97/96) should be -52.0 (mm, or the equivalent in inches if using imperial units).	0
4334	17	Probe options: 0x0001 – do not soft EStop if probe triggered unexpectedly 0x0002 – capture position at trigger point (else backoff point) 0x0004 – this probe is independently mounted (else is a spindle tool). Independent probes are controlled using M119 instead of M6. 0x0008 – this probe is for Z measurement only (else is 3-D). All probing operations may be initiated with a Z-only probe, however they will fail with an error code if there is any X or Y movement while actually probing. Probing an XY plane will work, since all scanning is vertical down. 0x0010 – this probe is a non-contact type, else is a touch probe. Often used with 'do not soft EStop' option, since this type of probe is often used without regard to potential “crashes”. 0x0020 – this probe ok for use with initial “contact”. Normally, a probe operation will fail if there is initial contact, however this flag permits the probe to move opposite the commanded direction until not in contact, whereupon the normal probe will start. 0x0040 – this probe has an “invalidity” signal on the auxiliary input 'A' of the main input (e.g. INPUT3A if INPUT3 is the main probe input). If set, then the input is used to check that the main probe signal is valid during probing operations. It does not trigger EStops, but will abort any probing operation with a specific error code. This setting is only relevant if inputs 1, 3 or 4 are used (also input 2 on the MA360). 0x0080 – invert the above probe invalidity signal. If zero, then the probe signal is invalid if the invalidity signal is 1.	0x0000
4335	17	Input number to use (1-4). If zero, then input 3 is used as a hard-coded default. NOTE: input 1 is hard-coded to be used for the THS, input 2 is for lathe spindle index. Best to use inputs 3 or 4 for probing.	3
4336	17	Approach speed, steps/sec	10000
4337	17	Backoff speed, steps/sec	3000

Hash var	Block	Description	Default
4338	17	cal Effective Z offset for this probe, if independently mounted, absolute machine steps. This is the value that the machine Z axis would obtain at the point of touch-off of the probe against the THS, assuming that there was a physical tip with the specified Z radius. M119 is used to control use of this setting.	
4340-4349	18	Secondary probe settings, as above	
Block 19 is for spindle options. These are only read at INIT time.			
4350	19	Spindle variant installed: 0: IMT 1: Varicon/Hanning Note that there is a jumper setting on the controller which must be set appropriately. If the jumper setting mismatches this setting, then an error is flagged.	0
4351	19	Max RPM. Spindle speed is controlled via a PWM DAC. At maximum duty cycle, the DAC puts out about 10V to the spindle, which is expected to then run at the speed configured here. There is assumed to be a linear relation between the analog output and spindle speed. In particular, spindle speed should be 0 at zero output.	60000
4352	19	Multiplicative factor to convert raw ADC reading to spindle RPM. This is used when the spindle provides speed feedback via an analog signal. If zero, then there is no analog speed feedback. Feedback is assumed to be linear, with zero intercept. The hardware will convert an input voltage of +10V to an ADC count of 20000.	
4353	19	Index pulses per revolution. This is used when spindle provides speed feedback via a simple index pulse. If zero, then the index pulse is ignored.	
4354	19	Encoder counts per revolution. This is used when spindle provides speed feedback via a quadrature encoder (wired to the INPUT2 connector on the DM6-SCL, or a configured encoder input on the MA360). If zero, then the quadrature encoder is ignored. Note that out of the above 3 methods of determining actual spindle speed, if more than one is non-zero, then they are used in following order of priority: 1. Encoder 2. Index pulses 3. Analog	
4355	19	Multiplicative factor to convert raw ADC reading to spindle load percentage. This is used when the spindle provides load feedback via an analog signal. If zero, then there is no analog load feedback. The hardware will convert an input voltage of +10V to an ADC count of 20000.	

Hash var	Block	Description	Default
4356	19	Minimum spindle speed allowed (when not off) - RPM	0
4357	19	Maximum spindle speed allowed – RPM. If zero, then uses value of setting #4351. This may be used to lower the maximum speed in the case that a heavy or unbalanced tool is in use. M3/M4/S-word check these settings and post an error on the CNC application if the requested speed is outside the bounds.	0
4358	19	Spindle options as follows: 0x0001 – Unidirectional (CW only) 0x0002 – Check speed feedback for speed reached when starting 0x0004 – Check Speed OK input when starting If neither check speed option is specified, uses a fixed wind-up delay. If both check speed options are set, the Speed OK input is assumed to be definitive, and speed feedback is ignored. Speed feedback check uses settings 4362/3 to determine an allowable speed tolerance.	0x0000
4359	19	Wind-up delay when starting, seconds. This is the time interval from stopped to max hardware speed. Spindle acceleration is assumed to be linear throughout this range.	6
4360	19	Wind-down delay when stopping, seconds. This is the time interval from max hardware speed to stopped. Spindle deceleration is assumed to be linear throughout this range.	6
4361	19	Overall timeout when starting spindle, seconds.	15
4362	19	Allowable bilateral speed tolerance, as a percentage	10
4363	19	Allowable bilateral RPM difference. Out of these two tolerances, the more permissive of the two is used at any given target speed.	100
Block 20 is read-only to the CNC application. It is intended to allow g-code programs to learn various machine settings, such as soft limits for axes, and steps per mm. These values are filled in at INIT (or axis reffing completion), so the values should be retrieved by M118 P0 Q20 only after axes are reffed.			
4370-4374	20	Positive soft limits for X,Y,Z,A,B. Absolute machine steps.	
4375-4379	20	Negative soft limits for X,Y,Z,A,B. Absolute machine steps.	
4380	20	Machine steps per mm.	1000
4381	20	Machine steps per degree.	1000

P0 – Load hashvars with current machine settings

This command retrieves current machine settings and saves them in G-code hashvars. This is useful before modifying any settings, since it will synchronize all settings between MotusCNC (G-code) and the machine.

- Q99 – all settings from hashvars;
- Q<block number> - just the hashvar settings in the specified block number (0-21);
- Q<hashvar number> - just the one hashvar setting (hashvar numbers are all over 4000);

P1 – Push hashvars to selected machine settings

This command updates selected machine settings based on current hashvar values in the CNC application. The Q parameter specifies which settings to update, and from where:

- Q99 – all settings from hashvars;
- Q<block number> - just the hashvar settings in the specified block number;
- Q<hashvar number> - just the one hashvar setting (hashvar numbers are all over 4000).

Certain settings ("calibration values") will not be updated by this command unless the R word is provided, and set to the value 99. This is to protect calibration values in the normal course of updating machine settings.

P2 – Update selected machine setting

This updates a machine setting without actually using hashvars. The parameter is passed directly in the R word.

Q<hashvar number> R<new value>

For example:

```
M118 P2 Q4177 R134000
```

would update the THS reference tool length offset (hashvar number 4177) with the value 134000 (134mm). Note that this command allows changing calibration values.

P3 – Print Setting

This prints the current value of the selected setting to the console:

```
M118 P3 Q4100
```

will print the value of hashvar 4100.

P4 – Save Settings to PC File (.var format)

This prints the value of all settings to a file on the PC, called

```
settings-######.var
```

where '######' is the serial number stored in hashvar 4200. The serial number will be 0 for an unassigned controller. Note that the serial number is normally read-only, so it will not be overwritten by any value stored in the .var file. Thus, the same .var file can be used to update multiple controllers.

The format of this file is decimal numbers in two columns. The first column is for the hashvar number (4060, 4061 etc. up to 4359) and the second column is the current value of that setting.

P5 – Save Settings to PC File (.ngc format)

This prints the value of all settings to a file on the PC, called

```
settings-######.ngc
```

where '######' is the serial number stored in hashvar 4200. The serial number will be 0 for an unassigned controller.

This command is similar to M118 P4, except that the file is written as a g-code file which can be directly executed by MotusCNC in order to update the entire controller configuration which was saved.

P6 – Activate Kinematics Settings

When block 0 (kinematics) are sent to the controller via M118 P1 Q0, the new values will not be used immediately unless this command is sent. M118 P6 will use the new settings to initialize forward and inverse kinematics computations, and update the current workpiece coordinates. Jogging in workpiece mode will use the new settings.

P7 – Set Jogging Mode

M118 P7 Q*n* R*m*

sets the current MPG jog mode to *n*, as follows:

- 0: Machine axis. This is direct motion of the single selected machine axis (X,Y,Z,A,B). This is the most intuitive mode, and works the same whether or not 5-axis kinematics are defined for the trunnion table etc.
- 1: Workpiece coordinate system. This requires a properly aligned trunnion table, with a defined workpiece coordinate system defined. The axes are moved with reference to the current workpiece orientation. If the workpiece is rotated (i.e. A != 0 or C != 0) then X, Y and Z will not generally align with the machine axes, which may be surprising to the operator.

  Jogging the A (4th) axis on the MPG changes the tool tilt (“altitude”) relative to the workpiece XY plane. The allowable range is from approximately -30 degrees to +90 degrees if the workpiece XY plane is parallel to the physical table plane (which is the default given an identity rotation matrix). 90 degrees makes the tool perpendicular to the XY workpiece plane, in the +Z direction. Jogging past these ranges depends on physical limits to rotation. The controller ensures that physical limits are respected.

  Jogging B (5th) axis on the MPG changes the tool angle w.r.t. the workpiece ZX plane, which can be though of as the “azimuth” of the tool axis about the workpiece Z axis. This angle ranges from 0 (tool axis in ZX plane) thru 360 degrees, rotating counterclockwise when looking down from the positive workpiece Z axis.
- 2: Mixed workpiece (A,C) and machine (X,Y,Z). This mode will move the linear axes in

the normal sense (i.e. disregarding workpiece orientation), but will change the tool axis orientation w.r.t. the workpiece if the rotary axes are selected.

Modes other than 0 require properly defined 5-axis kinematics:

- The trunnion table must be installed, referenced, and the kinematics parameters (block 0) must be set appropriately.
- The current tool length offset must be known with reference to the THS datum. This is done automatically if the current tool is mounted via M6.
- If the current tool is unknown, then an arbitrary value of 0 is used (as if M118 P8 Q0 was commanded). Since Z=0 is at the top of Z axis movement, this value represents a "very long" tool which would touch the THS if the Z axis was near its maximum height. This might create exaggerated linear axis movement in certain jogging modes, and will probably exceed Z axis soft limits. It is, however, a reasonably safe value which will keep the Z axis far from the machine table.

The R word sets the rotary jog options to m. Current values are:

- 0: normal mode where MPG axes 4 and 5 map to the primary and secondary rotary axes (A,B).
- 1: mapped rotary, useful when the MPG does not have a selector for rotary axes. This mode maps MPG axes X,Y,Z to rotary axes A,B,C.

P8 – Set Tool Length Offset

M118 P8 Q*z* R*r*

sets the current tool length offset (which must be known for 5-axis kinematics to work). The value *z* is defined to be the machine Z axis position **in absolute machine steps of 1um** that would obtain if the tool was touched off against the THS datum plane. This is normally a negative value for reasonable length tools. Shorter tools have more negative values than longer tools.

The *r* value is either 0 or 1. If 0, the z value is absolute. Otherwise, it is relative to the current offset. Relative setting can be used to temporarily change the location of the "controlled point" of a tool e.g. to move it to the center of a ball-nose tool instead of the tip.

Be careful setting this value. In absolute mode, it should only be set when the THS cannot be used.

P9 – Save All Settings to Flash

M118 P9

saves all current settings to a special flash memory location. Using this command ensures that any changed settings will be preserved over power cycles.

M119 – Tool Changer and Miscellaneous Functions

Up to 3 parameters are provided with M119 code (P, Q, R) to perform various tool changer functions manually. **This is only used for calibration and manual recovery etc. M6 is used for normal tool changes. <u>Some of these commands may be dangerous</u> since the normal tool change logic is overridden.**

M119 is also used to control some special probe setup, especially for independent (non-spindle mounted) probes. It also performs rotary axis 'unwind'.

Parameters are used as follows:

- P - function code
- Q - parameter
- R - parameter 2

P0 close spindle collet

This will set the "current tool" to "unknown" (-1), unless Q0.

Q0 No THS cycle, declare that the spindle is empty (no tool, T0).

Q1 Perform THS setting of a temporary ("unknown") tool.

Q2 Unknown tool, but skip THS cycle.

Note that the normal (and safer) way to manually mount tools which are not in the tool changer, or which require a different size collet etc., is by specifying these flags in the application tool table then using normal M6 Tn tool change command.

The Q1 option performs a THS cycle. This updates the CNC application tool table for the current T number, which should be 99 for an unknown (manual) tool. Thus, the g-code should follow this with G43 H99 to use the measured TLO. Generally, it is preferable to use M6 T99 rather than this option.

P1 open spindle collet

Without Q word, opens the spindle collet and sets the current tool to "unknown".

Caution: when using this, the tool might drop out and get damaged. Most manual tool changes should use M6 T99 which waits for the operator to select 'X' on the MPG in order to open the collet.

Q0 Set "current tool" to explicitly "none" (0). "None" means that the drop-tool cycle is omitted since the spindle is known to be empty. Be cautious when specifying this, since if there is really a tool in the spindle, then it might get damaged because it will not be dropped off before the next tool is attempted to be picked up.

This command turns off and inhibits the spindle. The spindle remains inhibited, so this command should be followed by M119 P11 Q0 to re-enable the spindle.

P2 tool cleaner air blast

Q<t> - timer (seconds) for air blast, in range 0.1-10

e.g. "M119 P2 Q1.3" gives 1.3 second air blast

P3 manual calibration of tool slot position

Q<1..8> specify tool slot position

These move the spindle over the selected slot, opens the tool cover, then require the operator to use the

MPG etc. to jog the spindle in Z. The action is confirmed by selecting axis 4 (or cancelled by selecting axis 6) on the MPG and pressing the enable button. Alternatively, since a dialog box is posted, OK or CANCEL can be pressed on the PC side.

The purpose of this is to allow the tool changer to be “floated” into the position expected by the tool changer code, using the current tool changer position settings.

NOTES:

- The expected position of tool slot 1 must be configured in the controller settings. Other positions are derived from this using the known spacing. Most of the time it is best to use Q1, since slot 1 is the reference position.
- If OK is selected, then the Z height at that point is saved in the machine settings (#4170 – Z_MIN). The updated settings should be flashed to retain this value over power cycles.
- Because this sets Z_MIN, it is important that the reference tool used to float the tool changer is mounted normally in the spindle, not projecting too far, otherwise all tools will be picked up with this amount of overhang.
- All other tool changer Z (height) settings are specified relative to this value, except Z_SAFE. As implied by the name, Z_MIN is the lowest Z height that the spindle will move to during a tool change operation.
- After OK or CANCEL, the spindle is moved up to safe Z. Since the collet will be closed, the reference tool will remain mounted.

This command turns off and inhibits the spindle. The spindle remains inhibited, so this command should be followed by M119 P11 Q0 to re-enable the spindle.

P4 stealth tool change

This command is for testing only and may be hazardous in normal situations.

M119 P4 will traverse Z up to maximum height, then the operator may perform a manual tool change by using the MPG to control the collet etc. On OK or CANCEL, the operation is terminated.

Unlike a normal tool change, this is “stealthy” in that it does not change the controller's current tool state. Thus, it may be hazardous since the current tool offset will be wrong, and the controller's knowledge of the tool state will most likely not match reality.

This command turns off and inhibits the spindle because the collet may be opened. The spindle remains inhibited, so this command should be followed by M119 P9 to re-enable the spindle.

P5 manually specify tool which is in spindle, by slot

Q<1..8> tool in spindle belongs to this slot (which must be empty, because next M6 will attempt to drop this tool off in that slot.) The max slot number is 8 or 10 depending on the tool changer.

Q0 there is no tool in the spindle. Note that T0M6 is also allowed when the tool is currently unknown, which will have the same effect as M119P5Q0. Spindle must really be empty, otherwise there will be a crash on next tool change and possibility of spindle damage.

Q<90..99> declare virtual slot in use, for a spindle-mount tool.

Note that **this can be dangerous**, because it overrides the normal M6 process checks. If M119P5 does not reflect the true state of tool mounting, then crashes will occur which break tools and damage the spindle.

P6 close tool changer cover

P7 open tool changer cover

P9 close collet and resume spindle (it is forced off by any of the above).

If the spindle is inhibited because the collet is (or was) open, then this command ensures that the collet is closed, and allows the spindle to resume (unless it is inhibited by the following command, which is designed to prevent the spindle turning for certain tools such as probes).

P11 set spindle inhibit for current tool

Q0 Inhibit off

Q1 Inhibit on.

Spindle is automatically inhibited for probes, and changes to other tools or empty spindle will enable the spindle. Thus, M119P11 should be used after M6 to change the default behavior.

Note that this uses the "probe inhibit" flag, which is distinct from the "collet open inhibit" flag which is manipulated by M119 P9 and other commands, such as M6 tool change.

P12 manual THS XY calibration

Q0 Do not set the THS Z datum

Q1 Set the THS Z datum as well.

This command is used with some tool changer hardware (currently, the HSK-10 tool changer) to define the position of the Tool Height Setter. Running this command will prompt the user for required action, which basically requires manually jogging to center the spindle over the THS. Optionally, the Z datum can be set as well: the captured position defines the "zero tool length".

P16-19 tool change check flags

These commands set or clear the following tool check flags:

- M119 P16: clear flags
- M119 P17: force a TLO check even if the same tool is being called out on the M6
- M119 P18: quiet mode; no operator prompt, just abort program on host if error in tool change.
- M119 P19: both of the above flags.

The P17 flag is useful if it is desired to check a tool for breakage at regular intervals. If this flag is set, then an M6 command for the same tool will run the tool over to the tool height setter and perform a measurement cycle. Otherwise, the normal action is to ignore changes to the same slot number.

If the THS cycle indicates the tool is out of tolerance, then the P18 flag determines the action to take: either the operator is prompted, or the program is automatically halted. The latter case is useful for application plug-ins which can recover from tooling errors.

P22 update tool table Z offset

M119 P22 Q*n* R*m*

This code sets the Z offset entry for tool slot *n* in the CNC application's tool table. If *m* is zero, then the current machine Z coordinate is assumed to be such that the tool would be just touching the THS (datum plane) at this height. Other values of *m* introduce an additional offset. For example, if the tool would be touching off on a "virtual" THS which was 10mm lower than the standard THS, then

R10000

would compensate for this difference. *m* is specified in machine steps. The R value is *positive* when Z is *lower* than the height it would have if touching off at the THS level.

The purpose of this command is to allow TLO to be set for tools which cannot be physically touched off against the THS for some reason (e.g. because the tool had a large X offset which prevents the X axis from being able to position the tool over the THS). This is most often used for independent probes.

P23,P24,P25 update tool table X,Y,Z offsets, absolute

M119 P23 Q*n* R*m*

M119 P24 Q*n* R*m*

M119 P25 Q*n* R*m*

Update slot *n* in the MotusCNC tool table, with the specified X, Y or Z offset (*m*). The offsets are independent of the current axis positions.

NOTE: the units of *m* are machine steps, which are converted to inch or mm when stored in the tool table, according to the current units mode (G20/21).

P27 set tool breakage/slippage detection flags

Q0 Prompt operator on breakage (gets 'press OK to continue') - default

Q1 Bypass operator prompt if breakage.

Q2 Prompt operator on slippage, and require manual tool drop-off - default

Q3 Bypass prompt on slippage - not recommended, since may have problems dropping off tool.

Q4 Prompt operator on slippage, but with normal tool drop-off (no open/close collet step).

In any case, the application is notified of the slot with broken/slipped tool. The prompt bypass options are useful for some programs that can automatically select new tools.

P30 Unwind B axis

M119 P30 will unwind the B axis. This means that a multiple of 360 degrees is added to the current B axis position so that the final position is within +/-180 degrees of zero. It does this without actually moving the axis.

It is recommended to place this at the beginning of a job so that the B axis does not unwind on the first G0 or G1 with a B word. Also, place at the end so that the next job does not wait for a physical unwind if it omits this command.

P31 Qn Unwind B axis to specified angle

This is a generalization of M119 P30. An integer multiple of 360 degrees is added to the current B axis position so that the final position is within +/-180 degrees of the specified Q number. It does this without actually moving the axis.

It can greatly improve program speed to use this command. For example, if the first B axis motion in the program is to the destination 10740 degrees, the following command will prevent more than 180 degrees of start-up rotation no matter the initial B axis position:

M119 P31 Q10740

Without the M119 command, if the B axis was near zero it would first rotate about 30 full turns.

P32 Qn   Force B axis to specified position

This forces the B axis to have the specified angular position, without moving the axis. This should not be used except as a last resort to recover correct angular positioning where position has been lost somehow (for example, a stalled stepper motor).

Use great caution with this command, since it defeats soft limit protection.

P34, P35 Qn, P36 Qn Unwind A axis.

These commands are analogous to M119 P30, M119 P31 and M119 P32, except they apply to the A axis. This is useful for 4th axis rotary tables which have unlimited A angle travel.

Do not use this on the trunnion table or any other rotary mechanism with a limited A axis travel, otherwise the software may halt with a soft limit error, or there may be a hardware crash.

Examples of use of some M119 commands:

Floating tool changer when commissioning new machine:

```
M6 T98   (Mount reference tool manually, skip THS cycle)
(Use MPG X select to open collet.  MPG A = OK, MPG C = cancel)
M119 P7   (Open cover)
M119 P3 Q1  (Moves spindle over slot 1, allows Z jog)
(Operator jogs down in Z, and tightens down tool changer when)
(reference tool is properly seated in slot 1.  Machinist's square)
(must be used to align the tool changer)
(MPG A = OK, MPG C = cancel)
```

```
M119 P6   (Close cover)
M119 P9   (Resume spindle)
M6 T0     (Unmount reference tool manually)
M2
```

Since the M119 P3 Q1 saves the Z position in the machine settings (as the appropriate height for pick-up and drop-off of tools), the settings need to be flashed to become permanent.

Setting step-by-step mode for testing out new tool changer algorithms etc.

```
M118 P2 Q4191 R5  (R=bitmask: 1=check THS on umount, 4=step by step)
M6 T1     (Try it out: MPG X advances to next step)
M118 P2 Q4191 R1  (Put it back to default setting)
M2
```

5-Axis Kinematics

General Information

Trunnion Table

The trunnion table is supported for simultaneous 5-axis machining. The rotary axes are named α (master axis, nominally aligned with the X axis) and β (slave axis, perpendicular to α). In the home position, the β axis is parallel with the Z axis, so the trunnion table is considered to be an AC table-table configuration.

The controller and host software make the following assumptions about the AC table:

- The α axis may be aligned in any direction in the machine XY plane, however a non-zero Z component is not supported. Usually, the α axis is accurately aligned along the X axis, but swiveling the table is permitted if required provided that the trunnion alignment procedure is performed.
- The β axis is perpendicular to α but is not required to intersect it. This allows for manufacturing tolerance in offset. The perpendicularity requirement ensures that all tool orientations are reachable, including singular. Internally, a parameter defines a tolerance angle for the singularity to account for mechanical tolerances.
- When the AC table is mounted on the machine, the table alignment procedure is performed so that both the controller and the host software agree on the position of the table with respect to the absolute XYZ scale positions, and α β home positions. The positioning parameters are stored in the kinematics group of the machine settings.

5-Axis Modes

MotusCNC implements TCPC (Tool Center Point Control) types 1 and 2. Interpolation modes supported are either standard interpolation or tool posture control.

G-Code	Description
G43.4 P0 H_	TCPC type 1 with standard interpolation. P0 is assumed if omitted.
G43.5 P0 H_	TCPC type 2 with standard interpolation. P0 is assumed if omitted.
G43.5 P1 H_	TCPC type 2 with tool posture control.

The interpreter implements these modes in a manner which is compatible with GE-Fanuc. It is recommended to use TCPC type 2 where possible, since this is more compatible with most CAD packages and post processors.

TCPC type 1 with posture control is not implemented.

TCPC Type 1

This mode controls the tool so that its controlled point interpolates linearly with respect to the workpiece, regardless of rotary axis motion. The rotary axis position is specified using the A and C words which directly control the tool orientation angles.

TCPC Type 2

This mode is similar to type 1, except that rotary axis motion is controlled by tool orientation vectors specified via I, J and K words. Tool orientation to actuator position is affected by the transform stack including TWP and WSEC offsets.

Tool Posture Control

There are two supported modes for tool orientation interpolation. The standard (default, or P0) mode linearly interpolates the position of the α,β axes between the start and end positions of a linear (G0 or G1) move.

The other tool posture mode interpolates the rotary axes so that the tool orientation vector remains in the plane defined by the start and end orientations. This mode is selected via the P1 option when TCPC type 2 is selected. This may be described as spherical or "great circle" interpolation.

Coordinate Systems

There is a stack of coordinate transforms that occur between the CAD (program) coordinates in the G-code, and the eventual machine actuator positions. These systems are described in the following table.

System Name	Axis Labels (by TCPC mode)			Description
	None	Type 1	Type 2	
CAD	XYZAB	XYZAC	XYZIJK	As specified in G-code program. Also known as "program" coordinates.
Canonical CAD (CCAD)	XYZαβ	XYZλφ	XYZIJK	Conversion from CAD to fixed (inch) units, also normalized angles or orientation vector.
Reference CAD	XYZαβ	XYZλφ	XYZλφ	Application of Tilted Work Plane (TWP). Also

System Name	Axis Labels (by TCPC mode)			Description
(RCAD)				conversion from orientation vector to angles for type 2.
Adjusted CAD (ACAD)	XYZαβ	XYZλφ	XYZλφ	Application of Work Setting Error Compensation (WSEC).
Table	XYZαβ	XYZλφ	XYZλφ	Application of global (G92) and fixture (G54 etc.) offsets. This coordinate system is fixed with respect to the machine table (and workpiece), taking account of its rotation. For 3-axis work, this is the main machine table; for 5-axis it is the rotary table.
Absolute	XYZαβ	XYZαβ	XYZαβ	Application of tool length offsets, and adjustment of XYZ according to the rotary axis positions. This coordinate system is accessible from G-code using the G53 modifier.
Machine	XYZαβ	XYZαβ	XYZαβ	Kinematics proper: non-linear, geometric and thermal correction if implemented.
Actuators	XYZαβ	XYZαβ	XYZαβ	Scaling from machine position to actuator steps.

Note that the conversions from CCAD to Table are all of the form of a 4x4 matrix transform, so the complete transform from CCAD to Table can be expressed as a single “short-cut” matrix. The RCAD and ACAD coordinate systems are not particularly important since they are merely intermediate logical steps, however it is important to note the order of application of TWP, WSEC and fixture offsets.

NOTE: currently, conventional fixture offsets and WSEC are mutually exclusive.

Rotary Axis Conversion

In some coordinate systems, rotary axes are expressed in terms of λ and φ. λ is the angle of the tool axis with respect to the XY plane (so that 90 degrees is the normal perpendicular tool position). φ is the angle of the projection of the orientation vector onto the XY plane, measured from the +X direction using the right hand rule.

There is a simple relation between λ,φ and α,β:

$\lambda = 90 - \alpha$

$\varphi = \beta + 90$

The relation between the tool orientation vector (IJK) and λ,φ is complicated by the fact that there is no unique mapping, since any multiple of 360 degrees added to φ maps to the same orientation vector. The mapping of

$(180 - \lambda, \varphi + 180)$

is also equivalent to that for (λ, φ). This represents the situation where there is a choice between changing sign of the α axis position, or rotating β by 180 degrees.

At $\lambda = \pm 90$ the orientation vector is (0,0,±1) regardless of the value of φ. Since the table physically cannot rotate to the negative position, the positive 90 degree point is the unique singularity point.

The issue with angle to orientation vector ambiguity is resolved by the following principles:

1. Let the current physical axis position be at α,β.
2. Convert IJK using standard trigonometry to λ,φ such that $-90 <= \lambda <= 90$ and $-180 < \varphi <= 180$, with φ set to $\beta+90$ if $\lambda = \pm 90$.
3. Then, if $\alpha < 0$ then set $\lambda \leftarrow 180 - \lambda$ and $\varphi \leftarrow \varphi + 180$. This allows table to remain on the "side" that it is currently at, without always trying to move to the positive side.
4. Finally, add or subtract integer multiples of 360 to φ until it is within ±180 of β. This resolves any whole rotations of the slave axis so that there is no sudden "jump" of over 360 degrees.

Singularity Crossing Behavior

In 3+2 axis modes, the behavior of the rotary axes is directly controlled by the program. In TCPC types 1 and 2, rotary axis position is indirectly controlled via local orientation angles or orientation vectors. In the case of TCPC type 2, the orientation vectors do not encode enough information to fully specify rotary axis position (because of angular wrap-around), so some heuristics are employed by the G-code interpreter. It is assumed that angular moves do not exceed 180 degrees in any axis.

- If the program commands a move where the singularity position must be crossed (i.e. the orientation vector attains (0,0,1) at some point not at the start or end of the move), then the software chooses to change signs of the α axis rather than rotate β by 180 degrees. Thus β is unchanged and only α moves.
- If the move starts and ends at the singularity (possible because the singularity is actually a zone with an angular width of about 1/500 degree) then there is no rotary motion commanded.
- If the move ends at the singularity, the value of β at the end will be unchanged from the start.
- If the move starts at the singularity and ends away from it, the first phase of motion will rotate β only (while staying singular) until it matches the required value of β at the end of the move. It then moves α only to move to the end point. In this case, there are two possible choices for the final β, 180 degrees apart, with α moving either negative or positive. The alternative chosen is the one which moves β by the least amount from its current position. The a α axis then moves either positive or negative.

As with any similar system of master/slave rotary axes, if a move passes close to the singularity, but not directly through it, then there can be relatively rapid movement of β when α is close to 0. In this case, α will not change sign.

Some operators prefer the α axis to remain on one side or the other (always positive or always negative). MotusCNC does not currently support this, since it is considered that superior efficiency and surface quality are obtained by allowing uninterrupted movement through the singular position.

The above considerations cause the execution of any TCPC type 2 program to depend on the initial position of the two rotary axes. Since the software tries to minimize β axis rotations, initially jogging the β axis by 180 degrees and negating the α axis position will affect whether the α axis stays more on the negative or positive side for the entire job.

5-Axis G-Code Commands

Tool Compensation Modes

All of the following commands change the Table→Absolute transform. The machine axes do not move, however the current program and table coordinates will change to reflect the new transform.

Except for G49, the tool compensation modes require an H word to specify the tool table entry whose length (Z), and X,Y offset values are to be applied. G49 is equivalent to G43H0.

If X, Y or Z words are specified in the same block as G43.x, and the motion mode is G0 or G1, then the requested motion is performed in the new (tool compensated) coordinate system, however the movement itself does not constitute tool center point control. That is, linear and rotary axes will all move with linear interpolation as for non-TCPC modes.

G49: Cancel Tool Length Compensation

G49 sets all tool offset values to zero, and cancels TCPC. This is equivalent to G43 H0 since the zero tool table entry is reserved for "no tool".

G43 H_: Standard 3- or 3+2 Axis Tool Length Compensation

G43 H_ compensates for tool length and XY offset, but does not coordinate the linear and rotary axes. Traditional 5-axis machining in this mode requires inverse time feed mode, since it is the responsibility of the post-processor to calculate rotary axis positions and an appropriate time to complete each move segment. The post processor must be re-run each time the tool offsets are changed otherwise part dimensions will change unpredictably.

Tool Center Point Control Modes

In TCPC modes, the controller moves the tool tip ("controlled point") so that the tool moves as commanded with respect to the rotary table, regardless of any rotary axis motion that is required to control the tool orientation with respect to the workpiece.

If there is rotary axis motion commanded (orientation change), then the controlled point generally follows a curved path with respect to a fixed frame, even for a linear (G0 or G1) move. The move is controlled to be linear with respect to the (rotating) workpiece, however.

There is no requirement to use inverse time feed mode, since the units/minute feed rate is applied to the motion of the controlled point relative to the table (and thus any workpiece mounted on the table).

Pure rotary motion changes only the orientation of the tool with respect to the table; the controlled point remains stationary. Rotary motion (orientation change) is performed without regard for feed rate, so pure rotary motion is performed at rapid traverse rates. A non-zero linear motion component will command the rotary axes to move as fast as required to maintain the commanded orientation.

NOTE: this behavior under "pure orientation change" move is different from Fanuc. MotusCNC always rotates at rapid rate provided that any compensatory linear (XYZ) motion can keep up. Other controllers re-interpret the current feed rate as being for degrees rather than mm/inch, however MotusCNC takes the approach that this is an undesirable and ambiguous discontinuity. In all cases, movement is constrained by the slowest actuator, and maintains tool position within specified tolerance.

There are two TCPC modes, described below.

G43.4 H_ P_: Tool Center Point Control Type 1

This TCPC mode allows direct control of the rotary axes, using A and C words. The A word maps to the α axis, and the C word maps to β. Both axes are expressed in degrees.

The P word specifies tool posture control:

- P0 – no posture control. Rotary axis positions are linearly interpolated in the same way as the linear axes. This is the default if no P word is provided in the block.
- Currently, P1 is not supported for TCPC type 1.

Examples:

```
M6 T1                 (Tool 1)
G49 G91 G30 Z0        (Cancel TCPC, move to Z safe)
G30 X0 Y0 A0 B0       (Move to home pos G30, which is above rotary)
G90 G21               (Absolute motion, mm)

G43.4 H1              (Start TCPC mode 1, tool 1)
G0 X0 Y0 Z0           (Move tool tip to rotational center)
X60                   (Move to edge of table, will move to right)
C180                  (Rotate C by additional 90 deg)
```

In the last movement above, the tool will maintain its position above the initial point of the table while the table rotates, ending up with the tool at the back of the table. In this case, X and Y motion is coordinated so as to maintain a constant tool position over the rotary table. The motion is maintained to within the accuracy specified by the "collinear tolerance" motion planner parameter.

The above example shows an initial move to the G30 home position. It is recommended to use this prolog, or equivalent absolute move or jogging, to position the spindle approximately over the rotary table before entering TCPC modes. In addition, the rotary axes should be homed as well.

Rapid moves (G0) which begin or end outside the workpiece envelope set in the motion planner settings are not performed with TCPC. This avoids the problem of unexpectedly large motion when the effective radius of the tool tip, measured from the table center of rotation, is large.

Note that G53 motion mode temporarily overrides TCPC. G28 and G30 homing also work like G53 and will override TCPC.

Recommendations for entering, exiting and living with TCPC modes:

- Make large traverse movements (e.g. after tool change when positioning near the rotary table) before setting TCPC modes, or after canceling TCPC. (Although this requirement is mitigated by the workpiece envelope setting).

- Alternatively, specify G53 (absolute positioning mode) when traversing long distances.
- Home position G30 is defined to be above the calibrated rotary axis centre, so after moving Z to a safe height, home X and Y to this position, as shown in the above example. Home position G28 is near the axis positive limits, so is not near the rotary table in standard machine configurations.
- While in TCPC modes, use G91 G30 (or G28) Z0 to move Z to a safe height. Don't forget to add G90 after the G28/G30 block to restore to absolute positioning mode.
- Do **not** assume that something like G91 G0 Z10 will safely move the tool up by 10mm. This is because the work plane may be tilted, so that the program Z axis is not aligned with the tool. Such a move might move X and Y as well.
- For the same reason as above, do not use G30 other than with G91 Z0. This is because the initial motion specified with G30 moves in program coordinates. Only G91 Z0 ensures that there is no motion in program coordinates.

G43.5 H_ P_: Tool Center Point Control Type 2

This mode is requires tool orientation to be specified using I, J and K words. There are some limitations to this format as explained in section 10.1.4 regarding disambiguation of conversion to angles, however this is the preferred mode for continuous 5-axis machining since it is the most natural mapping from CAD models.

The P word specifies tool posture control:

- P0 – no posture control. Rotary axis positions are linearly interpolated in the same way as the linear axes. This is the default if no P word is provided in the block.
- P1 – posture control. Tool orientation is controlled throughout the move, so that the orientation vector sweeps the plane defined by the start and end orientations. The sweep angle is linearly interpolated so that the orientation vector interpolates from the start to the end orientation.

Note that P1 posture control imposes some limitation on the allowable movement. The total change in orientation vector is not allowed to be greater than or equal 180 degrees. This is because it is impossible to define a single orientation plane in this case. In practice, the allowable orientation change is limited to less than this because of high sensitivity to rounding errors. It is very unlikely that a post processor would produce such drastic orientation changes, so this issue is unlikely to be encountered.

There is no requirement to normalize I,J,K in the program.

Because I,J,K words are used to define tool orientation, it is not possible to use "center format" arcs (G2/G3); only radius format arcs are supported.

Example:

```
M6 T1              (Tool 1)
G49 G91 G30 Z0     (Cancel TCPC, move to Z safe)
```

```
G30 X0 Y0 A0 B0      (Move to home pos G30, which is above rotary)
G90 G21

G43.5 H1 P1          (TCPC type 2, tool 1, posture control)

(Machine chamfers on square feature.)
m3 s10000
g0 x0y0z0 i-1j-1k3
g1 f600 x50 i1j-1k3
y50 i1j1k3
x0 i-1j1k3
y0 i-1j-1k3
m5 g53 g0 z0              (Z safe)
m2
```

In the above, tool orientation is (±1,±1,3) so the tool tilt up from the XY plane will be arctan(3/√2), or equivalently an angle of 25.2° from the perpendicular. Because tool posture control is selected, and linear motion is parallel to the tool orientation plane, the machined surfaces will be planar if milled using the side of a normal "square" end-mill.

Tilted Work Plane (TWP)

TWP is a general offsetting and tilt transform that allows program coordinates to be specified relative to any desired part location.

TWP is not restricted to 5-axis jobs. Even in 3-axis mode it allows for rotation of the program XY plane. There can also be rotations in the other planes, however this may produce unexpected results because the tool orientation cannot change.

Warning: be careful when programming TWP (or WSEC) where the XY plane is not parallel to the actual machine XY plane. This represents a tilt, so that any X or Y movement will generally cause a compensating Z movement so that the controlled point stays in the defined tilted plane, and vice versa. This can cause problems with canned cycles, since drilling cycles can only move parallel to one of the local X, Y or Z axes. The machine movement will be tilted, however, which risks dragging a drill sideways into the workpiece, breaking it. If a 5-axis machine is in use, then G53.1 can be used to bring the tilted work plane parallel to the machine XY plane (perpendicular to the tool), so canned cycles may be used. When only 3- or 4-axis hardware is in use, G53.1 cannot be used so this warning applies.

G69 – Cancel TWP

G69 sets the TWP transform to identity i.e. cancels it. It is recommended that the program include G69 in its prolog, and also before program end. Other programs which are not 5-axis aware will not generally include a G69 command, so any TWP setting left over could drastically affect the operation of such a program.

G68.2 – Set TWP

This command allows the program to alter the program coordinate system. It is a generalization of fixture and global offsetting (G54, G92 etc.) that allows not only linear offsets but also arbitrary 3-D rotation.

Unlike offsetting, however, TWP is intended for temporarily redefining the CAD coordinate system. It is not intended for aligning the CAD and workpiece like an offset or WSEC.

For example, suppose a workpiece was basically the shape of a hex bolt head, and that the same feature was to be machined on all 6 sides of the head, and a different feature on the top (XY plane). The traditional way of doing that would be to tilt the workpiece 90 degrees and then either:

- Index the workpiece by 60 degrees each time, and machine the side feature using the same XY coordinates each time, or
- Let the CAD program generate the same feature in multiple locations, adjusting X,Y,Z coordinates each time.

The first approach is obviously easier to understand, but is really a 3+2 axis mode. The second is full 5-axis, but generates a program that is 6 times larger.

TWP allows “indexing” to be simulated in 5-axis mode. For this example, a new program coordinate system is generated for each of the 6 faces, and then the same XY code can be run each time. Supposing the top face is the basic XY plane at Z=0. After running the code for the top feature, the TWP is set so as to rotate by 90 degrees and move the origin to the appropriate point on the side face. The side feature is machined, then the work plane is rotated by 60 degrees (and the origin moved again). This repeats until all sides are done.

The following table describes the several ways that the TWP can be defined in the program:

G-Code	Description

`g68.2 p0 x_ y_ z_`
`i_ j_ k_`
Note: indented lines indicate same program block as previous line.
	This is the default setting if the P word is omitted. Sets TWP using an initial offset of X,Y,Z i.e. the new origin will be at (x,y,z) in the current program system. Then the system is rotated by Euler angles i, j, k:
	1. Rotate by i degrees about the current Z axis;
	2. Rotate by j degrees about the new X axis;
	3. Rotate by k degrees about the new Z axis.
	In this and all other commands, rotation follows the right hand rule unless otherwise noted.

`g68.2 p1 q_ x_ y_ z_` `i_ j_ k_`	Similar to above, except Q word specifies axis order of rotations. The Q word is a 3-digit number consisting of permutations of the digits 1,2,3. '1' applies the next rotation about the new X axis, '2' about Y, '3' about Z. The angle data is taken from I,J,K in order. Example: `g68.2 p1 x0y0z0 q312 i30 j-40 k50` No offset, rotate 30 degrees about Z, rotate -40 degrees about X, then 50 degrees about Y.

`g68.2 p2 q0 x_ y_ z_ r_`
`g68.2 p2 q1 x_ y_ z_`
`g68.2 p2 q2 x_ y_ z_`
`g68.2 p2 q3 x_ y_ z_`
	Set TWP via three points and an offset. This command requires 4 blocks for completion. The Q word specifies the datum number for each point. The TWP is generated only when all 4 datum points are accumulated (Q0-Q3).

G-Code	Description
	This works in the following stages: 1. Initial offset so that Q1 is the new origin. 2. The direction vector (Q2-Q1) defines the direction of the new X axis. 3. The vector (Q2-Q1)×(Q3-Q1) defines the direction of the new Z axis 4. The new Y axis is Z×X 5. The origin is shifted by Q0 6. The XY plane is rotated about Z by the R word. Note that the R word can be specified on any of the 4 blocks. An error is flagged if any of the cross products are too small i.e. the transform is not well defined.

`g68.2 p3 q1 x_ y_ z_`
`            i_ j_ k_`
`g68.2 p3 q2 i_ j_ k_`
	Set TWP via two vectors and an offset. This command requires two blocks for completion. The first block (Q1) provides the XYZ offset and a vector IJK specifying the direction of the new X axis. The second block (Q2) implies the direction of the new Z axis.
	
	This works as follows:
	1. Initial offset so that Q1(x,y,z) is the new origin.
	2. Q1(i,j,k) is the X axis direction
	3. $Q2 \times Q1$ is the new Y axis
	4. $Q1 \times (Q2 \times Q1)$ is the new Z axis
	An error is flagged if Q2 is more than 5 degrees away from perpendicular to Q1.

`g68.2 p4 x_ y_ z_` `         i_ j_ k_`	Set TWP by projection angles. X,Y,Z provide an initial offset. IJK specify angles as follows: 1. The current X axis vector (1,0,0) is rotated by -i about the Y axis, forming a new X' axis. 2. The current Y axis vector (0,1,0) is rotated by j about the current (old) X axis. Call this W. 3. The new Z axis is computed as $Z' = X \times W$. 4. X' is rotated by k about Z'. 5. $Y' = Z' \times X'$. Note that the initial rotation is negative of the provided i value since it is counter to the standard right hand rule. A positive i value rotates the X axis into the first quadrant of the ZX plane.
Codes above this are standard Fanuc compatible. Following codes are proprietary to Menig Automation.	
`g68.2 p7 q1 r_ x_ y_ z_`	Match 3 points in the target CAD space to the same number of

G-Code	Description
`g68.2 p7 q2 r_ x_ y_ z_` `g68.2 p7 q3 r_ x_ y_ z_`	points defined in the current CAD space. The target coordinates are specified in the XYZ words of this command. The actual points (which may be data from probing etc.) are defined separately using the G10.8 or G38 commands, and stored in a single buffer called the probe buffer. The R words in this command index entries in the probe buffer. If any two commands use the same R word, then the same point is selected twice, which will probably cause an error because the points must be distinct and not collinear. This calculates the transform as follows: 1. For both target and measured points, construct orthogonal unit axes in the same way as done for g68.2 p2. The origin of both systems is specified by the first point (Q1); the X axis direction is specified by (Q2-Q1); the Z axis direction by (Q2-Q1) $\times$ (Q3-Q1). 2. Compute an overall transform that maps the target system onto the measured system.
`g68.2 p8 q_ r_ x_ y_ z_` Q and R values index 0..31.	This is an extension of the above, which permits additional point data to be matched. The optimum transform is computed which minimizes the sum of squared error between target and probed points. Up to 32 points may be specified for matching, using the Q word value of 0 to 31 inclusive. Similarly, the R word is used to index the probe buffer in the same range (although this may be any permutation). The TWP is calculated as soon as Q1, Q2 and Q3 are accumulated. Thus, if more than this many points are required, then hold off specifying Q1 until all other points are accumulated. If only Q1, Q2 and Q3 are accumulated, the result is not necessarily the same as g68.2 p7 for the same data. This is because the least squares fit will use the remaining degrees of freedom to adjust the transform. Specifically, a TWP transform has only 6 DOF, but specifying 3 by 3-D points to match is 9 DOF, which overdetermines the solution. g68.2 p7 follows a deterministic algorithm which discards the redundant data. g68.2 p8 always uses least squares fitting to determine the most accurate possible transform.

Note regarding multiple block accumulation: there is a single buffer which collects data for TWP. Thus, both g68.2 and g68.4 share the same buffer.

The data buffer is reset whenever the P word changes, and also after the TWP is calculated from a complete data set. If insufficient data is provided (missing Q words from the required sequence) then the pending incomplete TWP will not be generated.

It is recommended to put all G68.x data on contiguous program blocks. They do not need to be specified in order.

G68.4 – Compose TWP

This command is similar to G68.2, except that the coordinate transform is composed on top of the existing program coordinate system, if any, rather than the base workpiece coordinate system. G68.4 immediately following G69 has the same effect as G68.2.

TWP Example

The following example illustrates use of g68.2 p7. Other setting types are analogous.

Suppose a workpiece was defined in the CAD program to have measurable location features (such as bored holes) centered at XY = (10,10), (50,20) and (-10,60). Suppose then that this part has the location features created, but had to be removed from the machine before completion of a second machining operation. When re-mounted on the machine, it is possible to use TWP (actually, WSEC would be more appropriate, but this example uses TWP) in order to realign the CAD and actual workpiece coordinates:

```
(Enter actual probe results measuring the location features.)
(This would normally be from a G38 probe, but for this example)
(enter the data by hand...  where the part currently is...)
g10.8 r13 x11.5 y9.52 z0.14  (oops – it was a bit off!)
g10.8 r14 x52.9 y19.93 z0.17 ( - but no problem!)
g10.8 r15 x-8.76 y58.21 z0.28

(Now define where it should be in a perfect world...)
g68.2 p7 q1 r13 x10 y10 z0
g68.2 p7 q2 r14 x50 y20 z0
g68.2 p7 q3 r15 x-10 y60 z0

g53.1     (Tip it around so tool is perpendicular)

g0 x10 y10 z10  (Bingo – tool is exactly 10mm above first location!)

(Happily proceed with rest of machining...)
```

Work Setting Error Compensation

WSEC is similar to TWP except for the following:

- WSEC compensation (if any) is applied after TWP when transforming from program to table coordinates.
- There is only a single TWP transform, whereas there are 7 WSEC transforms, one of which can be selected at any point. This is analogous to conventional fixture offsets (CFO: G54, G55 etc.).
- The transform data for WSEC is provided by G10.2 or G10.4 commands. There are 8 buffers for WSEC transforms: buffer 0 is "global" and the other 7 are composed on the global buffer.
- The active WSEC is selected using G54.4 P_. The P word is 0 to cancel WSEC, otherwise P1..7 selects WSEC of that number, composed onto the global WSEC.
- When WSEC is active, CFOs (G54, G55,...) and the global offset G92 are ignored. In other words, WSEC and CFOs are mutually exclusive. This is different from Fanuc, which combines the action of both. MotusCNC considers this confusing and ambiguous when 5-axis TCPC modes are used. In any case, linear offsetting always can be incorporated into WSEC offsets.

When probing data is used to define a coordinate transform (TWP or WSEC), it is preferable to define a WSEC transform, and reserve TWP for moving around locally on the workpiece.

NOTE: in MotusCNC, if WSEC is in effect it is not currently possible to zero or set the DROs using the right mouse menu. This is mainly because WSEC involves rotations, which are not simple to apply. It is expected that WSEC will mainly be used in the context of G38 probing, rather than manual touch-off etc.

G54.4 P0 – Cancel WSEC

G54.4 P0 sets the WSEC transform to identity i.e. cancels it. It is recommended that the program include G54.4 P0 in its prolog, and also before program end. Other programs which are not 5-axis aware will not generally include a G54.4 command, so any WSEC setting left over could drastically affect the operation of such a program.

When WSEC canceled, any conventional fixture offsets (G54 etc.) are reinstated.

G54.4 P1..7 – Select WSEC 1..7

This command makes the selected WSEC active. Each of the 7 coordinate transforms is composed from the "global" WSEC (specified using G10.2 L0) and the selected WSEC from G10.2 with an L word matching the P word on the G54.4 command.

When MotusCNC is started, all of the WSEC transforms are initialized by reading the previously saved values. If there are none, then all WSEC transforms are identity.

WSEC transforms are retained from job to job, like the conventional fixture offsets.

G10.2 – Program WSEC Data

G10.2 L_ specifies transform setting data analogously to G68.2. The L word indexes the WSEC buffer which is to be set. L0 specifies the global WSEC, and L1..7 specifies the numbered WSEC.

Except for the L word, all other parameters of G10.2 are the same as for G68.2.

This command is not specified by Fanuc. It is a proprietary extension.

G10.4 – Compose WSEC data

This command composes a new transform onto an existing one. It is provided for symmetry with the equivalent TWP commands g68.2 and g68.4.

This command is not specified by Fanuc. It is a proprietary extension. Fanuc does not support composing on top of an existing WSEC transform. It is expected that the data is entered manually via the controller interface.

WSEC Example

The example for TWP above is modified to use a WSEC transform instead:

```
(This part is same as for TWP example.)
g10.8 r13 x11.5 y9.52 z0.14
g10.8 r14 x52.9 y19.93 z0.17
g10.8 r15 x-8.76 y58.21 z0.28

g69  (cancel TWP)

(For WSEC, use g10.2 instead of g68.2.  Also, add the L word)
(to index the appropriate target WSEC transform number [0-7].)
g10.2 L1 p7 q1 r13 x10 y10 z0
g10.2 L1 p7 q2 r14 x50 y20 z0
g10.2 L1 p7 q3 r15 x-10 y60 z0

(There is no command to explicitly clear WSEC data, but)
(can use the following code to achieve the same effect.)
(In this case, ensure there is no global [index 0] WSEC)
(because the global WSEC is always applied.)
g10.2 L0 p0 x0y0z0i0j0k0

(Enable the WSEC [index 1] that we just defined)
g54.4 P1
g53.1     (Tip it around so tool is perpendicular)
g0 x10 y10 z10  (Bingo – tool is exactly 10mm above first location!)

(Happily proceed with rest of machining...)

g54.4 p0  (Clear WSEC.  The transform can be turned on again)
          (with another G54.4 P1).
```

G10.8 Probe Data Buffer

The g68.2/4 and g10.2/4 commands with P7 or P8 refer to point location data stored in the probe result buffer. This buffer contains 32 entries, which are indexed by the R word in the g68.2 etc. commands.

Probe Data Buffer Coordinate System

Each entry in the probe data buffer is a vector of 3 or 6 components, which are interpreted as a 3-D point, a 3-D direction vector, or a point and direction.

The 3-D point data is normally stored in the "table coordinate system", which is a frame of reference that is fixed with respect to the workpiece. The units of measurement are also fixed. The reason for this is so that data buffer entries may be combined at arbitrary points throughout the program execution, and the relation of the data points to the workpiece do not change even if the rotary table position is changed.

If necessary, the P_ parameter may be used to modify the stored coordinate system, however this is advanced use and may cause unexpected behavior if entries with inconsistent units or frames of reference are combined.

Probe Data Buffer Set Command

The data in this buffer is normally obtained from G38 probing (c.f.). It may also be specified directly in the G-code program, using the G10.8 command.

```
G10.8 P_  R_  X_ Y_ Z_  I_ J_ K_  Q_  L_
```

Parameter word	Description
P_	P0 – This is the default if the P word is not specified. Coordinates are stored in the table coordinate frame, converted from current CAD units. P3 – Coordinates are stored without any conversion. P-3 – Coordinates converted from table frame to current CAD units. This is most often used with the Q6 option when reading a buffer entry into program-accessible hashvars.
R_	Index the entry in the probe data buffer to be set by this command. Acceptable values are 0..31.
X_	Point coordinate data (optional). If at least one is specified, any missing axes are set according to the Q word option. If none are specified, this entry is marked as "vector only".
Y_	
Z_	
I_	Vector (direction) data (optional). If at least one is specified, any missing values are set according to the Q word option. If none are specified, this entry is marked as "point only".
J_	
K_	
Q_	Options: The Q word has an optional digit after the decimal point:

Parameter word	Description
	.0: (default) – do not change the entry type or set it according to the given data; .1: Mark the entry as "point only"; .2: Mark the entry as "direction only"; .3: Mark the entry as "point plus direction". The whole number part of the Q word selects from the following options: Q0 – (default) Missing XYZ values set to current program coordinates; missing IJK values set to zero. Q1 – Missing XYZIJK values are not modified from current buffer entry values (expressed in the current CAD system) Q2 – Missing XYZIJK values obtained from the 6 contiguous hashvars starting at the L word. E.g. g10.8 q2 L100 will fill missing X from #100, missing K from #105 etc. Q3 – Missing values copied from buffer entry indexed by L word. In this case, all of XYZIJK may be missing, which effectively copies another entry including its type marker. Note that the L buffer entry which is copied is assumed to be in the table coordinate system, and is always converted to the current CAD system to provide the missing values. The following options do not permit any of XYZIJK to be specified unless otherwise noted: Q4 – Project the current point data to the plane specified by the L word. For example, point data obtained by probing a bore might be projected to the surrounding surface plane. "Projection" means finding the nearest point on the L plane to the current point. Q5 – Move the current point along the current direction vector until the L plane intersection is found. This entry must be "point plus direction". Q6 – do not make any changes, but write the entry data to hashvars 5061-5066 plus the type marker to #5060. The type marker value is interpreted thus: 1: point; 2: direction; 3: point plus direction. X,Y,Z,I,J,K values are written to 5061, 5062, 5063, 5064, 5065, 5066 respectively. A point entry will not alter 5064-5066, and a direction entry will not alter 5061-5063. Usually, this would be used with P-3 in order to transform the table coordinate system into the current CAD system, which would be more meaningful to programs. Q7 – Move the current point for the distance I along the direction vector of the entry indexed by the L word. If used with a plane, specifying the same index for R and L, this has the effect of displacing the plane by the given amount, keeping it parallel to the original.
L_	For Q0, Q1 or Q6: not permitted. For Q2: L word indexes the first hashvar (of 6 contiguous) to use for missing

Parameter word	Description
	values, in the order XYZIJK. For other Q word options: Index to another buffer entry (0..31 if specified). This is used by the Q3 option to duplicate values which are not specified. Also used by Q4 and Q5 to specify the reference plane. In this case, the L entry must be marked as planar i.e. point plus vector.

If at least one of XYZ *and* one of IJK are specified, then the entry is marked as "point plus direction", unless overridden by the Q word decimal.

G10.8 Examples

This example demonstrates how incoming "raw" probe data can be manipulated so as to define coordinates on the workpiece.

> NOTE: when using probe data to set local transforms (TWP, WSEC, fixture offsets) it is important to turn off these features first. This is because probing data is stored in the table coordinate system. The 3-point match feature expects the target and current point coordinates to be expressed in the same frame of reference. Thus, the program and table coordinate systems must be identical. This is accomplished using the following code, which should be printed out, laminated, and nailed to the workshop walls:

```
g69 (cancel TWP)
g54.4 p0 (cancel WSEC)
g92.2 (zero global offset)
g10 L2 p9 x0 y0 z0 a0 b0 c0 (clear offset 9)
g59.3 (select fixture offset 9)
```

> Tool offsets are permissible, and in fact required to be correct for the probe.

Supposing the workpiece has 3 holes bored in known locations, but we don't know exactly where the workpiece is positioned on the machine table, except that the top face is parallel to the XY plane, and we want to define that plane level as Z=0. We can probe the hole positions, but the hole center location is at an arbitrary Z level (it depends on where the operator jogs down to when manually positioning the probe inside the bores).

So, in order to use the "point matching" mode for setting WSEC, the probed bore centers need to be raised up to the Z=0 level.

Assume that the bores are about 6mm diameter and centers located at XY = (10,10), (50,20), (-10,60). The workpiece is located within +/- 1mm of the expected position, but we wish to find its exact location:

```
g69 (cancel TWP)
g54.4 p0 (cancel WSEC)
g92.2 (zero global offset)
g10 L2 p9 x0 y0 z0 a0 b0 c0 (clear offset 9)
g59.3 (select fixture offset 9)
```

```
m6 t97 g43 h97 (mount probe and compensate)
x10 y10
z-5       (move probe into first bore)
g38 p2 i1 j0 r3 L10      (probe and save to buffer 10)
z5
x50 y20
z-5
g38 p2 i1 j0 r3 L11      (probe and save to buffer 11)
z5
x-10 y60
z-5
g38 p2 i1 j0 r3 L12      (probe and save to buffer 12)
z5

(Now bring the “random” probed Z positions up to the top plane)
(which is Z=0.  This will create points that are the intersection)
(of the bore centerline with the top plane.)
g10.8 r10 q1 z0
g10.8 r11 q1 z0
g10.8 r12 q1 z0

(Now use the probe data to create a WSEC which exactly compensates)
(for the +/-1mm uncertainty in workpiece positioning.)
(L1 is the WSEC index number;)
(P7 is the “point matching” function code;)
(R_ indexes the g10.8 data entries above.)
g10.2 L1 p7 q1 r10 x10 y10 z0
g10.2 L1 p7 q2 r11 x50 y20 z0
g10.2 L1 p7 q3 r12 x-10 y60 z0

(Enable the WSEC [index 1] that we just defined)
g54.4 P1

m6 t1 g43 h1   (mount cutter)
g0 x30 y30
g0 z5
etc...
```

The above example relied on a few assumptions, the most significant being the assumption that the top plane of the workpiece was parallel to the machine XY plane. This is the normal case for 3-axis work, since the work is usually seated in a vice which is accurately parallel to the machine axes.

Since the probed Z position “should be” zero, it was sufficient to use the g10.8 q1 command to simply force the Z position to zero for each probe datum.

For 5-axis work, it might be more difficult to set the work precisely, so for accurate work it might be necessary to compensate for arbitrary tilt. In this case, the top plane of the workpiece can itself be probed. The following example replaces the first part of the above, to demonstrate dynamic plane intersections.

Assumptions: the XYZ positioning is within +/- 1mm (this is just so we can fully automate the probing: if not, then some manual jogging code needs to be inserted, but the principle is the same).

```
g69 (cancel TWP)
g54.4 p0 (cancel WSEC)
g92.2 (zero global offset)
g10 L2 p9 x0 y0 z0 a0 b0 c0 (clear offset 9)
g59.3 (select fixture offset 9)

m6 t97 g43 h97 (mount probe and compensate)
x10 y10
z-5       (move probe into first bore)
g38 p2 i1 j0 r3 L10      (probe and save to buffer 10)
z5
x50 y20
z-5
g38 p2 i1 j0 r3 L11      (probe and save to buffer 11)
z5
x-10 y60
z-5
g38 p2 i1 j0 r3 L12      (probe and save to buffer 12)
z5

(The above was same as for previous example.  Now add the top)
(plane probe.  Assume that we can probe the corners of an XY)
(rectangle at [5,5] with side lengths [50,40])
g0 x5 y5
z2
g38 p4 i50 j40 k-5 L9    (XY planar probe to buffer 9)

(Now bring the “random” probed Z positions up to the probed plane)
(This will create points that are the intersection)
(of the bore centerline with the measured top plane.)
(R_ is the buffer index to modify; L_ is the plane to project to;)
(Q4 is the “project point to plane” command option.)
g10.8 r10 q4 L9
g10.8 r11 q4 L9
g10.8 r12 q4 L9
```

Since the resulting WSEC might be tilted, for accurate 5-axis work a G53.1 command would be used to rotate the workpiece was perpendicular to the tool.

The above example assumes that the workpiece is not highly tilted, otherwise the initial bore probing might not be accurate because the g38 p2 (hole probe) assumes that the bore axis is fairly closely aligned with the machine Z axis. If this is not the case, then better accuracy would be achieved by splitting the probing into two steps:

- First, probe the top plane and compute an initial WSEC, followed by G53.1, to actually make the bores align vertically. This step would rotate the workpiece so that the bores are in the best position for probing.
- Then, probe the bores and compute the final WSEC.

Best accuracy would be achieved by probing both the bores *and* the top plane (again). The reason for the improvement is that best accuracy is attained by making all related probes without moving the rotary axes in between.

The next example demonstrates a technique for dealing with a workpiece which is lacking one or more distinct “features” with which to align.

In the case of a turned workpiece which is mounted on the 3-jaw chuck on the trunnion table, because of the rotational symmetry there is ambiguity in its rotational position about its own axis. Since one position is as good as any other, it makes sense to pick an arbitrary position and stick to that for the remainder of machining.

One approach to handle this situation is to:

- Probe the face of the part to establish the position and orientation of an XY plane. This is done with the part axis aligned vertically;
- Probe the part using the “boss” (or pin) operation to establish a point on the part's axis.

This gives a single point which is the intersection of the axis and the face plane.

Since 3 non-collinear points are required to unambiguously define the full 3-D position and orientation of the part, it is necessary to make 2 additional “virtual points”. The most robust way to do this is to programmatically define a g10.8 entry for points to the right (+X) and back (+Y) from the axis point.

The following example illustrates this technique for a 50mm diameter cylinder, set up so the face is roughly at Z=0 when the rotary table is in home position:

```
g69 (cancel TWP)
g54.4 p0 (cancel WSEC)
g92.2 (zero global offset)
g10 L2 p9 x0 y0 z0 a0 b0 c0 (clear offset 9)
g59.3 (select fixture offset 9)

m6 t97 g43 h97      (mount probe and compensate)
g0 a0 b0            (table to home pos)
g0 x-15 y-15
z5
g38 p4 i30 j30 k-10 L9 (probe face plane, to buffer 9)
z5
x-30 y0
z-5
g38 p3 i10 j0 r25 a10 b5 L10 (probe outside of cylinder, buffer 10)

g10.8 q4 r10 L9          (project center to face plane)
```

```
g10.8 q6 r10             (write center pt to hashvars 5060-5066)
(Now copy buf 10 to buffers 11,12 except modifying X and Y)
(We arbitrarily add 50mm in each dimension, it just needs to)
(match the values in the following g10.2's.)
g10.8 q3 r11 L10 X[#5061+50]
g10.8 q3 r12 L10 Y[#5062+50]
(Project new points onto face plane)
g10.8 q4 r11 L9
g10.8 q4 r12 L9

(Create WSEC by point matching)
g10.2 L1 p7 q1 r10 x0 y0 z0       (Define center as 0,0,0)
g10.2 L1 p7 q2 r11 x50 y0 z0      (Fake +X point at 50,0,0)
g10.2 L1 p7 q3 r12 x0 y50 z0      (Fake +Y point at 0,50,0)

(Enable the WSEC [index 1] that we just defined)
g54.4 P1
```

Admittedly the above seems fairly complex, compared to simply touching the tool off at the cylinder face and going from there. Using probing and WSEC gives the following advantages, however:

- Allows the chuck to be offset from the rotational center;
- Allows the workpiece to be offset in the chuck;
- Allows the workpiece to be slightly tilted in the chuck.

G10.8 Tips and Tricks

In the following, and throughout this document, the 'L' word is capitalized. This is not necessary, but is a convention to avoid confusion with the digit '1'.

```
g10.8 r1 q0.1
```

Set entry 1 to current program x,y,z.

```
g10.8 r1 q0.3
```

Set entry 1 to current program x,y,z, and direction to (0,0,0) i.e. no direction. Zero vectors are not very useful, and can cause errors in other operations, so typically e.g. K1 would be added to set the direction. Note that direction vectors are not normalized when stored, but will be normalized internally when required.

```
g10.8 r2 q3.1 L1
```

Copy entry 1 to entry 2 (as a point).

```
g10.8 r3 q0.2 k1
```

Set entry 3 to the just the unit Z direction (0,0,1), with no point. Q0.2 indicates that only IJK values are required, and missing values are set to zero.

```
g10.8 r4 L3 q3.2 p3
```

Set entry 4, as a direction vector, to entry 3 transformed to the current CAD coordinate system. Q3.2 indicates that unspecified IJK values are obtained from entry 3 (L3), and the .2 ensures that the result is also a direction vector. The output vector will have the same length as the input vector, since the transforms do not do any scaling.

```
g10.8 r5 L5 q7 i10
```

Move entry 5 (assumed to be a plane) by 10 units in its own direction. The I word specifies the displacement in the plane's normal direction. If L indexes a different entry, that entry's direction is used as the displacement direction for the R index entry.

G38 Probing

The G38 command is implemented directly in the interpreter. G38 does everything that M100 probing does, except it works slightly faster and is better suited for 5-axis work. It does not make use of hashvars for control, and is easier to integrate with TWP and WSEC transforms.

G38 probing uses some default parameters which are set via G10.7, and also uses parameters provided in the same block.

The probing results are stored in hashvars 5060-5064, and optionally stored in the G10.8 buffer.

G10.7 – Set default G38 Probe Parameters

```
G10.7 A_ B_ C_ P_ Q_
```

This command provides default values for parameters which are not specified on an individual G38 command. In addition, the P word provides an overall timeout value which cannot otherwise be specified with the G38 command.

Parameter	Equivalent M100 hashvar	Default	Description
A_	5007	10mm	Z hop distance. The specified distance is stored internally in fixed units, therefore the default parameter saved here is independent of the units mode in effect at the time of the G38. For example: `g20 g10.7 a1`

Parameter	Equivalent M100 hashvar	Default	Description
			sets the Z hop distance to 1 inch = 25.4mm whereas `g21 g10.7 a1` sets the Z hop distance to 1mm = 1/25.4 inch.
B_	5009	5mm	Stepover distance
C_	5008	3mm	Clearance distance
P_	5005	60	Timeout (seconds). G10.7 setting of this parameter is the only way to specify an overall operation timeout with G38, since the G38 P word is used to specify the probe operation type.
Q_	5000	0	Option flags. Same options used for G38 as M100, except that point cloud probing is currently not supported for G38. 1: Quiet. No user interaction if error. Program needs to check for errors. 2: Halt on error. G-code is halted if any error encountered. 3: No user interaction, but halt if error. Default (0) means that the user is prompted if there is an error, and the program is only halted if the user cancels.
n/a	5001		Tip radius units. Not used with G38: tip radius is always obtained from the tool table entry for the probe.
n/a	5004		Tip radius. Not used: see above.
n/a	5010-5012		Indices. If required, these are provided directly in the G38 block via the I,J,K words.
n/a	5013		Approach speed. Not used with G38: approach speed is always the configured value in the machine settings for the probe.
n/a	5014		Backoff speed. Not used with G38: backoff speed is always the configured value in the machine settings for the probe.

G38 – Probe Command

The G38 command starts a probing operation, waits for the result, then stores the result in hashvars 5060-5064, and optionally in the probe data buffer. The probe data buffer is the same point data buffer used by the G10.8 command. This permits probe result data to be used when setting TWP or WSEC transforms.

When the probe operation takes XYZ parameters, the destination point is transformed into machine coordinates the same as any other use of coordinates at that point. So, for example, if TWP is in effect which rotates the coordinate system, the probe direction will be rotated correspondingly.

On the other hand, if the probe operation takes IJK parameters, these represent a direction in machine coordinates and are **not** transformed by TWP etc. Thus, I is always along the machine X axis, and so on.

Probe Operations

```
G38 P_ L_ X_ Y_ Z_ I_ J_ K_ R_ A_ B_ C_ Q_
```

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>P_</td><td>Specify probe operation code. This is the same set of codes documented for M100, using hashvar 5006. G38 probing does not use hashvars so the probe operation is always provided using the P word.</td></tr>
<tr><td>L_</td><td>Result destination index for G10.8 probe data buffer. On successful completion of the probe operation, the L index (in the range 0-31) specifies the entry in which to place the resulting data. Coordinates are stored in the table coordinate system, expressed in current program units.

If L-1 is used, then the result is not stored in the G10.8 buffer, but is only written to the hashvars.

If L is omitted, the default is the same as L-1.</td></tr>
<tr><td>X_</td><td rowspan="3">If G90 mode: Generally used as an absolute destination point.
If G91 (incremental) mode: relative destination point.

At least one of these words must be specified for probe operations which use them. Words which are not defined will default to the current program position for the specified axis. The actual destination point is transformed using TWP or WSEC in effect, if any.</td></tr>
<tr><td>Y_</td></tr>
<tr><td>Z_</td></tr>
<tr><td>I_</td><td rowspan="4">I, J, K, R meaning depends on the probe operation selected. These are documented below. In general, IJK are used to specify displacement vectors, and R provides an expected result dimension. IJK are also used to provide indices to combine previous probe results.

At least one of I,J,K must be specified for probe operations which use them. Words which are not defined will default to zero. There is no default for the R word.

I,J,K vectors are not transformed by local TWP or WSEC. They always map to machine axes X,Y and Z respectively.</td></tr>
<tr><td>J_</td></tr>
<tr><td>K_</td></tr>
<tr><td>R_</td></tr>
<tr><td>A_</td><td>Optional override of default Zhop specified by G10.7 A_</td></tr>
<tr><td>B_</td><td>Optional override of default stepover specified by G10.7 B_</td></tr>
<tr><td>C_</td><td>Optional override of default clearance specified by G10.7 C_</td></tr>
<tr><td>Q_</td><td>Optional override of default option flags specified by G10.7 Q_</td></tr>
</table>

Parameter	Description

G38 operations by operation code (P word):

Operation code	Parameter words	Description	Result type

G38 P0	X_ Y_ Z_	Absolute (if G90) or incremental (if G91) straight probe to given destination point. For backward compatibility with RS274NGC, the G38.2 command works the same as this. For example:
		`g38.2 x5 y-6 z40`
		is the same as
		`g38 p0 x5 y-6 z40`
			Point

G38 P1	I_ J_ K_	Incremental straight probe	Point
G38 P2	I_ J_ R_	Hole: initial direction IJ, expected radius R	Point + radius
G38 P3	I_ J_ R_ A_ B_ C_	Boss: initial direction IJ, expected radius R	Point + radius
G38 P4	I_ J_ K_ C_	Rectangle in or near XY plane: probing rectangle opposite diagonal corner displacement IJ, max perpendicular distance K (-ve)	Plane
G38 P5	I_ J_ K_ C_	Rectangle in or near YZ plane: probing rectangle opposite diagonal corner displacement JK, max perpendicular distance I	Plane
G38 P6	I_ J_ K_ C_	Rectangle in or near ZX plane: probing rectangle opposite diagonal corner displacement KI, max perpendicular distance J	Plane
G38 P7	I_ J_ R_	Slot: initial direction IJ, expected half-width R	Point + radius
G38 P8	I_ J_ R_ A_ B_ C_	Ridge or web: initial direction IJ, expected half-width R	Point + radius
G38 P9		3-plane intersection (last 3 planes)	Point
G38 P13	I_ J_ K_ C_ B_	Corner: probing rectangle extents IJK	Point
G38 P14	I_ J_ R_ C_	Sphere: initial direction IJ (from top pole), expected radius R	Point + radius
G38 P15		Circle (from last 3 points)	Point + axis
G38 P18	I_ J_ R_ C_	Sphere (high accuracy)	Point + radius
G38 P20	I_ J_ K_	3-plane (indexed)	Point
G38 P21	I_ J_ K_	Circle (indexed)	Point + axis

Operation code	Parameter words	Description	Result type

Probe Result Storage

If a G38 probing operation is successful, the status code will be zero, and zero will be stored in #5060. Otherwise, the non-zero status code (as documented in the M100 section) is stored in #5060 and no further action is taken. Typically, a failed probe operation will result in a prompt in the CNC program, and the program will be halted. An option can be set (using the Q word) which allows the program to continue after a failed probe. In this case, the program should test for a non-zero status code and take appropriate action.

If the probe operation is successful, the following hashvars are stored:

Hashvar	Description
5060	Status code (=0 for success)
5061	X,Y,Z coordinate of point result, converted to current program units.
5062	
5063	
5064	Feature dimension converted to current program units. For example, hole or sphere radius.

If the L word of the G38 command is 0-31, then the probe result is stored in the G10.8 probe data buffer entry corresponding to the L word. This result represents the same data point as stored in the hashvars, however it is stored in the table coordinate system instead of program units. The result type (point or point+direction) is also stored. The feature dimension (if any) is not stored.

The G10.8 buffer entries consist of a 3- or 6-D value plus a result type indicator.

Planar results are converted to an arbitrary point in the plane (3-D) plus a plane normal vector (also 3-D). Circle results are stored as the center point of the circle, plus an axis direction vector. Point results are stored as just the point with no other data, so only the first 3 elements of the buffer entry are used.

Results in the probe buffer are stored in current program units, but in the table coordinate system. Thus, it is important that any use of the probe buffer results is made without changing program units. As with any G-code program, it is thus important to set the units (G20 or G21) at the top of the program and not change it anywhere else.

Interaction With Rotary Axes

Probing operations run somewhat autonomously in the controller. During a probe operation, the interpreter waits for the result and therefore cannot run any other code (such as moving the axes).

Since probe results are stored in the table coordinate system, they are still valid when the rotary axes are moved.

If possible, it is better to perform all probing operations without moving the rotary positions in between. Although the results are valid, they will be less accurate because of the uncertainty introduced by angular errors in the rotary actuators.

Probe Result Indexing

This is a possible point of confusion:

There are two probe result buffers. One exists on the controller, and the other is maintained by the interpreter (the G10.8 probe result buffer).

The G38 L word is used to index the destination in the G10.8 buffer.

On the other hand, the probe operations which rely on combining previous results either index the controller buffer directly (P20 and P21) or rely on the “most recent” results in the controller buffer (P9 and P15).