• Arduino based • Low Cost • Easy to build • Little or no experience needed!

ARDUINO

EARTHQUAKE Early Warning Alarm

Concept by Allan Linton-Smith • Circuit and software by Nicholas Vinen

Earthquakes can strike anywhere . . . and usually with very little warning. But these days there are ways that you can get an early warning, that may be the difference between getting to safety (eg, an open area) and possible injury or death. So how do you go about getting early warnings of impending earthquakes? Read on...

Probably the easiest way to get earthquake warnings is to install an early warning app on your smartphone.

The idea is that a network of seismographic sensors based around the world will pick up an earthquake soon after it occurs and determine its location (based on triangulation), depth and magnitude.

The app receives this data within seconds and compares it to your location.

Depending on your proximity to the earthquake and its magnitude, it can generate an alert, seconds or even minutes before the destructive waves of the earthquake arrive.

But this does rely on a few things working properly: you have to have a smartphone, it has to be charged and switched on, it has to have a working internet connection, the app needs to be installed and running properly.

And there's also the fact that, depending on where the seismic sensors are located geographically, significant

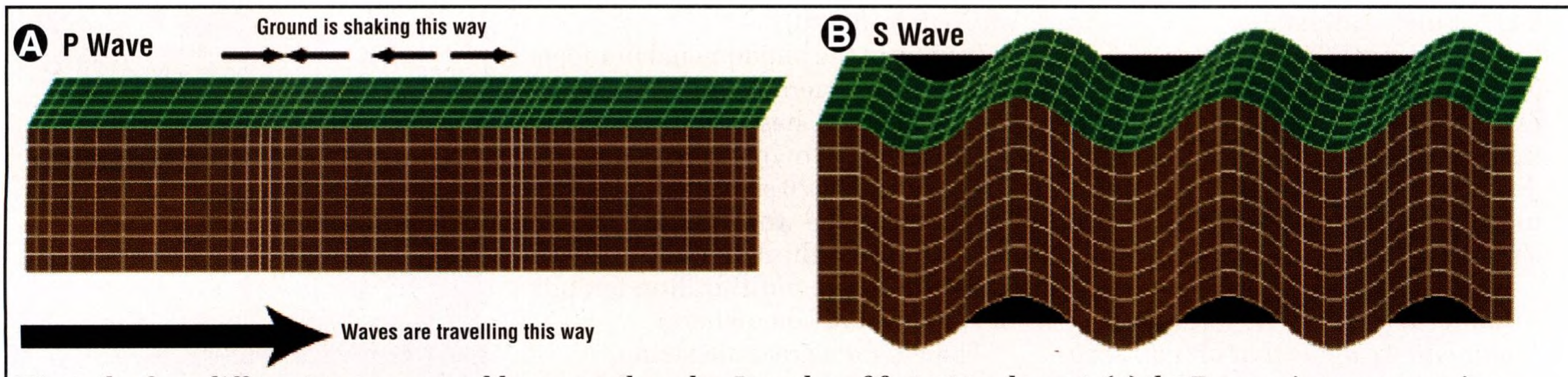


Fig.1: the four different waves caused by an earthquake. In order of fastest to slowest, (a) the P-wave is a compression wave, (b) S-wave is up-and-down and/or side-to-side motion.

A commercial earthquake early warning alarm, the FREQL (Fast Response Equipment against Quake Load), used by rescue teams in earthquake areas. Ours is very much simpler . . . and cheaper!

Here's another commercial detector – the Chinese-made XYB01A. It's not intended for first-responder use; in fact, it's designed for home use, mounting on a wall as shown. We found it tricky to set up and use.

time could pass before the alert is even raised.

We installed some popular earthquake early warning apps and set them up to warn us about earthquakes around the world. (There are, literally, hundreds of earthquakes occuring every day – only the largest make the six o'clock news . . .)

Timing!

We found that we sometimes got alerts many minutes after an earthquake had occurred – somewhat pointless, you'd agree!

Of course, even if the warning is timely, you might not hear the alert or you may not look at the screen straight away.

But there's another option and it may be much more useful, because it doesn't rely on remote seismic sensors, an internet connection or any software.

And you don't even need to own a smartphone.

Early warning using P-waves

Earthquakes cause a disturbance in the Earth's crust that you can feel. They are generally caused by a sudden rock fracture where the pressure has built up at the junction of two tectonic plates, due to continental drift.

When this energy is suddenly released, it causes waves to travel through the Earth's crust away from the location of the fracture.

You may not realise it but a single seismic event can cause at least four different waves to travel through the Earth and shake the ground beneath your feet.

Unless you are very close to the epicentre, these waves will arrive at different times and they will have different strengths and effects.

The first wave to arrive is the pressure wave or P-wave. This travels in a similar manner to the way sound travels through air – see Fig.1(a).

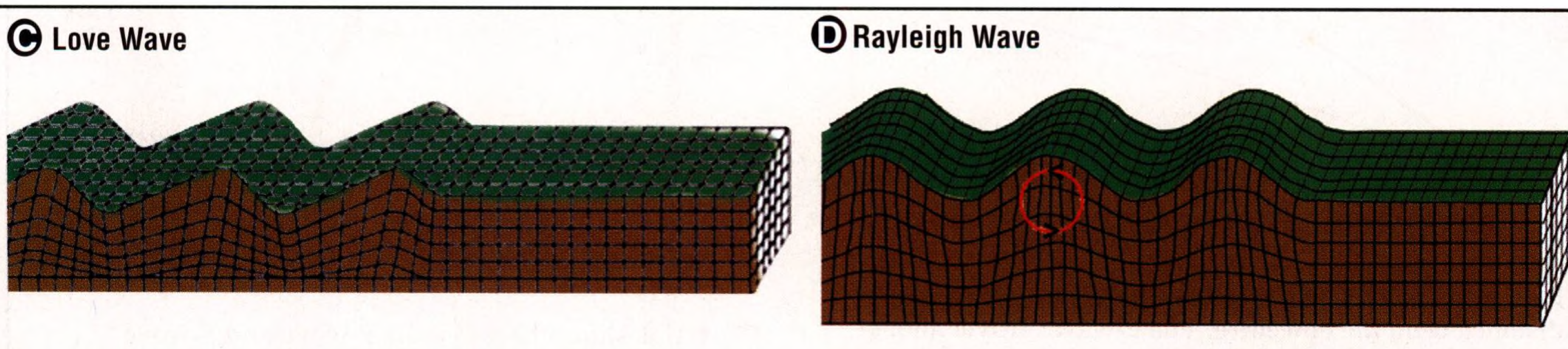
Part of the reason why it arrives first is that it can travel through solids and liquids, so it can take a direct path through the Earth to your location (ie, it doesn't have to follow the curvature of the Earth, despite the fact that there are liquid layers under the Earth's crust).

The P-wave is usually not terribly strong nor destructive but it certainly can be detected using seismic monitoring equipment and this will give you the most warning before the destructive waves arrive at your location.

The secondary wave is known as the S-wave and this is caused by rock particles moving side-to-side or up and down, similarly to the way that a wave travels through deep water – see Fig.1(b). Because the S-wave cannot travel through liquid, it can not pass through the Earth's outer core and so generally arrives after the P-wave. It is usually strong enough to be felt but is not the most destructive wave.

The third wave to arrive is the Love wave (named after A.E.H. Love) – see Fig.1(c). This is the fastest surface wave and is caused by the surface of the Earth moving side-to-side.

Because it has to travel along the surface, it takes the longest path and therefore arrives after the S-wave and P-wave.



(c) the Love wave is side-to-side and (d) the Rayleigh wave has a vertical, rolling action (and tends to be the most damaging). *Source: US Geological Survey.*

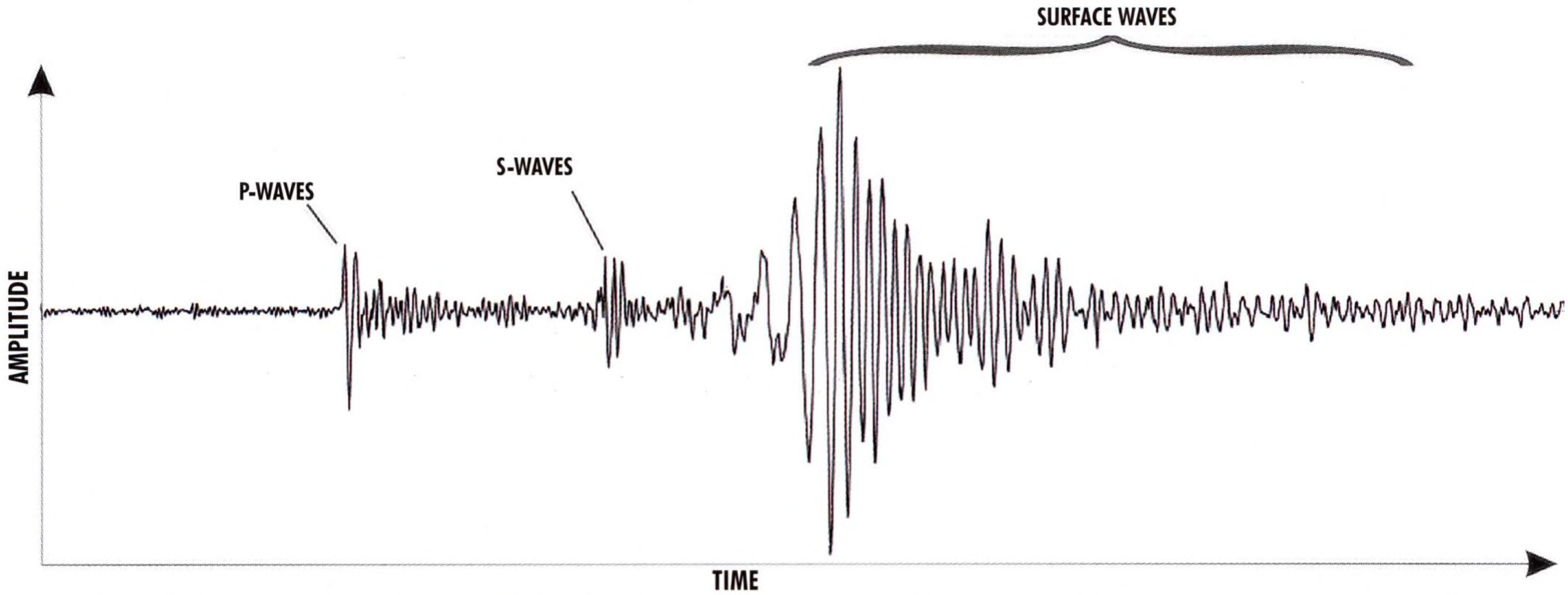


Fig.2: a seismograph plot taken some distance from an earthquake, showing that the P-waves arrive first, then the S-waves, then the surface (Love and Rayleigh) waves.Typically, the surface waves have the greatest amplitude and will be the most destructive. Source: US Geological Survey.

Shortly after the Love wave comes the Rayleigh wave, which also travels along the surface. It causes vertical motion as the ground "rolls", much like waves in shallow water – see Fig.1(d). This is the wave which is normally felt the most and causes the most destruction.

The relative speeds of the P-waves, S-waves and surface waves can be seen in the seismograph plot of Fig.2. Fig.3 gives more detailed information on the relative speeds of P-waves and S-waves while Fig.4 shows how the P-waves and S-waves travel at different speeds through different parts of the Earth's crust.

Notice though that the P-wave velocity is always higher than the S-wave velocity, so in most cases it will arrive much earlier.

Fig.3 shows how long a typical P-wave and S-wave take to reach a certain distance from the epicentre. As you can see, the S-wave takes around twice as long to reach a given point compared to the P-wave.

If we can detect the passage of the P-wave, then the interval between these two lines is the amount of warning we get before the larger S-wave arrives.

For example, if you are 200km from the epicentre, you would get around 30 seconds' warning while if you are 2000km away, you will get around five minutes' warning.

Unfortunately, the closer you are, the less warning you will get and the more destruction the earthquake will cause (as the waves drop in power as they travel away from the epicentre and expand).

For the most damaging 'quakes, you probably won't get more than one minute of warning.

Detecting the P-wave

Commercial P-wave detector devices do exist. One example is the port-

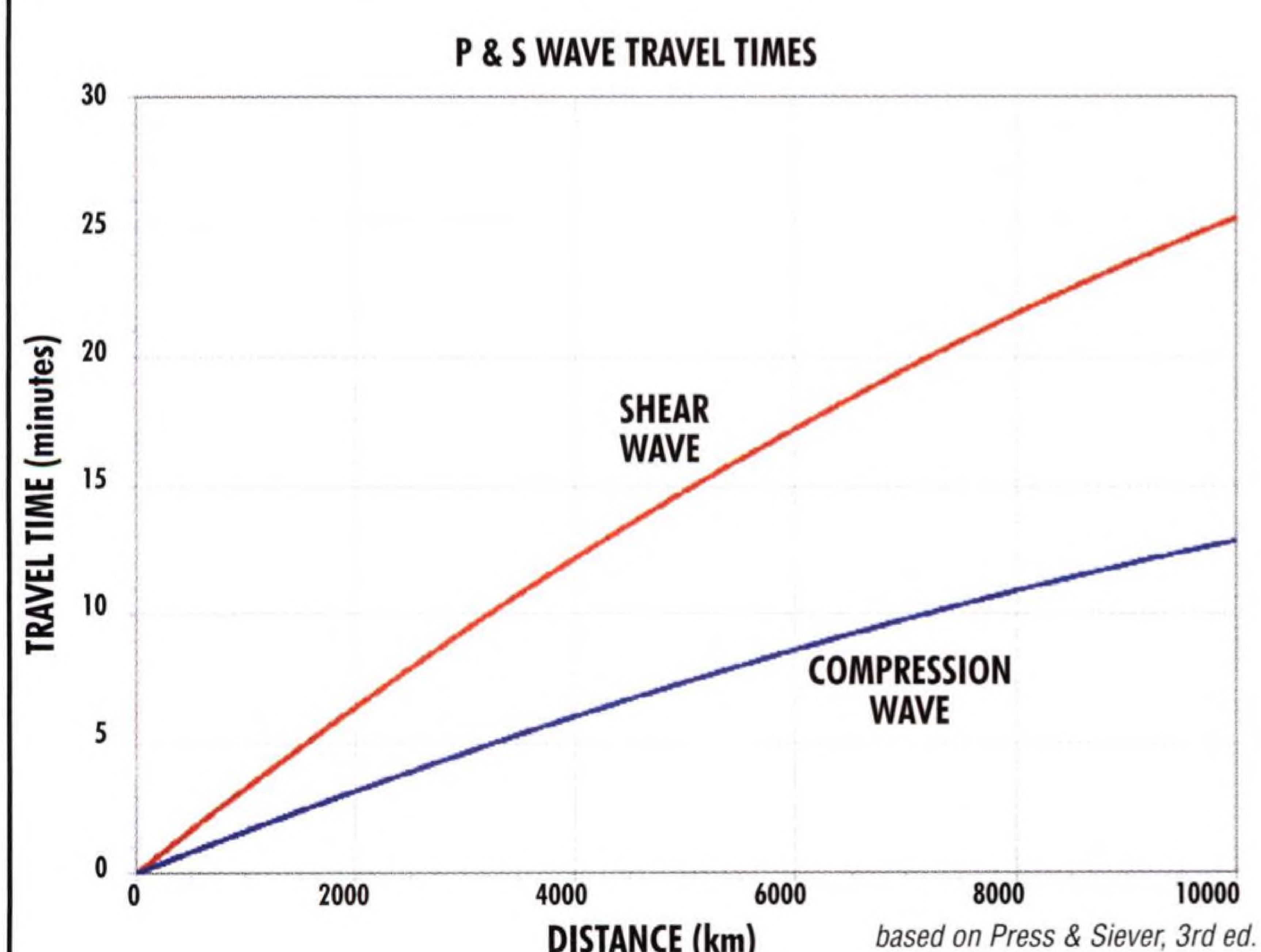


Fig.3: a graph of approximately how long it takes for the P-wave and S-wave to reach a point a certain distance from the epicentre. The P-waves travel about twice as fast as the S-waves so they reach the same distance in about half the time. The lines are curved due to the curvature of the Earth.

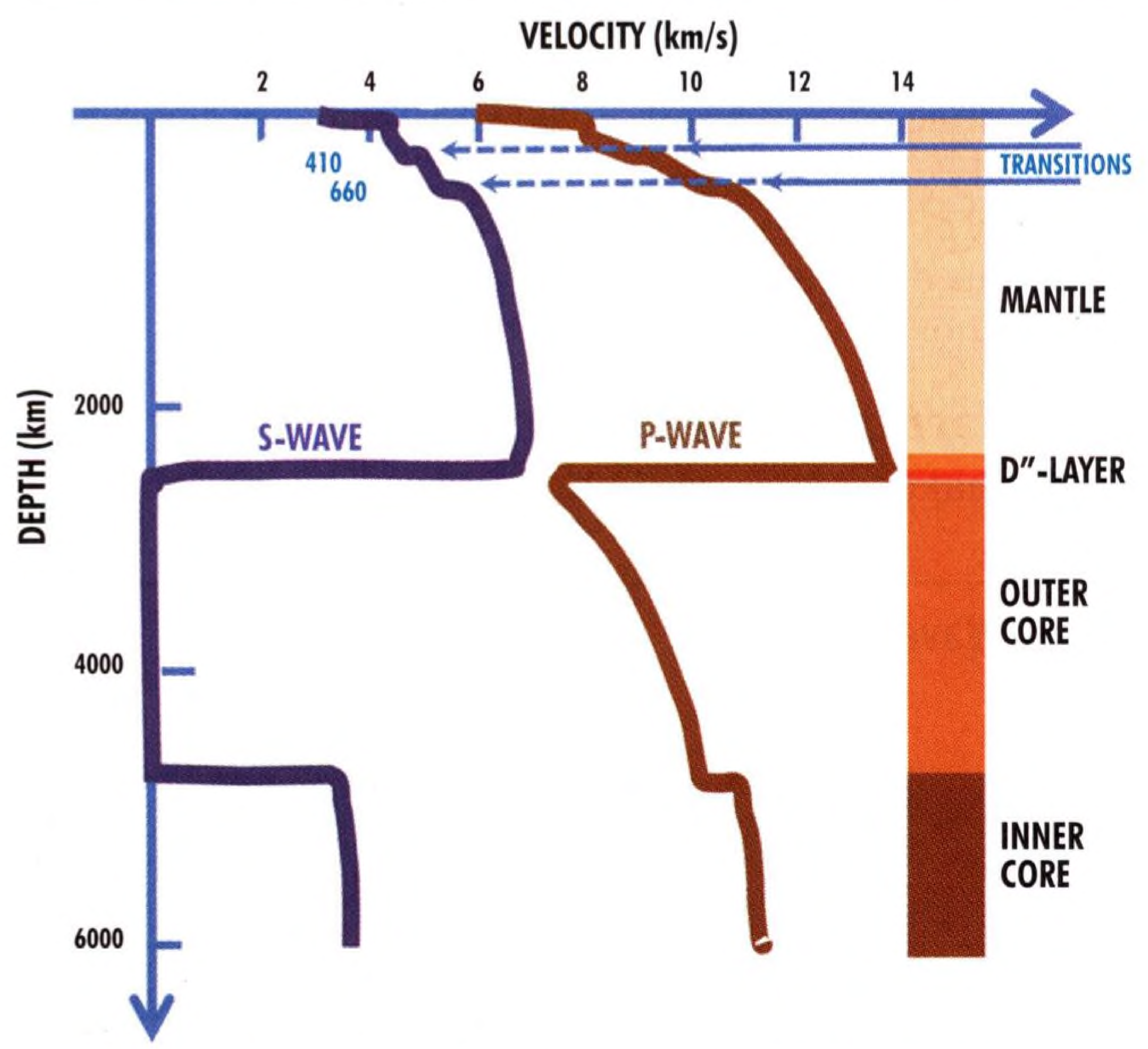


Fig.4: this shows how fast the P-wave and S-wave typically travel at various depths in the Earth's crust. The P-wave travels faster so it will arrive first.

able FREQL (Fast Response Equipment against Quake Load). This is used by rescue teams and fire departments in Japan and is especially useful for early warning of dangerous aftershocks during difficult rescue phases. It's shown overleaf.

You can also get consumer-grade devices such as the Chinese-made XYB01A detector. This is a wall-mounted unit which runs from a 9V battery and uses a pendulum to make contact when a P-wave is experienced, sounding the alarm.

It is mechanically adjustable but is a little tricky to set up.

The P-wave normally has a frequency of between one and five hertz (1-5Hz) and could consist of just a short jolt, a series of tremors or a continuous wave, depending on the nature of the earthquake.

So to give you the best chance, the device needs to be as sensitive as possible to signals in that frequency range and with the correct orientation, without being so sensitive that it could be set off by other vibrations.

Our detector

The electronic device we describe here uses a relatively inexpensive but very sensitive accelerometer combined with a regular Arduino board.

Depending on where you live, it may give you enough warning to find a safe place if it detects an oncoming earthquake or aftershock. And it may be useful if you live near an active volcano; volcanoes can generate P-waves prior to eruption.

No promises, of course: but it's much better to have a detector which could give you warning than have no detector and have no chance!

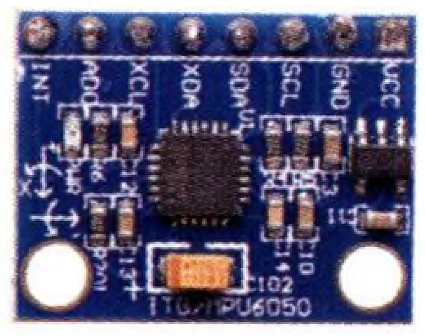
Besides, it's cheap, easy to build and requires very little soldering. You can put it together in about an hour or so, even if you aren't very experienced.

We considered designing the device around analog circuitry but P-waves can come from any direction and thus some fairly intense signal processing is required.

This is much easier to do with software and it doesn't require a custom-designed PCB.

Circuit details

Our circuit is shown in Fig.5. The two main components are the Arduino Uno (or equivalent) board, MOD1, and the Altronics Z6324 digital accelerometer/gyroscope, MOD2.



The tiny MPU-6050 3-axis accelerometer which is the "heart" of the project, detecting distant P-waves.

MOD2 uses the MPU-6050 IC and we've chosen this one in particular because it has an on-board 16-bit digital-to-analog converter (DAC). Note that we aren't using the gyroscope feature, just the accelerometer.

At maximum sensitivity, the full-scale reading of this device is ±2g on each of the three axes and the 16-bit DAC means this has a resolution of 0.0006g [(2 ÷ 32768].

That's what we need to detect the very small vibrations of a P-wave from a distant source.

P-waves are often so faint that you can't feel them with your sense of touch but this device can potentially detect such small tremors.

One of the handy things about the MPU-6050 is that it has configurable digital low-pass and high-pass filters. The low-pass filter can be configured with a -3dB point of 5Hz, 10Hz, 21Hz, 44Hz, 94Hz, 184Hz or 260Hz. We have chosen 5Hz as this suits our application.

Similarly, you can configure it for a high-pass filter of 5Hz, 2.5Hz, 1.25Hz or 0.625Hz.

We have used the last option, giving a response of 0.625-5Hz. We provide an additional 1Hz high-pass filter in the software (which also helps to remove any residual gravity from the readings, eg, if the unit is not mounted perfectly horizontally).

The Arduino makes a couple of dozen readings of the X, Y and Z axis acceleration figures each second and after processing them, it uses an RMS

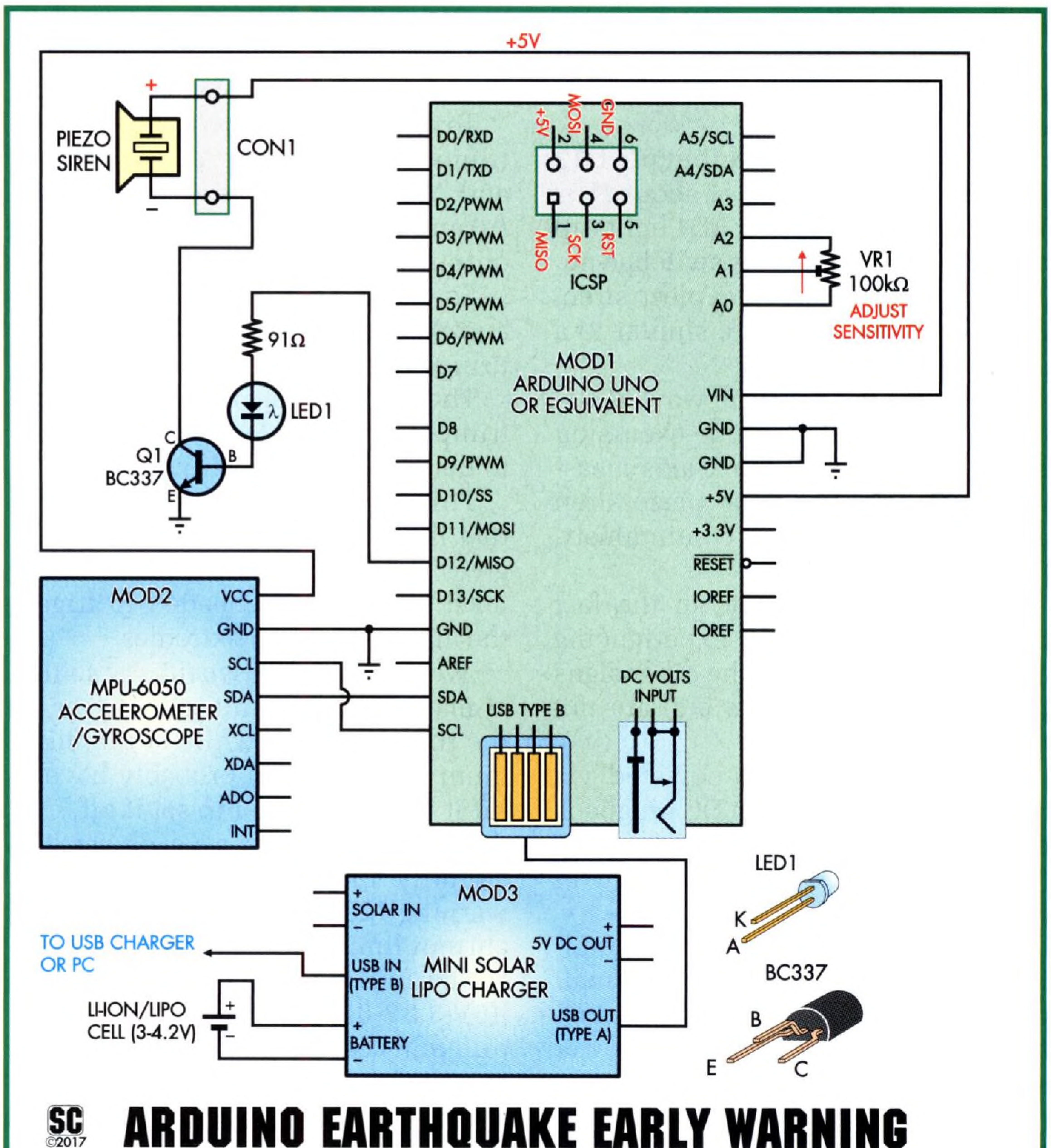
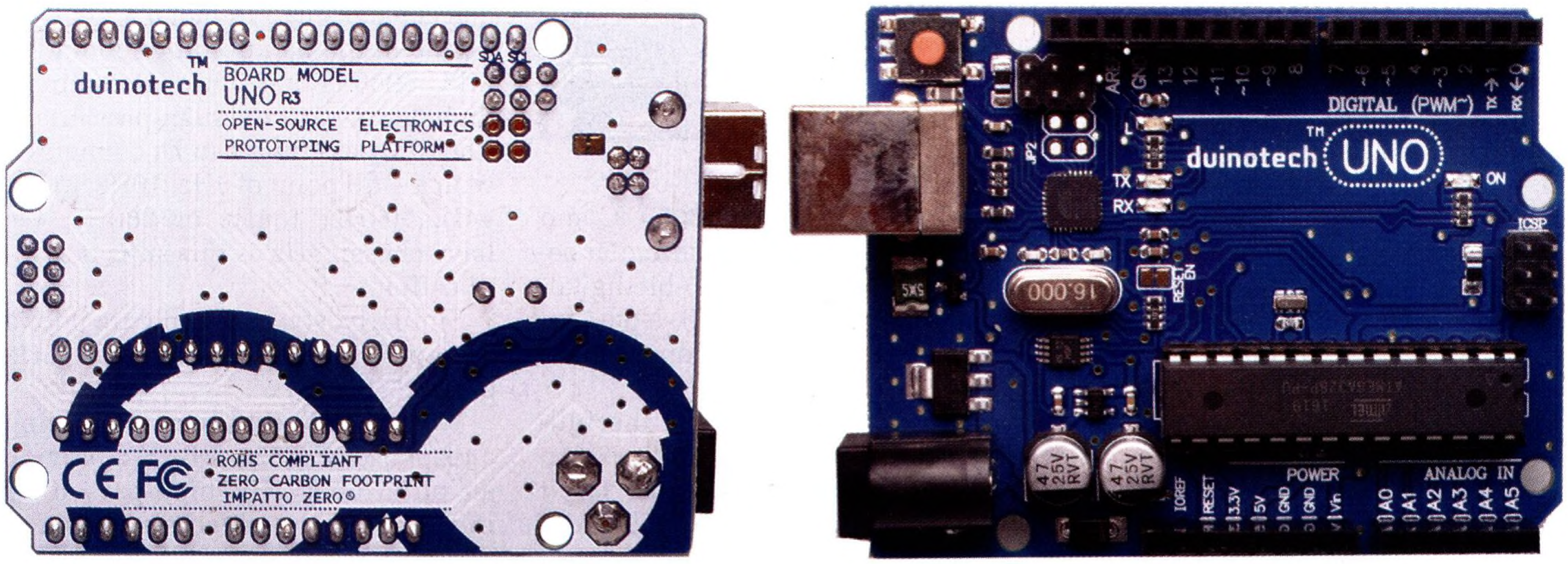


Fig.5: full circuit of the Earthquake Early Warning Alarm, including the components for the optional battery-backed supply, at bottom. The Arduino (MOD1) constantly reads the three acceleration values from MOD2, performs digital filtering and amplification, then decides whether to light up LED1 and sound the loud piezo siren.



The two sides of the Arduino Uno board, shown here close to life size (in this case the duinotech UNO from Jaycar – there are several compatible boards). The protoboard (opposite) simply plugs into the sockets on the edges of the board.

formula to compute the magnitude of the resulting X/Y low-frequency vector.

This is multiplied by a sensitivity factor, set using trimpot VR1, and if it exceeds an arbitrary threshold for more than about 200ms, the alarm is triggered.

To sound the alarm, output pin D12 is pulsed high and low at about 1Hz. When high, bright blue LED1 lights up and NPN transistor Q1 is switched on. This triggers the very loud piezo siren. Its volume and pitch are similar to a smoke alarm.

If an S-wave or surface wave is detected (by a similarly large excursion in the magnitude of the Z-axis measurement), LED1 and the piezo siren also light but they are on continuously, rather than pulsed.

This should alert you to the fact that you are currently experiencing an earthquake, in case the other signs (shaking, falling objects etc) are not obvious enough!

The unit can be mains-powered, via a USB port on a PC, from DC plugpack or the optional battery-backed supply (shown as MOD3 at the bottom of Fig.5) can be used.

This consists simply of a single Li-ion/LiPo cell combined with a small charger/power supply board. The battery is kept charged by the USB power supply when mains is present.

The battery can power the rest of the circuit for a few hours if there is a blackout.

While we haven't shown a solar panel connected there is provision for one – this could make the whole project fully self contained with solar-backed power if you wished to remotely use it. Virtually any 6V-12V solar panel could be pressed into service – the circuit only draws significant power from the battery when the alarm is going off . . . at which time a flattening battery is likely to be the least of your concerns!

To make construction easy, we wired trimpot VR1 directly to pins A0, A1 and A2. A1 is used as an input while A0 and A2 are programmed as digital outputs.

So we simply pull A2 high (to +5V) and A0 low (to 0V) just before measuring A1.

Therefore we read the position of the trimpot as a digital value and use that to determine the sensitivity.

This is computed exponentially so that the full range of rotation of VR1 gives about a 100:1 ratio between the level of vibrations needed to trigger the alarm at its two extremes.

We've set up the sensitivity so that at maximum, the unit will trigger on the tiniest tremor, while at the minimum setting, you'd probably have to hit it with a hammer to set it off.

Also note that to save power and simplify the circuit, we wired the warning LED in series with the base current limiting resistor for Q1.

The LED current is around 11mA [(5V-3.3V-0.7V)÷91Ω]. If you use a different colour LED, it will be driven at a slightly higher current, due to its lower forward voltage but it shouldn't be necessary to change the resistor value. (If you don't have a 91Ω resistor, 100Ω should be fine).

Construction

While you could build the device by wiring up the various components with flying leads, we used a protoboard to give a neater result, as you can see from the photos.

No component overlay is shown for the protoboard as there are so few components involved – all of the interconnection details are clearly shown in the photograph.

By soldering connected components close together, we only needed to run five wires, all of which you can see on the top of the board (two 0Ω resistors and three lengths of hookup wire; blue, green and red).

Wire links can be used in place of the 0Ω resistors if you prefer. (Wire links are also a tad cheaper!)

Start by soldering an 8-pin header to the MPU-6050 accelerometer module, then solder it to the prototyping shield.

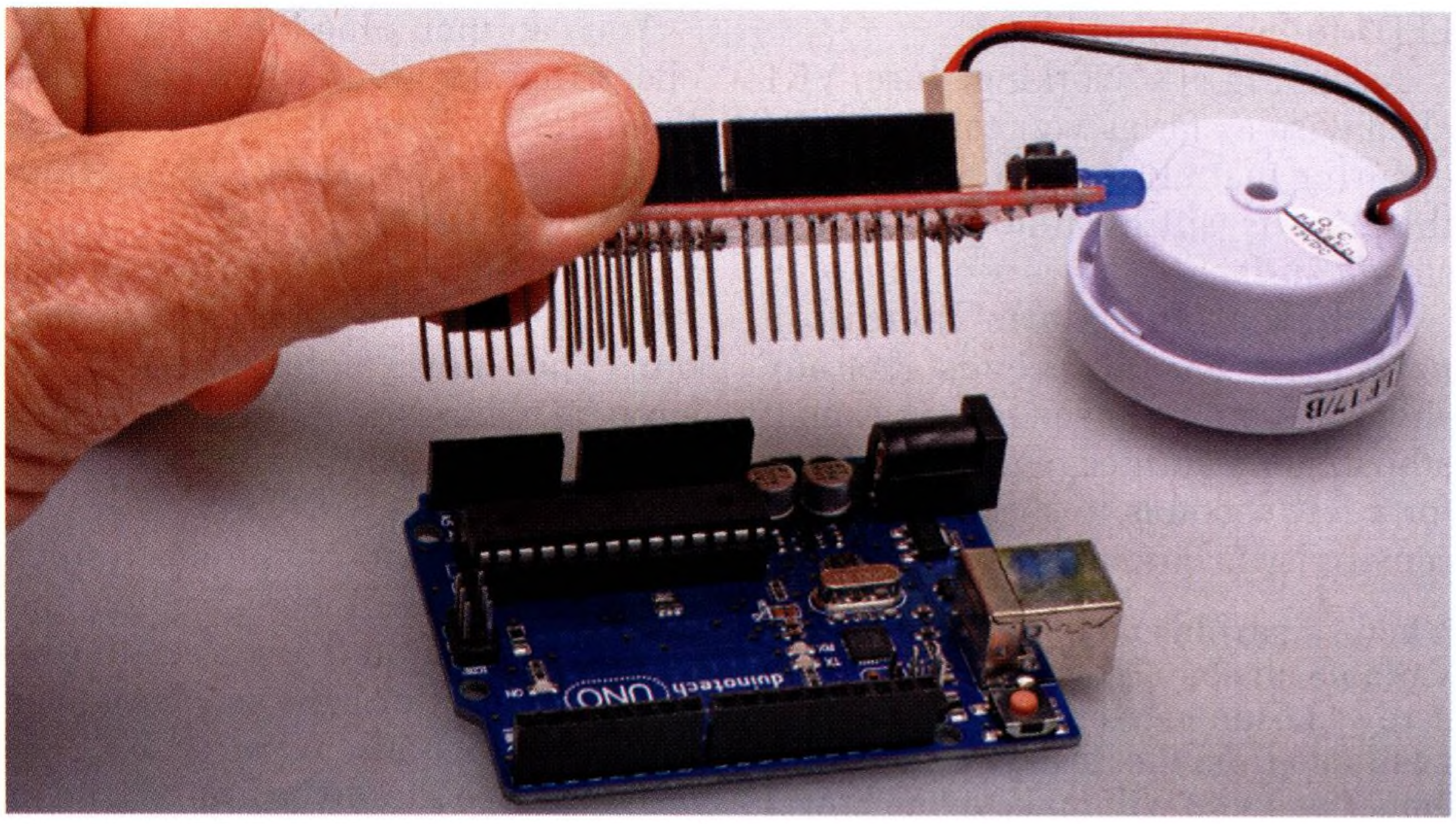
You will need to make four connections between this module and the shield headers: VCC to +5V, GND to GND, SDA to SDA and SCL to SCL.

Having done that, solder the 91Ω resistor from pin D12 to a pad near the edge of the board, then connect it to the LED anode. Connect the LED cathode to the base (middle pin) of Q1.

The collector of Q1 is the right-most pin when looking at its labelled face and this is connected to 5V. The remaining pin of Q1 goes to the negative pin of the piezo siren via CON1, with the positive pin wired to VIN.

We connected the piezo siren via a 2-pin polarised header. This is handy for testing since the siren is very loud,

If it's too loud while you're setting up, it can be temporarily muted by having something placed over its opening (a piece of sticky tape or insulation



The two PCBs simply plug into each other via the header pins on the top board and the matching sockets on the Arduino board, as shown here.

tape, for example) or placing it upside-down on your bench top.

If you don't want to solder the wires to a plug and the header to the PCB, you can directly solder the piezo wires to the board.

Finally, solder trimpot VR1 to pins A0, A1 and A2, as shown in the photo. That's it – those are all the connections you need. Solder the headers to the shield board, then put it aside while you program the unit.

Programming it

Download the Arduino sketch, named EarthquakeEarlyWarning.ino, from the SILICON CHIP website.

You will also need to have the Arduino IDE installed on your computer. The latest version can be downloaded for Windows, macOS and Linux from www.arduino.cc/en/Main/Software

Once it's installed, load it up and open the sketch.

There is one additional library that needs to be installed. It's called "Filters" and a zip file is included in the download package. Use the Sketch -> Include Library -> Add .ZIP Library menu option to install this file on your system.

Now plug the Arduino into your computer using a USB cable (without the shield, for now) and then go to the Tools menu and make sure the correct Port has been selected.

You can then use the Sketch -> Upload command to upload the code to the Arduino module.

Check the output at the bottom of the screen to make sure it has been compiled and uploaded without errors.

You can now unplug the Arduino module from your PC and plug the completed shield into it.

Then plug it back into your PC and open the Serial Monitor in the Arduino IDE. It's available under the Tools menu. Pretty soon, you should see an output like this:

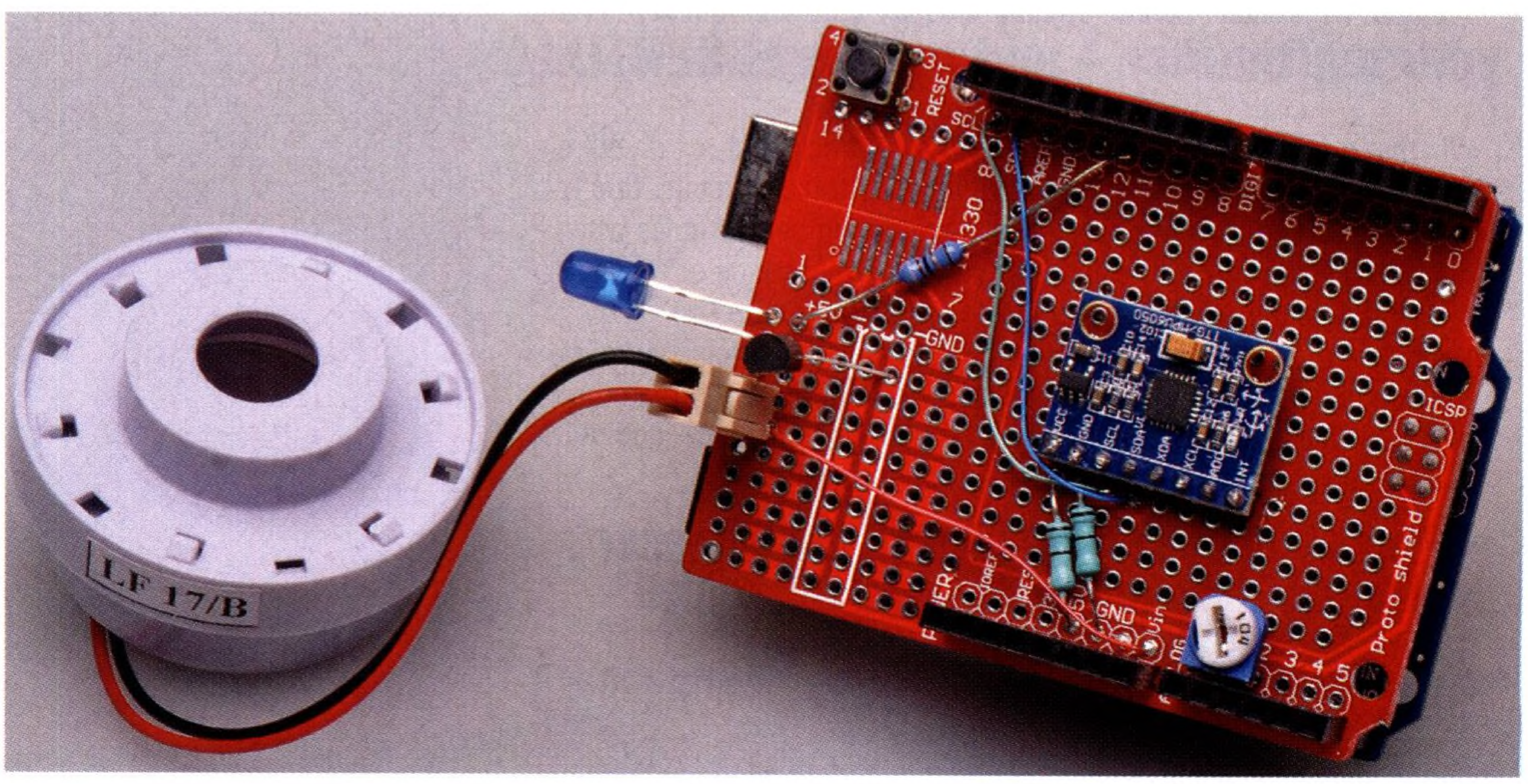
```
|XY| = 0.05, |Z| = 0.11
|XY| = 0.37, |Z| = 0.05
|XY| = 0.17, |Z| = 0.04
|XY| = 0.22, |Z| = 0.29
|XY| = 0.20, |Z| = 0.08
|XY| = 0.27, |Z| = 0.20
|XY| = 0.16, |Z| = 0.21
|XY| = 0.02, |Z| = 0.25
|XY| = 0.42, |Z| = 0.04
```

Parts list – Earthquake Early Warning Alarm

1 Arduino Uno or compatible board (MOD1)
1 MPU-6050 based accelerometer/gyroscope module with 8-pin header (MOD2; Altronics Z6324)
1 1-13V loud piezo siren (Altronics S6115)
1 Arduino prototyping shield PCB and header set
1 high-brightness 5mm LED (LED1)
1 BC337 NPN transistor (Q1)
1 100kΩ mini horizontal trimpot (VR1)
1 91Ω 0.25W resistor
1 2-pin polarised header and matching plug (CON1)
a few short lengths of light-duty hookup wire
1 small plastic box (eg, UB5 Jiffy box)
1 USB charger or other USB power source

Optional parts for battery backup

1 solar charger module (eg, SILICON CHIP Online Shop Cat SC4308)
1 small single-cell Li-ion/LiPO cell
1 short USB cable to suit solar charger module
1 6-12V mini solar panel, if required



Here's the protoboard with the LED, transistor, resistor and trimpot plus the MPU-6050 accelerometer board all mounted, as per the circuit overleaf. This assembly plugs into the Arduino Uno. The two light blue "resistors" (bottom of PCB) are actually 0Ω links.
The piezo "siren" is rather loud, as you would want it to be if it is to warn you of impending doom!
Not shown here is the optional battery and recharger – see full details of this in the article in SILICON CHIP, August 2017.

These are the readings from the accelerometer. |XY| is the dimensionless magnitude of the horizontal AC vector while |Z| is the magnitude of the AC component of the vertical vector.

If you shake the unit, you should see these values temporarily increase, then settle back towards zero. Rotating VR1 clockwise should cause them to increase and with VR1 fully clockwise, even the slightest nudge should cause LED1 to light up and flash.

Assuming it's working, turn VR1 clockwise as far as you can go while ensuring that LED1 remains off when the unit is sitting untouched on a steady surface.

Note that the alarm condition persists for several seconds after any shock so you will need to make small adjustments and leave the unit alone for a few seconds to see whether the sensitivity is correct.

A cheap 6V-12V solar panel, as shown here, a surplus mobile phone battery (both commonly available on ebay) plus one of the small "Elecrow" Li-Ion battery charger modules (available from the SILICON CHIP Online Shop, Cat 4308) will make a fine power supply for your Arduino-based Earthquake Early Warning Alarm, with the added advantage of making it completely self-contained: no external power supply is required!

You can then plug the siren in and check that it sounds when you bump the unit.

Setting it up

Mount the unit inside a box so that it's held firmly in place within that box. The "noise hole" of the piezo siren (ie, where the sound comes out!) should line up with a similar hole in the box.

The orientation of the electronics don't matter, as long as when the device is mounted on a wall (the preferred location), the accelerometer PCB is horizontal.

The device should be firmly fixed to a solid wall and if set correctly, it will sound the alarm when it experiences significant horizontal movement in any direction. Since the wall should be solidly fixed to the ground, that normally will only occur if the ground moves.

We can't rule out the occasional false alarm due to heavy vehicles, trains, nearby hammer blows or similar but you can turn VR1 anti-clockwise slightly if you are experiencing false alarms, reducing its sensitivity until they stop. **SC**