

02/26/2019

PCB Design Guidelines



Kyle Dlouhy
ELEMENTAL SCIENTIFIC

TABLE OF CONTENTS

| | |
|--|-----------|
| I. ALTIUM DESIGNER GUIDELINES..... | 2 |
| a. INTRODUCTION..... | 2 |
| b. YOUTUBE TUTORIAL | 2 |
| c. SCHEMATIC DESIGN | 2 |
| i. Hierarchical vs Flat Design..... | 2 |
| ii. Net Names & Ports..... | 2 |
| iii. GitHub Libraries..... | 3 |
| iv. Commonly Used Components..... | 3 |
| v. Creating & Importing Component Libraries | 7 |
| 1. Schematic Library | 7 |
| 2. Footprint Library..... | 7 |
| 3. 3D Models | 7 |
| vi. Test Points..... | 7 |
| d. PCB DESIGN | 8 |
| i. Design Rules | 8 |
| 1. Clearance..... | 8 |
| 2. Trace Width | 8 |
| 3. Via Sizes..... | 8 |
| 4. Polygon Connection Styles | 8 |
| ii. Common Design Practices..... | 9 |
| 1. Via-in-Pad | 9 |
| 2. Thermal Vias..... | 9 |
| 3. Polygons | 9 |
| iii. PCB Layout | 10 |
| iv. PCB Routing..... | 10 |
| 1. Auto-Router..... | 10 |
| 2. Re-Arranging Components | 10 |
| 3. Routing Groups of Traces | 11 |
| v. Silkscreen | 11 |
| 1. Text Type & Sizes..... | 11 |
| 2. Silkscreen Logos..... | 11 |
| vi. Multi-Layer PCBs | 12 |
| vii. Generating Production & Assembly Files | 12 |
| 1. ActiveBOM..... | 12 |
| 2. Gerber & NC Drill Files..... | 13 |
| 3. Pick & Place Files | 13 |
| viii. GitHub Source Control..... | 13 |
| II. STM32 DESIGN GUIDELINES | 14 |
| a. INTRODUCTION..... | 14 |
| b. DECOUPLING CAPACITORS..... | 14 |
| c. 5V-TOLERANT PINS | 14 |
| d. RESET CIRCUIT..... | 14 |
| e. CLOCK CIRCUIT | 14 |
| f. BOOTLOADER CIRCUIT | 14 |
| g. STM32CUBEMX..... | 15 |
| III. PCB ORDERING & ASSEMBLY GUIDELINES..... | 16 |
| a. INTRODUCTION..... | 16 |
| b. CREATING PROTOTYPE PCBs..... | 16 |
| i. Ordering PCBs and Stencils from JLCPCB..... | 16 |
| ii. Ordering Components..... | 16 |
| iii. Soldering Prototypes..... | 16 |
| 1. Solder Paste Application..... | 16 |
| 2. Component Placement..... | 16 |
| 3. Solder Reflow | 17 |
| 4. Second Side Soldering | 17 |
| c. ORDERING PRODUCTION-QUANTITY POPULATED PCBs..... | 17 |

I. ALTIUM DESIGNER GUIDELINES

A. INTRODUCTION

This section gives guidelines and tips for designing PCBs in Altium Designer. It includes guidelines for the entire PCB design process, including schematic design, PCB layout, PCB routing, creating a BOM, and generating production and assembly files. It is not meant to be a tutorial, but merely a set of guidelines to create consistency in the boards we produce. The goal of this document is not to teach the reader how to use Altium Designer, therefore detailed explanations of *how* to do things are not always given. Rather, this document specifies *what* you should do when designing a PCB. The *how* for most topics can be found in the YouTube tutorial link below or with a bit of quick Google research.

B. YOUTUBE TUTORIAL

An Altium Designer YouTube tutorial series can be found at the link provided below. This series is very in depth. Following along with the examples is a good way to learn the design flow in Altium Designer.

https://www.youtube.com/watch?v=8FDTVCIJm-o&list=PLxx4P_PUfAcC4exFOOPouMg1ZbL_txvGi

C. SCHEMATIC DESIGN

i. Hierarchical vs Flat Design

For most PCBs, the circuit schematic will not fit neatly on one sheet. In this case, it will be necessary to split the design among multiple sheets. It must be decided whether to use a flat design or a hierarchical design.

A flat design is like cutting a large schematic sheet into smaller sheets. All sheets are on the same level. Nets between sheets are connected by assigning them the same name.

A hierarchical design uses Sheet Symbols to create parent-child relationships between sheets. Connectivity between Sheet Symbols is through Sheet Entries. Hierarchical design has the advantage of ease of reuse of subcircuits. For example, the Mother board has 8 identical device circuits to convert UART to RS485.

This page shows how to use each type of design.

[https://www.altium.com/documentation/18.0/display/ADES/\(\(Multi-Sheet+and+Multi-Channel+Design\)\)_AD](https://www.altium.com/documentation/18.0/display/ADES/((Multi-Sheet+and+Multi-Channel+Design))_AD)

ii. Net Names & Ports

In a flat design, the net names determine connectivity between sheets. A setting in Project Options can be used to allow ports to name nets if desired.

In a hierarchical design, Ports on a sheet determine the Sheet Entries that will appear if a Sheet Symbol is created for that sheet.

iii. GitHub Libraries

The GitHub repository AltiumLibraries contains component libraries as well as circuit snippets for common circuit subsystems. Clone this repository to C:\GitHub\AltiumLibraries. Clone all project repositories to C:\GitHub as well. This will allow projects created on one computer to be opened and edited on another computer with no library reference errors.

iv. Commonly Used Components

These tables list components commonly used in our designs. As a general rule, 603 size capacitors and resistors should be used where possible. Part numbers for these are not provided as there are many different part numbers for a resistor or capacitor of the same ratings. Sometimes, a higher voltage rating or power rating is desired and is not available in 603. In this case, 805 or 1206 sizes may be used. The Altium libraries Panasonic_ERJ_603, Panasonic_ERJ_806, etc., provide resistor libraries for nearly all resistor values and power ratings. Likewise, the libraries capacitor_murata_603, capacitor_murata_805, etc. provide capacitor libraries. Using these part libraries does not mean the actual component ordered must match this part number. Simply find an identical part in the ActiveBOM.

| DIODES | | | |
|------------------|----------------------|------------------|-----------------------------------|
| Comment | Digikey PN | Manufacturer PN | Description |
| GREEN LED 603 | 732-4971-1-ND | 150060GS75000 | Green LED 603 |
| BLUE LED 603 | 732-4966-1-ND | 150060BS75000 | Blue LED 603 |
| RED LED 603 | 732-4978-1-ND | 150060RS75000 | Red LED 603 |
| SMTV01-R15G15B09 | 492-1970-1-ND | SMTV01-R15G15B09 | LIGHT PIPE ADAPTER RED/GRN/BLU |
| RGB Light Pipe | N/A (Vendor: Bivar) | FLP25-6.0 | RGB Light Pipe 6 Inch Fiber Cable |
| SK44BL-TP | SK44BL-TPMSCT-ND | SK44BL-TP | DIODE SCHOTTKY 40V 4A DO214AA |
| SD2114S040S8R0 | 478-7817-1-ND | SD2114S040S8R0 | DIODE SCHOTTKY 40V 8A SMB |
| 1N914BWT | 1N914BWTCT-ND | 1N914BWT | DIODE GEN PURP 75V 200MA SOD523F |
| SM12T1G | SM12T1GOSCT-ND | SM12T1G | TVS DIODE 12V 19V SOT23-3 |
| DF3A6_8F,LF | DF3A6.8FLFCT-ND | DF3A6.8F,LF | TVS DIODE 5V S-MINI |
| SMBJ24A | SMBJ24A-E3/5BGICT-ND | SMBJ24A-E3/5B | TVS DIODE 24V 38.9V DO214AA |

| FUSES | | | |
|--------------|-------------------|-----------------|--|
| Comment | Digikey PN | Manufacturer PN | Description |
| OZCM0010FF2G | 507-1818-1-ND | OZCM0010FF2G | Polymeric PTC Resettable Fuse 15V 100mA 1h Surface Mount 0603 (1608 Metric), Concave |
| RGEF800 | RGEF800HF-ND | RGEF800 | Polymeric PTC Resettable Fuse 16V 8A 1h Through Hole Radial, Disc |
| MF-PSML350-2 | MF-PSML350-2CT-ND | MF-PSML350-2 | PTC RESET FUSE 6V 3.5A 0805 |

| CONNECTORS & HEADERS | | | |
|---------------------------------|--------------------------|---------------------|---|
| Comment | Digikey PN | Manufacturer PN | Description |
| Pico SPOX 2-POS RA | WM7646CT-ND | 874380243 | CONN HEADER SMD R/A 2POS 1.5MM |
| Pico SPOX 4-POS RA | WM7648CT-ND | 874380443 | CONN HEADER SMD R/A 4POS 1.5MM |
| Pico SPOX 7-POS RA | WM7651CT-ND | 874380743 | CONN HEADER SMD R/A 7POS 1.5MM |
| Pico SPOX 2-POS | WM7662CT-ND | 874370243 | CONN HEADER SMD 2POS 1.5MM |
| Pico SPOX 4-POS | WM7664CT-ND | 874370443 | CONN HEADER SMD 4POS 1.5MM |
| Pico SPOX 5-POS | WM7139CT-ND | 874370573 | CONN HEADER SMD 5POS 1.5MM |
| Micro Fit 3.0 2-POS RA | WM1895CT-ND | 43650-0213 | CONN HEADER SMD R/A 2POS 3MM |
| Micro Fit 3.0 3-POS RA | WM1896CT-ND | 436500313 | CONN HEADER SMD R/A 3POS 3MM |
| Micro Fit 3.0 4-POS 2-ROW RA | WM6687TR-ND | 430450410 | CONN HEADER SMD R/A 4POS 3MM |
| Micro Fit 3.0 4-POS 2-ROW RA TH | WM3248-ND | 0430450401 | CONN HEADER R/A 4POS 3MM |
| Mini Fit Jr 3-POS RA TH | WM24038-ND | 39303036 | CONN HEADER R/A 3POS 4.2MM |
| Mini Fit Jr 4-POS RA TH | WM18447-ND | 39303045 | CONN HEADER R/A 4POS 4.2MM |
| Header 3 SMT | 952-1951-ND | M20-8770342 | CONN HEADER SMD 3POS 2.54MM |
| Shrouded Header 2x5 127 SMD | 1175-1629-ND | 3220-10-0300-00 | CONN HEADER SMD 10POS 1.27MM |
| Shunt Connector | S9337-ND | QPC02SXGN-RC | CONN JUMPER SHORTING .100" GOLD |
| PJ-002BH-SMT-TR | CP-002BHPJCT-ND | PJ-002BH-SMT-TR | CONN PWR JACK 2.5X5.5MM SOLDER |
| G51M08-P05QJ00-A080 | **N/A (Supplier - ODU)** | G51M08-P05QJ00-A080 | Fixed receptacle with two nuts (back panel mounting), 90° print contacts, Key (G), P series, Multipole Type with 5 Contacts |
| 1-1734346-1 | A114945-ND | 1734346-1 | CONN RCPT USB2.0 TYPEB 4P SMD RA |

| SWITCHES | | | |
|-------------------------------|---------------|-----------------|----------------------------------|
| Comment | Digikey PN | Manufacturer PN | Description |
| 4.5mm SMD SPST Tactile Switch | EG5350CT-ND | TL3305AF160QG | SWITCH TACTILE SPST-NO 50MA 12V |
| JS102011JCQN | CKN10721CT-ND | JS102011JCQN | SWITCH SLIDE SPDT 300MA 6V |
| CJS-1200TA | 563-1021-1-ND | CJS-1200TA | SWITCH SLIDE SPDT 100MA 6V |
| EVP-BB2A9B000 | P19888CT-ND | EVP-BB2A9B000 | SWITCH TACTILE SPST-NO 0.02A 15V |
| TL3301DF160QJ | EG5464CT-ND | TL3301DF160QJ | SWITCH TACTILE SPST-NO 0.05A 12V |
| L102021ML04Q | CKC5102-ND | L102021ML04Q | SWITCH SLIDE SPDT 4A 125V |

| MOSFETS | | | |
|------------------|-----------------------|------------------|------------------------------|
| Comment | Digikey PN | Manufacturer PN | Description |
| BSS138LT1 | BSS138CT-ND | BSS138 | MOSFET N-CH 50V 220MA SOT-23 |
| SI2308BDS-T1-GE3 | SI2308BDS-T1-GE3CT-ND | SI2308BDS-T1-GE3 | MOSFET N-CH 60V 2.3A SOT23-3 |

| MICROCONTROLLERS | | | |
|------------------|--------------|-----------------|---|
| Comment | Digikey PN | Manufacturer PN | Description |
| STM32L051K8T6 | 497-14901-ND | STM32L051K8T6 | IC MCU 32BIT 64KB FLASH 32LQFP (MFC & Mojave Boards) |
| STM32F410CBUX | 497-16240-ND | STM32F410CBU6 | IC MCU 32BIT 128KB FLASH 48QFPN (TMC Single-Axis Motor Board) |
| STM32H743VIT6 | 497-17720-ND | STM32H743VIT6 | IC MCU 32BIT 2MB FLASH 100LQFP (Mother Board) |

| POWER MANAGEMENT | | | |
|------------------|----------------|-----------------|---|
| Comment | Digikey PN | Manufacturer PN | Description |
| TPS54383PWPR | 296-23087-1-ND | TPS54383PWPR | IC REG BUCK ADJ 3A DL 14HTSSOP |
| LMZM23600V3SILR | 296-51379-1-ND | LMZM23600V3SILR | PWR MGMT SWITCHING REGULATOR 3.3V 500mA |
| LMZM23600V5SILx | 296-51380-1-ND | LMZM23600V5SILR | PWR MGMT SWITCHING REGULATOR 5V 500mA |
| TPS5405DR | 296-30579-1-ND | TPS5405DR | IC REG BUCK 5V 2A 8SOIC |

| CRYSTALS/OSCILLATORS | | | |
|----------------------------|---------------|----------------------------|-------------------------------------|
| Comment | Digikey PN | Manufacturer PN | Description |
| NX3225GD-8MHZ-STD-CRA-3 | 644-1178-1-ND | NX3225GD-8MHZ-STD-CRA-3 | CRYSTAL 8.0000MHZ 8PF SMD |
| SG5032CCN 16.000000M-HJGA3 | SER4363CT-ND | SG5032CCN 16.000000M-HJGA3 | XTAL OSC XO 16.0000MHZ CMOS SMD |
| FA-238_25.0000MB-K3 | SER3692CT-ND | FA-238 25.0000MB-K3 | Crystals 25MHz 50ppm 10pF -20C +70C |

| RS485 Transceivers | | | |
|--------------------|------------------------|-------------------|----------------------------------|
| Comment | Digikey PN | Manufacturer PN | Description |
| LTC2850CS8_PBF | LTC2850CS8#PBF-ND | LTC2850CS8#PBF | IC TXRX RS485 20MBPS 8-SOIC |
| ADM4853ACPZ-REEL | ADM4853ACPZ-REEL7CT-ND | ADM4853ACPZ-REEL7 | IC TXRX RS485/422 10MBPS 8-LFCSP |
| LTC2850CDD#PBF | LTC2850CDD#PBF-ND | LTC2850CDD#PBF | IC TXRX RS485/RS422 8-DFN |

| MISCELLANEOUS | | | |
|------------------|--------------------------|---------------------|--|
| Comment | Digikey PN | Manufacturer PN | Description |
| Test Point | A106145CT-ND | RCU-0C | PC TEST POINT NATURAL |
| LTC6994IS6-1_TRM | LTC6994CS6-1#TRMPBFCT-ND | LTC6994CS6-1#TRMPBF | IC DELAY BLOCK 8TAP PROG TSOT23 |
| TC7W7FK,LF | TC7W74FKLFCT-ND | TC7W74FK,LF | IC FF D-TYPE SNGL 1BIT US8 |
| MCP4726A0 | MCP4726A0T-E/CHCT-ND | MCP4726A0T-E/CH | IC DAC 12BIT NV EEP I2C SOT-23-6 Address: 0x60 |
| MCP4726A1 | MCP4726A1T-E/CHCT-ND | MCP4726A1T-E/CH | IC DAC 12BIT NV EEP I2C SOT-23-6 Address: 0x61 |
| NC7SZ04P5X | NC7SZ04P5XCT-ND | NC7SZ04P5X | IC INVERTER 1CH 1-INP SC70-5 |
| NC7SZ157P6X | NC7SZ157P6XCT-ND | NC7SZ157P6X | IC MUX UHS NONINVERT 2INP SC70-6 |
| MAX14759ETAT | MAX14759ETA+TCT-ND | MAX14759ETA+T | Analogue Switch, SPST, 1, 1 ohm, 3V to 5.5V, TDFN, 8 |
| M24C08-FMH6TG | 497-17568-1-ND | M24C08-FMH6TG | IC EEPROM 8K I2C 400KHZ 5UFDFPN |
| TMC5161 | 1460-1353-1-ND | TMC5161-AA-T | STEPPER MOTOR CTRLR/DVR SPI |
| RN41-I/RM | 740-1007-ND | RN41-I/RM | RN-41 Series 100 Meter Range 3 Mbps Class 1 Bluetooth Module |
| SA555DR | 296-14635-1-ND | SA555DR | IC OSC SGL TIMER 100KHZ 8-SOIC |
| FT232RQ-REEL | 768-1008-1-ND | FT232RQ-REEL | USB Full Speed to Serial UART IC, Includes Oscillator and EEPROM - QFN-32 |
| VNH5019A-E | 497-13073-1-ND | VNH5019ATR-E | VNH5019A Series 24 V 30 A Fully Integrated H-Bridge Motor Driver-MultiPowerSO-30 |
| NLAS2750MUTAG | NLAS2750MUTAGOSCT-ND | NLAS2750MUTAG | IC SWITCH DUAL SPDT 10UQFN |
| NTB0104BQ,115 | 568-8404-1-ND | NTB0104BQ,115 | IC TRNSLTR BIDIR 14DHVQFN |
| Peltier | 1487-1006-ND | 430874-503 | PELTIER CP14,199,06,L1 W4.5 |
| Thermistor | BC2715-ND | NTCLE305E4103SB | THERMISTOR NTC 10KOHM 3984K BEAD |

v. Creating & Importing Component Libraries

This link below gives a good description of how to make a component library. When making component libraries, save them in the “Libraries” folder of the “AltiumLibraries” repository. Make a new branch of the repository and merge all your new parts back into master once you’ve finished a board design and created all the parts.

[https://www.altium.com/documentation/18.0/display/ADES/\(\(A+Look+at+Creating+Library+Components\)\)_AD](https://www.altium.com/documentation/18.0/display/ADES/((A+Look+at+Creating+Library+Components))_AD)

Many components also already have schematic symbols and footprints available for download. SnapEDA, Digikey, and Altium Live all have part libraries for many components. The downloaded folder will include instructions on how to import the library into Altium. Usually, there will be a file named AltiumDesigner or something similar. That will be the name of the library project once it’s imported, so change that file name to the name you want *before* you import the part.

Components are defined in Integrated Library projects. An Integrated Library project consists of Schematic Libraries and PCB Libraries, both of which can contain one part or several parts.

1. Schematic Library

The Schematic Library is where the circuit symbol for a part is defined. This is also where you associate a footprint to a component, through the Properties tab.

2. Footprint Library

The Footprint Library defines the PCB footprint for a part. Pads and holes can be placed manually, or Altium’s built-in footprint wizard can be used to create footprints. Use the datasheet to verify the footprint is correct.

3. 3D Models

3D models of components may be added to the Footprint Library. Many parts with libraries available online include 3D models. Also, www.3dcontentcentral.com has free 3D models for nearly every footprint. A STEP file can be associated with a part by adding a 3D model to a Footprint Library. To do this, click “Place -> 3D Body”. Then hit the “Tab” key to open the properties. Choose “Generic”. This will open a window to choose a file. Navigate to the location of the STEP file and open it. Pressing “3” will switch to 3D view, where you can align the 3D model with the footprint. Pressing “2” will switch back to 2D view.

vi. Test Points

Test points should be connected to all signals of interest. This includes power (24V, 5V, 3.3V, GND, etc.), communication (RS485 D+ & D-, TX, RX, DE, SPI signals, I2C signals, etc.), and any other important nets that could be helpful for debugging. The test point

“RCU-0C” is preferred because it is very small and test leads can be connected to them easily. It is about the size of a 603 resistor.

When placing test points on the PCB, be sure to space them out enough that you will be able to clip the test leads to the board. Try to keep all test points on the top side of the board.

D. PCB DESIGN

i. Design Rules

1. Clearance

- 6 mil clearance

2. Trace Width

- Signals – 10 mils (minimum)
- Power – Depends highly on power consumption. Use an online calculator like <https://www.4pcb.com/trace-width-calculator.html> to decide minimum trace width.

3. Via Sizes

If space allows, make the hole size equal to the trace size and the diameter twice the hole size. If you must transfer a very wide trace (> 30 mils) to a different layer, use multiple vias in a grid (3x3, 4x4, etc.) to better transfer current to the other layer.

- 10 mil trace -> 10 mil hole, 20 mil diameter (smaller vias not recommended)
- 30 mil trace -> 15 mil hole, 30 mil diameter (for space-constrained boards)
- >30 mil trace -> multiple 15 mil hole, 30 mil diameter vias in a grid
 - i. Altium Designer can be tricky about placing multiple vias on a trace. Sometimes it will automatically delete redundant vias and traces. If this is a problem, two small polygons can be poured on the layers where the signal transfer will occur, and the vias can be placed here.

4. Polygon Connection Styles

The Design Rules can be used to determine how polygons connect to through-hole pads, SMD pads, and vias. To cause changes to these rules to take effect on existing polygons, click “Tools -> Polygon Pours -> Repour All”. For pads that don’t carry much current, it’s desirable to use a thermal relief connection style to make soldering easier.

- Vias – direct connect

- High-current signals ($> 1A$) – direct connect (through-hole pad & SMD pad)
- Low-current signals – relief connect (through-hole pad & SMD pad)

ii. Common Design Practices

1. Via-in-Pad

Vias placed in SMD pads are generally not recommended due to potential soldering issues. However, as a rule of thumb, it is permissible if the via size is relatively small in comparison to the SMD pad (less than half the pad size). For example, placing vias in SMD connector pads is usually OK. Also, sometimes it's necessary for dissipating heat. See Thermal Vias.

2. Thermal Vias

Some chips have a thermal pad (or multiple) on the underside. These pads should be soldered to a matching pad on the PCB. If heat dissipation is necessary (for example, for a high-current motor driver IC), vias should be placed in a grid in the pad connecting to a polygon of the same net on another layer. It's important that these vias use the direct connect polygon connection style so they draw heat out of the chip and use the polygon as a heatsink.

3. Polygons

Polygons can be used to fill the empty PCB space with copper connected to a net. For example, a GND polygon can be created on one side. Each IC that connects to GND on that side can just connect to the polygon (and will do so automatically, you shouldn't have to manually route a trace to the polygon). This makes routing easier and allows current to flow more easily for that net.

Polygons may be used for one net or for multiple nets. A GND polygon may be poured on the top and bottom. If this is done, **Via Stitching** can be used to connect the polygons together. A GND polygon may be poured on one side and a power (3.3V, 5V, 24V, etc.) polygon may be poured on another side.

Multiple polygons may be poured on one side. For example, if one area of the board uses mostly 5V and another area uses mostly 3.3V, two separate polygons can be created.

Polygons may overlap. In the Polygon Manager, polygons can be assigned priority so Designer knows which one to pour in case they overlap. For example, if a GND polygon covers the entire board, but you want a small 24V polygon underneath an IC thermal pad so that you can connect the pad to the polygon with thermal vias, you can place the 24V polygon within the GND polygon but give it a higher priority.

iii. PCB Layout

1. Place connectors on the outside of the PCB, keeping in mind how the board will be mounted in the physical product.
2. Place components in such a way as to minimize crossing nets on the ratsnest. This makes routing much easier.
3. Place components on both sides of the board.
4. If possible, place all heavy parts (connectors, large capacitors, etc.) on one side for ease of prototype soldering. When reflow soldering, the side with small parts should be soldered first. These parts will be upside down when soldering the next side, but the surface tension of the solder will be enough to hold them in place. There is no hard and fast rule as to what components will and will not hold to the board upside down. Some connectors will stay on the board even when reflowed upside down, but it's best to keep all connectors and other large parts on one side.
5. Right-click the PCB document tab and click "Split Vertical". Then open the circuit schematic page on the other tab. This helps in component placement so you know logically how the circuit *flows*.

iv. PCB Routing

1. Auto-Router

Use of the auto-router is generally not recommended by most PCB designers for any PCB design software. Many times, novice PCB designers use the auto-router as a crutch to make up for underdeveloped routing skills. As a result, they never truly learn how to route effectively.

The auto-router can be useful to help evaluate your component placement. If you run the auto-router and it completes easily, your placement is probably good. If it fails, you may want to re-arrange components.

However, using the auto-router to route the entire board and keeping the result is usually a bad idea. First, it will never route as well as an experienced designer. You can improve the routing quality using design rules, but it will still not be high quality. Second, it will inhibit development of routing skills. It's better to take a long time routing on the first few designs and learning the skill than to rely on the auto-router.

2. Re-Arranging Components

Do not be stubborn about component placement. It's easy to be reluctant about changing component placement once you have already done a great deal of routing and then you realize a different arrangement would have worked better. You may feel like all the work you did will be wasted if you undo it all and start over. However, that's how you learn and make real progress. It's better to take the time to improve your component placement than to try to plow ahead,

which usually results in large numbers of vias and traces that switch sides multiple times. As C.S. Lewis says, going back is the quickest way forward once you realize you've made a wrong turn!

"We all want progress. But progress means getting nearer to the place where you want to be. And if you have taken a wrong turning then to go forward does not get you any nearer. If you are on the wrong road progress means doing an about-turn and walking back to the right road and in that case the man who turns back soonest is the most progressive man. There is nothing progressive about being pig-headed and refusing to admit a mistake. And I think if you look at the present state of the world it's pretty plain that humanity has been making some big mistake. We're on the wrong road. And if that is so we must go back. Going back is the quickest way on." - C.S. Lewis

3. Routing Groups of Traces

Often times, you want to route a group of traces together, following the same path. For example, differential signals (such as RS485 D+ and D-) are important to route as close as possible so that they are tightly coupled (if noise is induced on one signal, it is likely induced on the other as well since they are tightly coupled. If this happens, the signal is not affected because the transceiver uses the *difference* between the two signals to determine a '1' or a '0').

To route a group of traces together, select the traces by holding SHIFT while clicking each trace. Then select "Route -> Interactive Multi-Routing". Click on one of the traces and all traces will begin routing.

v. Silkscreen

1. Text Type & Sizes

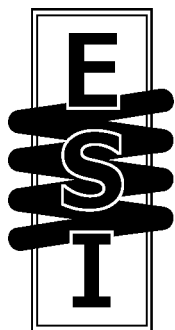
The default font type in Altium Designer is "Stroke". This is the classic, blocky PCB text. A much cleaner option is the font type "TrueType," which allows for actual fonts to be used. One of the first steps after creating the PCB document should be to select all text using the Find Similar Objects tool and convert it to TrueType Arial font. A Text Height of 45 mil is usually easily readable.

The designator should be the only text visible for each component. Additional labels, like "Power Connector" or "Reset" may be added manually. The exception to this is test points. It's much more useful to display the comment, which could be the signal name, or to manually place text describing the test point (such as "TX" or "RX"). This way, the test point can be easily identified by looking at the board.

2. Silkscreen Logos

Logos and pure black-and-white BMP images can be pasted onto the silkscreen layers. For most sources, copying the image and pasting into the silkscreen layer will *not* work properly. If that's the case, copy the image to a blank Word

document. Then, copy the image from the Word document into Altium. Once it's in Altium, it will probably be pasted as several independent shapes. To combine the image into one shape, select all the shapes. Right-click and choose "Unions -> Create Union from selected objects." Now, the image can be moved all at once. The image can be resized by selecting the image, right-clicking, and selecting "Unions -> Resize Union". The ESI logo below can be copied into Altium.



vi. Multi-Layer PCBs

Sometimes, due to space constraints or other reasons, a two-layer board is not adequate for routing all traces and maintaining signal integrity. An example is the single-axis motor board. This board contains a large number of components and connectors that must be connected on a very small board. In these cases, a multi-layer board can be used. Usually, a 4-layer board is sufficient.

A good practice is to use the internal layers for GND and power, and the outer layers for signals. One may think that the outer layers would be better for power due to the exposure to air and better heat dissipation (which is very true), but that would require a via for every single signal trace to go to the internal layers. One advantage of good component placement is that many connections between components can often be routed on one side of the board, without using any vias. Using internal layers for signals does away with that.

To use an internal layer for a GND or power signal, simply pour a polygon on the layer and connect it to the desired net. A power layer may also be split between different voltages, like 24V, 5V, and 3.3V.

vii. Generating Production & Assembly Files

1. ActiveBOM

Altium's ActiveBOM feature allows for easy BOM management. Rather than exporting a BOM and manually adding columns for manufacturer part numbers, ActiveBOM allows the user to associate an actual part from a supplier with each PCB component. It tracks unit costs and availability for each device, making it easy to see in one place the availability of all components in the design.

If the BOM needs to be emailed, it can be easily exported to an Excel spreadsheet in Altium.

2. Gerber & NC Drill Files

These files are sent to the board manufacturer to produce the blank PCB.

The Gerber files can be generated by clicking “File -> Fabrication Outputs -> Gerber Files”. Choose “Inches” for Units and “2:5” for Format. In the Layers tab, select “Used On” to include Gerber files for each layer that is used in the design. Click “OK” to generate the files. A .Cam file will be created. Save this as CAMtastic1. If you need to re-generate Gerbers, remove this file from the project and then save over it.

Generate the NC Drill Files by clicking “File -> Fabrication Outputs -> NC Drill Files”. Choose “Inches” for Units and “2:5” for Format in this window as well. Different values may be used; the important thing is they are consistent between Gerber files and NC files.

These files will be generated in the folder “Project Outputs for <Project Name>”. The Gerber files will be of type “CAMtastic <Layer> Gerber Data”. The NC drill file will be a TXT file with the same name as your project.

3. Pick & Place Files

When the PCBs are ordered in production quantities, pick and place data must be generated. This lets the company producing the PCBs know where each part goes. This is done by clicking “File -> Assembly Outputs -> Generates pick and place files”. The data can be exported in a TXT file and/or Excel spreadsheet.

The Center-X and Center-Y columns give the coordinates for the component’s center point. The Rotation column gives the rotation of the component. The Ref-X and Ref-Y columns give the coordinates for the component’s user-defined reference point. The Pad-X and Pad-Y columns give the coordinates for Pad 1 of the component. It’s a good idea to include all of these columns in the exported document.

viii. **GitHub Source Control**

GitHub should be used to store PCB projects as well as firmware. One GitHub repository should be created for each “project”. That means one repository that includes both the PCB design and firmware for one project.

The version history should be recorded in the README. PCB changes should be called “Revisions”, and firmware changes should be called “Versions”. A new branch should be made for each PCB Revision or firmware Version. Several commits should be made to the branch along the way. Once the work is finalized, it can be merged back into the master branch. If more changes are needed later, a new Revision or Version branch should be created.

II. STM32 DESIGN GUIDELINES

A. INTRODUCTION

This section includes guidelines for designing PCBs featuring STM32 microcontrollers. ST offers many different series of 32-bit ARM microcontrollers to suit different applications. They all share some design principles, and the HAL libraries are the same between series.

B. DECOUPLING CAPACITORS

Decoupling capacitors are placed near the power pins of ICs to filter out high-frequency noise in the power connection. As a rule of thumb, a 0.100 μ F capacitor should be placed next to each IC.

Each STM32 series has a hardware development application note (Example – “Getting started with STM32L0xx hardware development”). These have example circuit schematics showing what decoupling capacitors should be used. At a minimum, one 0.100 μ F capacitor should be placed next to each power pin, and one 4.7 μ F capacitor should be placed by one STM32 power pin.

C. 5V-TOLERANT PINS

Some of the STM32 I/O pins are 5V-tolerant. This means they can survive 5V voltages. For example, a 3.3V STM32 can often communicate with a 5V device via SPI, UART, etc. without the need for a logic level converter if the 5V device considers 3.3V a ‘1’. Check the datasheet for the STM32 to determine if a pin is 5V-tolerant. The abbreviation “FT” will appear in the “I/O Structure” column for the pin in the pinout table.

D. RESET CIRCUIT

In general, any microcontroller should have a reset circuit. For the STM32, this circuit should be a SPST switch connecting the RESET pin to GND and a 0.100 μ F capacitor from RESET to GND.

E. CLOCK CIRCUIT

STM32 microcontrollers usually support an external oscillator for more precise and accurate clock speeds. They also feature an internal PLL to scale up or scale down the input clock frequency. The Clock Configuration tab of STM32CubeMX is a useful tool for choosing an oscillator frequency. This article gives rules of thumb for choosing the values of capacitors for crystal oscillators. <https://blog.adafruit.com/2012/01/24/choosing-the-right-crystal-and-caps-for-your-design/>

F. BOOTLOADER CIRCUIT

An important feature of the STM32 is the ability to bootload over UART. However, the RS485 DE signal cannot be generated by the STM32 while bootloading. A special circuit using a timer IC has been designed that generates a pulse the exact length required to transmit one byte. It is triggered by the start bit on TX. This allows for bootloading over RS485 without the microcontroller generating the DE signal. A multiplexer is used to select between the automatically generated DE signal and the STM32’s DE signal. That way, the STM32 DE signal can be used during normal operation.

Most STM32 devices do not have dual-bank memory. In these devices, a jump to the bootloader code can be performed programmatically. However, some devices, like the STM32H743VIT6, have dual-bank memory. The jump to the bootloader does not work on these devices. Another method of getting to the bootloader is to connect an output pin to the BOOT0 pin, along with a 10k resistor and a 10 uF capacitor to GND. The firmware can set the output pin high, delay a short time (50 milliseconds), and then perform a firmware reset. The capacitor discharges through the resistor, and the STM32 reads a high on BOOT0 at reset, so it jumps to the bootloader. It's a good practice to include this circuit if you're unsure of whether or not a firmware jump to the bootloader can be performed.

G. STM32CUBEMX

The tool STM32CubeMX can be used to select a microcontroller to fit design requirements and to generate starter code for a project. Choose a microcontroller, perform the configuration, and output the project to SW4STM32. The project will be generated in C. To convert to C++, right-click on the project in SW4STM32 and choose "Convert to C++". Rename the "main.c" file to "main.cpp". Put code within the "User Code" comment blocks. If configuration changes are necessary, make the changes in STM32CubeMX and re-generate the project. This will recreate the "main.c" file. Use a diff checker like <https://www.diffchecker.com/diff> to determine the differences between "main.c" and "main.cpp" files. Copy any new changes in "main.c" to "main.cpp" and then delete the "main.c" file.

III. PCB ORDERING & ASSEMBLY GUIDELINES

A. INTRODUCTION

This section includes guidelines on producing prototype PCBs and ordering production quantities once a design is verified to work.

B. CREATING PROTOTYPE PCBs

Now that we have the tools (reflow oven, solder paste stencil printer, affordable PCB manufacturer) to produce affordable PCB prototypes, we assemble our own PCB prototypes. We order the blank PCBs, stencils, and components and solder the boards here. Once a design needs no more revisions, it can be ordered assembled in production quantities.

i. Ordering PCBs and Stencils from JLCPCB

We order PCBs and stencils from JLCPCB at <https://jlcpcb.com/quote>. The cost for 10 boards of one design, a stencil, and DHL shipping is usually ~\$30. The boards arrive approximately 6 days from ordering. If you don't have an account, create one at JLCPCB. Upload the Gerber and NC files. Choose to include a stencil with framework of size 400*300. Add the boards and stencil to the cart. Checkout and have Jeff or Doug add payment information. Choose DHL shipping.

ii. Ordering Components

Kyle has a collection of many components that are common to all boards. For new parts that are needed, send a spreadsheet with manufacturer part numbers and quantities to Jeff or Doug for ordering.

iii. Soldering Prototypes

Always solder the side with the light components first, because these parts will be upside-down when soldering the second side.

1. Solder Paste Application

Solder paste is applied to the PCB using the stencil and the stencil printer. The stencil clamps into the printer. The board is fixed into the platform of the printer, and the stencil is lowered over the board. Solder paste is applied using the solder paste spreader. The printer is raised up and the board is removed. Paste should be applied nearly perfectly to each pad. If the paste is smeared or not applied sufficiently, scrape the paste off the board, wipe it off with a paper towel, and try again. This step is the most critical step for ensuring all pins are soldered correctly without any bridges. It is even more important than component placement, as components placed imperfectly will "snap" into place while reflowing, as long as the paste is applied well.

2. Component Placement

Components may be placed with tweezers or the vacuum pick-up tool. It is helpful to print out a blown-up version of the PCB top and bottom to make

locating components easier. It's also helpful to make a BOM copy separated by layer, printing it out, and checking off parts as they are placed. Place all parts of one value on a side at the same time. For example, place all the 0.100 μ F capacitors on one side at the same time.

3. Solder Reflow

Once the components are placed on the PCB, place the PCB in the reflow oven. Turn the oven on and start the reflow profile.

4. Second Side Soldering

Once the first side has been soldered, repeat the process for the second side. If the mounting holes allow it, place standoffs in the mounting holes to keep the board elevated. If not, clip binder clips to the corners of the board in spots where there are no components. Don't worry, the clips will not melt.

C. ORDERING PRODUCTION-QUANTITY POPULATED PCBs

Once a PCB has been prototyped and revised as necessary, it can be ordered from CSS. Talk with Tyler to decide what quantities to get quotes for. Compile the Gerber files, pick and place data, 3D model snapshots, and BOM in one ZIP file and email to CSS. Ask for quotes at the decided quantities. Once the quotes are received, Tyler will handle purchasing the assembled boards.