YOUR FAVOURITE ELECTRONICS IN A MAGAZINE!

DIYODE

MOBILE EMERGENCY CHARGER

DOES NATURE'S BOUNTY PROVIDE THE ULTIMATE SOLUTION?

Issue 004
Oct 2017
AUS: $9.95 inc. GST

ISSN: 2207-8045

CHECK OUT AN AMAZING SENSORY DEVELOPMENT DEVICE FOR KIDS
Using Arduino and some creative thinking for a great result.

LEARN HOW TO USE CAPACITORS EFFECTIVELY
They're useful and powerful, and surprisingly easy to use!

BUILD OUR FIRST DISCRETE PROJECT - A LOGIC PROBE
Convenient and fast diagnostics for makers everywhere!


TESLA

EDITOR-IN-CHIEF
Rob Bell

EDITOR
Analee Gale

TECHNICAL EDITOR
Bob Harper

STAFF WRITERS
Oliver Higgins
Daniel Koch
Tim Blythman

PROOFREADER
John Bourne

CREATIVE LEAD
Jennifer Holland

CREATIVE & DIGITAL TEAM
Luke Parsonage
Kayla Gourlay
Melanie Heard
Mike Lewis
Danielle Bell
Jacqui Creasy
Mike Hansell

CONTRIBUTORS
Tom Biskupic
Sascha Grant
Miles Hitchen
David Such
Dawid Verwey

DIYODE MAGAZINE IS AN INDEPENDENT PUBLICATION.

Published by B.E. DIYODE Pty Ltd

ACN 616 556 622
ABN 50 616 556 622

Level 1-2, 156 Mann St,
Gosford NSW 2250 Australia

Printed by PMP.

Newsagent distribution within Australia managed exclusively by Gordon & Gotch.

ISSN: 2207-8045

Print subscriptions are available to Australian residents for $9.95 per issue including GST. See diyodemag.com for international subscription pricing.

All prices quoted in AUD including GST unless noted otherwise. All prices are correct at time of printing and subject to change without notice.

ADVERTISE WITH US
media@diyodemag.com

CONTACT US
help@diyodemag.com
+61 2 4326 0160

FOLLOW US
@diyodemag

DIYODE Magazine's First Discrete Project

As a child, I grew up building Dick Smith FunWay kits, the Jaycar Short Circuits Kits, kits from Electronics Australia, and also from our friends over at Silicon Chip Magazine. They were a key piece of the puzzle that ultimately formed my understanding of electronics. They were also the cause of a few show-off moments to friends and family, to say the least. The engineering hours that have gone into developing those kits over the decades, fill many lifetimes. The copper tracks laid out could stretch to the moon, and the solder fumes could... well, at least we have lead-free solder now.

To the engineers who have assisted in the development of these projects, and the teachers who have adopted them into classrooms, I take my hat off to you. Technology evolves so rapidly these days, but there are still component-level projects that are not only fun to build, but teach us valuable principles and skills that are relevant to today's maker landscape.

It's for this reason, we have commenced development on a series of discrete projects. Our first is a handy logic probe. While there are many designs for logic probes out there, they're more relevant than ever. They're an invaluable tool for testing Arduino/Raspberry Pi systems (and of course, anything else using logic signals too).

While they're handy, board-level projects, they will focus highly on the education around what makes them work, while still providing practical utilities for a maker. We want them to compliment or assist your microcontroller and maker projects, just as much as your computer does. Of course, we may publish something that's purely for fun too, because who doesn't like a fun project, regardless of whether it's microcontroller based or not?

Our Technical Editor Bob Harper has been putting his decades of experience to excellent use to help these new projects come to life. Thanks to our retail partners Jaycar Electronics and Altronics, you'll be able to purchase a kit of parts complete with PCB. Naturally, the entire design of each project will be made open-hardware, so you can modify and adapt the design to suit your own purposes if you wish to do so.

We're also trying to use some off-the-shelf items in creative ways, which is demonstrated by our hacking of a 5V DC-DC converter to provide a convenient emergency phone charger. A handful of parts and a 3D printed case - it's simple and effective.

You may also note that Alarmduino has taken a break this month. That's because we have some brilliant new features for it in the next installment, so it'll be back in issue #5!

Please enjoy Issue #4, and thank you for your continued support.

Rob Bell

WIN 1 OF 6 SPARKFUN INVENTOR'S KITS!

VALUED AT $175 EA.

Thanks to our friends at Altronics, we have 6 Sparkfun Inventor's Kits to immerse you into the world of Arduino, with everything you need to create dozens of experiments. There's a SparkFun RedBoard (equivalent to an Arduino UNO) included, as well as a breadboard. There are 16 circuits to build, with all of the parts included. There's LEDs, semiconductors, and everything down to the jumper wires.

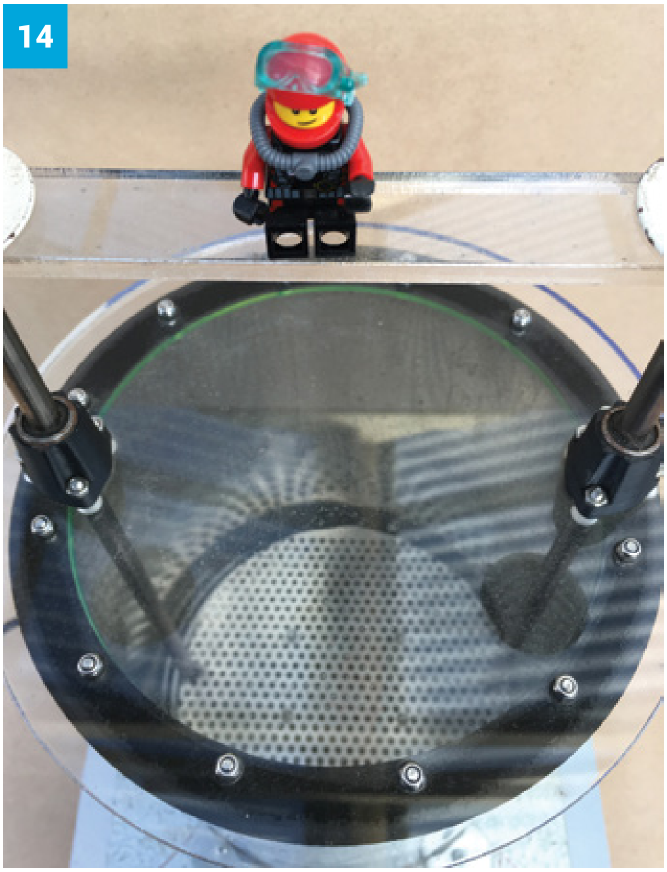
This is a great way to dive into Arduino, or expand your knowledge!

Head over to diyode.io/004comp and tell us why you want to win!

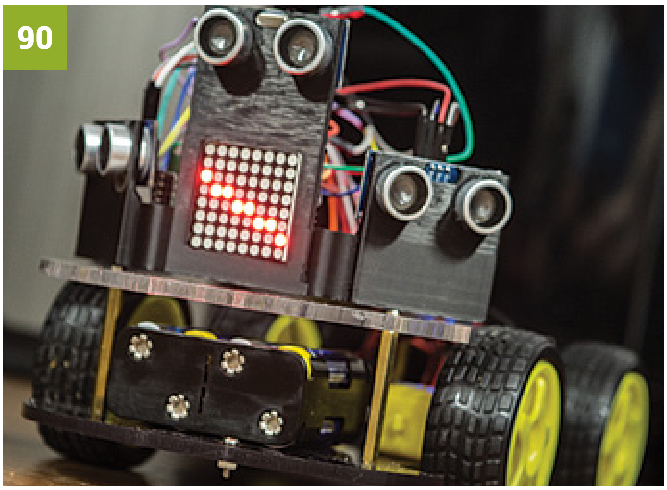
sparkfun ELECTRONICS

available at:

ALTRONICS


14

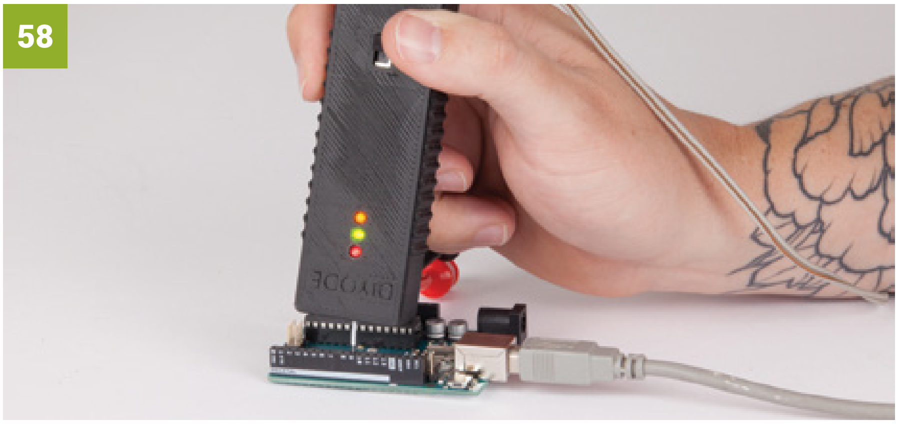

46
23.8


90


50


84
10.800V
0.001A
0.01W
0.005V
0.001A
0.00W
0.021V
-0.000A
-0.00W

58



32

36

40

FEATURES:

EDUCATION:

PROJECTS:

Spotlight on: Airblock

The Modular and Programmable Drone

The Airblock first appeared in 2016 on Kickstarter, pitched by Makeblock, the brains behind mBot and other cool robotics and STEM products. This clever innovation uses modules to create a six-axis drone or hovercraft, plus a handful of other weird and wonderful configurations. The Airblock has just landed (pun intended) at Jaycar Electronics, so we put one through its paces.

FIRST IMPRESSIONS

The packaging is very classy; solid construction, carry handle and a lid that magnetically closes. Inside, you're presented with a moulded cardboard insert securely holding six power modules (motors with rotors), and a master controller module, all hex in shape, 75mm wide and made from durable but light EPP foam. Within the box there is another moulded cardboard insert holding the hovercraft base, battery, charger, USB cable and protective covers for the power modules. There is also a colour user guide, a bag of spare rotor blades, and three sets of stickers for the fuselage, power modules and hovercraft. Cool!

INSIDE THE BOX

The hex-shaped control module sports an ultrasonic sensor, barometer, gyroscope, Bluetooth module, and microcontroller chip, with a battery compartment underneath. Up to six power modules (all supplied) can be connected to the control module via the electrical connections on each side. These modules have powerful coreless motors and plastic rotor blades and there is a protective cover for each side to stop mischievous fingers from being hurt. A 7.4VDC 700mAh Li-Po battery is included, along with a charger that plugs into any USB port, to provide the power for up to six minutes of full flight.

BUILD IT

Assembly is truly child's play. Our six-year-old helper assembled the hovercraft within minutes, just by looking at the photo on the box! Each module has a strong polarised magnet that fixes it firmly in its correct place on the master control module – genius!

THE APP

A tablet or smartphone is required to control the Airblock. The free 'Makeblock' app can be downloaded and suits all of the products in the Makeblock range. Once you choose Airblock you're asked if you want to Play or Create. The Play option asks if you're running in Air, Land or Water mode and provides a controller interface accordingly. The Create option enables you to customise and save your own controller interface depending on the mode selected. Pairing via Bluetooth is very easy also - it doesn't even require a pin code.

TIME TO PLAY

The fun part! At a touch of button, we had the hovercraft whizzing (quite fast!) across the tiled floor. Initially, the loud noise generated by the motors and spinning blades surprised us, especially Master Six who jumped a little in fright! It's quite surprising how much noise those tiny motors create, but you do get the impression of their mighty power. It handled the bumps into the walls and chairs with ease, and kept on going. Within no time we had a crowd fighting over who was going to control it next. Each of them were able to take control without instructions.

Before long, the thrill of operating the hovercraft soon turned to anticipation to build and fly the drone. Building the drone was quick and easy - just be sure to install the rotors in the right direction!

Choosing Air Mode in the app presents an interface similar to a handheld RC remote. Rise, descend, forward, backward, left, right. There are also three pre-programmed effects (roll, shake and circle) and a take-off/land button, which takes the complexity out of taking off and landing.

We found the Drone mode is better suited to teenagers and adults. We spent more time rebuilding the drone than flying it when the six-year-old was at the controls. It did demonstrate its durability though. There was no damage done to the modules, or the walls. Occasionally the rotor blades flew off with hard landings, but they are easy to pop straight back on. This also explains why spare rotor blades are included in the kit.

With an adult at the controls the drone was very nimble in flight. Take-off and landing was easy, and doing stunts at the press of the button was impressive. The rotor protective covers did fall off from time to time, which was annoying. It was easier to fly without them.

CODE IT TO THE NEXT LEVEL

This is where the Airblock really sets itself apart from the everyday drone. The app allows you to go code your own interface using Scratch 2.0 inspired drag-and-drop coding (as taught in many pre-schools now days). Brilliant! Set motor speeds, direction, and create events based on temperature, gyro, battery level, and much more. For those familiar with coding, it performs maths as well as the humble "if, then, else" and "loop" statements. It appears it's only limited by your imagination.

CONCLUSION

It is expensive if you compare it to a regular quadcopter or drone, but it makes up for it in its versatility and ability to be programmed. With the multiple configurations and being able to be combined with other toys such as LEGO®, there is endless hours of fun to be had (between charges of course). It is noisy, but it is fast, powerful and nimble. Because it's fun, and more importantly is able to teach kids coding we're giving the Airblock a big thumbs-up. ■

Shopping List:

Airblock by Makeblock is available at Jaycar Electronics:
www.jaycar.com.au

► **MAKEBLOCK AIRBLOCK MODULAR PROGRAMMABLE DRONE KIT**
KR9220 **$299**

On Location:
The Locals With A Leading Edge...

In the not-so distant past, the independent electronics store was commonplace in many towns and cities. It may be 2017 now, but it's important to remember they're still going strong.

Recently, we had the pleasure of attending the Leading Edge Electronics Annual Conference. It was awesome to speak to so many of the amazing people who are responsible for bringing the best in electronics to the doorstep of regional Australia.

When meeting these small business owners, what really stood out was the passion they have for what they do, and the life they bring to the industry. With time-tested values and honest service, they drive hard every day to bring the products to their stores that their customers want. Sure, Leading Edge Electronics might be boasting close to 100 stores, but walk into any individual store and you'll be greeted by the local owners, most of whom know many customers by name, and who take the time to make sure those people walk out with the very best solution they can possibly muster for their unique problem, whatever that may be.

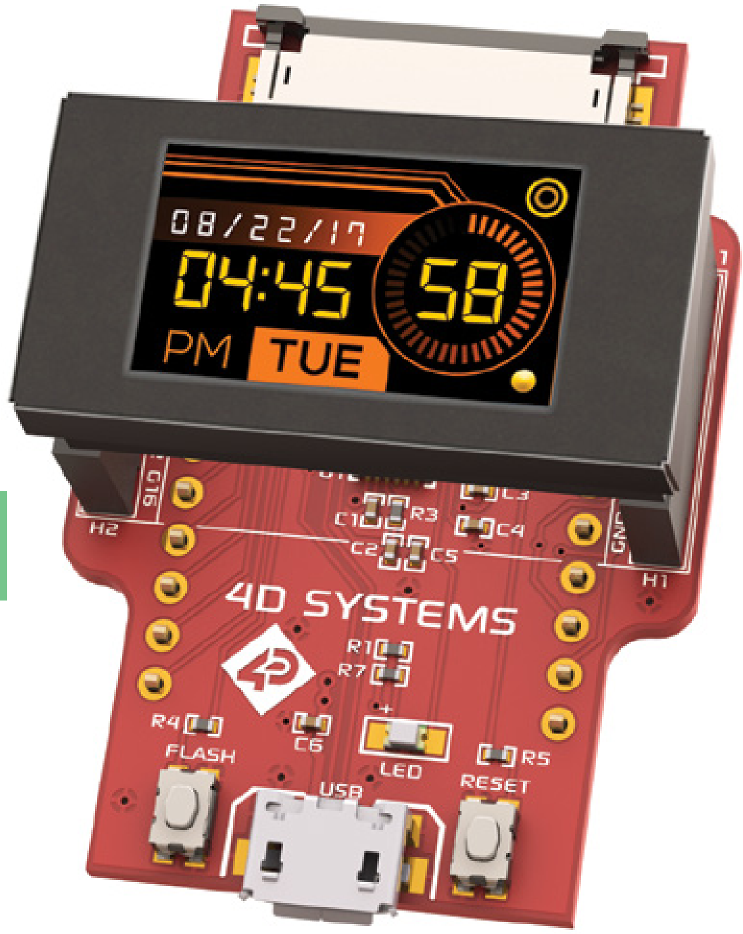
It's no secret that business has changed over several decades. The local store faces an increasing challenge in a cut-throat market. Their power now, stems from being a versatile group. They have the collective buying power that rivals large chain retail stores, but still have the flexibility to customise their stores to suit their market, their customers, and even their preferences.

It's an impressive model, and one we feel should be admired and supported. These business owners pour their heart and soul into delivering the best possible products they can to the marketplace. The Leading Edge Group was founded in 1986, and partners with world-leading brands to bring some of that muscle and might. But it's not all about business. Business can be a tough road, and the group provides comradery and a support network to keep the engines running at the heart of these businesses.

One thing that we never fully appreciated about The Leading Edge Electronics outlets was the vast range, and the quality of brands you'll find in each store. Walking through the trade exhibition, my eyes were opened; it's true, you'll find many of your favourite parts for hobby electronics. Arduino and Raspberry Pi products will be abundant, as well as soldering irons, components, and a huge range of the parts we use in many of our projects. But what was an unexpected find, are all sorts of amazing things from some of the biggest names in the industry. You'll find quality solar and power hardware, CB radios, surveillance equipment, high-powered WiFi gear, and so much more. It's a versatile and powerful range of products.

So next time you're considering where to go for your next purchase of electronics gear, remember the super hard-working, independently-owned retailers, who are part of the Leading Edge Group. ■

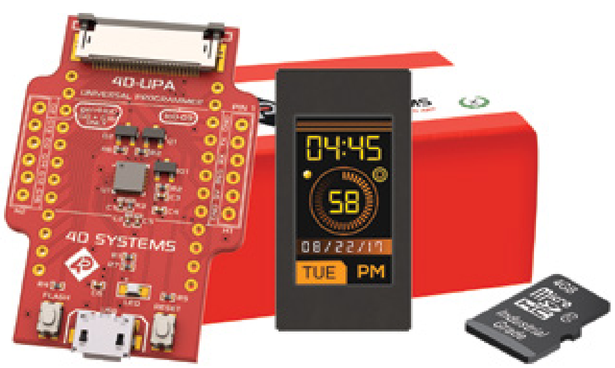
Coming Soon: The 4D Systems IoD-09TH



One of the limitations that currently surround Arduino, is its limited display capabilities. Absolutely, there are displays available and we often integrate them, but producing quality graphics can be a challenge. Obviously, in an Arduino board there is no on-board display processing, so an external display processor is really required for advanced display hardware.

Pushing display boundaries is part of 4D Systems' DNA, and their latest product is no different. The IoD-09TH integrates a 0.9" TFT LCD screen, on an ESP8266 WiFi chipset. This is a powerful combination, opening the door to an amazing array of applications.

One of the great features of this amazing device is programming flexibility. In order to make use of some advanced display features, you'll need to use their custom programming software (it's free), which is specifically designed to enable amazing graphics and leverage the hardware to its maximum potential. However, this device can still be programmed in the standard Arduino IDE, which makes it even more flexible with a familiar environment. The unit also integrates a MicroSD card slot for storage of images and icons too, so you don't have to worry about storage limitations either.



Now, anything you would normally task an ESP8266 to do, you can add a quality display for user feedback, input monitoring, and more. The display itself is a 80 x 160 pixel full-colour TFT display, and there's 6 x GPIO pins available too. Programming is also made familiar with the optional 4D-UPA programming adaptor, which provides USB communication, and power.

Weighing in at just 5g, these tiny but powerful devices are going to make their mark with makers. They're expected to be available this month and you can find all the details at: *http://www.4dsystems.com.au/product/IoD-09*

Shopping List:

IoD-09TH are available directly from 4D Systems, or you local 4D Systems distributor.

4dsystems.com.au

- **DISPLAY MODULE ONLY**
  ID-09TH **$19.95 USD**
- **DISPLAY MODULE WITH PROGRAMMING ADAPTOR**
  IoD-09TH + 4D-UPA + 4GB MicroSD **$39 USD**

VAPOUR CAPER

DAWID VERWEY

Technician and passionate PIC programmer.

http://UCSi.com.au

@UCSInnovation

When we learned that Dawid created a very simple vapour phase oven, for use with his non-contact soldering work, we couldn't wait to take a look.

Vapour phase technology is a powerful way to solder surface mount devices. By heating solder just above its melting point, there is a greatly reduced risk of component damage from excess heat, which you may get from soldering.

WARNING: Dawid's oven uses electricity and liquids at very high temperatures. It has no brain, so please use your own.

How did you learn about vapour phase ovens?

I first learned about vapour phase ovens (VPO) in 2010, when I came to Australia and started working at Fine-Tech. One of my colleagues, Peter, had made his own VPO based on a design he had seen on the internet a really long time ago. When I started my little business, he loaned his oven to me and I was immediately smitten. Previously, I'd used a little bake-oven for soldering PCBs at home, but it was really painful. Peter offered to help me make my own (he's got the coolest shed you've ever seen) and thus, my little oven was born. So this is basically version 2 of the oven; the only upgrade made from Peter's one, is the use of a more sturdy, slightly bigger chamber. His oven is made from a rectangular olive oil tin, so mine can do slightly bigger boards and would, hopefully, last longer. That said, his has been going for more than a decade, so longevity is clearly not a problem!

Yes, there's clearly no planned obsolescence there! Your oven seems very functional - how are you heating the Galden?

The heat comes from a cheap 500W halogen lamp. The glass cover was removed and the light is bolted facing upwards, to direct as much heat as possible at the base of the chamber. The Galden is only about 8mm deep, so it is very important to ensure that the chamber is sitting level to avoid hot spots. Note: If you do decide to build one of these, I suggest grabbing a handful of spare globes, as these are a dying breed and I don't know how long you'll still be able to get them.

That's about as simple as it gets! The Galden is expensive. Do you "use" much Galden in the process, or is it fully contained and re-used/re-cooled almost entirely?

I've been using my oven for about eight months now. I started with 300g of Galden and I've lost about 10g. My losses would probably have been less if I'd not been so slack and taken the Galden out after use.

The chamber of my oven doesn't seal perfectly, so the Galden does evaporate over time. The best thing to do is to wait until everything has cooled down, and then move the Galden to a sealed container; I actually use a syringe to suck up every last drop! Just don't do what I've done and leave it sitting in the chamber for weeks at a time!

Even losing a few grams in that period would hurt! Did it require any trial and error, or did it work first-go?

I did have to experiment a bit to find what works best for my oven. As the temperature is set by the Galden and the power is fixed, the only variable under user control is time. The size of the chamber, the mass of the boards in the oven, and the ambient temperature will determine how long it takes to bring the Galden to a boil. I use 230 degree celsius Galden, as it is suitable for use with both lead and lead-free solder paste. I just keep an eye on the board through the lid, and wait for the solder-paste to turn shiny. Then I give it another minute or so, just to be sure. Due to the nature of the process, you'll be hard-pressed to burn or damage a board unless you leave it in the oven for a very long time. Lately, I've been running a 10-minute cycle by default, and that works fine, regardless of the other factors.

10 minutes - that's less time than trying to solder by hand! Have you considered automating the PCB lowering/removal process, or is it too much complexity for such a simple process?

Considered – yes. Actually done something? It's on the to-do list! As you said, it's a very simple process; however, I'm thinking of automating it one day, just to make it a bit more convenient. Currently, the process is as follows:

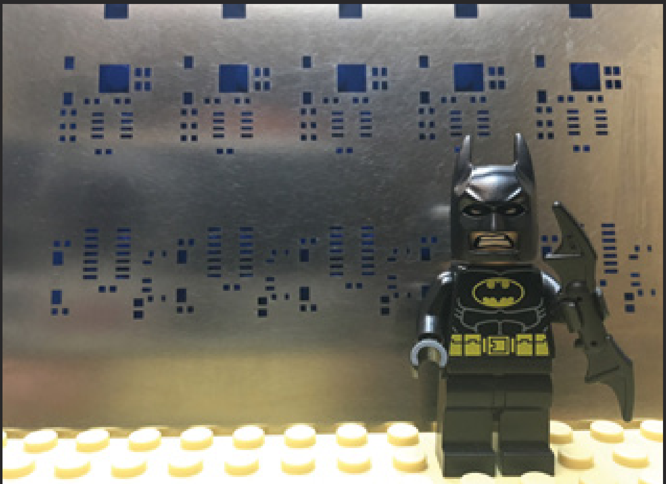
- Load the boards on the tray and lower them into the chamber.
- Turn on the lamp and set your phone timer for 10 minutes.
- After 10 minutes, turn off the lamp, raise the boards to the top of the chamber and wait two to three minutes for the Galden to condense and for the boards to cool down. Due to the low mass of the chamber, there is a huge temperature difference between the bottom and top of the chamber.

Automating this should be really easy, but if it ain't broke...

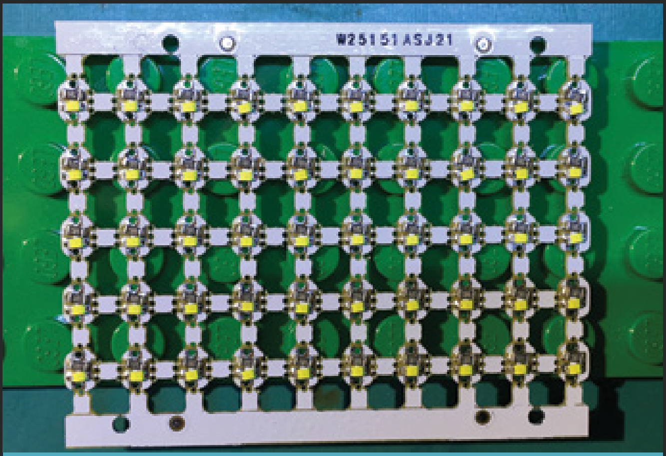
That's true... it could take many hours to finesse a basic automatic system, when it's not really necessary. Any tips for someone who would like to build their own?

Most of the parts I used, I picked up from my local Bunnings, so no great challenges there. The acrylic lid was laser cut by my mate, but you should come pretty close with a drill and some careful measuring. The hard part is getting hold of a small quantity of Galden – I need about 300g for my oven but I bought 500g, just to have a bit spare. The going rate for Galden is about $1500 for ››

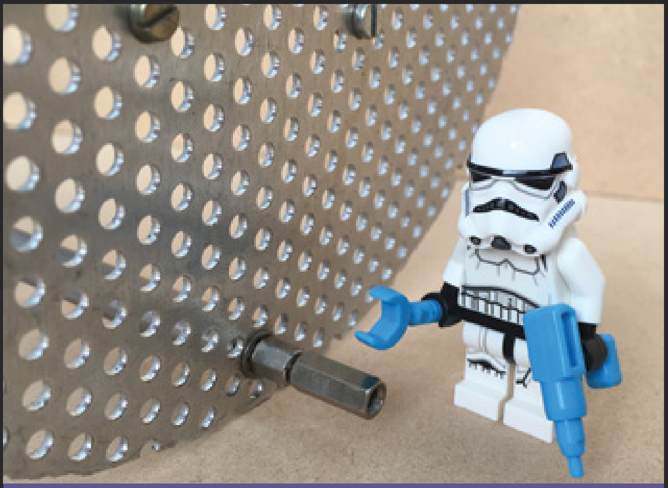
Dawid is used to vapour-phase technology as he uses it in his workplace at Fine-tech Electronic Solutions: www.finetech.com.au



As boring as it may seem, the printing of the paste onto a board during production is one of the most critical factors in determining the quality of your final product. Too much paste or bad alignment may result in short circuits; alternatively, not enough paste and you end up with dry joints. In the production environment, stencil design is a fine art. I ordered this small stainless steel stencil with my PCBs, and the PCB manufacturer set up the array for me and designed the stencil from my Gerbers. It didn't cost much and it's really convenient. Some PCB manufacturers even offer a service where you can have multiple PCB designs in one panel, so if you order a stencil to suit, you can cover multiple designs with the same stencil. Another option is to use a syringe to dispense paste onto the board, which I do on my little LED boards; but it does take time and practice to get it right.



The final product. You may see that the parts have drifted. This is due to poor PCB layout on my side. The molten solder tends to go towards the holes and pulls the components with it. Lucky for me, the boards still work fine, so I don't have to discard my PCBs. Some of us need to make the mistakes for others to learn from!

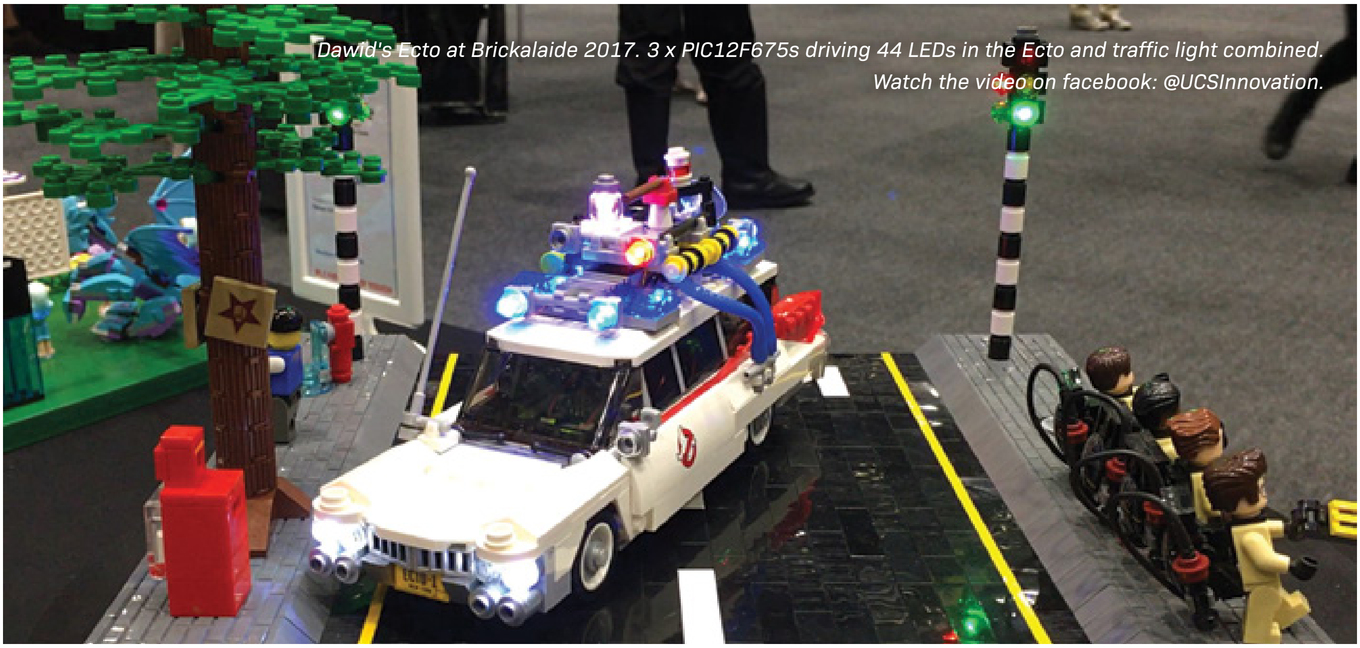


The PCB tray is suspended about 15mm above the bottom of the chamber and 5mm to 7mm above the surface of the Galden. I use the screws (in the tray) as supports when loading double-sided boards.

The heat source is a 500w halogen spotlight with it's glass cover removed.

The PCB tray slides on two linear bearings.

The complete oven on its base.


Dawid's Ecto at Brickalaide 2017. 3 x PIC12F675s driving 44 LEDs in the Ecto and traffic light combined. Watch the video on facebook: @UCSInnovation.

›› 5kg, and that's the minimum quantity you can buy. You can either persuade your friends to share the costs with you, or, if you have the means, buy a jar and sell the rest in small quantities online as a service to the community.

I'm sure there's someone doing just that! What's the coolest project you've used your VPO to solder?

I used my VPO to build the controllers and LED boards that went into my LEGO Ghostbusters Ecto for Brickalaide this year (Brickalaide is Adelaide's big LEGO show, held annually at Easter time). I watched YouTube clips to get the flashing sequences exactly correct; and 28 LEDs were driven from two 8-pin PIC12F675's. I'll have to work really hard to think of something cooler for next year's Lego show.

We'll be keeping an eye out to see what you come up with! Are there any circumstances when you think a traditional iron would be better suited than the VPO when using SMD? Or is the VPO really the way to go?

This little VPO uses exactly the same technology as what the big, industrial ovens use, so this is not a toy. It is ideal for volume work or for soldering fine pitch parts or parts with big ground pads. For Brickalaide I made 80 badge light boards. These have nine 0603 LEDs on the one side, and the PIC, battery holder, switches, caps and resistors on the other side. Yes, it took me a few nights and weekends to get them done, but it was still heaps faster than doing them one joint at a time; and the quality is superb. That said, I still needed my irons (two is way better than one – trust me!) to do rework on the odd tombstone. I have also found that the LEDs in particular, are very sensitive to overheating. None of them got damaged in the oven, but I did manage to melt a few with the irons, simply because they are so tiny and you can struggle to make a good thermal connection.

Sounds like it adds some valuable robustness to the process. If you were to start over and make a second one, would you change your approach using the knowledge you have now?

As I said, this is already version two. I do plan on adding some latches to the lid to get a better seal. For version three, I may look at automating it, but that's very far down on the list.

Awesome. Is there another cool project you're working on right now?

I'm preparing for the Adelaide Makerfaire in November, so the little oven will be running hard, doing production on the first batch of my new LED controller. It is 30mm x 20mm, runs off a single CR2032 and drives 6 LEDs. So far I have created flashing sequences for police cars and fire trucks, as well as a flickering flame effect. Each board comes with five different user-selectable programs, and I still have a few other ideas I want to try out; writing the software is my favourite part. Before the launch, I also need to design and build a display, write an instruction manual (my least favourite bit), and design the packaging. I plan on having my little oven and some of my production equipment at the Makerfaire for a bit of show and tell, for folks who are interested, so do pop in and say hi!

We're sure some of our readers will do just that! Thanks for talking us through your vapour phase oven, and thanks for the very entertaining pics! ■

WANT MORE?

For some info on Galden, or to discuss this feature, visit:
https://diyode.io/004sxhy

LEADING EDGE ELECTRONICS

Solutions for your everyday needs

Are you reminiscent of a time when you could buy all of your consumer electronics in one location? Or knowing and sharing a conversation with a store owner who was as passionate about your quirky specialities in "making things"?

Visiting a LEADING EDGE ELECTRONICS store in 2017 is still fun…where you can see, touch and discuss products in person.

LEADING EDGE ELECTRONICS stores are fully stocked and managed by experienced electronics professionals, some of whom have been trading for over 50 years.

These stores pride themselves in following the latest developments in electronics and technology and are dedicated to bolstering its range of products to ensure the retail group delivers the right items for the enthusiast and hobbyist.

LEADING EDGE ELECTRONICS has almost 100 strong, locally owned and operated resellers across Australia who sell a massive range of electronics. 30 of those stores are branded LEADING EDGE ELECTRONICS and carry products from major suppliers like Jaycar, Altronics, GME, Minelab, DJI, Uniden, Sony, WES Components, GoPro, Rasberry Pi, Traxxas, Hornby, Scalectrix, Sangean, Audio-Technica, Waeco, Garmin, TomTom, Swann, just to mention a few.

LEADING EDGE ELECTRONICS are passionate about bringing you the pieces to make your electronics and DIY projects possible and are always keen to hear from customers with suggestions for new products.

Visit us in-store for a huge range of:

ELECTRONICS • COMMUNICATIONS • COMPUTERS • HOME ENTERTAINMENT • SECURITY & SURVEILLANCE • POWER & SOLAR CABLES, LEADS & ADAPTORS • BATTERIES & CHARGERS • DRONES • CB RADIO & GPS • METAL DETECTING • WEATHER STATIONS • RC TOYS & HOBBIES • GIFTS & GADGETS

Here's one for all you electronic DIY'ers: Dreaming up your next big project? We would like you to post a pic and tell us about your current project for ...

YOUR CHANCE TO WIN 1 OF 10 $50 GIFT CARDS

to use at your local Leading Edge Electronics store

HERE'S HOW TO ENTER:

1. Scan the QR code on the right
2. Think of a creative way to take a photo of your project (include yourself, friends, family, pets, backdrops)
3. Post a pic of your DIY project with a comment on our Facebook page
4. Challenge your friends by tagging them in the post or share on your timeline

ENTRIES ARE NOW OPEN UNTIL NOVEMBER 17, 2017

Scan here to enter!

SCAN ME

https://goo.gl/hHs1GC

For full Terms and Conditions visit: www.leadingedgeelectronics.com.au

Interested in becoming a reseller?

Contact Support Office at recruitment@leadingedgegroup.com.au

NSW
Bathurst
Broken Hill
Burwood
Carlingford
Cessnock
Condobolin
Cooma
Cowora
Cowra
Deniliquin
Glen Innes
Griffith
Katoomba
Kyogle
Lavington
Lismore
Moree
Mudgee
Nowra
Oak Flats
Oberon
Parkes
Petersham
Singleton
Tea Gardens
Tempe
Tenterfield
Yass
Young

NT
Alice Springs
Tennant Creek

QLD
Biloela
Blackall
Bundaberg
Caloundra
Capalaba
Charters Towers
Gladstone
Gympie
Innisfail
Karumba
Kingaroy
Kirwan
Mareeba
Maryborough
Milton
Monto
Noosaville
Roma
Toowoomba
Woodford

SA
Balaklava
Barmera
Brighton
Cleve
Findon
Lameroo
Lock
Loxton
Mt Gambier
Naracoorte
Petersborough

TAS
Burnie
Miandetta

VIC
Ararat
Bairnsdale
Ballarat
Bendigo
Cobram
Dandenong
Echuca
Hamilton
Hoppers Crossing
Horsham
Laverton
Leongatha
Mansfield
Pakenham
Robinvale
Stawell
Sunbury
Swan Hill
Warragul
Wodonga
Yarrawonga

WA
Albany
Broome
Collie
Dowerin
Esperance
Geraldton
Kalgoorlie

The Autonomous Vehicle Conundrum

Many of us have this unnerving idea that machines can't drive better than us. But are we soon to be proven wrong? - by Rob Bell

When I was younger, I watched Back To The Future; and it's funny to think that back then we all believed it was somewhat realistic to have flying cars that were as stable and practical as the standard four-wheeled variety. Likewise, many of our favourite Sci-Fi movies featured spaceships that were held together with chewing gum and duct-tape as they hurtled through space. And they'd wrangle the steering of their million-tonne ship with the sweat and muscle of a World War fighter pilot, as if it makes a difference, providing sufficient distraction for us all to forget the obvious plot holes this represents! One thing is for sure though, while we're still struggling to take to the skies in daily life, humans are unlikely to seriously pilot these things once they exist.

These days, the semi-successful footage that we see of car-sized drones taking to the skies, are far from uneventful. These test flights made by cutting-edge technologies, usually see a very prototype-style contraption, with a rather nervous looking pilot strapped to the top of the device. They're making the smoothest adjustments they can to keep things under control, utilising 100% of their attention and focus, yet they still struggle to keep things from crashing to the ground.

Of course, it is fair to claim that this is a product of the technology being in its infancy, and it's to be expected while they figure out the systems and processes that will drive these things into the future; until they can reliably make them as simple to drive as a car.

But wait... driving a car is simple? Then why do over a million people all over the world, die on our roads, each and every year?

THE DANGER OF DISTRACTION

It would be fairly safe to say that close to 100% of the fatalities due to vehicles worldwide were caused by humans. Perhaps the driver misread the road conditions, or the vehicle's maintenance wasn't kept up (such as bald tyres) causing increased stopping distance. Or perhaps the driver's mobile phone rang. They know not to answer the phone, but it distracted them for long enough to take their eyes off the road. Or maybe they simply left the parking brake off and it rolled away on its own, taking out an unsuspecting pedestrian.

Every 25 seconds or so, somewhere in the world, someone dies in a motor vehicle accident.

It's already said that the youngest generation will likely never learn to drive. As a parent myself, I take solace in the fact that this is likely very true. The autonomy, convenience, and safety offered by a fleet of autonomous self-driving cars you can hail with an app on your phone, really is one of the best advances we can make, to ensuring our roads are less congested and safer for everyone in the future. It's entirely possible that in a few generations' time, we'll look back on this period of self-piloted vehicles on our roads, as one of the most wild and crazy times in transportation history.

Yet, as soon as a fatality happens involving a driverless car, the world goes into a tailspin slamming the technology as unreliable and unsafe. Of course, it makes for attention-grabbing headlines in the media, a chance for one company to advance ahead of another, but in reality it's an almost insignificant blip in the statistics of human fatalities involving vehicles. It also plays to our thinking that perhaps, somehow, we're not the bad driver that can cause these problems.

So why do many of us have a perception that a machine, specifically built, with world-class sensors and more driving experience programmed into it than a veteran taxi driver, is somehow less safe than a human driver? We can easily paint a picture of an 18-year-old, risk-taking, semi-distracted male driver, with a lead foot and an obsession with Facebook for comparison to see whether machine or human is more dangerous. But would that be fair? What about a 30-something mum taking her kids to school, obeying all the road rules and driving at five-below the speed limit? The reality is, regardless of our default personality and vigilance behind the wheel, we're all easily distracted sometimes - be it by the news on the radio, the kids arguing in the back seat, or daydreaming about the weekend while sitting in traffic on a Friday afternoon. Humans aren't designed to drive, we're designed to walk. So why not leave it to the machines?

THE HUMAN ELEMENT

One aspect that challenges our thinking, is who's making the decisions? As humans, we make all sorts of seemingly intelligent decisions, which stem from a mix of biology and education. Our education tells us to drive on the correct side of the road, while our biology would, if faced with the decision of hitting a person or a machine, create an almost involuntary and reactionary response, steering us away from the person and into a machine.

So could a programmer ultimately have a say in who lives and who dies? When the machine is left with no alternative but to crash into two people, how would it decide who lives and who dies? If it has to decide between crashing into a wall or crashing into another car, how does it decide? Could the car ultimately sacrifice you, the occupant, rather than crashing through a crowded pathway?

I think the reality here is that, forgoing some sort of conspiracy, it doesn't matter. Consider that first and foremost, a self-driving car's only task is to get you from A to B safely. Chances are it's not speeding, it's driving to the weather conditions, and it's not distracted by constant phone alerts or the conversation inside the car. The reality is, when faced with a challenging decision, the outcome should yield the best result, but by design it should rarely be faced with such decisions anyway.

NEVER-ENDING IMPROVEMENT

There are, of course, limitless scenarios we can conjure up regarding potential failure of the sensors: the computer could crash, or the system could somehow turn against us. It's easy when we need to try and justify our superior driving skills, but the reality is that machines are learning each and every day. Indeed machines can learn from each other's mistakes too, something humans are often reasonably poor at doing!

Personally, I can't wait for machines to take over the driving. As much as I love to get behind the wheel, I'd much prefer to spend the time doing something more interesting. A family road trip could involve playing games with the kids in the backseat until we arrive at our destination; to me that sounds much more fun than trying to distribute snacks while keeping a steady eye on the road.

When it comes to autonomous vehicles, I believe the future is bright, and personally I'll be there ready and waiting to sign up for the first autonomous option that will inevitably take to our roads soon. What about you? ■

GOT SOMETHING TO SAY?

To discuss this edition of Moonshots, visit: https://diyode.io/004xjvq

MADE THE THING?

Now Show it Off!

Each and every month we shine a spotlight on impressive projects from the electronics and maker community. We've already seen amazing projects like Thomas's Pi Powered Sprinkler and FRED the Tracked Robot from Royce in Issue 1, Liam's Toy Piano Conversion in issue 2, and Greg's Ultra-Accurate Voltage Reference in issue 3, just to mention a few.

Awesome ideas inspire others, so make all those late nights developing your project count even more. Not only will you get a buzz out of seeing your creation in print for the world to see, you'll know that you'll probably encourage other readers to get hands on to start or finish their own project.

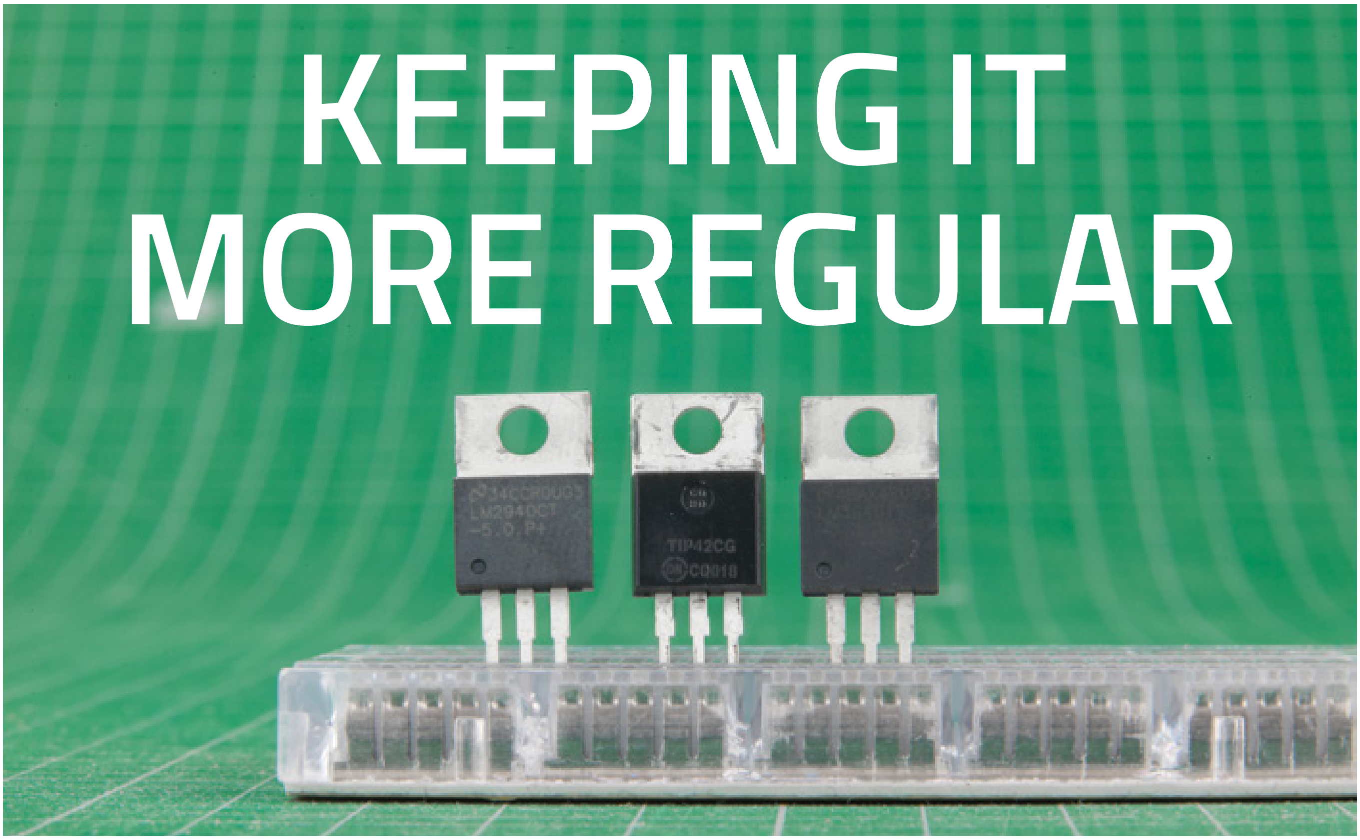
To show us what you've built, simply send us a few details along with some image of your project, and we'll get in touch if we have a place for it in a future issue.

Featured builds are also in the running for the $250 prize for Editor's Favourite - we'll pick one each issue!

DIYODE

SUBMIT YOUR PROJECT:
diyode.io/004submit

KEEPING IT MORE REGULAR



There's a range of other options with voltage regulators which are worth exploring. - by Daniel Koch

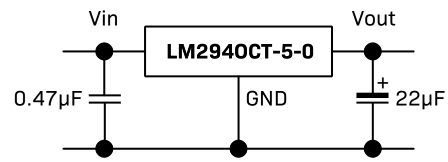
In last month's FUNdamentals, we presented several ways of deriving a fixed, clean, regulated voltage for your project when the power supply voltage is greater than your circuit requires; and spoke of some reasons why this may be the case. This month, we look at some different methods to regulate the voltage, as well as covering some devices that step the voltage up.

MORE ON VOLTAGE REGULATORS

Last month we covered the traditional, low-cost regulators that require the supply voltage to be at least 2.5V greater than the output voltage. However, this is not always practical. Take, for example, producing a 5V USB supply from four AA batteries, totalling 6V when new.

This would be impossible with a 7805; however, there is another three-pin, fixed regulator that can help us. The LM2940CT-5.0 is a member of a family of regulators referred to as "low-dropout" regulators, meaning that the difference between the input and the output voltage can get quite low before the devices "drop out". The LM2940CT-5.0 only requires an input 0.5V higher than the output, making them perfect for our scenario above **[1]**.

1



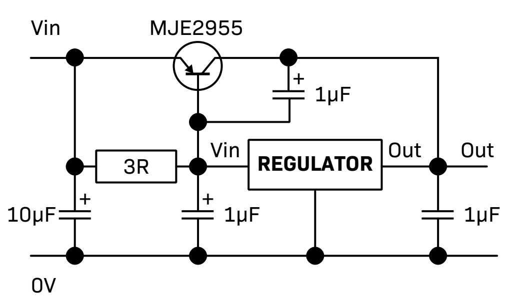
The LM2940CT-5.0 even has the same layout as the 78XX series, with input on the left, ground in the middle, and output on the right (when viewed from the front). The manufacturer's datasheet for the Texas Instruments product does recommend using both an input and an output capacitor, however. The family of regulators are also of interest to the maker because there is a 3.3V variant, the LM3940. The low dropout regulators are anywhere between three and ten times the price of the 78XX series though – depending on the item chosen, and the supplier – so they are not always ideal if their particular attributes are not essential. ››

MORE POWER!

One of the challenges of both the 78XX series, and the low-dropout regulators presented above, is that they only have a current rating of 1A (over which, they will self-limit the current.) This is not a big drama for a typical logic circuit, but is a challenge for anything driving light arrays, motors, actuators, and often much more. In such cases, there is a way to boost the current capacity of these devices. The manufacturer's datasheets from several suppliers list a similar circuit that has been commonly used for a long time. This involves the regulator being attached to the base of a power transistor, which handles the high current as the series pass element, while staying regulated by the regulator IC.

The circuit diagram **[2]** shows this (note: variations on this circuit exist and work fairly well). The choice of power transistor determines the current that can be passed and, in turn, this would rely on appropriate heat sinking.

2



One limitation of a circuit such as this comes from the fact that breadboards are not well suited to large component legs (even the TO220 package can stretch some breadboard springs and result in inadequate contact when used with fine resistor legs). Also, larger packages such as the TO3P don't fit at all. In addition, breadboards are not rated for very high currents, and 4A would be too much for many.

Fortunately, there is a solution. Both Altronics and Jaycar Electronics sell kits based on a low-voltage adjustable regulator, such as the LM317, which includes a circuit board, all required components, and instructions. These kits will, with added heat sinking, supply a voltage between 3V and 15V at a current of up to 4A. The kit has provision for fitting with screw terminals instead of the supplied PC stakes, so you could use fine wire to supply the logic on your breadboard and thicker wire to supply the high-current-driven load.

GOING UP!

All of this is well and good if you need to reduce a supply voltage, but what if you want to run your 5V device from a 3.7V lithium polymer battery? Sure you could put two in series for 7.4V and use a linear regulator, but this is inefficient on battery and far too close to the nominal value of 2.5V above output voltage for standard regulators (the 7.4V is nominal and is usually slightly higher when charged), leaving only the more expensive low-dropout types. In this case, the answer comes from a more recent switch mode power supply (SMPS), or switching regulator. The SMPS device uses a tiny coil to store energy when turned on and release that energy to increase the overall voltage out. SMPS devices switch on and off very fast as high speed allows them to use a smaller coil. SMPS can run between 20,000 and 50,000 cycles per second depending on the design. At high frequency, smaller copper windings are required, enabling tiny transformers to be used. SMPS can be designed to step-down the voltage (Buck mode) or step-up the voltage (Boost mode). Switchmode circuits can also be designed to increase or decrease voltage, Buck/Boost mode.

One such device from Jaycar is the XC4512 5V DC-to-DC converter. This takes a voltage from 2.5V to 5V, and steps it up to a regulated 5V DC. The current rating is stated in the datasheet as 500mA. This device is a small board, the size of many USB flash drives, and has a USB socket for the output, being aimed directly at microcontroller boards.

Another Jaycar product is the XC4609 DC-DC Boost module. This is a larger module with voltage display that can take an input from 3.5V to 35V and give an output between 5V and 56V, at a current of 2A. Altronics are also at the party, with the Z6334 DC-DC Converter Module being an adjustable switchmode supply, with input ranging from 3.5V to 35V and output from 5V to 25V. Also in the line-up is the Z6339 DFRobot Boost module, taking two AA batteries and yielding 5V, and the Z6337 DC-DC Boost/Buck module, which can be configured to increase or reduce voltage at currents up to 1A.

Of course, there are many other products available from online retailers; however, examples of these were not obtained. If in doubt, ask for datasheets, as these always contain the required information and more.

Hopefully this article has helped you understand that the supply voltage does not have to be limited to 5V from a USB port, and that supplying high power projects with a regulated supply rail does not need to be difficult or expensive. Likewise, lightweight, portable projects can be powered from single-cell battery supplies while still powering logic circuits. ■

NEXT MONTH:
RELAYS AND SOLENOIDS

We will return to a circuit building block that works in its own right but can be adapted or expanded. Remember, this is your space, so please keep the suggestions coming regarding what you want to read about.

GOT SOMETHING TO SAY?

To discuss this edition of Fundamentals, visit:
https://diyode.io/004gnzq

3D Print Server: Easy As Pi!

Make 3D printing easy and autonomous with this ultra small Raspberry Pi based print server. - by Oliver Higgins

3D printing is one of the most empowering elements in the maker space, and gives us the ability to rapidly develop our ideas. However, sometimes printing slows down the research and development process. So, we are going to use some basic hardware to reshape and redesign, in order to speed up this process.

THE BROAD OVERVIEW

Here at DIYODE Magazine, we use 3D printers a lot, as I am sure a lot of our readers and makers do too. In this situation, one thing that becomes apparent very quickly is the volume of resources consumed by computers and control units when running the printer. Some printers will allow you to run prints by SD card or inbuilt storage, but it can often be slower.

The next issue that we face is checking on the print. We must admit that since we have been using the Lulzbot printers we have been much less paranoid, but I think we all like to keep an eye on our print; just to make sure it's progressing as planned. But what do you do if the printer is in a different room or part of the workshop, or if you have a print that may take several hours?

THE SOLUTION

We have two issues, and we can solve them in many ways. The first issue is that of the required resources to control the printing. The actual printing process itself does not require any significant overhead; instead, it is the slicing process that will bog down a machine. However, you don't want to tie up a machine. You could easily shut down your printing software by accident, or overlay the machine with another process that you or someone else may be undertaking. Considering many printers can be run using an Arduino Mega, we can easily delegate this task to a Raspberry Pi. As a bonus, the Pi includes additional storage, network connectivity and of course Linux, which means we can run web server software.

Our second issue is that of needing to physically go and view the print. Placing a webcam near the printer and having a live stream can easily overcome this. There are many different webcams available, but in this instance, we will be using the Raspberry Pi Camera module.

PARTS REQUIRED:	JAYCAR	ALTRONICS
1 x Raspberry Pi 3	XC3630	Z6302B
1 x Raspberry Pi Camera Module	XC9020	Z6305
1 x 300mm (or longer) RPi Camera Cable	-	-

This assumes you have a working 3D printer too! You will also need an LCD screen and a keyboard in order to complete this setup on the Raspberry Pi. ››

›› THE CODE / SETUP / OTHER

Most workflows in this space go model, slice, GCODE then print. If you are using Cura and your printer is connected to the server or PC then the slice, GCODE, the print is done in one step. Both are then followed by walking and looking at the print, and then walking back a few minutes later.

To address the above problems, we plan to use the Raspberry Pi and set up a print server, which will allow us to set and forget. Of course, we are not the first to have these issues, and as such one group have already built a specific operating system for the RPi, and that is Octopi.

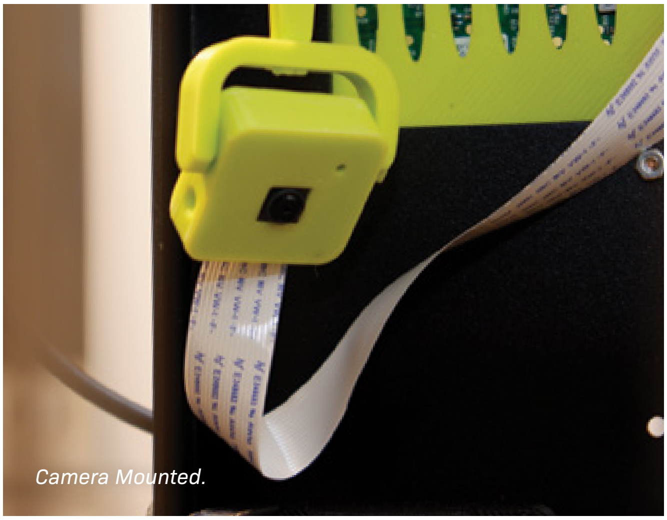
Here at the DIYODE workshop, we use the Lulzbot TAZ 6 and Mini. As such this article uses some resources that are specific to these printers; however, there are many cases for RPi use in Thingiverse that may exactly fit your printer.

Octopi is an open source RPi-based implementation of the Octoprint 3D print server environment. It offers full and remote monitoring of every aspect of 3D printing, and considerable extensibility well beyond this article through its plugin system. Best of all though, it's 100% open source, the degree of detail that it offers is unparalleled, and it is setup in just a matter of minutes.

Setting up is quite straight forward, although, if you have not installed or set up a RPi before we would recommend you explore and install NOOBS. To make the system run as efficiently as possible the operating system is installed without a desktop environment. You'll need to be comfortable executing some commands at the command line, and you'll also need to edit one text file.

Print the following from the digital resources (and yes, it is ironic that you have to print parts to make your life easier for printing).

Raspberry Pi 3 Case / Mount:	**RPi_case.stl**
Raspberry Pi Camera Case Mount:	**RPi_camera_case_stl**


Camera Mounted.

Download the latest version of Octopi: *https://octopi.octoprint.org/*

Prepare the SD and upload the Octopi image.

For Windows users: Win32 Disk Imager:
https://sourceforge.net/projects/win32diskimager/

For Mac user's we would recommend using Etcher:
https://etcher.io/

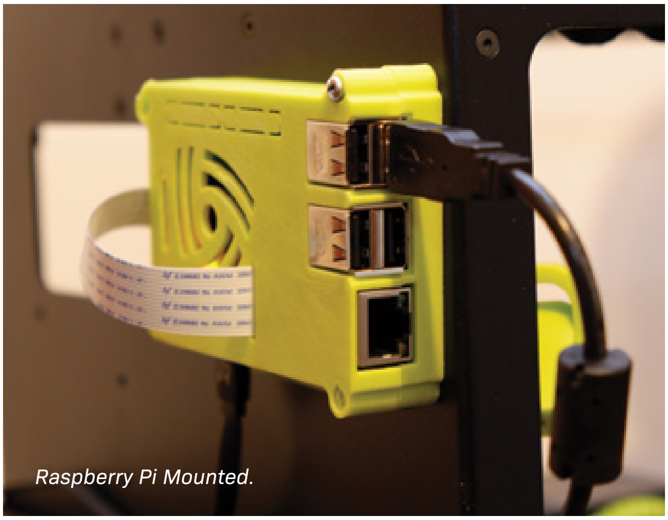
If you are using WiFi to connect, you will need to setup the SSID and passwords (skip this if you plan to use ethernet). Once you have imaged your SD card, navigate to the card in your operating system and locate the octopi-network.txt file located in the SD card's root directory.

```
##WPA/WPA2 secured
#iface wlan0 inte manual
#    wpa-ssid “put said here”
#    spa-psk  “put password here”
```

and change this to:

```
##WPA/WPA2 secured
iface wlan0 inte manual
    wpa-ssid “AWESOME_NETWORK”
    spa-psk  “1231456123456”
```

Note: make sure you are using text editing software, not a word processor or rich text editor. In Windows, we recommend Notepad++ or notepad. On Mac, use Atom or TextWrangler (TextEdit on Mac is known to put strange characters in for the “ character, and as such is not recommended in this case). Secondly, please note that by default most routers will have the 2.4GHz channel set


Raspberry Pi Mounted.

to auto, with 1-13 being available for use. There are documented issues with the RPi not working with 12 and 13, so if you have manually changed the channel on your router, you will need to make sure it is not 12 or 13.

BOOT IT.

Eject the SD card from your computer and insert into the RPi. Connect a keyboard and screen, connect the power supply and boot.

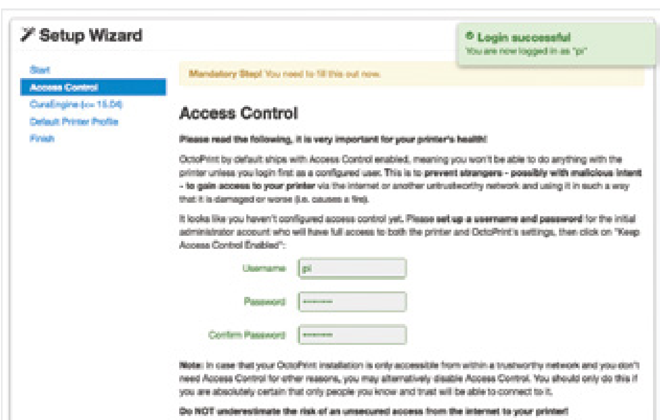
Once the system has booted, log into your RPI using the default User: pi, Password: raspberry. Run the following command:

```
pi@octopi:~ $ sudo raspi-config
```

If you are using the Raspberry Pi camera, select option 5 and the option P1, then enable. On a personal note, we also like to enable SSH here (option P2), to enable remote access via the terminal. When finished, select “Finish” then reboot. The RPi will reboot and will display the RPi.

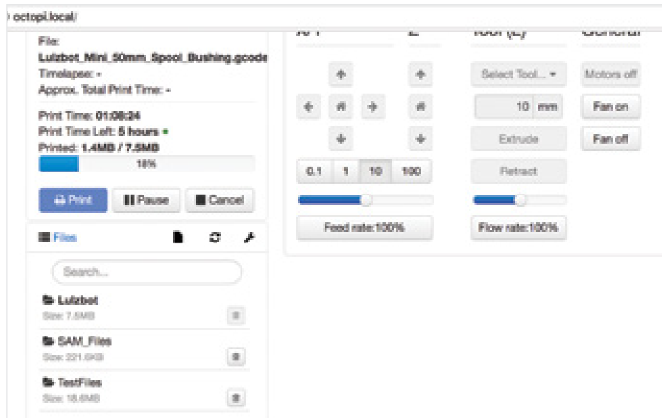
Connect to RPi via browser and connect to octopi.local or the IP address that is listed on boot.

Once you are connected to the RPi you will be greeted by the “Configure Access Control” page. Enter a username and password and then select the “Keep Access Control Enabled” button on the bottom right.



Next is the printer setup screen. If you are familiar with Cura, then the setup question will be familiar to you, as the Cura engine comes with octopi. If not you will need to know as a minimum:

- Name
- Model
- Bed shape (rectangular or circular)
- Origin point
- Volume (print bed dimensions)
- Is the bed heated?
- Nozzle diameter
- Number of extruders



Finally, elect “auto” to connect at login, and you have a wireless print server for your 3D printer! If you have not connected your RPi camera module or USB webcam, connect and you will then have a live stream on the control tab. The default RPi camera cable is quite short, so we would recommend getting a longer one (100, 200 and 300mm are available).

NEW WORKFLOW

This implementation has now become our permanent workflow and is currently being implemented on other machines in our workshop. Personally, we find that it speeds up the research and development phase considerably. For instance, we were developing a small part to work with an Arduino sensor. We designed a small bracket using our 3D software (TinkerCAD, Fusion 360, etc.). From here we exported our .stl file and opened it using Cura. With Cura we set up our print for slicing (or your slicing software). Once complete we export the GCODE file. If you have been using an SD card with your printer then nothing will have changed up to this point.

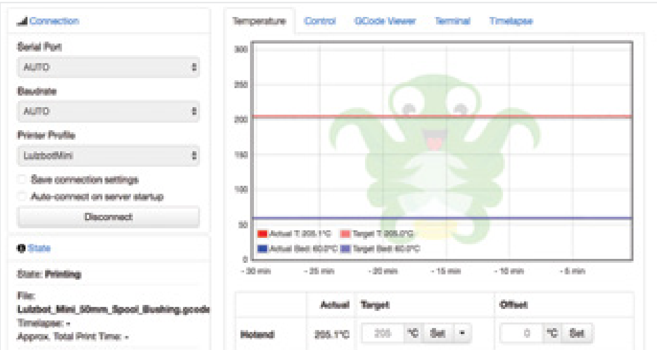
WORKING WITH OCTOPRINT

Once we have the GCODE file, you will need to navigate to *http://octopi.local* with your favourite browser. When you first go to the page you will be able to view the current status of the printer. To start the printer (assuming it is connected and turned on) you must log in using your previously set log in details, and connect to the printer (assuming you did not click select “auto print”).

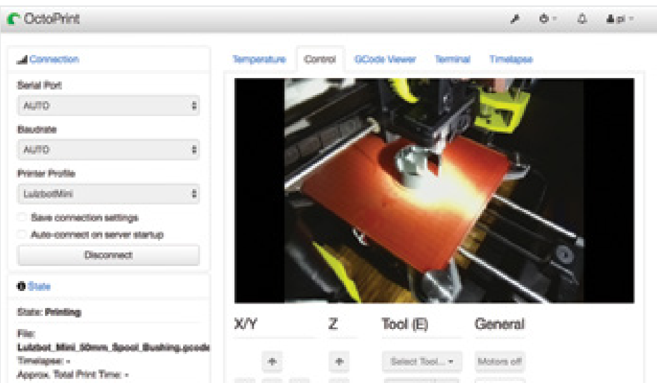
The upper left corner contains the printer control area. When logged in you will be able to connect to your printer. It usually takes a few moments to establish a connection and communicate with your ››

›› printer. While this happens, go to “files”, which is located at the bottom right of the screen. Click on “upload” and locate the GCODE file that you have generated. Once uploaded it will appear on your list.

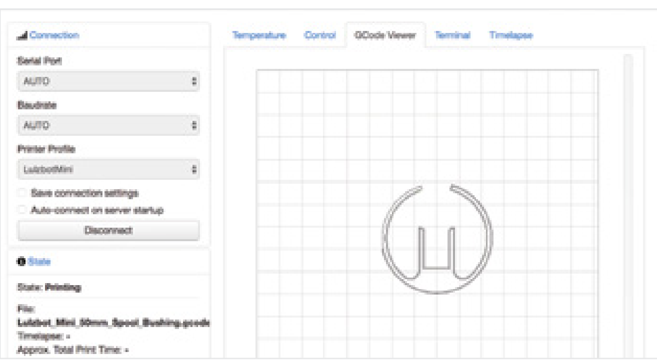
Click the “print” button and provided your printer is connected the system will start. Go to the temperature window and set your desired filament temperature.



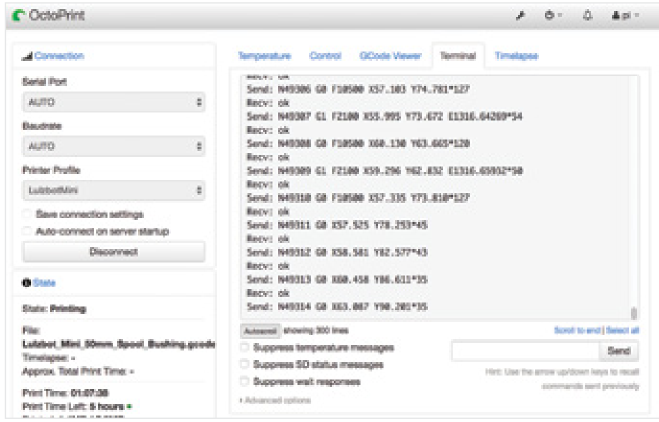
From here you are done, but that is not where the fun ends! If you have connected and enabled the RPi camera then click on the “control” tab and you will get a live stream of your print.



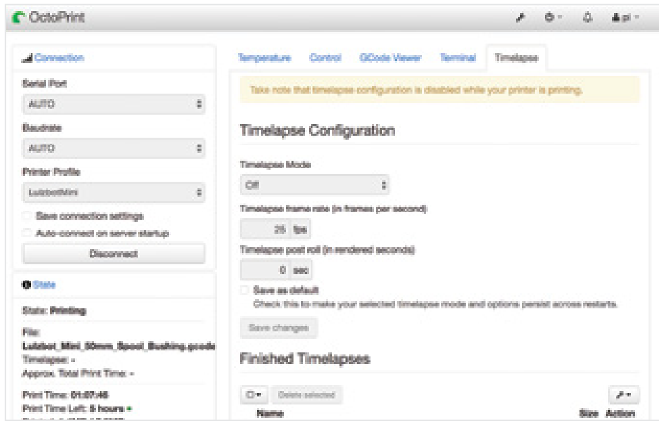
Clicking on the GCODE tab will show you the current GCODE output and a visual representation of where the print is.



The “terminal” tab will give the current GCODE code that is being executed.



Finally the “time-lapse” tab – now this is fun! You can set up the system to record your prints and create a video file, which can be used to access your print process or show it off on YouTube.



Finally, you can see the current status at any time, on the left side. This shows the current percentage printed, elapsed time and remaining time to finish. We can now design, print and view the entire process without ever leaving our desks, enabling us only to have to get up when the job is done. For some of us, that means saving precious time, every day!

WHERE TO FROM HERE?

This has just given a taster of what the Octopi system is capable of. Dive into the plugins, set up a live stream to the internet, install a desktop environment and connect with VNC. There is a lot more fun to be had! ■

WANT MORE?

For additional screenshots, 3D printer files and more, or to discuss this project, visit:

https://diyode.io/004jrkn

INTRODUCING Makeblock

AIRBLOCK MODULAR PROGRAMMABLE DRONE KIT

A 7-piece modular drone that can be turned into a hovercraft, car, spider and more! It is made of light but durable plastic foam so you can bump into walls without making dents. Control it from your Smartphone or iPad via Bluetooth using a freely available iOS or Android app. Control and program your aerial stunts through the Scratch 2.0 inspired mBlock app. Ages 8+.

• Watch the amazing videos on our website

KR-9220

Tablet not included

VIDEO

$299

NEW 2017

Bluetooth®

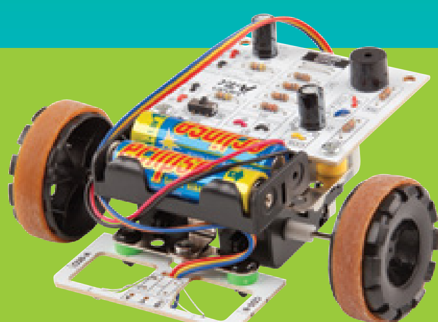
MBOT BLUETOOTH ROBOT KIT

An easy-to-assemble, entry level robot that can avoid obstacles, follow lines, play soccer, and more. Connect to a tablet via Bluetooth and start programming using the Scratch 2.0 inspired mBlock app. mBot can also be programmed with the Arduino IDE - it uses the same chip as the UNO. Ages 12+. **KR-9200**

ONLY $199

Bluetooth®

BUILD. CODE. PLAY.



$44.95

PC PROGRAMMABLE LINE TRACER KIT

Assemble the kit and you are able to use it in line tracing mode or program using the supplied software. Ages 12+. KJ-8906

Edison

$99.95

MEET EDISON ROBOT KIT

A pre-assembled robot, pre-programmed with 6 robot activities set by barcodes. Can be programmed using simple drag-and-drop programming blocks or a Python-like written language. Ages 5+. KR-9210

For our extensive STEM range, visit:

www.jaycar.com.au/stem

jaycar Electronics

jaycarelectronics @JaycarAU @jaycar_electronics

Prices shown in $AUD. Savings off original RRP. Valid to 31 Oct, 2017.

Miles has won $250 cash for this build!

DIYODE EDITOR'S FAVOURITE MAGAZINE

Sensory Development

MILES HITCHEN
Software engineer, electronics hobbyist, doting godfather and all round good egg!

Clever electronics provide a special needs boy with a sensory device to help him learn and grow. We were impressed.

Miles has a godson named Eiden who has a developmental delay, so he put his software development skills to fantastic use by building a sensory game. We were blown away by the creativity of the project, and for such a worthwhile cause; so we caught up with Miles to learn all about it.

Creating a sensory device for your godson sounds like an awesome project. Were you asked by his parents to assist, or did it come about some other way?

Sensory toys are very expensive and bulky, and many involve lights and sound but very few actually provide a game or require problem solving skills. Sasha - Eiden's mum - told me about visiting a sensory play area where Eiden had shown a lot of interest in a particular game. It was very large (roughly 5ft high x 3ft wide) and had a little bit of functionality. So I suggested I could probably make something of a more appropriate size, and add in lots of functionality, such as progressive levels, sound effects, and reward animations. Eiden has very specific visual and auditory issues; he is sensitive to certain sounds that lead to changes in his behaviour, so one requirement was to use sounds that excite and motivate him. Another was to ensure any buttons used did not have too much resistance or require too much effort to press, to cater to Eiden's hyperflexible joints.

Do you have any specific skills relating to special needs/developmental delays, and/or did you undertake any research?

Other than being involved in Eiden's play for six years, and having a good understanding of his individual preferences, I don't have any specific skills or training related to special needs. I worked with Sasha who helped provide me with some specifications, based on knowledge of his medical condition, research she had done into what's available on the market, and we identified changes in the game that could help enhance his developmental progress. The aim of this toy was for it to grow with his mental development.

How was the device received by Eiden?

Right from the start, he was very interested in it due to its large colourful buttons. He knows buttons can do exciting things when they're pressed so it captured his attention, even before it was switched on. It took him a couple of minutes to find a comfortable position in which to play with it, but once

▲ **ABOVE (INSET)**
Eiden using the sensory device.

The completed sensory device.

he did he loved it. He was giggling at the sounds the buttons made, and the reward animations, which he understood occur when he's solved the puzzle. Sasha took the lead in helping him initially, to play the game. We watched him evolve from just randomly pushing the buttons, to eventually making specific button presses based on what was happening in the game. This was proof that he was learning to understand the association between action and reward.

Working with Veroboard can be a challenge - did you encounter any?

Yes, there was definitely a challenge during circuit development. I was originally going to use perfboard (with individual solder pads) and just do point-to-point wiring, but after realising how many wires I'd need to solder I decided to go with stripboard, which I'd not used before. The main challenge was working out the layout of parts, and ensuring they could be connected either via the actual copper strips or by wiring on top of the board. I also had to make sure I considered any breaks that were required on the copper strips. I'd done a bit of reading and watched a few YouTube videos about working with stripboard, so that helped; and I'd also used the continuity test on my multimeter to ensure all the breaks I made were true. I used some stripboard PDF templates I found online, and that helped tremendously with my layout, I think I went through about five or six sheets before I got something that didn't have any errors on it. I did look at a couple of programs to help with layout, but I actually found it easier using the printed templates.

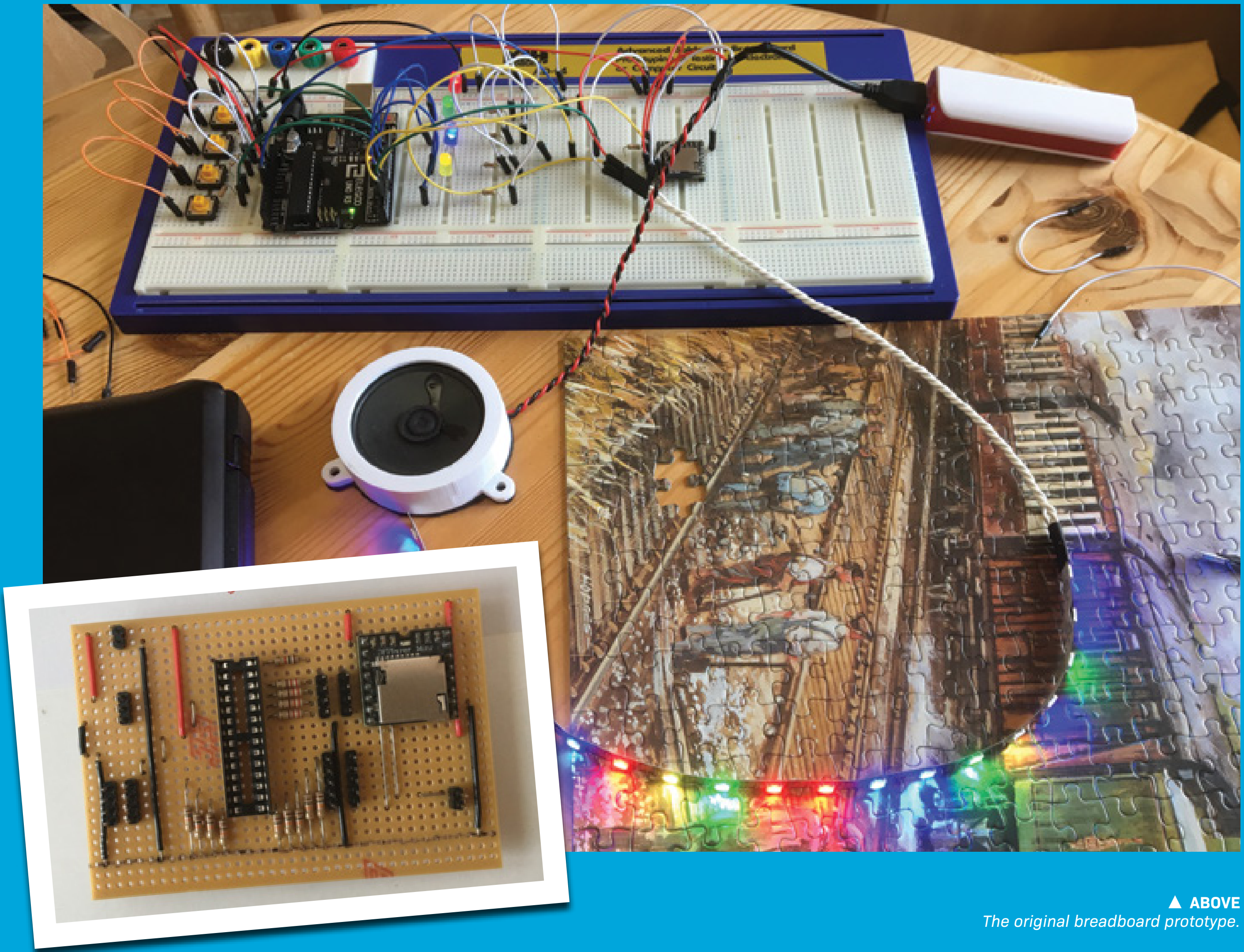
The templates can certainly help with Veroboard use. Did you prototype using an Arduino UNO before moving to a standalone 328p?

Yes, it's a great prototyping platform and I love using it. I did think about just using the Arduino, but wanted the challenge of making it standalone. In the end it wasn't much of a challenge, as it's well documented and easy to follow on the Arduino website.

It sure does make prototyping easier. Can you explain how the game itself functions (i.e. input versus output)?

The game has a 4x5 matrix of LEDs that are individually boxed and covered with opaque plastic to diffuse the light. Each LED can be one of four colours: red, green, blue or yellow. Under the LED matrix are four large backlit buttons, with the same colours. The game has five selectable levels and progresses automatically as each level is completed.

The basic idea is that at the start of the game, some or all of the LEDs will be lit and depending on which colours are displayed, the appropriate large button will also be lit. As the player pushes one of the lit buttons, a distinct sound effect (associated with that button) is played and an LED of that same colour is switched off in the matrix. When there are no more LEDs of a specific colour, then the correspondingly coloured button's backlight is switched off, and that ››



▲ **ABOVE**
The original breadboard prototype.

◀ **LEFT (INSET)**
The veroboard evolution of the prototype, which also uses a DIP ATMEGA328P

›› button no longer has any effect. When all the LEDs in the matrix have been switched off a short animation is displayed, which uses the matrix of LEDs and a longer sound effect. The animation played is randomly selected from four that have been programmed. The matrix is then relit, and the player's level is increased.

So, for Level 1 there are five random LEDs in the matrix and they are lit in a single colour (randomly chosen from red, green, blue or yellow), and the large button of the same colour is also backlit.

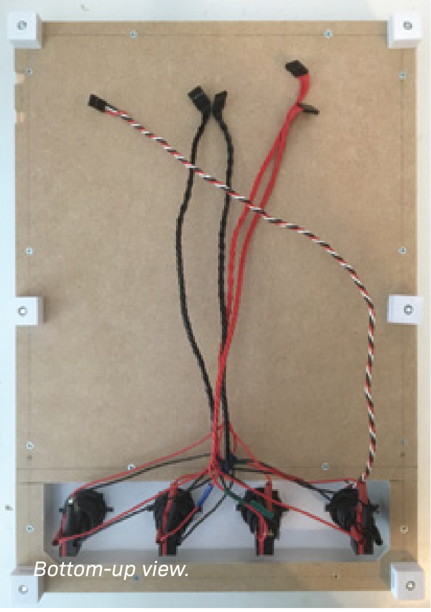
For Level 2, 10 LEDs are lit in two randomly chosen colours (five LEDs in each colour), and the associated buttons are backlit.

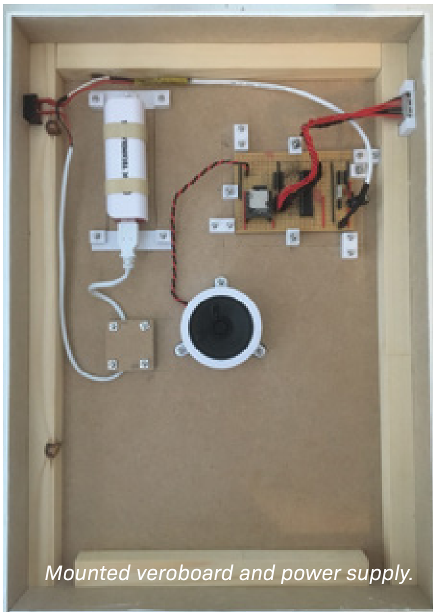
For Levels 3 and 4, the number of LEDs is increased to 15 and 20 respectively, and the number of colours used also increases to 3 and 4.

Level 5 is slightly different as all 20 LEDs are lit in randomly selected numbers and colours (e.g. there could be 3 red LEDs, 6 blue LEDs, 10 green LEDs and 1 yellow LEDs; or there could be 14 yellow LEDs and 6 red LEDs).

"Eiden has very specific visual and auditory issues; he is sensitive to certain sounds that lead to changes in his behaviour...."


Planning out the construction.


Bottom-up view.


Mounted veroboard and power supply.

That's awesome. We've uploaded the video you provided to the online article too, so everyone can see exactly how this works. Has this inspired any other projects that provide a similar benefits or outcomes for someone with developmental challenges?

I've actually, literally, just finished this project, so I don't think it has had chance to inspire anyone to build anything else yet. But there has been a lot of interest on special needs forums and Facebook, with over 400 likes already! I've also had many requests for the toy by other parents of special needs children, and the special needs school that Eiden attends also expressed an interest in using it.

We're sure every parent with a child who has special needs would be keen on something that may assist with their development and provide some fun along the way! Were there any unexpected challenges developing this project?

The first one I came across was how to wire up the five-way slide switch. As I'd not used one before, I initially thought that opposing pairs of pins would be connected as the switch was moved; but it was slightly more complicated than that, and so ended up taking a while to determine the correct way to the wire them. I knew designing an accessible case (i.e. accessible to me so that I could access the circuitboard and wiring) would be a challenge, and although I've made the circuitboard accessible, it's still quite tedious. As much thought needs to be put into the case design, as the original circuit layout and programming. Luckily, I have a 3D printer so that was put to good use making various fixings. I also used it to make the boxes surrounding each of the LEDs. I work as a software engineer, but I did find the programming of the game quite challenging - though it was a challenge that I enjoyed!

There's nothing quite like a challenge with a positive outcome! If you started it again from scratch, is there anything you would change for the new version?

There are quite a few things I would change; mostly around the design of the case. I'd also change the circuit slightly, as I'm not currently limiting the current on the input pins of the processor. I could also reduce the number of connections to ground for the level switch. And I'd change the way I've connected the power switch up to the circuit board - just to simplify things. I'd also change the way I've wired up the big button switches as the current method is complicated and makes swapping any of them out, almost impossible.

It sounds like it's been a huge success the way it is though. What other amazing creations might you be working on next?

Eiden's school use object and picture exchange communication, so to make something that could help reinforce this or even replace it, is something I'm currently thinking about.

That's fantastic, Miles. This project is such an exceptional use of your skills. Thanks for taking us through your project. ■

WANT MORE?

For video and more images, or to discuss this feature, visit:
https://diyode.io/004mdyq

Reading & Resources:

▶ **ARDUINO** *www.arduino.cc*

FUNCTIONS AND MULTI-DIMENSIONAL ARRAYS

Repetitive tasks can be made more efficient, and you can handle data in a convenient way. - by Oliver Higgins

In the world of programming, functions are one of the most valuable tools you can use. In our ultimate quest to do as much as we can with as little code as possible, the function is probably only second to the loop, for the sheer number of lines it can save.

WHAT IS IT?

When writing code, we will often be faced with repetitive tasks, and while loops can serve a purpose, they are not always the right tool for the job. The function is another tool in the arsenal; it is a simple but elegant way to do a single thing. In best practice, you should code your functions to do a single thing and do it well.

Functions provide us with a way of putting a repeated section of code in a single place. We can then "call" it from any point in our program. It serves multiple purposes: it provides a single point to make code changes to the related code, it promotes reusability of your code, and will significantly speed up development time.

THE ANATOMY OF THE FUNCTION

No matter whether a function returns a value or not, the structure is very similar. The following is an example of pseudo code for a basic function that does not return a value.

PSEUDO CODE:

```
void myFunction()
   do something;
```

ARDUINO:

```
void myFunction(){
   do something;
{
```

PYTHON:

```
def myFunction:
   do something
```

We would use this in your programs where you have a single defined output that needs to be done. It could be as simple as sending an error message to the user, or it may be to move a servo a given distance. For example, the simplest implementation for this would be based on the trusty "Hello World" example. Our typical "Hello World" includes the minimum code (about one line) required to start the program and output a message. We will modify this to create the function that prints "Hello World" for us.

HOW TO USE IT

ARDUINO/C:

```
void printHelloWorld(){
   Serial.println("hello, world!");
}
void setup() {
   Serial.begin(9600);
   printHelloWorld();
}
void loop() {
   //not used
}
```

PYTHON 3:

```
def printHelloWorld():
   print("Hello World")
   return
printHelloWorld()
```

You can see in this example that we put the function in the code before we call it. Now most modern languages will work no matter where you put it, but we have had experience with code that compiles fine, yet the execution is erratic simply because functions were out of sequence.

PASSING A VALUE OR VALUES

This is good for calling a simple function, but what if you frequently need to calculate something complex? We can send variables to the function via parameters, and we can get the function to output a variable of the final result.

Functions can only ever return one thing (I want to say only one variable here, but arguably you could return an object or even a dataset. The important thing to understand is that you can only ever return one of them).

Here is an example of a function that will take a single parameter; in this case it's an "int":

ARDUINO:

```
void myFunction(int x){
   Serial.println(x);
}
```

PYTHON 3:

```
def myFunction(x):
   print(x)
   return
```

MULTIPLE PARAMETERS

ARDUINO:

```
void myFunction(int x, int y, int z){
   Serial.println(x+y+z);
}
```

PYTHON 3:

```
def myFunction(x,y,z):
   print(x + y + z)
   return
```

RETURNING A VALUE

Functions are great at doing simple repetitive code and having it in a single location, but one of the most powerful elements of functions is to return a value to the user. As I've said, functions can accept multiple inputs but can only ever return one item. In very simple terms we can input several integers and return the result of our math as a single integer.

To instruct the function that will be returning something, we need to replace the void at the start of the function with the type of variable we will be returning. In the following example we are going to return an integer by specifying "int" at the beginning.

ARDUINO: ››

```
int myFunction(int x){
   return x+1;
}
```

›› **PYTHON:**

```
def myFunction(x):
   return x+1
```

WORKING EXAMPLE

This is a working example of a function that takes a single argument and then returns a value. In this case, we need to regularly calculate the circumference of a circle. The function takes one argument, an integer “d”, which is the diameter. The function then calculates and returns a float, calculating the circumference by multiplying d (diameter) with pi to 100 decimal places. In the case of the python example, the variable types are defined by the compiler.

ARDUINO:

```
float myPiFunction(int d){
   return d*3.14159265358979323846264338327950
   28841971693993751058209749445923078164062862
   08998628034825342117067 9;
}
void setup() {
   Serial.begin(9600);
}
void loop() {
   Serial.println(myPiFunction(3));
   delay(2000);
}
```

PYTHON 3:

```
def myPiFunction(d):
   return d*3.14159265358979323846264338327950
   28841971693993751058209749445923078164062862
   089986280348253421170679
print(myPiFunction(3))
```

NOTES ON FUNCTIONS

Most languages will allow you to place functions at any point outside the main script or loop. However, it is best practice, and I would strongly encourage you to place your functions above and before you use it, in a final sense. While you will not get a compile error, we have seen this be the cause for some strange issues, especially on the Arduino platform.

NAMING CONVENTIONS

Function names should start with verbs, such as “setTrap” and “getDeadVermin”, if the function changes the state of the program; and nouns if they’re used to return a certain value.

MULTI-DIMENSIONAL ARRAYS

In Issue 3 we discussed the role that arrays play in the world of programming. We can extend the array functionality to create a richer and more dynamic structure. We do this by creating a multi-dimensional array.

While an array has 1 dimension, an object with 2 or more dimensions is called a matrix. For the rest of this article we'll work with 2 dimensions only.

Please note: in this article, we will focus on ANSI-C style syntax, because Python 3 uses a different interpretation of its lists, and does not directly correlate with the general interpretation and syntax.

The declaration is similar to a single dimension array; we simply add a second square bracket with the number of elements required in the second dimension. The simplest way of thinking about the multi-dimensional array (i.e. matrix) is that it is an array of arrays; although, it may be better to think of them as a matrix or grid. When we declare an array, we give it the total number of elements in both the X and Y dimensions.

To declare a multi-dimensional array we using the following code, which has three rows with each row having four columns.

```
int myArray[3][4]
```

HOW DO WE PUT IN DATA?

You can directly access the by:

```
myArray[2][1]=5;
```

or

```
int myArray[3][4] = {
   {0, 1, 2, 3} ,   //row indexed by 0
   {4, 5, 6, 7} ,   //row indexed by 1
   {8, 9, 10, 11}   //row indexed by 2
};
```

or

```
int myArray[3][4] =
                  {0,1,2,3,4,5,6,7,8,9,10,11};
```

To access the information in a give variable we address the array like so:

```
myArray[2][1]
```

This is similar to creating a table in Word or using a spreadsheet. By addressing the X and Y coordinates, we can pragmatically and systemically traverse (that is, work our way through) the array and pull out the required information.

	COL 0	COL 1	COL 2	COL 3
ROW 0	[0][0]	[0][1]	[0][2]	[0][3]
ROW 1	[1][0]	[1][1]	[1][2]	[1][3]
ROW 2	[2][0]	[2][1]	[2][2]	[2][3]

A word of warning: there are two ways in which you can use this type of array; one is similar to a database or record set, where each row contains a subset of data, and each column is an element. To access the array in this manner, you use a nested loop conjugation, consisting of two loops. The first loop traverses the rows, the second traverses the columns. This is shown below.

```
void setup() {
   Serial.begin(9600);
}
void loop() {
   int a[5][2] = { {0,0}, {1,2}, {2,4},
                                     {3,6},{4,8}};
   int i, j;
   for ( i = 0; i < 5; i++ ) {
      for ( j = 0; j < 2; j++ ) {
         Serial.print("a["+String(i)+"]"+"
                               ["+String(j)+"] ");
         Serial.println(a[i][j]);
      }
   }
   delay(10000);
}
```

The alternative is to use the multi-dimensional array as a grid. The best example of this is the battleships game. If we were to implement a version of this game, we would need to create four matrices; one for each player's ships and one for each player's hits/misses. Each player would align their ships to the grid creating a "map" of the locations. Each time the other player "fires" a shot (e.g. "B7"), we need to take the reference and see if there is a ship located at that location. If there is then we mark the alternate matrix as a hit, otherwise we mark it as a miss. The game could include error checking to indicate that shots may have already been fired in a given location or supply alternative feedback.

In this context we do not access the information in a linear looped way; instead, we are addressing the individual elements directly.

```
myArray[2][1]
```

We need to clearly understand the purpose of the multi-dimensional array before we access it in our code. In this context, the multi-dimensional array is two dimensions: X and Y. Depending on your implementation you could make it three or more dimensions; however, better coding solutions exist for things such as this. The battleships game could easily work as a 3D array, with the 3D "Z" dimension allowing each of the players' boards to be allocated in this dimension.

CODE WITH INTENT

This might sound like a motivational speech, but it's not. What we mean here is that when you're designing your functions, give them a single purpose. Don't make several things happen or multiple variables change with each function. While this may sound trivial, it becomes a serious headache in large multi-file programs or object-orientated environments.

Functions can call functions, and those functions can call functions, so do not be afraid to break your code into atomic structures (an atomic structure is one that does a single job).

This simple change to your coding habits will make code supervision and bug tracking much faster and simpler. ■

NEXT MONTH:
MODULES AND LIBRARIES

GOT SOMETHING TO SAY?

To discuss this edition of Secret Code, visit:
https://diyode.io/004ymhw

Emergency Mobile Phone Charger

What started as a quest to debunk a "lemons charging phones" video, turned into something far more practical. - by Rob Bell & Daniel Koch

When life gives you lemons, you make a... phone charger? WAIT - that's not how that goes..

These days, mobile phone charging is rarely a problem when we have dozens of USB chargers, computers with USB ports, and various other ways to access power. But when the weather turns wild and the mains power shuts off, there's often a mad scramble for the nearest working chargers, to get the phones up and running again.

Recently we've seen a number of videos making their way around the internet regarding the theory of charging a phone using lemons. Some articles even landed on respected news sites, so they're convincing enough for sure. However, to us electrically-minded folk, at face-value it's difficult to believe this is possible; though it does beg the question - just how much "juice" can we get out of a lemon?

The theory here is fairly straight forward, but still warranted some investigation. The great thing about this exploration is that it yields some experiments that are easily undertaken at home, which are suitable for just about any age. We'll admit too, we got a little ahead of ourselves; on the surface, using these lemons like batteries we should be able to stack a few together to get something useful.

It's just feasible enough to be believable, right?
Well... let's take a look.

Note: If you want to skip to the end-project, where you'll find a useful emergency phone charger, we understand - just flip to page 44.

THE BROAD OVERVIEW

In reality, getting power from a piece of fruit isn't all that difficult. The old potato clock experiment continues to be a staple in electronics theory and experimentation. However, just because you can run a small LCD from a single potato for a substantial amount of time, doesn't mean you can power substantial electronics.

PARTS REQUIRED:

Lemons (we used up to 10)
Zinc Galvanised Nails
Copper Nails or Rod
Alligator Leads
Multimeter

PREPARATION

It can be difficult to find copper nails these days. Copper corrodes easily, and is relatively expensive, so mild steel is generally preferred for common nail use. However, it's quite easy to find copper rod. It might be in the form of copper welding rod, or even copper plated steel. We have used copper-plated welding rod. If you're using something similar, cut the rod into lengths of approximately 10cm long. The overall length doesn't make a whole lot of difference, providing they can be pushed all the way into the lemons, and leave room to attach your alligator leads.

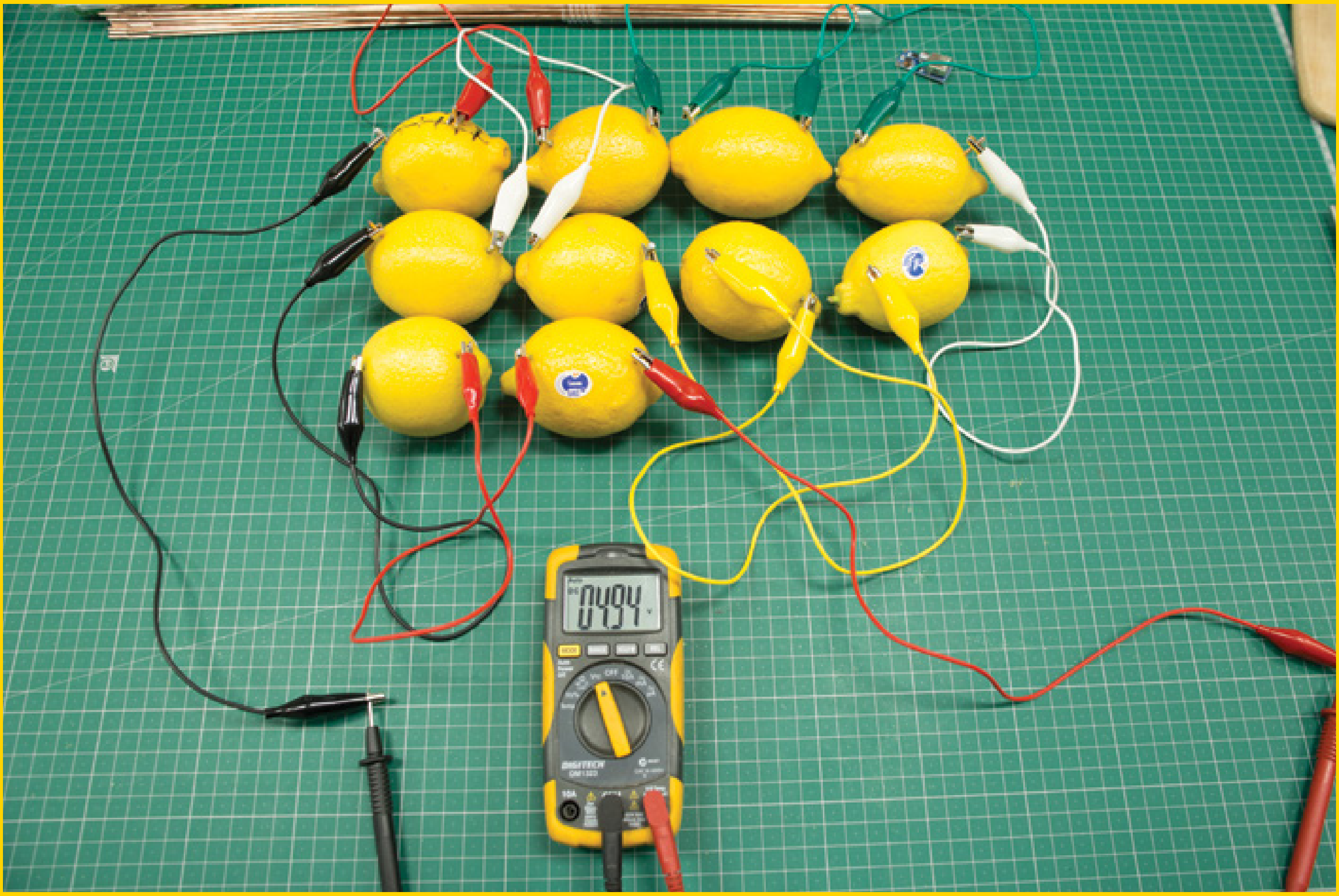
EXPERIMENT 1. 1 LEMON

Using a single lemon, insert one of the copper electrodes, and one of the zinc-plated nails. Use your multimeter to test the voltage and short circuit current of your lemon battery.

Our tests indicated between 0.5V and 1V. From a single lemon, we managed to produce a current of just 0.17mA. These figures vary with the lemon, the quality of the electrodes, and other such factors, but really the difference is insignificant.

Consider a "standard" minimum power USB outlet, which is capped at 500mA of current. At 5V, that's a mere 2.5W of power. Not really a lot, but our lemon is producing just 0.000085W of power.

When you put that in perspective it means our USB port produces around 30,000 times the power of a single lemon; and we're not even talking about newer, high current USB ports! ››



EXPERIMENT 2. 10 LEMONS

Now we'll use series wiring to try and boost voltage. This is really no different to wiring batteries in series.

When you test the voltage of the bank, you should see approximately 10 times the voltage you measured in Experiment 1. For our lemons, this meant just under 5V. The problem is, there is barely 0.5mA of current available. No, not 0.5A, 0.5mA... that's about 1% of the current available on an old-school USB port (or 0.2% of a higher current USB port). This presents a massive problem; we simply don't have the power available. The potential simply isn't there.

With 10 lemons we can illuminate a single, standard 5mm LED. That's hardly mind-blowing, working out to around 0.00085W of power.

ARE THERE ANY VARIABLES?

As with all electronics and experiments, there are variables.

ELECTRODES: As with a battery, the surface area of the electrodes does make a difference. We saw a marked improvement in current available when using our oversized electrodes, with an increase in voltage per lemon, as well as an improvement in short circuit current.

ROLL-FIRST: Lemons aren't just a bag of juice. They have internal membranes between each segment. Rolling them with a firm pressure helps disrupt these internal membranes and provides better consistency in the resulting voltage and current output.

TEMPERATURE: Temperature plays a role in all electrical circuits - even fruit-based ones.

SEASON: The season and ripeness of a fruit will change the electrolyte potency, and therefore the power available.

However... none of these variables are going to provide enough of a change to give us usable power to charge a battery with.

IT'S JUST NOT THERE...

We could mess about with different configurations, track down better electrodes to improve performance, but ultimately this solution really just doesn't provide enough electrical power to be useful for phone charging. Based on the results of this experiment, it would take thousands of lemons to create 500mA at 5V, which is around the power you would need to see sensible charging rates for your phone. The videos reportedly achieving this using only a handful of lemons are obviously fake; and that's all there is to it.

CONCLUSION - CHARGING SMARTPHONES FROM LEMONS IS CRAZY!

What you'll discover through this process is, while it's true you can quite easily obtain the 5V required to theoretically provide power, there is a serious lack of current available. Despite the relatively convincing videos doing the rounds on social media, it's fairly safe to say they are fake. They often, conveniently, don't show the entirety of the cables being used in the circuits.

In order to get enough power to charge a phone, you would, need a truckload (literally) of lemons. Around 1000 lemons in a series or parallel combination would be required to produce the most basic of charging currents.

So no, lemons are not a feasible way to charge your phone.

HOWEVER; is there a simple way we can build an emergency phone charger? You bet!

A USEFUL EMERGENCY PHONE CHARGER

Despite our experiment, we aren't about to let you down in terms of being able to charge your phone when little-to-no power is available! An emergency power source for your mobile phone might just save your life, or - more likely - at least make life a little easier when the mains power goes out. Of course, there are plenty of rechargeable power banks that can be used for this purpose, but most use some sort of lithium cell which you'll need to leave recharging, or else it will probably be flat when you need it. With the long shelf-life of lithium batteries (indeed some are sold as "guaranteed long life"), we have a sensible source of power that's shelf-stable, and provides enough power to get us out of trouble in an emergency.

CHALLENGES OF PHONE CHARGING

When first considering this purpose, a DC-DC boost module seems like a simple solution. It will take a 3V source from two batteries, and gives you 5V output right? Technically, that's correct, and you can power an Arduino project from it with no problems, for instance. However, Apple - in their infinite wisdom - decided that applying power wasn't good enough; so charging a phone purely from 5V isn't possible. You connect the power, and the phone ignores you.

As mobile phones evolved, lithium batteries grew in capacity. The old standard of 500mA charge-current quickly wasn't considered fast enough, and charge-rates quickly escalated. However, that meant the charge controllers within the devices needed to become a little more intelligent. While some phone manufacturers don't make it too difficult, Apple notoriously had specific requirements on the data pins of the USB socket in order for the phone to charge. For this reason, a number of USB outlets that are still available on the market won't actually charge your phone. So frustrating!

Fortunately, the secrets behind why this occurred became not-so-secret in a relatively short space of time, so the information is easy to find. Effectively a voltage-divider circuit is required on the data pins. The voltage divider tells the phone what power is available. While newer iPhones might be able to charge at a faster rate, our DC-DC boost module is only rated at 500mA, so we can appropriately configure the data pins accordingly. Regardless of how fast your phone can charge, this is designed to run at 500mA, which will still get any phone switched on and charging in a short space of time.

NOTE: We have tested this on various iOS devices and it works well. We also connected a few Samsung devices and it appeared to work fine with them too. You might find however, that your particular device is picky about these things; 500mA configuration appears to be relatively universal however, and should suit an overwhelming majority of devices. ››

›› THE BUILD

PARTS REQUIRED:	JAYCAR	ALTRONICS
1 x 5VDC Boost Module	XC4512	Z6366
1 x 2xAA Battery Holder	PH9202	S5025A
2 x AA Lithium Batteries	SB2355	S4906
1 x 75kΩ 0.5W Resistors	RR0617	R7603
2 x 51kΩ 0.5W Resistor	RR0613	R7599
1 x 43kΩ 0.5W Resistor	RR0611	R7597
1 x Slide Switch	ST0300	S2010
2 x M2 Screws	HP0390	H3100
1 x PCB Mount USB A Socket (optional)	PS0916	P1300
1 x 3D Printable Case (optional)	-	-

BUILDING A PROTOTYPE

Before we go randomly soldering up to our DC-DC converter module, it's prudent to prototype the data pin resistors just to check things are working as expected.

Using a breadboard, simply follow the diagram. It's quite straight forward and will only take a minute or two.

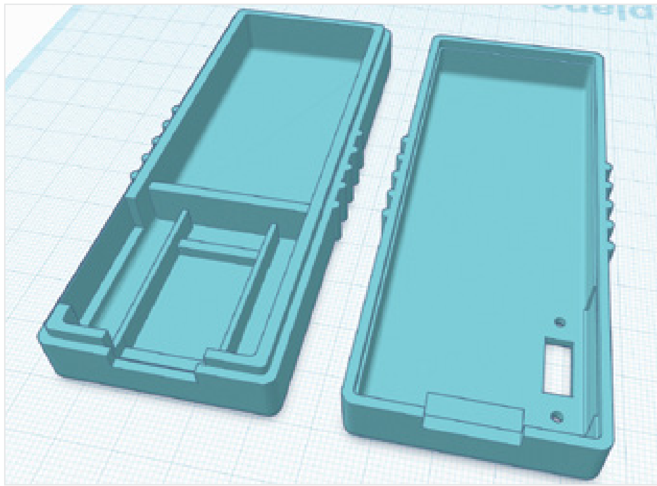
For clarity and convenience, the breadboard provides USB Ground, Data +, Data -, and Vcc, all in the same line as they appear in the USB connector. This simply makes life easier for ourselves than having them in random spots. Now apply a 5V power source to the breadboard, which can be from a spare Arduino or DC supply you have. Using your multimeter, test each of the four pins to ensure you have the correct voltages. Obviously pin 4 is your ground, while Data + should have approximately 2.1V relative to ground, and Data - should have approximately 2.7V relative to ground. We haven't tested the limits of where this will fail to work on a phone, but it definitely allows for tolerances with the resistors used, and we didn't encounter any problems.

IMPORTANT NOTE: It's critically important that you measure the voltages on all pins of the USB port. While the 5VDC module takes care of the heavy lifting, incorrect voltages on the data pins of the USB outlet could damage the device you're charging.

If you have a USB A socket, you can connect each of the pins to the pins on your breadboard, and connect a USB charging cable to your phone. It's easy to flip the connector and get your wiring backwards so be sure to check that you have things around the correct way. If your phone doesn't show charging, check your voltages again, to ensure you have the voltages on their respective data pins, and that you haven't wired the USB connector upside down.

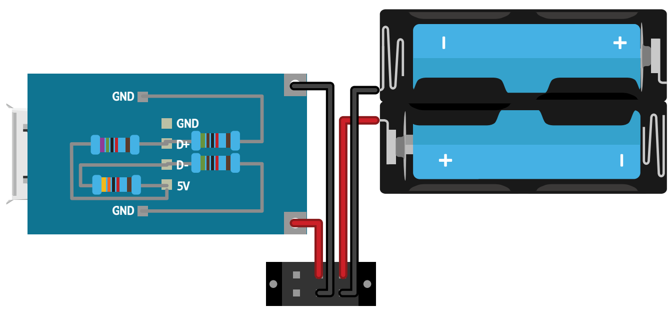
MAKING THE CASE

It's entirely possible to use this unit without a case, but we want to make it into a finished unit that's easy to use, and could sit packed into your emergency kit, glove compartment, or anywhere else. Therefore, we have designed a 3D printable case to house the entire project, and you can find the STL file in the downloadable resources. We basically build the circuit into the case, so it's prudent to print it first.

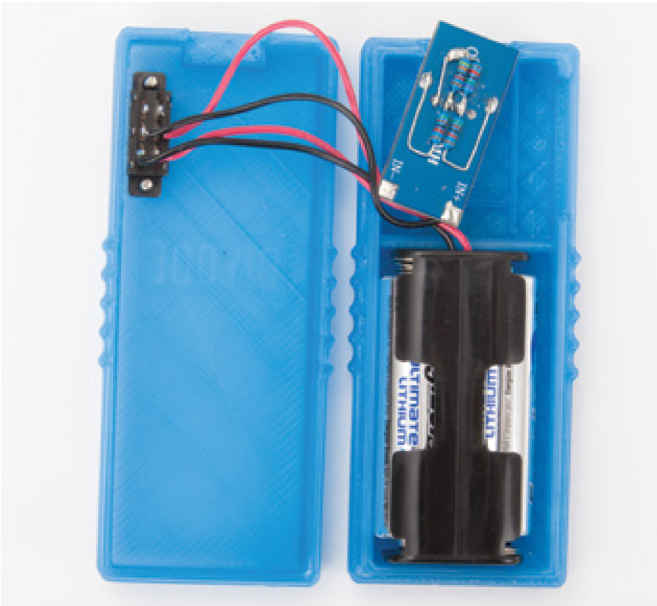


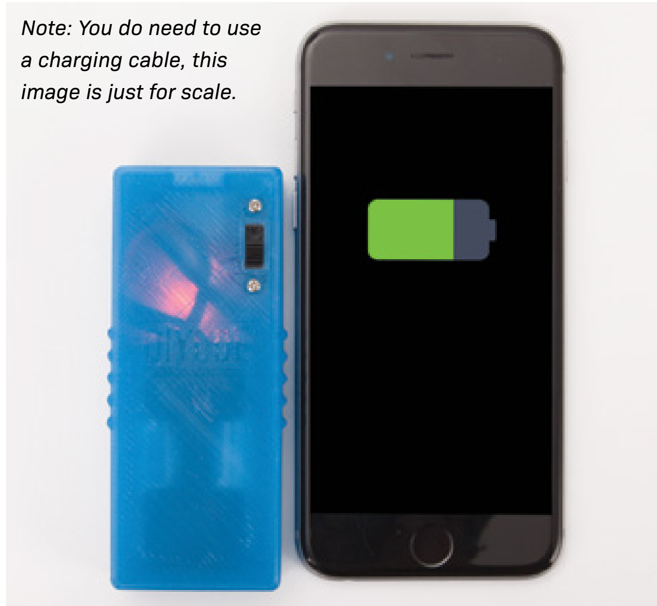
This case houses the DC module, two AA batteries, and provides a mount for a small switch. We have used a semi-translucent PLA filament to print our case too, so we can also see the small LED on the DC-DC module to indicate power on, without any additional electronics.

MODIFYING THE 5VDC MODULE



The prototype is great to test the theory, but isn't terribly useful past that point. Now that we know it'll charge the phone, let's make a real unit! Since, at the heart of this unit there are only four resistors, we decided it's best to just solder them to the base of the DC-DC converter, where the USB socket legs come through the PCB. This means we don't need a PCB, don't have wires everywhere, and it will keep it all neat and tidy. There is limited space under the USB socket, so we decided it's best to spread the connection points, as this allows us to reduce the risk of resistor legs accidentally shorting each other out. Once it's completed and tested, we'll coat





Note: You do need to use a charging cable, this image is just for scale.

it in some superglue, liquid electrical tape, or some other method to encapsulate what we've done, to reduce the likelihood of any shorting or problems down the road. You can see the diagram of the connections, as well as a photo of our prototype.

Once everything is soldered up, it's a good time to test things. Provide 3V to your DC-DC module (alligator clips and your battery holder will usually work just fine). You should see the on-board LED illuminate to indicate power. Now, grab your multimeter and check your voltages read as previously noted. If your module features an LED but it doesn't illuminate, it could be a sign of a shorted connection.

If your voltages check out to be reasonably accurate (we'd say, within about 0.1V of what we've described), then you should be safe to connect your phone and see if it charges. Grab your favourite USB charging cable. Connect it to your phone, and the USB socket. You should see your phone's charging icon appear. If so, IT WORKS!

IMPORTANT NOTE: You're connecting power to an expensive, sensitive device. Be sure to check things first. Most phones are intelligent enough to prevent any major issues, but don't risk it. Double check the voltages on the USB socket twice, then check them again.

It's important to mention that if you're using an old phone (good idea if you have access to one) that has been sitting in an discharged state for some time, it can take some time for the battery to have sufficient charge that the phone will turn on. Many phones such as an iPhone show you that it's charging at least, even while it's switched off. But just keep this in mind, to save yourself from chasing a problem that doesn't exist.

Mounting is fairly self-explanatory. Mount the battery holder with double-sided tape (though it's not really going anywhere). In order to make use of the switch, wire the positive lead from the battery holder through the switch, to the DC-DC module. Because everything is in such close proximity, we didn't have to use any additional wire, we just used the excess from the battery terminal's flylead. Then wire the negative wire from the battery hold, directly to the DC-DC module.

NOTE: We realised after some frustrations that the silkscreen on the under-side of our DC-DC module is backwards to what is marked on the top-side. Do your own investigation (even just trial and error) to ensure your silkscreen is the correct way around, as we're confident some of you will have the same module as we do.

WHERE TO FROM HERE?

Really, this is a complete unit, which is extremely simple to create. It would be possible to integrate some battery monitoring, a torch, or any other number of other features. For this purpose of providing emergency power however, we really wanted to keep the design as robust as possible. You could add a lemon-interface, but that's just crazy...!

It's also important to remember that this is designed as an emergency device. It's simply not practical to try and charge your phone from AA batteries all the time. The battery in your phone will leave a trail of AA batteries in its wake if you constantly try and recharge from them. But it should be enough to get your phone going in an emergency situation. ■

WANT MORE?

For 3D print files, additional media, or to discuss this project, visit:
https://diyode.io/004fczg

RETRO TEMP

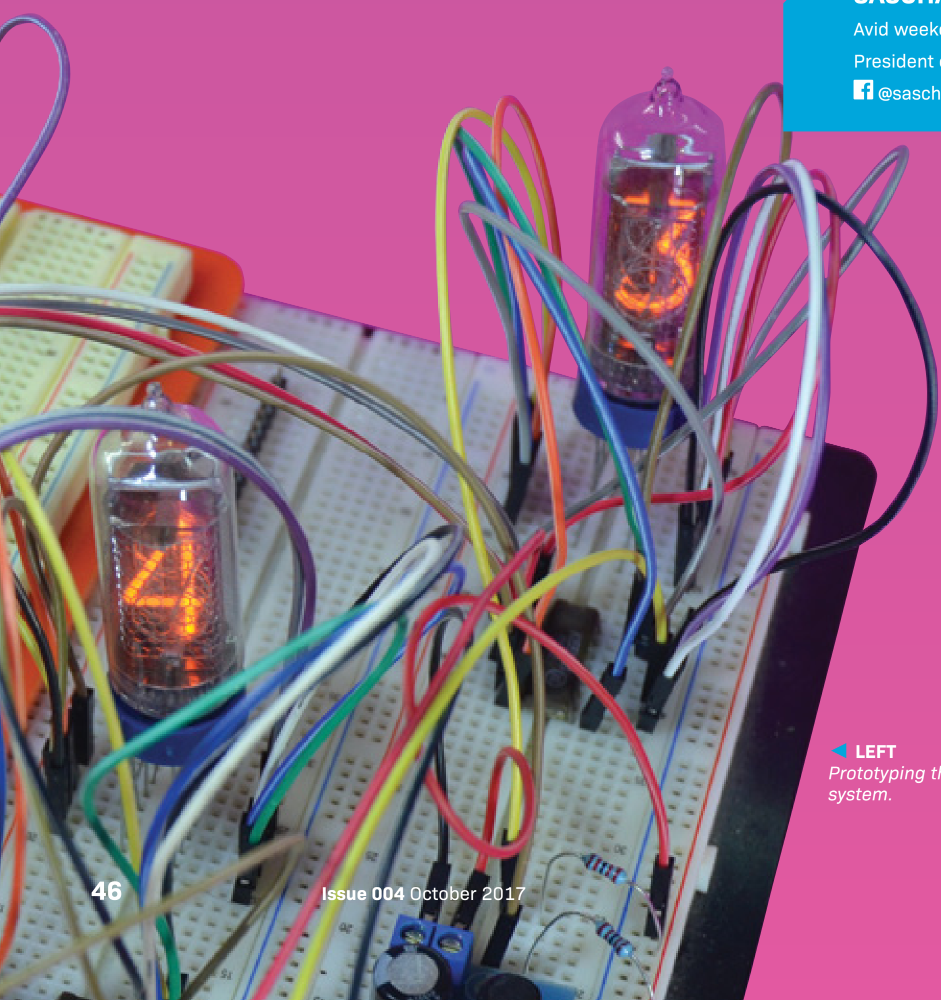
SASCHA GRANT

Avid weekend tech-explorer.

President of local makerspace M.E.S.H.

@sascha.grant

Digital displays might be the standard these days, but there's something special about the classic Nixie Tube.

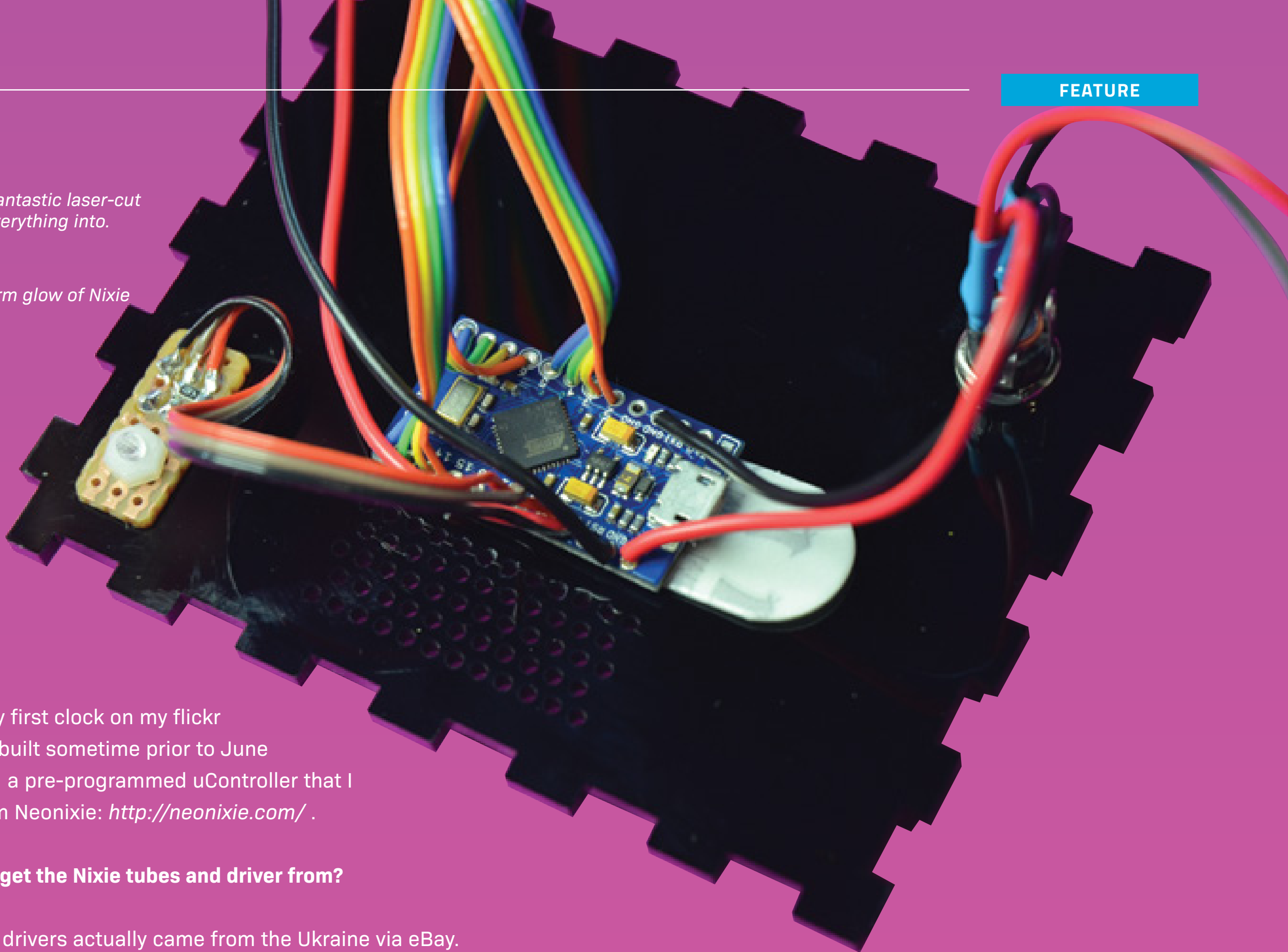


◀ **LEFT**
Prototyping the system.

We took a quick peek at Sascha's Nixie Tube Temperature Display, in our Community Submissions section of Issue 1. However, it received so much attention that we thought we should share the full story with you. So we spoke to Sacha about this fascinating application of Nixie Tubes.

How did you get started with Nixie tubes - have you used them before?

I discovered Nixie tubes a little over a decade ago, and thought they were very cool (that's a cold-cathode joke!).



RIGHT ▶
Sascha made a fantastic laser-cut case to mount everything into.

◀ LEFT
The stunning warm glow of Nixie Tubes.

You can see my first clock on my flickr stream. It was built sometime prior to June 2006 and used a pre-programmed uController that I purchased from Neonixie: *http://neonixie.com/* .

Where did you get the Nixie tubes and driver from?

The Nixies and drivers actually came from the Ukraine via eBay. There's always a good selection of tubes available, and so far I haven't had any problems with postage; you just have to be patient as international post from Europe can take a month or so!

Did it take much experimentation to get the prototype running?

There were a few steps I went through in putting the prototype together - I'd never dealt with the DS18B20 1-wire temperature sensor. I started by installing the Arduino library and playing with the demo code. Once I had the temperature displaying correctly in the console, I moved to driving the Nixies.

First step there, was to get a single digit running; just ticking over numbers sequentially, making sure that I had the Soviet K155ID1 Nixie driver IC wired correctly.

The prototype only ever got two digits installed, but from memory, I had the prototype running correctly over a few evenings after work.

Are you using a DHT sensor, or something else for temperature measurement?

I'm using the DS18B20 1-wire sensor in the TO-93 package, which looks just like your garden-variety transistor. The original intention was to have the sensor sitting just proud of the housing, but I discovered that the heat generated in the case runs at about five degrees above ambient, which meant I was getting incorrect readings. So I ended up moving the sensor about 10cm from the case, by using a short length of ribbon cable that was covered in heat-shrink tubing. Temperature readings are now pretty much spot-on with the comparisons I've done.

Yes we're getting used to displays not putting out much heat - that's an interesting point! How is the whole unit powered, and what's the power consumption overall?

A 9V 500mA power supply provides power to the high voltage circuit and the Arduino. I've found that the Nixie power supply produced by *LEDSales.com.au* is stable and reliable. In fact, this is the same power supply that runs my original Nixie clock from 2006 - providing more than 11 years of service!

The Nixie tubes run at approx 180V and around 2mA. The Arduino Pro Micro that I'm using is 9V tolerant.

Not really much at all for something that looks so amazing! Your laser cut case is great - what laser cutter are you using?

I bought my laser back at the end of 2009, to support my Model Rocketry hobby. I purchased it from WKLaser in China and learned more about importing, customs, quarantine, duties and paperwork than I ever thought I would! But by importing it myself (with the help of a freight agent), I saved a substantial amount of money in the process. The laser has a 60W CO2 water-cooled tube and a cutting area of 400mm x 600mm. You can see photos of this on my flickr stream.

At least you're putting it to good use! Did this project inspire any other Nixie-based projects? ››

›› I've built a few Nixie projects prior. I have a "people counter" that's still a work in progress. I'm the president of a maker space called MESH (Melbourne Eastern Suburbs Hackers), and I started building this to take along to maker fares and events like the Stringybark Festival (in the Knox Council area), for people to interact with. Like many projects though, I'm still tinkering away on it!

We often have as many projects on the go as we have completed! If you had your time again, what would you differently for a V2.0 of this project?

Oh, that's easy! Nixie tubes have a finite lifespan and there are some techniques used to maximise their life. So the first thing I'd do for version 2 would be to add an ambient light sensor, which would automatically dim the Nixies when the light in the room drops, and I'd add a "slot machine" type effect to cycle through all the digits every few minutes.

Both of these additions would boost the visual appeal and increase the lifespan of the Nixies.

I'd also love to build a wooden box to house everything; maybe in a lacquered walnut or something similar, as I feel the acrylic is a bit cold and the tubes would look much better with a nice warm timber surround. Maybe one day I'll try my hand at woodworking!

At least you have a plan! Increasing the lifespan is also a valid point - not something we really think about with an LED display (obviously). What are you working on now?

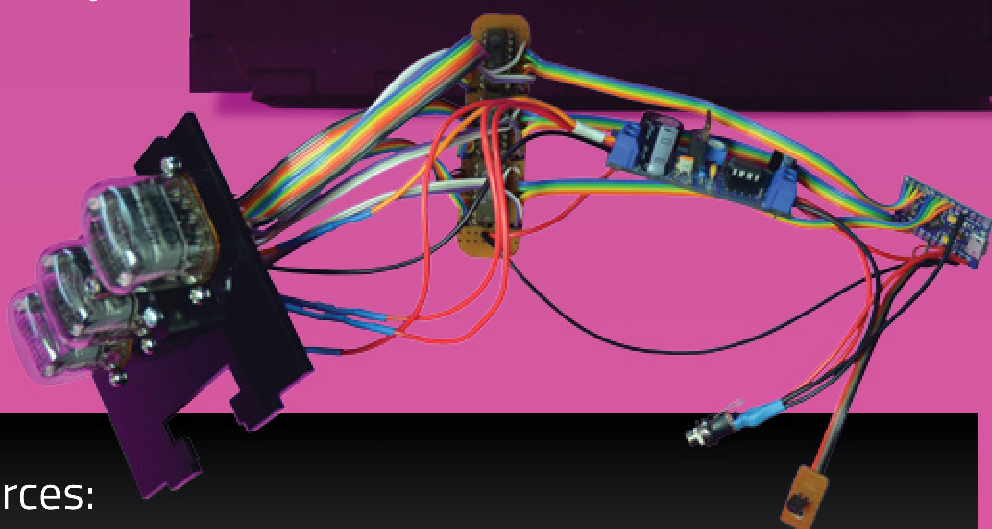
I'm currently focused on a Bartop Arcade Machine using a Raspberry Pi and 10" display. I'm a bit of a retro gamer and have seen some absolutely fantastic looking machines on the internet. I've got all the parts now sitting there, just waiting for me to dedicate some time to start.

Sounds awesome - we can't wait to hear about it in the future! ■

GOT SOMETHING TO SAY?

To discuss this feature, visit:
https://diyode.io/004cyww

ABOVE RIGHT ▶
The display is bold, sure to capture attention.



BELOW RIGHT ▶
The full harness before it was mounted into the case.

Reading & Resources:

- **NIXIE TUBE & NEON BULB POWER SUPPLY KIT** *https://www.ledsales.com.au/index.php?main_page=product_info&cPath=142_144&products_id=652&zenid=0th2h6de7pjeopi4f1br0ffcf5*
- **GITHUB** *https://github.com/ibuildrockets/NixieTemperatureDisplay*
- **NIXIE TEMPERATURE DISPLAY** *https://hackaday.io/project/3885-nixie-temperature-display*
- **NIXIE TEMPERATURE DISPLAY ON FLICKR** *https://www.flickr.com/photos/oflittleinterest/albums/72157650053263577*

TOOLS FOR MAKERS.

IT'S TRUE
Technician multimeter. True RMS. Autoranging. CATIII. 1000VDC, 750VAC. Up to 10A. **QM-1321**

PRECISION
High quality 32 piece precision driver set. Case included. **TD-2106**

THIRD HAND
PCB holder with LED illuminated 2x magnifying lens. **TH-1987**

POWER UP
0 - 30VDC regulated power supply. 0 - 5A. 1mV ripple voltage. **MP-3840**

HOLD IT!
Sturdy PCB holder. Holds PCBs up to 200(L) x 140(W)mm. TH-1980

MEASURE UP
Calipers with LCD and etched vernier scale. 150mm range.
BUDGET TD-2081 **$13.95**
PROFESSIONAL STAINLESS STEEL TD-2082 **$39.95**

FOR THE HOBBYIST
40W Temperature Controlled Soldering Station. Up to 450°C. TS-1620

ANTI-STATIC
Adjustable wrist strap. 1.8m coiled lead with banana plug/alligator clip. TH-1780

STARTER BUNDLE
DMM, 25W iron, solder, de-soldering tool, and much more. TS-1652

CIRCUIT REPAIR
Adhesive Copper Tape. 5mm x 10m. NM-2870

$6.95

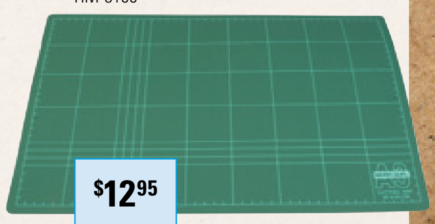
LIQUID ELECTRICAL TAPE
Seals and protects electrical connections. 28g tube.
BLACK NM-2836
RED NM-2838

$16.95 ea

CONDUCTIVE GLUE
A soldering iron in a jar. Lead-free. Cures overnight. 9ml. NM-2831

$9.95

WORK IT
Durable A3 size PVC cutting work mat. HM-8100



$12.95

For Tools & Test Equipment, visit:
www.jaycar.com.au

jaycarelectronics   @JaycarAU   @jaycar_electronics

jaycar Electronics

Prices shown in $AUD. Savings off original RRP. Valid to 31 Oct, 2017.

Coping With Capacitors

Storing electrical charge, a little like air in a tyre. Capacitors are valuable components, and easy to understand. - by Daniel Koch

Last month, we looked at resistors, arguably the most commonly found component on a circuit board. This month, we continue on to capacitors, which are a versatile component used in dozens of applications around even a moderately complex circuit. Capacitors are so versatile, in fact, that we only have space to talk about using them with direct current, which is by far the most common situation most makers will find or use them in. Their use with alternating current, therefore, will be left for another time. One final note – this article will no doubt look very similar in sequence and content to many textbooks or primers. This is because we are not proverbially reinventing the wheel; rather, we are describing it, and there is usually a lot of similarity when many people describe the same wheel!

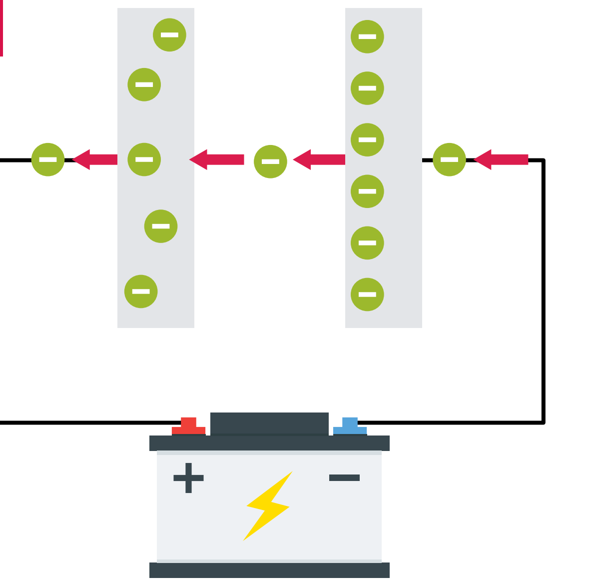
CAPACITANCE

It is said that a person must learn to crawl before they can walk. In the case of electronics, it is more that we need to learn what feet are and how they work before we can use them. So before learning about capacitors, first we will summarise capacitance.

The word capacitance is formed from the root word “capacity”, which we use across our lives to describe the ability to hold, carry, or store. This is no accident, as capacitance is the ability to store an electrical charge. This can occur in any situation where two conductors are separated by an insulator. This means that the power lines out in your street, if you are in an above-ground transmission area, have capacitance between them, separated as they are by the air gap between them.

At the other end of the scale, the windings of an electric motor also have capacitance between them. Even the tracks on a circuit board have a small amount of capacitance between them. In most cases the capacitance is either negligible or irrelevant; however, anyone working in specialised situations such as radio or medical electronics, which are usually highly sensitive, needs to remain aware of this.

1



To understand capacitance, consider two metal plates separated by an air gap **[1]**. When the DC power source is applied, electrons flow from the negative side of the supply to the plate they are connected to, causing that plate to have a negative charge.

The electrons at that plate repel the electrons on the other plate, which are drawn to the positive side of the battery. Therefore, current flows for as long as electrons are on the move, from the positive plate to the negative plate of the battery. This is referred to as the “charging current”. Most makers typically encounter this term and its associated maths when calculating timing circuits.

The quantity of charge (Q) in a capacitor is a function of the Capacity (C) of the capacitor, and the applied voltage (V). If the voltage is increased, the charge will also be increased. ($Q = CV$) However, the imbalance of electrons results in a stored charge in the form of a potential difference between the two plates. In theory, the capacitor will remain charged forever. In the real world though, even the best insulator is not perfect, and electrons will slowly leak across the gap, or back through the non-infinite resistance of the open circuit. A modern plastic encapsulated capacitor can store a charge for a very useful time, long enough, in fact, that people working on high voltage circuits need to deliberately discharge capacitors before working on them, to avoid lethal shocks, even after an entire day for some capacitive components.

The value of Capacitance is measured in “farads (F)”, in honour of the work of Michael Faraday. The Farad is a unit describing how much charge a capacitor can store. Those who have read The Classroom article in Issue 3 may remember that in DC terms, a capacitor value of 1F will take the equivalent of 1A of current for 1 second to charge, and result in a potential difference of 1V. Of course, these numbers are proportional, so a 1F capacitor charging at a potential difference of 10V at 1A will take one-tenth of a second to charge.

CAPACITORS

For now you can imagine a capacitor as two metal plates separated by an air gap, and for some capacitors this is true. In fact, old valve radios, and even transistor radios used a capacitor to tune the radio stations that was made from one set of fixed metal plates, and another set of moving metal plates, separated by nothing but air.

A modern practical capacitor has two thin insulating films called a “dielectric”, that have been metallised by coating one side with an Aluminium vapour. One type of capacitor construction places many layers of the film one on top of another, to be then cut into ‘blocks’. Another type rolls the two films together into one long sausage which is then cut into sections of the capacitance required. Either type is connected at the ends, or sides, by the component leads. The whole assemble is then dipped or molded into the component case.

The two construction methods each have pros and cons, but for average users, there is no difference in the effect. Do note however, that some are better for high frequency, others for low leakage, some for higher voltage and others for higher capacitance.

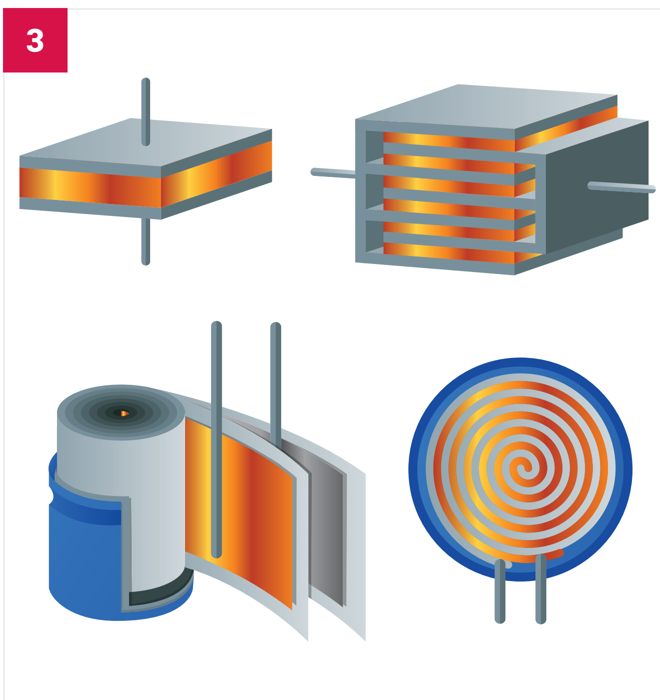
To the casual observer, they have a similar physical layout, being somewhat flat and broad in one plane and quite narrow in another **[2]**. Capacitor types use a variety of dielectrics, from ceramic, mica (a mineral insulator), and of course, polyester. In most cases they are named after the dielectric they use. It should be noted that these types of capacitors are all non-polarised. That is, they can be connected with negative or positive connections either way around, and will work the same way. Additionally, these types are usually ››

›› of lower capacitance, typically values under 1µF.

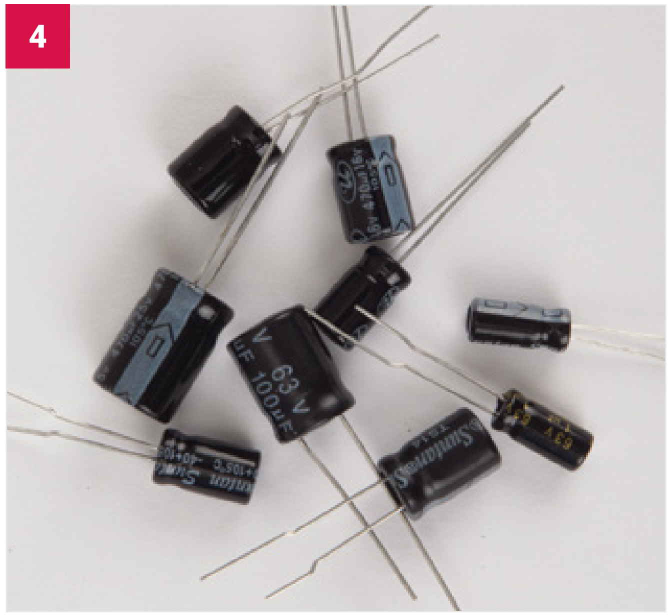
2

Electrolytic capacitors, which have a distinctive can-shape, use a chemical electrolyte as the dielectric material. The chemical action of the electrolyte forms an oxide layer on one sheet of foil during construction,but not the other. This means that electrolyte capacitor types are polarised capacitors, and they can only be used with the negative electrode more negative than the positive electrode. Electrolytic Capacitors are always marked accordingly, with through-hole types having a stripe down the negative side, and a series of '-'s marked down that side. Surface mount types having a black line to one side of the top of the case **[3]**.

3



4



Although it is normally loathsome to suggest Wikipedia as further reading, simply because it is open to editing by non-qualified people and issues can take time to identify (anyone who has been through university will remember the golden rule – Wikipedia in a reference list will either cost marks or constitute a fail!), in the case of electrolytic capacitors, there is actually a very informative and in-depth article that can be sourced from there. Check out *https://en.wikipedia.org/wiki/Capacitor*

For small physical size, but limited capacity, Tantalum capacitors **[4]** are an alternative to electrolytic capacitors. They have a very different construction, and are quite effective devices, if kept within their limitations. Tantalum capacitors are expensive compared with electrolyte capacitors, and therefore less common. Tantalums behave in a circuit the same way as electrolytic capacitors – they are polarised and usually of larger values than non-polarised types. Tantalum values are more consistent and more accurate, and are therefore are often used in timing circuits of longer durations.

USING CAPACITORS

As explained in the previous article on resistors, capacitors also have a variety of markings to show their value of capacitance, and other parameters, depending on type. Additionally, while a 1Ω resistor would be considered small and uncommon, a 1F capacitor would be considered large, a Super Capacitor.

Capacitors in common use range from less than ten milliFarads, still referred to as 10,000µF, down to 1 picoFarad, or 1e-12 Farads **[5]**! Yes, truly tiny numbers! For this reason, decimal dividers must be used.

	WRITTEN AS	VALUE IN FARADS	NUMBER OF DECIMAL PLACES FOR SINGLE SIGNIFICANT FIGURE WHEN EXPRESSED IN PICOFARADS	SCIENTIFIC NOTATION	FRACTION OF ONE FARAD
FARADS	F	1	0	NA	1
MILLIFARADS**	mF	0.001	3	x 10^{-3}	1/1 000
MICROFARADS	μF	0.000001	6	x 10^{-6}	1/1 000 000
NANOFARADS	nF	0.000000001	9	x 10^{-9}	1/1 000 000 000
PICOFARADS	pF	0.000000000001	12	x 10^{-12}	1/1 000 000 000 000

***millifarads are not commonly used, they are included for the sake of completion and clarity of sequence.*
Note: All references to pF, picofarad, nF, nanofarad, μF, microfarad, etc should use a capitol F in Farad.
Micro (u) should be represented by a 'μ' (Greek 'mu').

For most smaller capacitors, below 1μF, a multiplier system is used that works in the smallest common unit of capacitance, 'picoFarad (pF)'. The value will generally consist of three numbers, a letter, and a voltage rating in some cases. Some other capacitors only display the three-number value. As with resistors, capacitors are made in a set range of values, such as the e12, e24, e46 series. So, you could purchase a ceramic disc capacitor in 4.7pf, 47pf, 470pf, and so forth, but you are unlikely to find a 44.3pf capacitor.

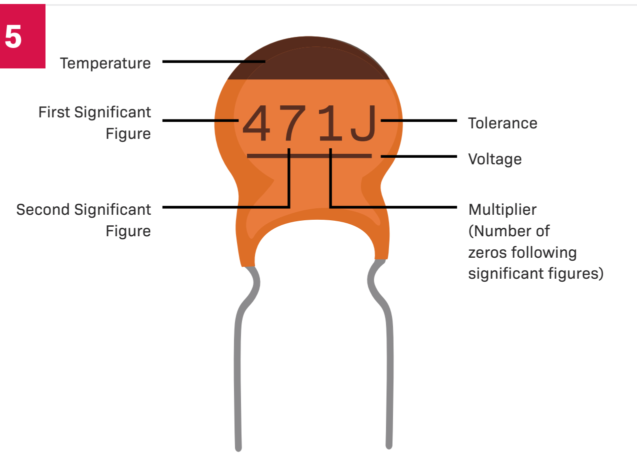
To read the codes, the first two numbers are the significant figures, while the third is the number of zeros used to give the value in picofarads. Therefore, a 470pf capacitor would have a value stamp reading "471". This is significant figure "4", significant figure "7", with the "1" corresponding to one multiple of ten, or one zero, following. Unfortunately – and very confusingly – some manufacturers use the "0" when there is no multiple; so 47pf can be stamped as "470". This does exist but is not common; in fact, in 14 years of selling electronic components, I only saw this a handful of times.

A hint: Whether Resistors, Capacitors or Inductors, the exponential function of the calculator is a great tool for beginners. The 471 above would be typed in as 47 [exp] 1 [enter]. Try it! Also for various other notations such as 333, 684, or a common group, 102, 103, 104 etc. For resistors, once you can translate the colours into numbers, you can also enter them straight into the calculator until you learn what each colour code means. Brown, Black, Black,Orange? Go on try it.)

Generally, a value of 47pF would be stamped as "47" and all higher values will be three digits. The codes work the same way on ceramic, MKT, greencap, and polyester capacitors **[5]**. Greencaps are a form of polyester capacitor; however, there are others.

On some capacitors, particularly greencaps, a letter may be found.

5

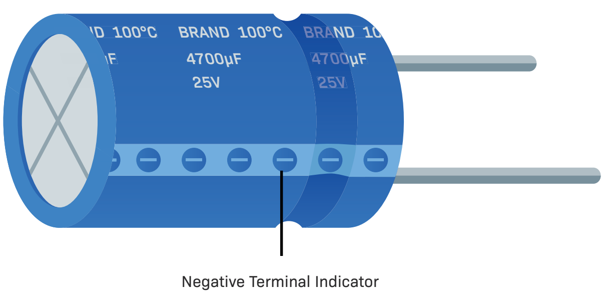


This is a tolerance designator. All components are manufactured to a tolerance, which means the value can be over or under the rated value by a percentage. The codes are generally the same as for resistors. Additionally, some capacitors have a voltage rating stamped on them. This is usually directly stamped, with no coding or multiple system. Some polyester capacitors, for example, have "100V" on them, which literally means they are rated to 100 volts.

For larger capacitors, usually electrolytic capacitors, the value is directly stamped. In most cases the value is in microFarads (μF). The "u" in this abbreviation is actually the Greek letter "mu", which has a longer tail on the left side extending down (μ), but many standard printing fonts do not cope with this. So even though 4.7 millifarads is a real value, it will be written as 4700 microfarads, or 4700μF. This has led to the tendency for most users to talk in microFarads instead of milliFarads. So, numbers of ten thousand microFarads are heard commonly, despite this being more mathematically correct when stated as ten milliFarads. ››

›› As with resistors, some manufacturers prefer to avoid decimal points, so you may see a value of 3.3µF stamped as 3u3, or even 3µF3 **[6]**.

6



The second piece of information on an electrolytic capacitor is its voltage rating. Electrolytic capacitors ALWAYS have a voltage rating on them, as well as a stripe to indicate the negative terminal. This is because electrolytic capacitors behave differently unless used at or close to their rated voltage.

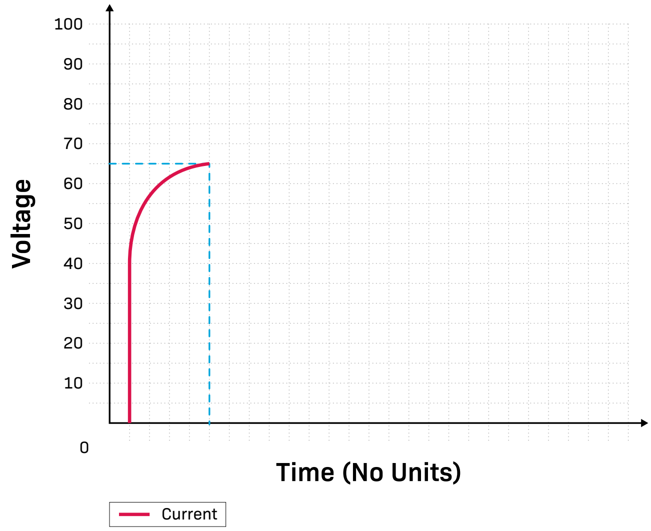
A curious property of electrolytics is that they only begin to show a curve in their charging current when operating closer to their design voltage. If used well below this, they may not respond as expected. This has implications in timing applications as discussed ahead. A 10µF 63V electrolytic may give a different timing result than a 10µF 16V example when used on a 12V circuit.

Electrolytic Capacitors also have a temperature rating. Mostly for higher power applications but essential knowledge if you are to repair a circuit such as a power supply, particularly a SMPS. Often such power supplies use a 105° component, and when it fails, it should be replaced with a 105° component. A power supply or amplifier that has used an 85° component to replace the original 105° value, will continue to fail, periodically, until the correct temperature rating is used.

POWER SUPPLIES AND CAPACITORS

One of the most familiar situations involving capacitors for many makers, will be the small cluster around the power supply on their breadboard's plug-in supply, or around a board that take a plug pack as a power source. In this situation, capacitors are employed so that their ability to store and then release a charge can be used to filter and stabilise the power rails. In very basic terms, they absorb spikes by charging up as quickly as they can, then slowly releasing this charge into the circuit **[7]**. There are many other reasons to use capacitors around power circuits; however, without the space to describe inductance, reactance, and resonance, this explanation is best left for later. The important thing to know is that capacitors across a power supply rail can help with stability and the reduction of electronic noise.

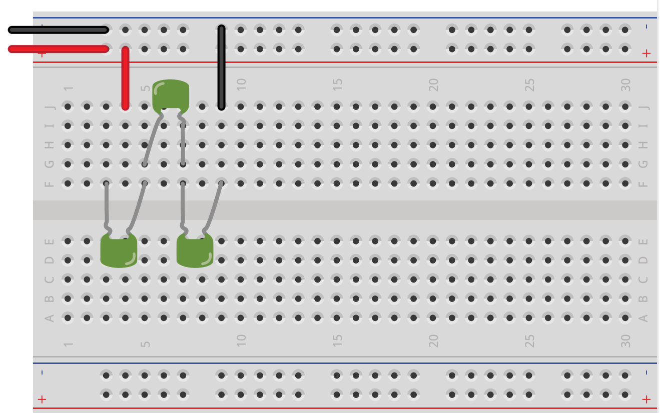
SOME QUICK MATHS

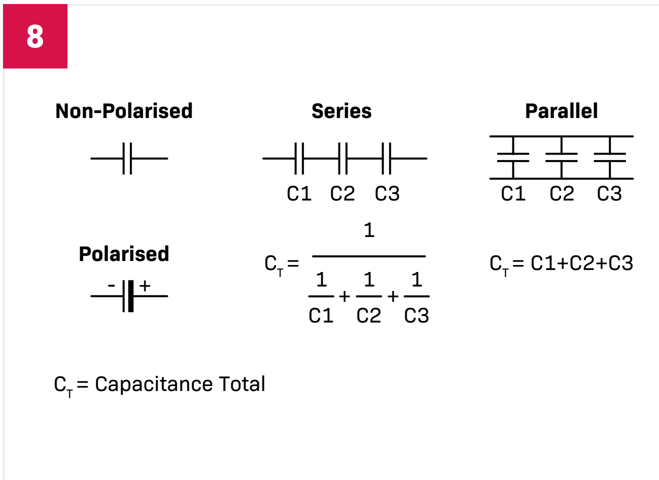


Capacitors can be used alone, or in combination with each other. If so, they can be connected in "series", or "parallel". There is nothing tricky about these names, they are what they describe.

When working in series, the total combined capacitance is actually lower than the smallest capacitor in the series **[8]**. The advantage to this is that the total potential difference that can exist across the combined chain, is the sum the voltage ratings of the individual capacitors. So three 100pF, 100V greencaps in series now yield a capacitance of 33.3pF, but can withstand 300V. The increase in voltage is the main reason for adding capacitors in series.

7

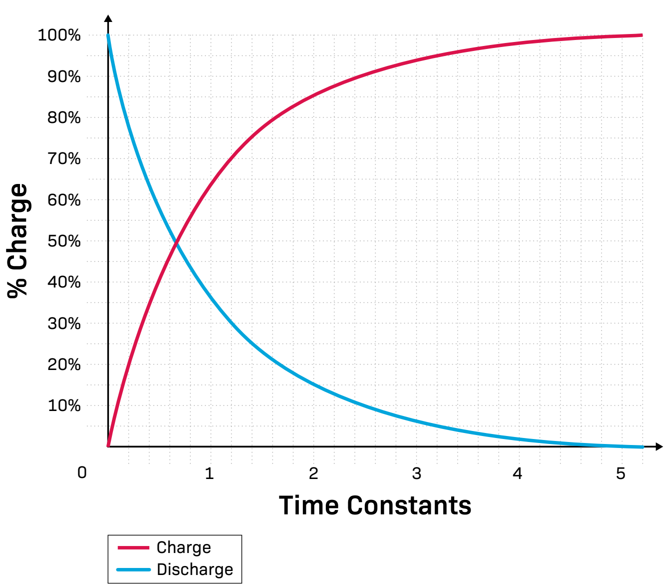




On the other hand, things are simpler for parallel circuits. The total capacitance is simply the sum of the individual values, while the voltage ratings are unaffected. Of course, the smallest voltage value determines the voltage that can be applied to the network, without causing damage to that capacitor.

TIMING WITH CAPACITORS

This same property of capacitors can be used to form a timing circuit. With the exception of electrolytic capacitors operating well below their design voltage, capacitors operate to what is known as an exponential, or "Universal Time Curve". To do this, they are used in series with a resistor, and called an RC network. The values of the capacitor and resistor do not affect the relationship curve; however, they will affect the length of the axes if graphed to scale. This is because of something called a Time Constant (TC). A TC is determined by an agreed standard, and it is defined as the time it takes for a capacitor to charge, via the resistor, to 63% of the voltage being supplied.



The maths for this is fairly simple: TC = R x C, with R in ohms and C in farads. Be careful to convert both the Resistance and capacitance to a whole unit, and not use the decimal multiplied or divided form.

The resulting number is the length of one time constant in seconds. This could also be a decimal with quite a few places, which you would usually convert for practical use, to milliSeconds or even microSeconds.

Once the time constant has been determined, the length of time a given RC network will yield can be determined. If your circuit will trigger at two thirds of the supply voltage, then your time delay is ever so slightly more than 1TC. If it is nothing like what you need, then you have an educated guessing point with which to try other values in the equation. For many makers, not being engineers, guess-and-check will be quite a valid way of working.

If you choose to trigger at around 63% and set your circuit to that value, you can choose a starting value of capacitor and a resistor, either from experience or by performing the maths. e.g. 1MOhm and 1µF have a TC of 1 second. You can begin by changing values in decades to get close to the value you want. Let us say you want a circuit to turn on in 200mS. You can drop the value of either R or C by 10, purely aiming at the next lower decade, and get 100mS. Then double one or the other value to get 200mS (e.g. 200kiloOhms and 1µF, or 2MegaOhms and 0.2µF).

Of course you would need to choose between standard values of 220kΩ, and 180kΩ. In engineering terms, there are other considerations that would determine the values used, where a large resistance may be undesirable, for example; but this will still be a good starting point.

To learn more, see the next page's "Hands-On Activity". Note: the maths for this involves transposing or rearranging the equation, so it becomes: R = TC/C.

That's all for now. However, on the next few pages we have included additional information, which may be of further interest, including a section specifically for educators. ■

GOT SOMETHING TO SAY?

To discuss this edition of The Classroom, visit:
https://diyode.io/004ymwg

For The Educators:

For educators, metalanguage is a central part of learning in any subject. These terms are all subject-specific, and important to students learning the field, as the words all have defined, often-used meanings; understanding of which, is vital for success.

Science is no different, whether being taught at kindergarten level or senior high school. What sets science apart, however, is the sheer volume of metalanguage that may be encountered. This is because of the almost limitless variety of topics that may be used to satisfy some outcomes. The question though, is how to teach it.

An example will, of course, be explained and defined at a point in the lesson; however, research shows that most students need to repeatedly engage with a technical word before its meaning is fully understood. Even if the concept sinks in, spelling may be an issue for some words, affecting fluent written communication.

In primary teaching, this is often overcome with a word wall, mind map, or graffiti wall that students can constantly refer to. These artefacts have the correctly spelled language, possibly with definitions, in a visible place that can be referred to at will. Younger students enjoy these brainstorming activities, and are usually in the same space lesson after lesson for a given topic, even if they change classes for subjects, as some primary schools do.

For high school teachers, the challenge is greater. Older students find such activities childish (although it is amazing how much we love them again at university), and often have no regular classroom for a given topic. In discussion with a high school teacher friend of mine, we adapted some of my primary school ideas for his robotics lessons in science class. This discussion eventually involved the whole science faculty, along with myself – a ring-in primary teacher! The result was a three-tier differentiated activity, which resulted in a cheat sheet that the students were able to access digitally at any time.

For struggling students, words were provided that they had to find definitions for. Some equations were also provided, which the students had to show worked examples of. The next tier was a handful of words and equations, where students were required to define and work examples. They also then had to find the same number of examples themselves. The final tier was a blank sheet with a required number of words and equations that the students had to find, define, and work as examples.

These three tiers were all marked against a rubric, constructed collaboratively by the teacher and students. They offered marks for thinking skills, processes, and creativity as much as for correct results. The resulting cheat sheet is allowed as a resource sheet during assessment which itself, aims to assess skills and processes rather than knowledge. This is proving very popular with the students at the moment; however, the unit of work had not concluded at the time of writing this. I do feel that it may be useful for other teachers who find that students do not absorb metalanguage well during direct instruction.

Hands On:
Demonstrating Capacity

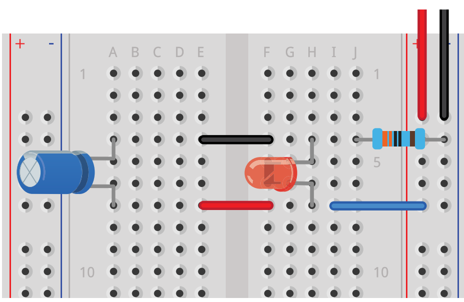
To get a feel for capacitors in general, and RC networks in particular, this simple, hands-on activity uses just a few components and a breadboard to demonstrate how the value of a capacitor affects the time it takes to charge and discharge; and the effects that a resistor can have on this.

For this experiment, we'll work with electrolytics. They're cheap and have a fantastic range of available values. Many makers will have an assortment of components on their workbench for testing or development, so the capacitors in the following parts list are a guide only. If buying values for a class, decade values are the best as they clearly show the effect of going up or down a decade.

PARTS REQUIRED:

Electrolytic Capacitors in:
10μF, 47μF, 100μF, 220μF, 330μF, 1000μF
1 x Diffused LED in any colour (low current draw)
1 x 220Ω Resistor (for LED)
Resistors in Various Values (we suggest 10Ωm, 10kΩ, and 1MΩ)
1 x Small Breadboard and Prototyping Jumpers

EXPERIMENT ONE



Follow the diagram above with just one capacitor. Connect your power supply to your breadboard, probably the 5V supply often used with Arduino and Raspberry Pi experimentation.

Note: Because the power supply may have capacitors also, we are using links to break the power to our test circuit. This ensures a clean application / shut off of power. If we simply turned the power supply on and off, it could skew our results by still ramping power up and down with the internal capacitors.

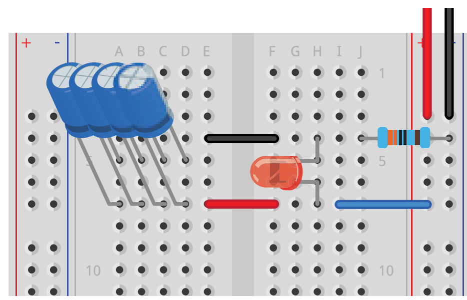
Following the diagram below, wire up everything but the blue jumper wire. This will be our power-control wire. You could use a switch in-place of the wire. Both options will work fine and produce the same result.

Using the smallest capacitor you have, insert it taking care of polarity shown and connect the blue link. Watching the LED, notice if it turns on instantly or slowly. Then keep watching when you remove the blue-coloured jumper wire. You should notice a nice gradation in the LED, turning on slowly, and off slowly.

Replace the capacitor with the next value up, and reconnect power. Continue until you have worked through all your chosen values. You can record the time the LED stays on, the plot the results on a graph. You should notice a pattern!

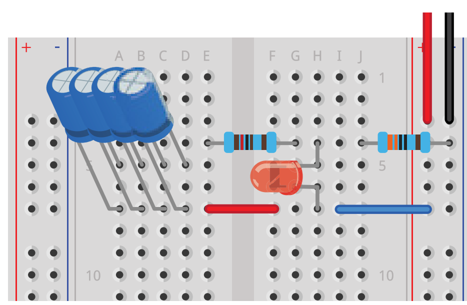
The capacitor values will have a predictable and reliable effect on the power to the LED, inline with their capacitance value.

EXPERIMENT TWO



Follow the diagram above. Add one or more additional capacitors in parallel, and note what effects this has on turn on / turn off time for the LED.

EXPERIMENT THREE



We'll simply replace one of the links with a resistor. Using the same method as before to apply and remove power, notice how the use of different resistor values between the capacitors and LED change the rate and which it turns on and off.

Design And Make:
The Ubiquitous Logic Probe

This is the first of our discrete projects! A logic probe is a powerful tool for any maker to have, since it simplifies debugging and testing. – by Bob Harper

Why do we use logic? Because all modern computers use logic. To not use logic would be illogical!

Logic sometimes fails, however, when the analog levels that control the inputs to gates, or the loading on the outputs, cause a misunderstanding. Logic works when the signal levels comply with logical requirements, (i.e., standards).

TESTING LOGIC AT THE HOBBY LEVEL

Many years ago I learned logic as a means to control stage lighting for rock bands. I knew electronics, but had never even heard of TTL or any of the digital logic technologies, except for switches and relays as I was, at that time, an apprentice electrician.

Later on I used logic ICs, 74xx devices in my first computer that consisted of seven ICs. It didn't take much more circuitry before I needed something better to test these circuits. Eventually I saw some articles about logic probes and built one, using just four transistors and a few resistors. Recently I was asked if I could make a logic probe and that is where this story began.

MAKER LOGIC

Robotics, 3D printers, CNC mills, routers, and so on are all applications of the inputs and outputs of computers, being used to drive mechanical and opto-electric devices. The logic analyser is a terrific instrument, but overkill; and usually over-priced for hobbyists. Multimeters often have a logic function, but they can be slow, insensitive, and inaccurate. They can also be inconvenient; for example when you're trying to hold a probe on a tiny pin, and you need to avoid touching the many other tiny pins that sit around it, when looking back and forth at the meter.

HOW IS A LOGIC PROBE BETTER THAN A MULTIMETER?

The logic probe has two or three coloured LED indicators near the probe tip, so the user can see the result without looking away. It also clearly shows three conditions using two of those LEDs. Also, multimeters read logic "0" or logic "1", but not all meters can handle levels in between. Therefore, they may make mistakes in interpreting an open circuit or dead gate.

WHAT DOES A GOOD LOGIC PROBE NEED TO DO?

As we mentioned, typically logic probes read logic "0" and logic "1". This may sound simple, but it's not necessarily the case.

Firstly, it will help to know what type of system it will be used on. On some logic systems, the probe would read up to 48V for a logic "1" on relay logic, and perhaps +/-12V on other systems. Many systems have existed, most are now gone, relay, DTL, ECL and so on.

So let us begin by defining our logic probe as intended for TTL. You might say 0 volts for logic "0", which although commonly agreed with, is not completely correct. In fact, most systems never actually reach exactly 0V. The truth is, TTL accepts from 0V to 0.8V as meaning a logic "0", and CMOS accepts 0V to 1.5V as logic "0", at least on a system with 5V rail voltage (and up to 4V on a system with 15V rail voltage). Then, we realise that the Arduino, along with many other computer systems, use either 5V or 3.3V rail voltage as a modern option. Thankfully, a 3.3V system still uses TTL levels for logic "0" (i.e., 0.8V).

The output voltages are another issue. TTL, either 5V or 3.3V systems, expect the output voltage to be from 0V to 0.5V, while the CMOS system expects the output voltage to be less than 0.05V – the main reason for lower power requirement of CMOS.

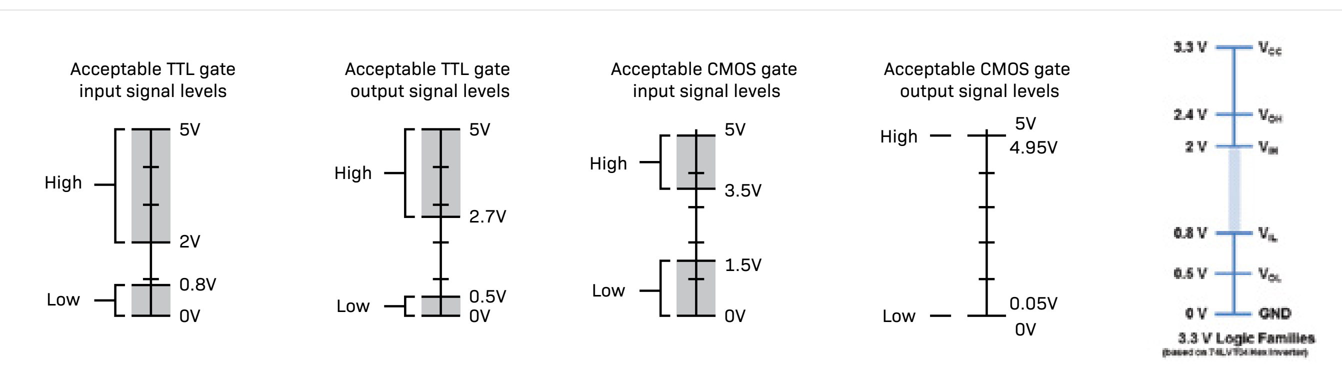
Logic "1" levels for TTL inputs (both 5V and 3.3V systems) need to be greater than 2V; but for CMOS, on a 5V supply, the inputs need to be above 3.5V to be a valid "1".

So a professional probe requires a different design for each system, or a switch to select the system in use. For hobbyists, unless chasing a glitch at the very edge of a system, or attempting maximum "fan-out" (loading), where noise can cause false logic, a hobbyist can suffice with 0V to 0.8V as acceptable logic "0", and above 2V as logic "1". In fact, a technician would probably pull out an expensive CRO or logic analyser by then anyway.

LOGIC SHIFTS

In a system where the logic is 'state determined' and the outputs don't change unless an input is changed, logic "0" and logic "1" are quite sufficient. On low speed systems, a flickering pair of red and green LEDs might be enough to say the clock is ticking, but even experienced technicians have difficulty on faster systems. The main issue is knowing whether the system is clocking, or held in a high impedance state or centre voltage (i.e., between a logic "0" and a logic "1"). To identify this, a logic probe can have a 'pulse' LED, usually orange, that may simply light when oscillating, or flash at a fixed frequency when a pulsed signal or clock is running. Some probes have a function to 'catch' a transition, meaning that the orange LED turns on and latches on until reset.

If the probe tip changes logic, or even if the logic momentarily changes then returns to the initial state, whether correctly following the logic, or due to a glitch, the latch will pick it up. This is useful for catching sporadic and unintentional pulses that may trigger logic circuits intermittently, and especially if the circuit is triggered unexpectedly and unpredictably. ››



›› OUR LOGIC PROBE DESIGN

While we would be proud to say we designed a new type of logic probe, it would be very likely that somebody else had already published or uploaded something very close to this design, even if we have done all the hard work ourselves. Anyway, the first step most engineers would take is to research what has already been done. This circuit uses the best method we know of, using a simple IC, and for very little cost.

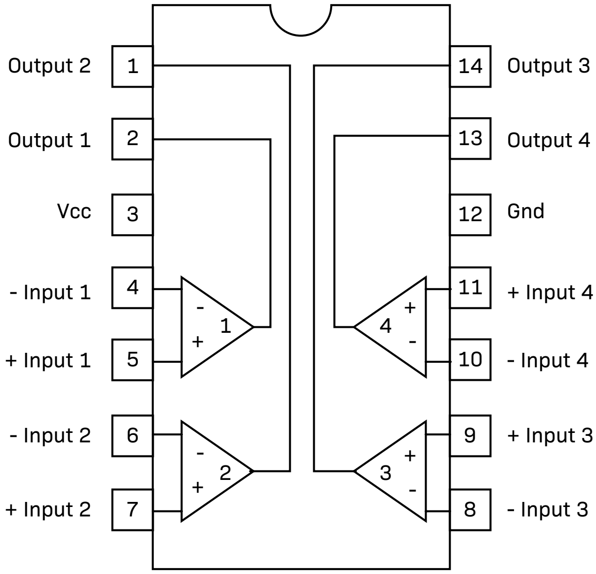
WHAT DO I WANT?

Honestly? I want something easy to make, which will last forever and cost me nothing! Or as close to that as possible! With this in mind, I researched many existing designs, some as early as 1975, to try to compare approaches.

Hobbyists could easily get away with two LEDs and two resistors, and that approach is good when you need something cheap and/or fast! So long as you realise that you will be getting very basic results, and will be loading the circuit so much you might actually cause logic faults in the process.

So, as in all test instruments, we want the probe to take as little from the circuit as practically possible, and try to be sure that the voltages are within the limits shown in the diagram below – at least for TTL and perhaps 3.3V TTL.

Another factor can be the speed of operation, but hobbyists are usually happy with basic technology, so although high speed quad comparators were researched, we settled for an old friend that is easy to find, cheap, and has good manners: the LM339, which is also used in many of the designs already available on the internet and elsewhere.



WHAT IS A COMPARATOR?

A comparator is a type of op-amp that compares the voltage on the two input pins, and decides which voltage is the highest. Then the output is changed to reflect that decision. Op-amps can do this, but are designed as analog devices on both input and output. Comparators are required to switch the output.

Comparators have a "-" input, and a "+" input, but either input can have a voltage anywhere between the supply voltages applied to them. The "+" and "-" do not refer to the voltage, but the effect the inputs have on the output.

DEVICE INPUTS	OP-AMP OUTPUT	OC COMPARATOR OUTPUT
+ input < - input	-ve value	Transistor Off
+ input = - input	Analogue	Switching
+ input > - input	+ve value	Transistor On

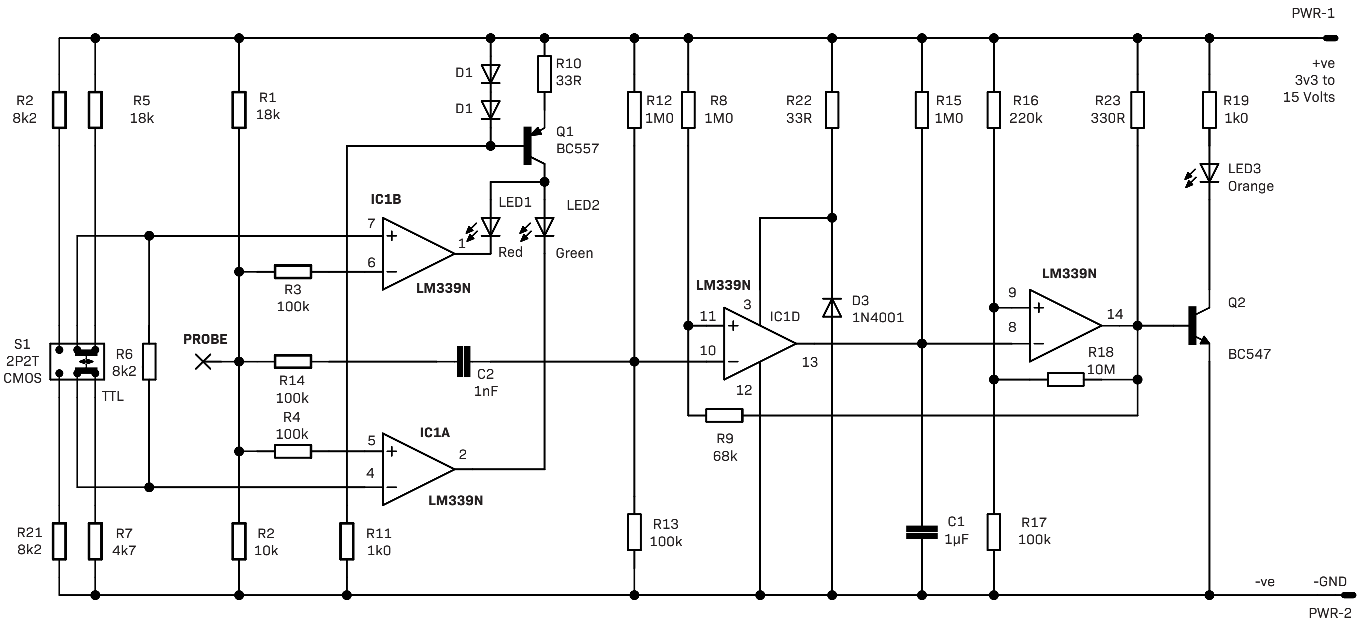
If the "+" is more positive than the "-", the output turns on, but if the "-" is more positive than the "+" the output turns off.

Note that the output itself is a transistor with an open collector (OC), unlike most op-amps. OC simply means there is an NPN transistor with the collector connected to that output pin, and the emitter connected to ground via the power supply pins. Therefore, the output does not go high or low, simply on or off. You will need to connect the load between the OC output pin and a positive supply, which can be a separate supply to the IC supply itself.

LM339 QUAD COMPARATOR

The LM339 has four comparators (hence "quad") in one 14-pin device; a DIP (dual inline package) device – a through-hole device – is my preference, but SMD and other versions are also available. The inputs do not require much current to detect the logic, typically 25nA (yes, that's "nano amps"), and the whole device takes 0.8mA total power, regardless of supply voltage, which can be from 2V to 32V. (Note: that does not include the output loads as they are supplied by a separate power source external to the IC power pins.) Therefore, the LM339 will work on 3.3V and 5V TTL systems, as well as CMOS and some others, (with adjustments to some resistor circuit values if you are keen). The output is compatible with TTL, MOS and CMOS, and the current draw per output is rated at 20mA, which is sufficient for directly driving 3mm LED indicators.

Switching speed is the least impressive of the specifications, but only if you can see and think at almost 2MHz. In fact, the design doesn't require that kind of speed even if the micro runs at 20MHz. The signals on the computer I/O lines will not normally be changing that fast, and pulses can still be detected. I do recommend that



The circuit of the Logic Probe and it's LM339 Quad Comparator core.

when you use any device, you download the datasheet/s and some application notes off the internet, and read them. It is a good way to determine what it all means, and at minimum the datasheets will tell you what pins to use.

THE CIRCUIT

Take a look at the schematic above. Note what the pins are and try to remember them. Particularly note that pin 3 is the positive or VCC connection, 5V in most cases, and pin 12 is ground.

Each comparator has two inputs (+ and -) and one output. Note the polarity of the inputs on either side of the IC. Comparator 1 has its inputs on pins 6 (-) and 7 (+) and its output on pin 1. Then note that comparator 3 is the same on the other side of the IC, except for the polarity of the pins. Once you become familiar with the symmetry of the pin-out, you'll find working with the device quick and easy!

For simplicity, we are going to describe the logic probe in two parts, and we'll start with level detection first.

Beginning with the comparator inputs, there are two voltage dividers at work. The first uses R1 = 18k, and R2 = 10k, to set the logic probe voltage when not connected, to Vcc x (10k/(10k + 18k) volts. When the rail voltage is 5V, the probe will be ~1.78V, or between 0.8V and 2.0V, so neither LED should be lit. The probe voltage is connected to the comparator inputs on pins 5 and 6, via two 100k resistors, R3 and R4. You'll note that another 100k, R14, takes the probe voltage to pin 10 for the pulse circuit. We'll explain that section later.

When the rail voltage is 3.3V, the probe will be ~1.2V, so as before neither LED should be lit, and 3v3 TTL can be tested without issue.

For a CMOS circuit, two resistors of the same value should be used for R1 and R2, but the values used for TTL will also work on CMOS, and no changes are necessary.

The second voltage divider chain using R3, R4 and R5, set the two reference values, 0.8V for "0" level TTL on comparator pin 4, and 2V for "1" level TTL on comparator pin 7. The string is designed for 5V TTL but works on 3v3 TTL with a little error in the absolute threshold values. For standard 5V TTL, R5 = 18k, R6 = 8k2 and R7 = 4k7 respectively giving reference values at pin 4 of 775mV, and 2.04V on pin 7.

The switch allows the user to swap between two references: standard TTL in the forward position, and a second option in the back position. Assuming CMOS is the second option, the resistors should be R20=R6=R21 = 8k2. If you want a more exact reference for working on 3v3 TTL, R6 remains as 8k2, as it is common to both switch positions and both options of CMOS and 3v3 TTL. Use a 10K for R20, and 5k6 for R21, only if you intend the second switch position to be used for 3v3 TTL.

CONSTANT CURRENT LED DRIVE

The LEDs on the outputs of the comparators are on pin 1 ("1") and pin 2 ("0"). To keep the brightness of the LEDs about the same from 3V to 15V, a constant current source has been built using D1, D2, Q1, R10 and R11. R10 limits the current to 20mA, but the supply is shared between LED 1 and LED2 as they do not come on together.

REVERSE SUPPLY VOLTAGE PROTECTION

In the dreaded situation where a technically-minded person connects the red wire to ground and the black wire to 5V (as happens from time to time), R22 and D3 limit the reverse voltage ››

›› across the LM339 to -0.6V, but R22 will start burning out and letting the smoke out; so consider it a sacrificial fused resistor!

THE PULSE CIRCUIT

The pulse detector has a basic function in what it can do, as we wanted to keep the logic probe simple. In reality, it did start to collect more parts than originally intended. The basic circuit configuration is a monostable multi-vibrator (a name that takes up more space than the actual circuit!). Put simply, a brief trigger at the input, causes the output to turn on for a timed period.

We begin at the probe, where we are more interested in the change in voltage than the actual voltage. Therefore R14, another 100k is connected to pin 10 via C2, a 1 nanoFarad capacitor. The capacitor will only pass a current when the voltage is changing.

R12, 1M0 and R13, 100k set up a reference voltage on pin 10 of about 450mV. Comparator pin 11, due to the voltage divider using R8, 1M0, and R9, 68k, has a set value of about 290mV, for reasons we'll get to later in the loop!

The output of comparator 3 remains off in normal conditions. When the pulse of current presents at pin 10, it becomes temporarily more negative than pin 11 and it's output, pin 13 turns on which discharges C1. This turns comparator 4 off, which lifts the voltage on R9, so pin 11 is again higher than pin 10, turning comparator 3 off.

For comparator 4, R16, 220k and R17, 100k set pin 9 to about 1.6V, with some feedback from R18, 10M pulling pin 9 a little lower in normal times, but a little higher when the LED is on. This is positive feedback and serves as a Schmidt trigger, to add hysteresis to the switch.

Pin 8 of comparator 4 has the RC timing circuit using R15, 1M0, and C1, 1uF. The timing constant of this pair is 1 second, for 63% of the supply voltage. However, pin 8 is compared to the voltage on pin 9, ~1.6V, so switches in about one-third of a second. A longer pulse would result if R16 were made 100k, for example.

So what happens to comparator 4, is comparator 3 grounds pin 8, turning the output transistor off.

Transistor Q2 is turned on by bias resistor R23, which has been reduced to 330Ω as Q2 was not turning completely off. It sounds wrong but comparator 4 output wasn't pulling the base of Q2 low enough.

The orange LED, LED3, has a simple ballast resistor, R19, 1k0, although it might have been powered by the constant current circuit, sharing current with the other LEDs.

C1 charges with the current flowing in from R15, until the voltage across C1 reaches the value on pin 9, and as soon as it exceeds that value, comparator 4 turns on, grounding the base of Q2. Everything is back to starting position except that C1 will continue to charge until it reaches rail voltage. The circuit is ready for another pulse.

ASSEMBLY

While this object can be built from proto board or similar, it will quickly end up as a rat's nest, so we recommend using a PCB or kit from one of the supporting suppliers. The PCB is shaped and sized so it will be a natural size to hold for testing a circuit, and the probe tip and LEDs are all at one end, while the power leads and switch are at the other end. So doing it this way means not only will you have somewhere to put the components, but you won't have to guess where the connections go!

HOMEMADE PCB

The PCB was in fact designed as a single-sided board, and those keen enough can download the PCB pattern from the website. Just be sure to use the "single-sided" pattern, which has wider tracks and larger pads for easier hand-drilling. You will need only two links of wire on the top-side of the PCB.

For commercially available boards, double-sided PCBs are often the same price as single-sided boards, so the links are integrated into the PCB already. There's not a whole lot of spare space on the board, so it does make life a little easier.

POWER

We did argue a little – but only between gentlemen – about options for the power supply. While a 9V battery would power it and be reasonably tidy, it also required voltage regulation and special consideration for CMOS above the supply voltage.

In the end any kind of battery was disqualified as unnecessary when the thing is used to test a powered circuit anyway. Using the supply from the "DUT" (device under test) is the norm for logic probes, as they don't require much current anyway.

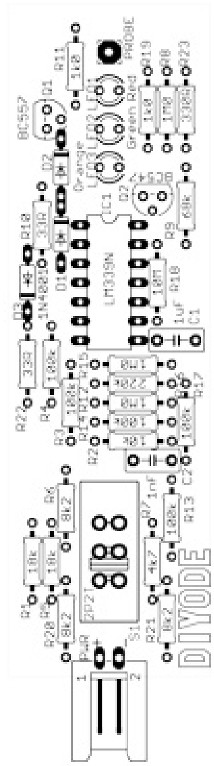
Still there may be options. My own probe had the leads ripped off it once, so I used a speaker cable with a pair of IC clips. When everything was TTL in DIP ICs, I simply clipped onto pins 7 and 14, or whatever the supply pins for the ICs were. I still think that's the way for ICs with legs!

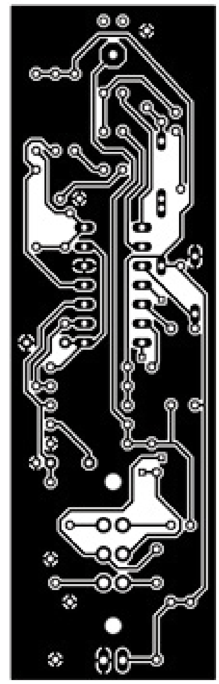
Those who work mainly on Arduino, Raspberry Pi and such, may prefer a pair of male pins or female sockets (PIN-header-style) to go straight onto the supply pins/sockets. Alternatively, you might like to use alligator clips, or even solder the wires onto the DUT power supply. Others may prefer having a USB plug and lead, and even use a USB plug pack. All these options can have the wires soldered directly into the PCB, or place a right-angle two-pin header into the power pads, and plug onto them with a two-way 0.1-inch socket. It all really comes down to what you intend to work on.

REF:	COMPONENTS:	JAYCAR	ALTRONICS
R1	18k Resistor	RR0602	R7588
R2	10k Resistor	RR0596	R7582
R3	100k Resistor	RR0620	R7606
R4	100k Resistor	RR0620	R7606
R5	13k Resistor	RR0599	R7585
R6	8k2 Resistor	RR0594	R7580
R7	4k7 Resistor	RR0588	R7574
R8	1M0 Resistor	RR0644	R7630
R9	68k Resistor	RR0616	R7602
R10	33R Resistor	RR0536	R7522
R11	1k0 Resistor	RR0572	R7558
R12	1M0 Resistor	RR0644	R7630
R13	100k Resistor	RR0620	R7606
R14	100k Resistor	RR0620	R7606
R15	1M0 Resistor	RR0644	R7630
R16	220k Resistor	RR0628	R7614
R17	100k Resistor	RR0620	R7606
R18	10M Resistor	RR0660	R7094
R19	1k0 Resistor	RR0572	R7558
R20	8k2 Resistor	RR0594	R7580
R21	8k2 Resistor	RR0594	R7580
R22	33R Resistor	RR0536	R7522
R23	330R Resistor	RR0560	R7546
C1	1µF Electrolytic 25V	RE6032	R4718
C2	1nF Capacitor MKT 100V	RM7010	R3001B
D1	1N4148 Diode	ZR1100	Z0101
D2	1N4148 Diode	ZR1100	Z0101
D3	1N4004 Diode	ZR1004	Z0109
LED1	Red LED 3mm	ZD0100	Z0700
LED2	Green LED 3mm	ZD0120	Z0701
LED3	Orange LED 3mm	ZD0119	Z0705
Q1	BC557 Transistor PNP	ZT2164	Z1055
Q2	BC547 Transistor PNP	ZT2152	Z1040
IC1	LM339N Quad Comparator	ZL3339	Z2532
S1	DPDT Switch	SS0823	S2060

REF:	OTHER HARDWARE:
PROBE	1-pin eg.Test Probe -or- Canvas Needle
PWR	2-pin Pin Header, 2pin-RA, polarised, (optional)
PWR	2-pin Pin Header, 2pin socket header
PWR-clip	IC-Clip IC Pin Test Clip Red
PWR-clip	IC-Clip IC Pin Test Clip Black
PCB	DIY1709-1 Printed Circuit Board, DIY17091-1
Dims	3.05" x 1.05" (88.9 x 26.7mm) (88.9 X 26.7MM)
Case	DIY1709-1-33D printed case, (see text)

Note: Our part numbers may denote a pack of parts, such as resistors which are often sold in a pack. We have suggested probe-tip hardware and other methods. Kits sold for this project may contain other suitable hardware for this purpose, and may not precisely match this list.





GROUND LEAD

If you intend to use a dedicated power supply, which isn't connected to the DUT, the logic probe requires a ground lead that should be soldered in with the negative power wire, or onto the pad beneath the PCB. Using the DUT power supply avoids the requirement of a ground lead. This convenience is why we used DUT for power.

PROBE TIP

The kind of probe you use is dependent on your needs for the logic probe itself. You might want a good strong probe, or a tiny pin probe for getting into socket headers (for example). With this second user preference in mind, the PCB is made using a largish pad for a wire, with two smaller pads (not connected to ground) for a wire loop, to mechanically support the probe tip.

In the past, my TAFE students have used 1mm^2 (and bigger) electrician's copper wire, copper boat nails, darning needles, MIG welding wire, and many more clever solutions. One little known trick to make soft copper wire become hard, is to beat it with a hammer, rolling it as you go, so it remains basically round.

The hole in the end of the 3D printed case allows you to use a probe from an old multimeter lead, or a new one if you like. Sometimes it's handy to leave the probe connected to a wire, pin or circuit, so you can work hands-free. A short lead with an IC clip on one end and an alligator clip on the other can also be handy. Be inventive! ››

›› PCB ASSEMBLY

The usual rule for manual assembly is “hardware, passive, active”, meaning that you fit sockets, switches, and pins first. If you do use a two-pin header or whatever, or fit the power now, you can perform tests as you go, to make sure you have it all in the right places and soldered correctly.

So with the switch, probe and power soldered in, connect it up and check that you have 5V as expected, and where expected. It is a good practice to assemble in stages and test as you go, rather than loading up every component and soldering everything in one go.

PASSIVE COMPONENTS

The resistors and capacitors come next, and a suggestion may be useful. If you think you might like to replace components at some stage, solder the component leads and cut them off close to the PCB. If you need to change one, it will be easier to get out.

The alternative, used in the industry and military, is to cut leads to about 1mm, bend them over, either flat against the PCB or at about 45 degrees, depending on just how harsh the mechanical forces on the PCB may be, and solder to cover the component pin as well as making the joint. This is a devil to remove without dedicated rework stations.

Make sure each joint is clean and shiny, and there are no “inclusions” (i.e., bits of nasty things unexpectedly stuck in the solder).

Out with the multimeter, and power up the circuit. There is a table provided with all the voltages to expect on the IC pads, for when the correct components are in the correct places, and are correctly soldered.

Compare your values to those in the table shown below.

ACTIVE COMPONENTS

Active components are the most sensitive to heat and static electricity, so they should be left until last.

Place and solder the diodes and transistors, as well as the LEDs, but leave the IC until later. Be very careful with the active components – make sure you have the polarity correct. Check all the connections as above, and then check the voltages.

Use a small piece of wire to short the pads of pins 1, 2 and 14, to the ground, pin 12. Each of the LEDs should light: LED 1 and LED 2 when the short is in place, and LED 3 when it is not in place. This has also tested the constant current transistor Q1, and transistor Q2.

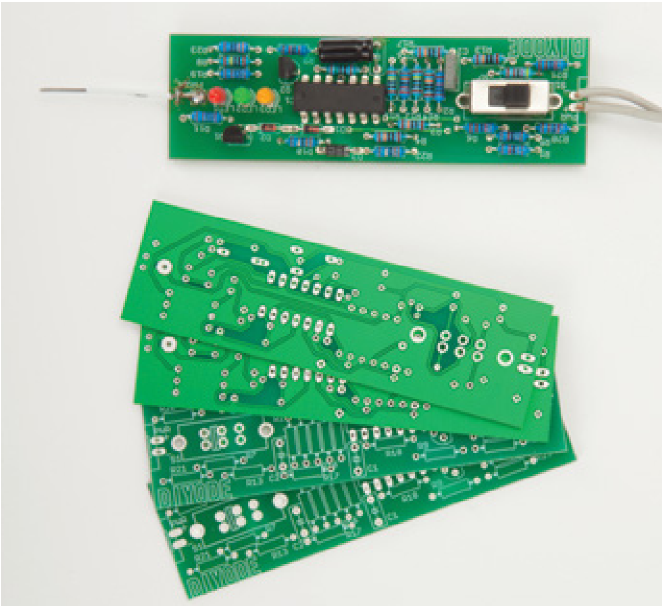
LM339 PLACEMENT

The LM339 quad comparator IC is fairly robust, and the PCB has some basic protection against reverse voltage, but if you are unsure about your soldering, you might like to use a low profile IC socket. This way you can make sure your soldering is clean and tidy, no shorts, whiskers, or dry joints, and then fit the 14-pin IC. The dimple near pin 1 (i.e., the little curved bit taken out of one end), goes where the little bite is taken out of the overlay print. Don’t laugh – even us old hands do it wrong sometimes!

FINALLY!

The big moment has arrived, and if you have passed all the tests so far, you should be looking at success. Power up, and nothing should happen – especially, no smoke. With a piece of jumper wire attached to the probe tip, touch pin 3 and the red LED should light. The orange LED should also come on briefly, about one-third of a second. Touch the ground, pin 12, and the green LED should come on. Note: I could have told you to put the switch in the forward position, but it doesn’t really matter if we’re looking for a good ground or positive rail.

CONDITION	NO IC	PROBE FLOATING	PROBE HI	PROBE LOW	PULSE LED OFF	PULSE LED ON
Pin 1	0.002	3.48	2	2.82	3.48	3.48
Pin 2	0	3.48	2.6	1.96	3.48	3.48
Pin 3	5.16	5.13				
Pin 4	0.795					
Pin 5	1.83	1.83	5.11	0.004	1.83	1.83
Pin 6	1.83	1.83	5.11	0.004	1.83	1.83
Pin 7	2.159					
Pin 8	4.7	1.586	1.586	1.575	1.586	0
Pin 9	1.84	1.61				
Pin 10	0.47	0.47	0.47	0.47	0.47	0.47
Pin 11	5.13	0.938	0.938	0.938	0.938	0.938
Pin 12	0	0				
Pin 13	4.7	1.587	1.586	1.586	1.586	0
Pin 14	5.16	0.651	0.29	0.29	0.29	0.29



LEVEL TESTING FIELD

If you really want to, you can check your actual levels by putting a potentiometer between ground and 5V, and the probe on the wiper pin, and adjust until the green “0” LED just goes out. Read, and record the voltage, and then adjust until it just turns on; then read and record again, and see how much hysteresis there is. Note that no hysteresis is actually designed into the circuit, on the levels at least.

Likewise, adjust until the red “1” LED just comes on, read the voltage and record, and then adjust it until it just goes off; then read and record again. The values should not be very different, and should be less than 0.8V for a logic “0”, and more than 2V for a logic “1”; this is with the switch in the TTL position.

You can try it also in the CMOS position, and for different voltage supplies up to 15V. Low should be about one-third supply voltage, and high should be about two-thirds supply voltage.

3D PRINTED CASE

We have designed a handy case for mounting the probe, so it can easily become part of your toolkit.

The case has two snap-ins to hold the PCB in place, while the case halves snap together for a neat fit. The case may want to stay that way, so make sure you’re ready to leave it together, or perhaps sand the snap-ins so it goes together easily, and therefore comes apart easily. If the PCB is tight, it forces the sides of the case out, so make sure the PCB fits well before tackling the lid. Checking you have no protrusions from your PCB will help ensure a neat fit too.

It could be perpetually fine tuned, but PCB manufacturers will vary, and some of the “fit” depends on the 3D printer and how it is set up. The original case was designed in OpenSCAD, and we've provided the SCAD file to make any changes relatively easily too. For those of you using Tinkercad or similar software, you can easily import the STL file and add/remove shapes from the core design too, though some modifications are much easier to make in OpenSCAD as it can re-render the shapes from scratch.

USING YOUR NEW PROBE

Obviously one caveat we must make is to never use this logic probe on voltages above 15V, or any device that runs on live ac current. IT IS NOT CatIII-rated, and isn’t meant to be. It's meant for TTL / logic levels only.

Now that we’ve clarified that point, take a probe around a running Arduino or such. Using the DUT’s power supply avoids requiring a ground wire, and the probe has enough input resistance to be kind to your circuits.

If you strike a test point that doesn’t light, or lights intermittently either high or low, but doesn’t hold at least one LED on, then it is probably spending a lot of time in between a logic high and a logic low, or has a high impedance. Either way, you may have found your fault!

Logic probes take a little learning before you really know how to use one well, but once you get your proficiency badge, you’ll always have that skill in your kit. ››

›› ADVANCED TESTING - ARDUINO

If this is the first time you've used a logic probe, some advanced testing might be worthwhile to fully appreciate the relationship between the logic levels in your code (or created by an integrated circuit), and the actual levels and voltages presented in a circuit.

For this reason, we have created a basic Arduino Sketch, which you can use as a powerful test tool. It will also put your probe through its paces in an Arduino specific environment.

The sketch is quite simple, and you can elaborate on it further to test all pins. This is also a valuable education exercise.

```
int lowPin = 5;
int highPin = 6;
int pwmPin = 9;
void setup() {
   // set these pins as outputs
   pinMode(highPin, OUTPUT);
   pinMode(lowPin, OUTPUT);
}
void loop() {
   digitalWrite(highPin, HIGH);
   digitalWrite(lowPin, LOW);
   analogWrite(pwmPin, 10);
}
```

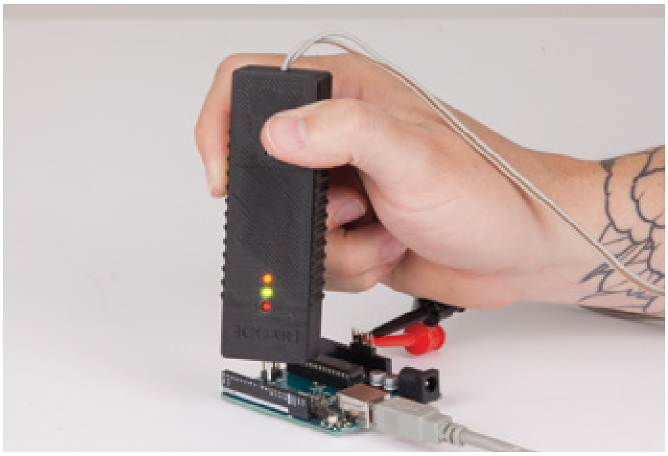
What we have effectively done here is quite simply, set pin 5 to LOW, pin 6 to HIGH, and pin 9 (which is a PWM-enabled hardware pin) to a low value.

You may notice that in the setup function, we have specified our high and low logic pins as OUTPUT, but not our PWM pin. This is because the analogWrite function does this automatically.

It's important to remember to use the pinMode function in setup for any standard digital outputs.

We have also included this sketch as probe_test.ino in the digital resources too. Load the sketch onto your Arduino.

Connect power for your probe to the 5V and GND pins on your Arduino. While you should have performed basic tests on your probe already, if you want to be sure your probe is working, you can touch the probe tip onto the 5V or GND. You should see the red or green LEDs illuminate respectively. As long as this occurs, move on to the pins we have specified in the sketch.



LOW: Touch the probe tip to pin 5, which we have set with a LOW output. You should see the green LED illuminate.

HIGH: Touch the probe tip to pin 6, which we have set with a HIGH output. You should see the red LED illuminate.

PULSE: Touch the probe tip to pin 9, which we have set with a PWM output.

You may also notice that the green LED is illuminated. This is because we have set it to around 4% duty-cycle (more accurately, 10/255, or 0.3921... but who's counting). This is because the probe features some great functionality. When a pulse width output is detected, the red and green LEDs act as a duty-cycle reference. When the duty cycle is low (that is, off more often than on), the green LED will illuminate alongside the yellow pulse indicator.

```
int ledPin = 9;
void setup() {
   // nothing happens in setup
}
void loop() {
   // fade in from min to max in increments of 5
   for (int fade = 0 ; fade <= 255; fade += 5) {
      // sets the value (range from 0 to 255):
      analogWrite(ledPin, fade);
      // wait 300ms to see the dimming effect
     delay(300);
  }
   // fade out from max to min in increments of 5
   for (int fade = 255 ; fade >= 0; fade -= 5) {
      // sets the value (range from 0 to 255):
      analogWrite(ledPin, fade);
      // wait 300ms to see the dimming effect
      delay(300);
  }
}
```

Likewise when the duty cycle is high (that is, on more often than off), the red LED will illuminate alongside the yellow pulse indicator. Of course, when the pulse width is somewhere in between, you will see all three LEDs illuminated.

Component tolerances will affect the precise points where the high and low LEDs switch on and off when the probe is in pulse mode. However, to demonstrate this, we can utilise a modified version of the FADING example sketch included with the Arduino IDE. Ours is not too different from the example provided with the software, however we have slowed the cycle down considerably so it's more visible and you can watch it go through the full cycle from 0% to 100% duty cycle, and back again. This sketch is provided in the resources as probe_pwm_cycle.ino. Load it onto your UNO and touch the probe to pin 9, watching what happens!

WHAT ABOUT A RASPBERRY PI?

The beauty of logic is that it's, well - logical. The process for coding a similar test on Raspberry Pi is fairly similar. But we've done it for you, providing the sample Python scripts in the resources.

Of course, you still need to tell the Raspberry Pi what you're doing, and set the various pins to have their respective outputs.

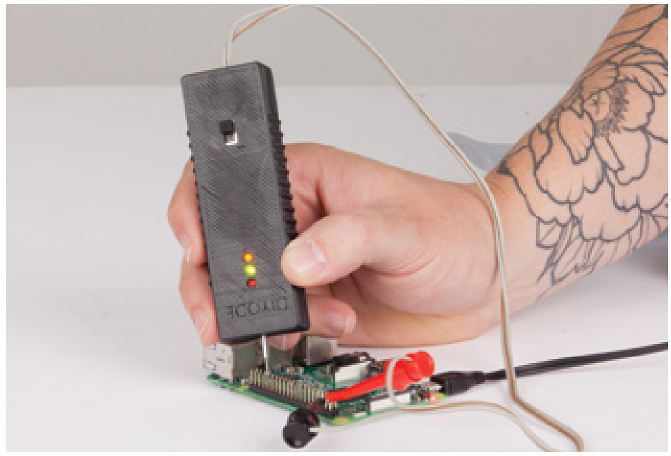
Here is a basic Python code to achieve something fairly similar to probe_test.ino, but it's Python so we'll call it probe_test.py (i know, we're very creative).

Load probe_test.py in IDLE and run. Since Raspberry Pi pins are unlabelled, it's easiest to use a GPIO breakout adaptor which provides full labels on each of the pins.

Again, you can verify your probe is powered and working properly by touching the tip to any of the GND or 5V pins on the Pi, and looking for the green or red lights respectively.

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
lowPin = 5
highPin = 6
pwmPin = 13
GPIO.setup(lowPin, GPIO.OUT)  # set up pin 17
GPIO.setup(highPin, GPIO.OUT)  # set up pin 18
PWM1 = GPIO.PWM(13, 100) # enable software PWM
GPIO.output(lowPin, 0)  # send pin 5 low
GPIO.output(highPin, 1)  # send pin 6 high
PWM1.start(10)     # start PWM at 10%
```

As in the Arduino examples, we'll also provide an example for a cycling PWM output so you can see the effect it has on the probe. Load and run probe_pwm_cycle.py. Pin 13 should start looping



through the cycle taking it from 0% duty cycle to 100% duty cycle, and continue on forever (well, until you forcibly quit the programme).

The interesting thing about Raspberry Pi PWM is that there's only two hardware PWM channels. Most PWM functionality is derived from software PWM via the GPIO library. For most purposes, it's perfectly adequate and opens up far more functionality than would otherwise be provided for this purpose.

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
PWM1 = GPIO.PWM(13, 100) # enable PWM at 100Hz
PWM1.start(0)      # start PWM at 0%
while 1:
   for level in range(0, 101, 5):
      p.ChangeDutyCycle(level)
      time.sleep(0.5)
   for level in range(100, -1, -5):
      p.ChangeDutyCycle(level)
      time.sleep(0.5)
```

So there you have it! Go forth and conquer logic with your new logic probe, and newfound understanding of how it works! ■

WANT MORE?

For additonal code, or to discuss this project, visit:
https://diyode.io/004tdqg

PLANE SPOTTING

DAVID SUCH

Engineer, Honorary Associate at USyd.

www.reefwing.com.au

@reefwing

Did you know you can track planes in the sky using free software? David does. And he's doing it!

There's a fantastic community of enthusiastic plane-trackers out there. David is part of that community. He won one of our prelaunch competitions, finding himself the proud owner of a pcDuino and Touchscreen! He's put it to great use.

Congratulations on being the lucky winner of our pcDuino competition. How did you feel when you found out?

I was very excited! I think it is the first competition I have ever won. I'd been tempted to give the pcDuino a try for a long time, but the cost had been a barrier, especially given you can get similar functionality with a Raspberry Pi (RPi) and Arduino. The price has recently come down but free swag is always a treat! I even had a useful project lined up that I thought would allow me to see how it performed.

We're so happy it went to a good home and we see you've put it to good use! For those who haven't heard, can you explain what you're doing with the pcDuino?

I am using the pcDuino and a cheap USB DVB-T TV Tuner to track aircraft flying nearby, and then displaying their real time position on Google Maps. In addition to the plane's position you can also see their flight number, transponder squawk code, altitude, speed and heading. Squawk codes are used by Air Traffic Control (ATC) to uniquely identify aircraft on radar; they are also used in emergency (for example, if a plane is squawking 7500 it means that the plane has been hijacked!).

In order to do this the pcDuino needs to receive and interpret the ADS-B data that most planes transmit from their transponder. A transponder is a piece of kit on a plane to help ATC identify a particular aircraft's position and altitude on a radar screen. This helps them maintain separation and prevent collisions. There are three civilian transponder modes, called A, C and S.

MODE A: Following an interrogation request, the transponder broadcasts the configured transponder code (or "squawk code"). A separate type of response called "Ident" can be initiated from the aircraft by pressing a button on the transponder control panel.

MODE C: This will send a pressure altitude response when interrogated.

MODE S: This "Select" mode is to avoid over interrogation of the transponder (if there are many radars in busy areas), and to enable automatic collision avoidance. Mode S transponders are compatible with Mode A and Mode C. A Mode S transponder is required to implement ADS-B, but there are other ways as well.

ADS-B stands for "Automatic Dependent Surveillance - Broadcast", which isn't really very helpful. In other words, the plane uses GPS to obtain its position it then broadcasts that position at rapid intervals, along with identity, altitude, velocity and other data. Dedicated ADS-B ground stations receive the broadcasts and relay the information to air traffic control for precise tracking of the aircraft. The pcDuino and TV Tuner are, in effect, another ADS-B ground station.

ADS-B data is broadcast every half-second on 1090MHz. The ability of a ground station to receive a signal depends on altitude, distance from the site, and obstructing terrain. The maximum range of each ground station can exceed 250 nautical miles; however, the dinky aerial which comes with the TV Tuner is not going to deliver that sort of range, particularly if it is indoors like mine. In practice all the plane data that I am receiving is within 80km. If you connect your base station to Flight Aware, it will provide you with these statistics, as well as much more.

In Australia, CASA is moving towards most aircraft requiring a Mode S transponder, which is ADS-B capable, but this will take years to come into effect, particularly for Visual Flight Rules (VFR) aircraft. All of the big jets certainly have it. You may see references to S-mode when reading about ADS-B plane tracking and sometimes S-mode is used interchangeably with ADS-B (which is not strictly correct).

TOP ▲
David's completed setup.

ABOVE ▲
David won a pcDuino (XC-4350) and compatible 7" touchscreen (XC-4356) in our prelaunch competition, thanks to Jaycar Electronics.

Interesting information! How do you pick up ADS-B signals using a DVB TV tuner? Is there a frequency crossover?

The RTL2832U-based Software Defined Radio (SDR) receiver is designed and marketed for DVB-T reception. However, it's possible to get raw samples from the device, rather than just a demodulated DVB signal. This means that a wide-band receiver can then be implemented in software. I used the USB DVB-T TV Tuner RTL2832U + R820T (you can get them on eBay for around $13). ››

›› The RTL2832U chip is generally paired with a tuner IC and in the case of the USB receiver above, it's an R820T, which enables reception from 24MHz to 1,850MHz. Remember we need 1090 MHz.

The antenna that comes with the USB Tuner is 100mm in length. This works okay, but is not optimised for receiving signals with a frequency of 1090 MHz. If you are having problems with reception, you could purchase the FlightAware 1090MHz antenna, or there are plenty of home brew designs online. The antenna should ideally be located outside, but I stuck mine near a window and it works okay.

Installation of the hardware is simple - just plug the SDR receiver into a USB port. As there is only one USB port on the pcDuino, you will probably need a hub so that you can also plug in the keyboard and mouse required for setup. I had a powered hub but it doesn't seem to need it. I think the pcDuino supplies enough current for an unpowered USB hub to work. Don't plug in the SDR receiver until you load the new driver. The driver we need is called RTL-SDR. To download it, fire up the pcDuino, open LXTerminal and type the following (assuming you're in ~/):

```
$ sudo git clone git://git.osmocom.org/rtl-
sdr.git
```

To install RTL-SDR:

```
$ cd rtl-sdr/
$ sudo mkdir build
$ cd build
$ sudo cmake ../
$ sudo make
$ sudo make install
$ cd ~
$ sudo cp ~/rtl-sdr/rtl-sdr.rules /etc/udev/
rules.d/
$ sudo ldconfig
```

The kernel that comes with Ubuntu already contains a DVB driver, which we don't want to use. To stop the conflicting Linux DVB-T driver from loading, we need to blacklist it. To do this:

```
$ cd /etc/modprobe.d/
$ sudo nano blacklist.conf
```

Then add the following line at the end of the file. You can either use the existing blacklist.conf file or create a new file; just make sure it's in this directory and ends in .conf.

```
blacklist dvb_usb_rtl28xxu
```

At this point you can plug in your receiver dongle and reboot. Rebooting will allow the blacklisting to take effect.

Thanks for the step-by-step instructions. We're confident there'll be a few doing this! You're using Dump1090 software. Is that what does the decoding, or is there another step in the middle?

The RTL-SDR driver gives us access to the ADS-B data and passes it to Dump1090, which does the heavy lifting. Dump1090 is a Mode S decoder that is specifically designed for RTL SDR devices. It provides robust decoding of weak messages, and an embedded HTTP server that displays the currently detected aircraft on Google Maps. The 1090 refers to the 1090MHz frequency that the signals are broadcast on.

Dump1090_mr is a FlightAware fork, of Malcolm Robb's fork, of Salvatore Sanfilippo's dump1090 program. FlightAware uses it as an important element of PiAware (a Debian package for forwarding ADS-B data to FlightAware - more on this later).

Note that dump1090_mr is no longer available on the FlightAware GitHub repository, so I used a copy sourced from CTassisF. Before installing Dump1090 there are some pre-requisite packages and libraries required. To install these:

```
$ sudo apt-get install git cmake libboost-
all-dev libusb-1.0-0-dev python-scitools
portaudio19-dev -y
$ sudo apt-get install tcl8.5-dev tclx8.4-dev
itcl3-dev tcl-tls tcllib automake cmake telnet
git gcc make
```

To build and install dump1090 and faup1090, and to configure the system to start them automatically whenever the system boots, change to the ~/ directory and type:

```
$ git clone https://github.com/CTassisF/
dump1090_mr.git
$ cd dump1090_mr/
$ make
$ make -f makefaup1090 all
$ sudo make -f makefaup1090 full-install
```

To test whether dump1090_mr is operational:

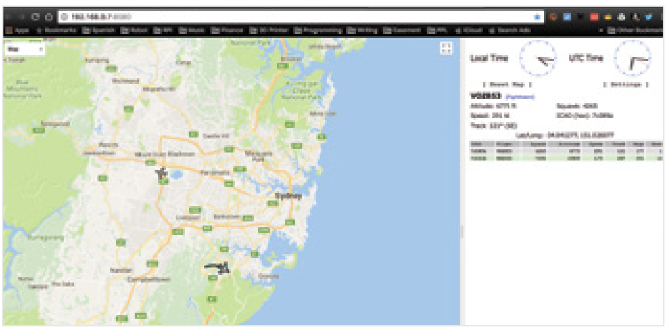
```
$ dump1090 --interactive
```

This should show you a list of detected planes along with their flight information.

That's awesome! How is the flight data then overlaid to Google Maps?

This is very easy as Dump1090 does it for you, and it includes an embedded HTTP server. If you are on the pcDuino, open the Chromium browser and point it at http://127.0.0.1:8080 to see the display.

You can also view this page from other computers on your LAN. To do this you will need the IP address of your pcDuino (type hostname -I on the command line). On the other computer you can then access the page via that IP address. So for example, my pcDuino IP address was 192.168.0.7 to view the dump1090 page I enter: http://192.168.0.7:8080 into the browser address bar.



Fantastic! Tell us a little about FlightAware and how the community works.

FlightAware is the world's largest flight tracking data company. By registering (for free), you can add your ADS-B base station results to the worldwide collection of other operators, creating a real time picture of aircraft movements globally. If you are picking someone up from the airport you can track their flight in real time, and minimise your parking fees! Connecting to FlightAware will also tabulate and graph statistics from your station.

You mentioned it's been done on RPi many times, were there any challenges getting it to work on the pcDuino?

Yes. FlightAware has made it very easy to get this system up and running on a RPi, and have a guide available on their website.

The difficulty is that the packages have been compiled for Raspbian (a Debian variant of Linux, which is most commonly used with the RPi), and the pcDuino uses Lubuntu (a light weight variant of Ubuntu). Fortunately, Debian and Ubuntu share a common ancestry so it is possible to move packages between the two.

PiAware is a FlightAware client program that runs on a RPi (or pcDuino!) to securely transmit dump1090 ADS-B and Mode S data to FlightAware.

I had some trouble getting PiAware to work on the pcDuino, but the following ended up being the best way to build the system for Ubuntu (this worked for version 3.5.1 of PiAware). To create the PiAware package, change to the ~/ directory and type:

```
$ sudo apt-get install build-essential
debhelper librtlsdr-dev libusb-1.0-0-dev pkg-
config tcl8.5-dev autoconf python3-dev python-
virtualenv libz-dev git tclx8.4 tcllib tcl-tls
itcl3
```

This will load some libraries you will need. You can then download and build PiAware. ››

```
$ git clone https://github.com/flightaware/
piaware_builder.git
$ cd ~/piaware_builder
$ ./sensible-build.sh wheezy
$ cd ~/piaware_builder/package-wheezy
$ dpkg-buildpackage -b
```

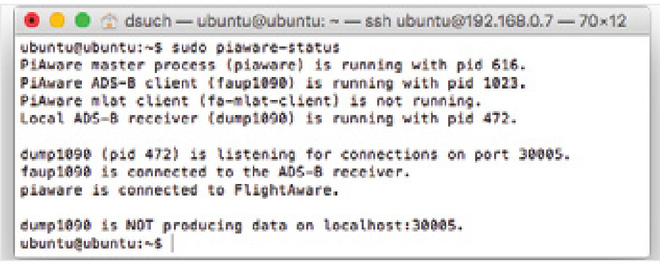
›› To install the PiAware package:

```
$ sudo dpkg -i ~/piaware_builder/
piaware_3.5.1_armhf.deb
```

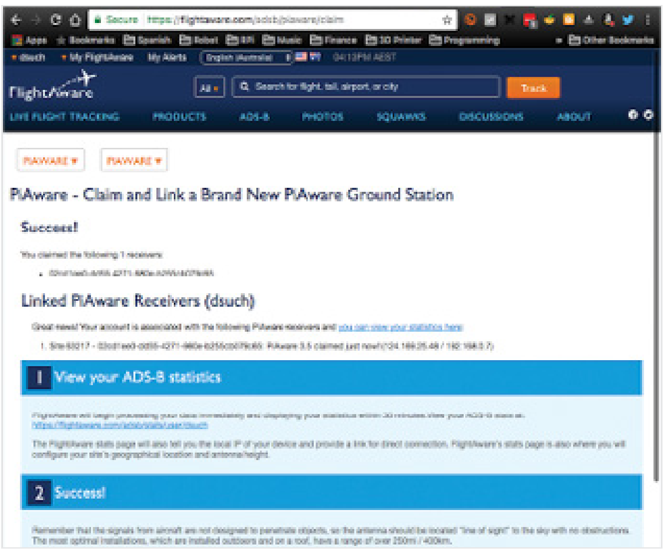
Reboot, and then to check that everything is working as it should, type:

```
$ sudo piaware-status
```

This should display something like the screenshot below. Note that you no longer need to add your username and password to the PiAware configuration file.



To link your new pcDuino ADS-B base station to your FlightAware account, log into the FlightAware website, and then go to the claim page to claim a new station.



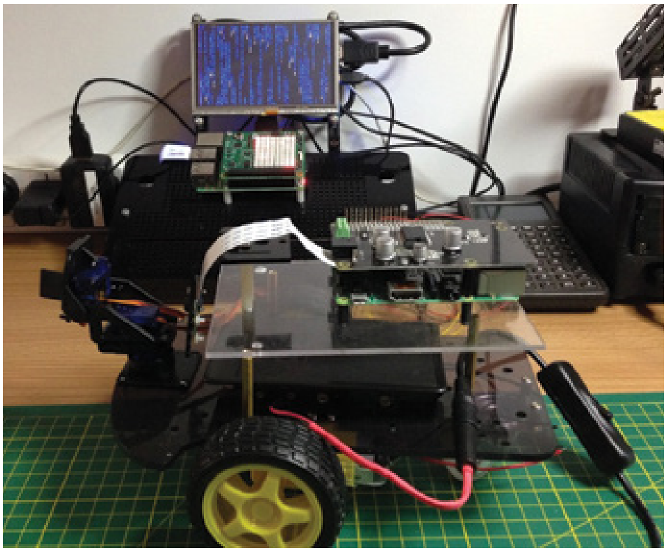
FlightAware works out which base station is yours, based on your IP address, so the computer you are using to claim the station must be on the same LAN as your base station; and the station needs to be transmitting data to FlightAware. Wait 15 minutes or so after booting the pcDuino to make sure that FlightAware is receiving your data, before trying to claim your station.

It sounds like an amazing community collating data together! Is there anything else you'd like to tell us about this system?

Just to note that if you do contribute your ADS-B data to the worldwide network, then Flight Aware provides access to the following:

- Live data on *flightaware.com* (subject to standard data processing delay of up to two minutes).
- Access to up-to-the-second live data received by the local device (accessible from the stats page with a local network connection).
- Data from local device highlighted on FlightAware track logs.
- Detailed statistics on site performance.
- A free Enterprise Account (USD89.95/mo value).

Wow, it sounds like a great deal all-round! What are you working on now?



A RPi-based robot, which looks a lot like SAM from Issue 3. Currently you can drive it around using a web browser, while viewing the onboard video, and I am working on an Amazon Alexa interface so that it will respond to voice commands.

That sounds awesome - we can't wait to hear about these in the future! ■

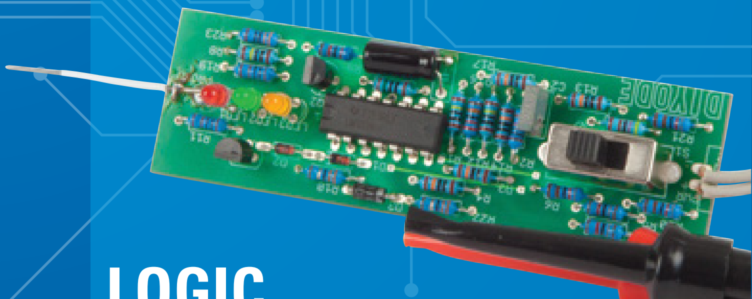
GOT SOMETHING TO SAY?

To discuss this feature, visit:
https://diyode.io/004sxpg

Reading & Resources:

▶ **FLIGHTAWARE** *www.flightaware.com*


CONNECT,
COMBINE,
CREATE!
NEW AND POPULAR MODULES & SHIELDS FOR YOUR NEXT INCREDIBLE PROJECT
$29.95
DISPLAY IT
A tiny monochrome 128x64 pixel display. Only needs 2 pins for control. XC-4384
$9.95
ADD STORAGE
SD Card interface. Works with the Arduino IDE's inbuilt SD library. 3.3V & 5V compatible. XC-4386
$29.95
COMMUNICATE
Bluetooth® V4.0 BLE module. Configurable as master or slave. XC-4382
$9.95
ADD COLOUR
RGB LED strip. Daisy chain up to 1000 LEDs run from one pin. 5V. 54mm long. XC-4380
$39.95
NETWORK
Ethernet expansion for wired networking. Get your Arduino online. XC-4412
$39.95
CONNECT A USB DEVICE
Connect to USB devices such as keyboard, mouses & even mobile phones. XC-4456
FROM $12.95
SWITCH IT ON
1 CHANNEL 5VDC XC-4419 $5.45
4 CHANNEL 12VDC XC-4440 $12.95
8 CHANNEL 12VDC XC-4418 $19.95
$19.95
SECURE ACCESS
RFID Card reader module with card and fob for testing. See what's on the smart cards in your pocket! XC-4506
FROM $3.95
GIVE OTHERS ACCESS
13.56MHz RFID NFC. Suits XC4506, XC4542 and some Android phones.
CARD ZZ-8978 $3.95
FOB ZZ-8970 $4.95
PROTOTYPE WITH POWER
Breadboard with removable power supply. Power from USB. 3V & 5V switchable. PB-8819
$19.95
DESIGNED FOR BREADBOARDS
70 stripped pieces of single core sturdy wire. 5pc x 14 different lengths. PB-8850
$13.50
WIRE IT UP!
A pack of 40 jumper leads of various colours for prototyping.
PLUG TO PLUG WC-6024
SOCKET TO SOCKET WC-6026
PLUG TO SOCKET WC-6028
$5.95 ea
ARDUINO STACKABLE HEADER
Build a stackable shield, or make your current shield stackable. Alternatively, shorten the pins to make female headers just like the Duinotech main boards. HM-3208
$4.50
NERD PERKS CLUB MEMBERS RECEIVE:
$25 JAYCOINS CARD
WITH EVERY $500 SPENT*
NERD PERKS
REGISTER ONLINE TODAY:
WWW.JAYCAR.COM.AU/NERDPERKS
jaycarelectronics
@JaycarAU
@jaycar_electronics
Call Us 1800 022 888
*Terms and conditions apply.



LOGIC PROBE KIT

You've read the DIYODE project on page 58, now you can get the kit from Jaycar. Kit supplied with PCB and all specified parts. You'll need to print your own case though.

KD-6100

Available early Oct.

$14.95

FOR OVER 40 EXCITING ARDUINO-BASED PROJECTS VISIT www.jaycar.com.au/arduino

BUILD AND SAVE: VOICE CONTROLLED POWER POINT

As described on page 76 of this issue of DIYODE, you can build something very similar to the Google® Home device using a combination of RPi and Arduino. We've made it easy and bundled the products together so you can save compared to buying the components separately.

RPI VOICE RECOGNISER BUNDLE

VALUED AT OVER $150

- RPI 3 XC-9000 $74.95
- USB TO AUDIO CONVERTER XC-4953 $24.95
- TIE CLASP MIC AM-4092 $19.95
- USB SPEAKER XC-5191 $14.95
- 16GB MICROSD CARD XC-4989 $19.95

$129

SAVE $25

POWER POINT CONTROL BUNDLE

VALUED AT OVER $110

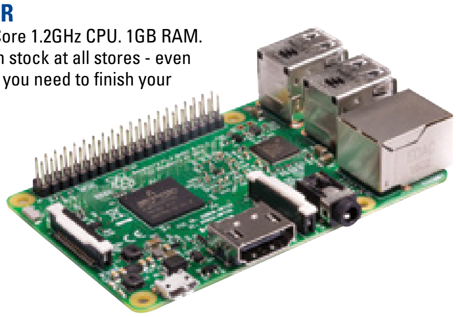
- LEONARDO BOARD XC-4430 $29.95
- PROTOTYPING SHIELD XC-4482 $15.95
- ESP-13 WI-FI SHIELD XC-4614 $34.95
- 433MHZ TX MODULE ZW-3100 $13.95
- WIRELESS OUTLET MS-6148 $19.95

$79.95

SAVE $30

GET PI AT JAYCAR

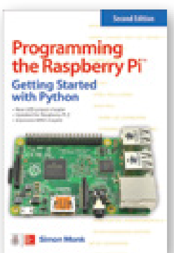
Raspberry Pi 3 Quad-Core 1.2GHz CPU. 1GB RAM. Wi-Fi and Bluetooth. In stock at all stores - even on the weekend when you need to finish your project! XC-9000



$74.95

GETTING STARTED WITH PYTHON

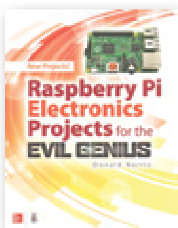
Programming the Raspberry Pi 2nd Edition. Softcover. 188+ pages. 230x150mm. BM-7160



$29.95

FOR THE EVIL GENIUS

Raspberry Pi Electronics Projects. Softcover. 280+ pages. 275x150mm. BM-7162

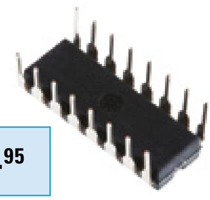


$49.95

$34.95

TINY WI-FI CHIP

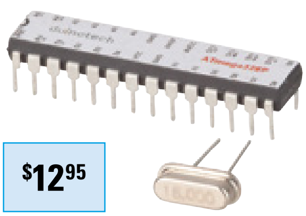
ESP-13 Wi-Fi Shield. ESP8266 IC. Integrated TCP/IP stack. AT command interface. Can be programmed directly with Arduino IDE. XC-4614



$4.95

DRIVE IT

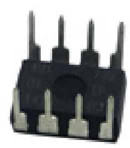
L293D quad half H-bridge motor driver. Four high power outputs to drive a 1 x stepper or 2 x DC motors. 3.3V or 5V. ZK-8880



$12.95

BUILD YOUR OWN

Bare ATMEGA328P micro with 16MHz Crystal. Great for prototyping. ZZ-8727



$49.95

ATTINY85

8 pin DIP micro. 20 MIPS at 20 MHz. 6 GPIO pins. 3 & 5V operation. ZZ-8721

Over 150 Arduino-Compatible Boards, Modules & Shields available.

www.jaycar.com.au

Jaycar Electronics

Prices shown in $AUD. Savings off original RRP. Valid to 31 Oct, 2017.

PROJECT

MQTT Voice Controlled Power Point

Turn on and off a mains appliance using voice commands? Yes, you can do that. - by Tim Blythman

You've probably seen the new Google Home® devices that are now available, but what you might not realise is that you can actually build something very similar with a Raspberry Pi (RPi). And not only that, it's very easy to interface it to your own hardware, such as a remote controlled power point.

The Google Voice Kit (see *https://aiyprojects.withgoogle.com/voice*) is a popular item, but you can achieve the same results with a RPi and a few other parts. This project however, is more than just a DIY version of the Voice Kit. Building on September's MQTT Light Switch project, I've built this project which can take voice commands and then transmit them via MQTT to an Arduino. In turn, this operates a remote controlled power point. While this sounds like an elaborate solution to a problem, the MQTT interface means that we can incorporate other features (such as the app control from the MQTT Light Switch) to expand the functionality even further. And of course, it's just plain cool to be able to speak a command and have it happen!

THE BROAD OVERVIEW

Just like the MQTT Light Switch, there are three parts to this project; and not surprisingly, they are all fairly interchangeable with the different parts of that project. One device (the publisher in the MQTT system) issues commands, which are relayed by a broker to a subscriber, which acts upon the command. In this case, the RPi running the Google Assistant voice recognition software creates messages, based on spoken commands which are transferred via the broker software to an Arduino. This then sends a radio signal via a 433MHz transmitter to a wireless power point to control an appliance.

While this is just one use of the RPi with voice recognition software, the options are really interesting. When set up correctly, the Google Assistant has many different skills. It won't be as capable as a Google Home® device due to hardware limitations, but it can respond to simple queries such as "Where's the nearest pub?". It can also be configured to do pretty much anything that can be done via a terminal command (and more), so the possibilities are well beyond what is mentioned in this article.

Of course, if you just want a really basic app-controlled power point, you can use an MQTT app for input to control a wireless power point, and it won't even interfere with the normal remote control operation. For more information on MQTT, revisit the article in Issue 3 of DIYODE Magazine.

HOW IT WORKS

As you might imagine, a very big part of this project is setting up and configuring the RPi to do the voice recognition. There is an SD card image, which is designed to work with the Google Voice HAT, but the hat appears to be hard to obtain. Fortunately, it's not hard to replace the hardware with something equivalent; in this case, a USB sound card connected to a microphone and speakers, as well as an LED for troubleshooting. There's some account setup (you'll need a Google account to be able to access Google's services), and finally some further configuring to get our specific commands working. There's also an extra step to set up the "OK Google" hotword, if you wish to use it.

The MQTT system requires a broker, and this is probably the easiest part to install. It can actually be installed in just two commands on a RPi, and this is the most obvious place to install it (although you might want to wait until we've created our SD card image).

```
sudo apt-get update
sudo apt-get install mosquitto mosquitto-clients
```

In addition to this, it is recommended to give the Pi a fixed IP address under your router, so that any apps can easily find it. Look under DHCP setting for DHCP address reservation options.

The other piece of our setup is an Arduino running a sketch that is similar to that in the MQTT Light Switch project; it looks for MQTT message packets, and then acts upon them. In this case, the Arduino is connected to a 433MHz transmitter and emulates the signal from a remote control, which operates a Jaycar MS-6148 Wireless Power Point. The code emulates the original product's 433MHz commands to the wireless power point, providing us with electronic-controlled mains power. The beauty of this hardware is that it takes wireless signal in and sends a different wireless signal out, and so it could be hidden just about anywhere.

For the voice recognition section of the build, I used the following parts:

PARTS REQUIRED:	JAYCAR	ALTRONICS
1 x Raspberry Pi 3	XC9000	Z6302B
1 x USB Sound Card	XC4953	D0290
1 x 3.5mm Microphone	AM4092	C0398
1 x USB Speakers	XC5191	D0806A
1 x 16GB Micro SD Card	XC4989	D0328

Of course, you will need other common RPi accessories, such as a HDMI cable and monitor, keyboard, mouse and power supply. It's also recommended to have some plug-socket jumper cables to emulate the push button on the Google Voice HAT hardware; and an LED wired in series with an appropriate resistor, to give feedback on the status of the voice recogniser program.

As noted above, the MQTT broker can run on the RPi, but there are versions available for other operating systems such as Windows and even OpenWRT. If you haven't already got one set up, I would use the RPi option, as it will need to be running for this project to work.

The Arduino side of things doesn't require much work. The prototyping shield is optional, but I found it meant that everything slotted together in a compact way, and could easily be pulled apart later for use in other projects. The alternative is to solder some plug-socket jumper leads onto the legs of the 433MHz Transmitter Module, and plug these into the headers on the WiFi Shield.

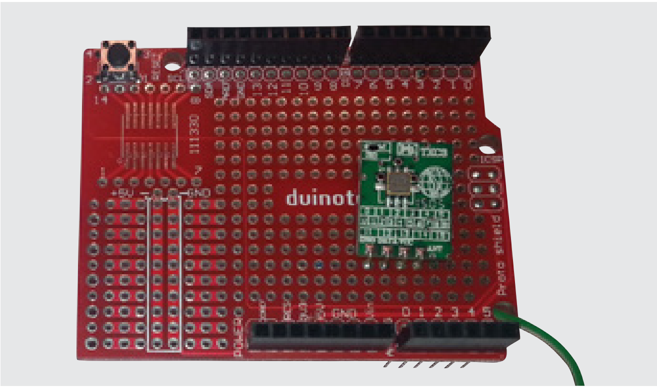
PARTS REQUIRED:	JAYCAR	ALTRONICS
1 x Arduino Leonardo Board	XC4430	-
1 x ESP8266 WiFi Shield	XC4614	-
1 x 433MHz Transmitter Module	ZW3100	Z6900
1 x Prototyping Shield (optional)	XC4482	Z6260
1 x Wireless Power Point	MS6148	-

BUILDING THE CIRCUIT

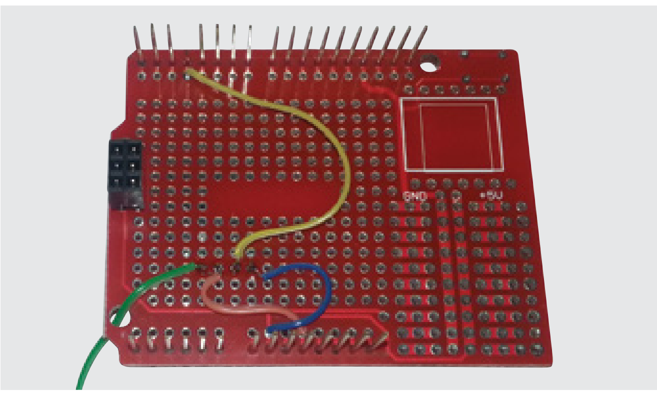
As the WiFi Shield can plug directly onto the top of the main board, the only connections that need to be made are soldering some jumper leads to the 433MHz Transmitter Module, or alternatively, plug 433MHz Transmitter Module and the jumper leads into a breadboard.

LEONARDO	433MHZ TRANSMITTER MODULE
5V	VCC
GND	GND
D3	DATA/DATA In

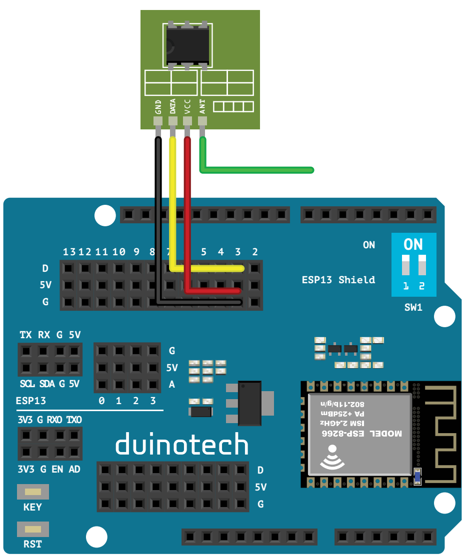
The ANT (antenna) connection does not need to connect to anything, but a short length of insulated wire (about 17cm is ¼ wave at 433MHz) should be attached to improve signal transmission. For the Prototyping Shield version, first solder the 433MHz Transmitter Module to the Prototyping Shield. ››



›› Next, make the necessary connections between the shield and module. Note how the green antenna wire is threaded through the hole in the Prototyping Shield; this helps to stop it flexing and breaking off.



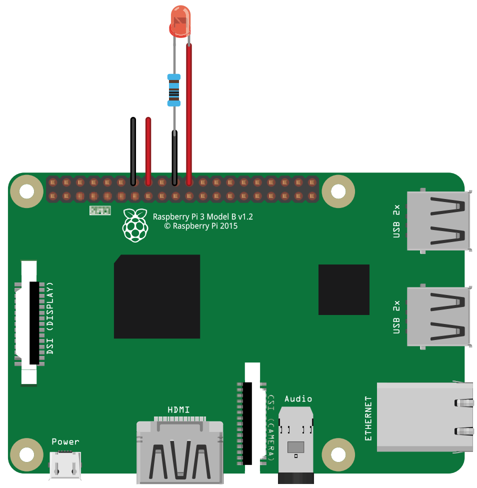
Alternatively, this is what it would look like if the connections are made directly to the WiFi Shield.



The shields are then attached to the top of the Leonardo to complete assembly of the Arduino side of the project.

The RPi doesn't need much in the way of mechanical build. The USB Sound Card is plugged into the one of the RPi's USB ports, and the microphone and speakers are plugged into their respective locations on the USB Sound Card. It will be assumed that you have a keyboard, mouse and monitor set up to use with the RPi. Once the build is complete, the keyboard, mouse and monitor can be disconnected and the entire project can be run headless.

I found it easiest to solder the LED and resistor (I used a blue LED in series with a 100Ω resistor) to the plug ends of the plug-socket jumper leads; and instead of a switch, I left the ends exposed, so I could briefly touch them together to emulate a button push.



CODE AND SETUP

The code for the Arduino is similar to that used in the MQTT Light Switch project, and setup is similar. The following lines will need to be configured to match your WiFi network and MQTT broker.

```
//spec WIFI network and MQTT server
#define SSIDNAME “SSIDNAME”
#define SSIDPWD “SSIDPASSWORD”
#define MQTTBROKER “BROKER IP ADDRESS”
#define MQTTPORT “1883”
```

If you haven't set up your broker yet, the easiest way is to use the RPi - just enter your RPi's IP address for the broker. A good idea is to get into your router settings and give it a reserved address under DHCP. This also makes it easier if you need to SSH in, to tweak a setting while it's running headless. The MQTTPORT shouldn't need to change as 1883 is the default for most MQTT implementations.

You might also want to change the below if you want to use different topic names and messages. The default topic names are the strings in the first five “if” statements and the messages are “on” and “off”. These correspond to the commands that are run on the RPi, so both need to be changed to suit your specific customisation.

```
   int cmd=-99;
//default fail
   if(strmatch(topic,"lounge/lamp")){cmd=0;}
//button A
   if(strmatch(topic,"bedroom/lamp")){cmd=2;}
//button B
   if(strmatch(topic,"spare/lamp")){cmd=4;}
//button C
   if(strmatch(topic,"spare/spare")){cmd=6;}
//button D
   if(strmatch(topic,"all/all")){cmd=8;}
//button ALL
(strmatch(message,"on")||strmatch(message,"off"))
{
   if(strmatch(message,"off")){cmd++;}
   if(cmd>=0){
      DEBUG.println("Action");
      sendrf(packet(address,rfcmds[cmd]));
   }
}
```

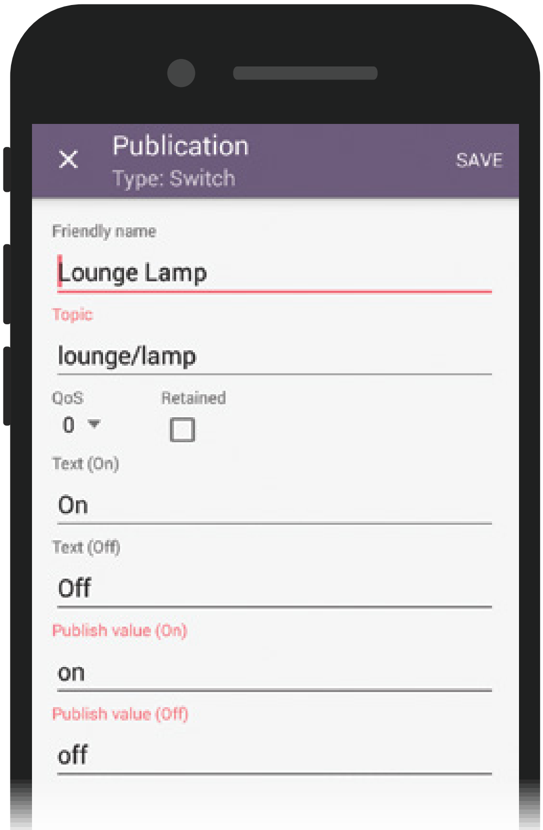
The Arduino uses an arbitrary code which is set in this line.

```
const unsigned long address=0x12340;
//this could be any 20bit value (not all tested)
```

This code will probably not be the same one as the remote you have, so the Arduino will have to be programmed as a second remote (these units can be paired with up to three remotes). After this, the sketch can be compiled and uploaded.

Next, ensure that the wireless power point operates correctly with the remote control. The wireless power point is in pairing mode for the first 30 seconds after it is turned on, so I found it easiest to plug the wireless power point into a switched power point and toggle the power to enter pairing mode. Pressing an "on" button during pairing mode will cause that button to be paired - you can tell that it works as the wireless power point will turn on at that time.

To set up the wireless power point, it is recommended to install an MQTT app or other form of client to be able to issue commands to manually trigger the Arduino. If you are using the MQTT Dashboard app, you will need to set up a broker and then "switch" controls on the "publish" tab.



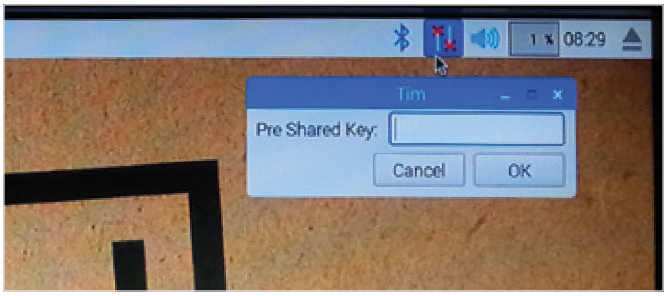
Create four more switch objects corresponding to the topic names and messages in the Arduino code. If you have the mosquitto clients installed on a PC or Pi, you can run the following commands (substituting the IP address of your broker for 192.168.0.223):

```
mosquitto_pub -h 192.168.0.223 -t
                        lounge/lamp -m on
mosquitto_pub -h 192.168.0.223 -t
                        lounge/lamp -m off
```

Replace "lounge/lamp" with the other strings from the Arduino code to emulate the other buttons. Monitoring the Arduino debug on the serial monitor can also assist in checking whether everything is functioning.

Once the Arduino side of things is working, we can move on to the RPi. The first step is to download the image from *https://dl.google.com/dl/aiyprojects/voice/aiyprojects-latest.img.xz*, and write the image to a 16GB or greater card using a program like Etcher or WinDiskImageWriter.

Boot up the RPi with the new card installed, and connect to a WiFi network with internet access. ››



›› Click on the icon marked “Start dev terminal”, and run:

```
sudo leafpad /etc/asound.conf
```

Replace the contents of asound.conf with the following, then save.

```
pcm.mic {
   type dsnoop
   ipc_key 1025
   slave {
      pcm “hw:1,0”
   }
}
pcm.speaker {
   type plug
   slave {
      pcm “hw:1,0”
   }
}
pcm.duplex {
   type asym
   capture.pcm “mic”
   playback.pcm “speaker”
}
pcm.!default {
   type plug
   slave.pcm “duplex”
}
```

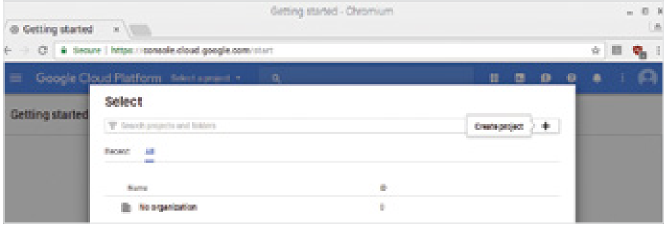
This changes the software to use the USB sound card instead of the non-existent Voice HAT, and also allows the hotword trigger to be used; it is documented in this thread: *https://www.raspberrypi.org/forums/viewtopic.php?t=183932&p=1167683.*

Reboot the RPi and then double-click the “Check audio” and “Check WiFi” buttons to make sure everything is working so far.

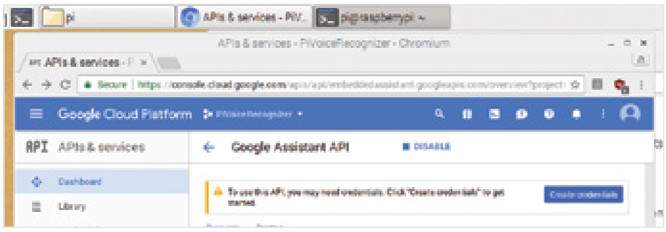
If you haven’t got a Google account, now is a good time to get one set up. The next step is to set up our Google account to allow the RPi to access the Google Assistant API. There is some information about this, on this website: *https://aiyprojects.withgoogle.com/voice#users-guide-1-1--connect-to-google-cloud-platform*

I found that these instructions didn’t match the web pages I was viewing, so I did the following:

Open *https://console.cloud.google.com/* on the RPi’s browser (there’s a link to it on the bookmarks bar), and make sure you’re signed into your Google account. Then create a new project.

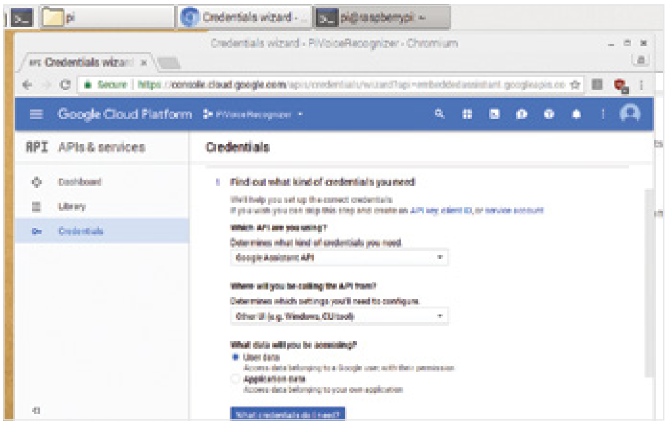


Open the project, and go to APIs and click “Services>Dashboard”. Click on “Enable APIs and Services”. Type “assistant” in the search box and click “Google Assistant API”.

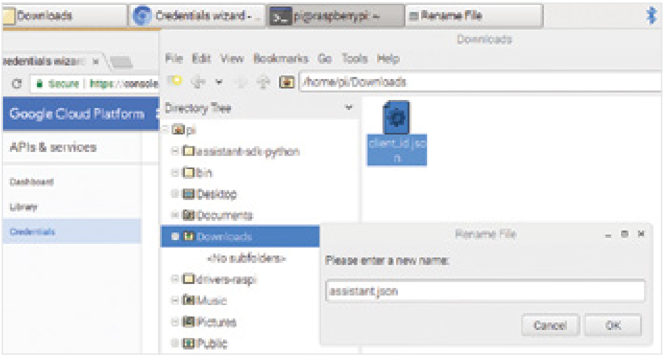


Click “Enable” and click “Get Credentials” under “Add credentials to your project”.

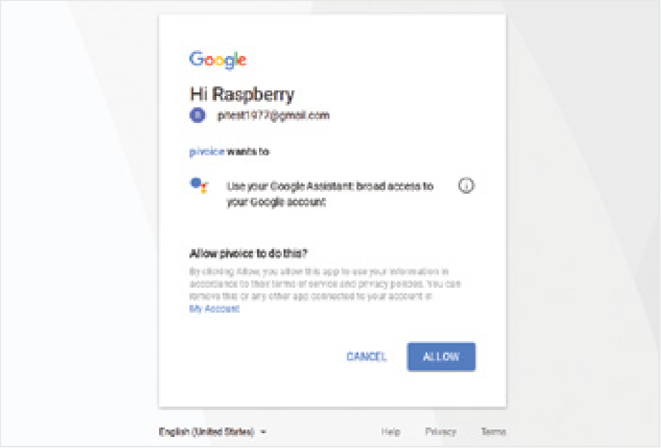
Select “Other UI” from “Where will you be calling the API from” and “User data” from “What data will you be accessing”. Then click “What credentials do I need”.



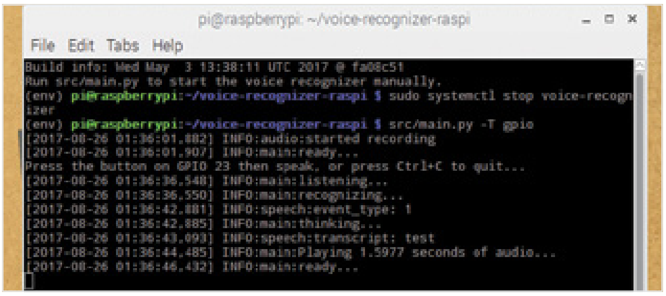
Click “Create client” and enter an arbitrary product name (I used “pivoice”), then click “Continue” and download credentials. The downloaded file “client_id.json” in /home/pi/downloads should be renamed “assistant.json” and should be moved to the /home/pi folder.



Click on “Start dev terminal” and run “src/main.py”, which will start the voice recogniser. A browser window will open so that you can provide authentication, so click “Allow”, and close the browser window once this has completed.



By default, the voice recogniser is set to trigger from the switch on the Voice HAT, so make sure that src/main.py is still running in the terminal window, and trigger it by touching the GPIO23 wire to the GND wire. The terminal should indicate that it is listening, so speak into the microphone. “What day is it?” is something that Google should be able to handle. You can stop the voice recogniser by pressing ctrl-C.



If you get a message like “Actually, there’s some basic settings that need your permission first”, open Google account settings in a browser, and click through “Personal Info and Privacy” to “Manage your Google activity”. Go to “Activity controls” and turn on “Voice and Audio Activity” and “Device information”.

By default, hotword activation (“OK Google”) is not installed. The process for adding this is documented on the RPi forum at *https://www.raspberrypi.org/forums/viewtopic.php?f=114&t=183932#p1164380*

It is recommended that this be viewed on a browser on the RPi, so that all the code can be copied and pasted. Effectively, we add a hotword.py file to perform this function, and modify main.py to be able to use hotword.py.

Once the new code is set up, test it using:

```
src/main.py -T hotword
```

This is where it helps to have the LED, as you will see it turn solid on when it has received the trigger correctly. To set the hotword as default trigger, change the ini file and run the following:

```
leafpad /home/pi/.config/voice-recognizer.ini
```

Then change:

```
#trigger = clap
```

to:

```
trigger = hotword
```

Test this by running src/main.py, and saying “OK Google”. There will be a delay of a few seconds before the phrase is recognised, and this will be seen on the LED and output on the terminal. If this all works as expected, then our setup is practically complete.

By default, the voice recogniser is not set to auto-start, as it would error with the credentials not set up. To start the service, run the following: ››

```
sudo systemctl start voice-recognizer
```

›› To enable auto start, run:

```
sudo systemctl enable voice-recognizer
```

If you make changes to the code (which we will be doing shortly, to add our commands to interface the MQTT broker), you can also run:

```
sudo systemctl stop voice-recognizer
```

Do this before making changes, and then run the above start command afterwards, to restart the service.

To add manual commands, we need a trigger sentence, and a command to execute. It's a good idea to test the trigger sentence while you have the voice recogniser running manually, as you can see what it thinks is being said. To add a set of commands to match what our Arduino is looking for, edit the "action.py" file leafpad/home/pi/voice-recognizer-raspi/src/action.py and place the new commands in this section, just before the "return actor" line.

```
    # ========================================
    # Makers! Add your own voice commands here.
    # ========================================
Return actor
```

The code we have inserted into this space is shown at the bottom of this page **[1]**.

Looking at the first line, the "lounge lights on" is the phrase that the recogniser is looking for, and the part in double quotes is the command that is run on the terminal. This actually runs two commands: the first part "mosquitto_pub..." is the MQTT message, and the second part "echo ok" is simply text printed out to the terminal; but because printed text gets read out by the RPi, it can also be used for audio feedback. The IP address, topic names and messages should of course, be changed to match your setup.

At this point, you should be able to restart the voice recogniser service and give voice commands to your power points.

WHERE TO FROM HERE?

Once the Google Assistant API is up and running, it can do much more that just control the lights. It has many of the features of a Google Home device, so you can try asking it things like "How's the weather?" and "Where's the nearest restaurant?". But that's not really what you set this up on a Raspberry Pi to do is it?

The src/main.py and src/action.py files are well commented with tips about how to add actions directly to the Python code, so if you are comfortable with Python, this is another place to add functionality.

It's worth noting too, that we have integrated MQTT here, and are using Raspberry Pi as well as an Arduino system to provide all the functionality. By having MQTT in the middle, we can effectively control the system using our familiar MQTT app and process. This is valuable in many ways, but also creates the requirement for additional hardware which, perhaps you don't want to use.

If you aren't interested in hooking up an MQTT broker and an Arduino, then you probably want to be able to control the GPIO on the RPi directly. This process would be faster, and require less hardware. Of course, there's no redundancy with a backup / remote app, unless you create that directly on the Raspberry Pi also.

To use the Raspberry Pi GPIO and instantly interface into the real world, you'll need to make a few changes to the action.py file. If you're solely an Arduino fan and aren't too familiar with Python, it's not too difficult to see the differences in convention and figure it out. As for any other alterations, remember to stop the voice recogniser before making changes.

1

```
actor.add_keyword(_('lounge light on'),SpeakShellCommandOutput(say,
                    "mosquitto_pub -h 192.168.0.223 -t lounge/lamp -m on && echo ok",_("fail")))
actor.add_keyword(_('lounge light off'),SpeakShellCommandOutput(say,
                    "mosquitto_pub -h 192.168.0.223 -t lounge/lamp -m off && echo ok",_("fail")))
actor.add_keyword(_('bedroom light on'),SpeakShellCommandOutput(say,
                    "mosquitto_pub -h 192.168.0.223 -t bedroom/lamp -m on && echo ok",_("fail")))
actor.add_keyword(_('bedroom light off'),SpeakShellCommandOutput(say,
                    "mosquitto_pub -h 192.168.0.223 -t bedroom/lamp -m off && echo ok",_("fail")))
actor.add_keyword(_('spare light on'),SpeakShellCommandOutput(say,
                    "mosquitto_pub -h 192.168.0.223 -t spare/lamp -m on && echo ok",_("fail")))
actor.add_keyword(_('spare light off'),SpeakShellCommandOutput(say,
                    "mosquitto_pub -h 192.168.0.223 -t spare/lamp -m off && echo ok",_("fail")))
```

Add the following line to rest of the imports at the start of the file:

```
import RPi.GPIO as GPIO
```

Just above the section marked “Makers! Implement your own actions here”, put this code:

```
class SetPinOn(object):
   def __init__(self,say):
      self.say = say
      GPIO.setmode(GPIO.BCM)
      GPIO.setup(24, GPIO.OUT)
   def run(self, voice_command):
      GPIO.output(24,GPIO.HIGH)
      self.say(voice_command)
class SetPinOff(object):
   def __init__(self,say):
      self.say = say
      GPIO.setmode(GPIO.BCM)
      GPIO.setup(24, GPIO.OUT)
   def run(self, voice_command):
      GPIO.output(24,GPIO.LOW)
      self.say(voice_command)
```

And finally, inside the “make actor” function (with the other lines we added earlier), add these lines:

```
    actor.add_keyword(_(‘on’),SetPinOn(say))
    actor.add_keyword(_(‘off’),SetPinOff(say))
```

Then restart the voice recogniser, and use the keywords “on” and “off” to turn GPIO 24 on and off. It’s a bit more involved, but it is also possible to set pins as inputs and read back their state. ■

WANT MORE?

For additonal code, or to discuss this project, visit:
https://diyode.io/004swnw

Reading & Resources:

- **MQTT** *mqtt.org*
- **GOOGLE AIY** *aiyprojects.withgoogle.com*

How Accurate Is A Voice Command?

One of the cool things about modern voice command technology is that it leverages deep learning and continuously improves accuracy thanks to machine learning.

Traditionally, software such as Dragon Naturally Speaking, which aimed to get us off the keyboards and on to the microphones, used self-contained voice data. It's efficient and effective, and they really carved out a niche (though arguably, was aimed at users before the large majority of us spent a day in front of the keyboard and many of us could type better than we could write with a pen).

Where this type of voice pattern recognition falls down however, is its ability to detect voice patterns when you change users. If you use something like Dragon Naturally Speaking daily, it will become increasingly accurate to your own voice. If you then hand the microphone to a colleague however, it will instantly lose substantial accuracy.

Of course, even the Google Assistant, Apple's Siri, and others, can be tripped up rapidly with a unique accent. But they learn, and learn fast. Once it's learnt, everyone benefits from the development.

Soon, these assistants will have better conversational artificial intelligence, and it will be able to decipher implied speech, sarcasm, and other nuances which usually fall short. The way we interact with these systems will become more natural, relaxed, and conversational. Even those of us who use them daily, tend to talk to them like robots.

> **Hey Siri! CALL-JOHN-SMITH.**
>
> **"Calling Jack Quiff Mobile"...**

We've all been there... but this is just the beginning of what's to come. Soon, we'll be able to have a conversation with a virtual assistant as casually as we would with a mate in a bar.

Background noise, slurred speech, random queries, will all be seamlessly handled by the virtual assistant, with the accuracy (or probably better) of a human. That's exciting, and it's right around the corner.

POWER OF THREE

TOM BISKUPIC
Sydney-based maker and enthusiast, PKI software developer.
@tom_biskupic

Creating your own bench power supply is no small task, let alone a three-channel, touch-screen controlled unit! With three fully independent channels, voltage adjustment from 0 to 30V, and up to 4.6A available, per channel! It's super steady and we we wanted to know more!

Tom was looking at expensive commercial power supplies, until he realised he could probably make his own, and learn a stack of new information in the process! With a Raspberry Pi (RPi) at heart, and a touch-screen for an amazing user-experience, we wanted to know more about Tom's amazing creation.

Other than your need for a power supply (and who doesn't need a good power supply?), what led you to embark on this project - instead of just buying one!

I recently took up electronics as a hobby and I was using a simple Jaycar single channel power supply for my experiments. But one day I needed a bipolar supply for an opamp circuit and realised I needed something more powerful.

I actually studied computer engineering many years ago, but I ended up bailing out. Since then I've been working as a software developer. So while I am not totally new to this, I am still very much an apprentice.

I've always enjoyed playing with electronics, having grown up hacking on Commodore 64s and the Commodore Amiga. A few years ago my son started playing with quadcopters and 3D printing, and I really wanted to learn more. I started watching Dave Jones on EEVBlog and things just took off again from there!

In addition to multiple channels, I also wanted a bit more current than the 3A my Jaycar supply could provide. I also found that although the Jaycar supply is very dependable, it's not very quick and takes quite some time to respond to load transients and the like. I looked at the Rigol DP832, but since the whole point was to reboot my electronic knowledge, I thought I would have a go at making one! As it is a one-off I had a pretty free hand in terms of budget and I didn't care about physical size. Basically I wanted a really quiet linear supply with very high accuracy. Being a smart and networked device seemed important too, because I had lots of ideas where being able to control the supply and coordinate it with other devices would be useful.

Making the project open source seemed important. I wasn't sure (probably am still not sure!) if it is something someone would want to copy, but if anybody did want to use my design then I felt it was important they could understand it fully. I love old Keysight and Tektronix equipment where you can get full schematics and even detailed descriptions of the circuit operation. It means you can really understand the limitations of your equipment.

We love Dave and the EEVBlog channels too. There are many choices when you have a combination of a microcontroller and analogue circuitry - what made you settle on the Raspberry Pi?

To be honest I didn't consider other Linux-based SBCs too much, as I was familiar with the Raspberry Pi (RPi) and quite liked the RPi community.

The RPi was attractive for a few reasons, but initially I was thinking I could easily get networking and the ability to store data on USB for free. I also thought I could build a user-interface toolkit targeted at electronic instruments. I thought it might be an interesting platform to build other test equipment. For example, I thought of building a home-brew arbitrary function generator or perhaps an electronic load.

In the end I found the Kivy Pi toolkit (*https://kivy.org*), and it provided loads of useful tools. I didn't feel the need to build a toolkit but it might still be worth it. I'd need to do another project I think to really understand what could be generalised.

Just being able to program the GUI logic in Python is very nice. You can do anything in Python. I work in Java and C++ during my day job, but for small things you can't go past Python.

Using USB to connect between the channels and the front-end made life really easy. Each channel has an Atmel ATMEGA328PU that runs the show. I used a MCP2200 USB to serial chip to interconnect these, as it is well supported by the Pi. Also using USB meant that during development I could connect the channel control boards to my computer, and just type commands at them to make them go. The USB hardware on each channel however, is isolated from the rest of the channel circuitry. The Pi powers the MCP chip, and then the serial signals from the MCP are passed through an isolator chip before they go to the MCU on the channel.

The downside of using the RPi is that it is a bit slow to boot, and you have to shut it down properly. The latter point led to the need for a standby supply and a soft-power switch, but it worked out okay in the end.

Now there's a valid point - we never really considered boot up/shut down when using Raspberry Pi for something like this. Nevertheless it's a great basis for control. A touchscreen is a great interface. Was that an easy decision?

I wanted to use an LCD user interface from the beginning, but to begin with I had other plans for input. My son told me about this flexible 3D printing filament and so my plan was to experiment with printing my own keyboard membranes.

Using a touchscreen however, means the user interface is totally flexible. For example, halfway through the project I reoriented the user interface from a vertical layout to a horizontal one, as I didn't have enough space below the screen for the binding posts.

The cost of the display is a factor but I bought the parts over time so it didn't matter too much to me. Also compared with what a comparable commercial supply would have cost, I felt it wasn't that much; but maybe I have more pocket money than some!

I probably should have implemented a dial to adjust voltages/current; I still might. I also thought about adding a virtual dial on the display - I've found a Kivy widget for doing this.

I probably didn't need a display quite so big. I had (have) plans for more graphic capabilities, and so the big screen was chosen with that in mind.

A PSU with live voltage and current statistics on the display? We could get behind that! What made you decide on three independent channels?

I originally planned to have two channels only, but when I was working though details of the box and heatsinks, I found I could do three. Having the analogue circuitry on separate boards makes this relatively easy.

Three channels gives me the ability to configure a bipolar supply (so for example +/- 15V for some analogue circuitry), and then I still have an additional channel for a digital supply. Also I can connect them in series to generate higher voltages.

I do like repairing old equipment and having three (or four if I use my old supply) independent channels, as it means I can bypass the power supply in a piece of broken equipment and run the unit from the lab supply for troubleshooting.

Having the 1mA/1mV resolution gives you some interesting capabilities also. For example it's a neat way to measure an LED voltage drop, as you can set the current limit to the current you ››

›› want the LED to consume, and then you can read off the voltage. So to enable a limit of 20mA, and then set the voltage to say 8V, you can just read off the forward voltage drop at 20mA from the supply.

Lately I've been thinking about upgrading the software to implement a sort of curve-tracer type feature (a poor man's SMU), where I can use one channel to sweep the bias voltage on the gate of a MOSFET (for example), while using the other channel to provide and measure current through the source/drain. I can imagine other configurations where three channels would be useful also.

The RPi software is pretty rudimentary right now, and I plan to add more features.

As long as it's stable and providing good functionality, the bells and whistles can come later! Does each channel have the same capabilities?

Yes the channels are all symmetrical; they are built on identical boards. You could probably reduce the size/weight/cost by using smaller transformers for one channel and adjusting the software current limits, but for me it was easy enough to make them all the same.

Awesome - that definitely provides great versatility. Tell us about any special features of this power supply that may not be evident?

The first thing is that the supply is very high precision. It uses an AD780 voltage reference, which is very stable. It uses 16 bit analogue devices ADCs and DACs, so I can set the voltage and current to within 1mV or 1mA, or read back the output voltage and current flow to the same accuracy. I spent some money and bought Vishay 4 terminal shunt resistors to measure the output current, and I use fancy low-offset voltage opamps (LT2050HV) in the current measurement circuit. Also there are four wires from the analogue board to the terminals, and there is a differential voltage amplifier to measure the voltage at the terminals, which eliminates voltage drops in the internal wiring.

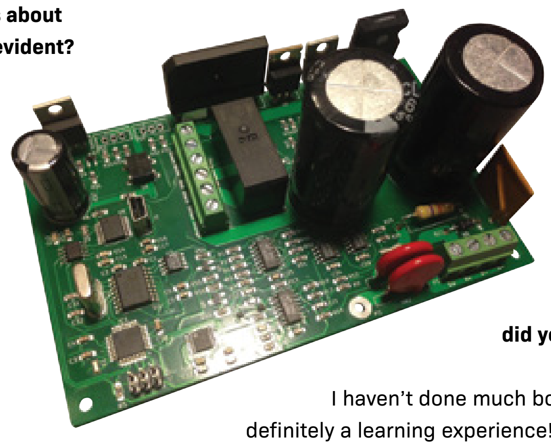
The voltage control loop is pretty quick. Keysight usually specify a settling time of 60 microseconds from a 50% change in load. I have measured less than 20 microseconds of settling time when the load switches from zero to full load.

It is a linear supply so the design has to manage heat. The design uses a MOSFET as the pass element, to reduce the required voltage drop. It also uses a pre-regulator to manage the voltage on the pass transistors and thus, to reduce heat losses. Transformer tap switching is used when switching voltage ranges, although this is more about increasing the current output at lower voltages.

The supply is designed to be very low noise on the output. Rather than using a switching pre-regulator, the design uses MOSFETs to switch off the current to the bulk capacitor once it reaches a target voltage. I initially tried using Jim William's design from AN-32, which uses an SCR to delay the turn-on time at each mains cycle. This generated a current surge as it switches at the part of the cycle where the current is highest. By switching off when the capacitors reach the target voltage, it is switching at the lowest current flow. The pre-regulator is based on a design I saw on the EEVBlog (put up by a guy who goes by "Black Dog" and hence it became known as the "BlackDog Pre-regulator" although I think he shared it from some other place initially).

I've tried to measure the output noise, and all I can say is that with my measuring equipment it appears to be less than 1mV RMS. The supply can provide 0V to 15V at 4.6A and 15V to 30V at up to 3A. It's pretty cool being able to set the output to 10mV to play with some devices.

Towards the end of the development I splurged on a bench digital multimeter (DMM), capable of being driven from Ethernet. The supply has a feature where you can connect the outputs to the DMM and put it in a calibration mode where it talks to the DMM directly. The supply reads back the voltage while it steps the DACs through a series of voltages. The supply then saves this data to NVRAM on the power supply channel as calibration data. Pretty neat!

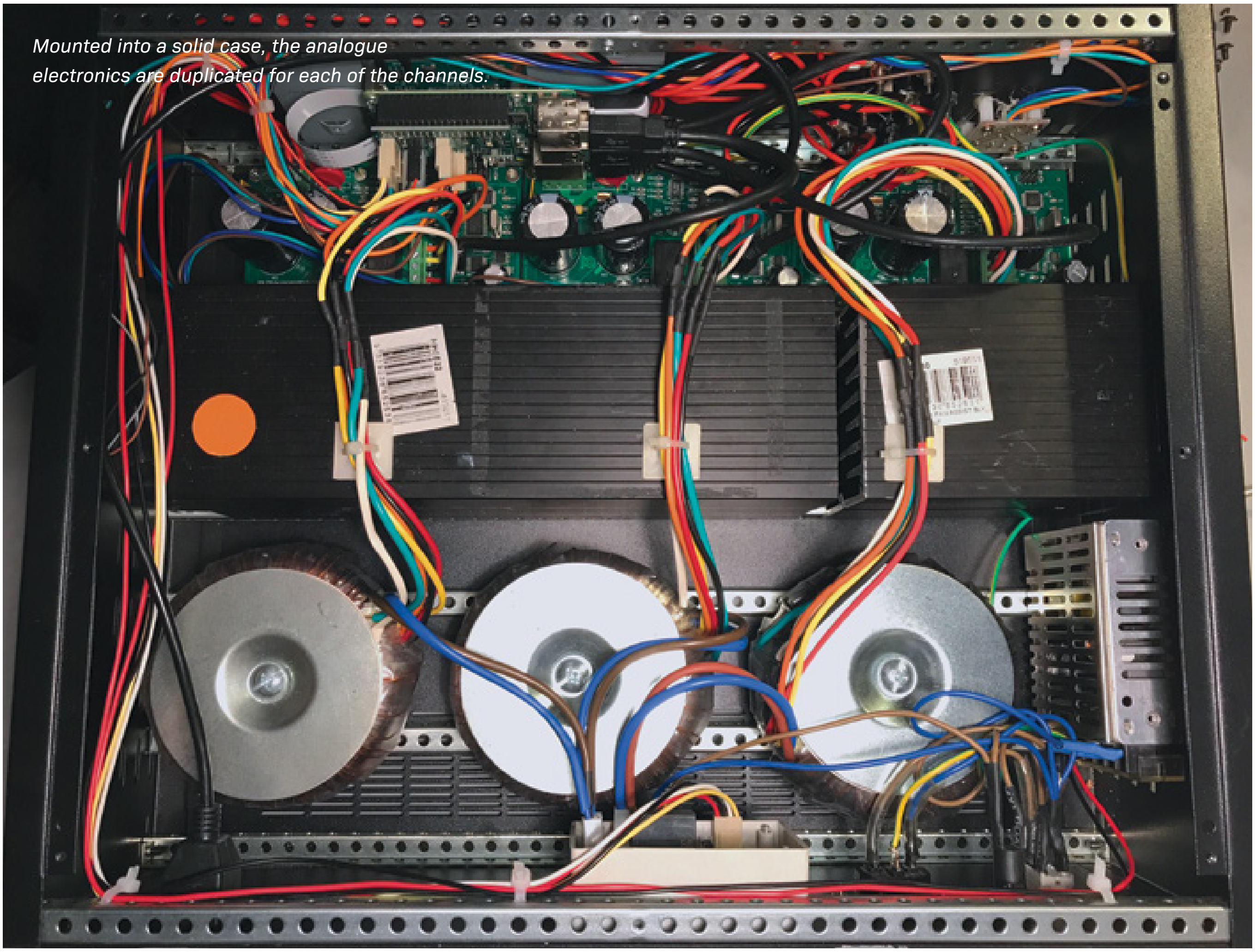


◀ **LEFT**
Tom's custom PCB.

Self-calibration (in effect) is a great thing to have; we're a little jealous right now! Did your analogue board take much revision, or did you nail it first-go?

I haven't done much board design so this was definitely a learning experience! I started in the free Altium Circuit Maker when it was in Beta. I really wasn't having much fun there and so switched to KiCAD. I love that KiCAD runs on my Mac and on my Linux computer. Last time I used anything like this it was called Protel. I quite enjoyed getting the hang of KiCAD overall.

So in the layout I have a digital section where I use a ground plane pour to reduce noise. In the analogue section, the grounding is more important so I run ground traces more deliberately. I wasn't sure what size traces to use for the current requirements and so have probably overdone this. It does make layout pretty hard when they're big. The power devices had to be at the back to get to the heatsink, and with the outputs at the front, I needed strong current paths between the power devices and the outputs. The digital had to be separate from everything else.


Mounted into a solid case, the analogue electronics are duplicated for each of the channels.

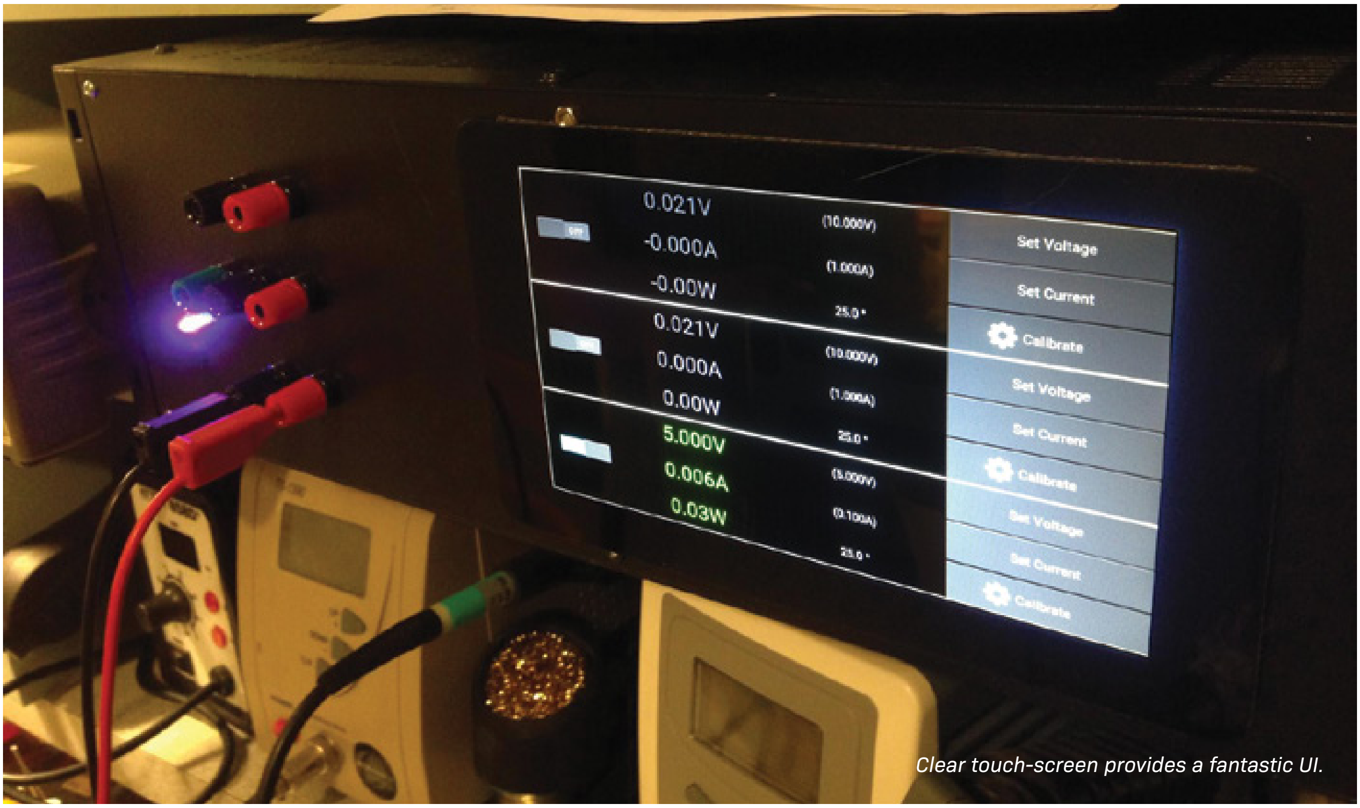
I did two revisions of the channel board - so many things went wrong! I stuffed up the footprint of the MCP2200 chip; it was a wide SSOP where I used a standard. One of the small SOT-3 transistors had the wrong pin ordering, so I had to bodge in a through hole for testing. And I forgot the diode across the relay, so it would reset the whole board when it switched transformer taps! Another rooky mistake was my footprint for the MOVs (they ensure the output never goes above 250V) and the PTC (to limit the current in case of failure) were backwards. They are asymmetric (point-symmetrical) but I misunderstood the datasheet, so the layout would actually have been correct if I'd soldered them onto the bottom of the board! I also didn't implement the reset circuit correctly either, so some of the SPI hardware didn't get reset. In hindsight, I am amazed this worked! But the fault would cause programming of the micro to fail occasionally.

The biggest problem however, was getting the grounding right. The current shunt really has to be the ground for everything to work correctly, and for the output to be accurate. Small resistances in the ground traces cause significant errors. The second revision improved this, but I still don't think I nailed it. The first version would wander by a few mV as the load increased.

The analogue circuitry runs off 44V in 30V mode, and 28V in 15V mode (transformer taps). The relay runs off 24V so I had an LM317 that provided this. I underestimated the heat this would generate and used a surface mount LM317 in the first revision. In the second revision I used a TO-220 on the heatsink.

I was using Welwyn wire current shunt resistors to measure the output current. But I found a problem in that they would significantly drift as they warmed up. I was surprised by this, as their datasheet says they have a low tempco. I figured out later it was the thermo-electric effect and it was voltage drop on the solder joints. They would have been fine at higher currents, but 1mA equated to 10uV at the shunt and I was seeing 25uV of drift. This totally blew my accuracy budget.

In the second revision I figured out the box layout more, and changed the board from having a narrow edge on the heatsink to having the long edge on the heatsink. This reduced the lengths of high-current tracks, and kept things together more. I also got a bit better at the layout process, so the board came out more densely laid out. ››



Clear touch-screen provides a fantastic UI.

›› Some valuable lessons in there, but an amazing effort for someone without much board design experience. If you had your time again, starting from version 1.0 - what would you do differently?

The biggest issue is that the UI is far too slow. I use a text protocol between the Pi and the channels and I think I need to slim this down a bit. I will probably fix this soon. This is also why I've not yet implemented an encoder or virtual dial - it would be too confusing right now.

I'm also having problems with the brown-out detection on the micro-controllers inside the power supply channels, which randomly screws up the calibration. I think it would be better to have the Pi store this data, and poke it into the channels at start-up.

I think if I were designing the channels again from scratch, I would change the voltage regulator design. Using an N channel MOSFET as the pass element means the gate voltage must be a few volts above the output. This then means the analogue circuitry and the gate needs to run at this higher voltage, and so the opamps are all 40V parts. This limited my choice of parts and made things more complicated.

If I was doing this again I would also use a floating bipolar supply for the analogue control circuitry, and make the positive output the ground for the control. This way the control circuit rides on the output voltage, and never needs to be more than a few volts above "ground".

I originally designed the supply to provide sense terminals as well as output terminals. The problem was I just couldn't make this stable if the leads were even 30cm long. Getting this to work would be a great feature, especially when coupled with the supply's high accuracy.

Some great upgrades to tackle in the future. At least with the Pi your software upgrades can be trialled in-place. What is the next amazing project you're working on?

Alas, work has me pretty occupied at the moment! But I'm slowly working through the restoration of a Phillips 115B valve radio from the 1940s for a friend. Valves are quite fascinating, although the circuitry terrifies me! Tag strip construction with 300V! I think it is pretty amazing that they can make a superhetrodyne receiver with just four or five valves.

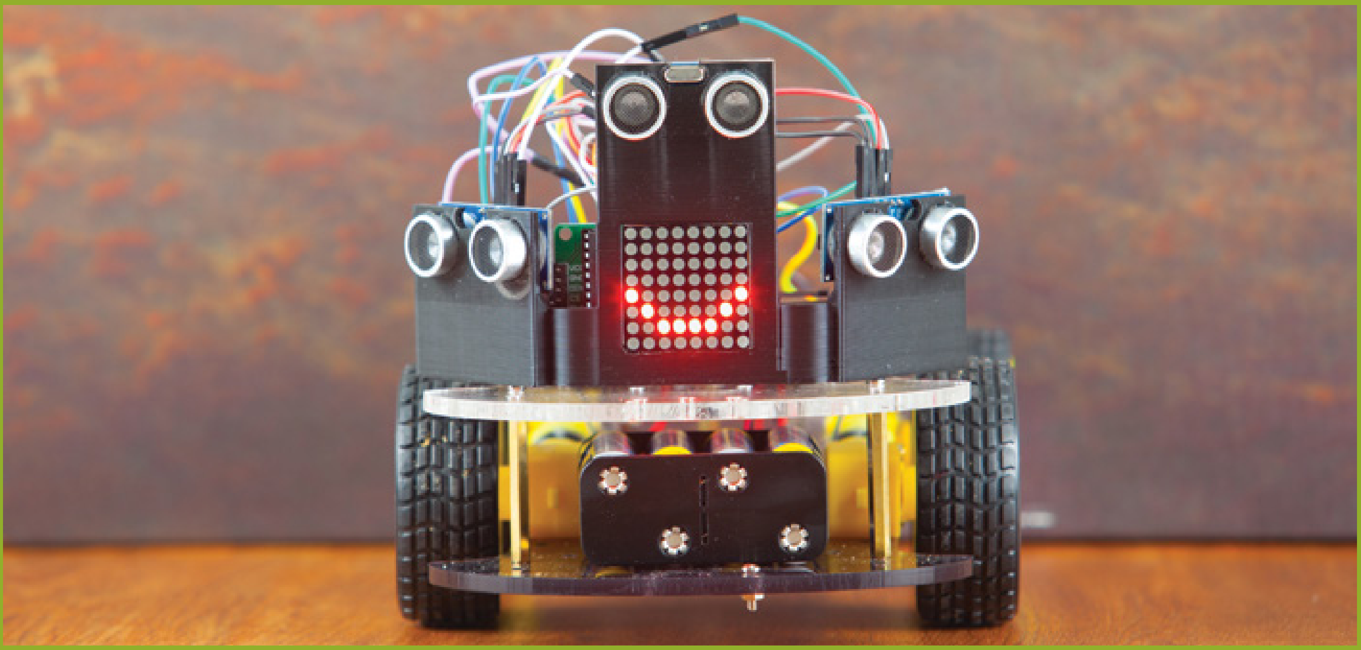
I have a long list of things I'd like to learn about next, including some RF circuit design.

Well it sounds like we'll be seeing some great projects from you in the future. Congratulations on your power supply, and thanks for sharing it with us. ■

WANT MORE?

For the circuit diagrams and code files, or to discuss this feature, visit:
https://diyode.io/004xpbk


MADE THE THING?
Now Show it Off!
Each and every month we shine a spotlight on impressive projects from the electronics and maker community. We've already seen amazing projects like Thomas's Pi Powered Sprinkler and FRED the Tracked Robot from Royce in Issue 1, Liam's Toy Piano Conversion in issue 2, and Greg's Ultra-Accurate Voltage Reference in issue 3, just to mention a few.
Awesome ideas inspire others, so make all those late nights developing your project count even more. Not only will you get a buzz out of seeing your creation in print for the world to see, you'll know that you'll probably encourage other readers to get hands on to start or finish their own project.
To show us what you've built, simply send us a few details along with some image of your project, and we'll get in touch if we have a place for it in a future issue.
Featured builds are also in the running for the $250 prize for Editor's Favourite - we'll pick one each issue!
DIYODE
SUBMIT YOUR PROJECT:
diyode.io/004submit

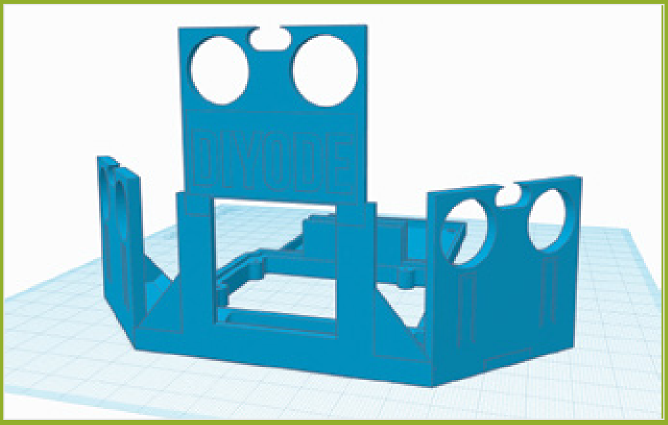


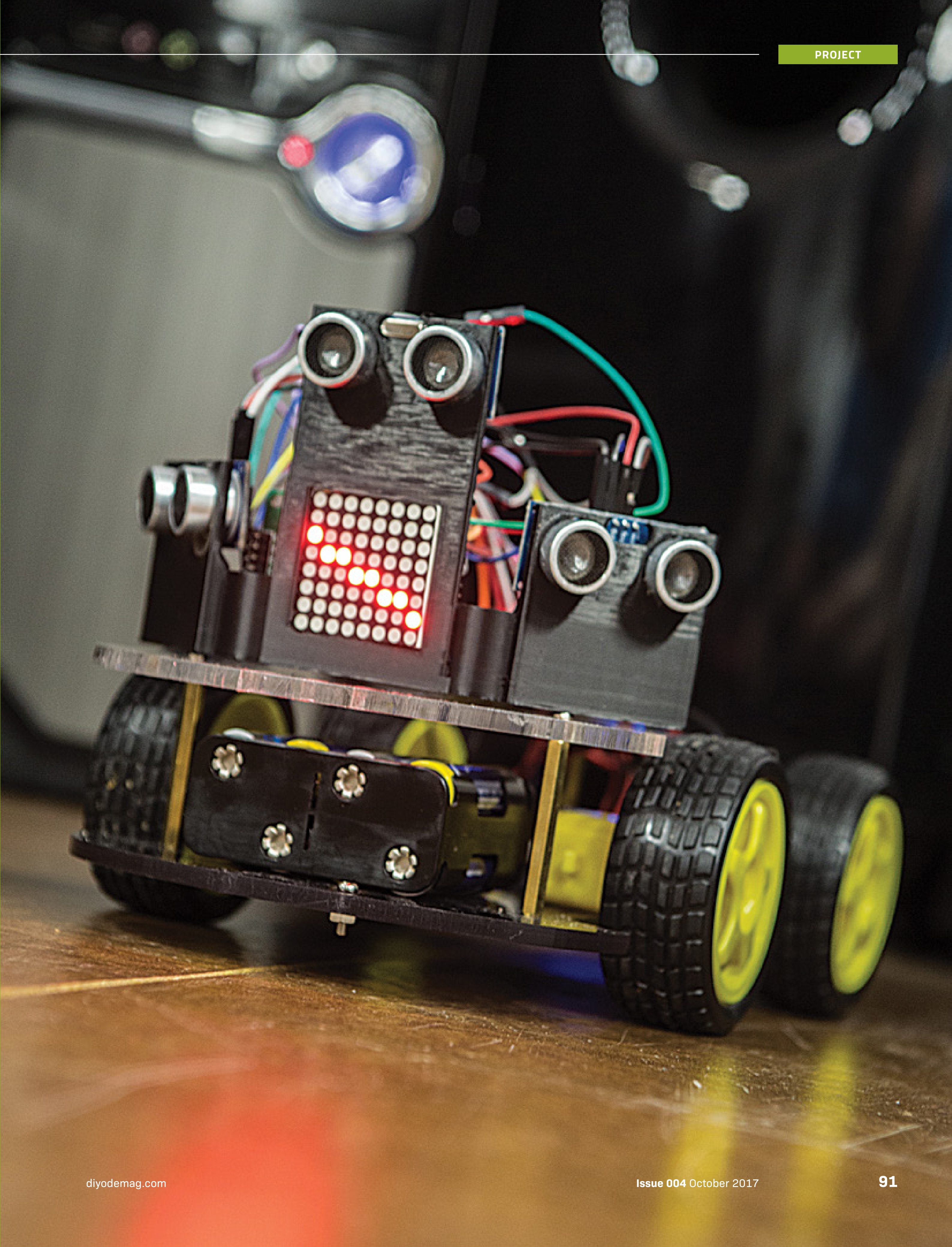
Part 2:
SAM v1.1
Upgraded Senses

In Issue 3 we built a basic obstacle-avoidance robot named SAM (Somewhat Autonomous Machine). This month we'll continue this project to give him some more intelligence and a lot more personality. - by Oliver Higgins

SAM's original brief was to build a basic autonomous robot that could sense an obstacle, react and then continue on its way. SAM achieved this objective, but we now have a new brief; be more aware of its surroundings and add some more functionality.

In last issue, we did solve the problem of basic obstacle avoidance, the resulting solution was quite abrupt. He would sense something and make an immediate retreat before turning around. This got us off the ground and used some simple code to achieve the desired result; however, there were some obvious issues. While he never hit anything (OK, almost never), the avoidance was very crude; we need to design a much better way for SAM to side-step an object and not lose his momentum. ››





›› HOW IT WORKS

ULTRASONICS

One of the main issues we faced after SAM's inception was that while he worked very well at undertaking his mission, he was not particularly smart when it came to driving. The only input was a very simple question: "is there something in front of me and how close is it?" And his response was "if it's close enough, then I'm going to turn and run in a different direction".

The two issues SAM faces is that he does not always drive particularly straight, nor is he aware of his surroundings. We can combat this in a number of ways: if we were using him outdoors we could implement a GPS unit that will provide a constant heading, allowing SAM to use his ultrasonic sensor to avoid things in front of him; or we could use a magnetometer to give us a heading to follow, but the magnetic interference from the motors can cause inconsistent readings. Alternatively, we could add a pan motor and allow him to move the ultrasonic scanner from side to side, although this would be quite complex to engineer and code.

A simple and cost effective solution is to add two more ultrasonic sensors. We will set this up as a polled operation, and it will continue to scan the surrounding environment to see if there are any objects nearby.

30° V 45° V 90°

This revision has placed two additional ultra sonic transducers on SAM's sides, to enable him to see what is to each side of him, and to allow him to side-step objects rather than stop, back up and turn around. We tested 30, 45 and 90-degree angles and found that the 45 seemed to give the best compromise. Be mindful that to detect an object it needs to be parallel to the ultra sonic sensor to reflect the sound waves back. We have included some variables that can be changed to help refine this process.

CONTROL LOGIC

SAM's original logic tree was simple: if an object gets with 20cm then stop, wait, back up, wait, turn, wait, then go forward. While this met the brief, it is fairly simple in its logic. The new brief we received was to make him avoid objects by navigating around them. The logic remains the same for the obstacle avoidance, with his front facing sensor, but we have the addition of the two extra sensors. Using these we scan each side individually and react based upon an obstacle being present outside of the range of the main sensor. We then use this information to slow one set of the motors on the chassis enabling the turn. This is achieved in a very simple way. The main loop now has three conditions: the first is a simple "if" statement for the object in front, which triggers the stop and reverse. The next condition is a while loop. This loop checks the left-hand sensor, and if anything is within the set range, then the right-hand pair of wheels are slowed down to 25%, thus instigating a side-step arc around the object. If it does not detect anything, it checks the right-hand sensor, repeating the while loop if required, before looping back to the first statement. Each while loop is governed by the distance to the object, so while ever the ultrasonic sensor in question finds an object too close, it will continue to make adjustments.

```
while (pingLeftSonar() <=
                      objectDistanceLeft) {
  Serial.print("L:");Serial.
                  println(pingLeftSonar());
  slowLeft();
}
while (pingRightSonar() <=
                      objectDistanceRight) {
  Serial.print("R:");Serial.
                  println(pingRightSonar());
  slowRight();
}
```

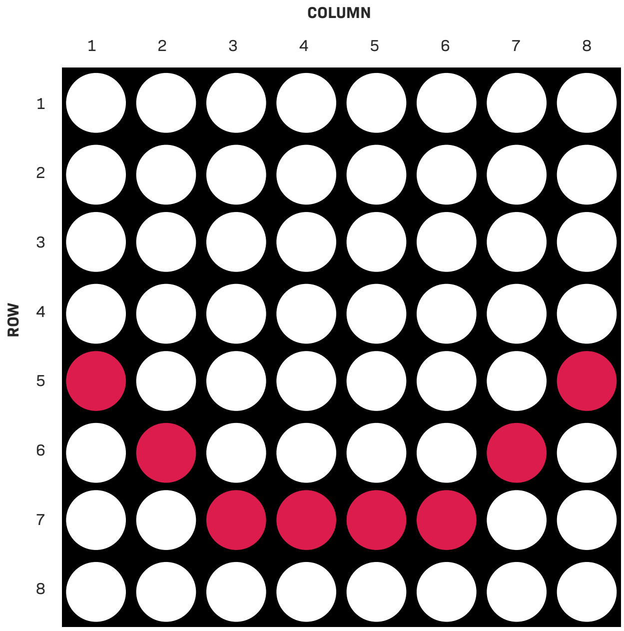
The above code calls the pingLeftSonar() during each while loop. The function returns the current distance that the ultrasonic sensor is reading. If it is less or equal to the defined distance variable, we call the slowLeft function, which will slow the motors and cause SAM to make a side-step or turn. The slowLeft function (below) is identical to the forward function with the exception of the analogWrite (ENA, motorSpeed) line. In this line, we have used a simple multiple to slow the motor down – 0.25 will reduce the motor to 25% of the motorSpeed variable.

The other addition to all of these functions is the printByte function, which we will discuss next. This is repeated for slowRight function, but the multiplier is added to the analogWrite (ENB), thus slowing the alternate motor pair on the chassis.

```
//slow Left
void slowLeft() {
  printByte(hmmmL);
  Serial.println("SL");
  digitalWrite(IN1, HIGH); // Turn HIGH motor
A
  digitalWrite(IN2, LOW);
  analogWrite(ENA, motorSpeed * 0.25);
  digitalWrite(IN3, HIGH); // turn HIGH motor
B
  digitalWrite(IN4, LOW);
  analogWrite(ENB, motorSpeed);
}
```

LED MATRIX

To add in a bit more personality, we have added an 8x8 LED matrix array, which gives SAM an injection of personality. We have programmed him with a series of expressions – happy, sad, annoyed and indifferent, and each expression is triggered in the various drive functions. So when he is moving forward, we call the happy icon. The matrix can be thought of as a grid of LEDs. Now if we were to connect all of them directly to the Uno it would take up 64 I/O lines, which is more than the Uno has available! However, in this case, the matrix also contains a matrix driver. The MAX7219 (part of the 72XX series) is a serial interfaced, eight-digit encoder. This makes the setup and execution of 8x8 icons easy.

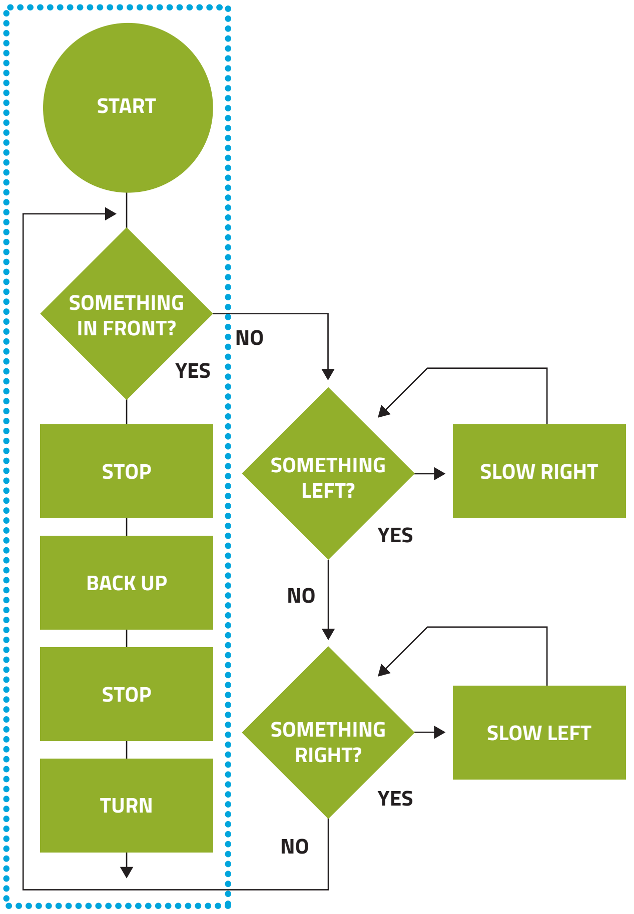


5 = 0 x 81 = 10000001
6= 0x42 = 010000010
7= 0x32 = 00111100

The LED matrix can be thought of as a series of eight rows of 8 LEDs. We feed the data to the matrix one byte at a time. Each byte represents 8 bits, each one being an LED on or off.

To declare our icons or emotions, we use an array of 8 bytes, where each byte contains 8 bits. Each byte here is represented by a hexadecimal (base 16) value. If you converted them to a decimal (base 10), you would get a normal number, but if you convert them to base 2 or binary, then you will get a series of bits. Each one of these bytes represents a row in the matrix. The first four rows are blank, the fifth row contains the first byte of data that will change the state of the LED row.

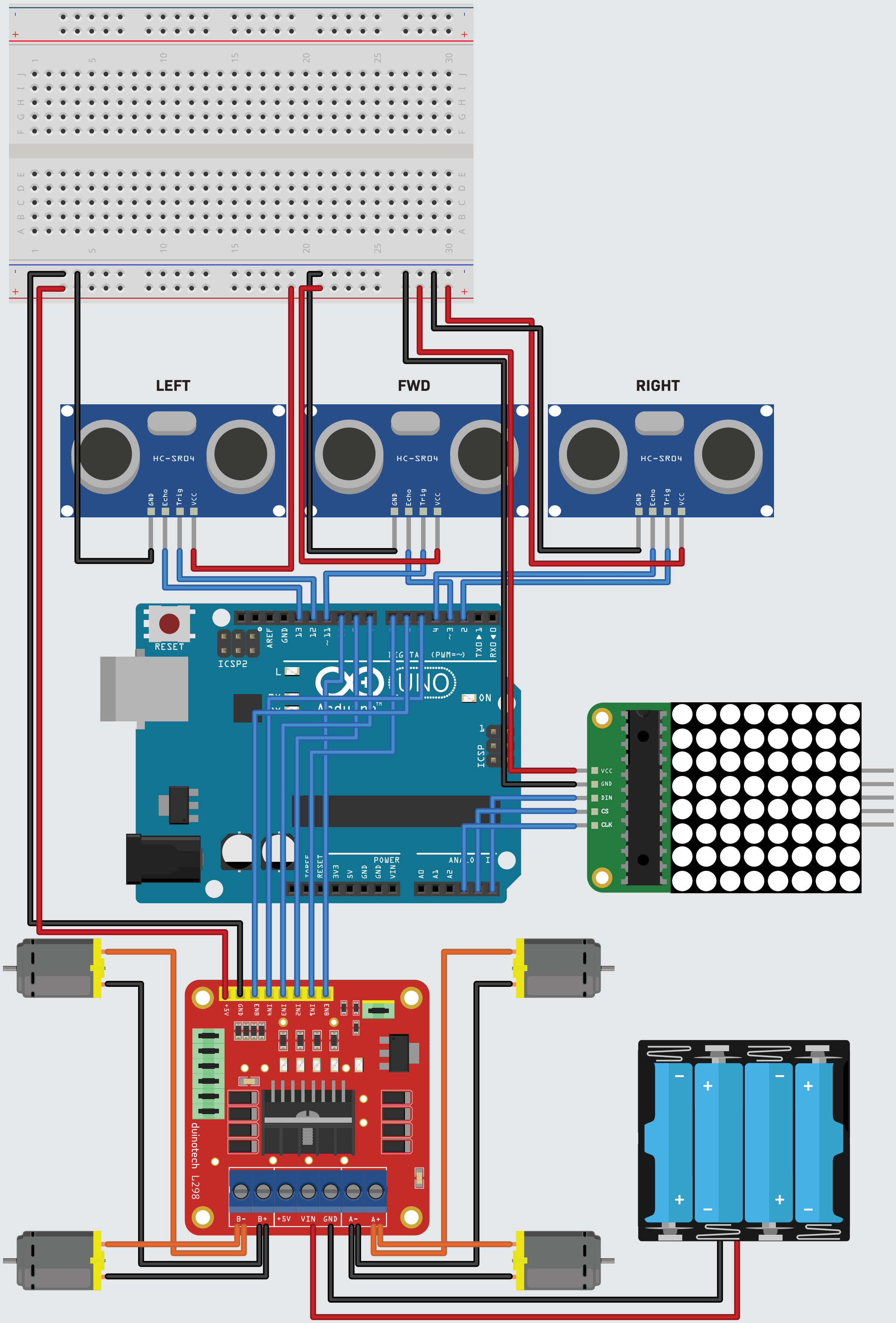
```
byte grin[8] = {0x0, 0x0, 0x0, 0x0, 0x81,
0x42, 0x3c, 0x0 };
0x81 = 10000001
```



To send this to the matrix-encoding chip we send it one byte at a time, and each time we index the row that we want it to be. To do this, we use a simple for loop, and loop through it eight times, since as we have eight rows.

```
void printByte(byte character [])
{
  int i = 0;
  for (i = 0; i < 8; i++)
  {
    lc.setRow(0, i, character[i]);
  }
}
```

If you wish to make your icons or patterns check out this generator web app project at *https://github.com/bernhard-hofmann/LedByteGenerator* ››


LEFT
FWD
RIGHT
HC-SR04
HC-SR04
HC-SR04
GND
Echo
Trig
VCC
RESET
ICSP2
AREF
GND
ON
ICSP
POWER
VCC
GND
DIN
CS
CLK
duinotech L298
B-
B+
+5V
VIN
GND
A-
A+

›› THE BUILD

ORIGINAL SAM PARTS LIST FROM ISSUE #3:

PARTS REQUIRED:	JAYCAR	ALTRONICS
1 x Arduino UNO	XC4410	Z6240
1 x Ultrasonic Module	XC4442	Z6322
1 x L298n Motor Control Module	XC4492	Z6343
1 x Chassis & Motor Kit	KR3162	K1092
1 x 8AA Battery Holder	PH9209	S5034
8 x AA Batteries	SB2333	S4955B

PARTS LIST FOR V1.1:

ADDITIONAL PARTS REQUIRED:	JAYCAR	ALTRONICS
2 x Ultrasonic Modules	XC4442	Z6322
1 x 8x8 LED Matrix Module	XC4499	Z6362
1 x Updated 3D Printed Caddy	-	-

BUILDING THE CIRCUIT

If you have already built the first version of SAM, then the hook up is quite simple. We have two additional ultrasonic sensors to add to the chassis, in addition to the 8x8 LED matrix.

As the unit is based on the UNO R3, we will be very close to having every GPIO pin used by the end of this project. As such we have run out of GND and 5V pinouts. A simple solution is to use a small breadboard which has power rails to expand the connection points available.

The chassis has been modified to allow the additional pair of Ultrasonic sensors and have them set off at a 45° angle relative to the main sensor. We have also raised the main sensor up high to allow for the LED matrix underneath. Please note that due to the addition of the extra ultrasonics and the angle they sit on, it is now very difficult to program the Arduino Uno while it is attached to the chassis (the USB port is obscured). For this reason, make sure you have the jumper on the motherboard off, and program the UNO before you attach it. The LED matrix is a very snug fit if you are using the module. Due to the encoder IC, you will find the unit just goes past flush on the print. You may find bending the input pins will also help to ensure clearance. The two additional ultrasonic sensors are much easier to work with if you place them upside down, and they will work just as effectively in either orientation.

THE CODE / SETUP / OTHER

SAM includes the addition of three new functions: 1) to deliver our emotions to SAM's LED matrix; and 2) to change the motors speed to allow them to side-step any objects. To facilitate the LED matrix, we need to include a new library that is specifically designed to work with the MAX7219 and MAX7221 (*https://github.com/wayoda/LedControl*). This has been included in the download package. Please extract and place in the libraries folder of your Arduino IDE installation.

In setting SAM up, you may need to fine tune the sensitivity settings. You can do this by adjusting the three object distance variables. These are integers and are representative of centimetres. SAM V1 had the objectDistance variable set at 20cm. In testing we found that the longer distance would prevent the side sensor from triggering so, we have set the defaults at 15cm for the front and 30cm for the side pods. This should mean that any object off to the left or right should trigger the function to make him side-step the object before the main sensor causes him to halt. This is very dependent on the range and angle of the object itself.

```
// defines variables
long duration;
int objectDistance = 15;       //distance in cm
int objectDistanceRight = 30; // distance in cm
int objectDistanceLeft = 30;  // distance in cm
int motorSpeed = 128;         // 0-255
```

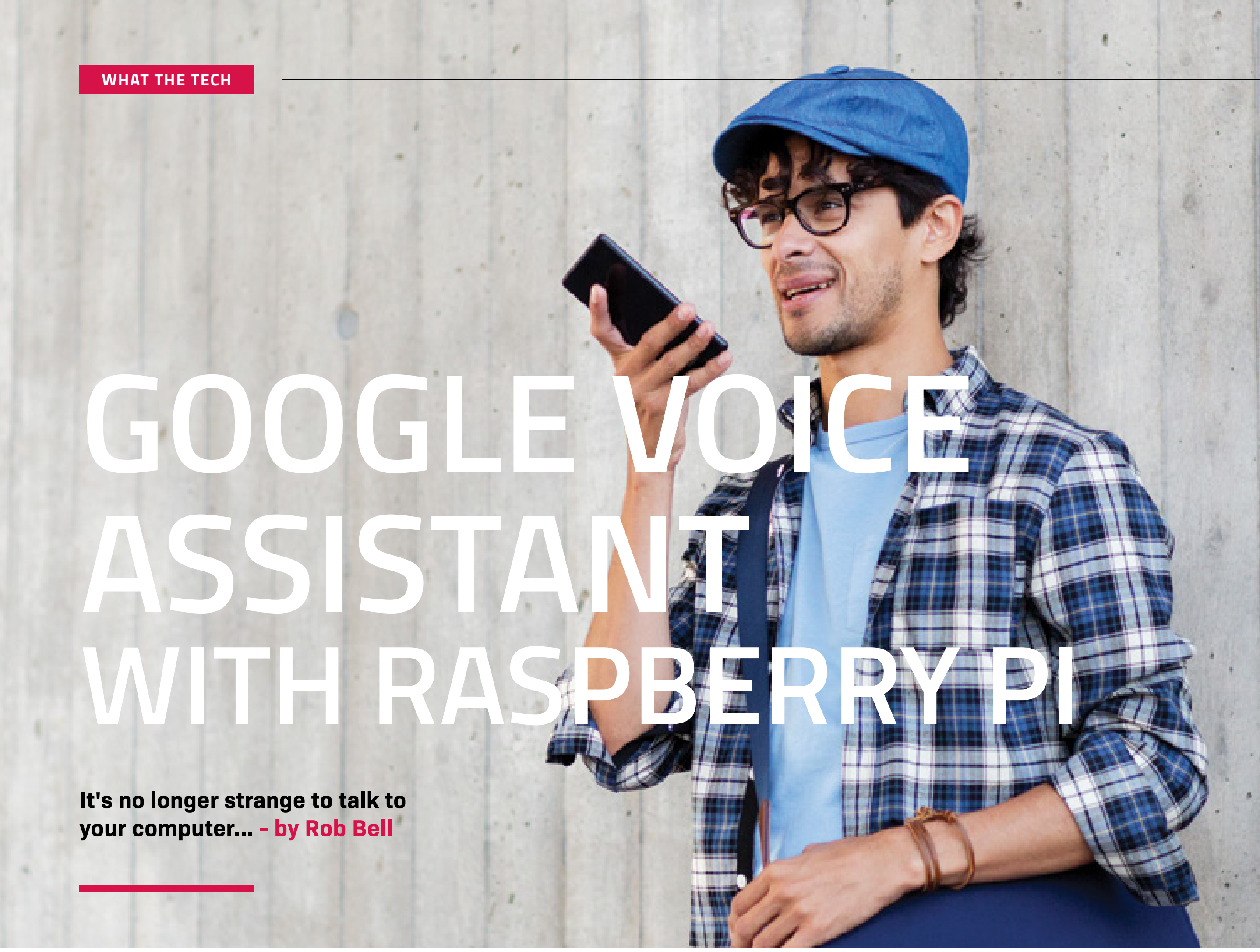
Note: make sure the batteries are fresh. Low batteries may result in false signals with the ultrasonic sensors.

WHERE TO FROM HERE?

We are starting to reach the physical limits of the Arduino Uno, with only three GPIO pins left to use. We will address this in the next iteration. However, the software provided with SAM is quite simple. Currently, the code is very block-driven; it checks a sensor and immediately implements a function. This means that it will always favour an object on the left before it evaluates an object to the right. A more intelligent way to address this would be to get the data from all three sensors and evaluate them based on values and tests before implementing an action for avoidance. What if there is an object left and right? SAM could evaluate the distance and adjust the speed multiplier on the fly, to create smaller movements and keep on track. ■

WANT MORE?

For additional media, code files and 3D modelling, or to discuss this project, visit:
https://diyode.io/004kmyy



GOOGLE VOICE ASSISTANT WITH RASPBERRY PI

It's no longer strange to talk to your computer... - by Rob Bell

As a kid growing up, I was fascinated by the potential for voice control. In fact, even VOX (voice operated relay) CB radios had me intrigued. Such a simple thing captivated my imagination from a young age.

In the 90s I was fortunate enough to get to experiment with what was rather advanced software for its time. Not only could it translate speech to text (which seemed amazing in itself), but it could open and close applications, and control the operating system in some amazing ways. Of course, back in the day, it was shareware, probably on CD (I think we'd progressed past floppy drives, but not as far as CD burners). Out of curiosity I recently ran a Google search for INCUBE, and it returned a few parts of the internet that needed dust blown off them! But I digress...

Back in those early days of the internet, multimedia was an emerging buzzword. But there were a few serious limitations: RAM was counted in the kilobytes, soundcards had limited quality, computers had clock speeds in megahertz - the technology was really letting the concept down. It was ahead of its time, and for many years voice technology stayed on the fringe.

Now, we know how clear and accurate voice commands can be. Sure, we could create a feature film around what Siri doesn't understand, but the reality is that this technology works exceptionally well. Whether it's a Google, Apple, or Amazon product, voice commanded technology is improving every day. With the inclusion of machine learning, we're going to continue to see improvements at a rapid pace, while the voice command blooper reel will continue to shrink.

HOW CAN YOU LEVERAGE VOICE CONTROL?

Google has long provided API access to some of their greatest technologies, which is fantastic. They develop ideas with billions of dollars of backing, that we could rarely dream of achieving on our own; rather than making the system proprietary and hiding it behind an expensive corporate fee,

Sure, just as INCUBE did in the 90s, it is indeed possible to create a useful, voice recognition system that's self-contained. Though if I recall correctly, INCUBE required configuration of voice files to get things running, so it could learn your tone of voice. I don't think it required individual voice commands to be recorded for every single action, but there was a reasonable level of setup (it was cutting-edge at the time though, let's not forget that).

This is where the Google Assistant SDK comes in. You can leverage it on just about any piece of hardware capable of running Python with fantastic speed and accuracy.

GOOGLE ASSISTANT WITH RASPBERRY PI

Where hardware-interfacing computers gain a huge advantage, it's clearly the GPIO. Raspberry pi is a logical choice for a Google assistant project, but in reality you can use any board you like which can run Python. As long as you have WiFi access to your pi, you could create a voice-commanded robot to follow your every command.

If you're really game though, all you need is internet access, so it is possible to use a 3G module or similar if you need access outside of a WiFi network. The data demands are reasonably low so you're not going to blow through your data any faster than a few cat videos and LOLs will!

But let's just consider what ramifications this has for hardware. You can effectively voice control ANYTHING. Lights on? Sure. Door open? Definitely. Turn on the TV? You bet. Of course, the applications need to be ones that aren't critical or where safeguards need to be employed. For example, if the voice assistant mishears you and suddenly your robot is running away forever, it's probably not running so fast that you can't catch it. There are loads of creative ways to use it, where failure of the system won't have any catastrophic implications.

CREATING COOL STUFF

We have our first project this month using the Google Assistant. It integrates MQTT protocol, so you can remotely control anything you could action with the MQTT system. However next month we'll look at some more simple voice-command functionality that you can get started with, in order to play around with the functionality. Even from an experimentation process, simple functionality such as turning an LED on or off, playing music via the raspberry pi's audio output, and many other simple projects take on a whole new life when voice commands are added.

WHAT ABOUT THE MIC?

Those of you familiar with the Raspberry Pi will know it doesn't have a hardware-level native microphone input. This doesn't present much of a problem however. USB sound cards are an easy go-to option. You can also use a USB webcam you might have lying around, which often has an audio input. Google Assistant won't access the camera portion, but it provides a USB audio input for your Raspberry Pi - problem solved.

GETTING WILD

You know what I would really like to do? Create two virtual assistants and initiate a conversation between them. Even if we have a little code to help them out with particular questions and answers.

Why? I think a more appropriate answer is "why not?"

While it's unlikely to yield anything substantial (it's a computer, remember), it's fun to hear two machines talking to each other, even if for a brief period and about nothing in particular.

What would be more interesting to investigate however, is creating a set of environment sensors and conditions which create a platform of discussion for these two sensorless devices. Providing them with real world input about the current temperature and humidity, could we then spark a conversation about the weather?

Of course, we can add real questions and answers, and follow-up questions as part of the Google SDK; coding a false-intelligence into them is achieved rather easily. But this isn't really the point, and represents no artificial intelligence outside of the precise speech patterns themselves.

WHERE IT'S HEADED

As the technology improves and continues to evolve, it's likely that we'll see more comfort and social acceptance around voice-commanding computers. Even though it's more commonplace now, talking to your phone when there's not an actual human on the other side of the conversation can still feel a little awkward when there are others around. Saying "hi mate, how are you?" is still far more socially acceptable than "OK Google, what's the weather like where I'm going?".

This paradigm of machine-response to speech is becoming ubiquitous to machine interactions however. So commonplace, that within a few years you can probably "tell" the parking meter that you need to pay for parking, and it'll know who you are from your speech patterns.

There's no telling when the general social attitude to talking aloud to your devices will be come as universally accepted as talking to another human via your phone, but I'd suggest that we're closer than you think. Until then though, go ahead and create a raspberry pi virtual assistant, and have a play with the system. It's easier than you think, and loads of fun! ■

GOT SOMETHING TO SAY?

To discuss this edition of What The Tech, visit:
https://diyode.io/004vscx

Reading & Resources:

- **GOOGLE AIY PROJECT**
  https://aiyprojects.withgoogle.com/voice/
- **GOOGLE ASSISTANT SDK**
  https://developers.google.com/assistant/sdk/

THE ADVENTURES OF CIRCUIT MODD