“Single Channel R/C” emulation - a blend of old and new technologies.

Phil_G philg@talk21.com

I thought we’d begin this little odyssey with a brief recap of Single Channel (S/C) operation as it was in the 60’s. If you were around during those early days a lot of this will be familiar and it might even jog a few dormant memories, some good and some not so good!
If you arrived on the scene a little later, bear with me, you might take some convincing that we were actually able to fly these things at all, let alone *enjoy* it…

In the 1960’s the most affordable and therefore the most popular radio set was a single channel transmitter and a super-regenerative receiver, with a rubber-driven escapement controlling just the rudder.

The super-regen type of receiver would respond to almost any transmitter in range and therefore models had to be flown one at a time. Remarkably, this rarely caused any problems whatsoever!

A typical 1960’s single-channel transmitter had just two features, and one of those was the on/off switch. The other, the one and only control, was a push-button, a bit like your doorbell.

In the receiver was either a relay or a transistorised switch, which followed the operation of the transmitter button, so in fact the whole outfit was basically a remotely operated on-off switch. The receiver relay was wired so that it switched power to the coil of the escapement when the transmitter button was pressed and so the armature of the escapement followed the operation of the pushbutton. The escapement was driven by a long rubber motor, wound up with a few hundred turns and attempting to spin the rotor of the escapement, which it was prevented from doing by a pawl on the resting armature. In a sequential escapement such as the Elmic Conquest, operating the armature released the rotor for a quarter of a turn, producing right rudder. Releasing the armature would permit the rotor to turn another quarter turn, settling at neutral rudder. The next armature operation would release the rotor for another quarter of a turn, giving left rudder, and on releasing the button (and therefore the armature) it would complete the 270-degree cycle and come to rest at neutral. Sequential escapements simply gave left and right alternately, and you had to remember which way you turned last. Operation was very fast as the rubber powered rotor was not slowed mechanically, the escapement emitting a loud ‘clack’ as it deflected the rudder.

A compound escapement such as the Elmic Commander had stops at 90 and 270 degrees, but omitted the ‘neutral’ position at 180 degrees. The rotation had to be mechanically slowed to allow a reasonable speed of keying the button, and this is how the ‘one press for right’ and ‘two-pulses for left’ method was achieved. The Compact had a further stop at about 355 degrees, where the ‘kick-up’ elevator function would be triggered on the third press of the button, and a ‘quick-blip’ circuit which triggered the throttle escapement if the rotation passed through 45 degrees whilst the button was released. Timing the keying of the button was quite critical and had to be well rehearsed - flying a 3-function single channel model was a real skill!

A development of the Elmic rubber-driven escapement was the motorized equivalent – for example the Climax ‘Unimite’ compound 'motorized actuator' (they didn't call it a servo back then). Both worked in much the same way, one press for right, two for left.
The Unimite was incredibly noisy, and could be plainly heard from a model flying hundreds of feet above.

Single-channel models were more often than not rudder-only and inherently stable models were the simplest and most successful.
On the other hand, by choosing Compound escapements, a fully equipped single channel model would have rudder, one way elevator (usually up but down was often used in gliders) and a 2 or 3-position throttle. All these controls were 'bang-bang' all or nothing, there was no proportional movement, no stick to follow. All the control surface movements involved nothing more than rubber bands and mechanical means, no electronics were used other than in the radio itself.

The radio equipment was rarely 100% dependable in those days, and the rubber band driving the escapement could easily run down during a long flight, which of course meant no further control was possible. Radio failures, flat batteries, interference and expired rubber meant that retrieving a fly-away was an occasional necessity. This is where the noisy Unimite was a godsend for finding a S/C slope-soarer buried under four feet of bracken. Made the sheep scatter like ninepins too.

Despite the shortcomings of the equipment, it was of course, a lot of fun – and fun is the reason we do this hobby, right? But wouldn't it be great if we could have a modern, reliable equivalent of the old single channel sets?

So.
Enough of the history and on to the project.

A Single Channel Transmitter Encoder for 2.4g Modules: **Phil_G philg@talk21.com**



The heart of the project is a software driven custom encoder written in pic assembler, which employs an accurate state-machine model of the escapement. From this model the encoder creates 4 channels of standard PPM for rudder, kick-up elevator and quick-blip throttle, and feeds a standard 2.4g DIY or Futaba-fitting RF module. The complex escapement emulation coded into the PPM stream enables a standard 2.4g receiver to drive the rudder and optionally the elevator & throttle servos in the exact manner of the old clockwork bang-bang escapements. Additionally it can drive a genuine period escapement or actuator, though obviously trading reliability for authenticity.

Note that all the emulation is in the transmitter encoder. The model carries a standard, unmodified 2.4g receiver and modern proportional servo(s). A further advantage of this approach is the weight saving – 'new age' S/C weighs a fraction of its bulky ancestors.

The four PPM channels created by the encoder each serve a different purpose. At the receiver, the required function is selected by plugging into the appropriate channel and several variations of equipment and functions are possible.

The standard (Futaba convention) encoder gives receiver channel assignments as follows:

Ch1=rudder, sequential or compound (Elmic Conquest or Commander/Compact emulation)

Ch2=optional kick-up elevator (Elmic Compact emulation)

Ch3=optional quick-blip two or three-position throttle (Elmic Corporal/Fred Rising emulation) or 15 second timed motor start in sequential mode

Ch4=follows the tx button press for the option of a real escapement driven via a relayless driver switch on the rx. 1ms=off, 2ms=on.

The optional JR/Spektrum version of the encoder inverts the PPM stream and gives receiver channel assignments as follows:

Ch1= optional quick-blip two or three-position throttle (Elmic Corporal/Fred Rising emulation) or 15 second timed motor start in sequential mode

Ch2= rudder, sequential or compound (Elmic Conquest or Commander/Compact emulation)

Ch3= optional kick-up elevator (Elmic Compact emulation)

Ch4=follows the tx button. 1ms=off, 2ms=on. This would probably be unused in a buddy box situation.

The encoder is based on the PIC 12F675 with software written entirely in assembler. It has connections for a PP3, a 9.6v nimh or 3S lipo power, an input from the Single Channel pushbutton, a PPM output to the module, a buzzer and two status LEDs. The module runs on the full battery voltage whilst the PIC takes its power from an on-board 5v regulator. A diode protects against accidental reverse polarity and capacitors smooth & decouple the supply. The two LEDs indicate motor status & rudder direction and are also used in the configuration mode.

The 2.4g module and its associated receiver are bound together using the normal method - the exact bind procedure varies between manufacturers. Both Futaba-fitting and DIY (“hack”) Corona or Frsky modules are ideal – in fact both are in regular use as S/C transmitters.

Operation is very simple.
The transmitter operates in one of two emulations – sequential like the Elmic Conquest, and Compound like the Elmic Commander or Compact. The current emulation mode is one of several user-selectable options – this is explained in the ‘Configuration’ section.

Sequential mode:
In “Sequential mode” each press of the transmitter button will give alternately left then right then left, etc. This drives just the rudder via a servo on receiver channel 1 exactly like an Elmic Conquest. In the old Single Channel days you had to remember which way you last turned and expect the opposite on the next press – however we now have a tell-tale LED which indicates the direction of the next turn – red for left, green for right.

The software has an additional feature for sequential flyers. If the model is electric motor powered then we need some way to operate the ESC. Consider for a moment what we are trying to emulate. A small diesel would be refuelled, started then run flat out until the tank drains a few minutes later, when it cuts and down we come.
For electric flyers we need to replicate this sequence, which the encoder does using the throttle channel, where of course the ESC would normally be connected.
For this purpose, a motor run timer is included which can be set to one, two, or four minutes.
Once one of the three timer values is selected, on the next power-up, the encoder is primed for a motor run and will emit a trill sound as a warning. Rudder operation can be checked, and when you’re happy and ready to go, hold rudder for a full 5 seconds. This will start a 5 second pre-start delay with more urgent audible pips before slowly building the throttle to full, where it remains for the duration of the motor timer, at which point the motor automatically cuts just as a diesel would run out of fuel.
Operational safety being paramount, during the pre-start delay a warning LED flashes and a buzzer sounds to indicate that the motor is about to start itself up. This is a final warning to get your fingers away from the prop. If at any stage before launch you wish to terminate the countdown or cut the throttle, momentarily switch off the transmitter. The throttle will immediately be closed and the timer will be cancelled. This is a safe ‘panic’ response to an unexpected arrival or crash, where the motor could be stalled, and there is no time to cycle through throttle settings. Simply switch off the transmitter and the motor will cut.

Compound mode:
In “Compound mode”, the rudder servo behaves just like an Elmic Commander, with one press for right and two presses for left.
In this mode it also emulates the Elmic Compact for kick-up elevator using three presses, and the Elmic Corporal for two-position throttle using a quick-blip of the button. There are no motor run timers in compound mode – you control the throttle throughout the flight.
A further option for throttle is ‘Fred Rising’ emulation, giving three throttle positions in sequence.
‘Half’ throttle can be configured to be between 20% and 70% as some motors have a very non-linear throttle response.
The current throttle position is indicated by the same LED used in sequential mode for the motor start – off is throttle closed, flashing is half throttle, on is full throttle. A ‘fast’ flash indicates that the next quick-blip will give high throttle. Conversely a ‘slow’ flash indicates that the next quick-blip will give low throttle.

Similarly the red/green direction indicator displays the current direction of turn – red for left, green for right, and a combined red/green when kick-up elevator is selected.

Compound escapements were cyclic, rotary devices with stops at 90 and 270 degrees for right & left. ‘Left’ cycled through ‘right’ before stopping, and ‘right’ returns to neutral via ‘left’. The software mimics the action of the rotary armature and pawl stops exactly.
A reasonably fast servo is necessary to accurately mimic the operation of an escapement – modern micro-servos are ideal.

In either emulation mode a genuine single channel actuator can be directly controlled via receiver channel 4. The transmitter button presses are repeated on channel 4 such that we can drive a plug-in channel switch (the type you might use for lights etc) to operate an actual rubber driven escapement or relay. The escapement itself can be either compound or sequential.
This series of projects includes a circuit for a suitable channel switch capable of driving escapement coils etc, up to one amp, although these are probably cheaper to buy than to build – they are often advertised for landing lights, night-flying lights etc.

Configuration mode:
Different types of model have different requirements, for example servo directions and throws may differ between models.
The configuration mode allows you to change the set up to suit your installation. There are eight user-selectable options:

The current emulation mode can be selected as sequential or compound.

In compound mode we have throttle control via the quick-blip, and so the ESC in an electric model will work as expected.
In sequential mode however we only have rudder control. The encoder therefore has an optional motor run timer which can be set to one, two or four minutes. For a glider or a non-throttled IC powered model, these settings would be de-selected.

In either mode, you may find that the surfaces move the wrong way, and so the encoder includes individual servo reversing for rudder & elevator. It also has a high or low travel volume option.

To enable single-handed range checking, 'range check' mode simply flips the rudder from side to side about once per second.
Once selected, you can then walk away with the model without any need to touch the transmitter. Whilst operating in range test mode, holding the button until it beeps will switch to a slow sweep. Holding again will return to the left/right flip. Reset this option before flying!
Sweep is better for finding nulls or breaks in transmission at long range, and left/right flip is easier to see and to hear.

The selection of options acts like an 8-way dipswitch, working down the individual switches from 1 to 8. At each of the 8 settings, the LED will flash either red or green. Green means that option is deselected, red means the option is selected. A push of the button 'flips' the current option – if it was enabled (red) the button will flip it to disabled (green).

To enter configuration mode, hold the button down briefly whilst turning on the transmitter.
If the button is untouched, the current configuration will be displayed as follows.
The yellow LED will flash once for each of the eight 'switches'. At the same time, a pip will sound and either red (selected) or green (deselected) will flash. After the eight flashes, there is a 'di di dit' sound, a brief pause and the cycle repeats.

'Di-di-dit' means that the eight configuration switches have been written to flash, in other words the current options have been saved for next time the transmitter is powered up.

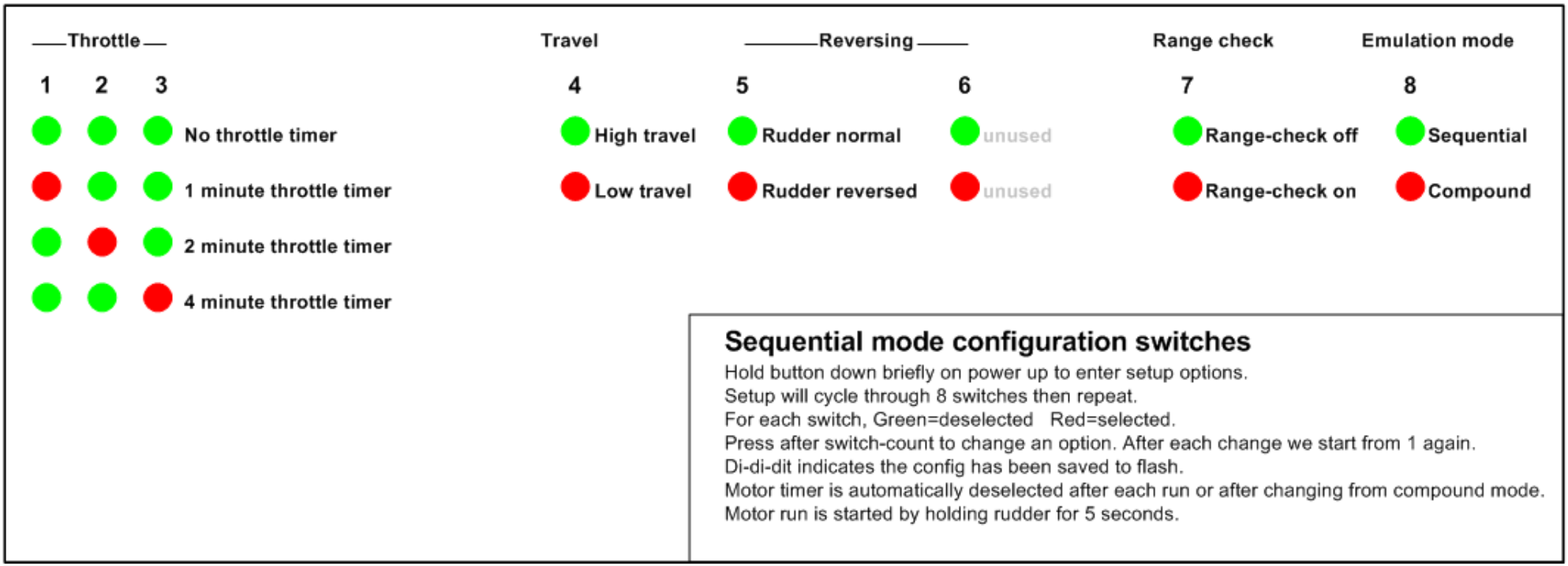
To change an option, simply count the pips and press & release the button immediately after the switch number you wish to change.
This will 'flip' that option from green to red or vice versa.
Remember to wait for the 'di-di-dit' sound if you want your changes to be saved.

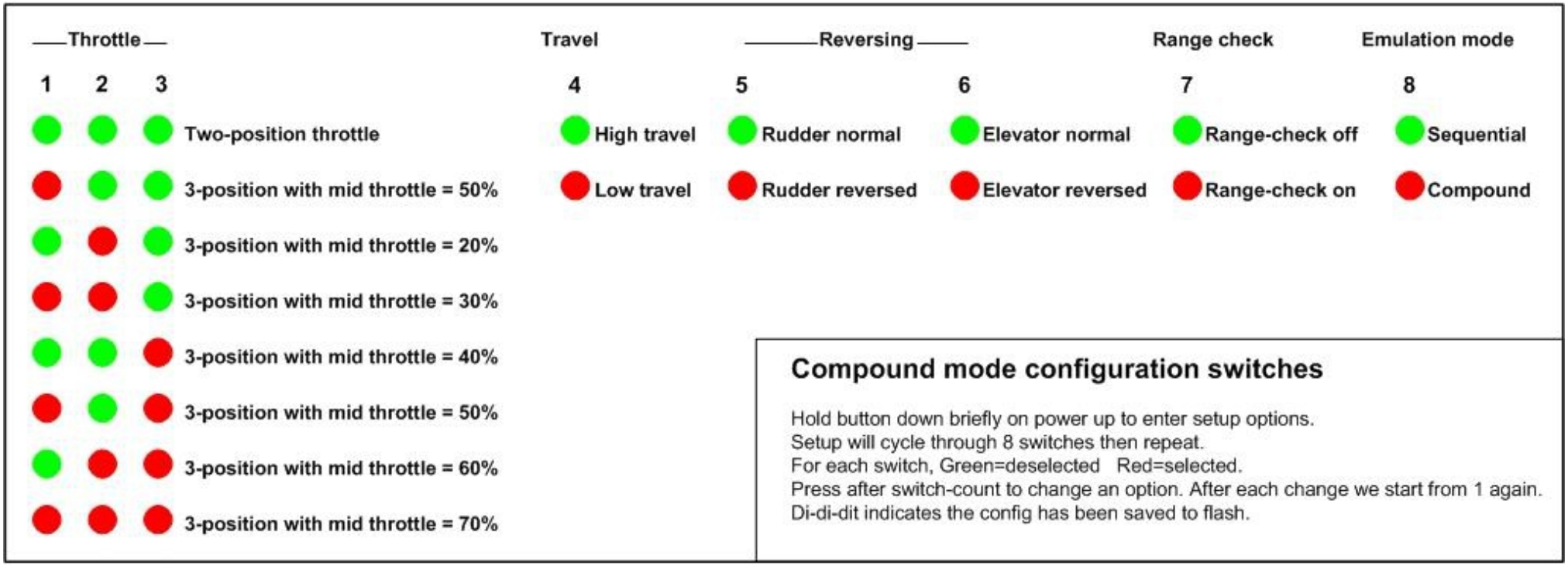
For example, to reverse the rudder servo, switch on whilst briefly holding the button down. Observe the current settings of the 8 switches, particularly the colour of the fifth flash. Wait for the 'di di dit' and the repetition of the 8 flashes. This time, count the pips and immediately after the

fifth pip & flash, press & release the button. The cycle will start again from switch one – again observe the settings of the 8 switches, and note that the colour of the fifth flash has changed. Wait for the “di-di-dit” sound as the new config is saved.
Switch off, switch on again and note that the rudder servo is reversed.

Another example, say you use sequential mode and therefore need a motor run timer to drive the ESC.
To enable a one-minute ESC run, set the first switch to red (selected). For a two minute run, set switch 1 back to green (deselected) and set switch 2 to red (selected). For a four minute run, clear switches 1 and 2 back to green and set switch 3 to red. Only select one timer option. If more than one time is selected, ie one minute and four minutes, only the shortest time will be used, ie one minute.
To disable the motor timer, set switches one, two and three to green (ie all deselected).
The motor timer settings are for sequential operation only. The timers can of course control the throttle servo on an IC powered model.
In compound mode switches 1,2&3 are used to select two position (on-off) or three position (half-full-half-off) throttle with a selectable mid-throttle point.

The eight switched options are numbered as follows. Note that switches 1, 2 & 3 serve two purposes depending on the emulation chosen:





Always allow the display to run through the full cycle of eight flashes to visually confirm your options, wait for the ‘di di dit’ to confirm that the settings are saved, then switch off. The saved settings become active when you next switch on – they only need to be changed when you need to change the configuration. Once set, all except the motor-run timers (which are cleared for safety reasons) remain in force even when the tx is powered off for long periods.

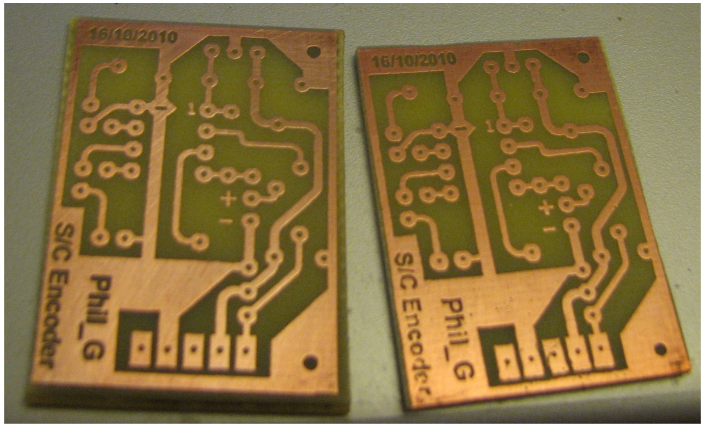
The encoder circuit diagram is very simple, having only a handful of components built on a custom PCB.
It can be easily built on stripboard, as were the prototypes.

Components:

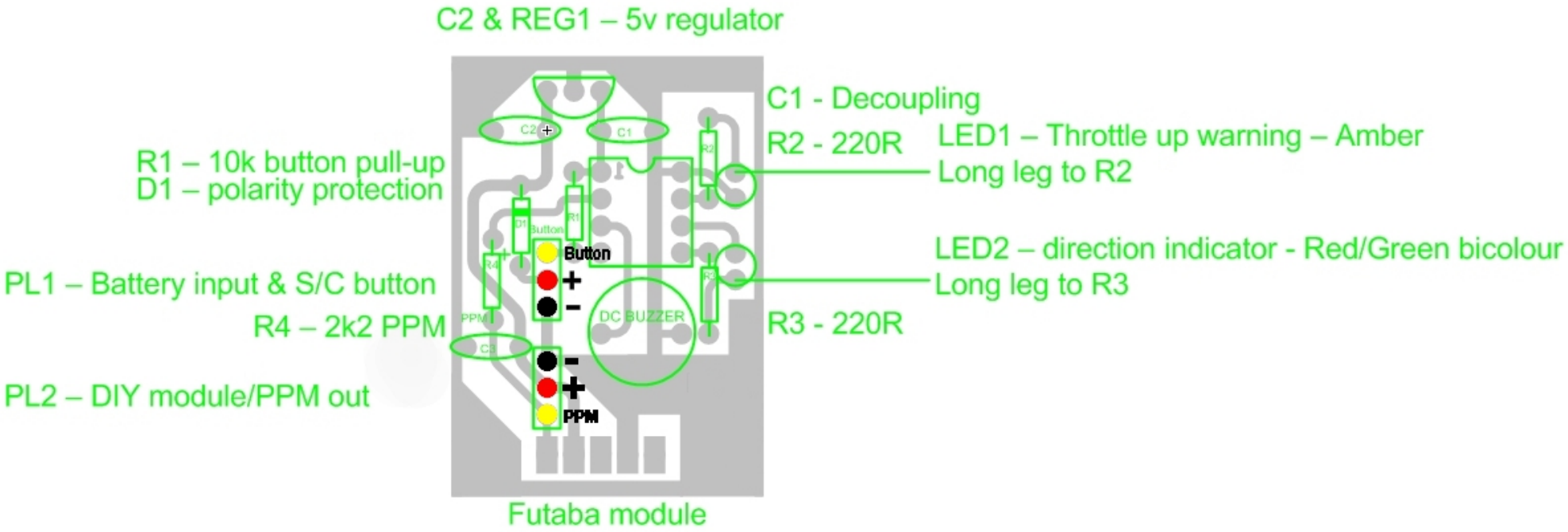
IC1	Pre-programmed PIC 12F675	(from Phil_G via philg@talk21.com & paypal, same address)
D1	1N4001 diode	
REG1	78L05	RS 810-295
C1	0.1uF mylar capacitor	RS 622-4397
C2	33-100uF Tantalum bead capacitor	
C3	0.1uF mylar capacitor	
R1	10k 1/8w resistor	
R2 & R3	220 ohm 1/8w resistors	
R4	BAT42 Schottky diode or 1k 1/8w resistor ***	
PL1 & 2	3 way servo header plugs	RS 251-8632
Buzzer	5v DC buzzer throttle-up warning	

LED1	Yellow ‘throttle-up’ ultrabright LED	RS 826-644
LED2	Red & Green Bicolour ‘direction’ ultrabright LED	RS 228-5641
Note:	LEDs can be optionally mounted in a header socket to permit extension - RS 488-1724	

*** R4 is a user selectable component used to match the PPM requirements of the chosen RF module or trainer socket input



PCB Layout (TOP VIEW from component side):



Construction:

Start by soldering the 8-pin IC socket, ensuring that it is oriented the right way – of course it will work either way, but if the socket is the wrong way round, it may lead you to fit the PIC integrated circuit the wrong way round too, which would be disastrous.
Fit and solder the two 220R resistors, coloured red, red, brown. These are the current-limiting resistors for the LEDs.
Fit and solder the 1K resistor, coloured brown, black, red. This one is the pull-up resistor for the single-channel button control.
Trim the excess leads using nail-clippers as you progress – with the exception of the Futaba module pins of course!

Check then fit the two diodes, ensuring that D1 (nearest the IC socket) is the 1N4001 and that R4 is the Schottky diode or resistor selected for your purpose from the table below. Note that both diodes have the banded end towards the regulator.
Solder in the plugs PL1 and PL2, and also the long-pin Futaba connector if it is required. Ensure that they are upright, fully home and seated on the PCB before soldering. Fit and solder the DC buzzer, noting that it is polarity sensitive, with its positive terminal nearest PL1 and PL2.
Fit and solder the 78L05 5-volt regulator, oriented with its flat side towards the edge of the board.

Solder in the capacitors, taking care not to overheat these components. C2 is the tantalum bead which is polarity critical, with its positive lead towards the regulator and its negative to the track surrounding the board. Polarized components usually have one lead longer than the other – the long one is positive. C1 and C3 can be either way round. The LEDs should be fitted last, for three reasons: they are very easily damaged by excessive heat, the length of the legs on which they stand should be chosen to suit the transmitter housing, and of course, being diodes, their orientation is critical. This is discussed further in the next section.

Finally, locate the notch in the PIC chip, and carefully align the PIC into its socket so that the notch is nearest the regulator. Make sure none of the legs have folded under or have become bent or trapped, then push it securely home. The encoder circuit board is now complete, and the wiring of the front-panel of the transmitter can be completed, following the diagram below.

Standard servo leads are used to connect the front panel to the encoder via PL1, whilst the output appears at both PL2 and the Futaba connector, one of which will have been chosen to suit your application. A DIY module or a buddy output is connected via PL2 using a servos lead patched to the three DIY module connections. If a cased ‘Futaba-fitting’ module is employed, then PL2 is unused and the whole PCB piggy-backs onto the Futaba module. This makes for a much simpler combination with no soldering or manipulation of the 2.4g module itself, and is therefore the preferred option. Note that a JR module cannot be fitted directly, and since it would require a different layout and a PPM voltage level shifter, it is not a supported option. The JR version of the encoder can however drive the buddyport of a JR/Spektrum transmitter.

It is vital that the plugs PL1 and PL2 (if used) are oriented and connected with the right polarity – even a brief mistake will destroy either or both the PIC and the RF module. Once fitted these plugs remain in place so it is worth taking particular care over these connections.

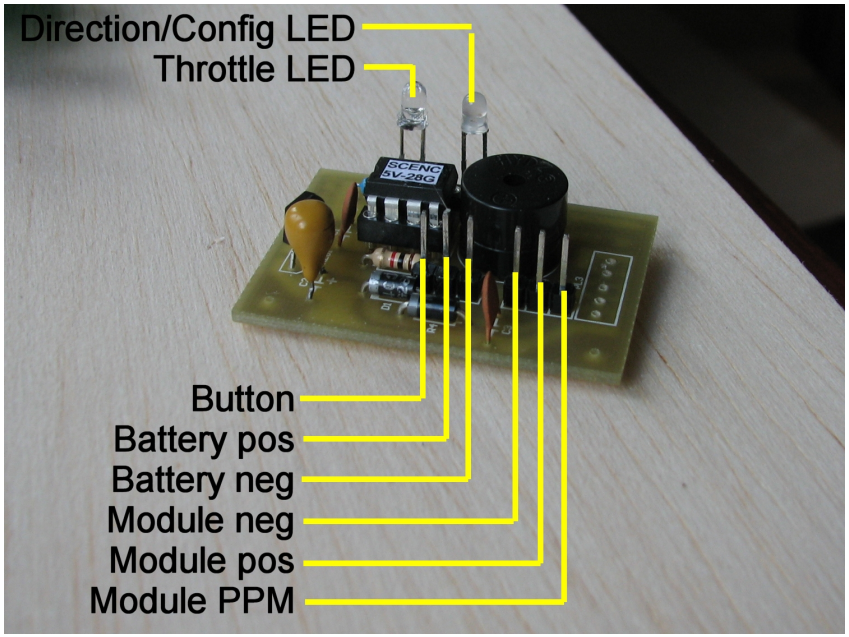
Bi-colour LED orientation:

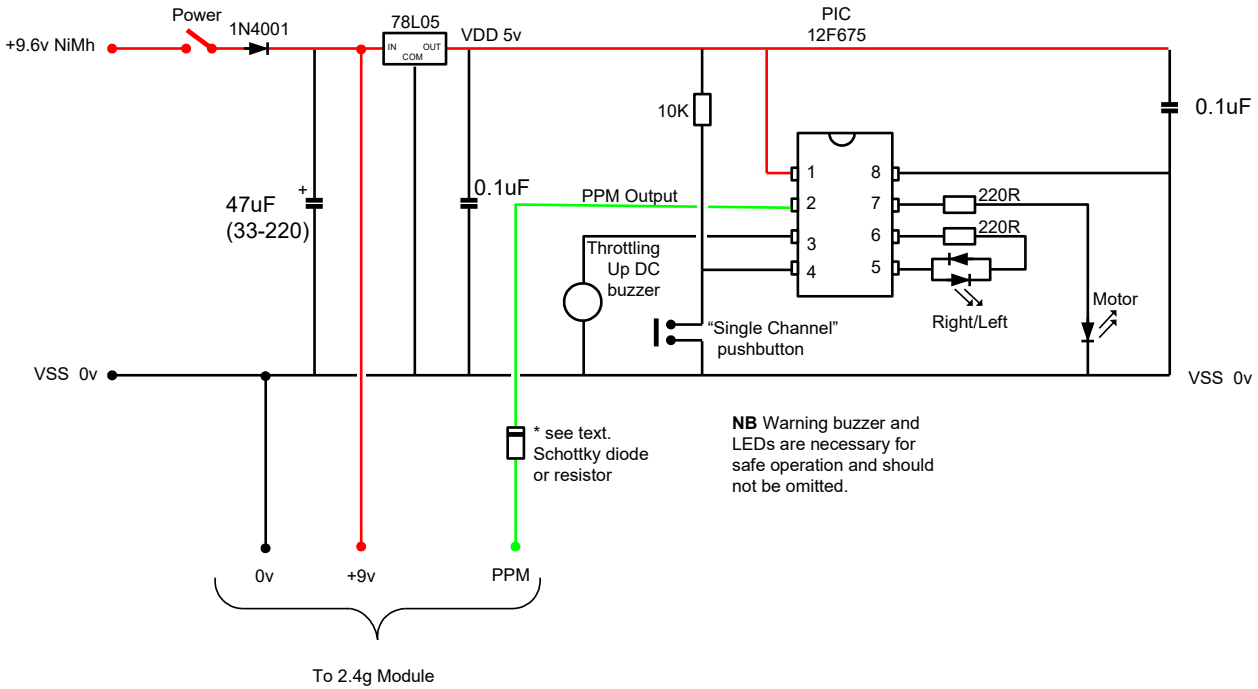
It is very important that the LED is correctly oriented as red/green is used in ‘configuration’ mode.
It is equally important that the LED colour matches the command direction, especially if the S/C transmitter is used with more than one model.
The shorter legs should be nearest to each other and the longer legs furthest apart.
To test, switch on the transmitter, then press the button down once and keep it held in.
Regardless of which emulation is active, the LED should be green during this button press.
If the LED colours are reversed, correct by removing the LED, turning 180 degrees and replacing.

In Compound mode, check that the bi-colour LED shows GREEN for one press and RED for two presses, ie Green for right, Red for left, as per convention. Don’t worry if the LED shows the correct colour but the rudder moves the wrong way – we can reverse the servo in ‘configuration’ mode. Of course it is also possible to match rudder operation to the LED colours, by choosing the appropriate side of the servo arm in the traditional way.

Go into the configuration and switch to sequential operation.
Check that the rudder movement still matches the LED colour – green for right, red for left.

When mounting the board inside the transmitter case we have found it best to mount the LEDs right next to the control button – in flight they can be very reassuring, especially on an electric model where the motor may be inaudible amongst other models.





"Single Channel" R/C emulation encoder for PIC 12F675. Generates 4 channels of PPM with a 20ms framerate and drives a standard 2.4g RF module, eg Corona, Frsky, etc.

Fuses - WDT enabled, code protect, Power on reset enabled, 4Mhz intRC osc.

Connections:
Throttle LED on GPIO0
Red/Green direction indicator on GPIO1&2
S/C pushbutton input on GPIO3
Throttle-up DC buzzer on GPIO4
PPM output on GPIO5

Channel assignments:
Ch1=rudder, sequential or compound (Elmic Conquest or Commander/Compact)

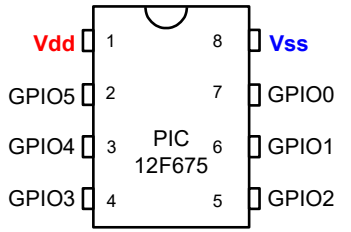
Ch2=optional kick-up elevator (Elmic Compact)

Ch3=optional quick-blip two-position throttle (Elmic Corporal)

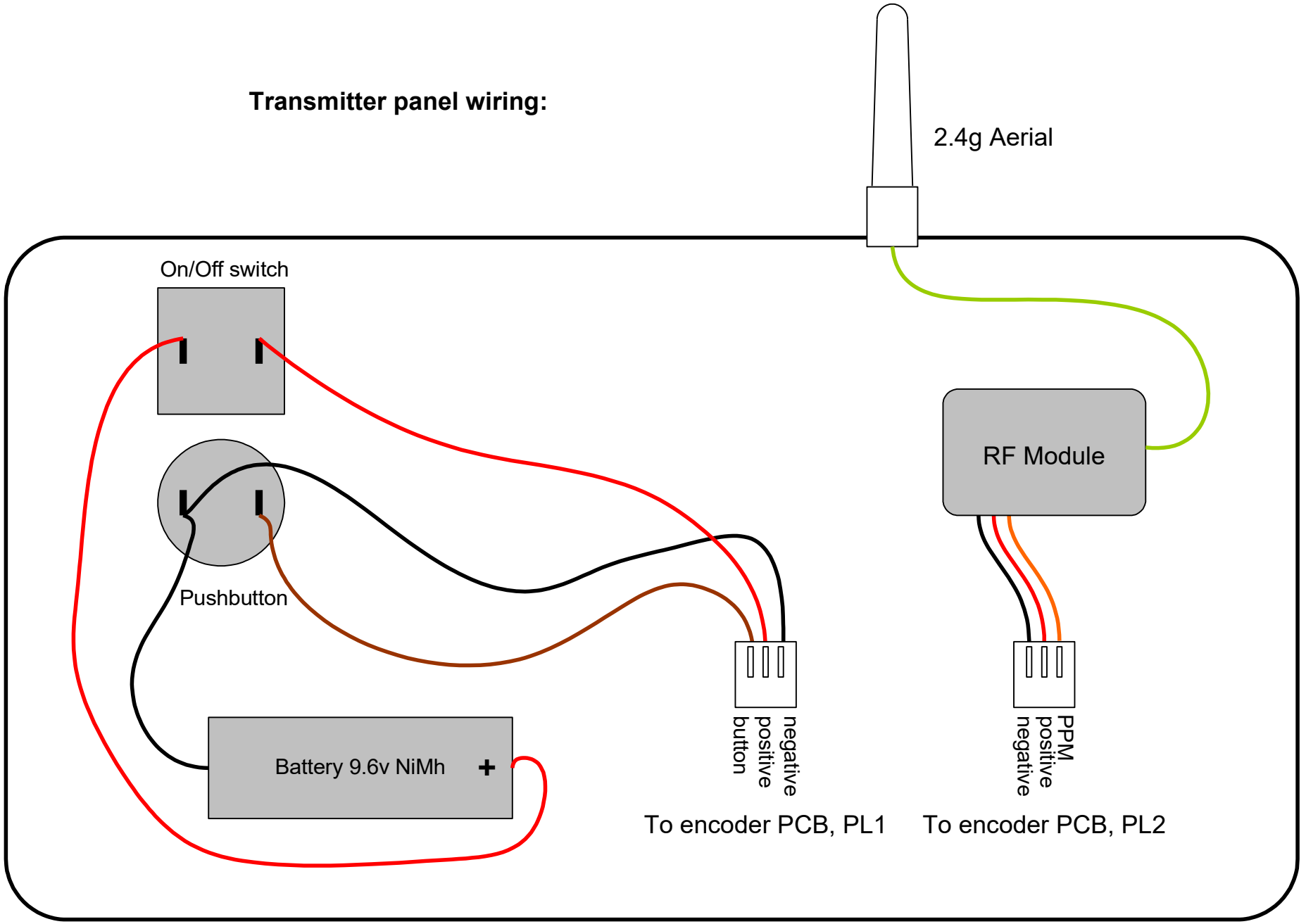
Ch4=in seq mode follows the button press for real escapement drive via a plug-in channel switch on the rx, 1ms=off, 2ms=on

Config options:
In sequential mode: ramped throttle up after 15 seconds with buzzer & LED warnings
One, two or four minute motor run timer.
Servo reversing.
Range test mode.

Author	Phil Green
Email	philg@talk21.com
Date	06/07/2010
Revision	Rev 2.0



Transmitter panel wiring:

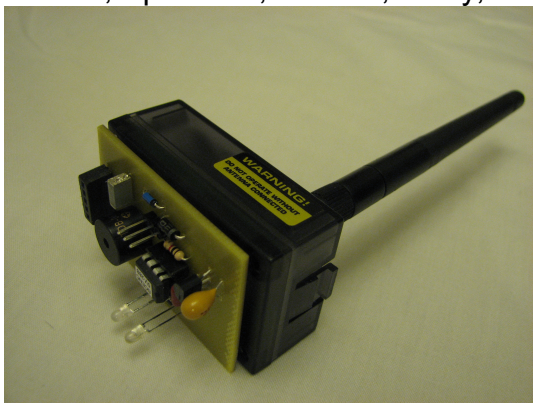


Until recently it was generally accepted that most 2.4g DIY or ‘hack’ modules would accept any polarity of PPM and at logic or battery voltage.
It has recently come to light that we have to avoid overdriving the 3.3 volt PPM input to both Corona and Frsky modules, as they are very much less robust than had been implied by the manufacturers. To their credit, Frsky have acknowledged the situation and have changed the design of their DIY modules to incorporate a schottky diode.

The encoder accommodates this situation by providing a user-selected protection component (R4) in series with the PPM line.
This makes the unit very versatile as the PPM protection component can be chosen to suit the particular application:
If the encoder is solely used to drive a Futaba, Spektrum, Corona or Frsky 2.4g module, the ideal series component is a schottky diode.
If the encoder is to drive a trainer socket as a buddy-box, instead of or as well as a 2.4g module, then a 1k ohm resistor is a good compromise.
Directly driving a JR module is not within the spec and will need an external level shift to the JR ‘full battery voltage’ PPM stream.
Without going into too much detail, we have to do one of 3 things: limit the PPM level to 3.3 volts, limit the PPM current into the module, or prevent the encoder supplying any positive voltage to the PPM input. The table gives an indication of the choice of component:

Module	**R4 suggestion**
Frsky DIY pre-manufacturers PPM diode modification	BAT42 schottky diode
Frsky DIY with manufacturers PPM diode modification	wire strap
Corona DIY hack module	BAT42 schottky diode OR 1k resistor
Other hack or DIY module	1k resistor, select on test
Cased Futaba compatible module, Futaba, Spektrum, Frsky etc	BAT42 schottky diode
Trainer or Buddybox feed, to Futaba or JR/Spektrum master	1k resistor
JR-fit module	Not straightforward. JR PPM needs level shifter to battery V
Flight Simulator	1k resistor

The PCB features two paralleled output connectors – PL2 is the most convenient for DIY modules or buddy leads, taking a standard servo lead socket. It also has provision for a Futaba long-pin connector which allows the PCB to piggy-back directly onto any Futaba-compatible module, eg Futaba, Spektrum, Corona, Frsky, etc. In this case PL2 is unused:





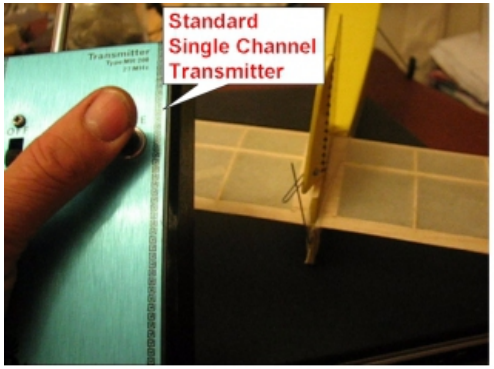
The Single Channel Emulation Encoder is available ready built & tested from Phil_G via email philg@talk21.com and the projects are discussed at length on http://www.singlechannel.co.uk, with supporting photographs and videos on http://www.youtube.com/user/philg2864/videos

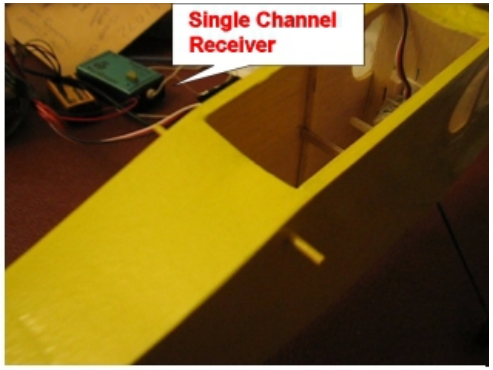
To support the Single Channel revival project, a collection of quite a few modules have been developed:

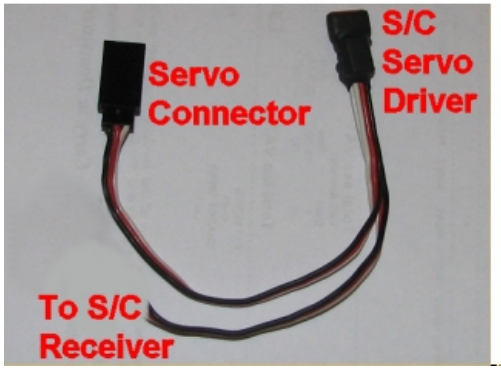
1) The encoder, as described above, emulating various escapements and driving a standard 2.4g RF module. A 35mhz module would also work but 2.4 is so convenient and enhances the contrast between ancient and modern.

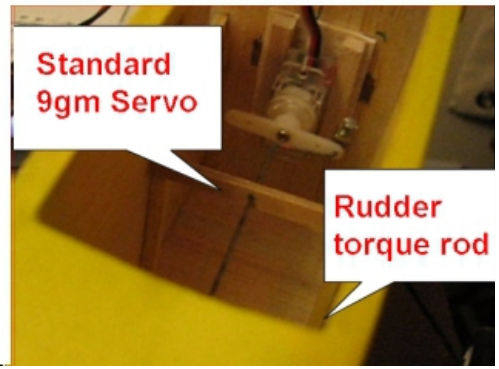
2) An optional ‘coder’ which does all the one-for right, two-for-left and quick-blip pulsing for you, like the old Macgregor Codamac.

3) An in-line module which converts any modern proportional servo into a sequential single-channel actuator. This would be driven by a genuine period relay or relayless single-channel receiver. Incorporates a ‘throw’ pot so that the travel can be adjusted (the rotary ‘bird-cage’ style linkage needs only a few degrees of movement either way). This module and a modern servo can be used to replace the Elmic Conquest. If you have an old single-channel set which works but is missing its actuator, this will fit the bill.



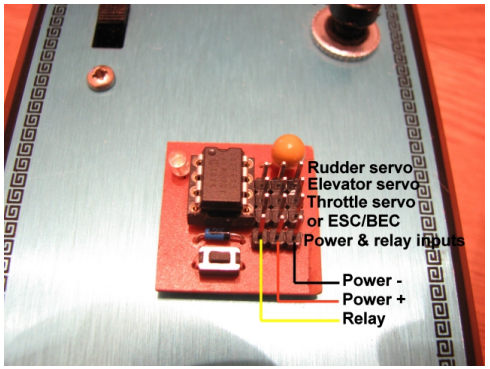






4) A similar in-line module which converts a modern proportional servo into a selective actuator such as the Elmic Commander or Compact. This would again be driven by a genuine period relay or relayless single-channel receiver.

5) A more comprehensive “Full House” compound actuator emulator which drives 3 modern servos, giving rudder, elevator and throttle (or ESC) from the relay or switcher output of a genuine single channel set. It incorporates a test button in parallel with the relay input.



6) A relayless driver switch which permits a real, period escapement coil to be driven by the channel 4 output of the 2.4g receiver, which is designed to follow the press-release operation of the transmitter button.

We would be delighted if others would give new-age Single Channel a try. You never know, it might catch on!

Finally please bear in mind that this isn't a commercial venture, I'm not a dealer, the modules are sold barely above cost to promote interest in vintage radio and model flying as it used to be. Everything is hand made on request, and in my leisure time which is limited. I will despatch orders as soon as I possibly can but I can't issue from stock, there isn't any!

Phil ☺ philg@talk21.com Phil_G on all the forums.

Encoder one-page, speed-read summary:

It has a fully featured software model of the escapement mechanism, and concentrates on an exact emulation of the escapement operation.
It features accurate emulation of either a sequential escapement, or compound escapements and actuators, with additional features relevant to each:-

In 'sequential' mode it alternately drives the rudder servo left and right as you would expect, but also has an automated throttle output to drive either a throttle servo or an electric speed controller. This is controlled by one of 3 optional timers giving a 1, 2 or 4 minute motor run. Prior to the motor starting, you must select a run timer, rather like refuelling an IC engine.
When the set is subsequently switched on, the motor is now primed, and an audible warning trill reminds you that the motor is ready for a run. Rudder commands will operate normally, allowing a last pre-flight check to be made – however if the rudder is held for 5 full seconds, a 5 second delay will start, with a more urgent warning. From the 5 second mark the motor will slowly ramp up to full speed, and the warning stops. After the selected runtime, the motor stops, just like an IC engine running out of fuel, and has to be 'refuelled' for the next flight by once again selecting a motor timer. The encoder has two LEDs - a bi-colour red/green used for left-right display and for configuration setting, and a yellow for motor run indication. In sequential mode, the red/green acts as a 'cheat' led showing which way the next rudder signal will go, with the yellow LED showing the throttle state - flashing during motor startup and steady during the motor run, extinguishing when the motor times out.

In 'compound' mode, the rudder servo mimics exactly the cyclic action of an escapement, the servo stopping at the left, right & kick-up pawl stops and the quick-blip operating the throttle, which can optionally be either two position (high, low) or 3 position (high, half, low, half etc).
In compound mode the red/green LED shows the currently selected direction, ie off for neutral, red for left, green for right, and mixes red & green to indicate when kick-up elevator is selected. The yellow motor LED is off for throttle off, flashing at half, and fully on for full throttle. Mid throttle of course only applies if the 3-way throttle has been selected. The nominally ‘half’ mid-throttle setting can be adjusted to a suitable ‘cruise speed’. A mid throttle, if the next quickblip will be ‘high throttle’ then the flashing is fast. If the next quickblip will give low throttle, then the flashing is slow.

In either sequential or compound there is an inactivity timer which sounds an alarm if the transmitter is left switched on and there has been no button press for 10 minutes. This is to prevent transmitters being accidentally left on, a mistake we’ve all made at least once!

It has a 'configuration' mode, selected by switching on with the button depressed, where for both sequential and compound modes, it has servo travel volume adjustment, individual servo reversing, and two range-check modes (slow sweep, or left-right clack).
The only control is the button, just like all the old single-channel sets, which also sets all the configuration options (rather like an audio/visual 8-way DIP switch) so everything is very easily configurable from the front panel using only the single-channel button. Each of the eight options can be either red or green, on or off. As the setup cycles through the eight options, pushing the button will 'flip' that option. Its very quick & easy.

Physically it is built on a neat printed circuit board which has two output options: it has a 3-pin connector suitable for DIY/hack modules such as Frsky and Corona, and it also has a standard Futaba module connector which allows it to piggy-back directly onto any Futaba-style module,, eg genuine Futaba, Spektrum for futaba, Corona for futaba, Frsky for futaba, etc. The output incorporates the schottky diode necessary for safe 3.3 volt module operation. The input is another 3 pin connector carrying switched battery pos, battery neg and the button input, which is simply grounded (to neg) when the button is pushed.
The whole thing runs happily from a PP3 drycell, or an NiMh pack of eight AAA cells.
The eight configuration options are saved in flash eeprom and so remain active regardless of switching the power on & off, with the exception of the motor timers which (for safety reasons) have to be set immediately before any flight, thats if they are being used of course, they are optional. The firmware is written entirely in assembler.

Despite its complexity the encoder is very simple to use, and configuration is a piece of cake from the front panel with no need to remove the cover to change anything.

I'm doing the encoders barely above cost as I have no wish to profit from a hobby, however any excess is used to fund additions to our growing collection of old-time radios which we present at R/C clubs with a talk and demos on request. Initially, whilst numbers permit, I’ll supply the boards ready made and tested. If demand makes this impractical, they will be issued as kits. This isn’t a commercial venture!

As is probably obvious, me and my clubmate Shaun Garrity are very enthusiastic about 'New Age' single channel equipment, it is so much fun without the problems expected of the old ‘period’ single-channel equipment. Shaun has an encyclopaedic recollection of commercial and homebrew radios, of old models and events and I would like to thank him publicly for his encouragement throughout the development of the project. I’d also like to thank Ashley Sheriff for converting my Visio pcb designs to Gerber files – an apparently trivial task that turned out to be anything but!
As for myself, following my late Dad's footsteps, I'm a class A radio ham and from an early age have always had an interest in all things electronic. I designed & built my own proportional sets back in the 70's, and have been actively involved in R/C electronics throughout.

Written for rom version `sc_5v_28M.asm` 06/12/2011

© Phil Green, 2011