1 Coordinate Systems

The coordinate system is inherently tied up with the particular physical machine on which the G-code is intended to run. Codes like `G28` (machine zero return) are impossible to interpret in a way that would be consistent with every machine. Various codes related to the coordinate system are accepted by Ger, but "translated away."

Machines typically have several notions of coordinate system origin. In theory, "machine zero" (or "home") is a position fixed at the factory[1] and `G28` should return the cutter to that position. Another form of origin is the Part (or Program) Reference Zero (PRZ). The PRZ is the coordinate system relative to which most commands are given.

Ger defines machine zero and the (initial) PRZ to be the location of the tool immediately before the first line of the program. As with most real machines, the location of machine zero can't be changed. So, the tool always starts at (X,Y,Z) = (0,0,0). Ger *does not accept the* `G28` **`command`**; there's very little reason to return the tool to (0,0,0), and it's always possible to move there with `G00` or `G01` in the usual way.

The important point is that Ger will produce code with all coordinates given (in absolute terms) using the location of the tool at the start of the first line of the program as the orgin. To run the output code on a physical machine, the easiest thing do will *usually* be to set the PRZ to the surface of the lower-left corner of the part. As always, consider the possibility of interference and tool crashes.

1.1 Changing the PRZ

Use `G52` to make the given values, expressed relative to machine zero, the new PRZ. Thus,

```
G52 X2.000 Y-4.000 Z1.00
```

means that the location 2 units to the right, 4 units below and 1 unit higher than machine zero will be treated as the PRZ in the code that follows. Saying

```
G52 X0 Y0 Z0
```

(or simply `G52`, with no arguments) resets the coordinate system to what it was when the program started.

The `G54-G59` commands work the same way, except that the arguments come from the work offsets table. That is, `G54` (no arguments) is the same as

[1] See Peter Smid, *CNC Progrmming Handbook*, 2008 (3rd ed.), p. 153, where he says that machine zero

> ...is the position of all machine slides at one of the extreme travel limits of each axis. Its exact position is determined by the machine manufacturer and is not normally changed during the machine working life.

In my experience that is not always true. In any case, it would be impossible for Ger to reflect the quirks of every possible machine.

```
G52 Xx Yy Zz
```

where `x, y` and `z` are taken from the work offsets table.

Use `G92` to indicate that the given values should be used as the current coordinates under a new PRZ. For example,

```
G92 X1.000 Y-1.000 Z2.000
```

means to reset the PRZ so that the cutter's current location is given by $(1, -1, 2)$. Saying

```
G92 X0 Y0 Z0
```

(or simply `G92`, with no arguments) is a convenient way to make the current tool location the new PRZ.

1.2 Tool Length Offset

The `G40`, `G43` and `G49` commands are used on many machines for tool length offset (TLO). This is used to adjust for the fact that, when tools are stored in a turret, the location of the cutting edges of the various tools will differ. TLO can also be used to take cutter wear into account.

Ger accepts these commands, but they pass through unchanged, without affecting the surrounding code. In theory, Ger could have been written to simulate the effect of the TLO commands in more detail, but doing so would require that the tool table reflect the irritating real-world quirks of various cutters and turrets. The value of Ger's translation and simulation is that it allows you to avoid these irritations – or at least, to avert your eyes until they can't be avoided.