# Board: LED Board

# Designer: Jack Reamy

# Reviewer: Grant Davis

# Date: 2020-02-11

|  |  |
| --- | --- |
|  | Symbols |
|  | Verify that the part can be assembled If you are soldering by hand, or sending to our assembler (who solders by hand), you want to make sure that every part can be assembled. For passives, this means 0603 or larger parts. For IC’s, you should be able to get a solder tip onto each pad/pin. If the pins are too close together (typically less than 10 mils), our assembler will not attempt it without a stencil (which can cost several hundred dollars). We used to say this …  *Large ground pads underneath the part are fine for our assembler, but if you want to solder them yourself, extend the pad beyond the outline of the part for access.*  *Finally, if you are using a QFN part (or some other package where the leads are hidden), these can be difficult for us, and our assembler recommends that you bring the pads out a bit beyond the package so that we/they can get a solder tip onto the pad.*  However, with the pick and place, reflow oven, etc., we have a better shot at these. |
|  | Verify pinouts Ensure that the pin numbers match the pin labels as shown in the datasheet. Seems obvious, but check it anyway.  **Need to switch IOREF on Regulator to 5V** |
|  | Show power and ground on schematic symbols. Sometimes symbols are designed without power and ground symbols (e.g. some Op-Amps). It is better on our schematics to indicate all power and grounds. |
|  | If a part has multi-function pins, show them on the symbol. Many parts (especially microprocessors) have pins that can be configured for different usages (e.g. general-purpose I/O, SPI interface pins, A/D, etc.). Attempt to label all of these on the symbol. When a particular usage is expected on the schematic, the symbol instance can be edited to reflect that choice. |
|  | Verify proper package type (surface mount, DIP, etc) for all ICs. A parts library can have multiple packages for the same part. Ensure that you have selected the appropriate package for the symbol. |
|  | Verify proper part number for BOM. The BOM (Bill-of-Materials) will be auto-generated. Make sure that your part number and other manufacturer information are correct. |
|  | Indicate active-low pins. Active low pins should have a bubble, or triangle, or other clear indication on the pin that it is active-low. |

|  |  |
| --- | --- |
|  | Schematics |
|  | Ensure that all components have values. Certain components (typically passives) have generic symbols. Avoid using these and ensure that the schematic identifies all values. |
|  | Ensure that components have a reference designator. Also ensure that it is visible on the schematic. |
|  | Label all power nets consistently. Use GND for ground. Use AGND for an analog ground. For single DC voltages, Vcc is common. When multiple voltages are used, the voltage value is frequently used 3V3 (for 3.3V supplies) and 5V0 (for 5V supplies). |
|  | All no-connect pins on IC's should be labeled NC This makes it clear that you intend to leave it disconnected – not that you forgot. |
|  | Show mounting holes on the schematic. Show electrical connections between mounting holes and the ground plane if they exist. Add tooling holes so they don't get lost between revisions of the layout. |
|  | Add fiducial markers If the board is going to be professionally assembled, fiducial markers are general required. We recommend at least 3 placed near three corners. |
|  | Ensure off-page designators for each signal going either on or off pages. Attempt to group common signals on the same schematic page, but if they must change pages, be sure to add page designators. |
|  | Make sure to show unused circuits for multi-circuit packages. Certain IC’s (like some Op-Amps or logic gates) have more than one circuit in each IC. Frequently, there is a separate symbol for each element in the package so that they can be placed in different areas of the schematic. If there are unused gates, those circuits need to be shown (not only to inform viewers, but the unused input pins generally need to be tied to something). |
|  | Connect by name on-page if it makes it more readable. Rather than draw long wires as interconnects, it is generally preferable to label signals and let the design tool perform connection by name. Similarly don't wire power/ground all over the page, use multiple power/ground symbols and/or labeled power/ground signals. |

|  |  |
| --- | --- |
|  | Pin names and attributes on symbols with multi-function pins should match actual design usage. That is, if the part is going to have multi-function capability, the pin names should list those options. If only a specific usage is used in the design, the symbol should indicate only that specific usage.  Example: In Altium, you can double click on a part and then click the "Edit Pins" dialog box to manually edit pin names and their types (e.g. Input/Output/Both...) for your specific design. |
|  | Ensure that all critical signals have test access This can be accomplished in the layout phase by providing a testable via point somewhere on the signal trace. (Directions on how to do this?) As this is a prototype board, ALL signals should have test access. Unused pins are of no use on a prototype board. If you have room, include headers for all of the pins so that you can easily make use of the pins in the event that you need to alter the design after fabrication. |
|  | Keep traces on either side of a part the same width If you don’t, one side can heat up faster than the other and it can cause tombstoning.  <http://metallicresources.com/documents/Tombstoning%20explained.pdf> |

**Change Trace with to power the LEDS to be bigger! (3V3S)**

|  |  |
| --- | --- |
|  | IC Connections |
|  | Verify Power and Ground pins Ensure that power pins are connected to the correct voltages.  **Change Regulator input to 5V, not IOREF** |
|  | Ensure that all inputs are always driven at all times. Any input that is not driven can float and can damage an IC. Watch carefully for inputs that are driven by a microprocessor or other IC that has programmable I/O pins. Typically, when a microprocessor resets, all of its I/O pins initialize as inputs. This means that you cannot count on the microprocessor to always be driving, and you should have a pull-up (or pull-down) on any input line coming from a microprocessor (some microprocessors have internal pull-ups for this purpose). This is especially true for prototype boards where the firmware is not yet verified. |
|  | Ensure that any net that can possibly be dual-driven has a limiting resistor A particular chip might be set up to drive a pin of the microprocessor. However, improper programming of the microprocessor could lead to this pin being an output of the microprocessor and thus, two chips will be driving the net, possibly damaging one or more of the IC’s. Therefore ensure that there is a current-limiting resistor wherever this can occur. By definition, any connections between two microprocessors with configurable I/O pins should have current-limiting resistors as they can both be configured accidentally as outputs. |
|  | Appropriate bypass caps for each IC As a rule, use one 0.1uF capacitor for every IC power pin and one bulk 10uF or 100uF for every 10 to 20 ICs. However – some IC’s require either/both small and large caps (large caps for large current change requirements, small for rapid current change requirements). Check the datasheet for each IC to see if there are any special instructions for bypass capacitors. |
|  | Avoid direct connect of mode pins to GND or VCC. Rather, connect them through a 0 ohm resistor (surface mount preferable) and, if possible, provide test vias for both sides of the net. This makes debugging a prototype far simpler in the event of an error. (Note – this is a bad idea for a final board as it puts extra parts onto the board. Consider the assembly cost as well – a small resistor that costs you less than $0.01 can cost you significantly more to be assembled onto the finished board). |
|  | Ensure that unused inputs are driven Generally, IC’s that have optional inputs will provide internal pull-ups or pull-downs. But check to make sure that every input is going to be driven. For most digital inputs, either pulling them high or low is sufficient. Analog inputs might require a different approach.  See [this link from analog devices](http://www.analog.com/static/imported-files/rarely_asked_questions/unused_op-Amp_article.html) for a discussion of how to tie unused op-amp inputs. |
|  | ISP reset lines are unique If a system has multiple microprocessors, ensure that ISP connections are constructed so that chips do not interfere with one another when one of them is being programmed. If both chips are connected to the same ISP lines and the same reset, then both will respond to the programmer. |
|  | SPI chip selects are pulled to an inactive state The AVR microcontrollers use the SPI lines for programming via the ISP interface. When the chip is reset, all of its pins are set to inputs. Therefore, to prevent any other IC’s from interacting with the programming of the microcontroller, the SPI chip selects should be pulled up or down via a resistor to ensure that they are inactive unless the microprocessor is specifically selecting them. |
|  | Control power-up states. Make sure that your system will power up into a known state. You will need to check the timing of your reset circuits to ensure that all of your devices do not require interactions before other parts are powered up. You can reference the “Start-up Times” in the “Clock Options” section of your microcontroller datasheet. |
|  | Reset designs are glitch-free. Debounce any mechanical reset using an RC circuit or a specially designed reset IC (or set your microcontroller start-up time to be beyond any bounce time). |
|  | Ensure that any clock values are correct. Make sure that any clock values are appropriate for the chips being used. Do not trust default values copied from other schematics. |
|  | Ensure that voltage ranges are appropriate for all signal lines. The I2C interface to Chirps requires a 3.3V interface. The ATMega microcontrollers can reliably run at 12 MHz on a 3.3V source. If you need to run faster, you must use a 5V source. To talk with the I2C, an I2C converter should be used. Some parts will require level translators to communicate because of different core voltages. Look that the min/max Vih/Vil and Voh/Vol on your various parts to see where level translators are required. |

|  |  |
| --- | --- |
|  | Board connectors |
|  | Verify that the connector pinout matches mating connector. Be sure to check the direction. It is easy to flip connectors and have them backwards when you go to assemble. |
|  | Appropriate ESD protection Provided. Whenever contact (especially human contact) is expected with the system electronics (connectors being the primary mode), ensure that you have adequate electro-static discharge (ESD) protection. Many standard busses (USB, Ethernet) have standardized ESD requirements.  For a basis, attempt to meet IEC 61000-4-2, which is explained here - <http://www.onsemi.com/pub_link/Collateral/TND410-D.PDF> and <http://www.littelfuse.com/data/en/Application_Notes/an9612.pdf>, but can be roughly be stated to require 15KV for an air discharge. |
|  | Filters provided to avoid radio frequency interference (RFI). Follow the data sheets any wireless device. Additionally, it can be a good idea to place extra 33pF shunt caps to ground (X5R or X7R ceramics work well). This is especially important on power pins and long signal traces (those longer than ¼ wavelength). It can also be good to add series ferrites such as BLM15HG601SN1 to these same types of traces if noise is an issue. Adding in a 0 ohm series resistor is a good idea in the event that you want to later replace it with the ferrite. |
|  | Ensure that signals and power have adequate return paths. Remember that electrical current flows in a loop. Any signal in a connector must have a sufficient return path to the system.  **Do 3V3S again** |
|  | Attempt to use connectors that ground first when connecting and lose ground last when disconnecting. This allows hot-plugging. |

|  |  |
| --- | --- |
|  | Passive Components |
|  | Verify that your components are 0603 or 0805 (or larger). Unless you have a very specific need for a very small layout, you should not have a need to use 0402 or smaller parts on a prototype like this. If you choose to use smaller parts, the assembler might have to purchase a stencil which can run several hundreds of dollars. |
|  | Verify voltage ratings for all caps. Derate at 25% of the required voltage.  *Voltage derating is expressed as the percentage that the applied voltage is less than rated voltage, e.g., a 450 V capacitor operating at 400 V would have 11% voltage derating. So, if you need 100V across your capacitor, find one that can tolerate at least 133.3V.* |
|  | Verify polarization for all capacitors. |
|  | Verify power ratings for resistors. |
|  | In general, use ceramic capacitors when possible, then tantalums, then aluminum electrolytics only if necessary. Unless you are in the group that prefers electrolytics to tantalums ☺ This is a rather gross over-generalization. You can read a lot more about capacitor types here: <http://www.radio-electronics.com/info/data/capacitor/capacitor_types.php>. |

|  |  |
| --- | --- |
|  | Power |
|  | In general, use a switching regulator for primary power To allow for wider voltage input ranges and higher current outputs without generating too much heat, switching regulators are recommended as the primary voltage regulation for your board. |
|  | in general, use linear regulators for secondary power If you have a 5V core running off of a switching regulator and you require a 3.3V power source for a small-current supply for a particular chip, use a small, linear regulator. The low current draw and small voltage drop (5V-3.3V) is a low-power requirement and a perfect application for a linear. |
|  | Ensure that your power supplies have enough margin. Example: If you are using a 3.3V switching regulator to power a 1.8V linear regulator, make sure that your linear has enough margin to power the parts of interest. If you are powering a microcontroller that has a 1.8V minimum, your linear regulator should generally be set for 1.9V or 2V to guarantee that the output voltage will be enough to power the system. |
|  | Do we need to fuse the power? If you are utilizing power from an expensive source (such as a USB connection), it is a good idea to fuse the power. |
|  | Is there sufficient power available for the board? Estimate the maximum power draw for all components and ensure that your power supply can deliver the required current. |
|  | Are the power lines capable of carrying the required current? Will you be able to accommodate the power with the traces or power planes on your PCB?  **Ensure ground and IOREF and 5V and 3V3S are big enough** |
|  | Do you have a Power LED? I mean, come on … you are an EE aren’t you? Unless you have some specific low-power requirements, put a power LED on that bad boy. LED’s are cool. |
|  | Is any cooling required (heatsinks, flourinert)? Ensure that any high-power devices (specifically power supplies) have adequate cooling. |
|  | Ensure sufficient capacitance on low dropout voltage regulators. Just to double-check. |

|  |  |
| --- | --- |
|  | Analog elements |
|  | Separate analog signals from noisy or digital signals. Switching power supplies can induce noise in analog systems. Analog power systems (motors, etc.) can induce noise in digital systems. It is best to electrically isolate these systems as much as possible on your PCB. |
|  | Use separate voltage planes for analog and digital supply voltages. If you can’t use separate planes, use separate areas of your pcb and only connect them at one point. This ensures that fluctuations in area A will generally be served by capacitors in area A and area B can remain mostly isolated from the these fluctuations. |
|  | Understand ground requirements for a mixed analog and digital system. Certain systems recommend that the grounding for analog and digital systems be shared to provide a direct return path for signals that move from the analog areas to the digital areas. However, other mixed signal designers with more sensitive analog systems desire separate grounds for analog and digital. It doesn’t appear to be a “one-size-fits-all” problem. So understand your circuit situation and research different approaches. |

|  |  |
| --- | --- |
|  | Prototype Design |
|  | When using surface mount components, try to use packages with larger pin spacing. Observe the specifications of the fabrication equipment you will be using and ensure that your parts can be fabricated using that equipment. For example, the Voltera V-One printer has a minimum pin-to-pin pitch of 32 mil and a minimum trace width of 8 mil. |
|  | Socket microprocessors if possible. If you are using a DIP microprocessor, solder in a socket so that you can change the microprocessor if required. If surface mount, buckle up. |
|  | Get processors as large as reasonable for memory and I/O needs. Increased memory is generally cheap. If you think an ATMega168 is close to what you need, order an ATMega328. Better safe than sorry. |
|  | Make headers and/or test points available as much as possible. You want as much flexibility as possible on a prototype. If you have room, pull signals out to headers for easy debug. You don’t have to populate the headers on your board – but you will be thankful you can when you get involved in some debug. You should definitely have some headers for SPI and I2C lines so that you can easily hook up logic analyzers. |
|  | PCB Layout |
|  | Place test pad or test via on every net for in circuit test For those signals that you cannot get to a header or test pin, place a via so that you at least have access for electrical testing and as a potential solder point if you have to cut a trace (two vias per connection often work better for this as you can cut the middle trace and then jumper the wires from the vias. The smallest wire we have available is 30AWG. For that, 10 mil hole size and 25 mil diameter is as small as you want to go. Those almost always work unless there is flux stuck in the drilled hole. Then you just need to melt the flux with a clean iron. However as a general rule, if you can get 30mil diameter holes with larger pads then you will have more flexibility with wire choices. |
|  | Ensure that all vias are through the entire board (specifically for 4 or more layer boards). Our production runs do not allow for “blind” or “buried” vias. These are vias between layers that do not go all of the way through. These can be done, but they are much more expensive. So – ALL vias have to be drilled through the entire board. |
|  | Do not place vias under components This makes the via unusable for debug, and can cause issues – especially with parts that have thermal or ground pads. |
|  | Ensure every component has a reference designator on the silkscreen. This allows placement of parts to be performed. It is essential if your board is being assembled for you. If you have room, place the designators outside of the component. |
|  | Ensure that every component has a placement marker. For IC’s, indicate pin 1 with a circle or other descriptive mark. For diodes/caps/etc., use appropriate polarity marks to indicate direction. |
|  | Place a logo in silkscreen Because logos are awesome. |
|  | Place PCB part number on silk screen. Good habit. |
|  | Attempt to orient all polarized components the same way This makes it easier for anyone doing placement. |
|  | Place ground planes where possible If you can’t place an entire plane, use a pour to cover as much space as possible. However, you do not want ground loops. So don’t add GND wires on a different level if you have a pour elsewhere |
|  | Check pours to ensure that they are connected Sometimes, when you add a pour as part of your process, and re-edit, and re-pour, you might forget to connect the pour to a signal. If a pour is not connected, delete it. |
|  | Fabrication Checks |
|  | Run a Design Rule Check (DRC) Some fabrication facilities ([Sunstone](https://www.sunstone.com/pcb-resources/cad-tools/altium)) publish design rules that you can download to your pcb design tool for a design rule check. You must either correct or understand and approve any violation of the DRC.  **Do this again when done.** |
|  | Run a Design for Manufacturing (DFM) test Some fabrication facilities ([Advanced Circuits](http://www.4pcb.com/free-pcb-file-check/index.html)) provide on-line services that perform design-rule checks on uploaded files. If you are having your boards fabricated at Advanced, run this test. If you are having your boards fabricated elsewhere, and that fab does *not* have a DFM check, run the one from Advanced. In any case, you must either correct or understand and approve any violation of the DFM test.  **Do for the first time.** |
|  | Validate your part footprints If you can access your parts before submitting your board for fabrication, print out your PCB to scale on paper and validate that the footprints match. |

|  |  |
| --- | --- |
|  | Bill of Materials |
|  | Verify that all parts are accounted for. Every part on your schematic should show up in your bill of materials. |
|  | Verify that all parts are the correct package. Parts often come in a variety of packages and the very last bits of the part name often specify the package type. Make sure that you purchase the part with the same footprint that you require. |

|  |  |
| --- | --- |
|  | PCB Delivery After receiving the PCB, but before soldering. |
|  | Ensure that all of your power nets are connected to each other. Do the same for your GND nets. |
|  | Ensure that there is not a short between your power and ground nets. |
|  | Where reasonable, check that your major nets are connected. |
|  | If you have a 4-layer board, verify that the interior planes have been wired correctly. Doing the above 3 checks should verify this. |
|  | Verify that your part footprints match the board. |
|  | Ensure that your connectors are oriented correctly. |

|  |  |
| --- | --- |
|  | Pre-Assembly Before sending PCB and parts off to the assembler. |
|  | Cross-reference every part on the BOARD with every part on the BOM. Do not simply trust that the BOM output is correct, validate that every part you need on the board is on the BOM or noted. |
|  | Prepare the kit for assembly. As you prepare the kit, you should check each part on the BOM, locate the part on the board, ensure that the footprint matches, and ensure that the silkscreen indicates the correct orientation of the part. If you have small and cheap parts (under $0.10), you should have some extras of those. Our assembler does an outstanding job of not requiring extras, so we don’t have to worry about this for larger or more expensive parts. |
|  | Create Project Outputs for the assembler. These should include the Gerbers, BOM, and any assembly drawings you have. Reference prior boards. |
|  | Create a README file for special instructions If you have special instructions for the assembler, including parts you do not want placed, add these to the README. |