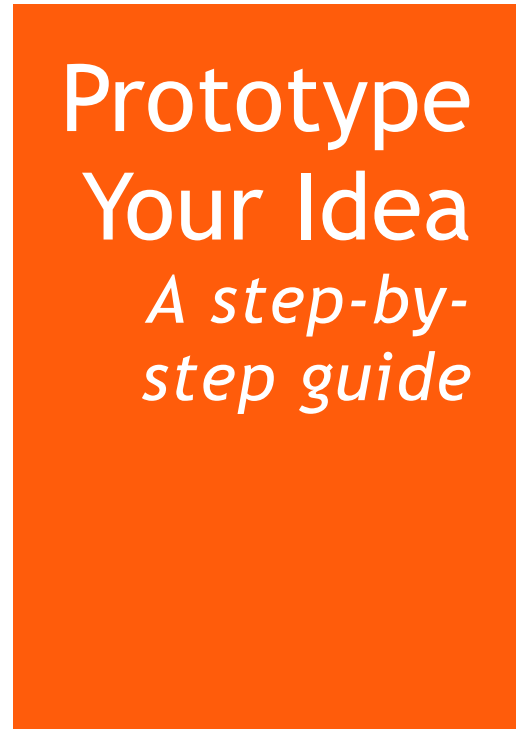




THE FINISHED PRODUCT



IN THIS GUIDE

by Greg Treseder - Founder, [Fab Forge Five](#)

## Introduction

You have a new product idea, and you're inspired to create it. It could be a fun project to share with friends and family. An item to sell at a Maker Faire or via a webstore. Perhaps you need a prototype to seek funding from potential investors. For all the above, this guide is for you. I'll cover a prototype design and build process from beginning to end, using lessons learned from a complex project of my own.

## Write a Product Summary

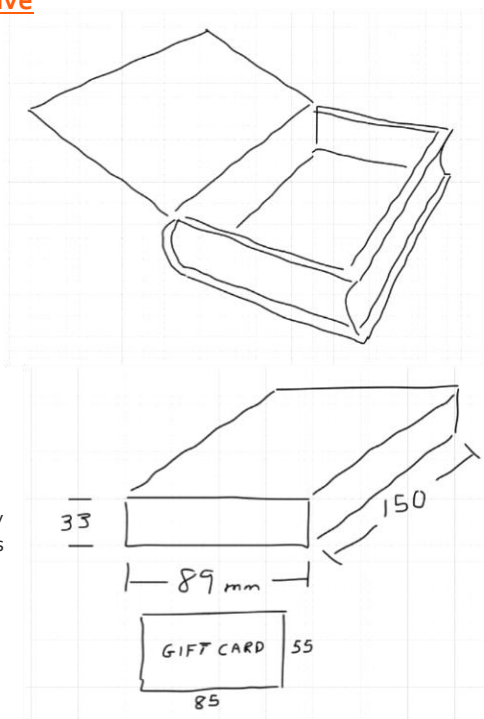
From this point on, spend a bit of time keeping all your work organized. Whether on a computer or in a notebook, keep track of versions and dates. This is helpful for reasons I'll explain later.

Start with a product summary. A crisp product summary will help you crystallize the main points of your idea. This will provide a foundation for your creative process, and will also help you explain your idea to others. A few sentences are plenty. No more than a paragraph. If it takes more than that, simplify your summary. It's also the first step in the later *Easy Marketing Materials* process. Here's an example of the product summary for my GPS Treasure Book idea:

*The VIA LIBRIS is a paperback-sized treasure book. It leads its user to a predetermined location, using a special compass dial and by giving COLD, WARM, or HOT clues. At the treasure location, a gift card pops out of a hidden slot in the book.*

## Create a sketch

Now get your idea on paper. You don't need a complex artwork – just a simple sketch. No drawing skills? Draw a simple diagram or two of a top (or side) view of your product. I can almost guarantee that this design will



change as you proceed. Perhaps many times. But a sketch will jump start your creative process, and your idea will begin to take shape. If your product will be constrained by size, draw a sketch of that volume, with dimensions. It will become the starting point for later work.

As you refine your design, come back to the sketch at some point – it's a necessity as you share your idea with others. And I promise it will help you brainstorm your idea, as your mind begins to visualize your creation in new and different ways.

Write a Product Summary

Create a Sketch

Your Product Parameters

Decide: Hardware vs Software

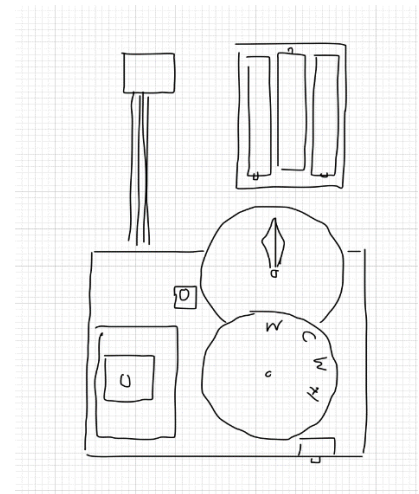
The Design Process

Build It!

Product Testing and Revisions

Documentation

Easy Marketing Materials



Even a simple sketch is powerful

## Your Product Parameters

These parameters form the basis of your design. If you spend a bit of time on this step early in the process, it will save you a great amount of time (and many revisions) later.

Think about how the product will be used, what the user interface should be, what environment(s) it will be used in (should it be waterproof?), what the power source will be, how long the power source should last, etc. Imagine a person using your product. How do they use it? Do they need to recharge it every night? Replace batteries occasionally? Will it need to be portable and rugged? Or will it sit on a table in a nice comfortable environment? Does it need lights? Sounds? You get the idea.

The parameters will likely change as your design evolves, but this first list will be a key step. As I thought through my list, I made some changes to my idea early on. For example, I decided I only wanted two motors (or servos) in my product. This required me to develop a design that did three types of movement with only two servos. A third servo or motor would have created a space issue, and would also add another set of magnets to a design that relied upon a sensitive electronic compass to determine direction. This was one of many tradeoffs in my design, and by identifying them early I was able to think them through and envision the ramifications of the different options. As you'll see later, this 2-servo decision created MANY challenges for me. I hope you'll learn from my mistakes!

- Must know its current location (latitude/longitude), must know the direction it's pointing
  - GPS module for precise latitude/longitude measurement
  - Electronic compass for directional measurement
- Must calculate bearing and distance to a predetermined latitude and longitude
  - Microcontroller (Arduino Pro Mini), 3.3-volt version (5-volt version would need four AAA batteries instead of three)
- Battery powered, at least 4.5 volts, but less than 9 volts. Need batteries with enough juice to power product on a treasure hunt for one hour. AAA alkaline batteries for size
  - Not rechargeable batteries – would require a charger and extra usage instructions. Need easily accessible, user replaceable alkaline batteries
  - Servos must run okay on 4.5 volts
- Location must be user-programmable without using a separate device or app, and must stay in memory when power is off
  - EEPROM in Arduino? Some type of switch or button that user can press to program a new location?
- Book must hide a gift card that user won't see until dispensed as a surprise
- Must guide user to the "treasure" location using a "compass" pointer, and a distance clue wheel (cold, warm, hot)
  - Two servos or motors – one for compass dial, one for distance clue wheel
- Must dispense gift card in a surprising way when the user arrives at destination, by somehow using the compass or clue wheel servo only. No third motor or actuator to save space, power, and compass interference

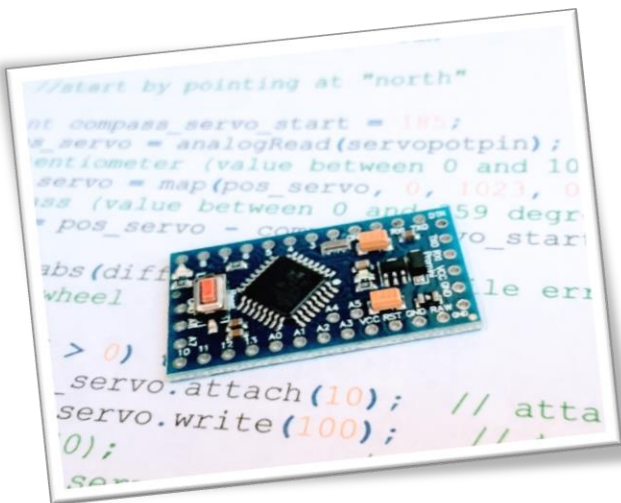
*Example: GPS Treasure Box Product Parameters*

### PARAMETER BULLET POINTS

Think of your parameters in levels of importance, like an indented bullet list as presented above. You'll have a few "no compromise" items, and many "gray area"

parameters that have some flexibility. Some parameters drive others. For example, the need to calculate bearing and distance drives the use of a microcontroller/software solution

### Arduino Pro Mini and code



## Decide: Hardware vs. Software

This is one of my favorite areas of the design process, because there are so many options. And those options evolve rapidly as the maker community innovates. You have a few decisions to make here, but here's one example of how to categorize them: does your product have components that move, or is it stationary? If it moves, it will do that with hardware. If it's stationary, will you implement features with software and a microcontroller (with lights, buzzers, or a display, for example). Will you have knobs or switches for user input (hardware), or will you use sensors (hardware and software)? Don't get scared by the need for a microcontroller like an Arduino. They're easy to use, affordable, and have a huge user community that has solved many challenges and has readily shared their solutions in online community tutorials. A microcontroller gives you the flexibility of a hardware/software combination, and allows you to build high-powered capabilities into your project for little money. Also, if your product must perform a calculation, make a decision, or connect to the Internet of Things, then you're going to need some computing horsepower. A cheap microcontroller can work wonders. A Raspberry Pi can work miracles for the money it costs.

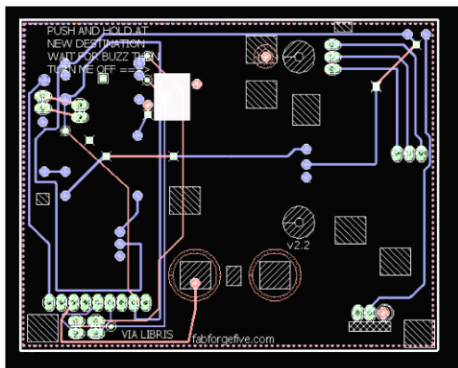
I love things that move. I like that good old-fashioned mechanical feel. I made a conscious decision with my GPS Treasure Book to implement many features in hardware (with servos that rotate indicator wheels). I could have implemented these indicators with a display or an LED light array. That would have been MUCH simpler for my design, and still would have looked good. But I decided to take a harder route and make things move. I felt like a display wouldn't be much different from a smartphone app. Decide what inspires you, and focus on that. Keep a list of the parts you need, including the source, their price, dimensions, etc. You'll thank me later.

There's another benefit of a hardware/software combination. You can often make a design or behavior change by simply changing your software. This is much faster than a hardware modification. For my example, I discovered that the sub-micro servos I use for my clue wheel have some variation in the way they rotate. A 20-degree rotation isn't the same from servo to servo. The COLD, WARM, HOT wheel locations varied from servo to servo (and therefore from box to box). I had two options: make a custom decal for each servo, with *slightly* different clue locations, or use one decal version and simply change the rotation values in the software for each box. It's easy to change the rotation values in software, and I had to use custom values for each box anyway for the electronic compass calibrations.

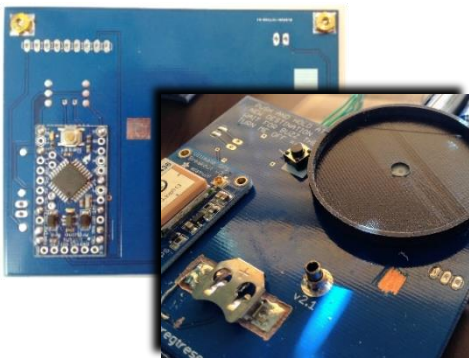
### BOARD GUIDES

Here's a detailed resource to help you choose the right microcontroller, based upon your product and its applications

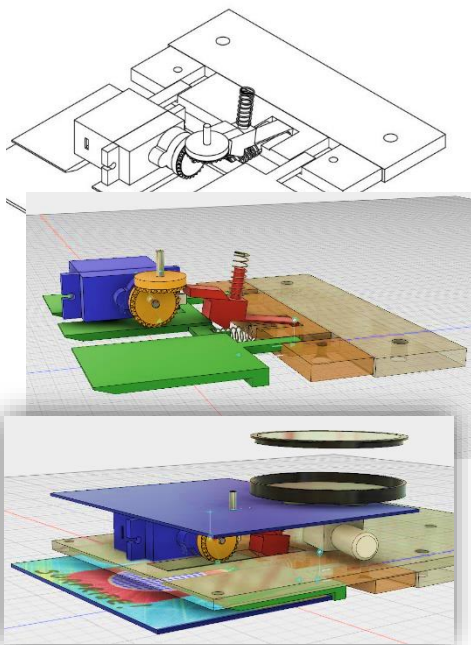
<https://makezine.com/comparison/boards/>



## PCB DESIGN AND FABRICATION



## 3D ASSEMBLY DESIGN



## FOR MORE INFORMATION

EAGLE PCB Design  
<https://www.autodesk.com/products/eagle/overview>  
 Autodesk FUSION 360 CAD  
<https://www.autodesk.com/products/fusion-360/overview>  
 Digital Design and Fabrication Services  
[fabforgefive.com](http://fabforgefive.com)

## The Design Process

So far, you've done some necessary "systems engineering" work in the initial stages of your project. You did some brainstorming, and made some lists. Those efforts will serve you well, no matter how you proceed. Now the real work begins - it's time to design the hardware and software components. Next, we'll cover 3D design, software design, and their integration. Yes, it's some involved stuff. If this is beyond your ability, don't despair. You have all the info you need to get help. Your Product Summary, Sketches, and Product Parameters are the foundation for a designer or fabricator to start your project for you.

A good digital designer can produce 3D models, photo realistic renderings, and 2D drawings within hours or days. This may be sufficient for an investor to understand your product. If you want something tangible, you might use 3D printed components. If you'll manage this yourself, be sure to ask for the proper file types (STL, OBJ, etc).

If you need help with an embedded microcontroller and associated software, a good developer can complete a first version within days or weeks. You may need to hire different developers for different components of your product. But a good digital designer/fabricator should have the skills to cover most of the work in one shop. I decided to do all this design and integration work myself. I'll walk you through my process and my lessons-learned.

This is also the time to decide what your prototype will be. Is it simply a non-working mockup (perhaps photo-realistic 3D renderings or a 3D printed model)? Or is it a fully-functional working prototype? Is it full-sized, or a scale version? Will it involve a smartphone app? If so, you'll likely want to mock-up only the user interface of the app in this early prototyping stage. A full app development is a much more involved process.

## EXAMPLE DESIGN PROCESS

My GPS Treasure Book product prototype required a custom Printed Circuit Board, off the shelf components and electronic modules, and 3D printed and laser-cut parts. It has enough variety to be a good example of the overall prototype process. I built 10 copies - enough to share with family and friends, but not too many to be a problem if my design was flawed.

**Printed Circuit Board.** I started with a custom PCB design. I had a crucial decision to make first. I would be using several different product "modules" - an Arduino Pro Mini, an Adafruit Ultimate GPS, a servo controller, and an electronic compass. I could have designed a PCB that would essentially rebuild all those products on one PCB. BUT that was well beyond my ability at the time. It would have required me to purchase (and solder) many tiny surface-mount components. So, I opted for a slightly different approach. I used EAGLE PCB (Autodesk) to design a master PCB onto which I could solder various modules (Arduino, GPS etc.). My design was still complex - it required a two-sided board, with GPS module on the top and Arduino on the bottom. But EAGLE has this capability.

Before sending my PCB design to a fabricator, I built my own board to test the system. It was crude, but it worked! I ordered 10 custom PCBs from a fabricator. Despite checking my design before ordering the PCBs, I realized as soon as I received them that I had made a

design error. I had to rework a few paths on every PCB. No big deal for 10, but it would have been a huge effort for 100 or more. Triple check your design before you order custom PCBs! It's a bit difficult, especially with multi-layer boards, but do it. Don't make my mistake.

**Laser Cut Parts.** I needed three clear acrylic pieces for my prototype. I ordered them custom-cut from a plastics supplier, and drilled my own holes with the help of a cardboard template. This was fine for 10 prototypes. But for larger quantities, I will laser cut the pieces (and the holes).

**3D Design.** This is where I made my first BIG mistake. Remember my earlier Product Parameters? I decided to use 2 servos/motors for three different motions - the compass wheel, the clue wheel, and the gift card dispenser. There was good reason for this (see the Product Parameters section). But it also created a complex design challenge for me. I needed the clue wheel servo to also trigger some type of dispenser. I wanted the dispenser to "pop" the gift card, in a surprising way, from a hidden slot in the bottom of the book. It had to have a mechanism triggered when the rotating clue wheel hit the "you've arrived" GPS location.

I decided NOT to invest my time in a 3D design, and instead tried building prototypes of different versions of dispensers. I tried five different versions, all using some sort of spring (for one I used a large safety pin). The design was constrained in several ways - mostly by the small space that the dispenser mechanism had to fit into, but also by the motion of the clue wheel.

Each time I created an updated version, it would have one of two problems. It would either work well but be ridiculously difficult to make, or it would be easy to make and work intermittently. I discovered this when I started making prototypes to give to friends and family. I needed a design that enabled easy "making". More about that later. I spent months of work on many versions, each with their own problems.

When version 5 was a failure, I threw up my hands and decided to invest my time in creating a good 3D Computer Aided Design (CAD) model. I used Autodesk Fusion 360. I spent a few hours over a weekend creating a complete assembly of my design, minus a dispenser. I added every part that would interact in any way with the complete system. The design allowed me to visualize the constraints, and to easily try different dispenser ideas without building anything. If it didn't work, I simply clicked "undo" and tried something else.

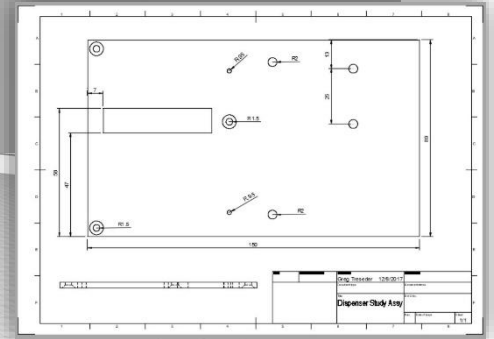
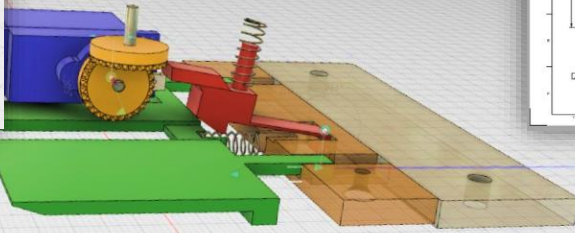
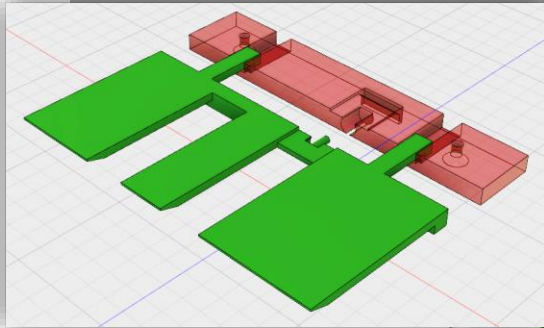
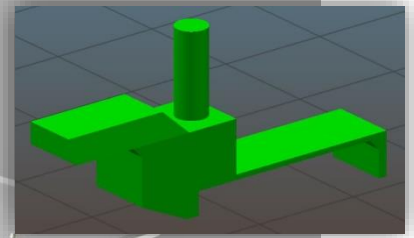
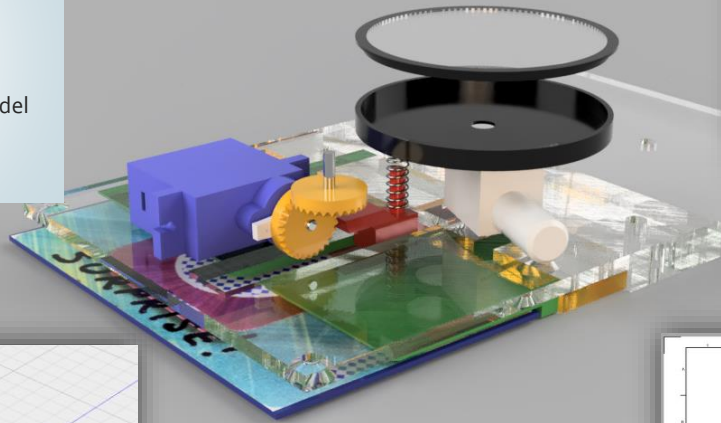
I spent a few hours over a week trying different dispenser ideas in my 3D model. I finally settled on a reliable, repeatable solution, using three 3D-printed parts. All total, I spent 10 hours on this design process. Had I done it at the beginning, I would have saved weeks and months of aggravation. Don't make my mistake. Start with a 3D model!

The model also enabled many valuable extras. I could easily 3D print the three components of my dispenser. I could also create photo-realistic 3D renderings, 2D drawings (for laser-cutting the acrylic mentioned earlier), AND I made a nice animation of the dispenser design and operation. This took some time but was worth it. I used that animation clip in a longer video to introduce my product. You can see all these items on the next page.

## 3D Design Results

Examples of the output of a complete 3D assembly design:

Photo-rendering. 3D parts. STL model for 3D prints. Drawing for laser cut. Animation video.



## Build It!

You've done a good amount of work to this point, but now the fun begins. Time to build your prototype! The work you've done so far will make this step easier than it would be with a trial-and-error approach. You'll still need to make revisions and perhaps some design changes, but there shouldn't be any "surprises". Surprises at this point cause aggravation, delays, and often increased cost. Better to handle them in the design phase when their impact is smaller. I picked my example project because it's a mix of hardware, software, 3D-printed and laser cut parts. Your product might be vastly different, but in general you'll follow the same process.

I started with the Arduino software code (my Arduino "sketch"), to make sure the Arduino could handle all the necessary functions. This included the i/o for a variety of external devices. The Adafruit Ultimate GPS Breakout uses serial communication. The electronic compass, a LSM303, uses I2C. The servos use digital pins, and my custom-designed compass servo also needed an analog pin to measure feedback. My project also includes a small disk vibrator to provide user feedback, which uses another Arduino digital pin.

By the way, the Arduino uses serial communication for programming, and there

is only one serial "port" hardcoded on an Arduino. The GPS module also uses that serial port. A connected GPS module interferes with uploading a program sketch to the Arduino.

Ideally, I would have shifted the GPS to a couple of digital pins using the Soft Serial Arduino library. It uses software to mimic the hardware serial interface. BUT when I built my first version, I found that Soft Serial and servos don't work together at all. They interfere with each other's timing. I added some simple jumpers to my PCB design, so I can disconnect the GPS for programming. An

### FOR MORE INFORMATION

The Arduino community is a huge and inspiring resource for help with project ideas, coding tutorials, etc. Start with these sites:

<https://www.arduino.cc/en/Guide/homepage>

<https://forum.arduino.cc/>

A full service digital fabrication shop can help: <http://fabforgefive.com/>

[Click for 3D animation video](#)

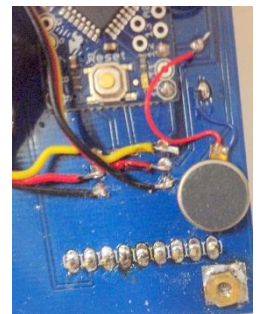
example of a hardware change to solve a software shortcoming.

During my build process, I documented EVERY step with photos and notes. Including any special tool setups. I therefore created an

assembly and manufacturing guide on the fly. I highly recommend this! You'll need it when you ramp up your production, or even if you're building only 10 items for family and friends. Doing it now hardly takes any extra time. Also, add lots of comments to any code that you write!

I also discovered during my process that some of the wire routing,

especially the servo cable, was difficult due to limited space. There was no way around this for the initial 10 PCBs that I had. But I immediately made a change in the design for version 2 of the PCB to fix the issue. Make those types of changes when they're fresh in your mind! Like most Makers, I work on a project in phases, move on to something else, then come back to it, sometimes weeks or months later. It's hard to remember needed changes at that point.

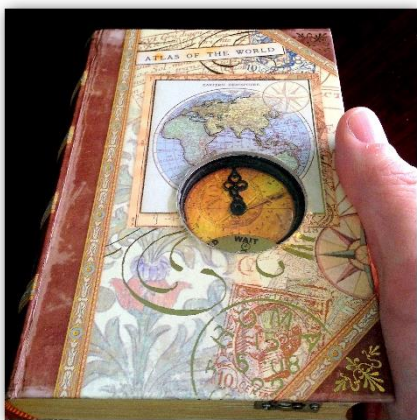


## Product Testing and Revisions

Once you've built a prototype, test it! Try it yourself in different environments, potential use cases, etc. Don't be discouraged if it doesn't work as well as you hoped.

Remember the gift card dispenser challenge I mentioned earlier? Use this self-test process to identify any possible issues. Then, have family or friends test your prototype. Before you do this, write up any special usage instructions that they might need. Pretend they're a customer that just unboxed your product. You want to test their entire experience, without your guidance. You won't be available to guide every potential user or customer anyway.

I had a fun time in this process. I packaged my prototype into a nice "book box" that I purchased at Michael's. I would soon buy MANY of these boxes. I picked different friends and family members, and gave them a GPS Treasure Book with a pre-loaded gift card for a merchant location near them. The location was pre-programmed into the EEPROM of the Arduino. I also gave them very minimal instructions. Remember that they had no idea a gift card was inside. I wanted to see if they could begin their "treasure hunt" just by turning on the box and following it, and I wanted them to be surprised. It worked! My favorite was a phone call I got from a friend – "I had no idea what was going to happen when I arrived at the place this box was leading me. But I'm here in front of *merchant*, and a gift card just popped out FOR THIS STORE! Totally cool!"

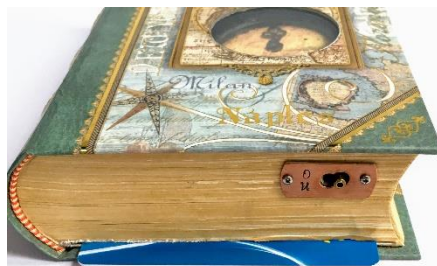


I also received some important feedback. My box doesn't tell its user how far they are from the "treasure" location. It could, but I like keeping it a bit of a mystery. It just gives COLD, WARM, HOT clues. One of my users

pointed out that they had no idea at the beginning how far they would need to travel. Car? Bike? Walk? I added icons on the clue wheel that represent a car, or footsteps. I also tell a person at the beginning roughly how far they will need to travel.



My mom lives far from me, and during her treasure hunt the compass wasn't pointing the correct location. I found this hard to believe. Math is never wrong, correct? I eventually discovered an error in my code in the trigonometry function that calculates bearing. It was a subtle and strange error that wasn't apparent around my home latitude/longitude. But it was a definite problem in other areas of the country. Easy to fix, but I otherwise wouldn't have known.



Another good suggestion was to add sounds to the box, to keep children engaged during the treasure hunt. Great idea! My updated version includes the Adafruit Audio FX Sound Board. It plays a bunch of funny sounds through a small speaker on a random basis. I'll need to update my PCB design again, but that's easy to do. Thankfully, there's room for the sound board just above the Arduino.

Keep careful track of your revisions, whether to hardware, code, etc. I have a revision history document that details the dates and changes made with each version. Its been a lifesaver – especially when I come back to this project after a few months off, and can't quite remember where I left off.

## PROJECT IDEA AND INSPIRATION SITES

### Adafruit

Visit the Adafruit site! Founder Limor Fried (Ladyada) and her team have created a fantastic catalog of products and associated tutorials. Their blog has daily updates with great ideas, and their learning forums are inspiring.

<https://www.adafruit.com/>

### Make: Magazine

Make Magazine is on "a mission to empower makers around the world". Make is the founder of the annual Maker Faires, now across the US and the world. Attend one if you can! If not, their website has a wealth of information, and has great projects for young makers as well.

<https://makezine.com/>

### Thingiverse

MakerBot's Thingiverse is "a thriving design community for discovering, making, and sharing 3D printable things." Thingiverse is a website of 3D objects. You can browse for an object of almost any type, download it, and 3D print it. No 3D printer of your own? Send the file to a 3D printing shop and they'll make it for you. Need an unusual part to fix something in your home? Check Thingiverse. You may just find it there, uploaded by someone else that had the same problem.

<https://www.thingiverse.com/>

### SparkFun

Like Adafruit, SparkFun manufactures and sells microcontroller development boards and breakout boards. All products designed and produced by SparkFun are released as open-source hardware. Great website with lots of tutorials.

<https://www.sparkfun.com/>

### Pololu

If you're building something that moves, check this place out. Pololu specializes in robots and associated motors, sensors, and electronics. They have a vast product catalog and a nice forum and blog.

<https://www.pololu.com/>

### Fab Forge Five

Fab Forge Five is a digital fabrication services company, specializing in digital fabrication • 3d design • prototyping • microcontrollers • coding. They're makers, and they want you to be one too.

<http://fabforgefive.com/>

## Documentation

Yeah, not the most inspiring topic. Don't despair. If you've followed the suggestions in this guide, you've already created a bunch of valuable documentation. At this point, depending on the nature of your product, you have:

- Product Summary
- Descriptive Sketches
- Product Parameters
- Printed Circuit Board designs
- 3D Designs and/or 2D drawings
- Animations or videos
- Build instructions (with photos)
- Revision History
- Bill of Materials (parts list with sources, pricing, etc.)
- Software code with lots of comments (version summaries, variable descriptions, etc.)

See, that wasn't so painful. Now, write a bit longer description of your product. You may need a few different versions, depending on the audience. Here's one for my GPS Treasure Book, focused more on a maker audience:

*The VIA LIBRIS uses an Arduino Pro Mini (under the circuit board), a magnetometer, an accelerometer (for compass tilt compensation), a GPS module, a sub micro servo (clue wheel rotation), and a very special continuous rotation gearmotor servo (compass pointer rotation) which I custom designed and built after trying many alternatives.*

Don't forget: give credit where credit is due. If you've used code, images, designs, sound files, etc. from others, be sure to give them attribution credit. For example, my Treasure Box uses code modules from Adafruit, and I have the following attribution on my website and in product documentation: *Contains code for Adafruit GPS modules using MTK3329/MTK3339 driver written by Limor Fried/Ladyada for Adafruit Industries. Copyright (c) 2012, Adafruit Industries.*

The license grants for software, 3D parts, etc. can often be found in the *license.txt* file that is included in the item you download. Be sure to check it, and don't forget to revisit when you make a change or upgrade to your product. For example, my latest Treasure Book version will have sound files licensed for commercial use via Creative Commons, and I'll be sure to credit the authors as they've requested.

## Easy Marketing Materials

One way or another, you're going to market your product prototype. This could be as simple as explaining it to friends and family. As complex as selling to a wide audience, made up of different interests and target communities. Perhaps you're seeking investors and/or venture capital funding. They'll want to see more than just a prototype. Thankfully, if you've followed this guide, you have written descriptions, photos, models, animations, videos, etc. Your content is ready for you. Now pull it together in different forms. A website, a brochure, social media, etc. Here's an example of my product website. Recognize the content? The vast majority was created during my prototype phase.

appear in your feed. Those companies have a process that works. Learn from them.

## Conclusion

I wish you much success in your efforts! Laugh at your mistakes, celebrate your successes, and share your lessons learned! And remember these quotes along the way:

*"The short successes that can be gained in a brief time and without difficulty are not worth much." – Henry Ford*

*"Do or do not. There is no try." - Yoda*

**TREASURE BOOK**

The VIA LIBRIS is a paperback-sized treasure book. It leads its user to a predetermined location, using a special compass dial and by giving COLD, WARM, or HOT clues.

**Not All Who Wander Are Lost**

Looking for adventure? Or at least a fun treasure hunt?

The VIA LIBRIS is a paperback-sized treasure book. It leads its user to a predetermined location, using a special compass dial and by giving COLD, WARM, or HOT clues. At the treasure location, a gift card pops out of a hidden slot in the book. It's a great surprise for the gift card recipient! A new location can be programmed anytime with the push of a button. Each book is handmade, one of a kind, beautiful, and reusable.

The VIA LIBRIS uses an Arduino Pro Mini (under the circuit board), a magnetometer, an accelerometer (for compass tilt compensation), a GPS module, a sub micro servo (clue wheel rotation), and a very special continuous rotation gearmotor servo (compass pointer rotation) which I custom designed and built after trying many alternatives.

When you arrive at your destination, a fun surprise happens!

Contains code for Adafruit GPS modules using MTK3329/MTK3339 driver written by Limor Fried/Ladyada for Adafruit Industries. Copyright (c) 2012, Adafruit Industries. <https://www.adafruit.com/>

Contact Us



## About the Author

Greg Treseder is the founder of Fab Forge Five, a digital fabrication services company. A former aerospace engineer, he has a background in electronic test and measurement equipment, CAD software, robotics software, and big data analytics.

Greg believes that the process of *making* is a fundamental aspect of human culture, and one that should be openly shared. Plus, it's good clean fun.

I'm no marketing expert, and there are many valuable guides on this aspect of your endeavor. The good news: there are many ways to market a product or idea, and they're easier than ever with the options of social media. Browse your favorite social media and notice how products you might like

**Send me an email!** I'd love to hear your feedback, your comments, your projects, and your wacky science jokes. You can reach me at [admin@fabforgefive.com](mailto:admin@fabforgefive.com).

*Greg Treseder*



digital fabrication • 3d design • prototyping  
microcontrollers • coding

[fabforgefive.com](http://fabforgefive.com)