BUILD NOTES

Within its steel frame and aluminum skin, our 15-foot backyard rocket encompasses microcontrollers and LEDs, pneumatics, vibration and sound effects, and the joy of making.

Rocket-Ship Treehouse

BY JON HOWELL AND JEREMY ELSON

■ **RARELY DOES BUILDING A TREEHOUSE** require welding, grinding, painting, riveting, bending, crimping, plumbing, brazing, laser cutting, sound design, printed circuit board fabrication, distributed network protocols, an embedded operating system, sewing, and even embroidery. Ours did: a backyard rocket-ship treehouse.

The Ravenna Ultra-Low-Altitude Vehicle (RULAV), named for our neighborhood in Seattle, Wash., is a hexagonal capsule 7½ feet high, atop a tripod of the same height, for an overall height of about 15 feet. The frame is welded mild steel with riveted aluminum skin and a hinged entry panel and window. A ladder made of steel cable runs from the ground to the rocket's floor. A rigid interior ladder lets kids climb up and peek out the top.

Inside the rocket are nearly 800 LEDs forming dozens of flashing numeric displays spread across 14 control panels, each with an acrylic face laser-cut and etched with labels such as "Lunar Distance" and "Hydraulic Pressure." Working buttons, knobs, and switches operate the rocket's software.

Underneath the capsule are 3 "thrusters" that shoot plumes of water and compressed air under the control of a pilot's joystick, simulating real positioning thrusters. Takeoff and docking sequences are augmented by a pneumatic paint shaker that simulates the vibration of a rocket engine (Figures A and B, page 146).

Sound effects complete the illusion, with a powered subwoofer that gives the rocket a satisfying rumble.

An Engineering Playground

"It's for Jon's son, Eliot," was our justification to friends and co-workers after describing the growing list of planned features. As our ambitions spiraled and months of construction stretched into 2 years, it became transparent that the treehouse was just as much an engineering playground for the adults, a place for us to share our joy of making and teach it to the kids. Now that the rocket is complete, it's a fun plaything, but the journey was even more rewarding than the result.

The rocket was conceived in 2008 after Eliot's mom suggested that Jon install a swing set under the trees in the backyard.

"A swing set? *Everybody* has a swing set."

Mom said, "Then what *are* you going to build? A rocket?!"

"Yes. Yes we are. That is exactly what we're going to build."

Chassis and Skin

We went to Boeing Surplus and brought home a few big aluminum sheets and some sturdy aluminum tube. Our first idea was to build a geodesic structure formed entirely by bending and riveting, but early prototypes wouldn't stand up, proving that we really didn't know much about mechanical engineering.

Finally, we realized weight wasn't a design

Jeremy Elson and Jon Howell

constraint for a rocket that never leaves the ground; it would be just fine to use steel. A new design was born: a welded steel chassis skinned with aluminum (Figure C).

The main challenge in constructing the chassis was the many compound angles required where the steel members meet. We proofed the design with a series of paper prototypes and built simple jigs that made it possible to make compound cuts with a steel cutoff saw. Wood jigs also helped greatly with accurate assembly (Figure D).

Once the frame was underway, we built a floor and a window screen from expanded steel, steel handles for the hinged pieces, and even a stainless-steel rope ladder (Figures E, F, and G). Eliot wasn't old enough to weld, but he pulled a lot of rivets (Figure H)!

Pneumatics and Plumbing

Once the chassis was done, we gathered a few friends around for an old-fashioned rocket-raising. We also dug a trench across our backyard and plumbed it with ½" copper pipe.

A big green handle lets air into the rocket with a satisfying whoosh, and a needlessly elaborate "distribution manifold" has dials that spin and twitch as compressed air is used (Figure I). Visit rocket.jonh.net for a schematic of the final plumbing system.

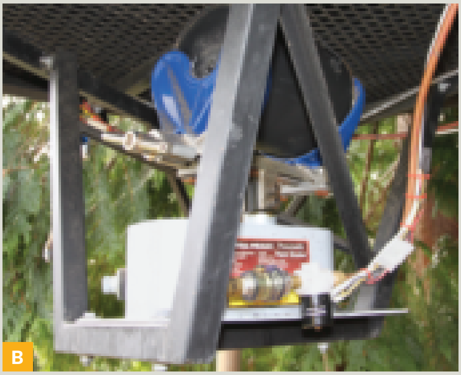
Water for the orientation thrusters is supplied from a jug (Figure J). To refill the jug, a water line runs through the trench from the garden spigot (Figure K), where an electric solenoid lets the pilot "refuel" by pushing a button in the cockpit.

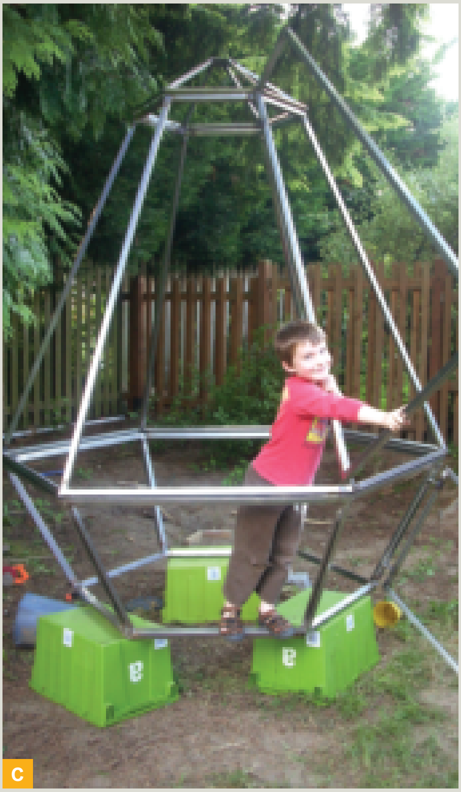
Electronics and Programming

While Jon was building the structure, Jeremy was working on the electronics. The original goal was modest: fill the rocket's interior with dozens of flashing lights and randomly changing numbers.

We designed a circuit board that would


A


B


C

WELD IT

Welding seems intimidating, but making basic joints in mild steel is pretty easy, and it's enough for a treehouse. You'll need a flux-wire welder, a helmet, gloves, an angle grinder, a wire brush, and perhaps an abrasive cutoff saw. Low-end versions of these can be had for around $300.

Buy mild steel 1/16"-wall square tubing from your local steel yard. (You can reclaim scrap if you clean off paint and rust, but don't mess with galvanized, chrome, or stainless; they make toxic fumes.)

Practice making a simple bead on one piece, then butt-joining 2 flat pieces, then joining on a right angle. Your goal is to use the heat of the arc to melt both work pieces. Cut your practice pieces apart to see that you really melted the metal on both work pieces, so that they froze back into one continuous piece of steel.

RIVET IT

Once you've got a steel frame, riveting sheet steel or aluminum to it is easy. You'll need a pop-riveter ($6), an electric drill, and a 1/8" bit, plus a few more for the ones you'll break. Drill a hole, place a rivet, squeeze the riveter until it pops, and repeat.

BRAZE IT

Some of our pneumatic plumbing is soft copper tubing with compression fittings that require only a pair of wrenches to install (ask your plumbing store how). But those fittings are pricy, so where possible, we used sweat-fit fittings on ½" copper water pipe.

To get the job done, you'll need about $40 worth of tools: a tubing cutter, a pro-pane torch, plumbing solder, flux paste, and a plumber's wire brush.

Brush the corrosion off the outside of the end of a pipe and the inside of the fitting, smear flux on both surfaces, and push them together. Heat the joint until the copper subtly changes color, then touch the solder to the joint; when the joint is hot enough, the solder will get sucked in. Practice on scrap, then cut the joint apart to ensure that the solder filled it.


D


E

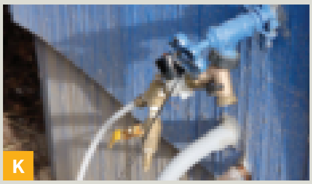

F


G


J


H


I


K

BUILD NOTES

light up 8-segment numeric LEDs. They're cheap, bright, and available in a wide variety of colors and sizes. Each board holds up to 8 digits. Each digit's 8 segments are attached to one output of a 74HC259 latch, a small memory device that can be programmed to start or stop current flowing through each of 8 outputs.

To turn an LED segment on or off, the processor specifies the desired LED digit number using 3 output pins, and the segment number using another 3. Three pins also specify the board number (each board has switches that set its ID). Another pin specifies the desired LED state (on or off), and a final "strobe" signal indicates that all the other signals are ready.

Two layers of 74HC138 demultiplexers feed the strobe only into the intended latch (Figure L), and a single segment turns on or off as a result. A ribbon cable distributes the 11 control signals from a single processor out to all 8 boards. This way, 512 LED segments can be controlled by a single processor.

An early prototype worked, but even with only 2 LED digits, it took a week of evenings to construct (Figure M). We had to carefully modify a prototyping board with a rotary tool, and solder in each component and wire connection. This manual labor was time-consuming, error-prone, and not very fun. If we built all the boards by hand, our goal of "dozens" of LEDs would likely remain out of reach.

Sane treehouse builders might decide to scale back their ambitions. We went the opposite direction: why not design our own printed circuit board (PCB) and have it fabricated in bulk? The only problem was, we hadn't done anything of the sort before — in fact, we'd only recently learned how to light up an LED.

The thriving DIY community came to the rescue. We learned basic PCB design from online tutorials (Instructables, Adafruit, Seattle Robotics Society, UK Electronics Club) and discussion boards (SparkFun, AVR Freaks). We scoured the web for the best deals on board manufacturing.

After a few weeks of learning CadSoft Eagle, we had a layout we liked: 6.25" by 2.8" boards, routed tightly enough that a row of 0.56" LEDs could be mounted without gaps (Figure N). We sent it out for fabrication and spent the next 2 weeks giddy with anticipation. There are a lot of good reasons to become a maker, but perhaps none better than the joy of moments like when our first boards arrived (Figure O).

The rocket's final design uses 12 boards controlled by 2 processors that coordinate over a TWI network (2-wire interface, compatible with I^2C). Not everything worked as expected: it turns out that running data lines in parallel over long distances without sufficient ground paths is not a good idea. We

ETCH IT

Though there is a learning curve, creating your own printed circuit boards (PCBs) is surprisingly affordable.

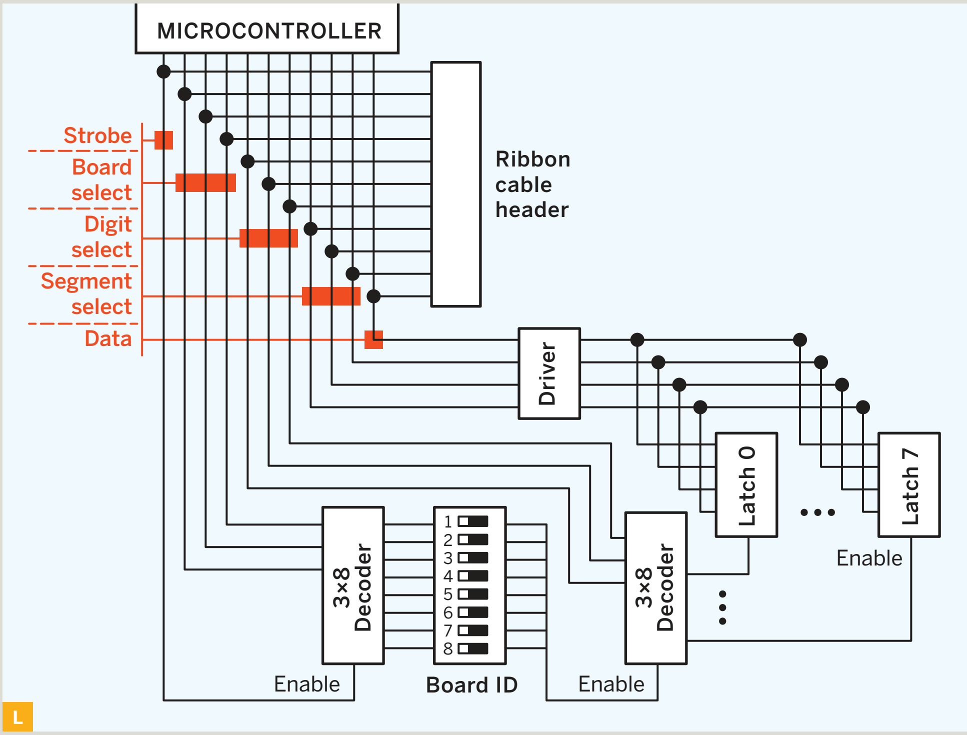
Design your PCB in CAD software such as CadSoft Eagle, popular among hobbyists, or an open source alternative such as KiCad or gEDA. As you add each component to your circuit's schematic, a corresponding component appears on a physical layout drawing. Position these on the physical layout to minimize the distance between connected components. Then run the "auto-router" that converts the connections you've specified into copper traces.

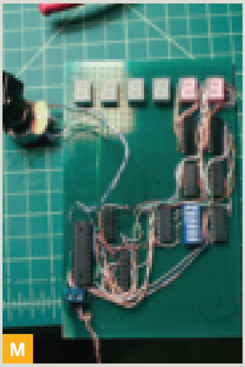
PCBs can be fabricated at home in just a few hours (*see MAKE Volume 02, page 164*). However, dangerous acids are required, and it's hard to drill all the holes precisely.

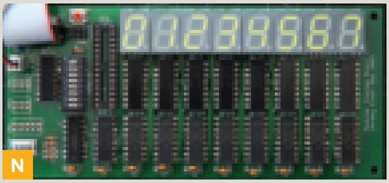
Professionally made PCBs are worth the wait: they're far more precise (allowing more compact designs), can have more than one layer, and are covered with "solder stop" (unsolderable lacquer) that makes correct soldering much easier.

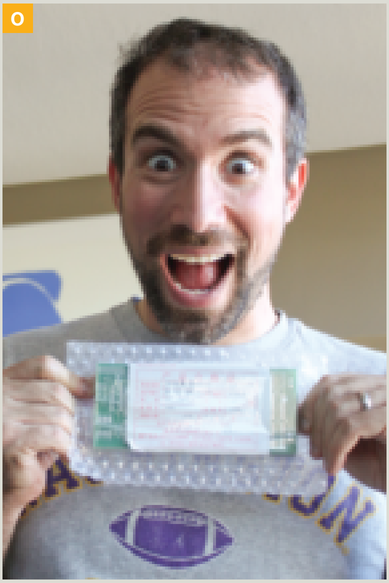
Sending your board out for fabrication is almost as easy as sending a PDF to a print shop. The Gerber format is the industry standard. If you have a small board, batch fabrication services can aggregate your design with other hobbyists', amortizing the setup cost; expect to pay about $5 per square inch for 3 copies. (DorkbotPDX runs an excellent service that anyone can use.)

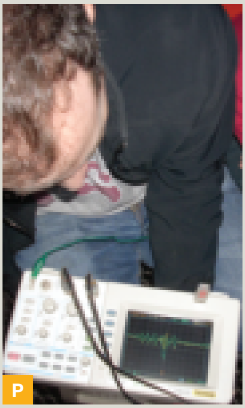
For larger sizes or quantities, it pays to go directly to a board fabricator. We love OurPCB; they charge $50 setup plus 15 cents per square inch. We bought 20 display PCBs for the rocket for about $150.











PROGRAM IT

It's never been easier to integrate microcontrollers into your DIY projects. Atmel's ATmega328 (the chip at the heart of the Arduino) has dozens of pins that you can use to drive logic, built-in processor-to-processor networking, a 6-channel analog-to-digital converter, a serial port, and much more. With 2K of RAM, it costs $4. For larger projects, the fancier ATmega1284 ($8) has 16K of RAM and more pins.

These chips come in easily solderable DIP packages and need no external components other than power/ground and a pull-up resistor on the reset pin.

Bring those 3 pins plus another 3 out to a standard 6-pin header, and your new circuit is programmable using any AVR in-circuit programming device, like Adafruit's $22 USBTinyISP. The software toolchain is free, including the compiler (GCC) and standard C library (*avr-libc*).

The Arduino is a great learning tool, but don't be afraid to add a bare microcontroller to your own custom board!

spent a few hours in the rocket with an oscilloscope (Figure P, preceding page).

Control Panels

We mounted the electronics in "control panels": aluminum boxes with transparent ⅛"-thick acrylic faces. We drew each face in Inkscape, placing rectangles where the circuit boards would go, and holes for standoffs and screws, then sent the drawings straight to a laser cutter to etch and cut the acrylic. (Laser cutters can be found at your local hackerspace, or try an online service.)

We made the pattern for each box by extending rectangles out from each edge of the face and adding ½" flanges all the way around. We printed the pattern and taped it to a sheet of aluminum, then marked the corners and holes with a punch, scored the lines with an awl, cut out the shape, and bent it on a homemade sheet metal brake. Each box is open at the bottom for cables to enter (Figures Q and R).

Standoffs attach the circuit boards to the acrylic face, and sheet metal screws attach the face to the box. The aluminum boxes are riveted to the rocket wall. The resulting control panels are sturdy enough to withstand a hapless foot.

Booster and Thrusters

Next we turned to the pneumatics that would power the "orientation thrusters" and "booster." The booster is a paint shaker that gives the rocket vibration during "takeoff." The thrusters are automotive engine-cleaning wands that aerate water using a supply of compressed air, producing a convincing jet blast of mist (Figure S).

Our original idea was to place air valves near the pilot's seat and direct air through hoses to the jets. A prototype didn't work very well — a hose itself can store enough air that the jets gave us unsatisfyingly soft whooshes rather than the short, sharp blasts we wanted. Jon suggested putting the valves next to the jets, actuating them at a distance with bicycle cables leading down from the cockpit.

"Why not buy electrically actuated valves instead?" Jeremy asked.

"I don't know," Jon said skeptically. "Aren't they expensive?"

"I found some for only $15," Jeremy said. "Plus, the controller will be electrical rather than mechanical, meaning it becomes my problem."

"Sold!"

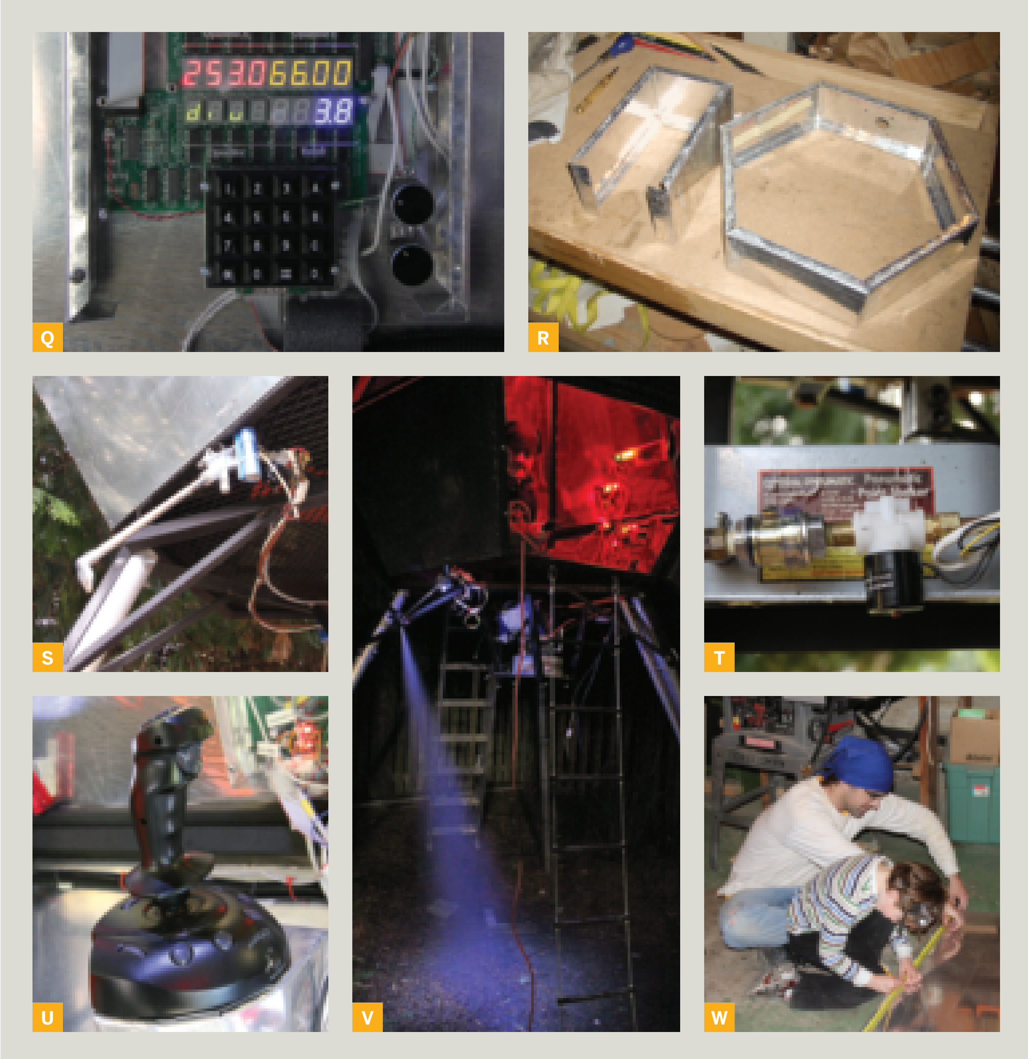
We bought a handful of solenoid valves from eBay (Figure T) and hand-built a simple electronics module that lets a processor control high-current devices. It has several power transistors, each with its gate connected to one of our boards' latch outputs. This lets software toggle the thrusters in exactly the same way as the LEDs.

Electrically actuated valves have another benefit: the pilot can use a joystick (Figure U). Vintage PC joysticks that use game ports rather than USB have simple electrical interfaces: 2 linear potentiometers, which vary between 0Ω and 100kΩ as the control stick is moved through the *x* and *y* axes. We connected those pots in series to fixed-value resistors, forming voltage dividers whose values are read by a processor's analog-to-digital converter. As the joystick is moved toward each corner of the rocket, the thruster in the opposite corner fires. The joystick's numeric *x*-*y* position is displayed on a panel, too.

Blastoff ... and Beyond

By now, the floodgates had opened. With all the lights and thrusters under software control, we realized the rocket's electronics could do far more interesting things than just display random numbers.

We created a takeoff sequence: once a launch code is entered into a 16-key keypad, a countdown is displayed on a control panel while audio from the real Apollo 11 sequence is played. At zero, the lights start to flicker, and the rocket starts to rumble from the movement of the paint-shaker and the bass



from our subwoofer.

In docking mode, the pilot guides a target onto a "docking clamp" represented by cross-hairs drawn on a LED matrix display. Joystick motions produce action in both the real world (thruster firing) and the virtual one.

We even created a rocket version of the classic video game *Pong*, to keep crew morale high during long trips to the Moon.

By the project's completion, we'd written about 30,000 lines of C code, including a miniature operating system and a simulator that lets us test and debug the software on our desktop computers.

The completed rocket is quite a sight! At night, the glow of the LEDs is otherworldly, and the illuminated water jets conjure dreams of space flight (Figure V).

Since Eliot was with us every step of the way, he also learned that toys aren't just something you buy, they're something we can build — together (Figure W). Because in the end, it's never really about the having. It's about the making.

For more photos, schematics, PCB layouts, and the continuing saga, visit rocket.jonh.net.

Jon Howell studies operating systems by day; after work he goes on adventures with his three junior astronauts. Jeremy Elson spends his free time flying airplanes, riding bicycles, and building electronics.