Software specifications describe:

- The requirements
- Interfaces to other software
- How the software implements the requirements

A Requirements Document Outline

The following is an outline for a requirements document that will fit most products. This document describes the product as a "black box"—that is, what the product does, not how it is done.

Overview. A brief description of the document, such as "This document describes the requirements for the ABC corporation swimming pool timer."

Related/reference documents. Related internal documents, such as product specifications, environmental specifications, and the like. Related industry specifications such as ANSI or IEEE specifications.

Specifications. These could include the following:

Agency approvals. List agency approvals that the product must meet, such as FDA requirements, IEC 950, UL 1950, shock/vibration specifications, and so forth.

Requirements. List system requirements. The following items are typical of the sort of thing that might be listed, and obviously all of these items will not apply to all products. This section is the core of the document and may run to dozens of pages.

MTBF (mean time between failure)

MTTR (mean time to repair, usually applies to products that are serviced by a field service organization)

Speed (How many things per minute/hour/day must be done?)

Operator interface (LCD? touch panel? barcode readers? mouse/keypad?)

External interfaces (interfaces to other systems, to a controlling host system, or to a slave subsystem? Ethernet? RS-232? Proprietary?)

Available options (may be lengthy if several need to be described)

Input power (list input voltages, frequencies, and current; include international requirements)

Export restrictions and requirements (applies if using controlled technology; also, requirements for the product to be marketed in certain countries may limit technology that can be used)

Input requirements (What size bottles does it use? What sizes of paper can it handle? How big or how small can the block of steel be that goes into the input hopper?)

Capacity (How many blocks of steel or bottles or pieces of paper can it handle at a time?)

Error handling (What happens if the operator puts in too many bottles or a block of steel that is too heavy? What happens if power goes off halfway through the process?)

Weight (usually applies only to large or portable products)

Size (Does it have to fit through a standard door or on a standard elevator? In a standard briefcase?)

Safety requirements (Does it have to operate in standing water with no danger of electrocution? Does it need a safety mat to stop the robotic arm when a person steps inside the fence? Are there rotating mechanisms that must be covered or stopped when a door is opened? Must the operator be protected from high temperatures?)

External interfaces (interfaces to external systems, like a 100 base-T Ethernet interface to a computer network or an IRDA interface to transfer data to and from a PC)

Note that there may be other requirements as well, such as media requirements, customer versus field engineer maintenance items, and the like. However, since we are concentrating on embedded systems, these requirements are outside the scope of this outline.

Finally, there is an additional type of requirement that deserves mention but that is outside the scope of this book. These requirements may be called "business requirements." These include such things as the requirement that the product have all the features of competitor's product X or that certain features be left off so the product won't compete with sister product Y. Like all requirements, these are sometimes hard to quantify, but they do filter down to the design requirements at some point.

In a complex design, it is often useful to include, with each requirement, a description of what drives that requirement. A requirement for an RS232 serial interface may be needed because the product must interface to product XYZ. If product XYZ becomes obsolete, or if another interface is used instead, that requirement can be deleted. Similarly, if someone suggests that the RS232 be removed, the original requirement to include it can be traced back to its source, and you can determine whether the requirement is still valid. The connection between requirements and their source can be documented in an appendix. As mentioned earlier, this can be beneficial in finding the real requirements.