RasPi

DESIGN
BUILD
CODE

14

Get hands-on with your Raspberry Pi

Plus
WIRELESS PRINTING with Pi

PROGRAM
ROBOTS

6 DIY kits & challenges

Welcome

Following its huge success last year, Pi Wars is back again this December. In case you missed it, the Cambridge Raspberry Jam organised an amazing event that saw people bringing in their home-made robots and then taking on a number of challenges designed by the Pi Wars organisers, including speed and power tests, obstacle courses, a three-point turn test and more. It was huge fun and we're excited for Pi Wars 2015 – and if you want to get involved too, we're here to help. Check out this month's feature to find six brilliant robotics kits, each one put to the test with a new challenge that you can download the code for. And if you do decide to take your robot down to Cambridge, send us a photo!

Editor

Get inspired

Discover the RasPi community's best projects

Expert advice

Got a question? Get in touch and we'll give you a hand

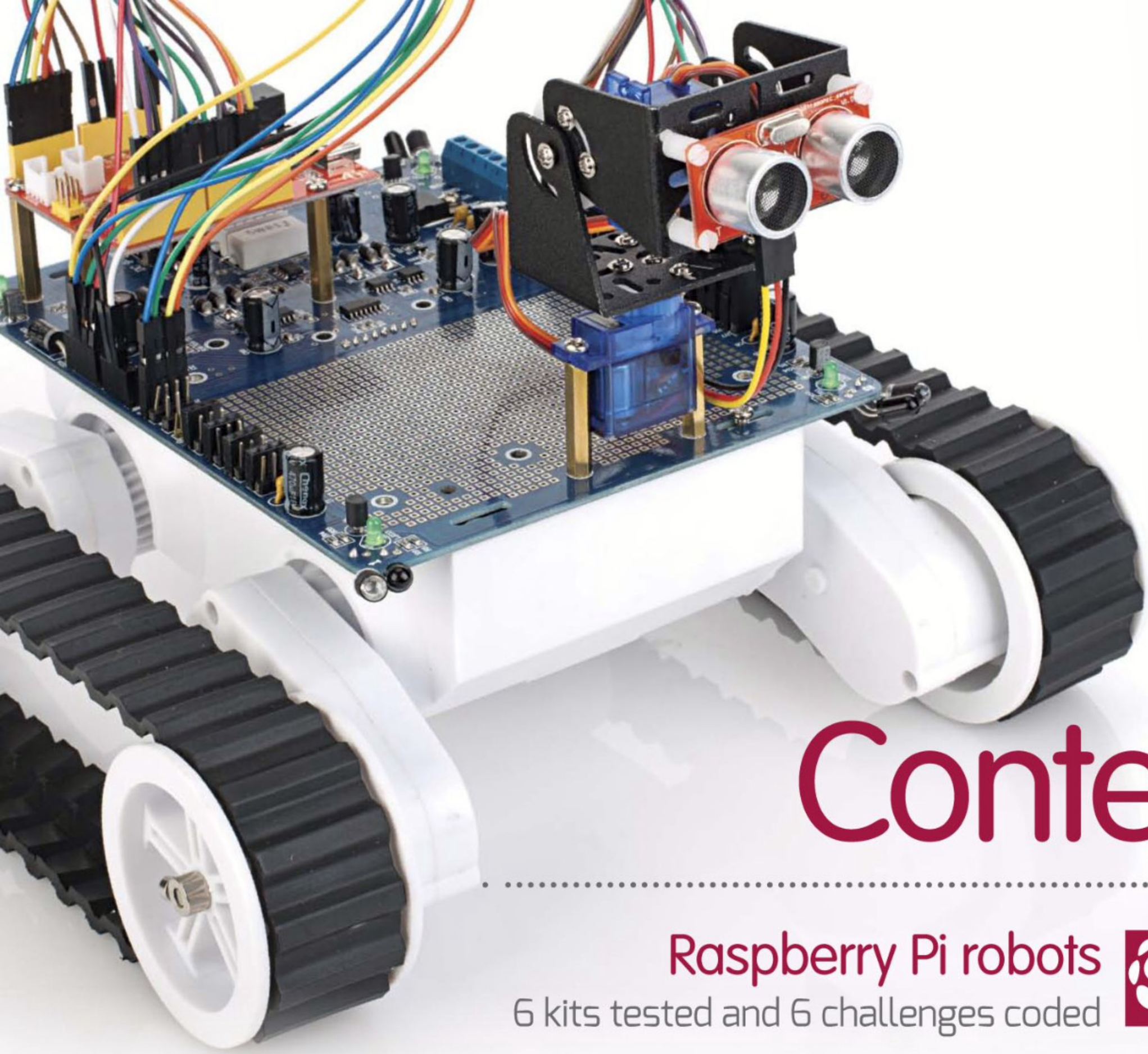
Easy-to-follow guides

Learn to make and code gadgets with Raspberry Pi

From the makers of **LinuxUser & Developer**

Join the conversation at...

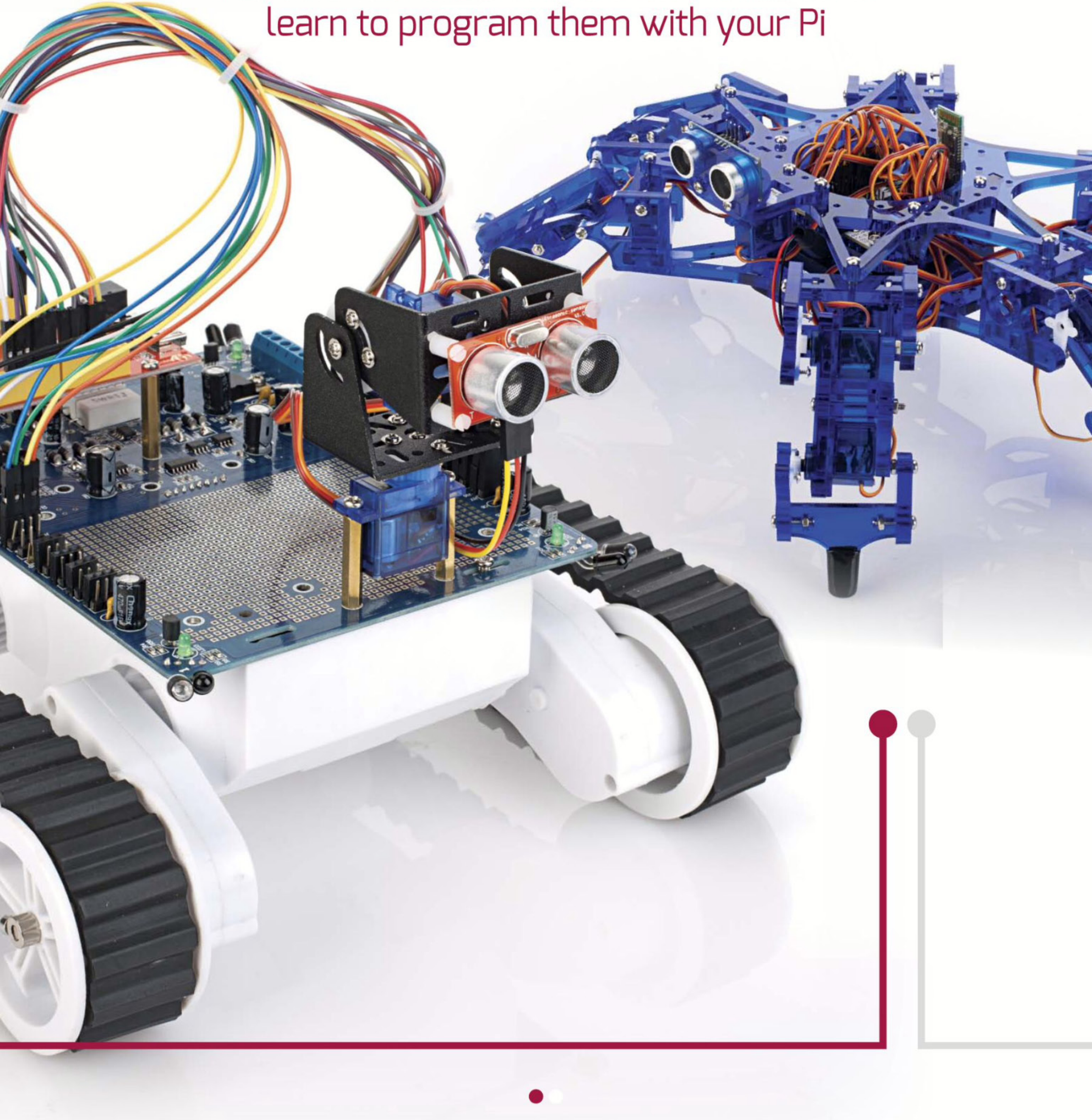
@linuxusermag | Linux User & Developer | RasPi@imagine-publishing.co.uk



Contents

Raspberry Pi robots

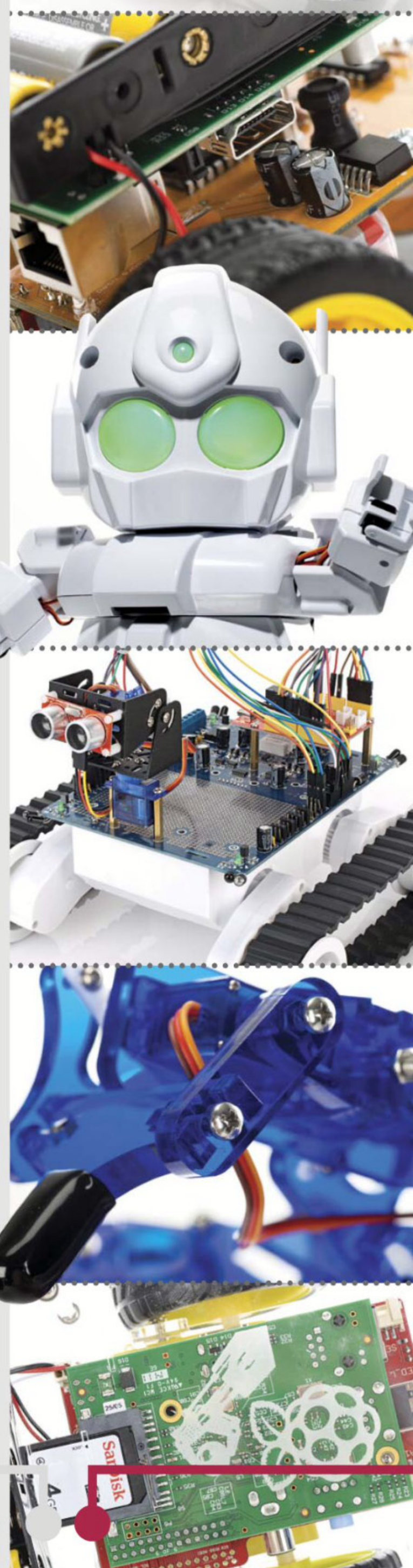
Discover the best robotics kits around and learn to program them with your Pi

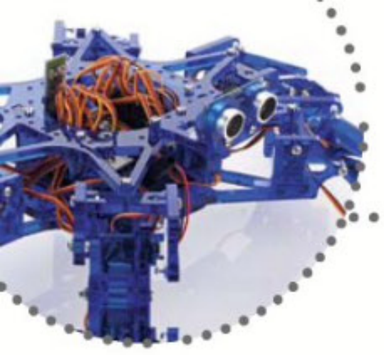


The rise of our robot overlords is well underway – give it another five years and we'll all be watched over by Pi-powered machines of loving grace. In the meantime, though, we've rounded up the very best DIY robotics kits available to buy right now that are designed to work with your Raspberry Pi, so you can get a head start on the inevitable revolution. Whether they're Arduino or Raspberry Pi-based, we're getting all of our robots to listen to our master Pi controller and showing you how to do the same with your kit. We'll also be scoring these robotics kits to identify their strengths and weaknesses in terms of their build quality, functionality out of the box, the construction process and of course their programmability, to help show you which kit is right for you and where you can get hold of your own.

And what then? Well, we thought we'd put our robots to the test with a series of challenges inspired by the Cambridge Raspberry Jam's Pi Wars event (**www.piwars.org**) – a line-following challenge, a proximity alert test, a tricky obstacle course and a three-point turn examination. Not content to stop there, we also got one of our robots to play a fine round of golf and another two to battle each other (sumo style).

So it's time to introduce you to our hand-picked team of robots – over the next few pages you'll meet Rapiro, our most humanoid and delightfully articulate Gundamesque robot; GoPiGo and Pi2Go, two nippy little two-wheel tricars with ball-bearing casters for stability at the rear; Frindo, the sensor-loaded, open source mobile robotics platform; Rover 5, the rugged two-track tank with a Seeeduino brain and an inexorable top speed of 1km/s; and Hexy, the six-legged, crab-walking, Thriller-dancing (**http://bit.ly/1lj2CqR**) force of robotic nature.





Rover 5 Seeeduino

A relative monstrosity, the Seeeduino is fully kitted out and makes a great gift

If you recall issue 2, where we made our first robot, you'll remember that the kit we used to build it was from Dawn Robotics – the Rover 5 is sort of a successor to that kit. The Rover 5 is a lot larger and generally has a few more functions than that particular Raspberry Pi robot. Said Pi is not needed for the Rover 5 as it is fully powered by the Seeeduino, another ATmega 328P. Construction is not the easiest and requires an extra hand at times. There's no soldering involved but there is

TECH SPECS

Manufacturer
Dawn Robotics

Height
170 mm

Width & depth
225 x 235 mm

Weight
1.05 kg

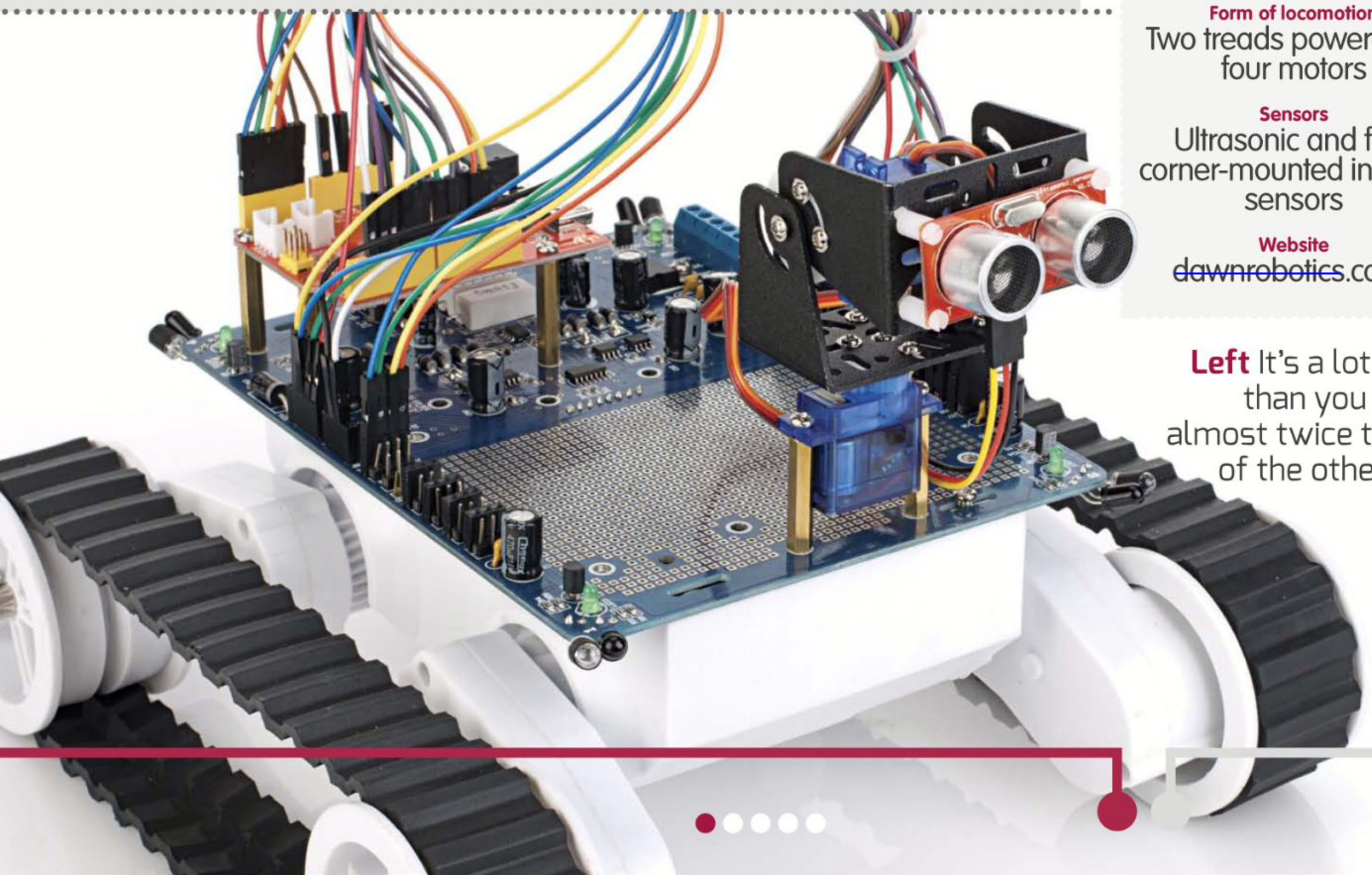
Power
9 volts from 6 AA batteries

Control board
Seeeduino Arduino (ATmega 328P)

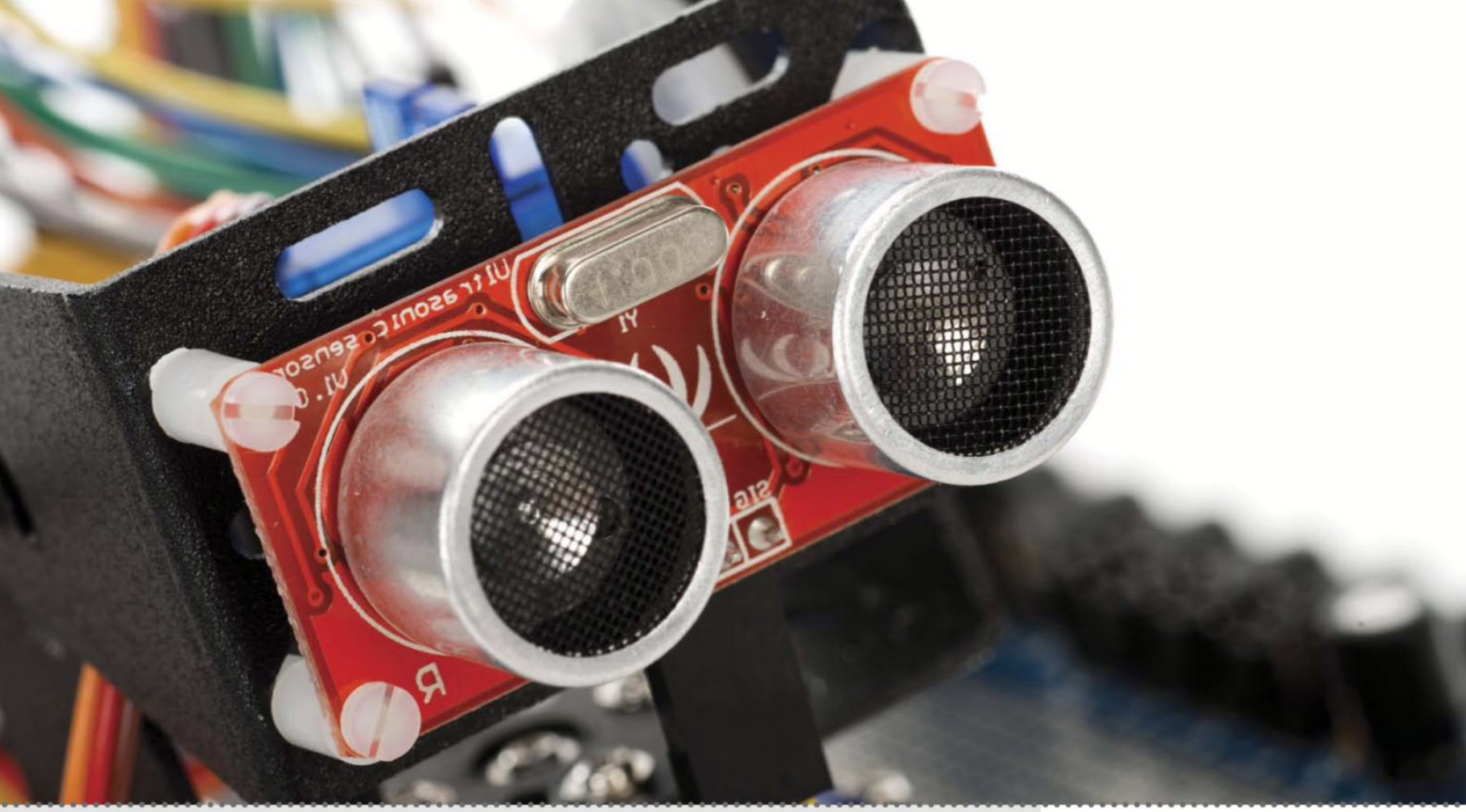
Form of locomotion
Two treads powered by four motors

Sensors
Ultrasonic and four corner-mounted infrared sensors

Website
dawnrobotics.co.uk



Left It's a lot bigger than you think – almost twice the size of the other bots!



Above The ultrasonic sensors enable the Rover 5 to sense distances to objects

an enormous amount of wires that connect up the board. Couple this with some fiddly nuts and bolts, a sometimes unhelpful manual, the odd cheap screw and you get a few problems that take a bit of lateral thinking to solve. The whole kit is a mix of different components manufactured separately, which explains some of the discrepancies in the screws and how cobbled together the entire thing is.

The big board sits on top of the Rover 5 and is quite well suited for the kit, but it does contribute to the DIY, mashed-together look of the Rover 5 with all the wires flying around. Programming it is slightly harder than other robots due to using pure Arduino rather than having serial commands or Python functions. You'll need to get right into the code to start programming, but there are preset tutorial scripts that give pointers on how to create your own code. With the sensors on each corner of the Rover 5, the board and robot can react to almost any obstacle thanks to their coverage, not to mention the ultrasonic sensor also attached to it.

HOW IT SCORED

Assembly

Build quality

Programmability

Functionality

Best for Big projects

The challenge: Obstacle course

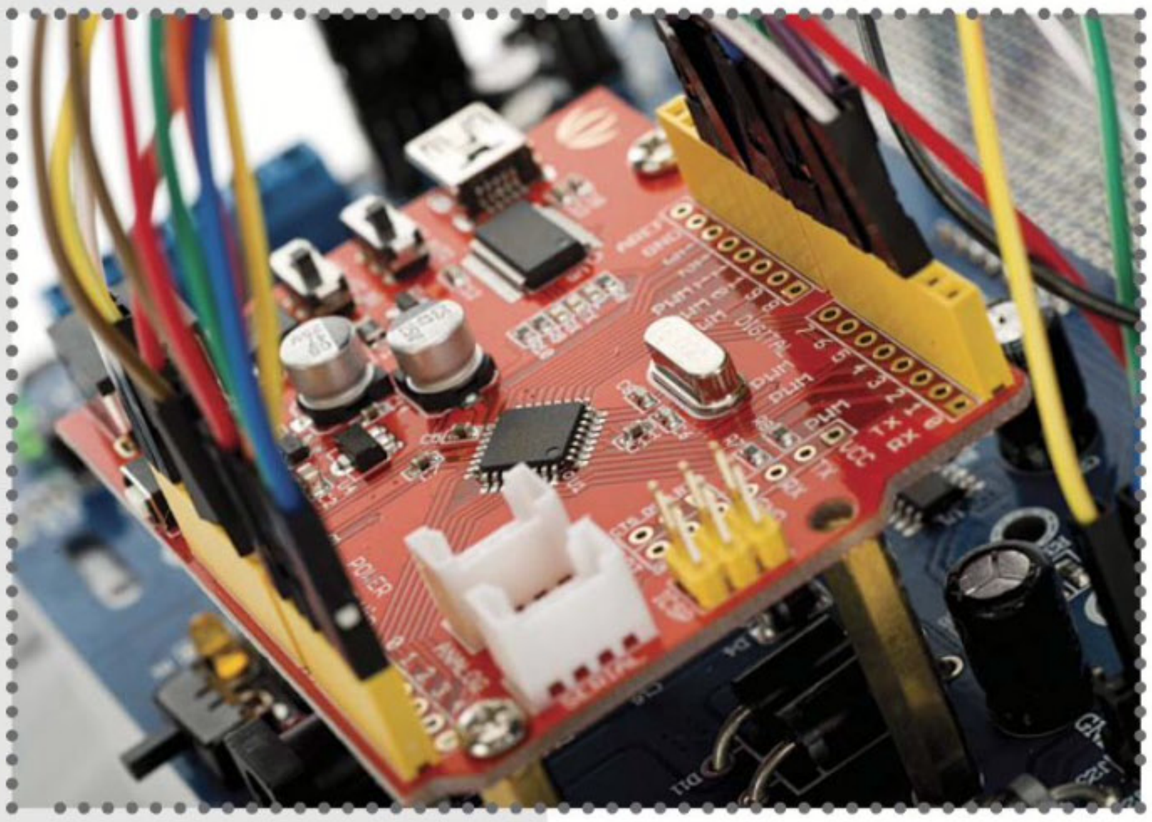
The challenge they have set at Pi Wars is for remote-controlled bots, but we're going to change the rules slightly and rely a bit more on automation. In our scenario, we're using a walled maze that runs a specific yet random course that the robot needs to navigate without any kind of lines to guide a robot with line sensors. It's a little more tailored to the Rover 5's capabilites, but how so? Well, the Rover 5 is perfectly equipped to handle pretty much any course we can throw at it. Thanks to its array of proximity sensors, it will know if there's a wall in the way around its body. We can also use the ultrasonic sensor to figure out its distance from the wall and the way that the wall will change as you travel.

There's some great code for the Rover 5 that allows you to follow a wall along the right of the robot. You can grab it now from **http://bit.ly/1vp2LLZ**. Upload it via Arduino; it's a bit of a long task, but we will help you out by explaining some parts of it here. Firstly, there are a lot of setup integers created to begin with. These include defining minimums for speed, wall proximity and defining the sensors in general.

Next, we have a small part of the sensing code. All the readings are taken and turned into useful data for the rest of the code – in this case, integers. After that, we have one of the various movement sections. This is used to just move forward, looking to see where an obstacle/wall will be and beginning the chain of events that occurs after noticing or bumping into a wall. This will include backing up, turning left and turning right.

Finally, the code ensures that the sensor keeps an eye on the wall as it runs along it to make sure it's still there.

"The Rover 5 is perfectly equipped to handle any course"



Below The main control board connects to the rest of the robot and is easily accessible to add more components

The Code OBSTACLE COURSE

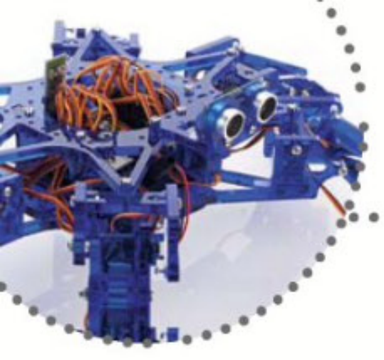
Get the code
bit.ly/1uQzsNa

```
const int NUM_IR_SENSORS = 4;
const int IR_LED_PINS[ NUM_IR_SENSORS ] = { A0, A0, A1, A1 };
const int IR_SENSOR_PINS[ NUM_IR_SENSORS ] = { A3, A2, A4, A5 };

...

float ultrasonicRange = gUltrasonicSensor.measureRange();
    gRoverIRSensors.takeReadings();
    int frontLeftIR = gRoverIRSensors.lastFrontLeftReading();
    int frontRightIR = gRoverIRSensors.lastFrontRightReading();
    int rearLeftIR = gRoverIRSensors.lastRearLeftReading();
    int rearRightIR = gRoverIRSensors.lastRearRightReading();

...

case eRS_DrivingForwardsLookingForWall:
        {
            // Check to see if we've hit an obstacle we didn't see
            if ( gLeftMotor.isStalled()
                || gRightMotor.isStalled() )
            {
                enterBackingUpState();
            }
            else
            {
                // Check to see if we've found a wall
                if ( ultrasonicRange <= CLOSE_ULTRASONIC_RANGE
                    || frontLeftIR >= CLOSE_RANGE_IR_VALUE
                    || frontRightIR >= CLOSE_RANGE_IR_VALUE )
                {
                    enterTurningLeftState();
                }
            }
```

The Code OBSTACLE COURSE

```
                break;
            }

...

void enterFollowingWallOnRightState()
{
    // Point the ultrasonic sensor to the right
    gPanAngle = LOOK_RIGHT_PAN_ANGLE;
    gTiltAngle = LOOK_RIGHT_TILT_ANGLE;

    gPanServo.write( gPanAngle );
    gTiltServo.write( gTiltAngle );

    gLeftMotor.clearStall();
    gRightMotor.clearStall();

    gLeftMotor.setTargetRPM( BASE_WALL_FOLLOWING_RPM );
    gRightMotor.setTargetRPM( BASE_WALL_FOLLOWING_RPM );

    gStateStartEncoderTicks = gLeftMotor.getLastMeasuredEncoderTicks();
    gStateStartTimeMS = millis();
    gRobotState = eRS_FollowingWallOnRight;
}
```

"Finally, the code ensures that the sensor keeps an eye on the wall as it runs along it to make sure it's still there"



Pi2Go Lite

One of the smallest robots in our test, yet the Pi2Go has a few tricks

The Pi2Go Lite is a very interesting little bit of kit. Coming in a tiny little box and utilising no chassis, in favour of construction via its PCBs, you'd think it would be a super simple robot that follows commands and doesn't really react much to the environment. It makes it sound like a remote control novelty more than anything else. That couldn't be further than the truth, as the Pi2Go is probably the most featureful robot in this feature.

TECH SPECS

Manufacturer
4tronix

Height
90 mm

Width & depth
130 x 145 mm

Weight
0.40 kg

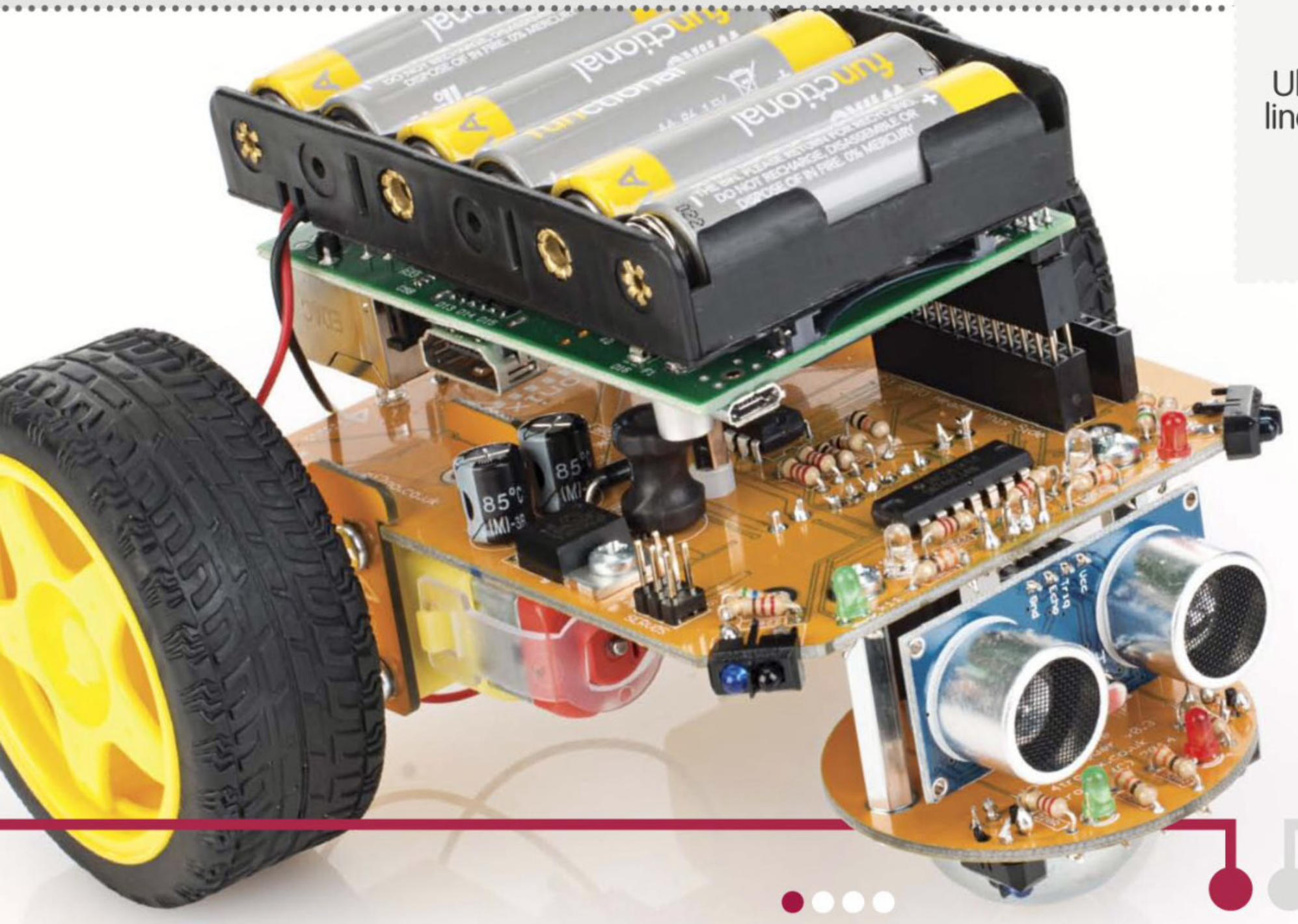
Power
9 volts from 6 AA batteries

Control board
Raspberry Pi

Form of locomotion
Two-wheel drive

Sensors
Ultrasonic sensor, two line sensors and two IR obstacle sensors

Website
pi2go.co.uk



Left The Pi2Go is actually one of our favourite robots – it's a lot of fun!

All this functionality comes at a price though, as it's the only robot that requires a lot of soldering and pre-preparation before construction. You'll need to be a bit handy with a soldering iron to do it, although you don't need to strip any wires and such. There are about 50 components to fit, possibly more, and it can be a little time-consuming. The instructions are not extremely helpful, but the individual components are actually listed on the PCB as a rough guide to where things should be fitted. Once the soldering is done though, you just need to put the few parts together to complete it. The website lists a 90 minute construction time, but we found it took somewhat longer – it was no longer than any of the bigger or more complicated robots on the other pages though.

It's a sturdy, compact little thing and it's powered purely by the Raspberry Pi via a custom Python library. Sensing, turning on the LEDs, activating the motors and other physical functions have their own corresponding Python functions. It lets you create scripts that can make it fully autonomous, as long as the autonomy only requires distance, line and proximity sensing to operate. At least it can take extra timed or web information from the Raspberry Pi if that's set up correctly.

For the price, functionality and relative ease of programming, it's a fantastic piece of kit that's great for getting into starter-level robotics and slightly beyond. Some soldering skills required though.

"Sensing, turning on the LEDs, activating the motors and other physical functions have their own corresponding Python functions"

The challenge: Line following

Line following is very easy: you put a line on the floor and you expect the robot to follow it. This includes turning as well, following a course accurately to its destination or to accumulate laps. The Pi2Go Lite is the only robot we're

HOW IT SCORED

Assembly

Build quality

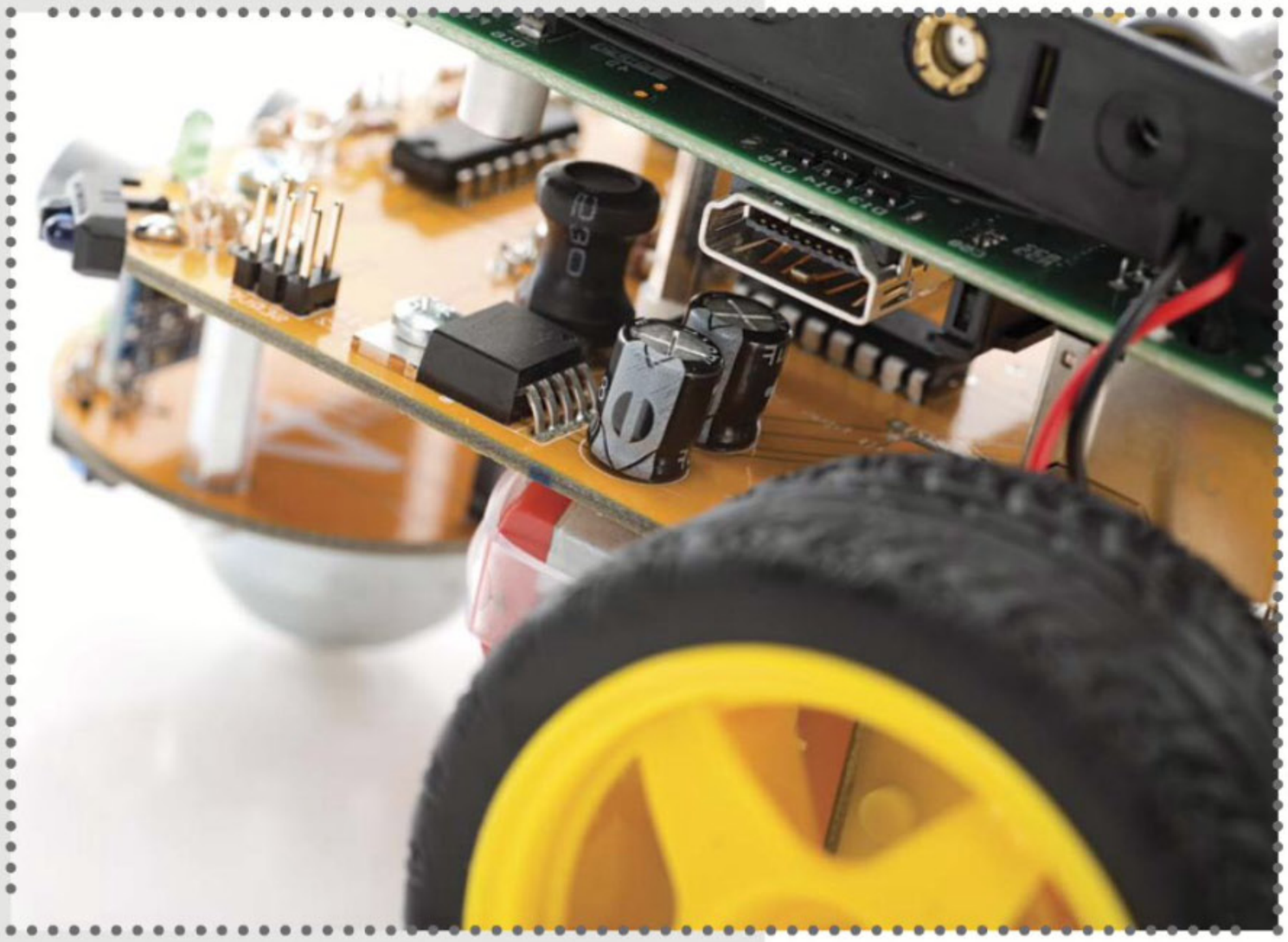
Programmability

Functionality

Best for Makers

looking at this month which comes with line-following sensors, although it is the main unique feature. Sounds like it should be quite simple then, however there's no line-following function in the Python script so we need to build a script for it.

As we said, the solution involves the line-following sensors – these are IR sensors located on the underside of the smaller PCB where the caster sits. We'll assume we've placed the Pi2Go down on the line and you want to go straight forward along it.



Above The PCB makes up the bulk of the chassis with each of the components fully visible

One of the problems we're going to run into is that the motors will likely not run at the exact same speed – with a bit of trial and error you can maybe fix this in the first forward command, but for now we'll keep it at 50, which is 50 per cent of its full speed. You can tweak this to be faster or slower as you see fit.

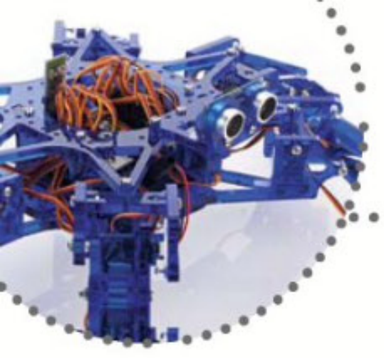
The loop is quite simple: it sees if any of the line sensors are activated. As we're assuming that the line is under the caster wheel, we'll need to correct course in a specific direction for each true statement. You can set the individual speed of the motors (left and then right in the turnForward function), and then we have it pause a bit before returning to full speed.

The code ends when you stop it and cleans up the GPIO port settings before exiting. The code requires the pi2go Python files, which you can grab from the 4tronix website: **http://4tronix.co.uk/blog/?p=475**.

The Code LINE FOLLOWING

Get the code
bit.ly/1z1REHW

```
import time, pi2go

pi2go.init()

pi2go.forward(50)

try:
    while True:
        if pi2go.irLeftLine() = True:
            pi2go.turnForward(45, 50)
            time.sleep(2)
            pi2go.forward(50)
        elif pi2go.irRightLine() = True:
            pi2go.turnForward(50, 45)
            time.sleep(2)
            pi2go.forward(50)
    else:
            time.sleep(0.5)

except KeyboardInterrupt:
    print

finally:
    pi2go.cleanup()
```



Hexy the Hexapod

The Kickstarter success story with six legs, 19 servos and some mad dance moves

We were really impressed by this all-in-one kit that lives up to its Kickstarter promise of being easy to assemble for people of any skill level, including beginners. Everything is neatly packaged and there's even a tiny screwdriver (though to be fair, those servo horns ended up breaking ours, but more on that later).

For the most part the instructions are excellent but there were a couple of occasions where a slight lack of clarity

TECH SPECS

Manufacturer
ArcBotics

Height
100-140 mm

Width & depth
300-400 x 200mm approx (depending on stance)

Weight
0.90 kg

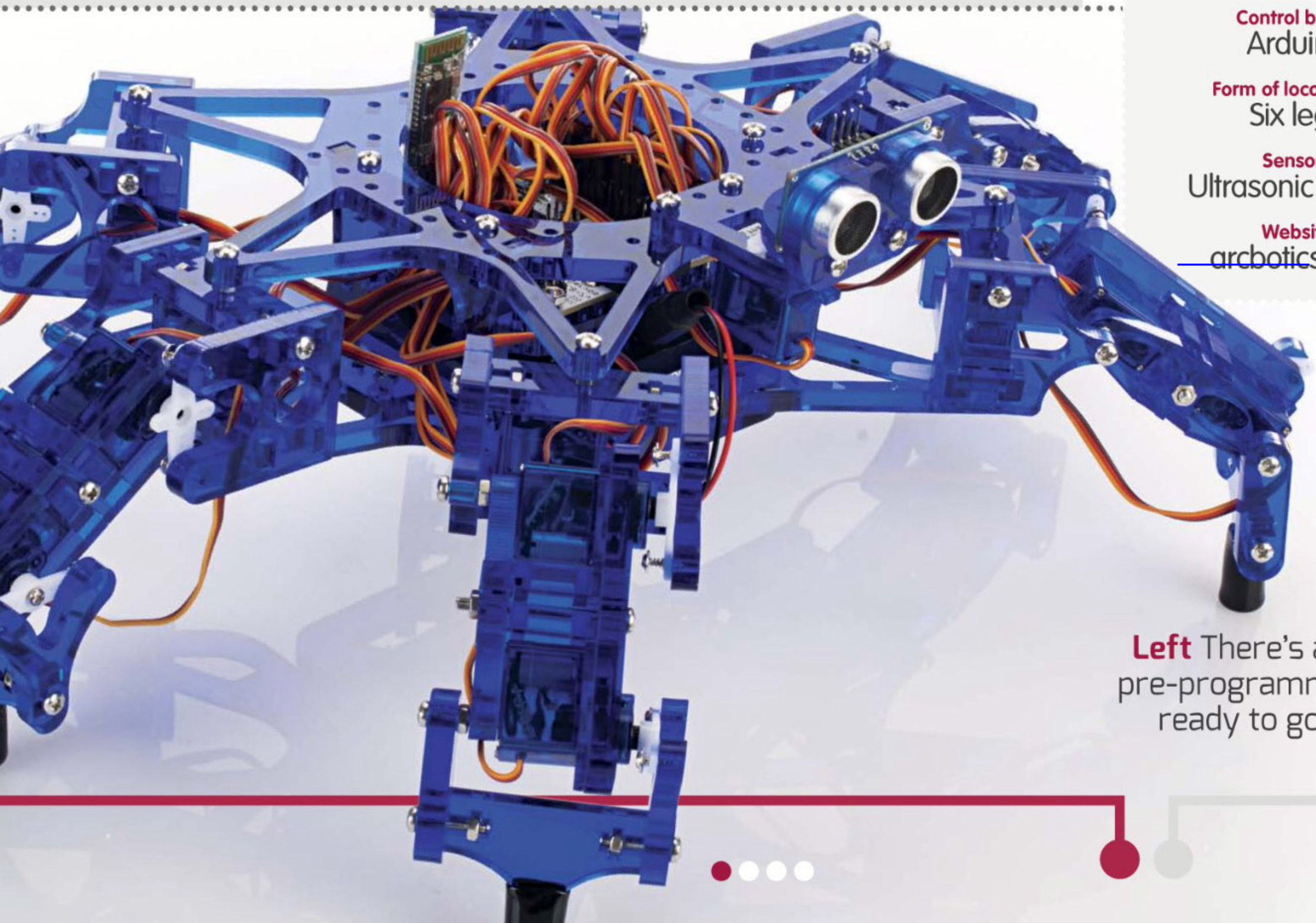
Power
6 or 7.5 volts from 4 or 5 AA batteries

Control board
Arduino

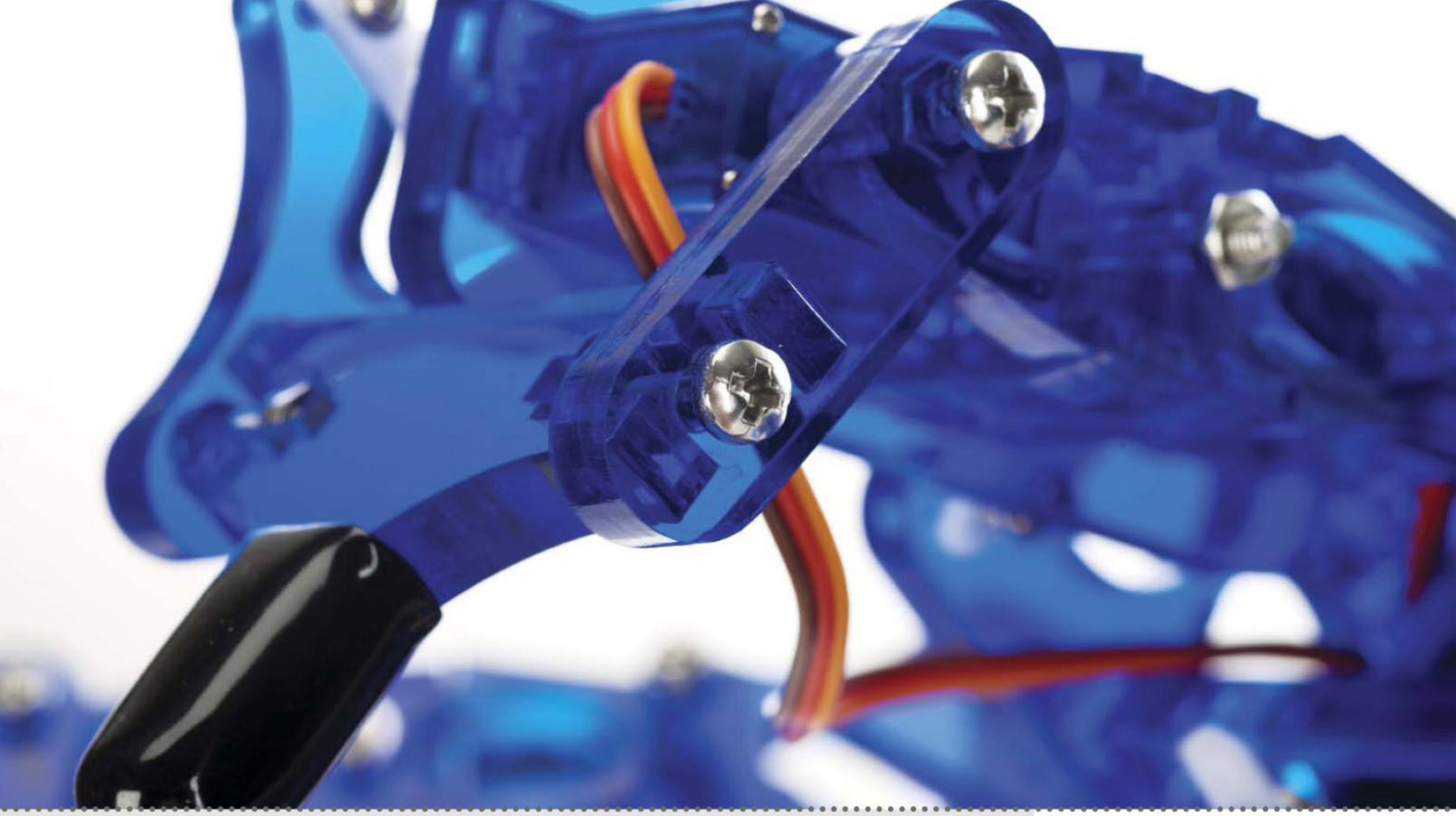
Form of locomotion
Six legs

Sensors
Ultrasonic sensor

Website
arcbotics.com



Left There's a library of pre-programmed moves ready to go with Hexy



meant that we just followed the images instead, though they were generally spot-on and very useful. You can see the thought that's gone into it, from the strong, lightweight plastic to the razor-sharp design. The wiring instructions are perfect and the software tutorials are really useful – you can get an Arduino IDE set up and also dive straight into PoMoCo, ArcBotics' position and motor controller software that's preloaded with actions (including dance moves).

There's only one real criticism of this kit – the screws are all wrong. There is a big bag of various size screws provided but you don't even use a quarter of them, instead being forced to open each individually packaged servo and borrow the medium screws from those, as the small holes on the servo horns are far too tiny for the recommended screws. The slightly smaller ones from the servo packs fit, so we used those, but you still have to widen the pinholes with brute force. It brings the otherwise speedy build process to a total halt, but all in all, Hexy is worth the trouble.

Above There are three parts to each leg and each part contains one servo. This could potentially enable Hexy to climb over small objects

HOW IT SCORED

Assembly

Build quality

Programmability

Functionality

Best for Mobility

The challenge: Three-point turn

This is usually a tricky challenge for robots, especially if it has to be done autonomously like in Pi Wars. The challenge requires the robot to walk out of a designated area and travel just over two metres before performing a three-point turn in an area only 750mm deep. When done, it must then return to the starting area. To do this classically, you'd need to know the speed and distance of your robot and travel with extreme accuracy to make the 180 degree turn easier.

The Hexy has an advantage in that it can literally spin on the spot, or at least shuffle its way around. To do this we made a very simple script where the Hexy walks a few 'steps' forward before attempting a full 180 degree turn and doing the same number of steps back to its starting position. We'll go over some parts of it here but you can grab the full thing from via the link on the next page.

First we define a few basic parameters: the way the Hexy will walk and some of its speed parameters. We've also got a rotation parameter which we've set to 180, but you may need to tweak it for your Hexy. There's also the steps variable created just to make the code slightly easier. Next we create a loop where for the first and last ten steps, the legs are articulated in order to make the Hexy move forward. This is a quarter of the walk forward section of the code, and once all parts have been completed we increase the step value by one. When it has reached ten steps, we do a load of code like in the last part to perform a full 180 degree turn, and then it does ten steps back with another if statement stopping the loop when a further 20 steps have been made.

> "Walk out of a designated area and travel just over two metres before performing a three-point turn in an area only 750mm deep"

Below The bluetooth sensor on the Hexy provides an easy way to connect wirelessly

The Code THREE-POINT TURN

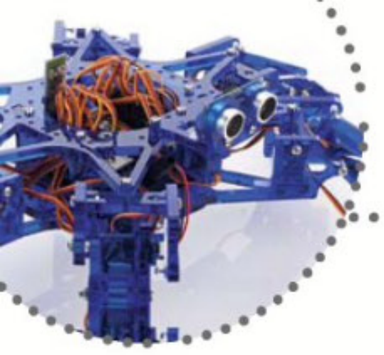
```
deg = 25
midFloor = 30
hipSwing = 25
pause = 0.5
rotate_deg = 180
rotate_step = 30
steps = 0
rotations = 0

...

While True:
   if steps != 10:
      # re-plant tripod2 forward while tripod1
      # move behind
      # re-plant tripod 2 forward
      hexy.LF.replantFoot(deg-hipSwing,stepTime=0.5)
      hexy.RM.replantFoot(hipSwing,stepTime=0.5)
      hexy.LB.replantFoot(-deg-hipSwing,stepTime=0.5)

...

else:
      # set neck to where body is turning
      hexy.neck.set(rotate_step)
      # re-plant tripod1 deg degrees forward
      for leg in hexy.tripod1:
         leg.replantFoot(rotate_step,stepTime=0.2)
      time.sleep(0.5)
      # raise tripod2 feet in place as tripod1
      # rotate and neck
      for leg in hexy.tripod2:
         leg.setFootY(int(floor/2.0))
      time.sleep(0.3)
```

Get the code

bit.ly/129HXMK

The next step

This step-and-turn code can easily be adapted into a full-on catwalk routine, with tilts, leans, belly flops, audience-pointing and even a dance routine, should you wish to go all-out. Just download the PoMoCo source code available at **http://bit.ly/1ykuLQF** and work those Python chunks into your main script wherever you like.



Frindo

The puck robot with a low profile and plenty of front-facing sensors

The Frindo is sold more as a robotics platform than an actual all-inclusive robot on its own. Out of the box you can do a fair bit with it, and it's extremely easy to build upon thanks to its support of Arduino and Raspberry Pi boards. Construction is straightforward and quick, although you will have to solder on wires to the motor during the construction. The chassis construction and fitting of the motors, boards and wheels is quickly done and with very few components.

TECH SPECS

Manufacturer
Frindo

Height
85 mm

Width & depth
160 mm diameter

Weight
0.55 kg

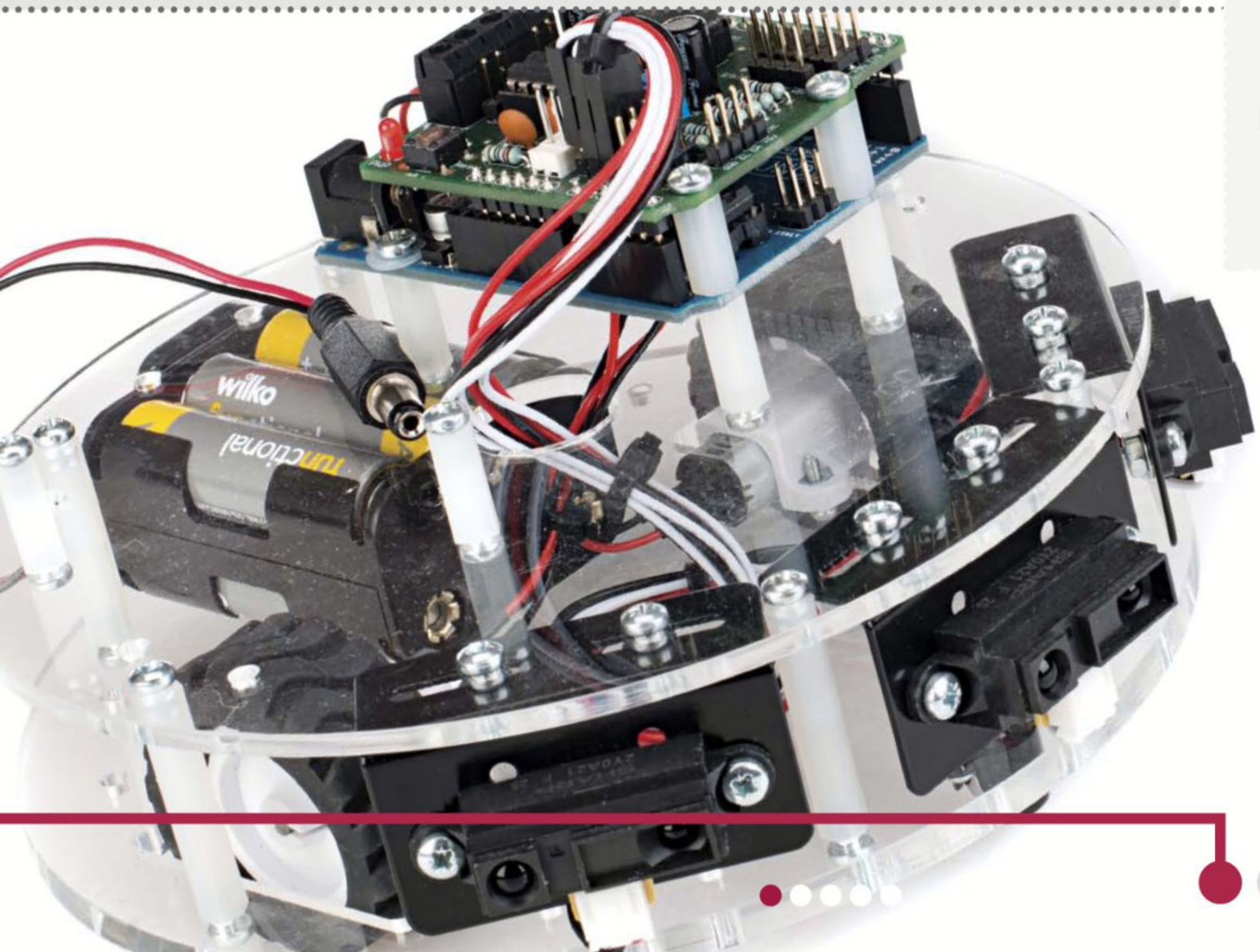
Power
9 volts from 6 AA batteries

Control board
Arduino and/or Raspberry Pi

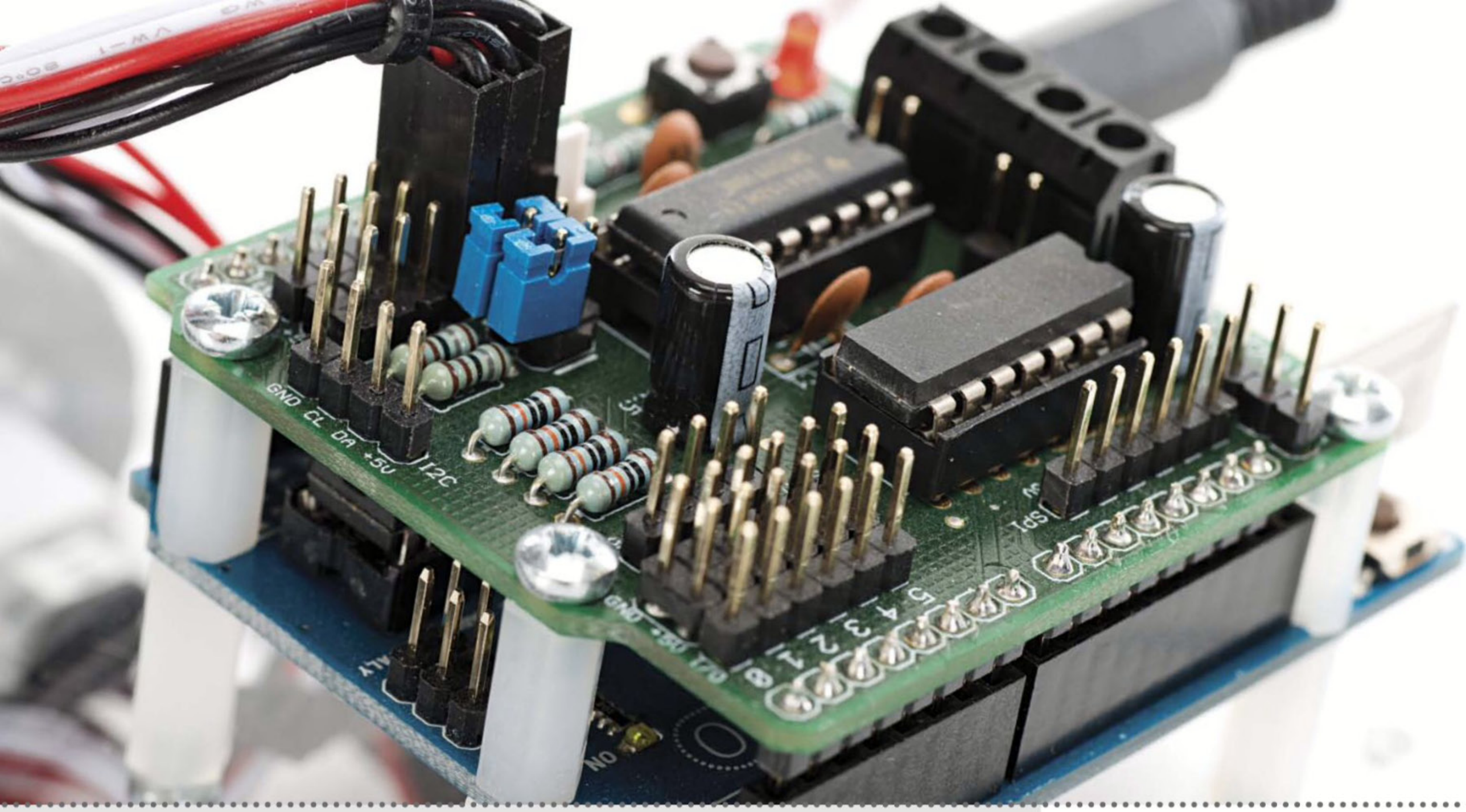
Form of locomotion
Wheels

Sensors
Four infrared proximity sensors

Website
robotbits.com



Left Hugely customisable, the Frindo is great for advanced projects



You can add a Pi to the system either with or without the supplied Arduino. This can be mounted on the opposite side of the board using holes specifically cut for the original Model B (though unfortunately not the B+ or 2B). There's room for four more proximity sensors, attachable in the spaces between the back and front sensors for complete 360-degree coverage. The Uno and Pi can take a lot more inputs and outputs as well, so adding components is easy.

Due to the dual controller support, the Frindo can be programmed in both Python and the Arduino IDE. Arduino uses the standard libraries and commands, making it great for those already up to speed with Arduino. The Python program uses the serial library, which uses terminology similar to Arduino, and there's a good, basic example on the website that can help you understand exactly how the sensors and motors can be operated in this fashion.

The Frindo is the most accessible robot we have here. Very simple yet very good, and excellent to learn with.

Above The Robot Shield has been donated to the Frindo project as an open-source version

HOW IT SCORED

Assembly

Build quality

Programmability

Functionality

Best for ~~Hackers~~

The challenge: Proximity sensor

This challenge is somewhat simple: drive right up to a wooden wall and stop before hitting it. The closer you are before you stop, the more points you get. No touching of the wall is allowed. Should be easy with all those proximity sensors, right? Well it's not as easy as you would think, as the proximity sensor is not analogue. Surely there must be a way around it though?

The Frindo's sensors have some form of distance sensing on them, although it's by no means perfect. The other thing you'd have to calibrate for is program speed and stopping distance – and that's assuming you're heading straight on to begin with. The motors on the Frindo are unlikely to be in full sync, making it likely that you'll be heaving at a slight angle

That helps us in multiple ways as the Frindo has three sensors on the front, and we can use the left and right sensors to detect the extremes of the wall and turn the Frindo itself to get the perfect stop.

In the code snippets here, you can see that we first define what constitutes the Frindo stopping – this can be modified with trial and error to get a more accurate reading for your situation. The numbers do not correspond to a distance value. Next is one of the parts where we define how we look at the readings from the sensors so that they can be used in the final part. This rotates the Frindo as it finds any obstacles in its path. The full code for this script can be downloaded via the link on the next page.

"The Frindo's sensors have some form of distance sensing on them"

Below The sensors are all around the circumference of the Frindo, giving it amazing detection abilities and mobility

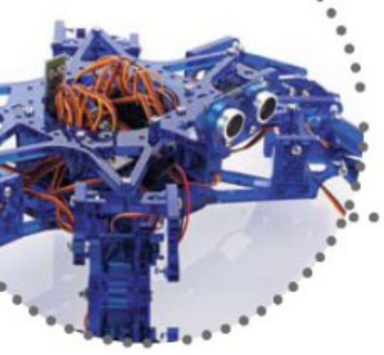
The Code PROXIMITY SENSORS

Get the code
bit.ly/121Xa38

```
int frontTrigger = 200;
int sideTrigger = 100;
int rearTrigger = 100;

...

int front_bump() {
            bump = analogRead(FrontBump);
            if(bump > frontTrigger){
                return 1;
                }
            else {
                return 0;
                }
            }

...

void loop() {

            Serial.println("Here we go...");

            while(!front_bump()){
            // while there is no bump keep going forward
            // (about 10cm with GPD120)

                if(!left_bump() && !right_bump()) {
                    Serial.println("NO bump detected - move forward");
                    rs.forward(500, 200);
            // move forward for 500 mS at speed 200
            // (200/255ths of full speed)
                }
```

The Code PROXIMITY SENSORS

```
            else if(left_bump() && !right_bump())  {
                Serial.println("LEFT bump detected - wrong angle");
                rs.rot_ccw(100, 200);
          // turn right for 100 mS at speed 200
          // (200/255ths of full speed)
                }

            else if(!left_bump() && right_bump())  {
                Serial.println("RIGHT bump detected - wrong angle");
                rs.rot_cw(100, 200);
          // turn left for 100 mS at speed 200
          // (200/255ths of full speed)
                }
          }
```



Rapiro

It stood up! The bipedal, humanoid, glowing-eyed, Arduino and Pi-powered robot straight out of Japan

The Rapiro is very unique on this list, even when compared to something like the Hexy. We were actually discussing in the office the difference in its design: Rapiro looks like a proper robot with its vacuum-formed shell, which in a way puts form over function. Not that it lacks function, but it's clear its creator Shota Ishiwatari fitted the motors around a design idea rather than design around the functions. It's a bit life-imitating-art, with Rapiro's design referencing robots in Japanese media compared to the hyperfunctional American and British robots with their ultrasonic sensors, line sensors and better stability that are more in line with Hollywood films.

Construction of Rapiro is quite simple; you attach the myriad motors to different parts as you assemble the shell around them and thread the wires into his chest cavity where the Arduino

lives. It's not really that fiddly, and there's no soldering or wiring involved. All the motors plug into the board using the straightforward labelling you're asked to do in the manual.

While the assembly manual is not written by a native English speaker, the repetition and illustrations are generally easy enough to follow along to. Connecting a Raspberry Pi is not covered in the manual, but the Wiki shows where the connections between the Arduino and the Pi should be made, while the mount points are pretty obvious while constructing the head.

Programming the motors and servos are quite easy, with a number of preset serial commands enabling you to create custom scripts for the Rapiro to move or react a certain way to different inputs. This kind of autonomy can be achieved by using the Pi and its camera to detect motion or specific objects, or respond to commands sent wirelessly to the board. It's not the most sophisticated robot on this test, however there's nothing else that can properly walk on two legs either, or grip things. It's unique and useful for different tasks in comparison to the wheeled robots in our selection.

Left You can pull off some surprisingly ~~delicate~~ manoeuvres

TECH SPECS

Manufacturer
Kiluck

Height
257 mm

Width & depth
196 x 159 mm

Weight
1.00 kg

Power
7.5 volts from 5 AA rechargeable batteries

Control board
Custom Arduino (ATmega 328P) with optional Raspberry Pi

Form of locomotion
Bipedal walking

Sensors
Support for Pi camera

Website
~~rapiro~~.com

The challenge: Robot golf

It's actually more of a putting challenge, with the robot tasked to manoeuvre a small ball across a defined space into a goal. The goal is of mouse-hole design, meaning it just needs to be pushed in. While this challenge was envisioned with wheeled robots in mind, we decided we could take it a step further and have the Rapiro knock the ball into the hole with some well-placed swings of a tiny and light gold club. Time is the measure of success, so how would the Rapiro best complete the challenge?

While the Rapiro has superb articulation, it doesn't really have the ability to adopt a traditional golfer stance. Its arms can't cross and it doesn't really bend down. So what we plan to have it to do is hold a golf club and twist its body to hit the ball – very simple, yet effective. Not particularly accurate though, but one step at a time.

You'll see an excerpt of the Arduino script we're using to control the Rapiro, using the test script you can grab. It allows you to set eight movements for the Rapiro to make – this includes the angle of the 12 servos listed in a specific order, the three RGB values of the light and the time the action takes.

In our code, the Rapiro's eyes turn purple (with the mixture of 100 red and 150 blue) and it raises its arm quickly. We have two of the same pieces of code both taking '1' unit of time for this to occur. After that it opens its hand and changes colour to green, giving you time to put a 'golf club' in its hand. It then grips it, turning its eyes yellow to let you know it's getting ready to swing. Finally, it twists its waist to swing the club. The full code is available to download, although you may have to tweak it to work with your Rapiro's calibrations.

"Programming the motors and servos are quite easy"

HOW IT SCORED

Assembly

Build quality

Programmability

Functionality

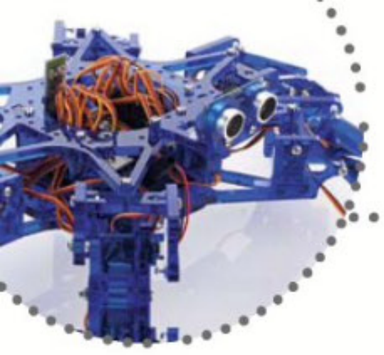
Best for Party piece

The Code ROBOT GOLF

Get the code

bit.ly/1HKBXeb

```
...
void setup()  {
  servo[0].attach(10);    // Head yaw
  servo[1].attach(11);    // Waist yaw
  servo[2].attach(9);     // R Sholder roll
  servo[3].attach(8);     // R Sholder pitch
  servo[4].attach(7);     // R Hand grip
  servo[5].attach(12);    // L Sholder roll
  servo[6].attach(13);    // L Sholder pitch
  servo[7].attach(14);    // L Hand grip
  servo[8].attach(4);     // R Foot yaw
  servo[9].attach(2);     // R Foot pitch
  servo[10].attach(15);   // L Foot yaw
  servo[11].attach(16);   // L Foot pitch
  eyes[R] = 6;            // Red LED of eyes
  eyes[G] = 5;            // Green LED of eyes
  eyes[B] = 3;            // Blue LED of eyes
  for( i = 0; i < MAXSN; i++) {
    targetAngle[i] = motion[0][0][i] << SHIFT;
    nowAngle[i] = targetAngle[i];
    servo[i].write((nowAngle[i] >> SHIFT) + trim[i]);
  }
  for(i = 0; i < 3; i++) {
    targetBright[i] = 0 << SHIFT;
    nowBright[i] = targetBright[i];
    analogWrite(eyes[i], nowBright[i] >> SHIFT);
  }
  Serial.begin(57600);
  delay(500);
  pinMode(POWER, OUTPUT);
  digitalWrite(POWER, HIGH);
}
...
```



GoPiGo

The simple and straightforward Pi project robot with WASD control

GoPiGo is one of the simplest kits in the array we're testing here – simple in a good way though, with none of the negative connotations. The promised 20-minute build time is no exaggeration and we were up and running with this robot in no time at all. With no soldering required either then this really is an ideal gift for anyone interested in putting their first bot together. Given the sub-$100 (£63.80) price point it also makes an excellent base upon which to build more advanced

TECH SPECS

Manufacturer
Dexter Industries

Height
85 mm

Width & depth
130 x 210 mm

Weight
0.60 kg

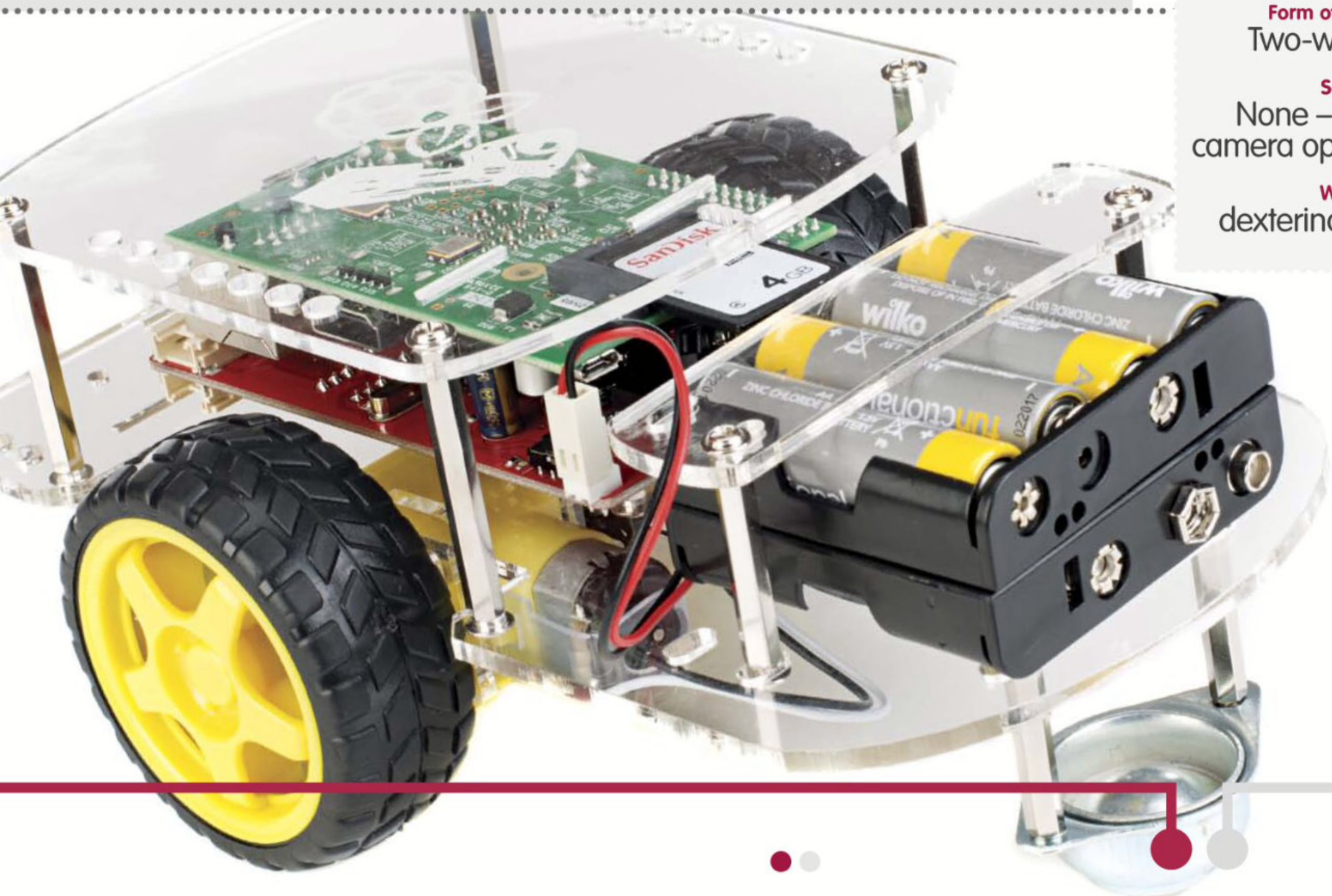
Power
12 volts from 8 AA batteries

Control board
Arduino (ATmega328) and Raspberry Pi

Form of locomotion
Two-wheel drive

Sensors
None – optional Pi camera optical sensor kit

Website
dexterindustries.com



Left This is pretty ideal robot for anyone who is learning robotics

projects, with plenty of room around the Pi and controller board within the open-sided shield for your own sensors and augmentations.

Straight out of the box, GoPiGo will work with Dexter Industries' firmware to give you WASD control of the two-wheel robot (the ball bearing caster at the rear making this a tricar of sorts), though nothing else beyond basic speed control. Being a Rasperry Pi-based project though, developing more advanced motion scripts and control for things like the optional ultrasonic sensor and camera module is a straightforward task.

There is one criticism to make, however: it seems there's a flaw with the design in that we found it impossible to connect the wheels properly. The wheels themselves simply slip onto the end of the axles, and can very easily be popped off with a quick knock. The short axle length and nuts that graze the inner tyre wheels mean that it's difficult to actually push the wheels far enough onto the axles to give you the confidence that it'll hold together while driving. But that aside, and given the otherwise sterling quality of GoPiGo, we still feel that this is one of our favourite kits.

HOW IT SCORED

Assembly

Build quality

Programmability

Functionality

Best for Makers

The challenge: Sumo battle

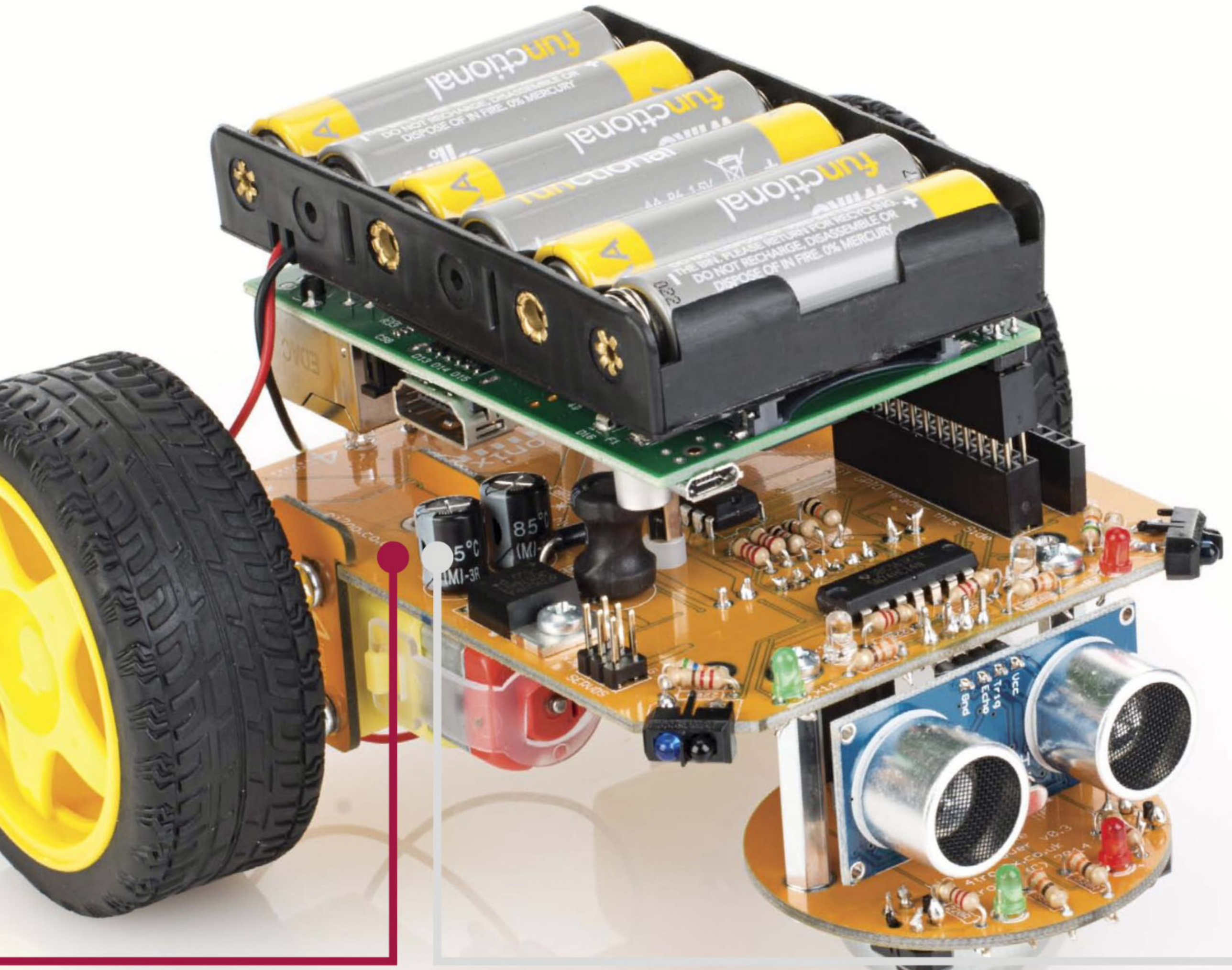
Our 'gentlerobots of strength' tested their torque in the fine tradition of sumo wrestling. The rules were simple, though slightly different to those of the more well-known Homo sapiens variant of this popular robosport. Matches could not be won by forcing the opposing automaton to touch the ground with any part of their body other than the soles of their feet – this would be impossible in most cases – but were instead focused on forcing them out of the dohyo (our tape-marked arena). It's a test of pure power, with each robot driving forth and attempting to push back the other.

Get the code

bit.ly/1ypCQod

What is a Raspberry Pi robot?

Is the Pi robot a specific product or just a concept? An easy answer for some, but not everyone knows the score



Q So you have some Raspberry Pi robots every now and then in the magazine. This may sound like a stupid question but… what is a Raspberry Pi robot?

A It's quite simply a robot that is powered, or can be powered, by a Raspberry Pi. Nothing too complicated to it – well, apart from actually making the robot and getting it to talk to the Pi. That can be very complicated.

Q So it's not a special model of the Raspberry Pi?

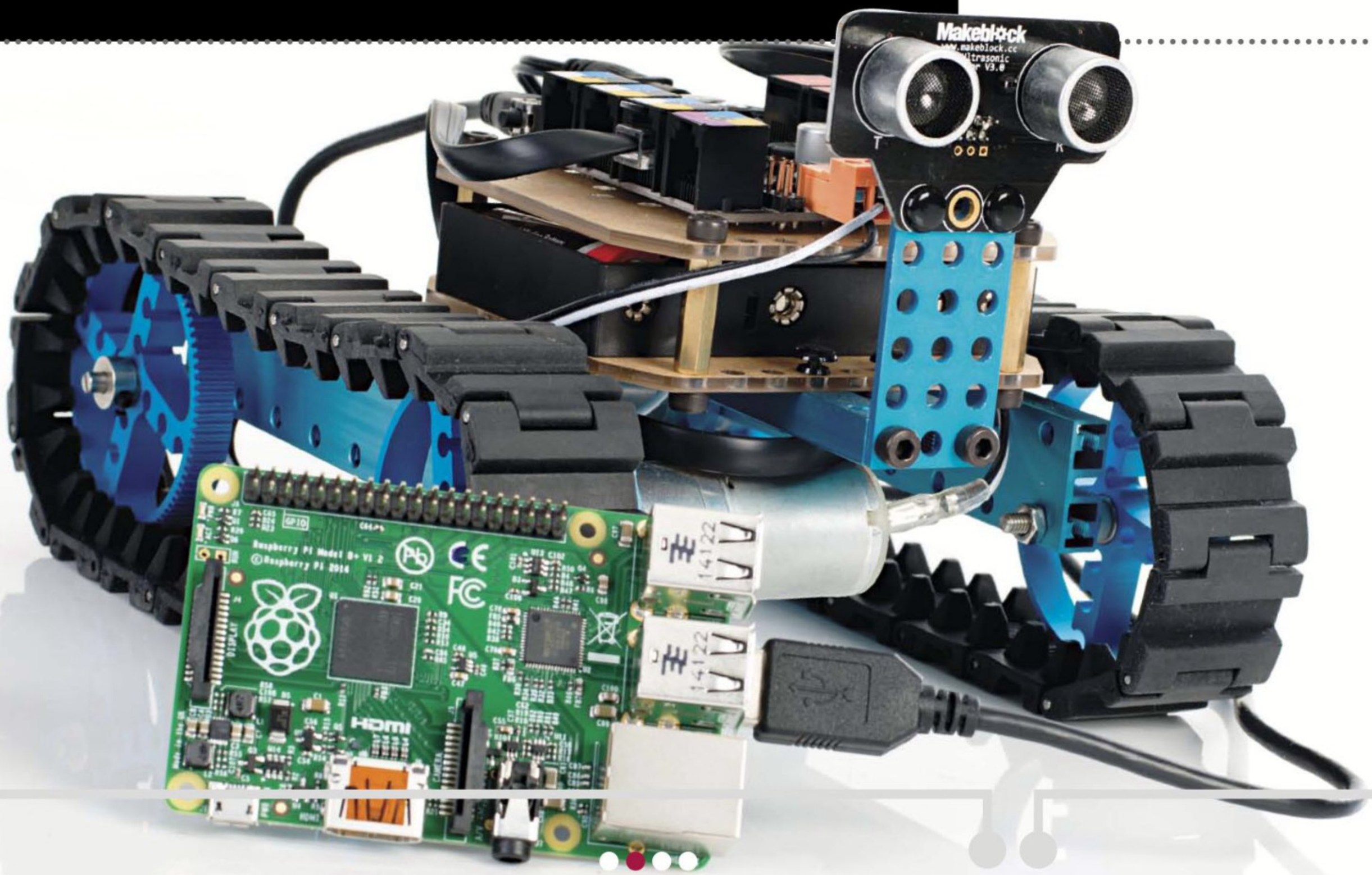
A No, it's not a Model Bot Raspberry Pi or whatever they would call it. Just a plain Raspberry Pi Model B or B+, connected up to a robot chassis via the usual mount points or with a bit of sticky tack or glue depending on how cavalier you're feeling.

Q What about the compute module?

A Yeah the compute module works as well – in fact it can work a lot better as it has 200 GPIO pins when connected to the I/O board, compared to the 26 or 40 of the other

"Arduino is usually good for controlling servos as that's what it's designed for"

Below Makeblock is another Arduino robot that's programmable on the Raspberry Pi



Raspberry Pi models. Although with a bit of know-how and a fair few components you can get it to run a robot without actually needing the I/O board.

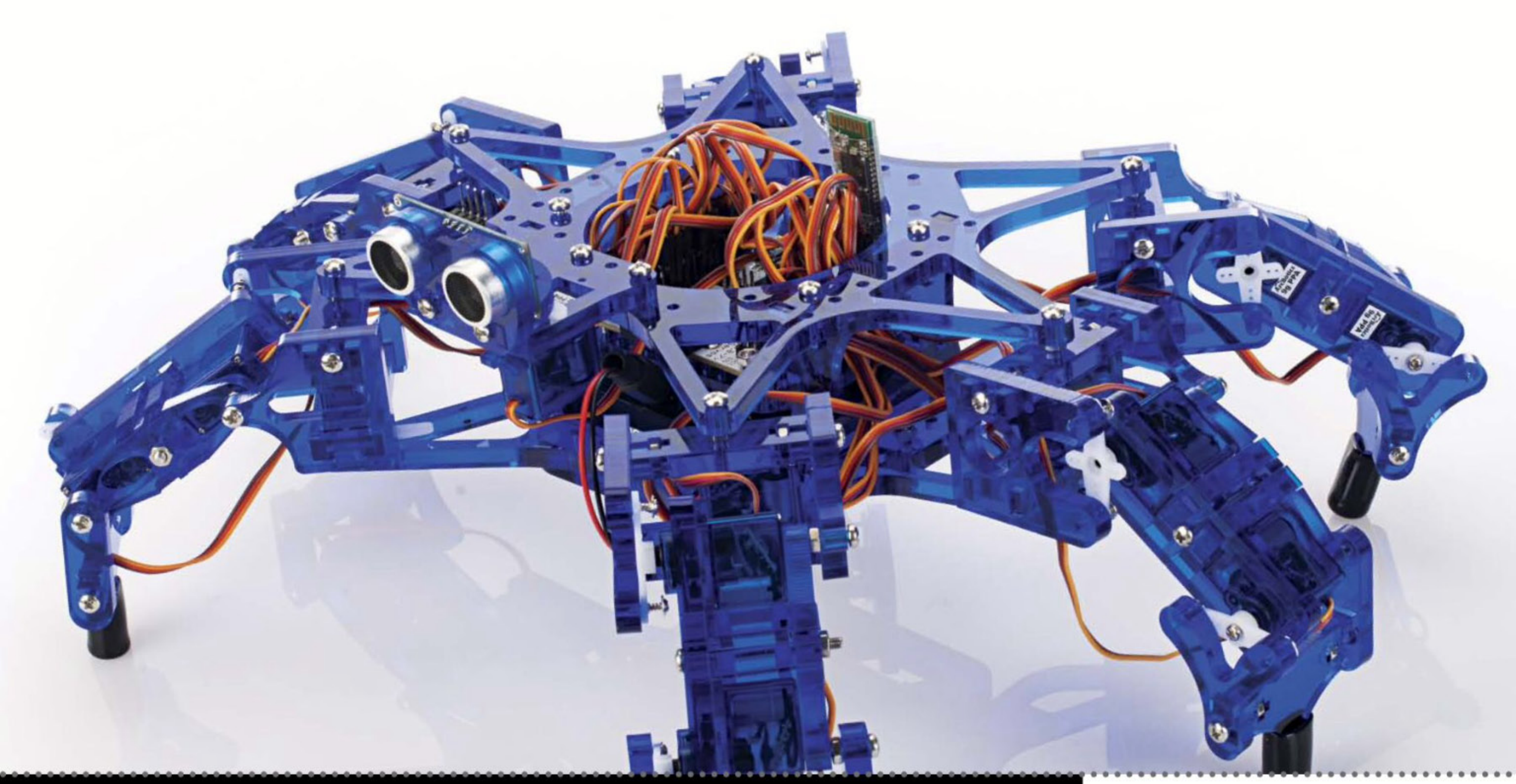
Q So the GPIO port is what powers these robots then?
A The pins allow you to read data and activate circuits and motors, which is just about everything you need to do on a robot. You could use the camera port and USB ports for things like a video feed or connecting plug-and-play devices. However, the majority of what you will be able to make a robot do will be via the GPIO.

Q Can these robots use anything else to power them?
A Sure, some are pure Raspberry Pi, some are a mixture between Pi and Arduino, and others can still be controlled by either a Pi or with an Arduino device like an Uno or Leonardo. The Arduino stuff is usually good for controlling servos as that's what it's designed for, whereas the Pi has more processing capabilities for analysing data.

Q Can all Arduino robots be controlled by a Pi and vice versa?
A With enough wire, components and patience it probably can be, but there are varying levels of difficulty in that. Some of the robot parts will require more control than others, for example. The easy answer is no, not really.

Q Can any robot be powered by a Raspberry Pi?
A That's a similar answer to before really. The Raspberry Pi has a lot of connectivity options and there are a few robots we've seen that have been hacked to use the Pi instead of their original intended hardware. One of the limits you start to get is processing power on some robots.

"The Arduino stuff is usually good for controlling servos as that's what it's designed for, whereas the Pi has more processing capabilities for analysing data"



Above Hexy is a relatively complex robot, with all those servos to articulate each joint of each leg

Q Why is that a limit?

A The more sophisticated the robot, the more inputs and input processing is needed. There's also the hard limit on things to output to because of limited GPIO pins. We do have some quite smart robots this issue, so the limit on the Pi isn't too low.

Q That's convenient.

A Very!

Q What do I need to get started with my very own robot then?

A Well kits are usually the best idea and we cover a lot of those in our big robot feature this issue. We have small, cheap ones and bigger, expensive ones that can do some really cool stuff. All of them involve varying levels of difficulty while building. See which robot in our feature appeals to you and then head over to the maker's website for more info.

Print wirelessly with your Raspberry Pi

Breathe new life into an old printer by using your Raspberry Pi as a wireless print server

Wireless printing has made it possible to print to devices stored in cupboards, sheds and remote rooms. It has generally shaken up the whole process of printing and enabled output from smartphones, tablets, laptops and desktop computers alike. But you don't have to own a shiny new printer for this to work; old printers without native wireless support don't have to end up in the bin, thanks to the Raspberry Pi. The setup is simple. With your Raspberry Pi set up with a wireless USB dongle, you connect your printer to a spare USB port on the computer. With Samba and CUPS (Common Unix Printing System) installed on the Raspberry Pi, all that is left to do is connect to the wireless printer from your desktop computer, install the appropriate driver and start printing.

CUPS gives the Raspberry Pi a browser-based admin screen that can be viewed from devices on your network, enabling complete control over your network printer.

THE PROJECT ESSENTIALS

Latest Raspbian image
raspberrypi.org/downloads

USB printer

USB wireless card

01 Check your printer works

Before starting, check that the printer you're planning to use for the project still works and has enough ink. The easiest way to do this is to check the documentation (online if you can't find the manual) and run a test print.

02 Detect your printer

With your Raspberry Pi set up as usual and the printer connected to a spare USB port, enter:

```
lsusb
```

This will confirm that the printer has been detected by your Raspberry Pi. In most cases you should see the manufacturer and model displayed.

"CUPS gives the Raspberry Pi a browser-based admin screen that can be viewed from devices on your network"

03 Install Samba and CUPS

First, install Samba to enable file and print sharing across the entire network:

```
sudo apt-get install samba
```

Next, install CUPS:

```
sudo apt-get install cups
```

With a print server created, begin configuration by adding default user 'pi' to the printer admin group:

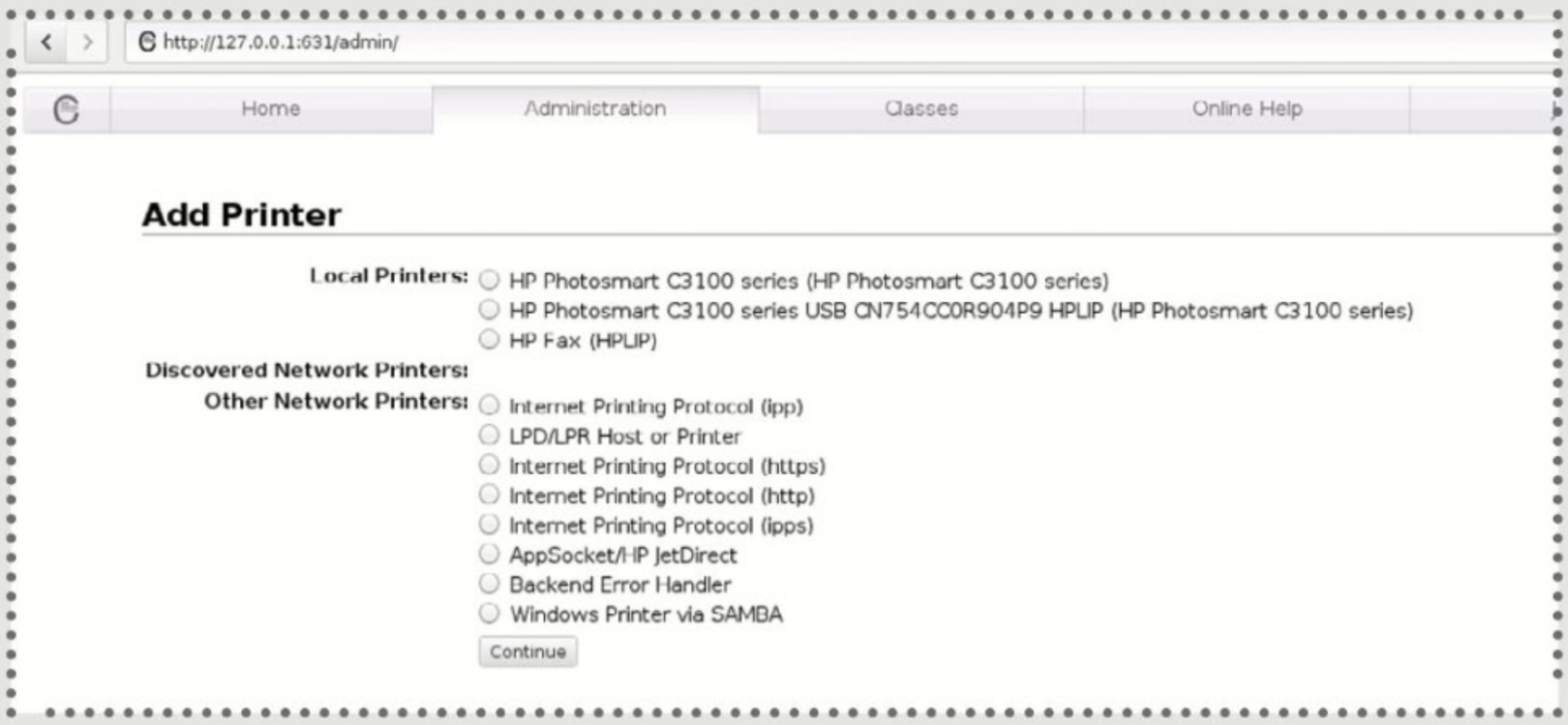
```
sudo usermod -a -G lpadmin pi
```

04 Set up print admin

Set up the CUPS print admin tool first. Boot into the GUI (startx) and launch the browser, entering 127.0.0.1:631. Here, switch to Administration and ensure the 'Share printers' and 'Allow remote administration' boxes are selected. Next, select Add Printer and enter your Raspbian username and password when prompted.

05 Add your printer

A list of printers will be displayed, so select yours to proceed to the next screen where you can confirm the details, add a name and check the Share This Printer box. Click Continue to load the list of printer drivers and select the appropriate one from the list.



Left This is the CUPS admin sceen, where you can find and select your printer

"Begin configuration by adding the default user 'pi' to the printer admin group"

06 Configure Samba for network printing

Using a Windows computer for printing? Samba will need some configuration. Open '/etc/samba/smb.conf' in nano, search (Ctrl+W) for '[printers]' and find 'guest ok' which you should change as follows:

```
guest ok = yes
```

Next, search for "[print$]." Then change the path as follows:

```
path = /usr/share/cups/drivers
```

07 Join a Windows workgroup

With these additions made, search for "workgroup" in the configuration file and add your workgroup:

```
workgroup = your_workgroup_name
wins support = yes
```

Make sure you uncomment the second setting so that the print server can be seen from Windows. Save your changes and then restart Samba:

```
sudo /etc/init.d/samba restart
```

08 Accessing your printer on Linux

Meanwhile, it's a lot easier to access your wireless printer from a Linux, Mac OS X or other Unix-like system, thanks to CUPS. All you need to do is add a network printer in the usual way and the device will be displayed.

09 Add AirPrint compatibility

It's also possible to print wirelessly from your iPad using Apple's AirPrint system. To do this, you need to add the Avahi Discover software:

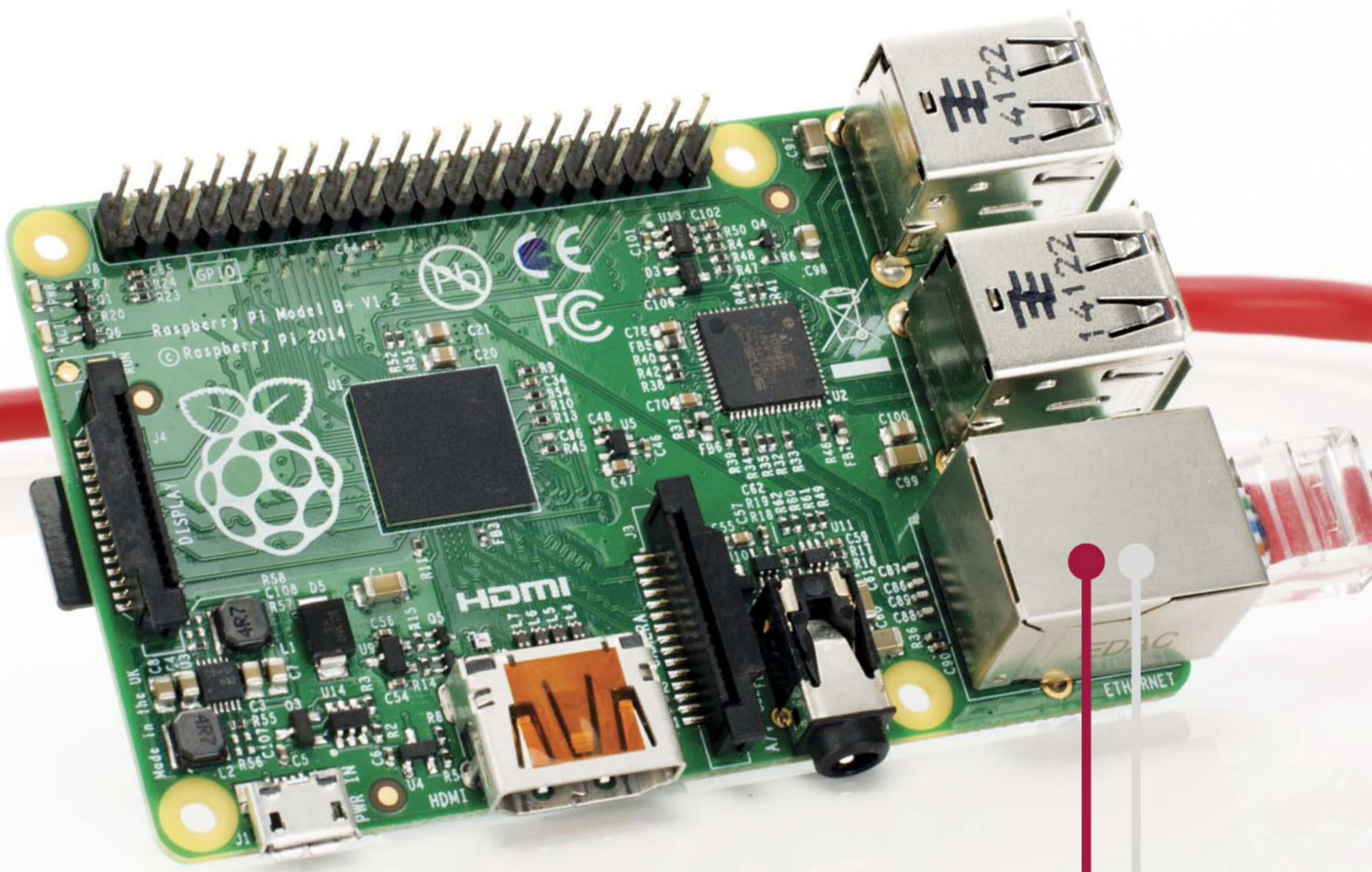
```
sudo apt-get install avahi-discover
```

Your wireless printer will now be discoverable from your iPad or iPhone and will be ready to print.

"It's a lot easier to access your wireless printer from a Linux, Mac OS X or other Unix-like system, thanks to CUPS"

Host your own website on Raspberry Pi

Don’t pay for web hosting. Configure your Raspberry Pi to act as a web server and host modest websites



Need a lightweight, low-cost web server? Your Raspberry Pi is all you need! Whether you're planning on hosting a static homepage (or one with minimal database use) or need an easy home for development websites, setting up your Raspberry Pi as a web server is surprisingly easy.

Ideal as an always-on device thanks to its low-power requirements, the Raspberry Pi can sit beside your router and serve a basic website to visitors, allowing you to put hosting fees to better use. You might wish to serve pages for some of your Pi projects, or even a personal page to host photos or your CV.

If you're planning on using it as a web-facing device, your Pi will need to be set up with a static IP address. You'll also need to ensure your Internet provider offers static IP addresses for their users. Often a price is charged for leasing a static IP, but there are services you can use (such as **www.noip.com**).

THE PROJECT ESSENTIALS

Latest Raspbian image
raspberrypi.org/downloads

Internet connection

External hard drive (optional)

USB flash (optional)

Ethernet cable for reliability

01 Connect your Ethernet cable

For this project it makes more sense to use an Ethernet cable. You may need your existing USB ports to attach flash drives or an external HDD to serve your web page. With Ethernet you will need to rule out any wireless issues that are causing interruptions for your visitors.

"If you're planning on using it as a web-facing device, your Pi will need to be set up with a static IP address"

02 Get Raspbian updates and Apache

As ever, begin by checking for Raspbian updates:

```
sudo apt-get update
```

You'll then need to install Apache and PHP:

```
sudo apt-get install apache2 php5 libapache2-mod-php5
```

Finally, restart Apache:

```
sudo service apache2 restart
```

Your Raspberry Pi is now ready to be used as a web server.

03 Check your Pi web server

With Apache installed, open the browser on another computer on your network and enter your Pi's IP address to view the Apache confirmation page. As things stand right now, all you will be able to view is the Apache index.php page. To add your own HTML and PHP pages, you will need FTP.

04 Install FTP for uploading files

Create a www folder, then install the following vsftpd FTP server software:

```
sudo chown -R pi /var/www
sudo apt-get install vsftpd
```

You'll need to make some changes to Very Secure FTP Daemon, so open it in nano. First, switch:

```
anonymous_enable=YES
```

…to…

```
anonymous_enable=No
```

Next, uncomment the following by removing the # symbols:

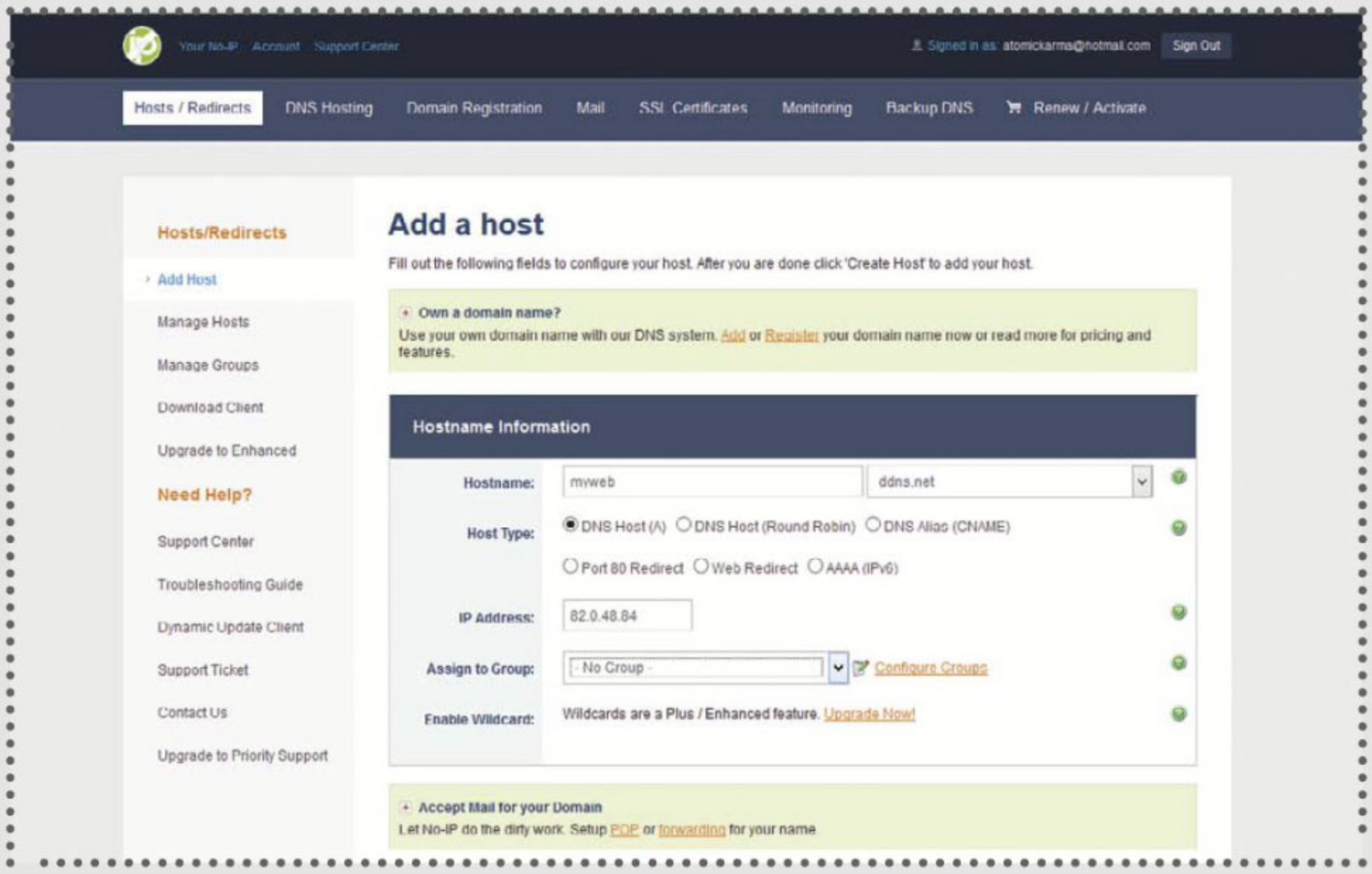
```
#local_enable=YES
#write_enable=YES
```

05 Restart the FTP Server

Complete configuration of the FTP software by adding a command to the end of the file which will display server files starting with "." such as .htaccess:

```
force_dot_files=YES
```

"Begin configuration by adding the default user 'pi' to the printer admin group"



Left With NoIP you can set up your own hostname using their DNS system

Save and exit nano (Ctrl+X) and restart FTP:

```
sudo service vsftpd restart
```

Using the default Raspbian credentials you can upload files to /var/www.

06 Make Pi a LAMP server

By adding MySQL into the mix you can use the Pi to host a database-driven website or even WordPress (although this is best limited to using the device as a development server).

```
sudo apt-get install mysql-server mysql-client
php5-mysql
```

The LAMP bundle is useful of course, but for the best results your site should remain streamlined.

07 Get your site online

Can't afford a static IP for your router? A great solution is available with the free service from **www.noip.com**. This enables you to point a hostname at your computer by using a client application that will remain in contact with the No-IP servers.

"By adding MySQL into the mix you can use the Pi to host a database-driven website or even WordPress"

08 Install No-IP

Make a new directory and switch to it:

```
mkdir /home/pi/noip
cd /home/pi/noip
```

Download No-IP on your Pi with:

```
wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz
```

Extract:

```
tar vzxf noip-duc-linux.tar.gz
```

Next, navigate to the directory and use sudo make and sudo make install, following any instructions. Finish by running this:

```
sudo /usr/local/bin/noip2
```

09 Change your password for security

Before using your Pi as a live web server, it's a good idea to change the default password to something more imaginative than 'raspberry'.

In the command line, enter passwd and then follow the prompts to add your new, secure password. You're doing this step because you obviously would not want your Pi web server to get hacked!

"Before using your Pi as a live web server, it's a good idea to change the default password to something more imaginative than 'raspberry'"

Pi Glass

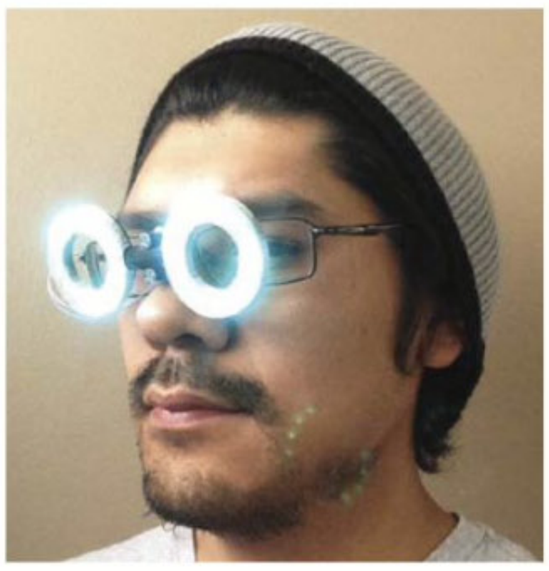
Adafruit creatives Noe and Pedro Ruiz hack video goggles to make a 3D-printed Google Glass-like attachment

How did you get started with Adafruit?

Noe It was about a year ago, we came on their show-and-tell and we wanted to show people what projects we were working on. At the time it was a simple wearable project – a 3D-printed belt buckle with Adafruit LEDs and their GEMMA microcontroller, so the thing there is mixing self-expression and design with the electronics and making it flashy and cool. Adafruit really liked that, and what they liked even better is that we happen to make videos as well, so Phil [Torrone], a cofounder of Adafruit, wrote to us asking if we'd like to be an author on their Adafruit Learning System. We said sure, we can write documentation, but we can also do video too. And that sort of led to starting another project and it gained momentum from there, and every week we've been coming out with a new 3D-printed project since. Recently it's been getting so much bigger, and it's always a challenge because every week we're upping our skills – it's like, can we design that, will that print? So far it's been more successes than failures. We do a good job learning from the bad stuff and capturing the really good stuff and telling people in our guides how to keep moving on. It's very hard stuff but we try to make it look like it's not so hard so that people try it out and learn from it.

Pedro We try to make the guides as repeatable as we can – step-by-step guides that are easy to use.

Noe It's so open sourcey! We had no idea of the open source hardware movement and it's completely empowering to give away our design files. When we started out as Pixil 3D we really didn't give away our designs – we'd hold on to them because it was our stuff. But now it makes so much sense to give away our designs since you have that incentive to. It's like hey, here's a cool project idea – just buy some parts and then follow along with our circuit diagrams and tutorials.



Noe Ruiz has designed industry products since the age of 14, including UI design for iOS apps, websites and audio production for video games

Pedro Ruiz is a media designer and has worked on iPhone apps, video games, 3D modeling, video production and social media

Pedro It really speaks to what 3D printing is becoming – it's the shell that holds all of the individual components that bring it to life inside.

So what exactly does your DIY Glass do?

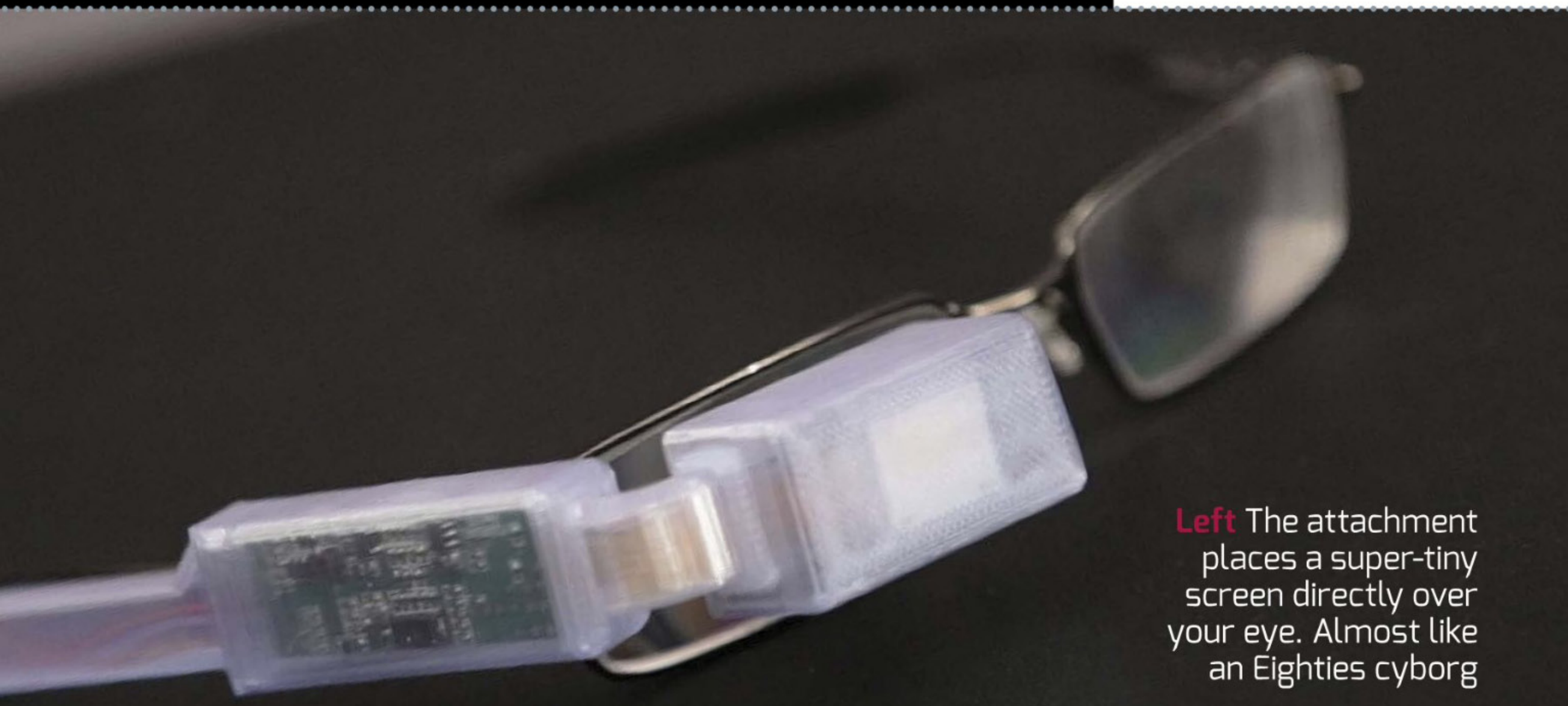
Noe The idea was sort of inspired by Limor [Fried] herself – she has these hundred-dollar video glasses on the shop and she said: "You know what? Let's take it apart. Let's take the guts out, the actual circuitry, and make a new format for it. Instead of being two glasses let's make it clip on to your existing glasses kinda like Google Glass, but let's make it for the Raspberry Pi." So that was the original idea and it was rather simple because there really wasn't much programming or software involved – it was just repurposing this component inside this hundred-dollar pair of glasses and making it more Google Glassy and more DIY. From a design standpoint it was really challenging because the tolerances and things for that was kinda hard, especially for different machines – you're

If you like

If you want to make your own Pi Glass then check out Noe and Pedro's tutorial on the Adafruit Learning System: **bit.ly/1fbHhfw**

Further reading

To see some of the crazy electronic costumes and 3D-printed gadgets that the Ruiz brothers make, check out their main page over at the Adafruit site: **bit.ly/1yBSECn**



Left The attachment places a super-tiny screen directly over your eye. Almost like an Eighties cyborg

always looking at different tolerances; even when you're slicing it, things will come out a little bit differently. So that's why it's so important to give away the files and to tell people that you can modify it and you can make it work for you. And quite a few people have made their own and printed it for their application.

Pedro It's a good foundation for anybody who can build on top of it. So we've seen different Pi UIs that mimic the Google Glass UI – this is something that somebody could take and sort of adapt.

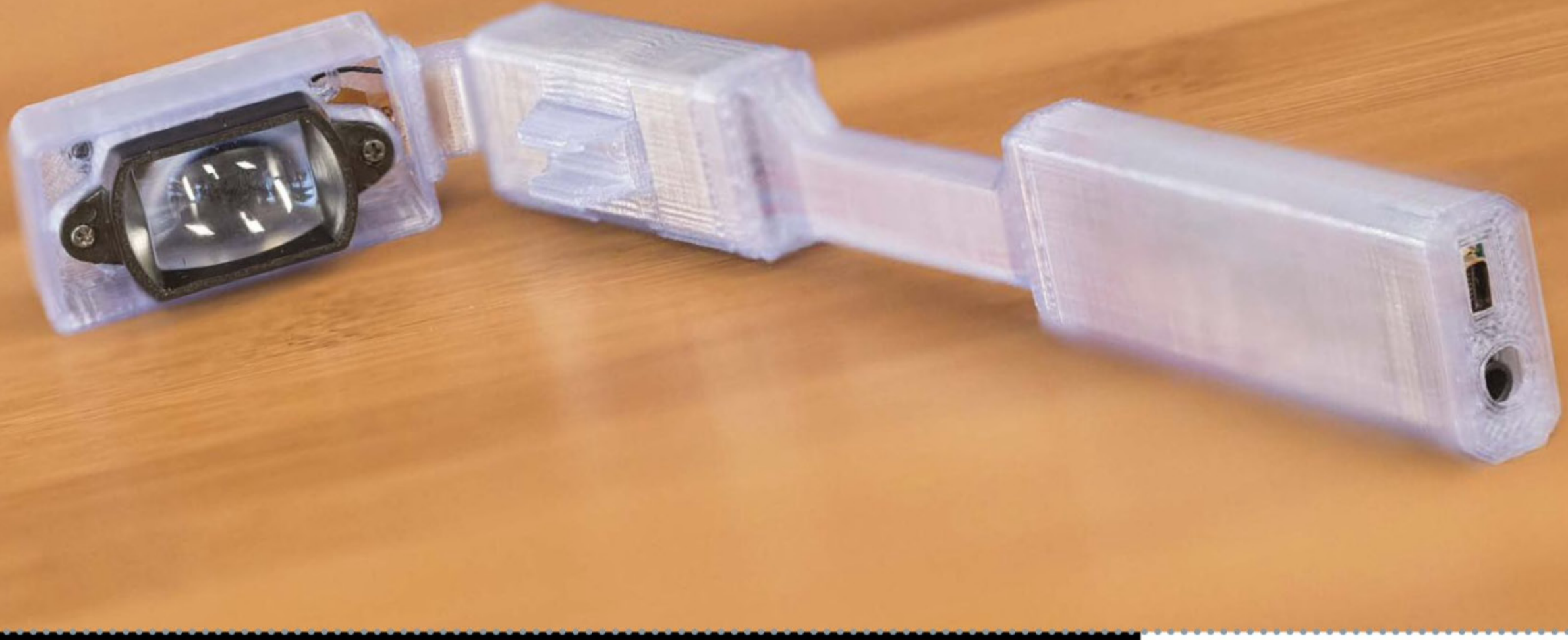
Noe It's a great example – so someone who's not super ace at designing or printing but maybe has the software chops can take this project and make it even better. We'd really like to see that.

How did you make the attachment after you split up the original video glasses?

Noe So I guess for starters we bust out the calipers and we start measuring like crazy. From there we designed the components, we remake them in CAD – our favourite

"We've seen different Pi UIs that mimic the Google Glass UI – this is something that somebody could take and sort of adapt"

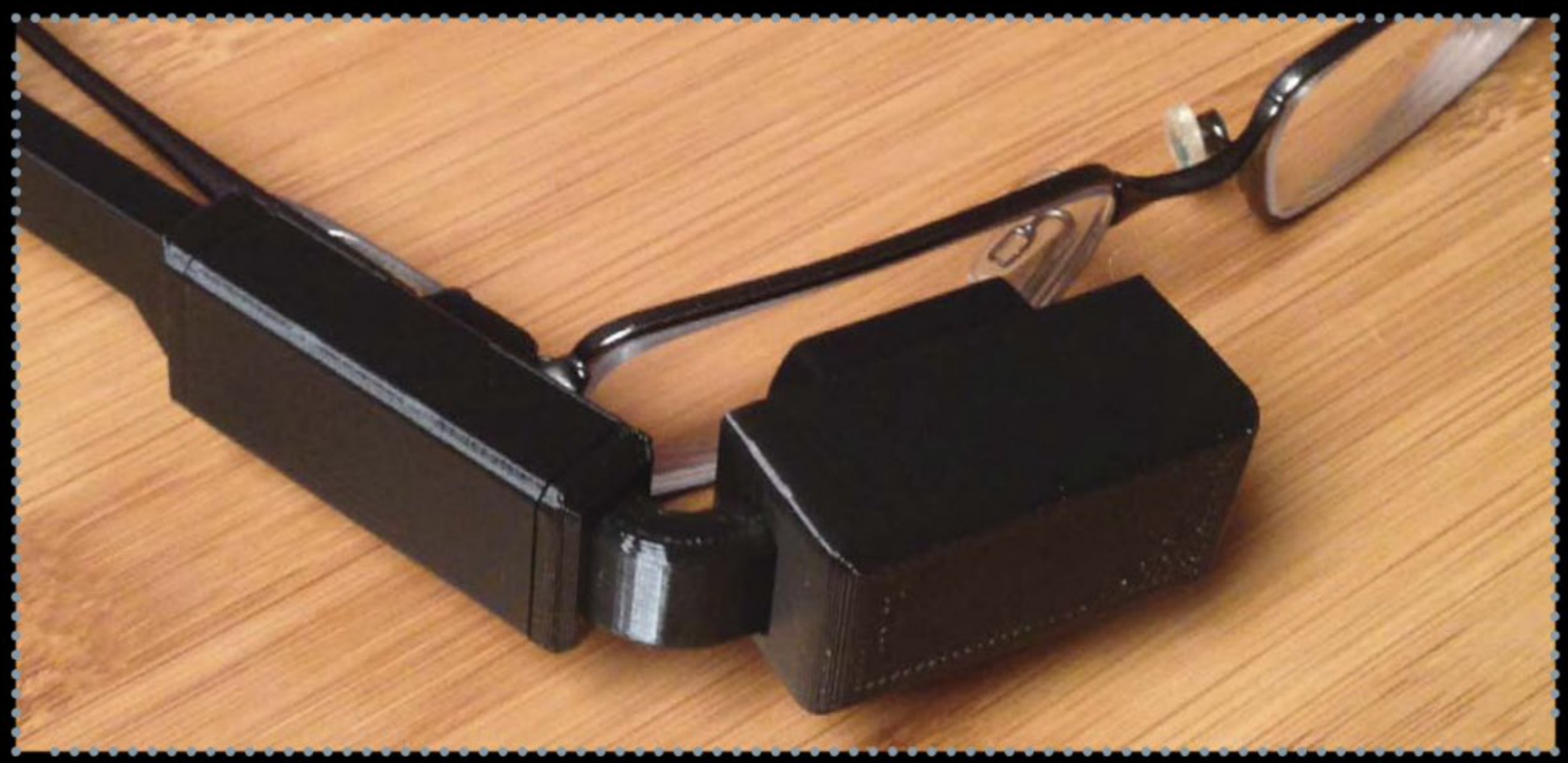
Left These video glasses cost about $110/£70 and, while not exactly suited for long-term use, they're perfect for repurposing into other optical projects like the Pi Glass



CAD right now is 123D Design, which is from Autodesk. We use it literally on a daily basis. But you start off by making the components and modelling them out, and creating the enclosure on top of that and just chiselling away and creating features, figuring out how to split it up into pieces so that it can print without any support material. We really strive to make our designs with no supports – that way you can get a really clean design that looks beautiful and doesn't require that extra bit of waste. And it is 3D printing – it's rapid, right? So we prototype the piss out of our projects! We're so fortunate that we have the time to do it. It's hard to keep it under two weeks, but it really feels like it's a rush and we do step back and take the time to make sure it's right.

Pedro We always have a buffer of at least a month with projects already in the works and we make sure we keep the pipeline full – sort of like a TV schedule. And sometimes if we can't finish quite in time, we let people know and say hey, you can finish this for us by all means – go ahead and pick up where we left off.

Above The Pi Glass attachment is affixed to your regular glasses using the small clip on the inside of the central 3D-printed block



Left The tiny screen on top of your glasses provides a 320 by 240 display and feels like you're looking at a 52-inch screen a few feet away

How is the Pi talking to the attachment?

Noe It's just plugged in through HDMI, really – it's just an add-on to the Pi to make it mobile. You just plug into a battery bank, so it's really simple in that way.

Is there an easy way to control the output?

Noe We have a small wireless keyboard that we sell in the shop, so we thought we'd keep it as simple as possible and use that to control things on the Pi.

Do you think this is a project you'll revisit?

Pedro Well, with the release of the A+ we might revisit it in a future episode.

Noe Maybe something more enclosed and more specific to the Pi.

Pedro Yeah, so build more libraries for talking to sensors and things like that. We might try and incorporate some eye-tracking, things like that.

Noe We have a really cool remote team that does different projects as well, so we're just now starting to collaborate with them because they're more skilled and disciplined in software engineering, so it's really cool to bring those two minds together – the design and videography and then the software engineering.

"More libraries for talking to sensors and things like that. We might try and incorporate some eye-tracking, things like that"

Talking Pi

Join the conversation at...

@linuxusermag | Linux User & Developer | @ RasPi@imagine-publishing.co.uk

The Raspberry Pi encourages a lot of hands-on work and this means it doesn't quite work like – or as easy as – your laser focus-tested smartphone interface, or even a normal computer. So we're answering your burning Raspberry Pi questions each issue – get in touch with us on Twitter, Facebook or by email.

What's the limits of the Raspberry Pi?
Tsukasa via email

That's a tricky one to answer – what's the upper limit you're thinking of? With enough optimisation and clever code, you can get a Raspberry Pi, or multiple Raspberry Pis working together, to do just about anything. Desktop, server, super computer, you name it. You'd be better off getting systems created for the tasks though, and in some cases they may be slightly cheaper. With enough ingenuity though, you can get a Pi to do many things.

Can I install my Raspberry Pi into my TV?
Matt via Facebook

Well it depends on your TV and your level of skill. If you're talking about actually taking apart your TV and soldering it to it, then you might be able to but that's an incredibly complex process and many things can go wrong. It will also be hard to access for other maintenance. You're best just keeping it outside the TV but plugged in via HDMI for whatever you plan to do with it.

Keep up with the latest Raspberry Pi news by following @LinuxUserMag on Twitter. Search for the hashtag #RasPiMag

JUST A SCORE
WHAT'S YOUR JUST A SCORE?

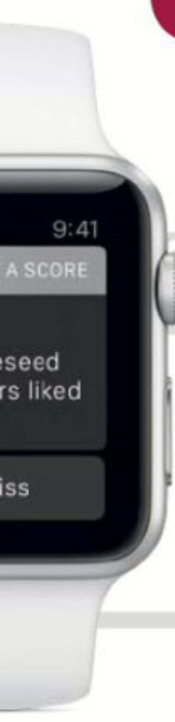
Have you heard of Just A Score? It's a new, completely free app that gives you all the latest review scores. You can score anything in the world, like and share scores, follow scorers for your favourite topics and much more. And it's really good fun!

Can I use specific GPIO pins for multiple purposes?
Mel via Twitter

In theory, yes. With the way you program stuff that uses the GPIO pins, you can technically have it do different tasks at different points in the code as long as everything is wired up correctly from there. You can't have it do multiple things at once though, and the wiring will be quite complicated to route the different functions to different parts of a circuit or system. You're better off using other GPIO pins for the tasks if you have some to spare.

I'm still using my very original Raspberry Pi, should I upgrade?
Michelle via Facebook

If you're still using it just fine then there's no hurry to upgrade. The only thing you'll have to be aware of is that more and more tutorials will not work so well on the first Model Bs. You'll still be able to use it for less resource heavy uses though, and it still makes a fine media centre for your TV if you do plan to upgrade to do more amazing things in the future.



Next issue

Get inspired · Expert advice · Easy-to-follow guides

CONTROL ROBOTS

Get this issue's source code at:

www.linuxuser.co.uk/raspicode