Badgelife 101

Slides & Resources: https://1337.fyi/badgelife101

About Me

- Hacker & Maker
- Tech Lead, Google Red Team
- Mostly self-taught in electronics
  - Not an EE, so you don't need to be either!
- @matir@infosec.exchange on Mastodon

Any opinions included in this presentation are my own, and not my employer's.
Alphabet/Google does not endorse any of the products/services discussed.

Quote

“Hack the Planet!”

- Dade Murphy

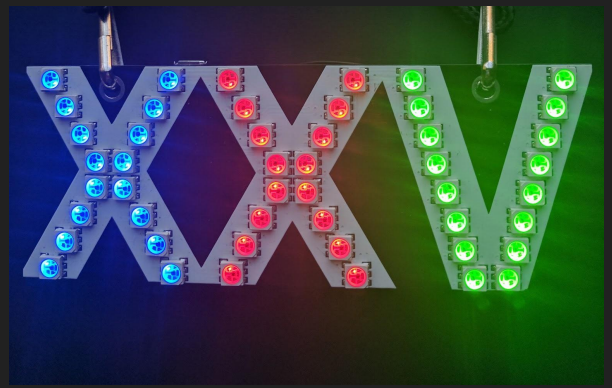
(The Unicorn speaking before me reminded me I need a quote.)

What is Badgelife?

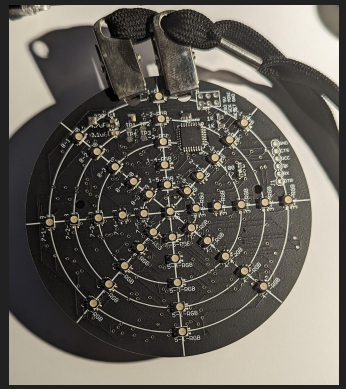
- Wearable Electronic Art
  - Non-electronic badges may be seen as well, but we'll only consider electronic badges
- Self-Expression
- Interaction
- Sometimes Conference/Event Admission
- Hackaday Documentary: https://youtu.be/G2fHKRONc6U

Images: http://www.grandideastudio.com/defcon-14-badge/, https://hackaday.com/2021/08/05/hands-on-def-con-29-badge-embraces-the-new-normal/, https://hackaday.com/2019/09/19/pictorial-guide-to-the-unofficial-electronic-badges-of-def-con-27/

My Badgelife



*



*

* Collaboration

Other Applications

DIY Electronic Ornaments Anyone?

Name Badges

Setting Expectations

- I’m assuming little knowledge of electronics, but stop me with questions anytime!
- I hope to get you an understanding of the workflow and basics, but there will be some gaps left. This could be a whole day training :)
- I’ll link resources to help fill those gaps.

This will **NOT** cover anything involving "mains" voltage! Use only batteries, USB, or a commercial DC power supply for your Badgelife projects.

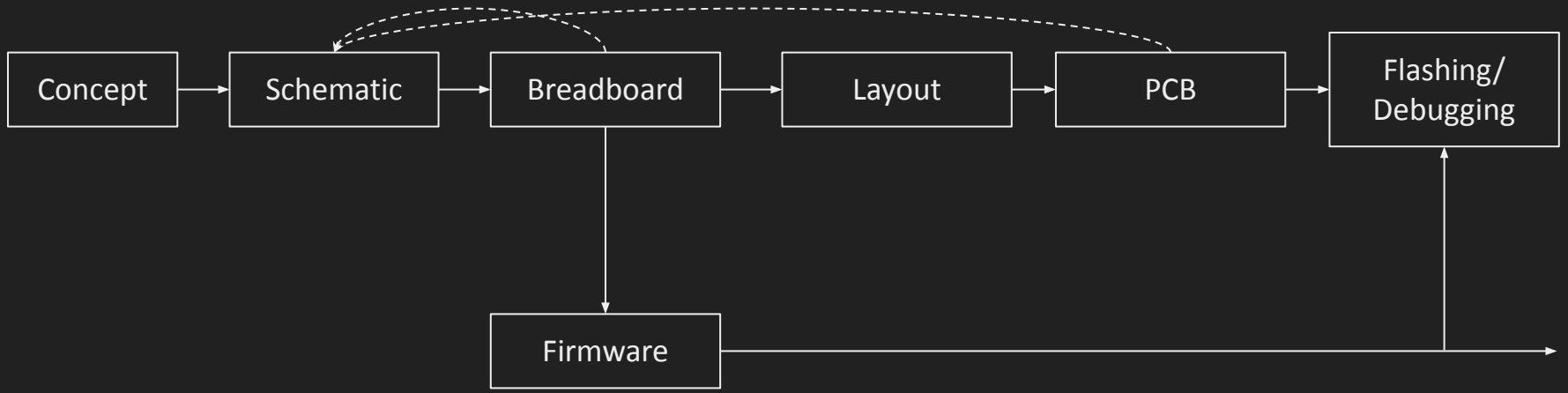
Quick Glossary

- Badge - PCB, usually with electronics, designed to be worn at an event
- SAO - S****y Add On – A semi-standardized connector for building small “add-ons” to badges
- PCB - Printed Circuit Board, the base material for electronic boards
- MCU - Microcontroller, a small embedded processor containing CPU/RAM/Flash in one package
- IC - Integrated Circuit, a component that integrates multiple “basic” components into one. Often black rectangles with many leads.
- UART - Universal Asynchronous Receiver/Transmitter. Basically an embedded serial port.

Steps Involved

1. Concept
2. Schematic
3. Prototype (Breadboard)
4. Firmware
5. Layout
6. Prototype (PCB)
7. Flashing/Debugging
8. Distribution

Steps Involved



The Concept

Goals

- Style/Artwork
  - “Just for Fun”
- Group Membership
  - Identity
  - CTF Teams
  - Village Supporters
- Interaction
  - Infrared
  - RF
  - Physical Connections
- Event Access

Constraints (Hard Limits)

- Budget
  - BOM cost – Cost of components on each unit
  - Total cost - BOM Cost + Production Costs + Tooling
- Time Budget
  - It **will** take longer than you think, I promise.
  - Manufacturing takes time
- Power Budget
  - Battery powered, so have limits
  - Newer tech is making a lot more possible in this space

Considerations

- It’s a wearable
  - Attachment: Lanyard Holes? Pin?
  - Sharp Edges are no fun
  - Through-hole component pins are sharp
  - Size/Weight – again, it’s a wearable :)
  - Heat – can be a concern if high power drain
- Overreaching
  - You may have a great idea, but can you implement it in time?
- Hardware Supply Chains
  - Some components can easily be substituted – resistors, capacitors, LEDs, etc.
  - Others not so much – microcontrollers, radios, specialty chips

Power Options

- Coin Cell
  - 3V, can run most microcontrollers
  - Very low current capacity
  - Cheap
- Alkaline Batteries
  - Need 3+ to reliably get ~3.3V from a voltage regulator
  - Cheap but heavy
- LiPo Pouch Cells
  - Flat
  - Physically Unprotected
  - Buy **only** with protection board!
- Specialty Batteries
  - A123 Lithium Battery, etc.

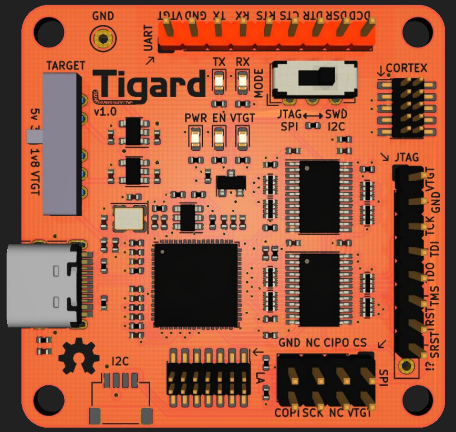
Tools

Software

- Electronic Design Automation (EDA)
  - KiCad - FOSS, popular in badgelife circles
  - Eagle - Now part of Autodesk Fusion
    - Very Limited Free Version
  - Altium - High End Professional Software
  - EasyEDA - Web Based, Limited
- Firmware Tools
  - gcc-based tool chains (open source, many architectures)
  - MPLAB for Microchip Devices
  - Arduino for supported MCUs
- Debugging Tools
  - OpenOCD (JTAG/SWD)
  - gdb (GNU Debugger)
- Flashing Tools
  - OpenOCD
  - flashrom

Hardware Tools

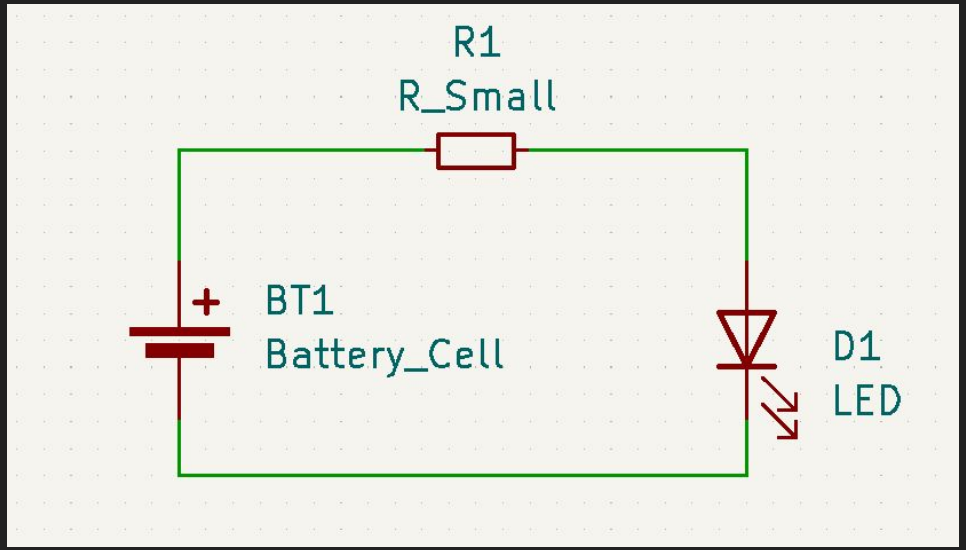
- Multimeter
- Soldering Iron
- Programmer/Debugger
  - Depends on MCU Choice
- UART/Serial Adapter
  - Unless MCU supports USB
- Logic Analyzer (nice, but optional)
- Hot Air Station (nice, but optional)



Schematic Design

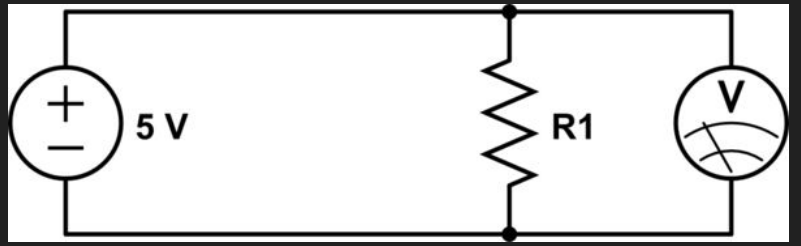
Schematic Concepts

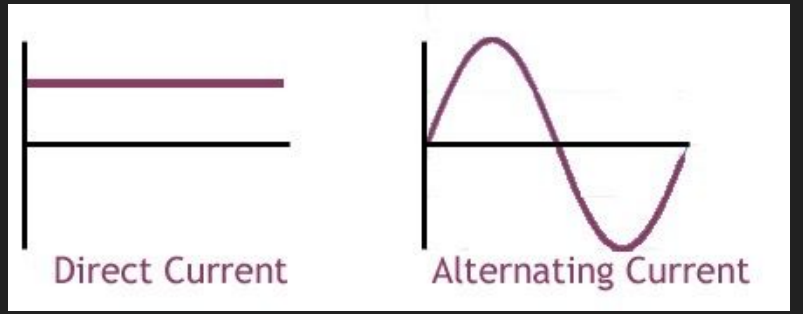
- Representation of **logical** electrical connections between components
- Drawn in EDA tool
- Labeled connections are “virtual” connections
- Standardized symbols



Electronics Basics: Voltage

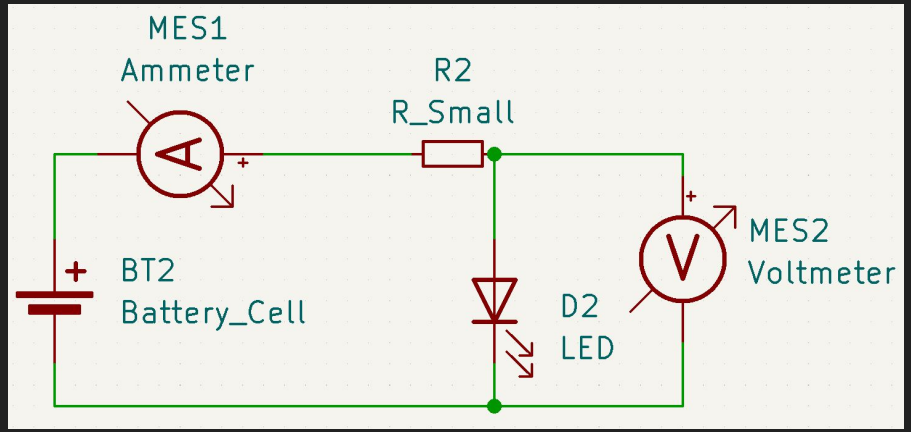
- Voltage, also called potential, is the difference between charges in two parts of a circuit
  - DC circuits have a continuous supply voltage
  - AC, as in wall outlets, has voltage regularly cycling
- By convention, an arbitrary point is designated as “Ground” or “0V”
- In most DC circuits, this will be the lowest voltage point in the circuit (so all other voltages are positive)





Electronics Basics: Current

- Current is the rate of flow of electric charge through a conductor
- Measured in “amps”
- Current is “pulled” – you can safely use a battery or power supply with a higher rating than the current you need to draw

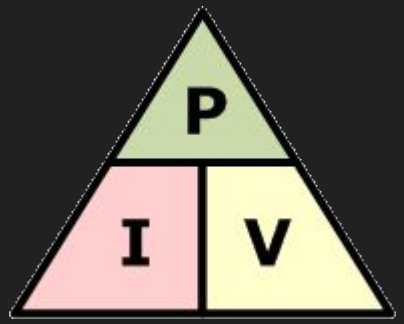


Basic Components

- Battery, Power Source
  - Provides power to run
- Resistor
  - Resists the flow of current in a linear fashion
- Capacitor
  - “Buffers” electrical charge
  - In Badgelife, mostly for stabilizing power supplies
- Diode
  - Current flow in one direction
  - LEDs also emit light!

Relationships

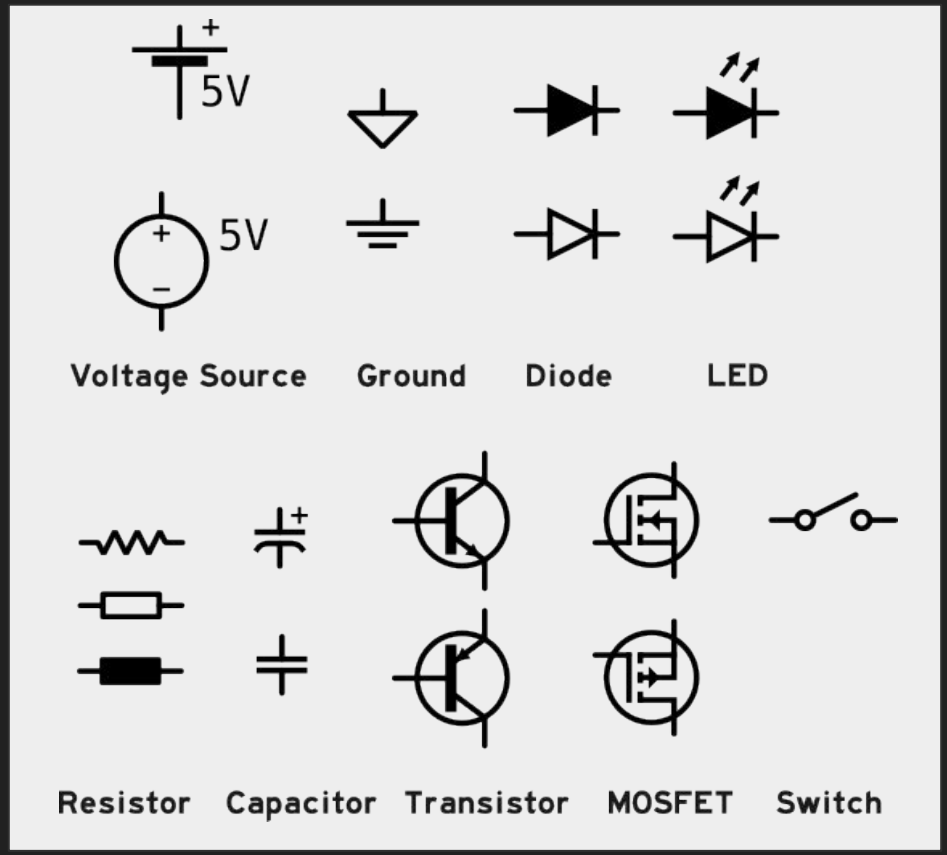
Power (Watts) = Voltage (Volts) * Current (Amps)



Voltage (Volts) = Current (Amps) * Resistance (Ohms)

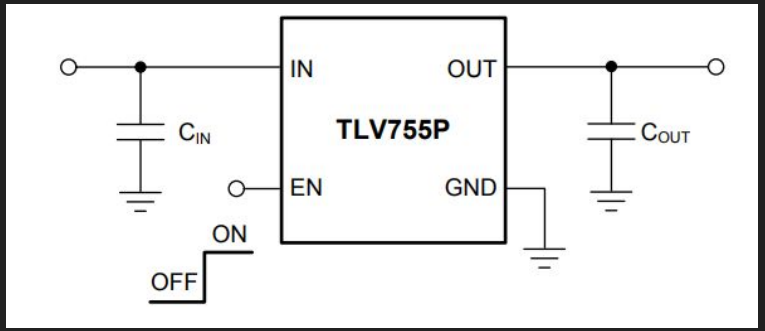
(Ohm’s Law)

Schematic Symbols

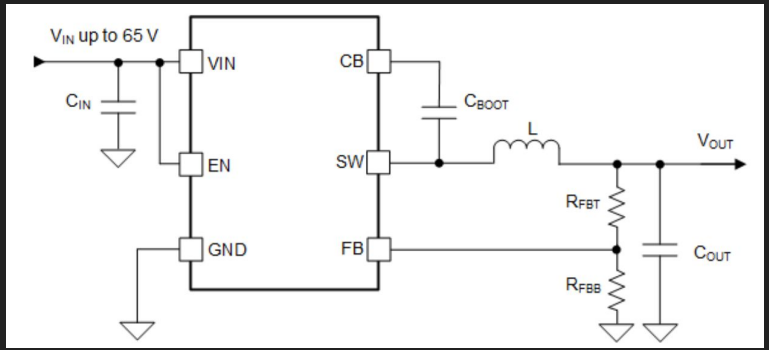


Voltage Regulation

- You cannot just use a resistor to “drop” voltage – the voltage drop depends on the current
- Voltage regulators actively control output voltage
- Linear regulars are simple, but waste excess power as heat
- Switching regulators are complex, but much more efficient

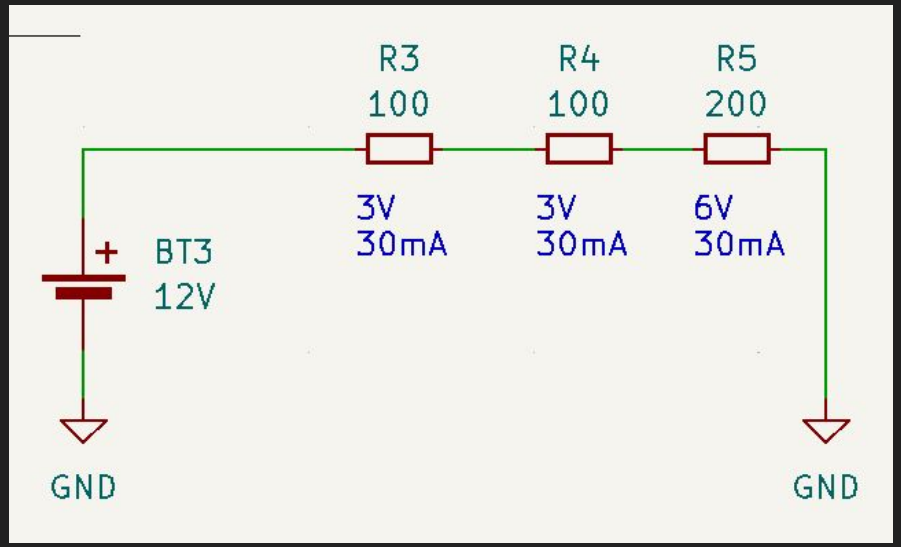


Linear

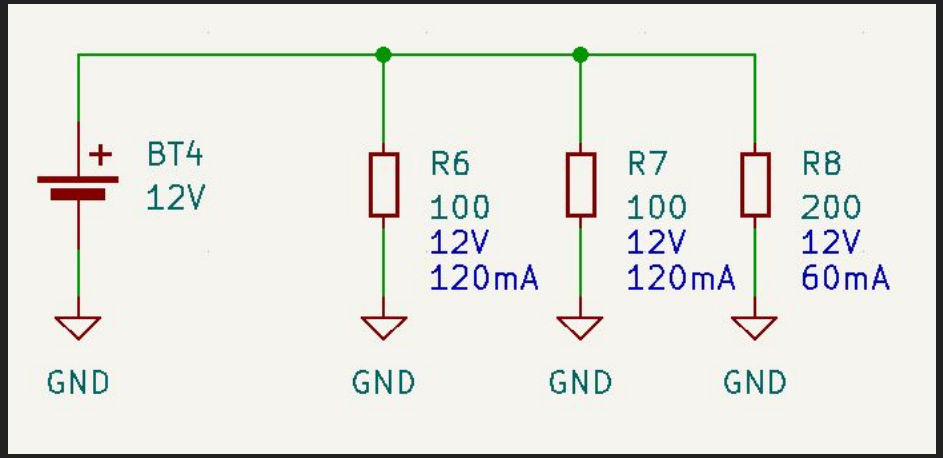


Switching

Series/Parallel Circuits



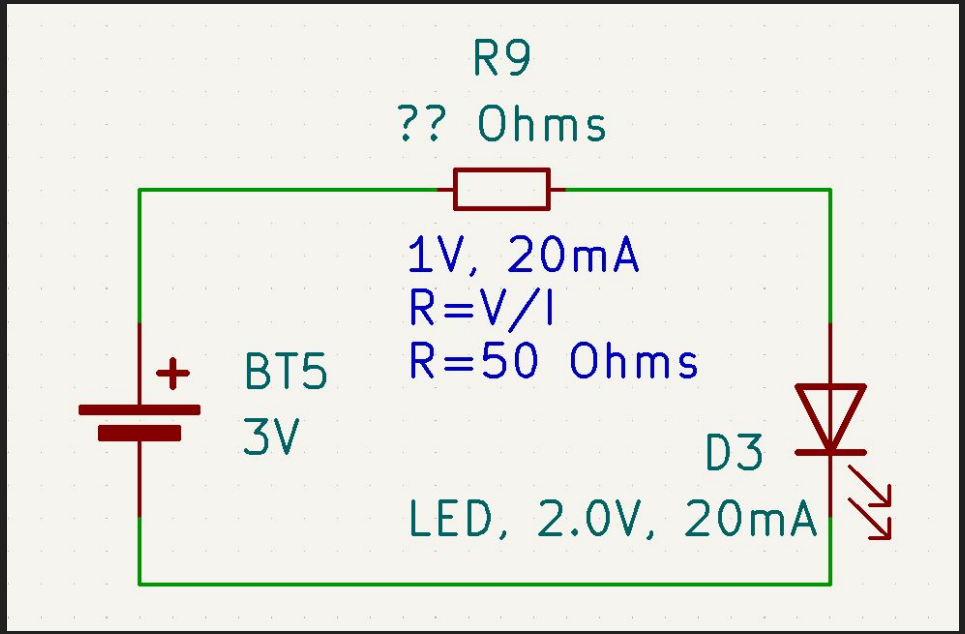
Series
Voltage Divided, Current Same



Parallel
Voltage Same, Current Separated

LEDs!

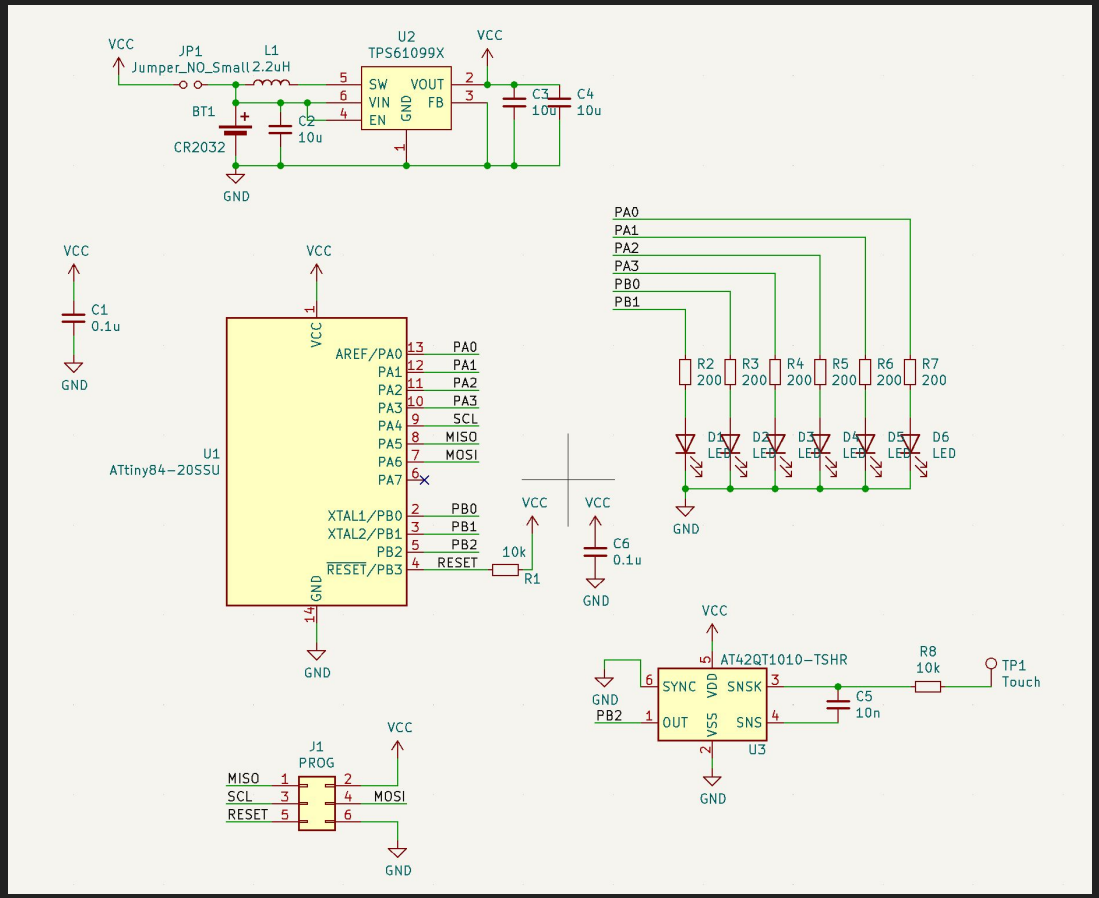
- Have a roughly fixed voltage drop across them
- Need to have current limited (or they glow **very** brightly **very** briefly)
- ~20mA is a typical max, but check the datasheet, 5mA is usually plenty with modern LEDs
- Vsupply = Vled + I*R



Schematic Rules of Thumb

- Group things by function
  - Power Supply
  - Microcontroller & supporting (reset, programming header, etc.)
  - LEDs/visual
  - Peripherals
- Positive voltage at top, ground at bottom
- Current in from left, out to right
- Use named nets instead of dragging wires everywhere

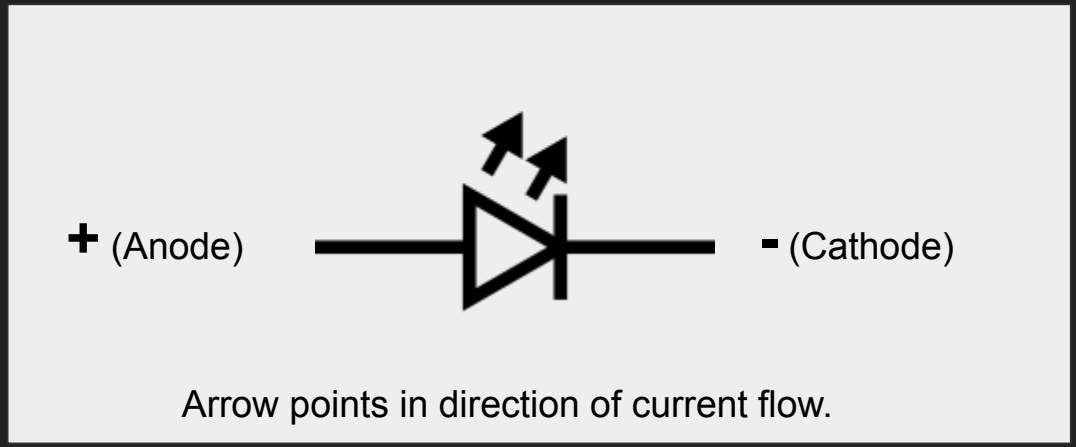
Example Schematic



Live Demo!

Your Turn 😺

- Using KiCad, create a schematic with the following:
  - 1 Battery
  - 6 parallel sets of 1 LED + 1 Resistor in Series
  - Connect positive side of series circuits to positive of battery and negative to negative



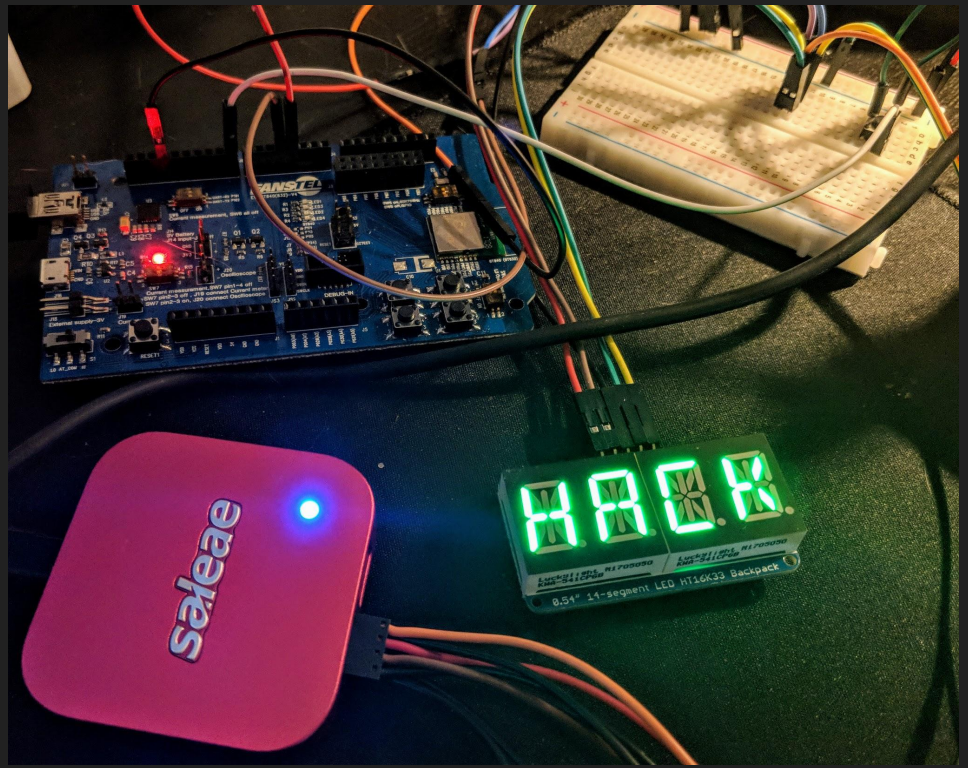
Early Prototype

Breadboard Prototype

Can be whole circuit, or just some parts for testing.

Useful:

- Dev boards from MCU vendor
- Breakout boards for components
- Variable power supply



Firmware

Firmware Basics

- Firmware is the code executed by your microcontroller
- Without it, nothing happens
- Needs to be flashed to MCU (or external storage)
- Good time to start with breadboard
- May need to remember bitwise operations

```
  // Enable timer for next frame
  OCR0A = TIMER_INTERVAL;    // Interval
  TIMSK0 = 0x02;             // Interrupt on compare
  TCCR0A = 0x02;             // CTC Mode
#if TIMER_PRESCALER == 1
  TCCR0B = 1<<CS00;          // No prescaler
#elif TIMER_PRESCALER == 8
  TCCR0B = 1<<CS01;          // Prescale by 8
#else
# error "Unknown prescaler."
#endif

  // Reduce power consumption slightly
  PRR |= (1<<PRTIM1) | (1<<PRUSI) | (1<<PRADC);

  // Enable the touch sensor
#ifdef TOUCH_ENABLED
  // INT0 on rising edge
  MCUCR |= (1 << ISC01) | (1 << ISC00);
  GIMSK |= (1 << INT0);
#endif
```

Firmware Challenges

- No OS Facilities
- Flash to iterate (unless emulated)
- Not threaded, no process scheduling
  - Real-time OS (RTOS) can get you cooperative multi-tasking
- Limited human I/O (maybe a serial port)
- Documentation may be… variable

Controlling Pins

- Exact mechanism depends on your MCU
- Generally, 2 things need to be done
  - Set pin to output or input mode
  - Set pin state high or low to control
  - High = input voltage
  - Low = 0V

```
24  // the setup function runs once when you press reset or power the board
25  void setup() {
26    // initialize digital pin LED_BUILTIN as an output.
27    pinMode(LED_BUILTIN, OUTPUT);
28  }
29
30  // the loop function runs over and over again forever
31  void loop() {
32    digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
33    delay(1000);                       // wait for a second
34    digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
35    delay(1000);                       // wait for a second
36  }
```

Hardware Peripherals

- **VERY** MCU Dependent
- Timers
  - Count without using main CPU
  - Trigger Interrupt
- Hardware PWM
  - Great for controlling brightness of LEDs
- SPI/I2C
  - Useful for interfacing with other chips

PCB Layout

PCB Basics

- Printed Circuit Boards are the layered boards that hold all the components
- Main part of a badge
- Layers
  - Silkscreen (markings)
  - Soldermask (main color, protects copper)
  - Copper (electrical connections)
  - Fiberglass (typically)
- 2 Copper Layers most common
  - More adds cost and complexity
  - “Vias” are connections between layers

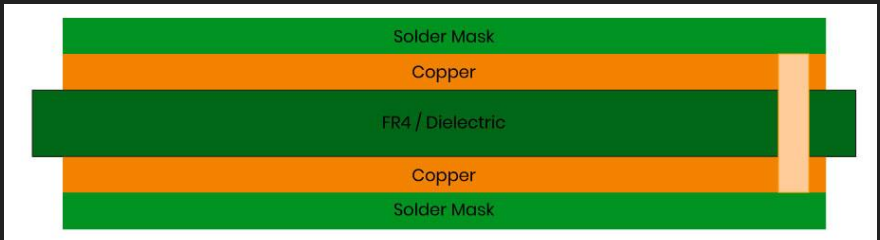
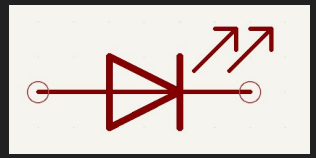
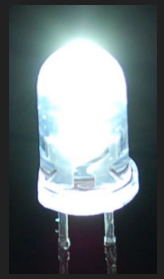


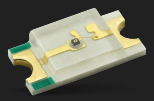
Image: SF Circuits

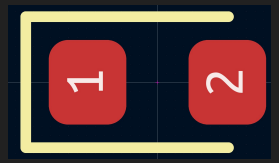
Footprint Matching

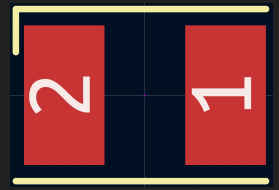
- Schematic has logical representation, but nothing about physical form factor of components
- Assigning “footprints” is the process to match the logical to the physical
- Most common components have footprints in libraries, but you might need to create your own sometimes (though distributors like Digikey may offer footprints for many components)









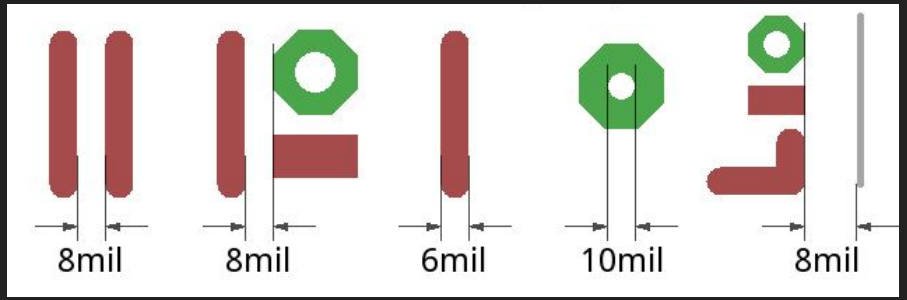


EDA Layers

- Copper
  - Carries Electrical Current
  - Usually a "positive" layer in software (where drawn, there will be copper)
- Soldermask
  - Prevents shorts, solder from sticking
  - A "negative" layer in software (where drawn, there will be holes in soldermask)
- Silkscreen
  - Printed on top of soldermask
  - A "positive" layer (where drawn, there will be ink)
- Paste
  - Solder paste for assembly
  - Positive layer (where drawn, there will be paste)
- Edge Cuts/Mechanical
  - Defines outline
  - Drilled/milled/etc.

Design Rules

- Each “fab” (fabricator) has rules for what they can produce
- Defines minimums for attributes
  - Trace width
  - Trace Separation
  - Drill Holes (vias)
  - Soldermask width
- 8 “mil” is usually safe for traces and clearances
  - 8/1000 of an inch
  - 0.2mm
- Design Rule Check (DRC) can help spot errors!



Badgelife PCB Design

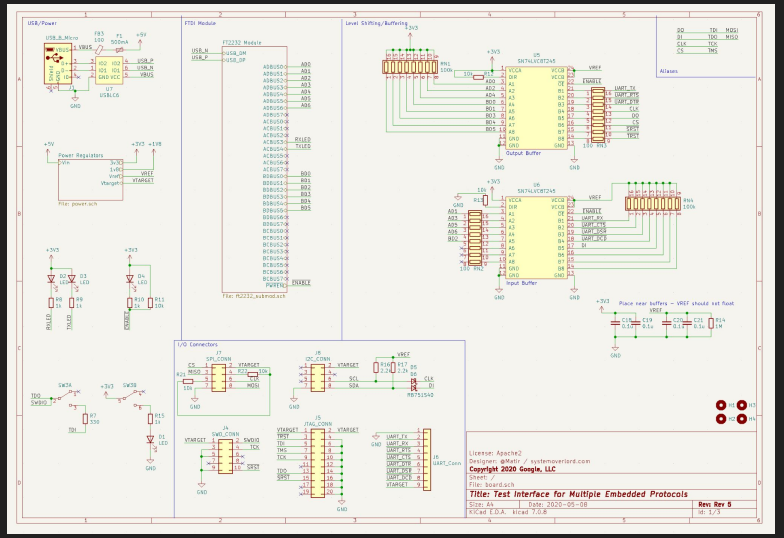
- Art & Outline First
  - Most non-badgelife PCBs will worry about this *last*, but since Badgelife is about design, start here
  - KiCad can directly import SVG for outline layers
- Tools for Art on PCB
  - KiCad SVG Import
  - KiCad Image Converter (for raster images)
  - Gerbolyze will do halftone images
  - svg2shenzhen

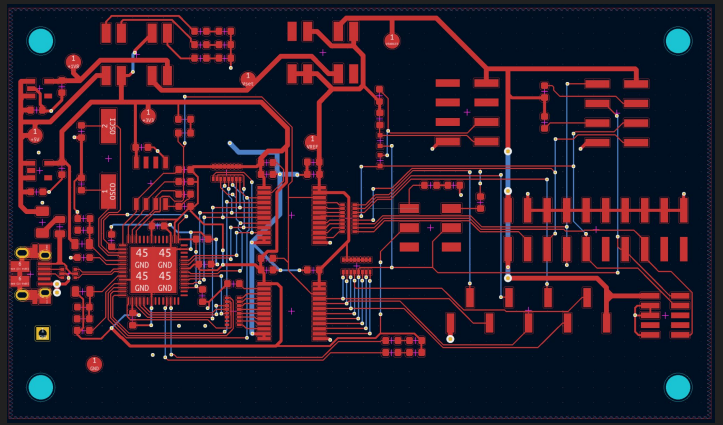
Source: https://www.tindie.com/products/makerqueenau/space-girl-soldering-kit/

Schematic -> PCB

- Import Schematic into PCB Tool

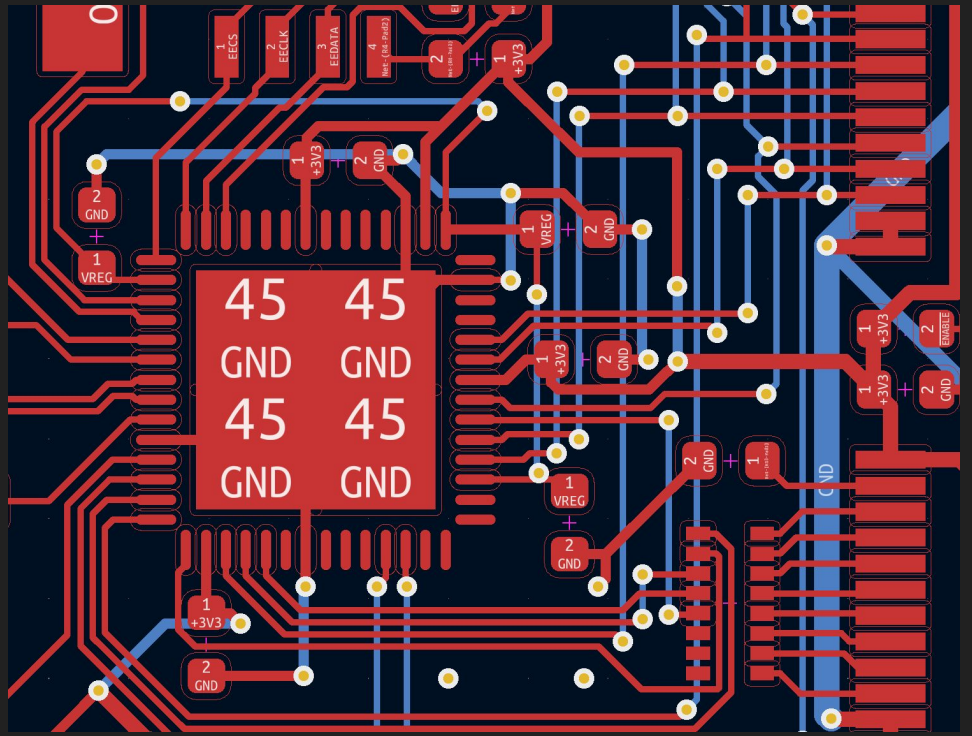
- Layout components first
  - Visually critical (LEDs, displays, etc. that are part of the art)
  - Major components (microcontroller, other ICs)
  - Add minor components (resistors, capacitors)
- May need to shift during layout





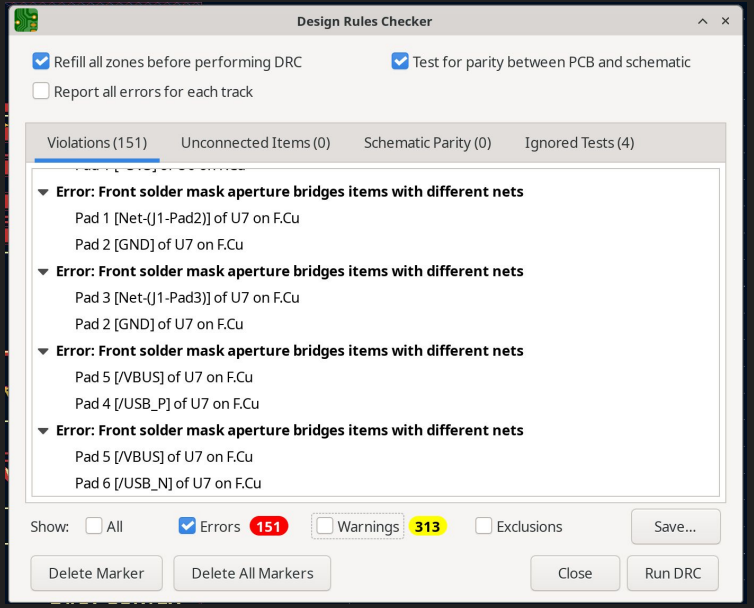
Routing Traces

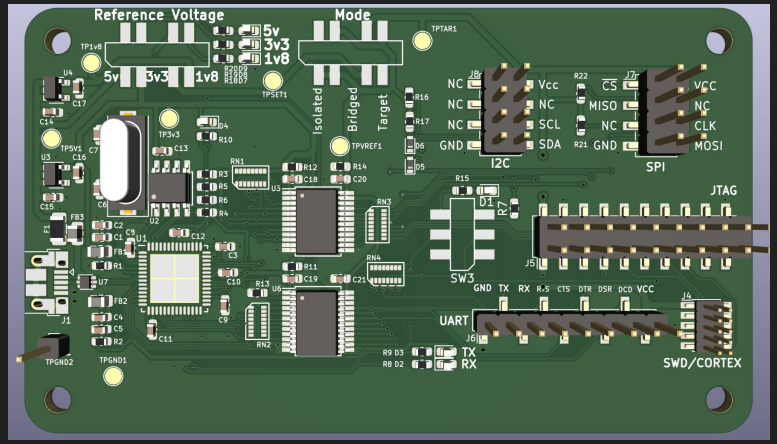
- Traces connect all the components
- You will see thin lines prompting you to connect certain components – this is the “rat’s nest” you are cleaning up
- Cannot cross other unrelated traces without switching to another layer (short circuit)
- A bit of an art form
- You **can** use the “autorouter” but the results may not be appealing



Pre-Fab Checklist

- DRC Run and Clear?
- Outline Layer complete? (Edge.Cuts)
- Mounting Hardware?
- Corners Rounded?
- Connections for Debug/Programming?
- Renders look right?



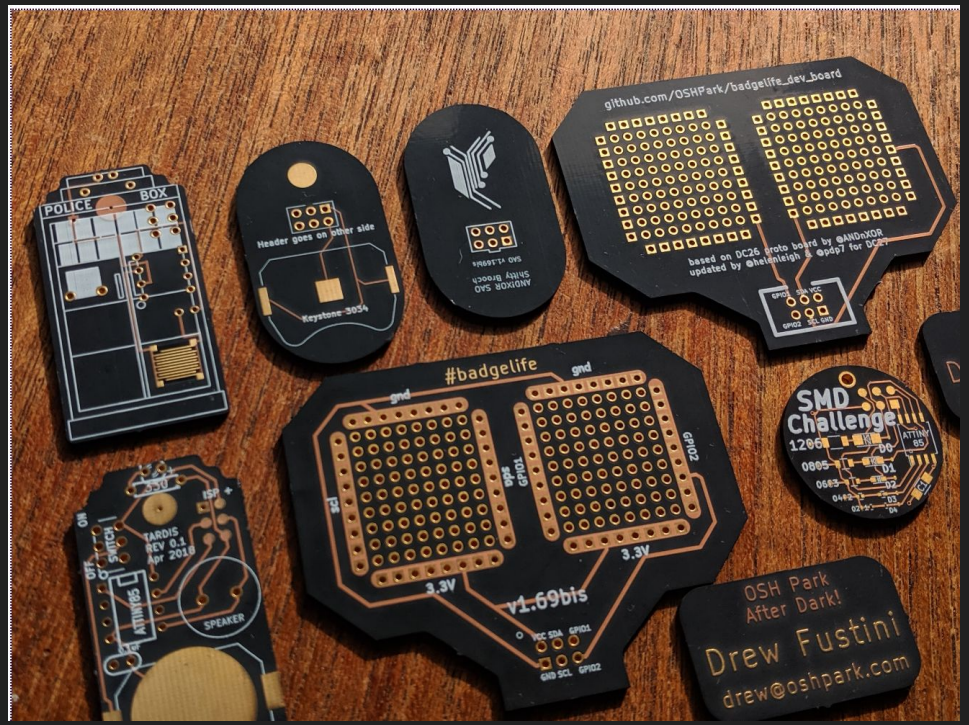


Sending to Fabricator

- Export as Gerbers and Drill Files
  - Industry standard formats for PCBs
- Use a Gerber Viewer to verify everything looks okay
- Upload to Fabricator
- Pick Options, Send Order
- (Hopefully Not) Realize your mistakes as soon as fabrication is in process.

Which Fab?

- OSH Park
  - Quick Prototype Turns
  - Very High Quality
  - Pricey in Quantity
  - Purple or Clear mask
  - Excellent Support
- JLCPCB (Shenzhen)
  - Very popular
  - Many colors
  - Much cheaper
- pcbshopper.com

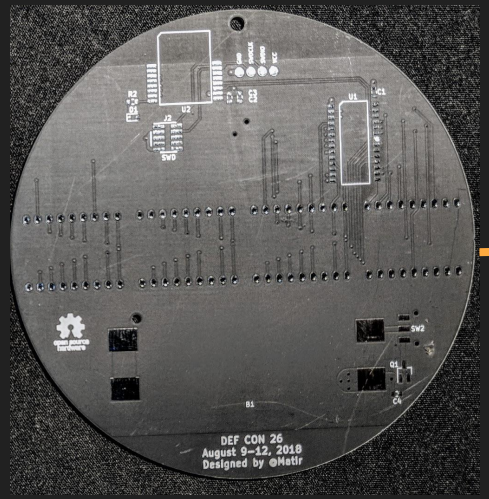


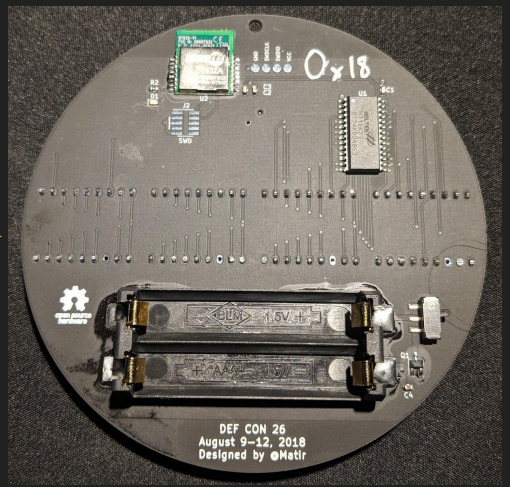
Assembly

PCB to Badge?


ATTACKER COMMUNITY
HACK THE PLANET


ATTACKER COMMUNITY
HACK THE PLANET


DEF CON 26
August 9-12, 2018
Designed by @Matir


0x18
DEF CON 26
August 9-12, 2018
Designed by @Matir

DIY

- Requires soldering skill and tools
- Great for prototypes
- Not great at scale
  - It’s doable for less complex designs if you plan enough time
- If you’re using surface mount components, much easier with hot air

PCB Assembly

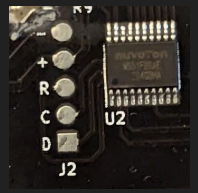
- May limit choice of components
- Setup cost may be high for prototypes
- Great for production runs
- Through-hole components will dramatically raise costs
- Requires panels that can run through automation

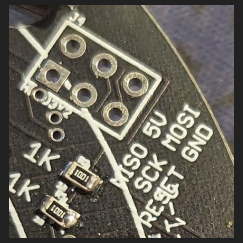
Flashing/Debugging

Flash and Debug Interfaces

- JTAG is a standard for debug of a variety of cores, requires 6+ pins
- SWD is a reduced form of JTAG, requires as few as 3 pins, and is ARM-specific
- SPI is commonly used for interfacing between chips, but also for programming AVR microcontrollers, with 5-6 pins
- Tigard is a great tool for both Flash and Debug

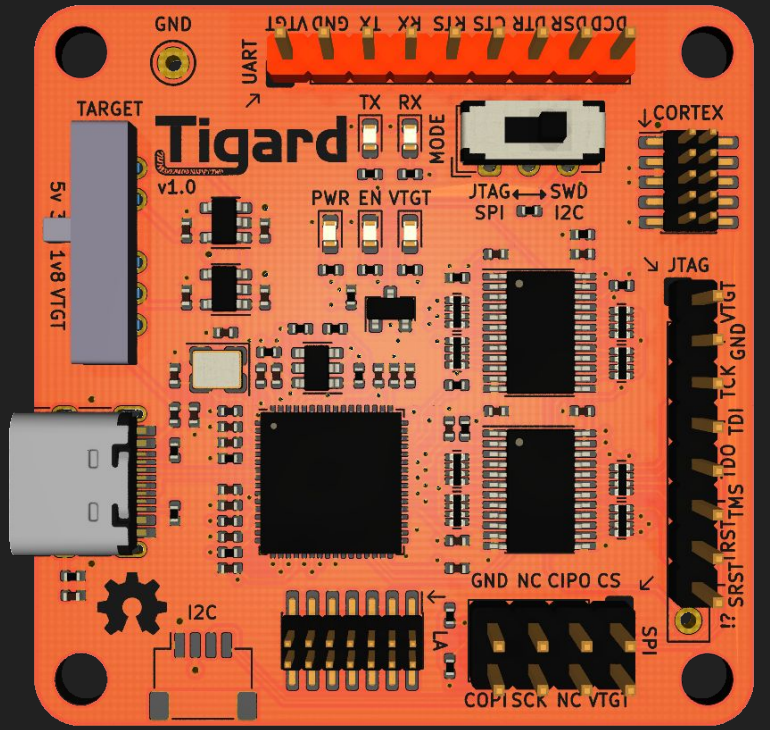
- Hopefully you have a program/debug header :)





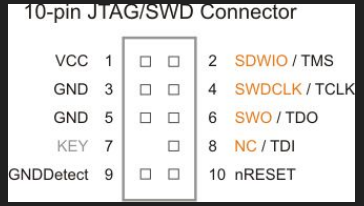
Flashing

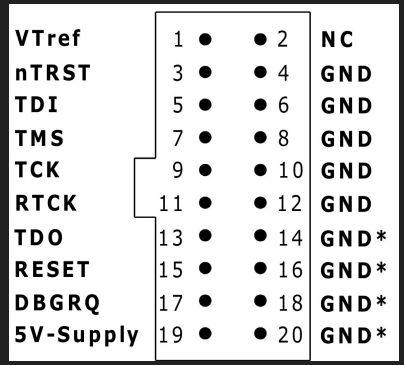
- Very microcontroller dependent
  - Most ARM can be flashed over debug port (SWD)
  - ESP32 family over Serial Port (UART)
  - AVR via SPI-like interface
- Different hardware to interface with each, although FT232H-based boards can talk to all 3 with various software



Debugging

- Often same interface as flashing
  - ESP32 needs JTAG for debug, UART for Flash
- Sometimes can use gdb, sometimes need specialized tools
- Almost all MCUs have serial (UART) ports, so printf debugging might be possible
  - Unless you crash before then
  - Or corrupt memory used by the UART





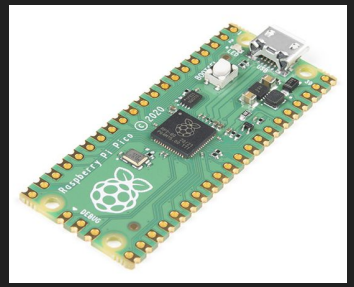
Pitfalls/Lessons Learned

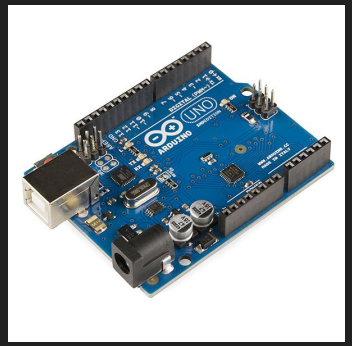
Basic Tips

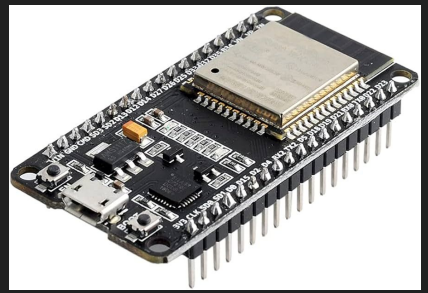
- Assume everything will take much longer than you expect
- Keep your design in a git repository so you can roll back if needed
    - KiCad files are plain text, but not really editable traditionally, so don’t try merges/diffs/etc.
- Make your first project(s) simple
- Assume a couple of prototype cycles
- Print PCB layout on paper and place major components if you have them (check footprints/spacing/size)

Recommended MCUs

- ATMega Series
  - Used in Arduino, Simple, Widely Documented
- Atmel SAMD21/SAMD51
  - ARM-based, well documented, some Arduino compatible
- Raspberry Pi RP2040
  - ARM-based, well documented, good community resources, **very** capable
  - Needs external flash though
- ESP32
  - Use **only** if you need BT/WiFi
  - Battery life **will** suffer







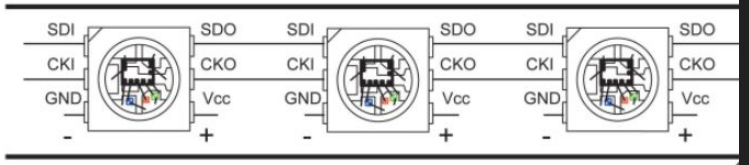
Datasheets Are Critical

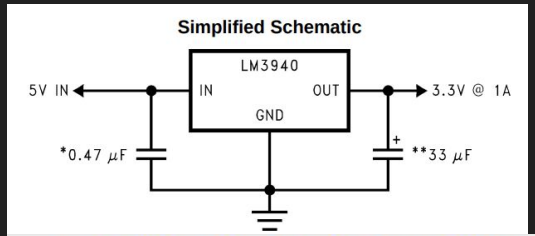
- They can be hard to read, but are full of useful information
  - Voltage Limits
  - Current Consumption
  - Pinouts
  - Recommended/Reference Designs*

● 最大額定範圍（Absolute Maximum Ratings）

Supply Voltage————————–0.3V to 6.0V
Input Voltage————VSS–0.3 to VDD+0.3
Operating Temperature———— –40℃ to 70 ℃
Storage Temperature————–50℃ to 125 ℃
Note： Stress above those listed may cause permanent damage to the devices

● 應用線路圖（Application Circuit）





6.1 Absolute Maximum Ratings

over operating free-air temperature range (unless otherwise noted)(1)

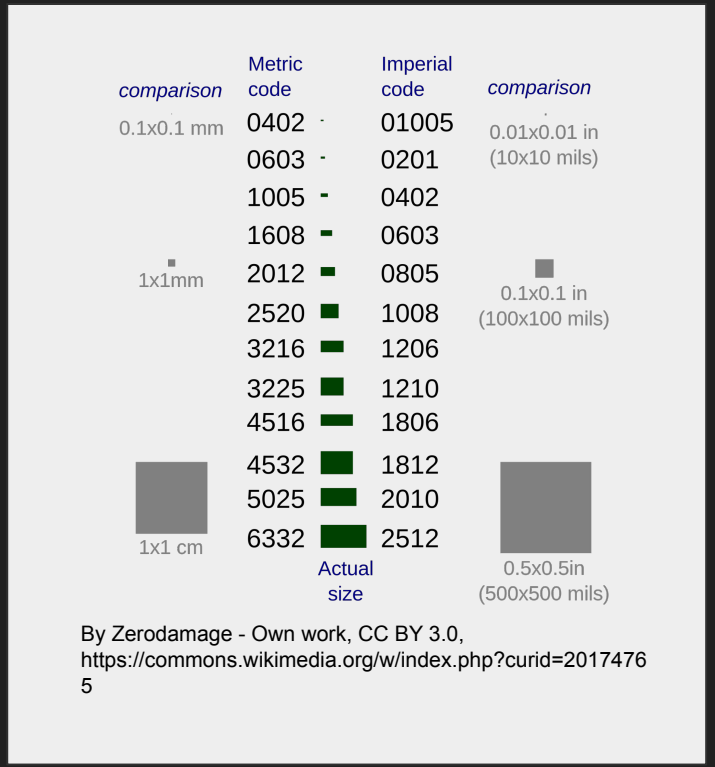
	MIN	MAX	UNIT
Power dissipation(2)		Internally Limited	
Input supply voltage	–0.3	7.5	V
Storage temperature, T_{stg}	−65	150	°C

My #fails

- LED brightness/current curve is highly variable from exact part to exact part
- Double check footprints
  - And not just the pins/pads, also the physical outline
- Imagining I will hand-assemble 50 badges
- Ordering PCBs without key components already ordered
- Wrong footprint for same component
  - Many components are available in several packages
- Ordered wrong part (1.8V regulator instead of 3.3V)

Near-fails (caught before production)

- Forgot a programming pin
- Battery holder loads from side, but points at another component blocking the holder
- Not measuring how far from the edge lanyard holes should be
- Wrong footprint for same component
  - Yeah, that's been done more than once… or twice



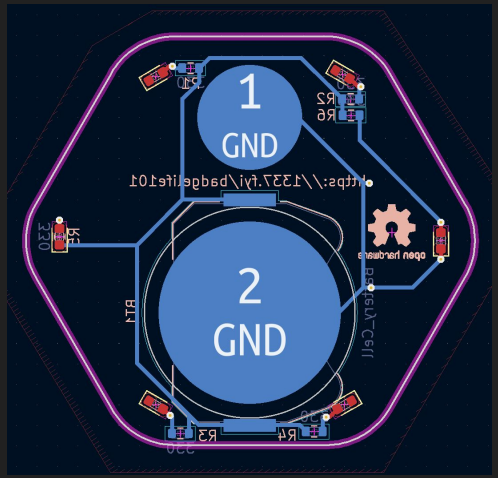
By Zerodamage - Own work, CC BY 3.0, https://commons.wikimedia.org/w/index.php?curid=20174765

Hands On!

Suggested Footprints for Demo

- LEDs: LED_0603_1608Metric
- Resistors: R_0603_1608Metric
- Battery: BatteryHolder_Keystone_3034_1x20mm

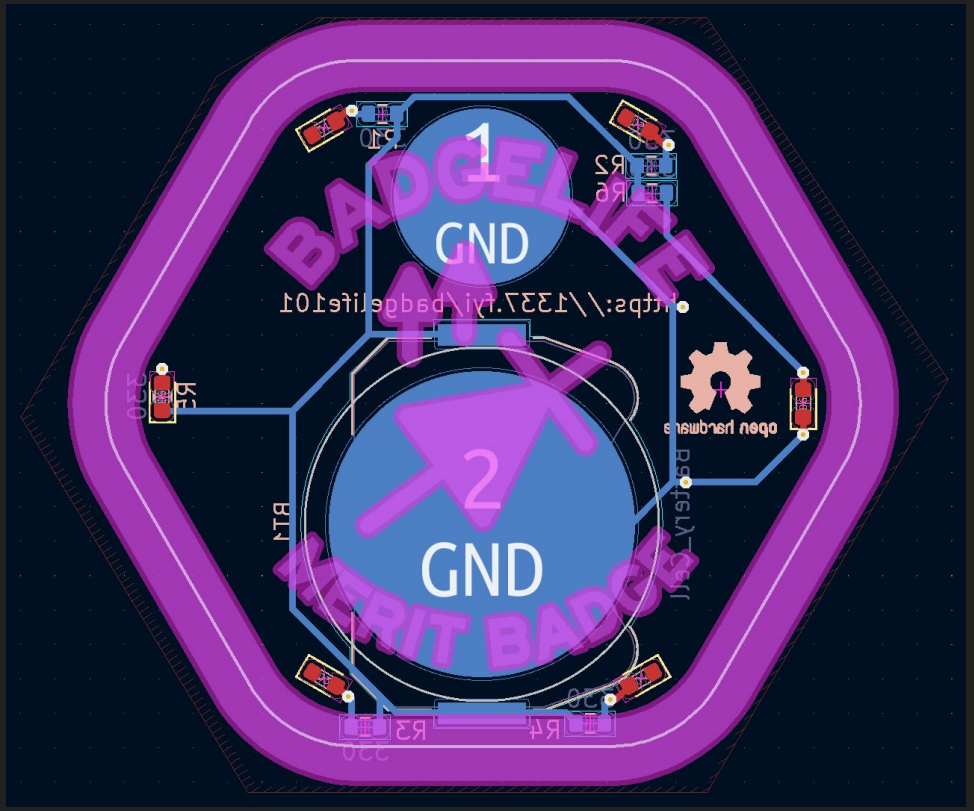
Finish PCB Design



- Assign Footprints
- Start PCB Layout
- Import Components
- Import SVG for Outline
- Place Components
- Route Components
- (Bonus) Export Gerbers and use Gerber Viewer to validate
- (Bonus Bonus) Finalize your design, add art of your choice, and fab with OSHPark (or similar)!

You can use the example schematic from GitHub if you didn't get it finished.

Final Design



Thanks!/Final Questions?

Supplemental Resources

Resources - Electronics Design/Analysis

- Hyperphysics Electronics Pages
- All About Circuits
- LED Resistor Calculator (DigiKey)
- How to Use a Multimeter (SparkFun)

Resources - Schematics

- Electronic Symbols (Wikipedia)
- How to Read a Schematic (SparkFun)
- Adafruit & Sparkfun publish most of their designs as open source, great references

Resources - EDA

- KiCad
  - Learning Resources
  - Third-Party Libraries
  - KiCad Cheatsheet
- Eagle (part of Fusion360)
  - Guide to make PCBs with Eagle (Adafruit)
- Comparison of EDA Software
- Online Gerber Viewer (GerbLook)

Resources - Hardware

- Parts
  - Digikey (Reliable, direct partner with many manufacturers)
  - Mouser
  - LCSC (Lots of China-based parts)
  - Beware eBay/AliExpress – Many Counterfeit/Recycled/etc. parts
- Datasheets
  - How to Read a Datasheet (SparkFun)
  - List of Integrated Circuit Packaging Types (Wikipedia)

Resources - Prototyping

- Breakout/Dev Boards
  - Adafruit (Great learning resources, open source designs)
  - SparkFun (Great learning resources, open source designs)
  - SeeedStudio (Wide variety of products, much ships from China)

Resources - PCB Art

- Gerbolyze
  - Produces halftone art in gerber format
- svg2shenzhen
  - Convert SVG to complex KiCad Shapes

Resources - PCB Fabrication

- Pre-Fab Checklists/Guidelines
  - PCB Basics (SparkFun)
  - OSH Park - Preorder Checklist
- Fabricators
  - OSH Park (Very high quality, fast turn around, local to Portland; Design Rules)
  - JLCPCB (Shenzhen-based, lots of options; Design Rules)
  - PCBWay (Shenzhen-based; Design Rules)
- Process
  - Strange Parts tours JLCPCB - PCB Fabrication (YouTube Video)
  - Strange Parts tours JLCPCB - PCB Assembly w/ Components (YouTube Video)