

ISOCNC Project

ISO listing – Specifications

This document describes the programming language interpreted by ISOCNC

Version: **1.1.1.0**

Update: **14.02.2017**

I.1. ISO Program. Accepted characters. Uppercase and lowercase letters	2
I.2. Format of an ISO line	3
I.2.0. Line numbers	3
I.2.1. Numerical values	3
I.2.2. Words	3
I.2.3. Comments	4
I.3. Modal and Modeless G's	5
Table 2. Groups of modal G's interpreted by ISOCNC	6
Table 3. Modeless G's interpreted by ISOCNC	6
I.3.1. Unit of measure for linear axes (G70, G71)	6
I.3.2. Frames of reference: machine frames of reference, workpiece frames of reference (G9) and relative frames of reference (G24, G27, G28, G29).....	7
I.3.3. Absolute (G90) and incremental (G91) dimensions	10
I.3.4. Movement G's (G00, G01, G02, G03)	11
I.3.4.1. Quick positioning (G00)	11
I.3.4.2. Linear interpolation (G01)	11
I.3.4.3.a. Circular interpolation (helical), clockwise and counter-clockwise (G02, G03), and plane selection (G17, G18, G19)	12
I.3.4.3.b. Helical interpolation	14
I.3.4.3.c. Circular interpolation (helical) with a variation of the position of the rotary axes.....	15
I.3.5. Type of interpolation (G48, G49)	15
I.3.6. Feed rate assignment mode (G93, G94, F)	15
I.3.7. Stop at the end of the block (G09).....	16
I.3.8. Application of MDI instructions (G909).....	16
I.3.9. Hold (G04).....	17
I.3.10. Updating of technological parameters from cycle (G990).....	17
I.3.11. Set machine state – "Pending" at the end of the program (G200).....	17
I.4. M functions	17
I.4.1. Program end (M02)	17
I.4.2. Spindle start and stop (M03, M04, M05), speed control (S)	17
I.4.3. Tool change (M06)	18
I.4.4. M defined by the user	19
I.5. Instruction of conditioned jump to label – IF condition GOTO label	19
I.6. Registers.....	20
I.7. ISO errors	20

These specifications define spelling, syntax and logic rules for the ISO language interpreted by ISOCNC. Any violation of these rules, in the moment it is being identified, entails the interruption of the execution of the program and the concurrent report of the detected error to the user. Each eventual kind of error is associated to a code, a short error message, and an additional explanatory message (see I.6).

I.1. ISO Program. Accepted characters. Uppercase and lowercase letters

The programming language interpreted by ISOCNC is based on text lines ended by an end character (carriage return, line feed, or both). A line including commands and/or data for the numerical control is also called "block"; it may also contain lines that are entirely commented (see I.2.3) or blank (i.e., eventually containing only spaces and/or tab characters): in both cases, they are ignored by the interpreter. Multiple lines of code in sequence (ISO listing) make up a program, that can be stored in a text file. Below, you can see, as an example, a simple ISO program complying to the following specifications¹:

```
G0 X100.00 Y100.00 Z+5.65 ; quick positioning
G28 X100.00 Y100.00 Z0.00 ; workpiece reference system
S5000 ; rpm (setting of the spindle rotation speed)
M03 ; spindle rotation start (clockwise)
G01 Z-3.00 ; interpolation movement
; a square in plotted:
G91 (step by incremental coordinates) G1 X+10.50 (interpolation)
(a G1 is active) Y+10.50
X-10.50
Y-10.50
M05 ; spindle rotation stop
M02 ; program end
```

While reading an ISO listing, we need to distinguish between sections strictly consisting of instructions and comment sections (see I.2.3). The first ones allow only uppercase letters²; lowercase letters are allowed in the comments only. Loading an ISO file, lowercase letters outside of commented areas are converted by ISOCNC into their corresponding uppercase letters; in a guided editing, only uppercase letters are allowed. In the core listing, uppercase letters only will be displayed, and they will be stored as such saving the file. Font formatting in the comment areas is ignored.

A *word*, made of one letter followed by a number (i.e., **G02**, **T503008**, **S5000**, **M30**, **X-150.11**), can alternatively give a command or provide an argument to a command. For example, the sequence of the two words **G0 Z-5.65**, part of a valid line, defines a command (**G0**: quick movement, see I.3.4.1) and an argument of this command (**Z-5.65**: the value of Z at the end of the movement must be -5.65). An *instruction* is defined by a command and any associated argument.

¹ By using the comments (see I.2.3), you summarily indicate the actions requested to the compiler. The instructions in this short example of a listing are shown in fuller details later on in these specifications.

² For the alphabetical characters allowed, see section I.2.2.

Most instructions interpreted by ISOCNC start by **G** or by **M**. The corresponding words are called "G codes" (G stands for "general") and "M codes" (M for "miscellaneous").

Spaces and tab characters are allowed anywhere within a line of code and, within an instruction, they do not change its meaning whatsoever: e.g., writing **G 0 Z - 5 . 65** is entirely the same as writing **G0Z-5.65**.

A program consisting of blocks with their words, grouped in instructions, must obey the rules of structure, syntax and formatting. Any non-explicitly allowed item included in the instructions of the listing is reported as an error by the ISO interpreter.

I.2. Format of an ISO line

I.2.0. Line numbers

They are allowed (although they are in no way obligatory) and are specified by a sequence consisting of the character **N** followed by a non-negative integer without separators (**N20000** is valid; **N-20000** and **N20,000** are not valid)³.

The sequence that specifies the line number can be in any position within the block (on condition that it does not break the series of arguments of the G's that require them; see further), although it is usually positioned at its start. Subsequent blocks should necessarily have different, increasing line numbers. However, in this respect, no check is currently performed.

Note that the sequence that specifies the line number is not a proper word (see I.2.3), although it has its format.

I.2.1. Numerical values

A number recognized by ISOCNC is a sequence of characters possibly started by a sign ("+" or "-") and followed by a variable number of digits with an optional decimal separator. As for the format of the decimal numerical values, integers and decimals are to be separated by a dot ("."): the comma (",") is not allowed as a decimal separator. For decimal values (in module) between 0 and 1, the separator may or may not be preceded by zero (both **0.35** and **.35** are valid). There is no limit to the number of significant digits⁴.

For both integer and decimal values, any initial or final zero is simply ignored: writing **G001 Z-05.6500** or **G1 Z-5.65** is absolutely equivalent.

I.2.2. Words

A word validly interpreted by ISOCNC consists of one of the characters listed in the following table⁵ followed by a number⁶. The letter is also called *code* of the word.

Letter	Meaning
A	A axis (rotating around X), component of the \mathbf{i}_F unit vector referred to X (G24)
B	B axis (rotating around Y), component of the \mathbf{i}_F unit vector referred to Y (G24)

³ If you specify a line number in an invalid format, an error is reported [IS0056; see I.6].

⁴ A number that does not follow the aforementioned specifications generates an error report [IS0006; see I.6].

⁵ If ISOCNC detects a word that does not begin with one of the listed letters, it reports error [IS0003; see I.6].

⁶ If a letter is not followed by a number, an error is reported [IS0004; see I.6]; if there is a number that is not preceded by a letter, an error is reported [IS0004; see I.6].

C	C axis (rotating around Z), component of the \mathbf{i}_F unit vector referred to Z (G24)
F	Feed rate
G	General function (see I.3)
I	X coordinate of an arc centre (G02/G03), component of the \mathbf{j}_F unit vector referred to X (G24,G28)
J	Y coordinate of an arc centre (G02/G03), component of the \mathbf{j}_F unit vector referred to Y (G24,G28)
K	Z coordinate of an arc centre (G02/G03), component of the \mathbf{j}_F unit vector referred to Z (G24,G28)
M	Miscellaneous function (see I.4)
P	Component of the \mathbf{k}_F unit vector referred to X (G24,G28), waiting time in seconds (G04)
Q	Component of the \mathbf{k}_F unit vector referred to Y (G24,G28)
R	Radius of a circle arc (G02/G03), component of the \mathbf{k}_F unit vector referred to Z (G24,G28)
S	Spindle speed
T	Tool selection
X	X axis (linear), X coordinate of the plane origin (G24), component of the \mathbf{i}_F unit vector referred to X (G28)
Y	Y axis (linear), Y coordinate of the plane origin (G24), component of the \mathbf{i}_F unit vector referred to Y (G28)
Z	Z Axis (linear), Z coordinate of the plane origin (G24), component of the \mathbf{i}_F unit vector referred to Z (G28)

Table 1. Alphabetic characters interpreted by ISOCNC as valid starts for a word

Letter G (see I.3) can only be followed by an *integer of no more than three digits*⁷; letter M (see I.4) can only be followed by an *integer of no more than three digits*⁸. Letter T (see I.4.3) can only be followed by an *integer of exactly six digits*⁹. All other letters can be followed by both integer and decimal numbers without any length limits.

I.2.3. Comments

Comments are accepted in two forms:

- a) in round brackets: all that is within brackets, opening and closing brackets included, is ignored in the compilation. Below is an example of a valid line containing a comment in this form:

G1 X82.5393 (comment) Y87.9137 (other comment)

⁷ If this is not the case, an error is reported [ISO007; see I.6].

⁸ If this is not the case, an error is reported [ISO008; see I.6].

⁹ If this is not the case, an error is reported [ISO009; see I.6].

Z0 The round opening bracket (“(”) must correspond to the round closing bracket (“)”) within the end of the same line and vice versa¹⁰.

b) after the semi-colon (“;”), all that follows in the line is considered as commented:

X82.4165 Y88.0227 Z0 ;C48.4241 B87.742

I.3. Modal and Modeless G’s

G functions can be classified in *modal* and *modeless*.

Modal G’s

Modal G functions define a persistent operating setting: they remain “active” until they are replaced by the “use” of other specific G’s. I.e., there are typically groups of alternative and mutually exclusive G’s (*modal groups* or *families*). E.g., **G70** and **G71** (see. I.3.1) define how to interpret the values of coordinates, *as an alternative* to, respectively, imperial or SI units of measure, and constitute therefore a modal group.

For almost all families of modal G’s there is a *default* modal G that is implicitly active since the very beginning of the program (an exception is the motion group, see I.3.4): in the case of **G70** and **G71**, for example, **G71** is the one active by default (SI unit).

Reiterating the concept with other words, we can tell that the modal G’s can be logically arranged into *modal groups*: at any given moment, for each of these groups, only one member can be “in use” (in other words, it is logically impossible that two items of a modal group are active at the same time). The groups are listed in the following table (where available, the G default code is written in bold).

Group	G code	Meaning
Movement ¹¹	G00	Quick movement
	G01	Linear interpolation
	G02	Circular clockwise (helical) interpolation
	G03	Circular counter-clockwise (helical) interpolation
Selection of the circular interpolation plane	G17	XY plane (default)
	G18	ZX plane
	G19	YZ plane
Assumption of reference frame for the face	G24	General orientation of the axes: origin defined by the instruction.
	G27	Coinciding with the same as the piece (default)
	G28	General orientation of the axes: origin in the current position
	G29	Rototranslation of the reference frame
Profile reduction	G38	Inactive (default)
	G39	Active

¹⁰ If this is not the case, an error is reported [ISO001, ISO001; see I.6].

¹¹ No default.

Mill-radius compensation	G40	No compensation (default)
	G41	Left of the profile
	G42	Right of the profile
Interpolation type	G48	3 axes (default)
	G49	5 axes
Unit of measure	G70	Imperial: in, in/min.
	G71	SI: mm, mm/min. (default)
Dimensions	G90	Absolute (default)
	G91	Incremental
Assumption of reference frame of the piece	G92	G92 X0 Y0 Z0 is active by default (the frame of reference of the piece coincides with that of the machine, see I.3.2)
Feed rate	G93	Reverse of the execution time
	G94	Unit of measure/min (default)

Table 2. Groups of modal G's interpreted by ISOCNC

Special mention should be made of **G92**: the group it belongs to, as illustrated in table 2, seems to consist of one single item. This group, actually, considering that one can specify arguments for **G92** (as mentioned in § I.3.2), is made of an infinite number of items.

Non-modal G's

Other **G** functions shown as *modeless* are only active in the block inside which they are programmed.

The following table shows the modeless **G**'s interpreted by ISOCNC¹².

G code	Meaning
G04	Wait
G09	Stop at block end

Table 3. Modeless G's interpreted by ISOCNC

Order of the arguments

Please, note that those **G** instructions requiring or allowing two or more arguments (**G00** [see I.3.4.1], **G01** [see I.3.4.2], **G02**, **G03** [see I.3.4.3], **G28**, **G29**, and **G92** [see I.3.2]), these arguments can be specified in any order. E.g., writing

G00 X3.57 Y-8.99 C45.7 B-18.9

or

G00 B-18.9 C45.7 X3.57 Y-8.99

is absolutely equivalent, although we recommend you establish an order and follow it constantly (also to facilitate the identification of omitted coordinates).

Below, the single groups of **G**'s are discussed according to a logical order.

I.3.1. Unit of measure for linear axes (G70, G71)

You can specify the type of unit of measure for the coordinates associated to linear axes (**X**, **Y**, **Z**, **I**, **J**, **K**, **P**, **Q**, and **R** words) or speeds (**F** words, see I.3.7) using either **G70** (Imperial: measurements in *inches*, *inches per minute*) and

¹² G04 and G09 are discussed after modal G's (see I.3.9 and I.3.8, respectively).

G71 instructions (SI: measurements in *millimetres, millimetres per minute*)¹³. As in the table 2, SI units (**G71** in use) are used by default; the use of Imperial units must be explicitly requested by means of **G70**.

G70 and **G71** can be found in any line; however, if in this there is any G which may have linear dimensions as arguments, it must necessarily precede it¹⁴. E.g., we may write

G71 G00 X2.27 Y30.99 (correct)

but not

G00 X2.27 Y30.99 G71 (wrong)

In the first case, X and Y coordinates will be interpreted in inches.

If **G70**, or **G71**, is repeated while being already in use, a warning is reported.

I.3.2. Frames of reference: machine frames of reference, workpiece frames of reference (G9) and relative frames of reference (G24, G27, G28, G29)

Piece and machine frames of reference

When starting, the control assumes that the coordinates given by the user concern the *machine frame of reference* (origin Ω_M , coinciding with the setpoint, and Cartesian orthogonal axes X_M , Y_M , and Z_M , forming a Cartesian orthogonal right-handed system).

Since, however, it may be convenient to employ a specific frame of reference for the workpiece on which time and again the working (program) is programmed, **G92** allows specifying the Cartesian orthogonal coordinates of the origin Ω_P (X_{Ω_P} , Y_{Ω_P} , and Z_{Ω_P}) – specified in the machine reference system – of a *workpiece (or program) frame of reference*, simply *translated* with reference to the first one. After line¹⁵

G92 *X*X Ω_P *Y*y Ω_P *Z*Z Ω_P

for example, the execution of **G00 X0 Y0 Z0** will bring the tool tip to the position (X_{Ω_P} , Y_{Ω_P} , Z_{Ω_P}) of the frame of reference of the machine; subsequently, and symmetrically, **G00 X-X Ω_P Y-y Ω_P Z-Z Ω_P** will bring the tool tip back to machine home position (Ω_M).

G92 instruction must be the only one in a line¹⁶: in other words, in a line in which **G92** is programmed, no other words are accepted beside those that specify the arguments. Of these, at least one is required (also a null one, if any)¹⁷; those omitted are considered null¹⁸.

As already mentioned, the axes of the piece frame of reference (X_P , Y_P , and Z_P) remain neatly parallel to their corresponding in the machine frame of reference (X_M , Y_M , and Z_M). Please, note that the arguments of the **G92** are *always absolute* (non-incremental) *coordinates*, regardless of whether **G90** or **G91** is in use (see I.3.3.)¹⁹.

¹³ As for angles, the dimensions are still expressed in degrees.

¹⁴ If this does not happen, an error is reported [IS0010; see I.6].

¹⁵ Here and on, *Italics* indicates *symbolically* a numerical value (of a coordinate or other).

¹⁶ If this is not the case, an error is reported [IS0018; see I.6].

¹⁷ In case it is missing, an error is reported [IS0025; see I.6].

¹⁸ For this instruction, but even more so for those requiring a few arguments, although it is possible to write them in any order, we recommend writing them in a univocal sequence.

¹⁹ The presence of a **G91** (or **G90**), which would offer the possibility of a misinterpretation, is not accepted anyway in a line that specifies a change of frame of reference, in deference to the general rule requiring the univocity in a line of any instruction that causes a change of frame of reference.

To go back to using coordinates of the frame of reference of the machine, simply write ²⁰

```
G92 X0 Y0 Z0
```

The frame of reference of the piece coincides by default with that of the machine (in other words, when it starts, a **G92 X Y0 Z0** is used, as specified in table 2).

Please, note that to move from a piece reference frame to another, you do not need to “disassemble” the system in use (unlike what happens for the reference frames of the face, as shown below). I.e., it is possible that two **G92** follow without passing through a **G92 X0 Y0 Z0**:

```
;activation of a piece reference frame
;origin in the point (15.33, -8.9, 25.00)
;[coordinates of the machine reference frame]:
G92 X15.33 Y-8.9 Z25.00
[execution of movement instructions with coordinates referring
the frame in use]
;activation of a piece reference frame
;origin in the point (-29.82, 0.77, -11.3)
;[coordinates of the machine reference frame]:
G92 X-29.82 Y0.77 Z-11.3
[execution of movement instructions with coordinates referring
the new frame in use]
```

Face frames of reference

Once you have set a frame of reference for the piece, it may be convenient each time (to facilitate the writing and understanding of the instructions) to take frames of reference specific to the piece faces to work. In order to do that, three modes – corresponding to **G24**, **G28**, and **G29** – are possible.

Face frames of reference: G24

First mode (**G24**) allows you to set a frame of reference whose origin is in the point with X, Y, Z coordinates and X_F , Y_F , Z_F axes, generically oriented as the Cartesian orthogonal axes of the piece frame of reference. Once the ordered triplets of the components of the three unit vectors \mathbf{i}_F , \mathbf{j}_F , and \mathbf{k}_F have been specified as (i_{XF}, i_{YF}, i_{ZF}) , (j_{XF}, j_{YF}, j_{ZF}) , and (k_{XF}, k_{YF}, k_{ZF}) respectively, with reference to the three axes X_P , Y_P , and Z_P of the piece frame of reference, we set this frame of reference with the following line

```
G24 X Y Z AiXF BiYF CiZF IjXF JjYF KjZF PkXF QkYF RkZF
```

taking as an origin (Ω_F) the position defined by coordinates X, Y, Z.

Please, note that the arguments of **G24** *always* express *absolute* (non-incremental) *coordinates*, whether or not **G90** or **G91** (see I.3.3) are in use, and that these coordinates refer to the piece frame of reference.

In a line where a **G24** has been programmed, no other words are accepted other than those that specify its arguments²¹. At least one of these for each unit vector is required (and not a null one) ²²; those omitted are considered

²⁰ We reiterate that arguments of **G92** are always absolute and referred to the machine frame of reference, not the one you are currently using.

²¹ If this is not the case, an error is reported [IS0018; see I.6].

²² In case of complete omission of the components of a unit vector, the corresponding error is reported [IS0019, IS0020, e IS0021; see I.6].

null. To define a valid frame of reference, the three unit vectors \mathbf{i}_F , \mathbf{j}_F , and \mathbf{k}_F are required to have a non-null module²³, and furthermore they must not be coplanar²⁴. With respect to the coordinates of the application point of the reference plane, if they are omitted, it is considered null.

Face frames of reference: G28

The second mode (G28) allows you to set a frame of reference whose origin is in the point where currently the tool bit is, with X_F , Y_F , and Z_F axes, generically oriented as the Cartesian orthogonal axes of the piece frame of reference. Once the ordered triplets of the components of the three unit vectors \mathbf{i}_F , \mathbf{j}_F , and \mathbf{k}_F have been specified as (i_{XF}, i_{YF}, i_{ZF}) , (j_{XF}, j_{YF}, j_{ZF}) , and (k_{XF}, k_{YF}, k_{ZF}) respectively, with reference to the three axes X_P , Y_P , and Z_P of the piece frame of reference, we set said frame of reference with the following line

G28 Xi_{XF} Yi_{YF} Zi_{ZF} Ij_{XF} Jj_{YF} Kj_{ZF} Pk_{XF} Qk_{YF} Rk_{ZF}

by taking as an origin (Ω_F) the position defined by the X, Y, Z coordinates.

In a line where a G28 has been programmed, no other words are accepted other than those that specify its arguments²⁵. At least one of these for each unit vector is required (and not a null one)²⁶; those omitted are considered null. Additionally, to define a valid frame of reference, the three unit vectors \mathbf{i}_F , \mathbf{j}_F , and \mathbf{k}_F are required to have a non-null module²⁷, and they must not be coplanar²⁸.

Face frames of reference: G29

The third mode for assuming a face frame of reference allows you to specify both the origin ($x_{\Omega F}$, $y_{\Omega F}$, and $z_{\Omega F}$) of the piece frame of reference, and the laying position of the face plane (indicating with angles C and B the orientation of the normal to the plane itself):

G29 $XX_{\Omega F}$ $YY_{\Omega F}$ $ZZ_{\Omega F}$ CC_F Bb_F

The values admitted for C's are included between -180° and $+180^\circ$, for B's between -90° and $+90^\circ$ ²⁹.

In a line where a G29 has been programmed, no other words are accepted other than those that specify its arguments³⁰.

Finally, please, note that the arguments of G29 *always* express *absolute* (non-incremental) I, whether or not G90 or G91 (see I.3.3) are in use, and that these coordinates refer to the piece frame of reference.

Switching from a frame of reference to another

It should be noted that for G24, G28, and G29, differently from G92, you cannot specify a face frame of reference while another is already being used: you

²³ Moreover, a unitary module is not strictly required. If the module is essentially null, an error is reported [ISO022; see I.6].

²⁴ If this happened, it would not be possible, of course, to define a spatial frame of reference, and therefore an error would be reported [ISO023; see I.6].

²⁵ If this is not the case, an error is reported [ISO018; see I.6].

²⁶ In case of complete omission of the components of a unit vector, the corresponding error is reported [ISO019, ISO020, and ISO021; see I.6].

²⁷ Moreover, a unitary module is not strictly required. If the module is essentially null, an error is reported [ISO022; see I.6].

²⁸ If this happened, it would not be possible, of course, to define a spatial frame of reference, and therefore an error would be reported [ISO023; see I.6].

²⁹ Indicating values outside these ranges is reported as an error [ISO060; see I.6].

³⁰ If this is not the case, an error is reported [ISO018; see I.6].

must first explicitly “disassembly” the current one, going back to the piece frame of reference using **G27**³¹. As for **G28**’s and **G29**’s, **G27** must be alone in one line, as well³².

As shown in table 2, the face frame of reference coincides with the one of the piece (**G7** in use). As regards to the above, the following listing is valid:

```
;activation of a face frame of reference:  
G28 X1.0 Y1.0 I-1.0 J1.0 R1.0  
[execution of the movement instructions with coordinates  
referring the frame in use]  
;disassembly of the system in use  
G27  
;use of another one:  
G29 X10 Y10 Z10 C-90.0 B+45.0  
[execution of the movement instructions with coordinates  
referring the frame in use]
```

It is also required that a non-obvious (i.e. coinciding with that of the piece) face frame of reference is disassembled before changing the piece frame of reference³³. Due to what just described, the following listing is valid:

```
;activation of a piece frame of reference:  
G92 X100 Y50  
;activation of a face frame of reference:  
G28 X1.0 Y1.0 I-1.0 J1.0 R1.0  
[execution of the movement instructions with coordinates  
referring the frame in use]  
;disassembly of the face frame of reference in use;  
G27  
;activation of a new piece frame of reference  
G92 X200 Y50  
[...]
```

In this case, in accordance to what mentioned above, **G27** is necessary³⁴.

G27 is normally used to “disassembly” a face frame of reference that was put in use by **G24**, **G28**, or **G29**. If in the ISO listing there is a **G27** not preceded by any **G24** (or **G28**, or **G29**), it is ignored for the purposes of computation: this circumstance is anyhow reported to the user by a warning.

I.3.3. Absolute (G90) and incremental (G91) dimensions

Using the instructions **G90** and **G91**, we can set the interpretation mode of the dimensions specified for the coordinates. As shown in table 2, absolute values (**G90**) are used by default, i.e., referred to the frame of reference currently in use. By **G91** you can express the coordinate values as algebraic increments of the current position. E.g., the following listing

```
G90 G0 X0.00 Y0.00 ;moves to (0,0)  
G1 X5.00 Y0.00 ;linear movement to (5,0)  
G1 X5.00 Y5.00 ;linear movement to (5,5)  
G1 X0.00 Y5.00 ;linear movement to (0,5)
```

³¹ If this does not happen, an error is reported [**ISO024**; see I.6].

³² If this is not the case, an error is reported [**ISO018**; see I.6].

³³ If this does not happen, an error is reported [**ISO055**; see I.6].

³⁴ The omission would lead to an error reporting [**ISO055**; see I. 6].

G1 X0.00 Y0.00 ;linear movement to (0,0)

is equivalent to the next one, as for the results:

G90 G0 X0.00 Y0.00 ;moves to (0,0)

G91 G1 X5.00 Y0.00 ;linear movement with +5 on X, Y unchanged

G1 X0.00 Y+5.00 ;linear movement with X unchanged, +5 on Y

G1 X-5.00 Y0.00 ;linear movement with -5 on X, Y unchanged

G1 X0.00 Y-5.00 ;linear movement with X unchanged, -5 on Y

G90 and **G91** instructions can be in any line: if a **G**, allowing dimensions as arguments, is present in the same line, **G90**, or **G91**, must necessarily precede it (as in the examples above)³⁵.

I.3.4. Movement G's (G00, G01, G02, G03)

Movement **G** functions form a modal group that, because of the peculiarities in comparison to other groups, does not have a default. This means that, while it is possible to specify a **G** belonging to this group in a line and then omit it in the following lines, you cannot start an ISO listing by specifying some coordinates, as a **G** were in use³⁶.

For each movement **G**, omitting explicitly a coordinate is equivalent to requesting it to be constant³⁷.

I.3.4.1. Quick positioning (G00)

G00 specifies a quick movement leading the tool tip from the current position to the position specified by the arguments, concerning the frame of reference currently in use. **G00** can be preceded by a **G90** or a **G91**, even in the same line, indicating how to interpret the values (absolute or incremental, respectively) (see I.3.3).

As arguments, **G00** requires the coordinates of the ending point (X, Y, Z, B, and C). You must specify one position, at least³⁸: those omitted are considered coincident with their initial value (when **G90** is active) or as null (when **G91** is active).

As already mentioned, this function is modal: therefore, you can write, for example, (if **G90** is theoretically in use):

G00 X0 Y0 Z0 (quick from current position to [0,0,0])

G00 X10 (quick to position [10,0,0])

Y10 (quick from [10,0,0] to [10,10,0])

Z10 (quick from [10,10,0] to [10,10,10])

if **G00** remains active, as far as it is not cancelled by a function of the same group (**G01**, **G02**, or **G03**).

I.3.4.2. Linear interpolation (G01)

G01 specifies an interpolating movement leading the tool tip from the current position to the position specified by the arguments, concerning the frame of reference currently in use and the specified speed (feed rate, see I.3.7). **G01**

³⁵ If this limit is not respected, an error is reported [IS0061; see I.6].

³⁶ If this is the case, an error is reported [IS0011; see I.6].

³⁷ Strictly speaking: if a **G90** is being used, this means that the value of an omitted coordinate is confirmed (i.e., as if the coordinate with the same value were repeated); if a **G91** is being used, the omission is equivalent to specifying its null increment.

³⁸ Without it, an error is reported [IS0026; see I.6].

can be preceded by a **G90** or a **G91**, even in the same line, indicating how to interpret the values (absolute or incremental, respectively) (see I.3.3).

As arguments, **G01** requires the coordinates of the ending point (X, Y, Z, B, and C): you must specify one position, at least³⁹. More specifically, if you are in 3-axis mode (I.3.5.), you must specify at least one linear position (X, Y, or Z)⁴⁰.

If you request a variation of the position of the rotary axes, the 5-axis interpolation mode must be enabled (see I.3.5).

As already mentioned, this function is modal; therefore, you can write, e.g.:

G00 X0 Y0 Z0 (quick from current position to [0,0,0])

G01 X0.01 (interpolation on position [10,0,0])

Y0.02 (interp. from [0.01,0,0] to [0.01,0.02,0])

Y0.04 Z-0.01 (interp. from [0.01,0.02,0] to [0.01,0.04,-0.01])

if **G00** remains active, as far as it is not cancelled by a function of the same group (**G00**, **G02**, or **G03**).

I.3.4.3.a. Circular interpolation (helical), clockwise and counter-clockwise (G02, G03), and plane selection (G17, G18, G19)

The execution of the arc of a circle on a plane that is parallel to a coordinate plane is required by means of **G02** (clockwise arc) or **G03** (counter-clockwise). The axis around which you consider the clockwise or counter-clockwise nature of the circle arc path, is therefore orthogonal to one of the coordinated planes:

- a) XY plane, if **G17** is active (which, as in table 2, is the default option);
- b) YZ plane, if **G19** is active;
- c) ZX plane, if **G18** is active.

If one among **G17**, **G18**, or **G19** is repeated while already in use, a warning is reported.

For the execution of **G02** or **G03**, the axes of the circular interpolation plane must be orthogonal⁴¹. There is no constraint, instead, for the third axis. In this regard, a potential problem arises, of course, only if **G17** is active. For instance, if **G17** is active, the unit vectors indicated as \mathbf{i}_F and \mathbf{j}_F of axes X_F and Y_F must be orthogonal, while the execution of one between **G02** and **G03** does not require from the vector unit \mathbf{k}_F any particular constraint. When **G19** and **G18** are active, the orthogonality constraint stands for \mathbf{j}_F and \mathbf{k}_F , and for \mathbf{k}_F and \mathbf{i}_F , respectively.

To define the circle arc, whose starting point is the ending point of the instruction of the previous movement, you need first to specify the ending point:

- a) when **G17** is active, with coordinates X and Y;
- b) when **G19** is active, with coordinates Y and Z;
- c) when **G18** is active, with coordinates X and Z.

Obviously, you must specify at least one of the two coordinates⁴²: the omitted one is implicitly considered as coinciding with its initial value (when **G90** is active) or as null (when **G91** is active).

³⁹ Without it, an error is reported [IS0026; see I.6].

⁴⁰ Without it, an error is reported [IS0047; see I.6].

⁴¹ Otherwise, upon processing **G02/G03** an error is reported [IS0068; see I.6].

⁴² Without it, an error is reported [IS0026; see I.6]. In addition, the absence of explication of the coordinates required by **G02** (/ **G03**), with reference to the **G17/G18/G19** in use, is reported as an error [IS0027; see I.6].

In case of five-axis interpolation, **G02** and **G03** are broken in an ordered sequence of **G01** (micro-linear interpolating segments), coherently with the maximum chord error allowed.

Having set the starting and ending points, the full specification of the circle to which the arc belongs admits then two alternative formulations⁴³:

- a) setting the value of the radius;
- b) setting the coordinates of the centre.

The data for at least one of the two formulations must be set⁴⁴. Below, both modes are separately discussed.

Assignment of a circle by specifying its radius

As for the first formulation, just to give some concrete examples, let us consider the following. Assuming that **G17** is used, the line

G02 Xx₁ Yy₁ Rr_c

assuming that at the start of the block the tool is on the point P₀ with coordinates (x₀,y₀), requires the execution of the arc of a circle with radius r_c, clockwise around an axis orthogonal to XY plane and oriented with reference to the vectorial product $\mathbf{i}_F \wedge \mathbf{j}_F$, ending on the point P₁ with coordinates (x₁,y₁) [x₁=x₁ and y₁=y₁, in **G90** mode; x₁=x₀+x₁ and y₁=y₀+y₁, in **G91** mode], so that the amplitude of the described arc is less or equal to 180°. An arc greater than 180° can be programmed only if its centre is specified, as mentioned below. For consistency, it is obviously required that $((x_1-x_0)^2+(y_1-y_0)^2)^{1/2} < 2r_c$ ⁴⁵.

Assignment of a circle by specifying its centre

Still assuming that **G17** is used, for the second formulation the line

G02 Xx₁ Yy₁ Ii_c Jj_c

assuming that at the start of the block the tool tip is placed on point P₀ with coordinates (x₀,y₀), requires the execution of an arc of a circle, clockwise around Z, whose centre is the point C with coordinates (x_c,y_c) [x_c=i_c and y_c=j_c, in **G90** mode; x_c=x₀+i_c and y_c=y₀+j_c, in **G91** mode] and ending on point P₁ with coordinates (x₁,y₁) [x₁=x₁ and y₁=y₁, in **G90** mode; x₁=x₀+x₁ and y₁=y₀+y₁, in **G91** mode]. Obviously, in this case it would make no sense indicating a value for the Z of the centre: the presence of a **Kk_c** in the previous line would be an error.

Always referring to the example in the previous ISO line, please note that⁴⁶:

- a) if x₁ is different from x₀, to specify the centre, the centre Y coordinate alone (therefore j_c) is enough: the presence of **Ii_c** is redundant (because x_c can be calculated starting from P₀, P₁, and y_c) and so can be omitted;
- b) if y₁ is different from y₀, to specify the centre, the centre X coordinate (therefore i_c) alone is enough: the presence of **Jj_c** is redundant (because y_c can be calculated starting from P₀, P₁, and x_c) and so can be omitted.

If also the redundant coordinate is given, this one is considered anyway, in order to ensure that it is correct, i.e. checking that the point (x_c,y_c) – specified in this way – is on the axis of the segment joining P₀ and P₁⁴⁷.

⁴³ Although unusual, it is possible to enter the data for both descriptions, provided that they are congruent (otherwise, an error is reported [**ISO029**; see I.6]).

⁴⁴ Without data, an error is reported [**ISO028**; see I.6].

⁴⁵ If this condition, whose geometric interpretation is obvious, is not complied, an error is reported [**ISO030**; see I.6].

⁴⁶ What is here said is discussed, with plenty of calculations, in appendix A.

⁴⁷ Any inconsistency, if detected, is reported as an error [**ISO031**; see I.6].

If **G18** were in use, similar considerations should be applied for

G02 X_{X1} Z_{Z1} I_{i_c} K_{k_c}⁴⁸

while, when **G19** is used, for

G02 Y_{X1} Z_{Z1} J_{j_c} K_{k_c}⁴⁹.

To sum up, considering all three cases:

- if **G17** is active, **I** and **J** can be given (**I_{i_c}** alone is sufficient, if y_1 differs from y_0 ; **J_{j_c}** alone if x_1 differs from x_0 . If both are specified, a correctness check is made), never **K**;
- if **G18** is active, **I** and **K** only can be given (**I_{i_c}** alone is sufficient, if z_1 differs from z_0 ; **K_{k_c}** alone if x_1 differs from x_0 . If both are specified, a correctness check is made), never **J**;
- if **G19** is active, **J** and **K** only can be given (**J_{j_c}** alone is sufficient, if z_1 differs from z_0 ; **K_{k_c}** alone if y_1 differs from y_0 . If both are specified, a correctness check is made), never **I**⁵⁰.

Please note that, on the contrary, that

- if **G17** is active: if **I** is given but not **J**, y_1 is required to necessarily differ from y_0 ⁵¹; if **J** is given but not **I**, x_1 is required to necessarily differ from x_0 ⁵²;
- if **G18** is active: if **I** is given but not **K**, z_1 is required to necessarily differ from z_0 ⁵³; if **K** is given but not **I**, x_1 is required to necessarily differ from x_0 ⁵⁴;
- if **G19** is active: if **J** is given but not **K**, z_1 is required to necessarily differ from z_0 ⁵⁵; if **K** is given but not **J**, x_1 is required to necessarily differ from x_0 ⁵⁶.

I.3.4.3.b. Helical interpolation

As a superset of the circular interpolations, you can also consider interpolations for *helical arcs*:

- around axis **Z** (active **G17**, by default). With

G02 X_{X1} Y_{y1} Z_{Z1} R_{r_c} or

G02 X_{X1} Y_{y1} Z_{Z1} I_{i_c} J_{j_c}

⁴⁸ In this case, if x_1 differs from x_0 , the centre **Z** coordinate is sufficient to specify the centre (therefore k_c): the presence of x_{i_c} is redundant (because x_c can be calculated starting from P_0 , P_1 , and z_c) and therefore can be omitted; if z_1 differs from z_0 , the centre **X** coordinate is sufficient to specify the centre (therefore i_c): the presence of k_{k_c} is redundant (because z_c can be calculated starting from P_0 , P_1 , and x_c) and so can be omitted. If also the redundant coordinate is given, this one is considered anyway to ensure that it is correct, i.e. checking that the point (x_c, z_c) , specified in this way, is on the axis of the segment joining P_0 and P_1 [**ISO031**; cfr. I.6].

⁴⁹ In this case, if y_1 differs from y_0 , the centre **Z** coordinate is sufficient to specify the centre (therefore k_c): the presence of y_{j_c} is redundant (because y_c can be calculated starting from P_0 , P_1 , and z_c) and therefore can be omitted; if z_1 differs from z_0 , the centre **Y** coordinate is sufficient to specify the centre (so j_c): the presence of k_{k_c} is redundant (because z_c can be calculated starting from P_0 , P_1 , and y_c) and so can be omitted. If also the redundant coordinate is given, this one is considered anyway to ensure that it is correct, i.e. checking that the point (y_c, z_c) , specified in this way, is on the axis of the segment joining P_0 and P_1 [**ISO031**; see I.6].

⁵⁰ Specifying a coordinate that was not provided for the centre is obviously an error and it is reported as such [**ISO038**, **ISO039** and **ISO040**; see I.6].

⁵¹ If the result is $y_1=y_0$ it is reported as error [**ISO048**; see I.6].

⁵² If the result is $x_1=x_0$ it is reported as error [**ISO049**; see I.6].

⁵³ If the result is $z_1=z_0$ it is reported as error [**ISO050**; see I.6].

⁵⁴ If the result is $x_1=x_0$ it is reported as error [**ISO051**; see I.6].

⁵⁵ If the result is $z_1=z_0$ it is reported as error [**ISO052**; see I.6].

⁵⁶ If the result is $y_1=y_0$ it is reported as error [**ISO053**; see I.6].

we specify a z_1 coordinate for the end point of the arc [$z_1=z_1$ in **G90** mode; $z_1=z_0+z_1$ in **G91** mode]. The overall change of position (z_1-z_0) is linearly accumulated along the curvilinear abscissa of the arc.

b) around axis X (**G19** active). With

G02 Xx_1 Yy_1 Zz_1 Rr_c or

G02 Xx_1 Yy_1 Zz_1 Jj_c Kk_c

we specify a x_1 coordinate for the end point of the arc [$x_1=x_1$ in **G90** mode; $x_1=x_0+x_1$ in **G91** mode]. The overall change of position (x_1-x_0) is linearly accumulated along the curvilinear abscissa of the arc.

c) around axis Y (**G18** active). With

G02 Xx_1 Yy_1 Zz_1 Rr_c or

G02 Xx_1 Yy_1 Zz_1 Ii_c Kk_c

we specify a y_1 coordinate for the end point of the arc [$y_1=y_1$ in **G90** mode; $y_1=y_0+y_1$ in **G91** mode]. The overall change of position (y_1-y_0) is linearly accumulated along the curvilinear abscissa of the arc.

Implicitly, all interpolations with **G02** and **G03** are helical: the omission of the position parallel to the rotation axis equates to confirming it as unchanged.

In case of five-axis interpolation, **G02** and **G03** are broken in an ordered succession of **G01** coherently with the maximum chord error allowed.

I.3.4.3.c. Circular interpolation (helical) with a variation of the position of the rotary axes

A further complication to the above-mentioned cases arises when a change of the position of one or both the rotary axes is indicated (this is possible only in the case of a 5-axis interpolation); e.g., like in

G02 Xx_1 Yy_1 Zz_1 Ii_c Jj_c Bb_1 Cc_1

In this case, **G02** (**G03**) is broken in an ordered succession of **G01** coherently with the maximum chord error allowed⁵⁷. The overall position changes on C (c_1-c_0) and B (b_1-b_0) are linearly accumulated along the curvilinear abscissa of the arc.

I.3.5. Type of interpolation (G48, G49)

ISOCNC controls three-axis interpolations by default, i.e., it does not allow changing the positions of the interpolating rotary axes (no constraint instead for the quick positioning). To activate an interpolation on 5 axes you need to request it explicitly by using **G49**. To return to the three-axis only interpolation, we use **G48** (in use by default)⁵⁸.

G48 and **G49** must be in their separate line⁵⁹.

If one of **G48** and **G49** is repeated while being in use, a warning is reported.

I.3.6. Feed rate assignment mode (G93, G94, F)

For the *interpolating* movements, it is also possible, or required (according to the cases, see later on) to specify a value for the feed rate⁶⁰. This is done by using the word F.

⁵⁷ Being this a five-axis interpolation, this one must be enabled (see I.3.5).

⁵⁸ Any instruction that includes an interpolation movement (**G01** or **G02/G03** with variation of the position of a rotary axis) requiring five axes without this option being enabled (**G49**) generates an error [**ISO032**; see I.6].

⁵⁹ If this does not happen, an error is reported [**ISO033**; see I.6].

⁶⁰ It is not possible to specify a feed rate for quick movements, because the value for the feed rate is meaningless.

ISO/CNC supports two modes to set the feed rate: unit per minute (**G94**, by default in use, as specified in table 2) and inverse time (**G93**).

If one between **G93** and **G94** is repeated while already being in use, a warning is reported.

In the first mode (unit per minute), the numerical value associated with an **F** word specifies that the tool tip should move to the indicated number of inches per minute (if **G70** is in use), millimetres per minute (if **G71** is in use), or degrees per minute (regardless of **G70** or **G71**), depending on which axes are moving.

In the second mode (inverse time) the numerical value associated with an **F** word specifies that the required movement should be completed in a timespan equal to that of its reciprocal (in minutes). For example, writing

G93 $X_{\Omega F}$ $Y_{\Omega F}$ $Z_{\Omega F}$ C_{CF} B_{bF}

G01 should be completed in a $\frac{1}{4}$ of a minute timespan (assuming that the speed limits for the axes involved do not require a greater timespan).

When the inverse time feed rate mode is active (**G93** in use) a value for **F** is required to be specified in each line in which **G01**, **G02**, or **G03** is present⁶¹; an **F** in this line that does not contain any of these is simply ignored [warning: not requested **F** is ignored].

A value for **F**⁶² must be specified before this first interpolated movement instruction, or at least in it.

In both cases, values for **F** must be positive⁶³.

F value makes no sense for a **G00**: if specified, it is ignored.

You are not allowed to specify the **F** value⁶⁴ more than once.

I.3.7. Stop at the end of the block (G09)

To specify a stop of the axes at the end of an interpolating block, a **G09** shall follow this block. Please, note that **G09** must immediately follow the arguments of the movement instruction, in the same or next line. In 3-axis systems (**G48** is active), the effect of **G09** differs from that in 5-axis interpolations: if you are interpolating 3 axes, as they stop, the interpolation channel is cleaned up and we will need to wait for it to be filled before the movement starts again; in 5-axis, the effect is the same as that of a movement stop, but the interpolation channel is not emptied up of all the next up movements, i.e., the movement resumes immediately when reaching the stop position.

I.3.8. Application of MDI instructions (G909)

G909 is used at the end of a block of instructions in MDI mode to guarantee its correct execution.

If the block of MDI instructions contains **M** functions only, or **G** movement instructions (**G00**, **G01**, **G02**, **G03**), it can be omitted; anyway, in these cases its presence does not generate any error.

⁶¹ If this does not happen, an error is reported [ISO041; see I.6].

⁶² If this does not happen, an error is reported [ISO066; see I.6].

⁶³ The attempt to set a negative feed rate value is reported as an error [ISO042; see I.6].

⁶⁴ If there is more than one **F** value, an error is reported [ISO065; see I.6].

I.3.9. Hold (G04)

To specify that the program remains on hold for a certain amount of time, use **G04** followed by letter **P** and the required value in seconds⁶⁵. Obviously, this value must be positive⁶⁶. No other words are accepted in the line where **G04** appears⁶⁷.

I.3.10. Updating of technological parameters from cycle (G990)

To force an update of the technological parameters from cycle and to stop the listing compilation and the execution of the ISO program until the parameters are updated, use **G990**.

I.3.11. Set machine state – “Pending” at the end of the program (G200)

[This functionality is only managed from a remote client]. In order to instruct the cycle to immediately start the next up ISO program or wait for a new program to be entered in the list, add in the program currently running the instruction **G200** before the instruction **M2**. In this way, the cycle is not interrupted and, if in the list there is another program following the current one, this will be automatically launched. Otherwise, the control board will be waiting for a new start (soft) from the remote client.

I.4. M functions

The table below lists the **M** functions, directly interpreted by ISOCNC. The meaning of these **M**'s cannot be redefined by the customer.

M code	Meaning
M02	End of the program
M03	Start of spindle rotation (clockwise)
M04	Start of spindle rotation (counter-clockwise)
M05	Stop of spindle rotation
M06	Tool change

Table 4. M functions interpreted by ISOCNC

The above listed functions are discussed below in more detail.

I.4.1. Program end (M02)

The end of the program is indicated by **M02** in an individual line⁶⁸: the presence of a **M02** is needed to every program⁶⁹. Everything that might follow an **M02** is ignored.

I.4.2. Spindle start and stop (M03, M04, M05), speed control (S)

To set the rotation speed of the spindle (rpm), use the letter **S** followed by the valued required: a positive value is required⁷⁰.

⁶⁵ It is compulsory to indicate the value for the hold timespan by means of the “P” word: if it is missing, an error is reported [ISO064; see I. 6].

⁶⁶ Specifying a negative value is reported as an error [ISO036; see I.6].

⁶⁷ If this does not happen, an error is reported [ISO037; see I.6].

⁶⁸ If this does not happen, an error is reported [ISO043; see I.6].

⁶⁹ If a listing ends without the M02, an error is reported [ISO045; see I.6].

⁷⁰ This value cannot be negative, nor null. If this was the case, an error is reported [ISO015; see I.6].

The **S** word can be specified in any line, except for those where are instructions that explicitly require to be alone. It can be written in any position, with the following exception: if a **G** with arguments is in the line, the **S** word is required to be written before the **G** or after its arguments. E.g., the following ISO lines are valid:

```
G1 X+1.01 Y-2.0 Z+5.65 S500 ;S stated after G arguments
S500 (S before G) G1 X+1.01 Y-2.0 Z+5.65
G17 S500 (S before G) G02 X100.0 Y100.0 R300
```

The following, instead, are incorrect:

```
G1 S500 (before G arguments) X+1.01 Y-2.0 Z+5.65
G1 X+1.01 Y-2.0 S500 (between G arguments) Z+5.65
G17 G02 X100.0 Y100.0 S500 (same as above) R300
```

In the first case, **G1** arguments would be reported as missing, in the second one, the presence of a coordinate (**Z+5.65** not referable to a **G** requiring it; substantially, **Y-2.0** would be interpreted as the last, properly interpreted argument of **G1**), and similarly in the third case.

If in a line there are both an **S** and a **G** for movement, the rotation value of the spindle is anyway set before the execution of the movement **G**, regardless of whether **S** is before or after **G**.

In a line, it is admitted the presence of one **S**⁷¹ alone.

The instruction to set the rotation speed of the spindle only updates the value of a GPL variable (**SValue As Double**), whose presence is required in the custom.

To control the start of the rotation, we use **M03** (clockwise rotation) or **M04** (counter-clockwise rotation); to control the stop of the spindle rotation, we use **M05**.

I.4.3. Tool change (M06)

The tool change is specified by an **M06** immediately followed by a **T** word that indicates, with its value, the number of head, toolholder and tool⁷²: **T** must be followed by a numerical code of six digits⁷³: the first digit specifies the identifier of the head (9 options: from 1 to 9), the next two digits, the identifier of the toolholder (99 options: from 1 to 99), the remaining three digits the identifier of the tool (999 options: from 1 to 999). As set out above (see I.1), for the ease of reading, you can leave spaces between the groups of digits by writing, for example, **T 1 30 207** (which is more readable than **T130207**, with the same effect anyhow). Null values are reserved to indicate missing tooling; for this reason, non-null identifier for toolholders are not admitted after a non-selected head⁷⁴, or non-null tool identifiers after a non-selected toolholder⁷⁵.

The couple **M06 T[indices]** forms an instruction.

The instruction for the tool change simply updates an array GPL variable (**TValues[3] As Integer**), whose elements match in an orderly way the

⁷¹ If in a line there is more than one **S**, an error is reported [ISO046; see I.6].

⁷² If a **T**, that must necessarily follow **M06**, is missing, an error is reported [ISO014; see. I.6]. Error reported again [ISO054; see I.6] if a **T** is not preceded by an **M06**.

⁷³ As already mentioned, (see I.1.2.2), if this constraint is not fulfilled, an error is reported.

⁷⁴ Example: **T 0 30 207**. A similar case is reported as an error [ISO012; see I.6].

⁷⁵ Example: **T 1 00 207**. A similar case is reported as an error [ISO013; see I.6].

indices of head, toolholder and tool. Control is left to GPL. Moreover, you need to specify the indices matching a head, a toolholder, and a tool that have been actually configured in the parameters⁷⁶.

Before the first movement instruction available in the ISO listing, an M06 specifying the initial tooling is required⁷⁷.

I.4.4. M defined by the user

The user can employ all the remaining M functions – codes between 1 and 999 – for private functions. You can assign parameters to individual M functions. If one or more parameters are associated to an M function (parameter limits=20, indicated by P1=value to P20=value), M function must be the only instruction written in the line of the ISO listing.

Parameter values shall be written in a variable of the ISOCNC library called "MxMFunParams[999] as double:P1 _ double:P2 _ ... _ double:P20". If the variable is not available in the library, the parameters will be ignored.

I.5. Instruction of conditioned jump to label – IF condition GOTO label

We have implemented the possibility to perform conditioned jumps to label, in order to be able to execute part of the code under prescribed conditions only. The syntax of the instruction is as follows: *IF condition to check GOTO label*.

As '*condition to check*' we mean the set consisting of: *comparison variable* (**VGPL** if you want to check a condition based on the GPL variable defined in the ISOCNC library "Var_IF[1] as double -", or you can specify the **V1-V30** register to use as comparison variable); *comparison operator* (you can compare the following operators: =, <, >, <=, >=, <>); *comparison numerical value*.

As '*label*', we mean the word identifying the line number Nxxxxx.

Where the condition is fulfilled, the compiler of the ISO listing does not execute the instructions between the line defining the jump to label and the line specified in the label of the instruction IF -> GOTO.

More specifically: you cannot jump backwards to label – the number of the line specified in the label must come after the number of the line of the instruction IF -> GOTO. This means that you cannot define multiple lines with the same line number – the line number must be univocal. The instruction IF -> GOTO must be the only instruction defined in that line.

The instruction IF -> GOTO entails a short interruption of the execution of the ISO listing: the compiler stops the compilation of the ISO listing until that instruction is reached. When it reaches the line, it values the condition in the instruction IF -> GOTO and continues the compilation, jumping to label if the condition contained in it is fulfilled.

This is an example of IF -> GOTO instruction:

```
N1 IF VGPL = 34.567 GOTO N30
```

⁷⁶ Where the tooling elements have not been configured, an error is reported [IS0067; see I.6].

⁷⁷ If a specification for an initial tooling is missing, in correspondence with a movement G, it is reported as an error [IS0044; see I. 6].

I.6. Registers

ISOCNC provides the user 30 decimal registers, numbered from 1 to 30. The user can use the instruction **V** to set the value of a register: writing **V** followed by the register number⁷⁸, the character "=", and a value (usually decimal)⁷⁹, the user can assign the required value to the register⁸⁰. E.g., **V25=78.99** assigns value 78.99 to the 25^o register.

I.7. ISO errors

Below is the list of errors that can be reported in an ISO file.

Code	Meaning
ISO001	Comment not closed. There is a comment opening brace ("(") not followed by the corresponding closing brace (")").
ISO002	Lack of comment opening There is a comment opening brace ("(") not preceded by the matching opening brace ("(").
ISO003	Word not recognized On the line, there is a word beginning by a letter not included in the specification.
ISO004	Word without number On the line, there is a letter (seems to to the beginning of a word) not followed by a number.
ISO005	Word without code ì. On the line, there is a number that is not preceded by a letter.
ISO006	Wrong numerical format. On the line, there is a number of characters between the initial letters of two subsequent words) showing an incorrect format.
ISO007	Invalid number associated with a G. G letter can be followed by an integer only with max. three digits.
ISO008	Invalid number associated with an M. M letter can be followed by an integer only with max. three digits.
ISO009	Invalid numerical sequence associated with a T. A number or a non-integer number or, although it is an integer, a 6-digit number not equal to the 6 requested digits comes after a T.
ISO010	Wrong positioning of G70/G71. In a line where there are Gs accepting linear positions as arguments, G70/G71s must come before them.
ISO011	Position or component not referable to any command. Onn a line there is a position (or a component) without that any G is used that may require it as argument.
ISO012	Head not selected. After a non-selected head, identifiers of non-null toolholders are not accepted.
ISO013	Toolholder not selected. After a non-selected toolholder, identifiers of a non-null tool are not accepted.
ISO014	Failure to specify the tooling An M06 is not followed by a T (making up his argument).

⁷⁸ A valid register number must be specified: first of all, it must be a non-negative integer (if the format is wrong, an error is reported [ISO057; see I.6]), secondly, the number must be between 1 and 30 (otherwise, an error is reported [ISO059; see I.6]).

⁷⁹ The assignment must always come after the valid indication of a register, otherwise, an error is reported [ISO063; see I.6].

⁸⁰ The presence of a "=" character in the listing, not preceded by **V**, is reported as an error [ISO058; see I.6].

IS0015	Rotation speed value of the spindle negative or null. A negative or null value is specified for an argument requiring a positive value.
IS0016	Argument repeated. The same position (or speed) is specified more than once of the arguments of the instruction.
IS0017	Lack of arguments. None of the requested arguments comes after a G or an M.
IS0018	Non-alone specification of reference system. In a line where there is an instruction specifying the use of a reference system (G27, G28, G29, G92), this instruction must be the only one.
IS0019	Lack of components of i_F unit vector. The indication of at least one non-null component of the i_F unit vector must come after a G28.
IS0020	Lack of components of j_F unit vector. The indication of at least one non-null component of the j_F unit vector must come after a G28.
IS0021	Lack of components of k_F unit vector. The indication of at least one non-null component of the k_F unit vector must come after a G28.
IS0022	Null module unit vector. You have tried to assign a unit vector by means of a actually null vector.
IS0023	Coplanar unit vector triple. You have tried to specify a spatial reference system by means of an actually coplanar unit vector triple
IS0024	Reference system of a face already in use. You have tried to specify the reference system of a face without disassembly that previously used.
IS0025	Lack of origin coordinates. You have tried to specify the reference system of a piece without giving any coordinate for the origin (at least one is requested).
IS0026	No position specified. You must specify at least one position for a motion G.
IS0027	Inconsistency on the specified coordinates for the endpoint of the arc (G02/G03). For the end of an arc you have specified some coordinates that do not belong to the set plane of interpolation.
IS0028	Lack of coordinates of the circle (G02/G03). You have not specified either radius or centre of the circle to which the trajectory of the interpolation of a G02/03 belongs.
IS0029	Inconsistent and redundant description (G02/G03). To identify a circle both coordinates of the centre and his radius are given, but both the descriptions are inconsistent.
IS0030	Radius of the circle is not sufficient (G02/G03). The specified radius defines a too little circle for the starting and end point to belong.
IS0031	Inconsistency on the two coordinates specified for the centre of the arc (G02/G03). Two coordinates are given to the centre of the circle, but the distance between the starting point and the centre differs from the distance between the same and the end point (in other words, the end point and the starting point cannot be found on an arc belonging to the circle whose centre is the point specified).
IS0032	5 axis interpolation not enabled. You have programmed a movement that would require 5 axes, but the use of them is not enabled.
IS0033	G48/G49 not on an individual line. G48/G49 are required to appear on a line without any other instruction.
IS0034	Wrong positioning of G41/G42. G41 and G42 can be found on an individual line or before a motion G.

ISO035	Wrong positioning of G40. A G40 must only appear on an individual line or be located after the last motion G for which the correction is provided (arguments included).
ISO036	Negative dwell time. specifies a negative value for a dwell time (without sense).
ISO037	Wrong positioning of G04. On the line where G04 appears, no other words are accepted.
ISO038	Coordinate not provided for the centre. G 17 is active: I and J can only be supplied, not K.
ISO039	Coordinate not provided for the centre. G 17 is active: I and J can only be supplied, not K.
ISO040	Coordinate not provided for the centre. G 19 is active: J and K can only be supplied, not I.
ISO041	Required F not specified (feed rate inverse time feed rate). When the inverse time feed rate mode is active (G93 in use) a value for F is required to be specified at each line in which a G01, G02 or G03 is present.
ISO042	Negative or null value for the feed rate. For the feed rate, a positive value is always required.
ISO043	M02 in wrong position. M02 must be on an individual line.
ISO044	Failure to specify the initial tooling. It is imperative that the first motion G is preceded by M06 specifying the indexes on the head, toolholder and tool in use.
ISO045	Missing M02. In each listing a M02 must be present concluding the processed part.
ISO046	Statement of a repeated S. On this line two or more S words are present: one only in accepted.
ISO047	Failure to specify the linear position. In 3 axes interpolation mode, one linear position at least should be specified for a G01.
ISO048	The position of the start and end points does not allow the identification of the centre depending on the function of the provided coordinate. If I is given but not J, being G17 active, y_1 must differ from y_0 .
ISO049	The position of the start and end points does not allow the identification of the centre depending on the function of the provided coordinate. If J is given but not I, being G17 active, x_1 must differ from x_0 .
ISO050	The position of the start and end points does not allow the identification of the centre depending on the function of the provided coordinate. If I is given but not K, being G18 active, z_1 must differ from z_0 .
ISO051	The position of the start and end points does not allow the identification of the centre depending on the function of the provided coordinate. If K is given but not I, being G18 active, x_1 must differ from x_0 .
ISO052	The position of the start and end points does not allow the identification of the centre depending on the function of the provided coordinate. If J is given but not K, being G19 active, z_1 must differ from z_0 .
ISO053	The position of the start and end points does not allow the identification of the centre depending on the function of the provided coordinate. If K is given but not J, being G19 active, y_1 differs from y_0 .
ISO054	T without M06. A T word appears, not preceded by M06 of which it constitutes the argument.
ISO055	It is impossible to switch to a new reference system of the piece: the reference system of the face in use. You have required to switch to a new reference system of the piece, without having previously disassembled the reference system of the face in use.
ISO056	Invalid format of the line number. You have tried to specify a line number (N "word") by giving the number in an invalid format.

ISO057	Invalid format for the number of a register. You have tried to specify the number of a register (V "word") by giving the number in an invalid format.
ISO058	Alleged alignment without specifying the register There is a sign of equality that for a correct interpretation should be interpreted by the indication of a register ("V" word) that it is missing, instead.
ISO059	Invalid number of register The number of register you have indicated is null or greater than the number of available registers.
ISO060	Invalid angular values for G29. For the angle determining the position of the XY plane of the rotated reference system, you have tried to specify the values outside the permitted ranges.
ISO061	Wrong positioning of G90/G91. In a line where there are Gs accepting positions as arguments, G90/G91s must come before them.
ISO062	More than a motion G. On a line a motion G only is accepted.
ISO063	Lack of assignment of a value to a register. The valid appearing indication of a register is not followed by his assignment of the value.
ISO064	Failure to indicate the dwell time. A G04 that is not followed by his argument appears.
ISO065	The value of F has been specified more than once. You are not allowed to specify the F value more than once.
ISO066	Failure to specify the F. Before this first instruction of interpolated movement or in the same, a value for F has to be specified.
ISO067	The indices of tooling cannot be managed. You have specified the tooling indices matching a head, or a toolholder or a tool that have not been configured in the parameters.
ISO068	Circular interpolation required in a non-orthogonal plane. Plane axes parallel to which the circular interpolation associated to the execution of the arc of a circle (or, more generally, of ellipse) must be orthogonal.

Table 5. ISO errors