

## Reeduino NRF - an FHSS 10-channel Reeds encoder with S/C emulation using the 'lockdown' NRF24L01

I've done a Reeds variation of the NRF24 FHSS project, emulating up to 10 channels. It suits any of the 'ebay boards' such as the DIY-More. I've deliberately worked to the spec of the existing Reeduino PPM encoder, and other than being 10 channels rather than 12, its functionally identical.

In use it will be totally familiar to anyone who's used to the Reeduino or the previous PIC encoder boards – it remains an accurate portrayal of a period reeds set which is easy to assemble and dead simple to use. It has all the same facilities as the Reeduino, variable servo-slow via a pot, half-speed throttle & aux channels, proper elevator trim via the trim toggle and 'cheating trims' (with pips) on all the other channels by simultaneously holding elevator trim and the required channel to trim, range-test sweep, etc – plus a few extras such as ATV (adjustable travel volume), and the Single-Channel compound escapement emulation mix. Reversing and elevon mix etc now happens in the receiver but transmitter reversing by power-on with the toggle thrown is still provided for use with the old FHSS receiver code. Because the NRF module uses D9 to D13, some pins have had to be moved compared to the PPM Reeduino.

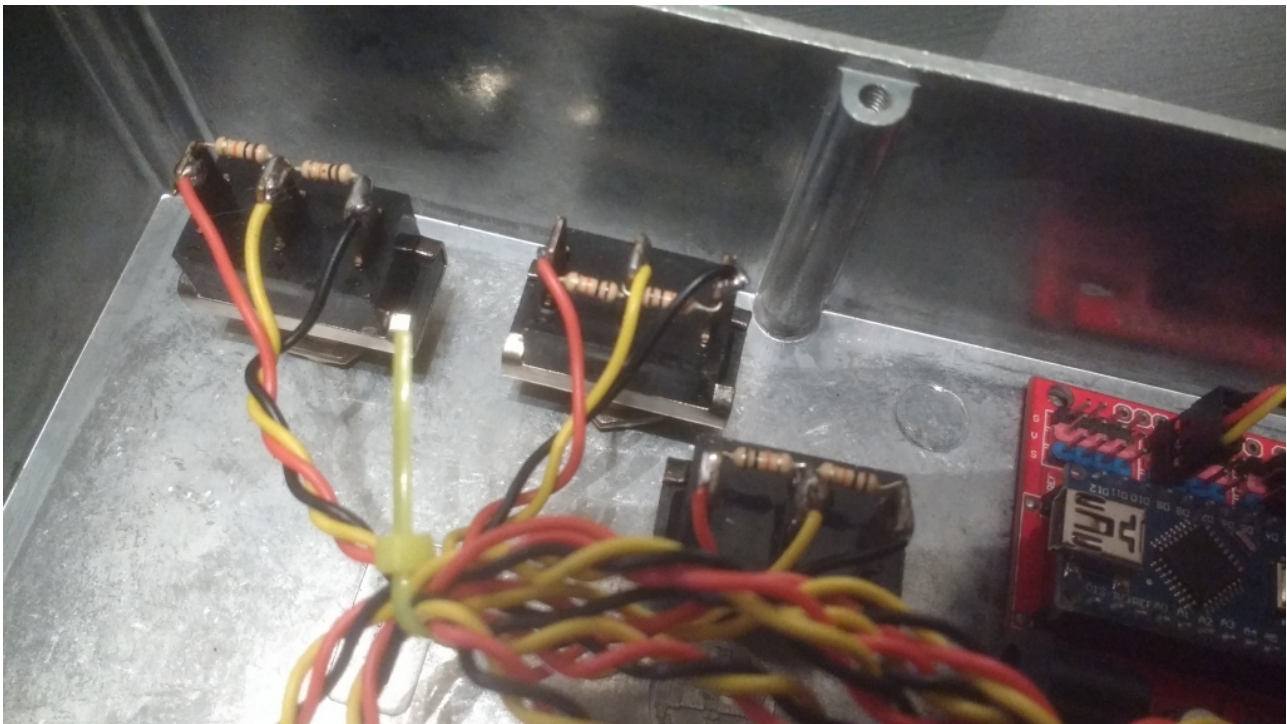
As before we have resistors across the toggles to enable three positions to be read via one analogue pin - and as before the actual analogue value has no direct bearing on the servo, its quantised to just 3 values so we can tell if the toggle is pushed one way, pushed the other way, or in the middle.

***This is not 'switched proportional'. Its a true Reeds emulation.***

There are crude reed implementations out there where resistors hold a propo channel at neutral and simply switch to either travel extreme - sometimes with capacitors for servo-slow - but thats not the case here. "High - mid - low" is just an economical method of reading the three possible switch positions via just one Arduino input pin.

### Assembly:

First prepare the toggle switches by soldering 10k resistors between either 'make' contact and the common. All the toggles are wired with servo plugs, pos and neg to the 'make' contacts, and signal to the common.



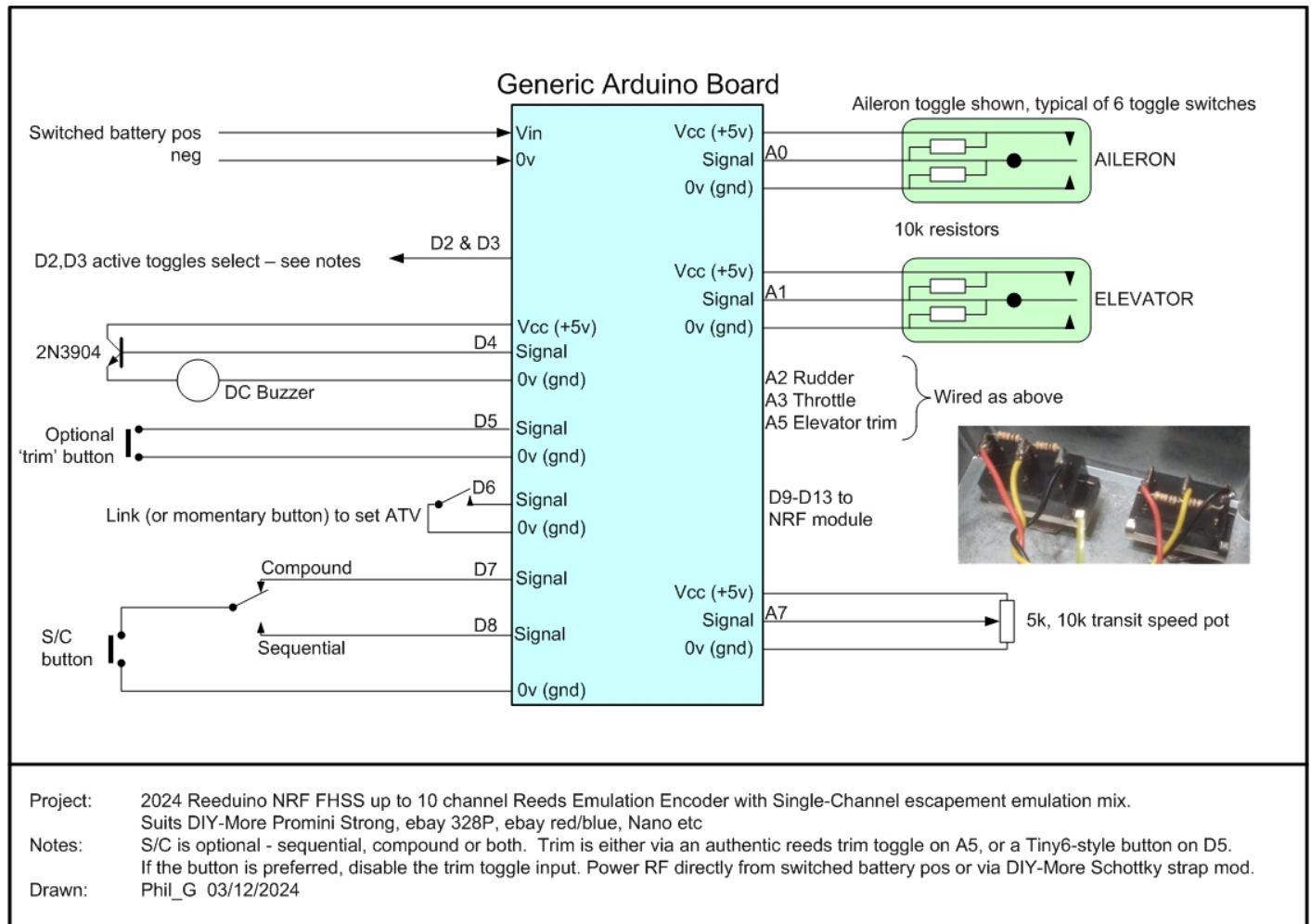
The single-channel button is a push-to-make. One button contact is ground, and the other is wired to D7 for compound escapement emulation. For Sequential, connect the button to D8.

The buzzer on D4 is best buffered using a transistor. This is most easily done on the buzzer itself – see addendum. Please don't omit the buzzer – its unobtrusive and trims would be awkward without it.

The transit-speed pot is not optional, it is essential to set and hold the required speed.

Whilst its an 'up to 10 channel reeds' encoder, some projects dont need the full 10 channels and you can't just leave unused toggles disconnected as the inputs are then indeterminate as they float around picking up extraneous voltages. So we need to enable the toggles you're using, and disable the ones we're not. This is done via option links on D2 & D3. Note that this setting *must* match your hardware – for example, if you ask me to supply a 10 channel encoder, you *must* wire all 5 toggle inputs.

A 6 channel set could be rudder/elevator/throttle for a powered model, or rudder/elevator/trim for a glider. Remember that in a RET model you're using the aileron channel for rudder. Think 'primary steering control' .



## Trim options:

Back in the day, elevator trim was a separate progressive servo which used two Reed channels.

The Reeduino NRF has two options for trim, both of which permit you to trim all channels, unlike a genuine reeds set where only elevator trim would be available.

- 1) a Tiny6-style trim-button which will be totally familiar to anyone presently using a Tiny6.
- 2) A conventional elevator-trim toggle

Without any code changes you have the choice of either the realistic reeds trim toggle, exactly like a period reeds set, OR the Tiny6 style trim button, whichever you prefer. (actually you could have both but its unnecessary! ...)

The trim button is a hardware option and can be omitted if you prefer the authentic up/down trim toggle, but it does simplify a basic Rudder/Elevator/Throttle 6-channel reed set as you no longer need an extra toggle for the elevator (and other channels) trims.

If you're building a full-house 10-channel set, for authenticity I'd suggest fitting all the toggles.

If you dont fit the trim button, its the familiar full-house reeds layout. Elevator trim works as you would expect, just like a period reeds set, and using a simultaneous combination of trim and other channels, they can be trimmed too.

If however you're building a simple 6 channel RET set, I'd use the button rather than have another toggle switch which would spoil the appearance of a 6-channel set, as traditionally they would have only three toggles.

To trim using the button, simply hold it in and blip the controls in the direction you need trim. Release when done.

The trim button can be quite discreet, as on my own RCS Inter-6: (between the RCS logo and the 'Inter 6' decal)



### Transit-speed pot:

Typical period reed servos such as the Bonner Duramite had a transit speed of slightly over 1 second lock to lock. Climax ServoMites were a little faster, as were Graupner Bellamatics.

The technique for flying reeds smoothly is to pulse for gentle control, and servos which were too fast did not give the desired proportional effect.

With gentle pulsing, a Duramite would almost hover at mid travel, giving a gentle turn.

So whereas for conventional proportional flying, fast transit servos are desirable, for reeds we need a slower transit to enable the servo to average out the pulsing of the toggle. This pot adjusts the servo transit speed.

For ease of throttle setting, the throttle moves at half the selected transit speed.

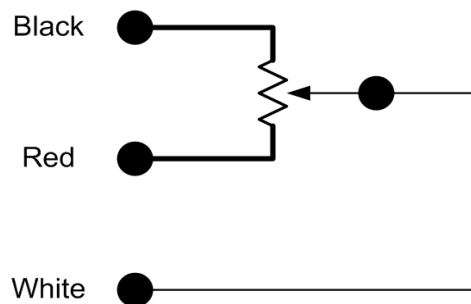
The speed takes account of ATV – a servo with half as far to travel would otherwise get there in half the time!

The transit-speed pot is a potential divider so its value isn't critical, 5k, 10k, 20k will be fine. Transit speed isn't a feature you would normally change in flight, so the pot can be mounted internally and set to personal preference – it could even be a preset.

The pot is wired with 5v and ground across the two outer tags and the wiper (centre tag) via the signal wire to A7:



Servo-slow Transit-Speed Pot



## ATV Adjustable Travel Volume

Adjustable Travel Volume or ATV as Hitec call it, its a settable end-point for each servo.

Its not variable 'in flight' like rates on a propo set - thats not what its for - its a configuration setting for use on the ground when setting up a new model, to allow you to pre-set the maximum servo travel & hence control-surface movements on each channel. ATV also affects the single-channel emulation throws.

To change ATV, you switch on, check everything is working, then stick a Spekky bind-plug into D6 or press the ATV button if you've provided one. This grounds the D6 signal wire. It beeps 'A' for ATV.

Then, two things happen.

First, all channels become progressive, ie they do not self-neutralise but stay where you left them like the way the throttle works, and second - regardless of the pot setting the servo transit speed is temporarily set to its slowest setting.

This allows you to slowly nudge any or all of the servos to whatever extreme limit of movement you require. Say you want 1/4" or aileron movement, you nudge the aileron toggle which is now progressive, left & right until the aileron is up or down by exactly 1/4". Same for elevator, rudder, throttle & aux.

Say for example you have an IC engine and the throttle arm has only one hole, and the servo also has only one hole. Without ATV the servo might hit the carb stop and stall. With ATV you can set the end point so the carb closes perfectly without doing any mechanical adjustments.

These adjustments can be nudged, twiddled & repeated as long as you like until all the throws are where you want them.

Then, when you're happy, you pull the link off D6 or release the ATV button. It saves the settings to flash, beeps 'V' for 'end of ATV', the servos neutralise and everything reverts to normal operation using your new servo throws. Any toggles you dont touch will retain their previous ATV setting. Any toggles that you try to set below 20% throw will also revert back to their previous ATV setting.

Thereafter, when you switch on, and it reads your reversing and trim settings from flash, it will also read the ATV settings you just made.

The saved servo-throw limits and are kept until you want to make changes by doing the 'link on D6' or ATV button thing again.

Servo transit speed is proportional to the set ATV for each channel, this means end-to-end transit time is the same regardless of how far each servo moves under its ATV. The reason this is necessary is because with a constant speed setting, a servo that had half as far to move would get there twice as fast.

In practise the servo transit pot works exactly as before although strictly it controls transit-time over all the different channel ATV's, rather than rotational speed.

I've limited ATV to 20% minimum so you cant accidentally set 'no travel' which would be a bit exciting.

As before the flash reset command (power on with up-trim held) clears all reversing and centralises all the trims and resets all the ATV's to 100%, which is 1100 to 1900uS, plus or minus 100uS total trim.  
The ATV range is from 20% to 112.5%

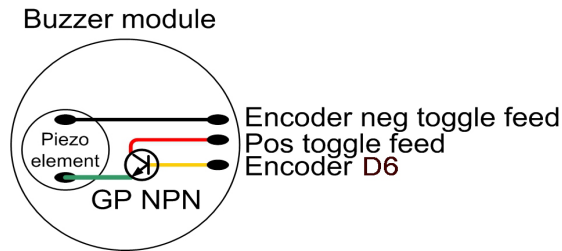
Throttle is treated exactly like any other channel, with reversing & ATV, but as before is slowed by half with respect to the other channels to make throttle selection easier.

Because an ESC 'trains' itself to the throttle channel, there should never be a need to change throttle ATV on an electric, and since almost all ESCS are 1mS low, 2mS high, there should never be any need to reverse an electric throttle, but in any case please, please remove electric props before doing any throttle reversing or ATV settings!

If you dont want ATV, just ignore any reference to it and the set will happily default to 100% travel on all channels.

### Buzzer:

The buzzer is used for trim-pips, mode selection and for the inactivity warning. It is normally a two-wire device but in this case a transistor has been added to buffer the buzzer drive. The whole thing as a unit is supplied pre-wired as follows and can be considered a 3-wire device as per the previous diagrams:

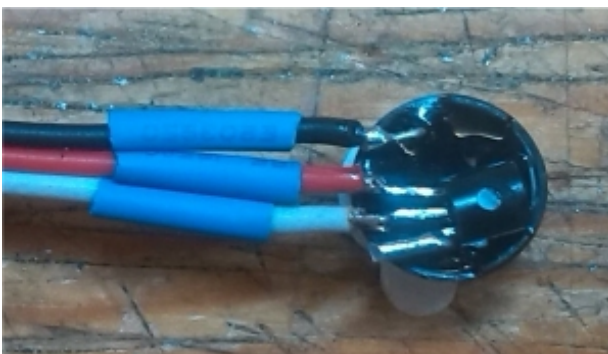


The buzzer assembly (buzzer and transistor buffer) is supplied ready made & tested, but here's how I assemble them, you may come up with something better:

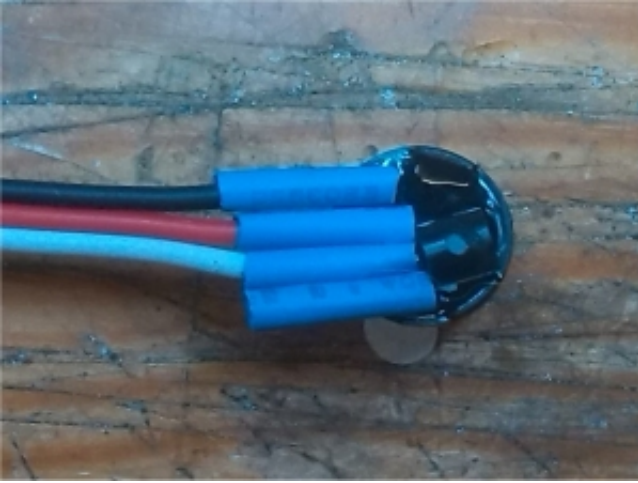
First, with the buzzer positive at the bottom, the leads are kept parallel whilst being bent to the left along the surface of the buzzer:



Next the 2N3904 transistor is laid flatside-down with its emitter touching the buzzer positive, then soldered: The leads are cropped to the edge of the buzzer and sleeved servo cable soldered as follows: black to the buzzer neg, red to the collector, white to the base. There's no wire to the buzzer /transistor junction.



Add a similar length of sleeving over the emitter/buzzer negative:



Next heat-shrink the sleeving, making sure it remains pushed up against the transistor body. Then add an overall sleeve covering all the connections:

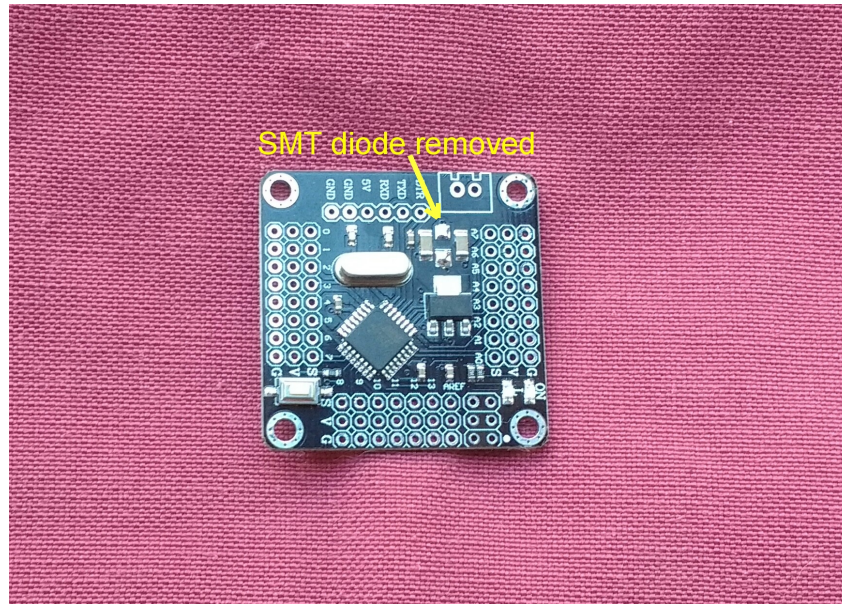


When connecting the assembled buzzer module, red is positive 5v (from the toggle switches positive feed) black is negative (again from the toggle feeds) and white goes to D4 on the encoder.

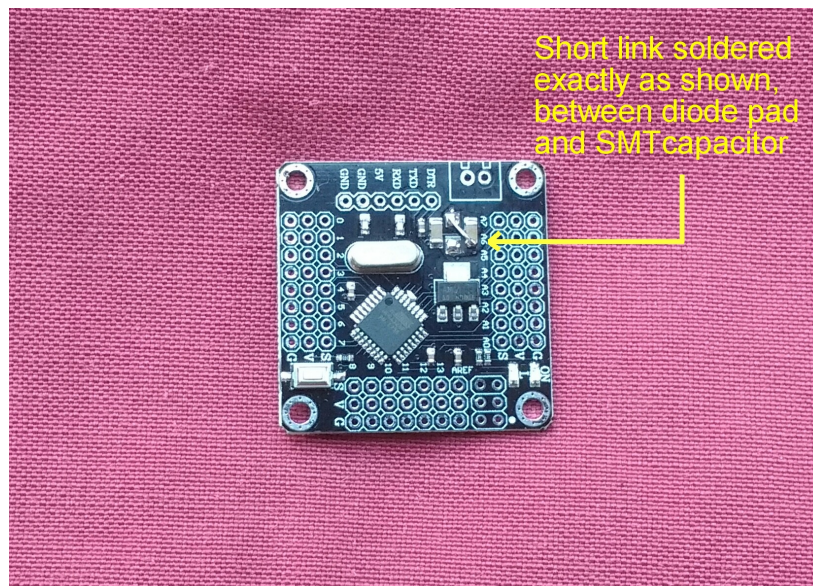
### The 2018 DiyMore 328P board:

This board was introduced in 2018 and is widely available via Amazon and Ebay, this one is easily distinguished by having the 'servo headers' on three sides, and its 'Pro-Mini-Strong' label. Functionally its identical to previous boards, with some advantages – it has a proper 16mhz crystal rather than a resonator, so timing should be more accurate, and it has a separate power in connector. It also has a 6-pin ICSP port. It needs a small modification to provide battery and neg for the NRF module, which is most easily done before populating the board with the headers.

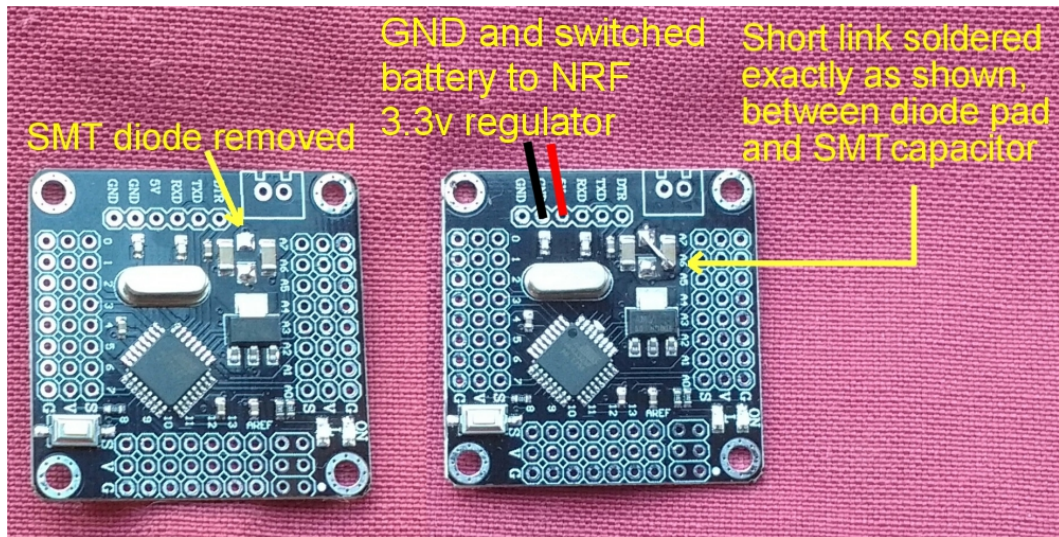
The mod is to remove an SMT schottky diode, then to bridge one of the vacated schottky pads to a nearby SMT capacitor:



Having isolated the 5v input header, we can now use it to provide battery voltage to the NRF module. This is done with a short diagonal strap from the upper diode pad to the lower end of the SMT capacitor:

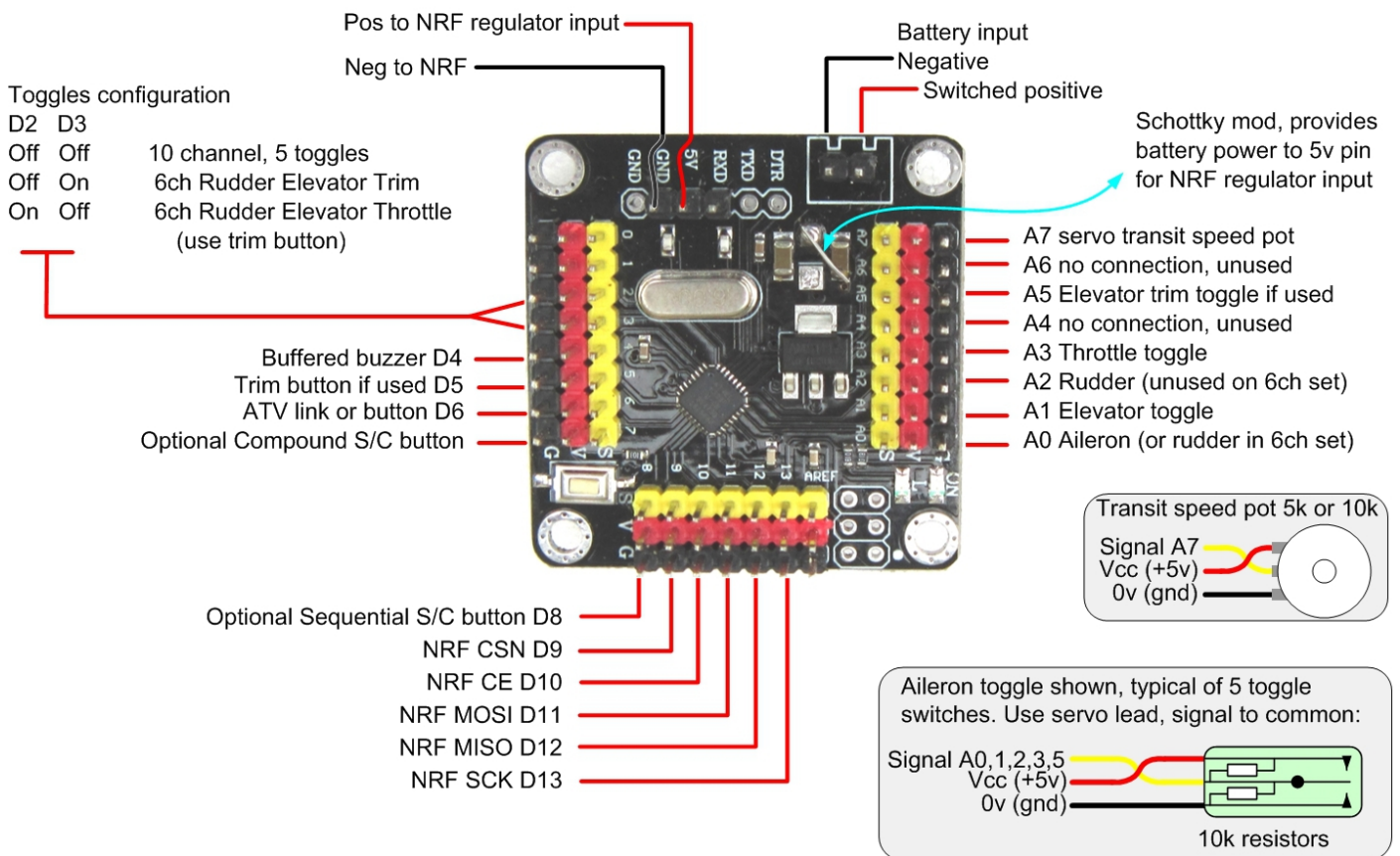


Heres a closer look, before the header pins are fitted - we dont have PPM with the NRF FHSS system, we just use GND and the pin marked 5v which after the mod carries switched battery voltage.  
This is just a convenient way to power the NRF from the PCB, but its not essential and if preferred power can be taken directly from the main on/off power switch:



# NRF FHSS Reeduino 2024

Phil\_G



The ATV, Trim and Single-Channel, buttons are push-to make (normally open 'NO')

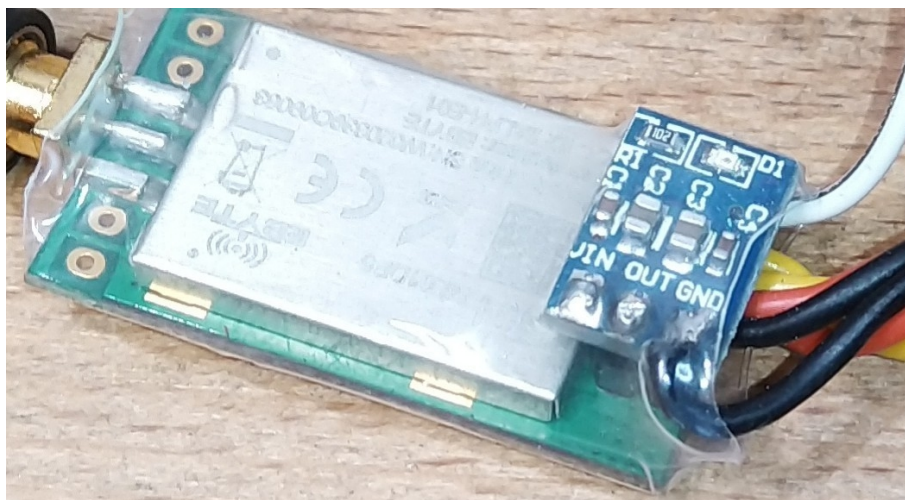
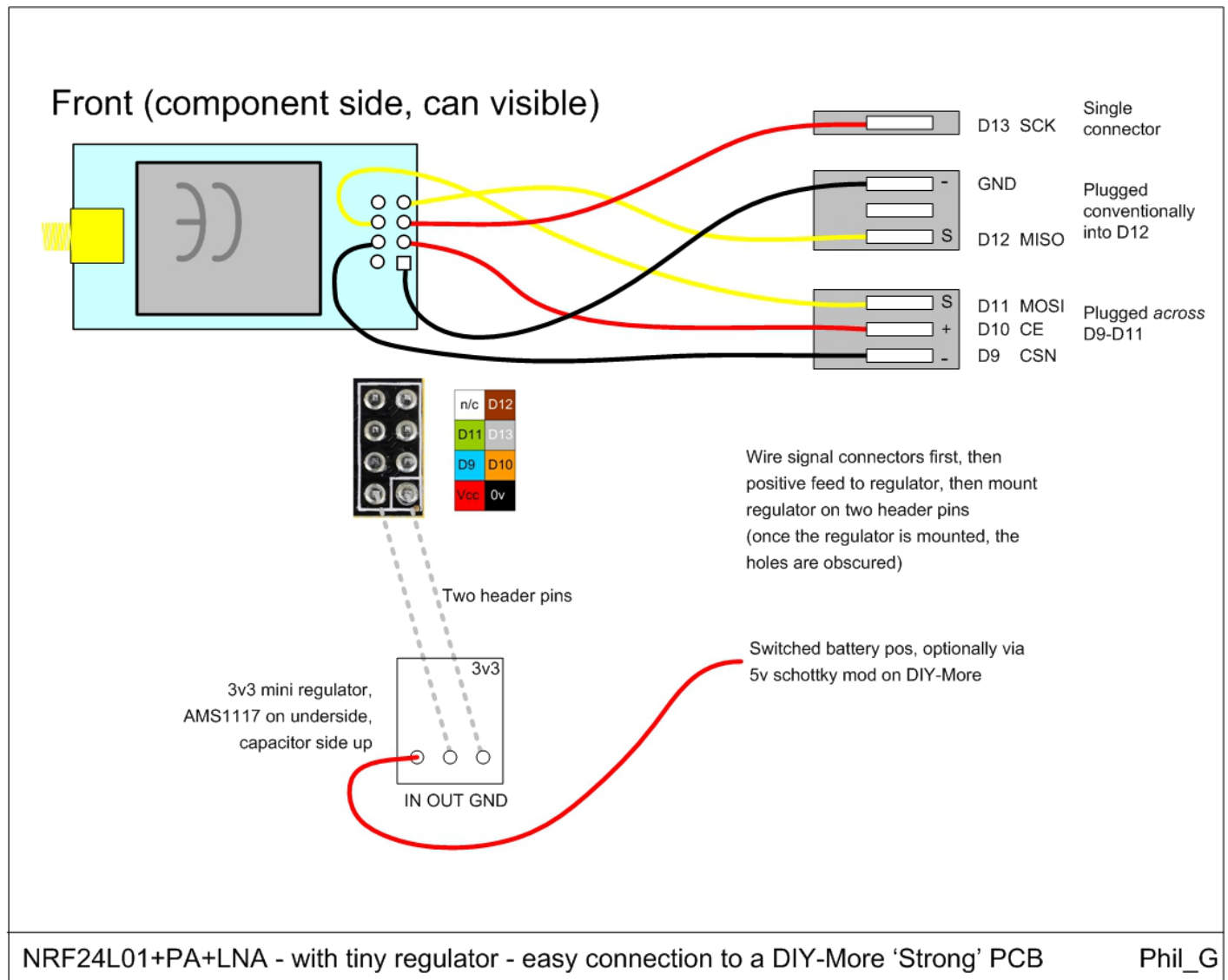
Toggles are all push to make. Be careful not to short the toggle 5v and ground connections.

The transit speed pot value isn't critical, 5k, 10k, 20k is ok.

Ensure that the links on D2 & D3 match your hardware.

The NRF connections are exactly the same as for the Propo, S/C, 1+1 and 2+1 projects.  
I'd recommend doing it in this order, since once the 3.3 volt regulator is mounted, the signal wires are obscured.

Remove the NRF header pins, then solder the signal wires from the servo plugs,  
Mount the regulator on two header pins (pos and neg) so it just clears the NRF board, capacitor side up  
Add the ground wire to GND and the pos battery input connecti on to the regulator IN.  
The pos battery feed can come directly from the main on/off switch or via the Schottky mod on the DIY-More PCB.



Further discussion on the 'Strong' DiyMore 328P board starts here:  
<http://mode-zero.uk/viewtopic.php?f=42&t=241>

## Reeduino-NRF Facilities summary:

### 2.4Ghz Frequency Hopping

Up to "10 channel Reed" emulation. Other configurations by D2/D3 selection links.

Optional full single-channel compound or sequential escapement emulation via a button,

Variable servo transit speed to match period servos such as the Bonner Duramite

Resolution is 1uS and is super smooth, even when the servo transit speed is set very slow.

Servo reversing by switching on with the required toggle thrown. Saved to flash. NOTE this is now redacted.

Trims on every channel, by simultaneous use of elevator trim and the required channel. Saved to flash.

Trim-pips with longer neutral-pip

Elevator trim lock to make trimming channels other than elevator much easier

Optional alternative 'Tiny-6 style' trim button for Rudder/Elevator/Throttle sets with no trim toggle

Adjustable Travel Volume (ATV) on every channel, range 20% to 112.5% travel. Default is 100%.

10 minute inactivity timer, sounds if no keys are thrown for 10 minutes. Any key resets the timer.

Range-check sweep mode, whereby the transmitter can be left whilst you walk away with the model

Master flash reset, reverts all reversing, ATV and trim settings back to neutral.

One character Morse identifiers are used to indicate the mode:

D for reset to Defaults, R for Reversed, A for set ATV, V for ATV end, S for Scan, K for OK

If the set is held in 'reset' for 10 seconds then a full ID is sounded

## Channel Assignments, Mixers & Reversing

From 2024 receiver-based memory is used in place of the previous 'transmitter-based' reversing & mixing methods.

Its identical in effect but its a different sequence.

With the transmitter powered on and operating normally, hold the transmitter toggle over whilst powering on the receiver to reverse a function. The setting is retained in that one particular receiver's flash, and of course only affects that receiver. Nothing changes in the transmitter.

The elevon mixer is also in the receiver and is configured at programming time by un-commenting the mixer section, to suit the model. This makes conventional transmitter-based "model memories" redundant.

*(Transmitter reversing is actually retained, in case a toggle is reverse-wired, but is deprecated with the new FHSS receiver code)*

The development is discussed in this thread:

<https://mode-zero.uk/viewtopic.php?p=15783#p15783>

Please post your builds on there!

Cheers

Phil\_G

Last update: 03/12/2024