

## HD2 - ROLLING CODE 2-CHANNEL UHF REMOTE CONTROL

This kit is similar to the kit described in the accompanying article which was published in the Australian electronics magazine *Silicon Chip* in 7/2002.

There have been some major improvements and modifications made and the article is for reference only.

Up to 15 Transmitter units can be learned by one Rx unit. (The article says 16 but the technical manual says 15.) Press button 1 (the button all by itself) while simultaneously pressing the **LEARN** tact switch on the main board. You only have to do this briefly for under a second. But note it takes about **15 seconds** for the two units to internally connect and recognize each other. (During this 15 seconds it seems that one and only one keypress of the Tx unit will be recognised. Just disregard this. Wait the full 15 seconds until the two units have connected. Do not press the LEARN button again. Just wait 15 seconds.)

Tx units attached to any Rx unit can be unattached by pressing the LEARN button continuously for 8 seconds. The **VALID DATA** LED is on during these 8 seconds. As soon as the LED goes off then you know that all Tx units previously recognized by the Rx unit have now been unattached from the Rx unit.

The article makes reference to assembling the remote control Tx units. However, it is supplied here full assembled, tested, with a battery included and ready to go.

**Assembly.** Here are some more details.

- we have supplied 3 pins which you may use if you wish in the ANTenna, Ground and 12V+ positions. We have put two places for the Ground connection: one is next to the 12V+ point, and the other is on the opposite side of the PCB. Use which ever one best suits you best.

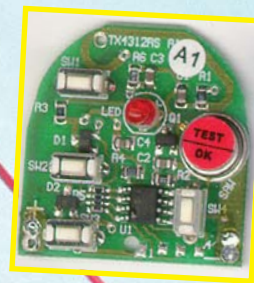
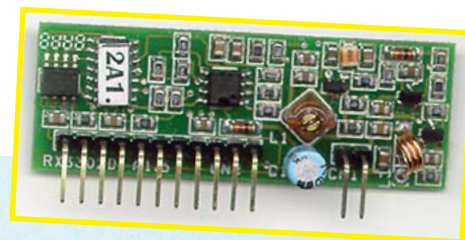
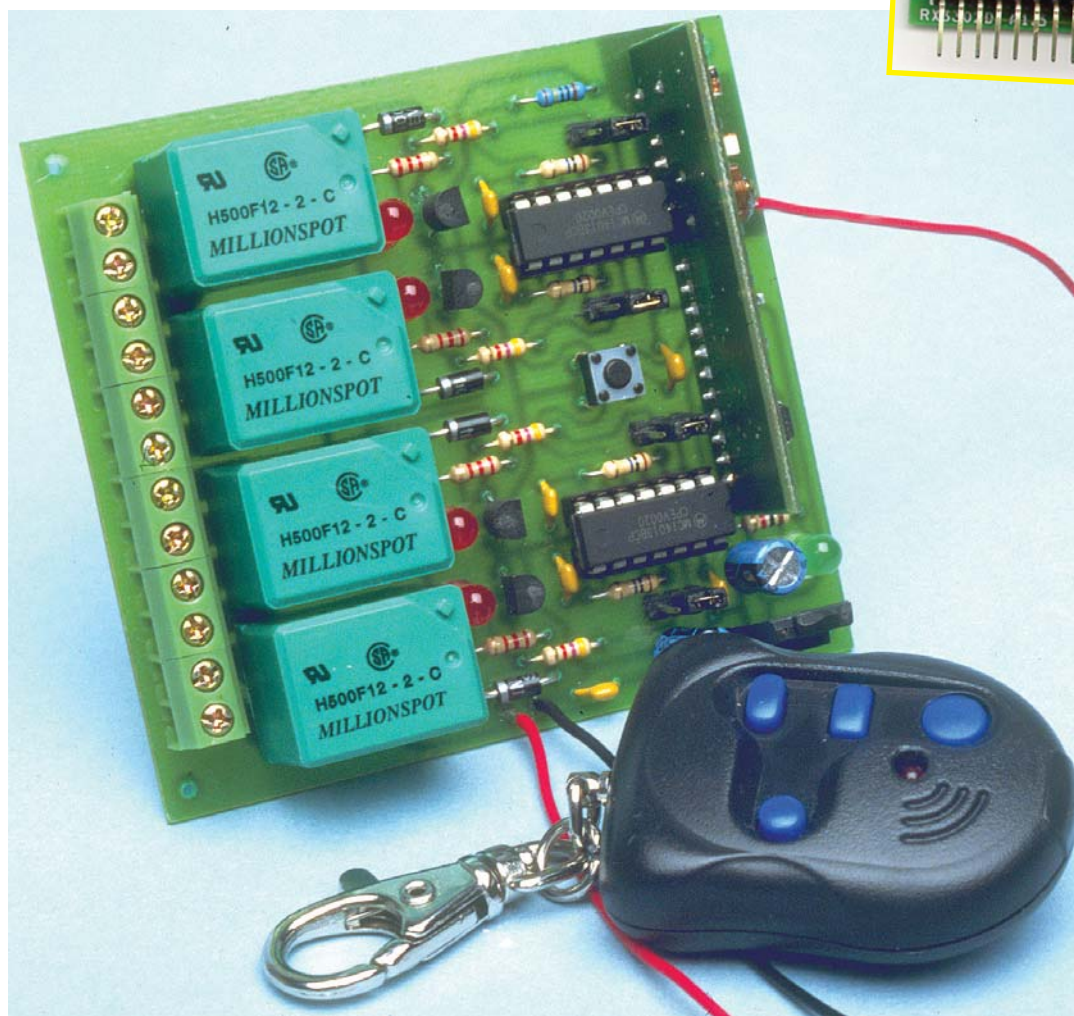
- follow the overlay for component placement.

The nearest thing you can get to “unbreakable” . . .

# A Rolling Code 4-channel UHF Remote Control

This is one very clever remote control. With rolling code, it's close-to-impossible to electronically “crack”. With four channels, all either latching or momentary operation, it's extremely versatile. With a sensitive prebuilt receiver, it's long range. With up-to-16 keyring-size transmitters, it's go-anywhere. And the kit even includes the keyring!

By Ross Tester



Whether you want to control a garage door or gate, a car and/or home alarm, or perhaps remotely turn lights or anything else on or off, this high-security system is just what you're looking for! Inset top right are the pre-built, aligned and tested receiver (top) and transmitter (bottom) modules, shown here same-size.

**W**e've presented a number of remote (radio) control devices in the past. None has been more secure than this one. To guess the code combination, you're going to need something like 23 billion years. But don't bother: the next time it's used, the code will have changed anyway.

That's the advantage of a rolling code (or "code hopping") system. We explain what this means, and does, later in this article.

Suffice to say at this stage that it makes one v-e-r-y secure system. For all intents and purposes, it is impossible to electronically "crack". Go on, give it a go – we'll see you in a few million years or so!

### The transmitter

It's probably not necessary to say it but there are two parts to this project, a transmitter and a receiver.

First of all, there is the tiny 4-channel "key-ring" transmitter which, fortunately, comes 99% pre-assembled.

We say fortunately because it's just about all SMD (surface mount devices) which, while not impossible for the hobbyist to work with, requires some rather special handling. You are spared that!

All you have to do with the transmitter PC board is solder on the two battery connectors and place it in the case (with battery).

The battery contacts are slightly different: the one with a spring is for the negative battery connection – it goes on the righthand side of the PC board with the only straight side of the PC board at the bottom.

You may find, as we did, that some of the holes for the battery connectors are filled with solder. This is easily melted during installation.

Once this is done, it's just a matter of assembling the board in its keyring case. Incidentally, the keyring case and battery are all supplied in the kit.

The transmitter itself is in the licence-free 433MHz LIPD band (it's

actually on 433.9MHz). As with most devices of this type these days, it is based on a SAW resonator (that stands for surface acoustic wave, so now you know!). This keeps the circuit very simple but enables excellent performance.

Without wanting to get into the nitty-gritty of SAW resonator operation, in essence it controls the RF side of things while a dedicated chip controls the complex digital coding.

The receiver (which we'll get to shortly) can handle up to 16 transmitters so if you have a really big family or maybe have a secure company carpark you want to give a certain number of people access to, you can do so simply by purchasing more transmitters.

The transmitter has four pushbuttons, one for each of the four channels.

Of course you don't have to use all four channels – just one will control

fact, anything your little heart desires.

### The receiver/decoder

Now we move on to the heart of the system, at least the bits you have to put together to make it work.

In fact, there are two parts to the receiver as well. There is a 433MHz receiver module which comes assembled, aligned and ready to go. This solders into an appropriate set of holes on the main PC board once you've finished assembling that board.

The main PC board contains the electronics which process the output from the receiver.

The receiver checks the incoming code and if valid, sends a signal to one of four outputs depending on which button was pressed on the transmitter).

From here, depending on how the four jumpers are set on the board, the signal goes either direct to an NPN transistor relay driver (for momentary operation – the relay is energised while

the button remains pressed) or to a D-type flipflop and then to the transistor relay driver (for alternate operation – press once and the relay latches, press again and the relay releases).

The flipflops change state (toggle) each time a positive going pulse appears at the clock input. This is achieved by the connection from the Q-bar output to the D input via an RC network.

The circuit has a power-up reset. When power is first applied, the Q outputs of the flipflops are reset low by the 0.1µF capacitor and 1MΩ resistor on the reset (S) inputs.

Reset is caused by sending the reset inputs of all flipflops high. Once the capacitor is charged, the voltage at the reset inputs of the flipflops falls to virtually zero, allowing normal operation.

It is perfectly acceptable to have a mixture of momentary and latched modes amongst the four channels. It's up to you.

But if you only require momentary action (for example, as needed by

### SPECIFICATIONS

- UHF (433MHz) licence-free (LIPD band) operation
- Long range – prototype tested to 100m+
- Pre-built and aligned transmitter & receiver modules
- Rolling-code ("code hopping") operation ( $7.3 \times 10^{19}$  codes)
- Receiver "learns" transmitter coding
- Receiver can handle up to 16 remotes
- Transmitter can handle any number of receivers
- 4 channels available, each either momentary (push on, release off) or latching (push on, push off) via jumpers
- Code acknowledge LED and channel status LEDs
- Each channel relay contacts rated at 28VDC/12A (single pole, changeover)
- 12V DC operation (6mA quiescent; 150mA all relays actuated)

most garage door openers, for example – but it's nice to know there are four channels available.

And before we move off the transmitter, up to three channels can be pressed simultaneously and the receiver will react to all three (it won't handle four at once, though).

Finally, as well as multiple transmitters, you can use more than one receiver if you wish.

Each receiver "learns" its transmitter(s) so you can have a multiple system controlling, for example, the garage door, the car doors, the car alarm, the home security system – in

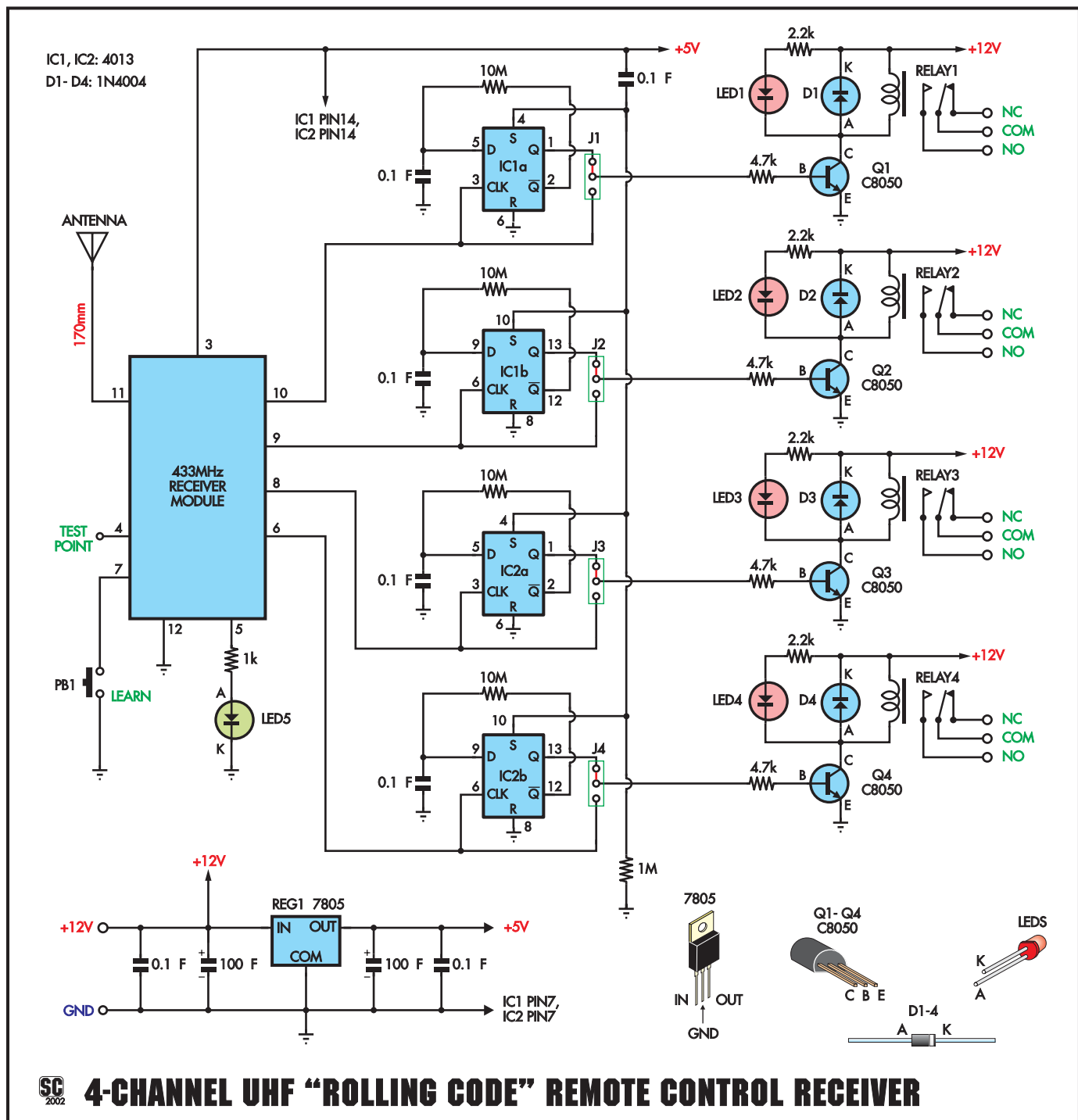


Fig.1: the circuit of the "control" section of the receiver unit. We haven't attempted to show the 433MHz receiver itself, nor the transmitter, as these are both pre-assembled modules, saving you a lot of difficult work!

some door openers/closers) the flip-flops, along with their associated RC network components and the four header pin jumper sets, could be left out of circuit. (You'd then need four links on the PC board to directly connect the receiver outputs to their respective transistors.)

Along with spike suppression diodes across each relay coil, part of each relay driver circuit also includes

an acknowledge LED to give a visible output of what's happening.

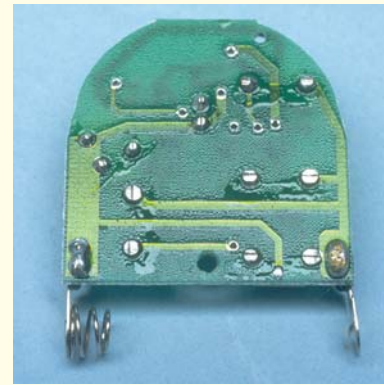
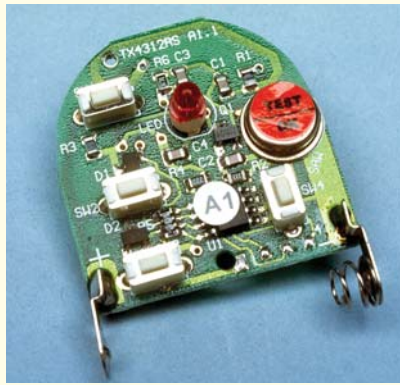
There is also a "valid signal acknowledge" LED attached to the 433MHz module, which lights when valid code is being received.

Each of the four identical relays has contacts rated at 28VDC & 12A, so can be used to control significant loads. The wide track widths on the PC board also allow high currents.

The relay contacts could, of course, also be used to switch higher-rated relays or you could replace the acknowledge LED with an opto-coupler.

The relays themselves are single pole but have normally open (NO) and normally closed (NC) contacts. These states refer to the unenergised state of the relay (ie, the NC contacts go open when power is applied to the relay coil and vice-versa).





## ASSEMBLING THE REMOTE CONTROL:

The photo above shows seven of the eight parts you should find when you take the bits out for the remote control (the battery is missing!).

Above centre shows the two battery connectors soldered in place on the top of the PC board, above right shows the same thing from the other side.

Don't mix up the connector with spring and the connector without. Finally, the photo at right shows the PC board in place, with battery, in one half of the keyring case. The blue pushbuttons are all on one plate – they fit in as shown but can easily fall out.

As you push the two halves of the case together, make sure the pushbutton plate stays in place. The keyring itself also fits into the notch in the case as you push the two halves together.



The only other components on the board are a simple 5V regulated supply, consisting of a 7805 3-terminal regulator and a couple of capacitors. This supply powers the 433MHz module and the 4013 flipflops. The relay coils are powered direct from the 12V supply.

## Construction

Start by soldering in the two battery terminals to the transmitter PC board, in the positions shown in the photographs.

Place the completed board in the keyring case, making sure the push-buttons stay in position.

Push the two halves together with the battery in place (and the right way around – see pictures), with the keyring clip sandwiched between the two halves.

One screw holds the two halves of the transmitter case together.

Press each of the four buttons and ensure that the LED lights each time.

If it does, you can be reasonably sure that the transmitter is working properly. Put it to one side while we move on to the receiver.

## Receiver board

As usual, check the receiver PC board for any defects before assembly. Then solder in the resistors, capacitors, diodes, IC sockets (if used) and the four header pin sets (which select momentary or latching function).

If you use IC sockets, make sure they go in the right way around – the notch is closest to the edge of the PC board.

The “learn” pushbutton switch solders in place between the IC sockets. These have two pairs of pins which are not identically spaced – the switch should be an easy fit in the PC board if you get it the right way around. If in doubt, check the “closed” state with your multimeter.

Now solder in the semiconductors – the regulator, diodes, transistors and

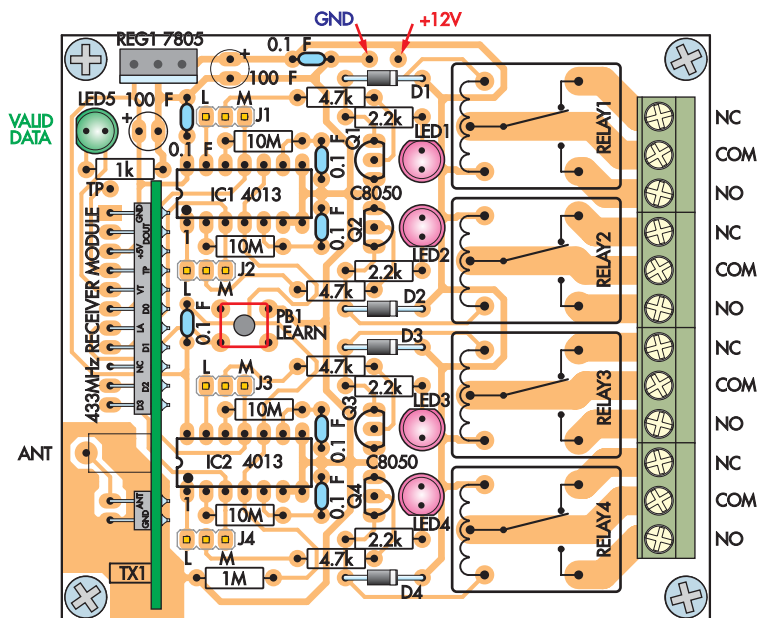
the LEDs as shown on the component overlay. Watch the LED and transistor polarities – each is opposite to its neighbour!

The last things to be soldered in place before the 433MHz receiver module are the four relays and the six output terminal blocks. The relays will only go in one way but the terminal blocks could be mounted back-to-front, making it almost impossible to get wires into them! (The “open” side of the terminals go towards the edge of the board, in case you were wondering!)

At this point, check your assembly for any solder bridges, dry joints or missed joints.

You might also now solder in the three wires – two connect 12V power while the third is the antenna. Make the power leads the necessary length to reach your supply.

When the antenna wire is soldered in, measure exactly 170mm from the PC board and cut the wire to this



length. This makes it resonant at 433MHz.

You should not have any bare wire(s) emerging from the end of the antenna – this could short onto something nasty and do you/it/something else some damage! If necessary, wrap a little insulation tape around the end of the antenna wire – just in case!

Plug the two ICs into their sockets, again watching the polarity. The notches should line up with the notches in the sockets (assuming you got the sockets right!)

OK, we're almost there. Place the receiver module in its appropriate holes along the edge of the PC board. It will only go one way (incidentally, take care not to move the coil or touch the trimmer capacitor).

Solder each of the module pins into position (there are 13 of them – don't forget the two by themselves) and your receiver is finished.

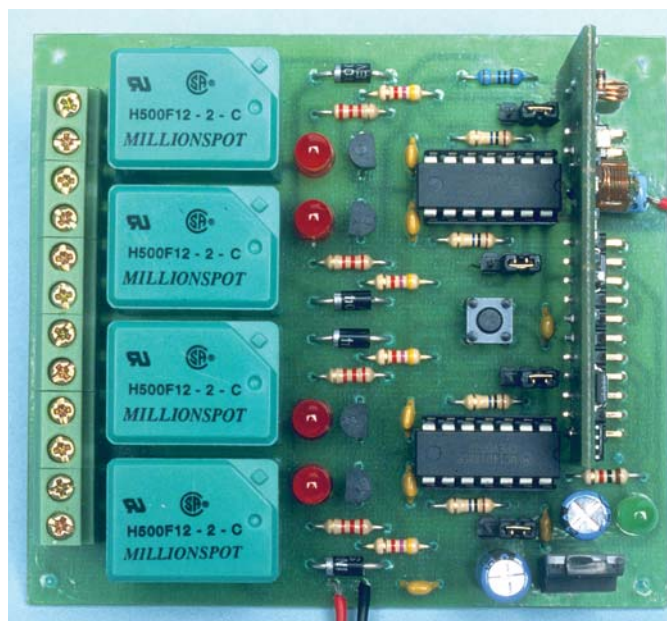
## Power supply

The receiver unit is designed for 12V battery operation and power requirements are pretty modest. At rest, (ie, no relays operating), it draws only 6mA and even with all relays actuated, the current is just a smidgeon under 150mA.

Therefore, most alarm-type batteries (eg, SLAs) will be more than adequate.

We had it operating for a couple of weeks on a 7Ah 12V gell cell, periodically pressing the remote control just for the hell of it, without recharging the battery. In fact, at the end of this

**Fig.2 (above): the component overlay of the receiver module with the full-size photograph at right. Just to confuse you, we've shown the board turned 180° compared to the diagram above!**



time the battery voltage changed only a few tens of millivolts – probably not much more than you would expect during shelf life.

Therefore, just about any 12V battery would be acceptable, even a couple of 6V lantern batteries in series or even 10 C or D-size Nicads.

Of course, you could also use just about any garden-variety 12V or 13.8V DC (nominal) plug-pack supply.

The relays won't worry about a few extra volts and the circuit has the on-board 5V regulator to ensure the electronics get the right voltage. Any DC plugpack over about 200mA capacity should be fine.

## Learning and testing

Looking at the board with the outputs/relays on the left side, move all header pins to the right side (latching).

Apply power and you should see absolutely nothing happen. So far, so good.

Now press the “learn” button once, then within 15 seconds press button one on the keyring transmitter for a second or so. Button one is the one all by itself on one side of the transmitter. The receiver then learns the encryption from the keyring transmitter – and remembers it.

Now all four buttons on your transmitter should alternately close and open the appropriate relay and light/switch off its associated LED.

Change the four jumpers over to the

opposite way and all four buttons should now pull in a relay and light a LED while ever they are pressed – and release it/dim it when let go.

And that's just about it. Now all you have to do is select the jumpers the way you want them and connect the external devices you wish to control. Note that each relay has a normally open and normally closed connection as well as common, so you have a lot of flexibility at your disposal.

## Want even more security?

We mentioned before the one major drawback with any remotely controlled security application, whether that



# What is “Code Hopping” or “Rolling Code”

These two names usually refer to the same thing – in a nutshell, a security system for a security system.

It's a way of preventing unauthorised access to a digital code which might be transmitted via a short-range radio link to do something: open a garage door, lock or unlock a car and perhaps turn its own security system on and off – and much more.

But before we look at these terms, though, let's go back in time to the days before code hopping and rolling code.

Short-range radio-operated control devices have been around for a couple of decades or so (at least, in any volume). The earliest ones that I remember simply used a burst of RF, at a particular frequency, with an appropriate receiver.

It's not hard to see the shortcomings of such devices. Simply sweeping the likely band(s) with an RF generator attached to an antenna would more often than not achieve the desired result (desired for the intruder, that is).

It didn't take long for crooks to latch on to this one (do you like that metaphor?). So manufacturers decided to make it a bit harder for them by modulating the RF at a frequency (or indeed multiple frequencies in some cases) “known” to the receiver.

Some used the standard DTMF tones generated by phone keypads because they were very cheap and made in the millions.

“Oh, gee,” said the crooks. Now we'll have to use an RF oscillator with a modulator. Or maybe even a DTMF keypad!”

Duh! (Still, it probably seemed like a good idea at the time. . .)

Ever one step ahead, the manufacturers went with this (then) new-fangled digital stuff and made each transmitter send a particular code which was matched to the receiver. This was usually done by way of DIP switches in both transmitter and receiver.

With eight DIP switches (probably the most common because 8-way DIP switches were common!), you would have  $2^8$  or 256 codes available. So you and your next-door neighbour could have the same type of garage door opener on the same frequency and the odds would be pretty good that their door would stay down when you pressed your button.

The problem with this, though, is that the transmitter spurted out exactly the same code every time (unless, of course, both sets of dip switches were changed). Enter the crooks again.

With a suitable receiver, called a “code grabber”, if they got within a few tens of metres of you they could scan for the RF signal and record your code without you knowing anything about it (for example, as you left your car in a carpark and pressed the button on your remote to lock the doors and turn on the alarm).

Once you'd gone, they simply “played it back” using the same code grabber. Presto, one missing car. Or one house burgled, etc etc.

Even without a code grabber, a smart intruder with the right equipment using digital techniques and trying eight combinations per second, could crack the code in no more than 32 seconds – and probably much quicker.

It's hard to believe the gall of some organisations openly flogging such devices, euphemistically disguising them (justifying them?) with names such as vehicle lockout recovery systems or disabled vehicle recovery systems. Then again, lock picks are sold for professional locksmiths, aren't they?

Now we move on a little. Microchip, the same people who brought you those ubiquitous PICs, invented a system called KEELOQ – better known to you and me as a rolling code.

What this does is simply present a different code every time the transmitter button is pressed. Of course, that's the easy part. The really clever part is that the receiver “learns” the algorithm which controls the code so it knows what code to expect. Once learnt, the receiver is effectively “locked” to that transmitter.

Actually, it's even cleverer than that, because the transmitted code is, for all intents and purposes, random (as far as any external device is concerned). But the receiver can still work out what the code is going to be in advance. If it gets the right code, it actuates. If not – you're out in the cold, baby!

The chances of the same code being transmitted twice in a person's lifetime is possible – but remote (at four transmissions per day, every day, it's reckoned to be about 44 years!)

Heart of this system is a Microchip proprietary IC, the HC301. It combines a 32-bit hopping code generated by a nonlinear encryption algorithm with a 28-bit serial number and six information bits to create a 66-bit code word. The code word length eliminates the threat of code scanning and the code-hopping mechanism makes each transmission unique, rendering code capture and resend techniques useless.

Even if it didn't code-hop, 66 bits allows  $7.3 \times 10^{19}$  combinations, which according to Microchip would only take 230,000,000,000 years to scan!

The chip itself is also protected against intrusion. Several important data are stored in an EEPROM array which is not accessible via any external connection. These include the crypt key, a unique and secret 64-bit number used to encrypt and decrypt data, the serial number and the configuration data.

The EEPROM data is programmable but read-protected. It can be verified only after an automatic erase and programming operation, protecting against attempts to gain access to keys or to manipulate synchronisation values.

If the code is changed every time a button is pressed on the transmitter, what happens if, say a child starts playing with the remote control and continually presses buttons away from the receiver? OK, here's where it gets really clever (and you thought it was clever enough already, didn't you?).

If the button is pressed say 10 times while out of range of the receiver, no problem. But if it is pressed more than 16 times, synchronisation between the two is lost. However, it only takes two presses of a button in range to restore sync. No, we don't know how either. That's Microchip's secret!

And speaking of button presses, there are a couple of other clever things they've done. At most, a complete code will take 100ms to send (it could be as low as 25ms). But if you manage to hit the button and release it before 100ms (difficult, but possible), it will keep sending that complete code. If you hold down the button, it will keep sending that same code. And if you press another button while the first is held down, it will abort the first and send the second.

As you can see, KEELOQ is a very robust system. Sure, it's not absolutely foolproof – nothing is (eg, there's not much protection if they simply steal your transmitter!). But for most users, it gives almost total peace-of-mind. That's why the system has been adopted by so many vehicle entry/exit and alarm system manufacturers, access controllers and so on.

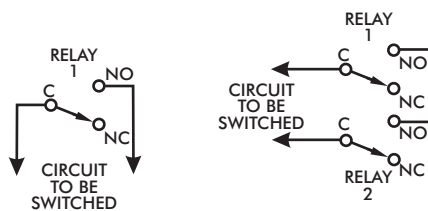
And that's the system that's used in the remote control unit presented here.

be for a car, a building or anything else: what happens if someone pinches your remote control?

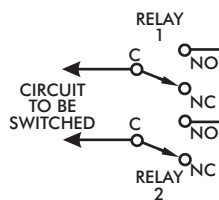
It is possible to protect yourself against the casual button pusher on a stolen control – at least to some degree.

Having four channels at your disposal, in this remote control system, gives you the possibility of increasing security rather significantly, simply by using a combination of keys on your remote.

It is “normal” to use one button to achieve a certain function. But what if you used two buttons? It’s possible because when you press the second button, even while holding down the



**Fig.3a (left):** conventional device control with one relay. Adding a second relay in series (fig 3b, right) increases security against the casual button pusher. Both buttons must be pressed at the same time for the device to actuate.



first, the second button’s code is sent.

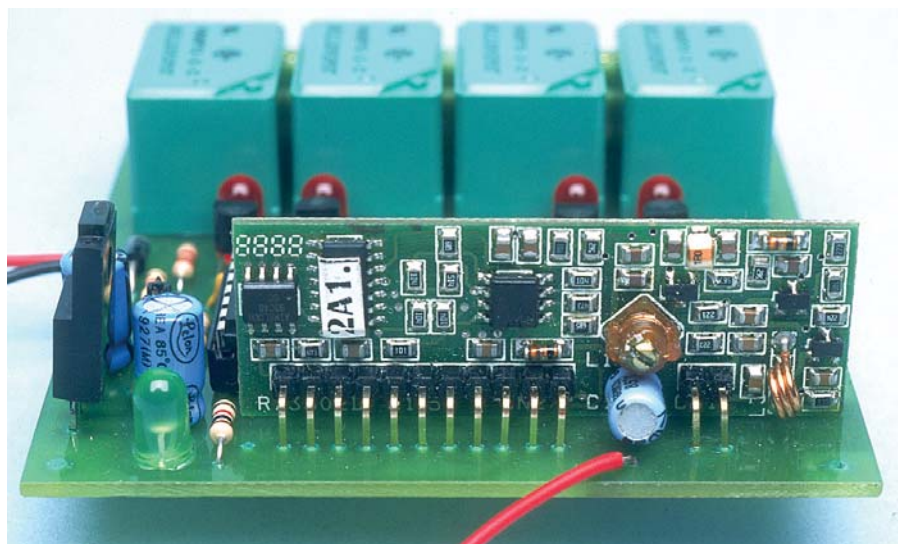
So if you made one button a “momentary” and linked another button’s relay contacts through the first button’s relay contacts, you have the situation where pressing single buttons (as most people would do) wouldn’t

achieve a thing.

Only you know which two buttons (or even three buttons) have to be pressed to achieve a certain function.

Fig.3 shows what we mean – the exact combination of buttons is entirely up to you!

SC



A close-up look at the receiver module soldered into the main PC board. Do this last, as explained in the text.

## Parts List – 4-Channel Code-Hopping Remote Control

- 1 TX-4312RSA 4-channel keyring rolling code transmitter assembly
- 1 RX3302D A1.5 433MHz rolling code receiver module
- 1 PC board, coded K180, 86 x 78mm
- 4 miniature relays, SPDT, PCB mounting, 12V coils (Millionspot H5000xx)
- 1 ultramini pushbutton switch, PC mounting, N-O contacts
- 6 interlocking 2-way terminal blocks, PC mounting
- 2 14-pin DIL IC sockets (optional)
- 4 3-way header pin sets, PC mounting
- Red & black insulated hookup wire for power connection
- 1 200mm length insulated hookup wire for antenna (see text)

### Semiconductors

- 2 4013 dual “D” flipflops (IC1, IC2)
- 4 NPN general purpose transistors (C8050 or similar) (Q1-Q4)
- 1 7805 3-terminal regulator (REG1)
- 4 1A power diodes, 1N4004 or similar (D1-D4)
- 4 red LEDs, 5mm (LED1-LED4)
- 1 green LED, 5mm (LED 5)

### Capacitors

- 2 100μF, 16VW PC mounting electrolytics
- 7 0.1μF polyester or ceramic (monolithic 5mm)

### Resistors

- 4 10MΩ
  - 1 1MΩ
  - 4 4.7kΩ
  - 4 2.2kΩ
  - 1 1kΩ
- OR