

Reference: Basic G-Code

The following are a basic set of G-code descriptions from Wikipedia. For more, visit:

<https://en.wikipedia.org/wiki/G-code>.

G = address for preparatory commands. M = miscellaneous functions.

| Code | Description | Corollary info |
|------|--------------------------------------|--|
| G00 | Rapid positioning | On 2- or 3-axis moves, G00 (unlike G01) traditionally does not necessarily move in a single straight line between start point and end point. It moves each axis at its max speed until its vector quantity is achieved. Shorter vector usually finishes first (given similar axis speeds). This matters because it may yield a dog-leg or hockey-stick motion, which the programmer needs to consider, depending on what obstacles are nearby, to avoid a crash. Some machines offer interpolated rapids as a feature for ease of programming (safe to assume a straight line). |
| G01 | Linear interpolation | The most common workhorse code for feeding during a cut. The program specs the start and end points, and the control automatically calculates (interpolates) the intermediate points to pass through that yield a straight line (hence " linear "). The control then calculates the angular velocities at which to turn the axis leadscrews via their servomotors or stepper motors. The computer performs thousands of calculations per second, and the motors react quickly to each input. Thus the actual toolpath of the machining takes place with the given feedrate on a path that is accurately linear to within very small limits. |
| G02 | Circular interpolation, clockwise | Very similar in concept to G01. Again, the control interpolates intermediate points and commands the servo- or stepper motors to rotate the amount needed for the leadscrew to translate the motion to the correct tool tip positioning. This process repeated thousands of times per minute generates the desired toolpath. In the case of G02, the interpolation generates a circle rather than a line. As with G01, the actual toolpath of the machining takes place with the given feedrate on a path that accurately matches the ideal (in G02's case, a circle) to within very small limits. In fact, the interpolation is so precise (when all conditions are correct) that milling an interpolated circle can obviate operations such as drilling, and often even fine boring. Addresses for radius or arc center: G02 and G03 take either an R address (for the radius desired on the part) or IJK addresses (for the component vectors that define the vector from the arc start point to the arc center point). Cutter comp: On most controls you cannot start G41 or G42 in G02 or G03 modes. You must already have compensated in an earlier G01 block. Often, a short linear lead-in movement is programmed, merely to allow cutter compensation before the main action, the circle-cutting, begins. Full circles: When the arc start |

| | | |
|-----|---|--|
| | | point and the arc end point are identical, the tool cuts a 360° arc (a full circle). (Some older controls do not support this because arcs cannot cross between quadrants of the Cartesian system. Instead, they require four quarter-circle arcs programmed back-to-back.) |
| G03 | Circular interpolation, counterclockwise | Same corollary info as for G02. |
| G04 | Dwell | Takes an address for dwell period (may be <u>X</u> , <u>U</u> , or <u>P</u>). The dwell period is specified by a control parameter, typically set to <u>milliseconds</u> . Some machines can accept either X1.0 (<u>s</u>) or P1000 (<u>ms</u>), which are equivalent. Choosing dwell duration: Often the dwell needs only to last one or two full spindle rotations. This is typically much less than one second. Be aware when choosing a duration value that a long dwell is a waste of cycle time. In some situations it won't matter, but for high-volume repetitive production (over thousands of cycles), it is worth calculating that perhaps you only need 100 <u>ms</u> , and you can call it 200 to be safe, but 1000 is just a waste (too long). |
| G17 | XY plane selection | |
| G18 | ZX plane selection | |
| G19 | YZ plane selection | |
| G20 | Programming in <u>inches</u> | Somewhat uncommon except in USA and (to lesser extent) Canada and UK. However, in the global marketplace, competence with both G20 and G21 always stands some chance of being necessary at any time. The usual minimum increment in G20 is one ten-thousandth of an inch (0.0001"), which is a larger distance than the usual minimum increment in G21 (one thousandth of a millimeter, .001 mm, that is, one <u>micrometre</u>). This physical difference sometimes favors G21 programming. |
| G21 | Programming in <u>millimeters</u> (mm) | Prevalent worldwide. However, in the global marketplace, competence with both G20 and G21 always stands some chance of being necessary at any time. |
| G28 | Return to home position (machine zero, aka machine reference point) | Takes X Y Z addresses which define the intermediate point that the tool tip will pass through on its way home to machine zero. They are in terms of part zero (aka program zero), NOT machine zero. |
| G52 | Local coordinate system (LCS) | Temporarily shifts program zero to a new location. It is simply "an offset from an offset", that is, an additional offset added onto the <u>WCS</u> offset. This simplifies programming in some cases. The typical example is moving |

| | | |
|------------|---|---|
| | | from part to part in a multipart setup. With G54 active, G52 X140.0 Y170.0 shifts program zero 140 mm over in X and 170 mm over in Y. When the part "over there" is done, G52 X0 Y0 returns program zero to normal G54 (by reducing G52 offset to nothing). The same result can also be achieved (1) using multiple WCS origins, G54/G55/G56/G57/G58/G59; (2) on newer controls, G54.1 P1/P2/P3/etc. (all the way up to P48); or (3) using G10 for programmable data input, in which the program can write new offset values to the offset registers. ^[8] The method to use depends on shop-specific application. |
| G53 | Machine coordinate system | Takes absolute coordinates (X,Y,Z,A,B,C) with reference to machine zero rather than program zero. Can be helpful for tool changes. Nonmodal and absolute only. Subsequent blocks are interpreted as "back to G54 " even if it is not explicitly programmed. |
| G54 to G59 | Work coordinate systems (WCSs) | Have largely replaced position register (G50 and G92). Each tuple of axis offsets relates program zero directly to machine zero. Standard is 6 tuples (G54 to G59), with optional extensibility to 48 more via G54.1 P1 to P48. |
| G90 | Absolute programming | Positioning defined with reference to part zero. Milling: Always as above. Turning: Sometimes as above (Fanuc group type B and similarly designed), but on most lathes (Fanuc group type A and similarly designed), G90/G91 are not used for absolute/incremental modes. Instead, U and W are the incremental addresses and X and Z are the absolute addresses. On these lathes, G90 is instead a fixed cycle address for roughing. |
| G90 | Fixed cycle, simple cycle, for roughing (Z-axis emphasis) | When not serving for absolute programming (above) |
| G91 | Incremental programming | Positioning defined with reference to previous position. Milling: Always as above. Turning: Sometimes as above (Fanuc group type B and similarly designed), but on most lathes (Fanuc group type A and similarly designed), G90/G91 are not used for absolute/incremental modes. Instead, U and W are the incremental addresses and X and Z are the absolute addresses. On these lathes, G90 is a fixed cycle address for roughing. |

| Code | Description | Corollary info |
|------|-----------------|---|
| M00 | Compulsory stop | Non-optional—machine always stops on reaching M00 in the program execution. |
| M01 | Optional stop | Machine only stops at M01 if operator pushes the optional stop button. |
| M02 | End of program | Program ends; execution may or may not return to program top (depending on the control); may or may not reset register values. M02 was the original program-end code, now considered obsolete, but still supported for backward compatibility. ^[10] Many modern controls treat M02 as equivalent |

| | | |
|-----|--|---|
| | | to M30 . ^[10] See M30 for additional discussion of control status upon executing M02 or M30. |
| M03 | Spindle on (clockwise rotation) | <p>The speed of the spindle is determined by the address S, in either revolutions per minute (G97 mode; default) or surface feet per minute or [surface] meters per minute (G96 mode [CSS] under either G20 or G21). The right-hand rule can be used to determine which direction is clockwise and which direction is counter-clockwise.</p> <p>Right-hand-helix screws moving in the tightening direction (and right-hand-helix flutes spinning in the cutting direction) are defined as moving in the M03 direction, and are labeled "clockwise" by convention. The M03 direction is always M03 regardless of local vantage point and local CW/CCW distinction.</p> |
| M04 | Spindle on (counterclockwise rotation) | See comment above at M03. |
| M05 | Spindle stop | |
| M06 | Automatic tool change (ATC) | Many lathes do not use M06 because the T address itself indexes the turret. Programming on any particular machine tool requires knowing which method that machine uses. To understand how the T address works and how it interacts (or not) with M06, one must study the various methods, such as lathe turret programming, ATC fixed tool selection, ATC random memory tool selection, the concept of "next tool waiting", and empty tools. ^[5] |
| M07 | Coolant on (mist) | |
| M08 | Coolant on (flood) | |
| M09 | Coolant off | |
| M13 | Spindle on (clockwise rotation) and coolant on (flood) | This one M-code does the work of both M03 and M08 . It is not unusual for specific machine models to have such combined commands, which make for shorter, more quickly written programs. |
| M19 | Spindle orientation | <p>Spindle orientation is more often called within cycles (automatically) or during setup (manually), but it is also available under program control via M19. The abbreviation OSS (oriented spindle stop) may be seen in reference to an oriented stop within cycles.</p> <p>The relevance of spindle orientation has increased as technology has advanced. Although 4- and 5-axis contour milling and CNC single-pointing have depended on spindle position encoders for decades, before the advent of widespread live tooling and mill-turn/turn-mill systems, it was not as often relevant in "regular" (non-"special") machining for the operator (as opposed to the machine) to know the angular orientation of a spindle as it is today, except in certain contexts (such as tool change, or G76 fine boring cycles with choreographed tool retraction). Most milling of features indexed around a turned workpiece was accomplished with separate operations on indexing head setups; in a sense, indexing heads were originally invented as separate pieces of equipment, to be used in separate operations, which could provide precise spindle orientation in a world where it otherwise mostly didn't exist (and didn't need to). But as CAD/CAM and multiaxis CNC machining with multiple rotary-cutter axes becomes the norm, even for "regular" (non-"special") applications,</p> |

| | | |
|------|--|---|
| | | machinists now frequently care about stepping just about <i>any</i> spindle through its 360° with precision. |
| M30 | End of program, with return to program top | Today, M30 is considered the standard program-end code, and returns execution to the top of the program. Most controls also still support the original program-end code, M02 , usually by treating it as equivalent to M30. Additional info: Compare M02 with M30. First, M02 was created, in the days when the punched tape was expected to be short enough to splice into a continuous loop (which is why on old controls, M02 triggered no tape rewinding). ^[10] The other program-end code, M30, was added later to accommodate longer punched tapes, which were wound on a reel and thus needed rewinding before another cycle could start. ^[10] On many newer controls, there is no longer a difference in how the codes are executed—both act like M30. |
| M104 | Set Extruder Head Temperature | Sets the extruder Head Temperature |
| | | |
| M108 | Extruder Speed | |
| M109 | Build Plate Temperature | Sets the Build Plate Temperature if the machine has a heated build plate |