



Hardware Design Guide

guide-hardware

Rev: A01

Gabriel Smolnycki

2018-11-14

Table of Contents

| | | |
|----------|------------------------------|----------|
| 1 | Introduction | 3 |
| 1.1 | The Design Process | 3 |
| 1.2 | EDA Software | 3 |
| 2 | Components | 5 |
| 2.1 | Footprints | 5 |
| 2.2 | Symbols | 5 |
| 2.3 | Devices | 5 |
| 3 | Schematics | 6 |
| 4 | PCB Layout | 7 |
| A | Glossary | 8 |

1 Introduction

The design of a high performance rocket demands an accompanying high performance control system. As a result, it is often necessary to design either semi-custom or fully custom hardware for this purpose. While it is impractical to cover the entire scope of digital and analog electronics design techniques required, this guide will attempt to convey some best practices, as they relate to hardware design and documentation.

Hardware design is a multi-stage process, all stages of which must be methodical. Unfortunately, unlike software, hardware cannot be changed at a moment's notice, and therefore more care must be taken at earlier stages to ensure proper functionality. Using proper design practices will help minimize errors, minimize wasted money spent on components and PCBs, and speed up testing and diagnosis of faults.

Hardware design must start with a list of functional requirements. This list will ultimately drive decisions throughout the design process. Following this, the design can be realized, ideally via the processes detailed below. The same process will assist in future updates, design changes or modifications, and new designs.

1.1 The Design Process

The design process as it relates to engineering has been covered in numerous publications before, and as such this guide will not go into detail, nor describe the many variations or implementations of the concept called "the design process". Instead, a brief overview is presented for posterity; any reader wishing to know more will find ample resources available.

A typical example of the design process is shown in figure 1.

Figure 1: The design process, illustrated

1.2 EDA Software

Electronic design automation (EDA), also known as electronic computer-aided design (ECAD) software, is very useful for all stages of electronic hardware design. Typical functionality includes circuit design, simulation, schematic capture, PCB layout, manufacturing, and testing, among others. Proper use of EDA tools can greatly simplify the design process, and in some cases enable design techniques which would not otherwise be possible.

Various EDA software is available, ranging from standalone schematic capture and PCB layout tools, to full-featured suites including simulation, auto-routing, RF tools, and other features, all in one integrated tool. A brief summary of tools is shown in table 1.

| Software Name | Pricing | Schematics | PCBs | Simulation |
|--------------------------------------|----------|------------|------|------------|
| Altium Designer | \$9000 | Yes | Yes | Yes |
| Autodesk EAGLE Premium | \$510/yr | Yes | Yes | Partial |
| Cadence OrCAD Professional w/ PSpice | \$10660 | Yes | Yes | Yes |
| Novarm DipTrace Full | \$995 | Yes | Yes | No |
| KiCad | Free | Yes | Yes | No |
| NI Circuit Design Suite Power Pro | \$6320 | Yes | Yes | Yes |

Table 1: Summary of popular EDA software

While any EDA software may be used, this guide will primarily refer to EAGLE terminology. It is a mature tool, with significant third-party support, and decent functionality. Additionally, it is fairly easy for new designers to learn, and is available free to students.

1.3 Design Style

Hardware design is very flexible, and not significantly constrained by anything but the designer's wishes. For consistency, it is helpful to have a style guide, similar to that used by software developers.

2 Components

Before discussing schematic or PCB layout, it is necessary to discuss electronic components, as they form the basis of such documents. Messy, inconsistent, or incorrect components will result in a messy, inconsistent, or incorrect layout. With ideal component designs, schematic and PCB layouts will be relatively quick and painless. Errors or omissions will require corrections, complicate the process, interrupt the workflow, and ultimately slow down the designer.

It is worth noting at this point that schematic symbols and PCB footprints are not fixed. Different designers may prefer different symbols for the same component, even simple components such as resistors, capacitors, and inductors. While a good designer should be able to read and understand a variety of symbol conventions, their layouts should be self-consistent, never using two different symbols when one will do. Similarly, PCB footprints may vary based on manufacturer, board house requirements, assembly methods, and once again designer preference. Ultimately, the best source of information for many of these issues is the manufacturer's datasheet, though any designs contained therein may of course be changed at the designer's discretion.

Components are generally divided into three parts. These are the schematic symbol, the PCB footprint, and the device which links the two. The schematic symbol is a functional representation of the device, and should be easily human readable, as its purpose is solely to serve the person or persons reading it. The PCB footprint represents the physical component and its connections to the PCB, and while it may have some human readable information attached, it first must be accurate to the component's package. Finally, the device is the most flexible part of the three: it may have one or multiple schematic symbols, may have one or a variety of package options, and typically contains information about usage and applications, ordering information, and external links to manufacturer information or datasheets.

2.1 Footprints

2.2 Symbols

2.3 Devices

Parameters

Reference Designators

2.4 EAGLE Style Guidelines

3 Schematics

3.1 EAGLE Style Guidelines

4 PCB Layout

4.1 EAGLE Style Guidelines

5 Manufacturing

A Glossary