

*Writing a deviation Tx model*

## ***Table of Contents***

### Introduction

[What's special about deviationTx?](#)

[What's in this document](#)

[Intended audience for this document](#)

[Giving credit where credit is due](#)

### Overview of the parts

#### Outputs

[Transmitter Channels](#)

[Virtual Channels](#)

#### Inputs

[Sticks](#)

[Knobs and Sliders](#)

[Trims](#)

[Channels](#)

[Switches](#)

[Physical switches](#)

[Channels as switches](#)

[Button switches](#)

[PPM](#)

### Overview of mixes

[A mixer](#)

[A mix](#)

### Reading .ini files

[Model description](#)

[First sample mix](#)

[Curved sample mixer](#)

[The last mix setting](#)

### Basic model setup

[Simple mixers](#)

[Expo&DR mixers](#)

[Setting up flight modes](#)

[Channel configuration](#)

[Throttle hold](#)

[Other channel configuration](#)

### Other niceties

[Sticky throttle hold](#)

[A simple mix](#)

[Timers](#)

[Channel monitor](#)

[Toggle icons](#)

### Finding RTF & BNF channel mappings

[Experimenting to find the channels](#)

[Throttle](#)

[Other control surfaces](#)

## Writing a deviation Tx model

[Rearranging channels](#)

[Other controls](#)

[WLtoys channel hacks.](#)

[Reversing a channel](#)

[Throttle safety](#)

[Advanced model setup](#)

[A panic switch](#)

[Stagility mode](#)

[Dialable stagility mode](#)

[Initialize the flight controller](#)

[Advanced Flight Mode Settings](#)

[Tricks with trim buttons](#)

[A momentary button](#)

[A toggle button](#)

[A switch](#)

[Adding an analog input](#)

[A three position switch](#)

## ***Introduction***

Let's start with a bit of non-technical material.

### **What's special about deviationTx?**

deviationTx's standard mixer mode is nothing special. If anything, it seems a bit underpowered when compared to proprietary aircraft transmitter firmware. The advanced mixer mode is much harder to set up, because you have to know how the desired behavior of the aircraft controls is expressed as channel output values, and then how to use the deviationTx mixers to create those outputs. However, it also provides direct access to the internal workings of mixes in deviationTx, making it incredibly powerful.

### **What's in this document**

After this introductory material and an overview of what's used in a deviationTx model config, this document will walk you through a complete setup for a modern multirotor aircraft using the advanced mixer. It starts with the basic four channel configuration plus a flight mode selection channel, covering all the features that any computerized radio should have. After a detour explaining how to figure out the channel bindings when they aren't known, it dives into using the advanced features of deviationTx. Along the way, it will cover how to set up channel configurations, the main page configuration, automatically changing output values, and anything else I know how to do that might be useful. As a lagniappe, there's a section covering all the things you can do with trim buttons besides use them as trims.

### **Intended audience for this document**

This document is intended for experienced RC pilots. In truth, at this time, I don't believe that deviationTx is suitable for beginning pilots, though progress is being made in that area all the time. Because of this intended audience, I will not be covering the standard mixer of deviationTx.

### **Giving credit where credit is due**

The deviationTx community is also special. They freely share their model configurations and mixing tricks in the forums. The information in this document comes almost exclusively from those forums. Some small bits of it are originally mine, but the bulk of it comes from the forums, though I've changed it to meet my needs and put it into my own words. While I can remember who I got some of it from, I can't remember that for everything, or in some cases even if that it wasn't mine. So rather than probably leave out someone who deserves a mention, I'm going to credit the community at large. My thanks and gratitude go out to everyone in that great group.

If anything here is wrong, it's because I screwed things up.

## ***Overview of the parts***

This is an introduction to the various moving - or at least changing - things that show up in a deviationTx model. This information is in the deviationTx manual, but I've included my rewrite here for your convenience.

Throughout this document, names that are shown as they appear in the firmware are shown in a `FIXED` font. Names - or parts of name - that are variable will be in *ITALIC*. Most of them are upper case, because that's the way they appear on your transmitter screen.

The values of inputs and outputs usually run from 100% to -100%, meaning going from on to off, or from full value in one direction to full value in the other. Many can be set to take on values up to 150 instead of 100, and the outputs can have their maximum and minimum values adjusted.

## **Outputs**

The end goal of any mixer program is to get a specific behavior for the outputs of the transmitter, which are the inputs of the model you are controlling. So let's go over those first.

## ***Transmitter Channels***

The transmitter output channels are the values actually sent to by the transmitter. The number usually matches the number of channels on the receiver, and can vary from as few as 1 to as many as 12.

## ***Virtual Channels***

One of deviationTx's uncommon features is that it also has 10 virtual channels. These can be mixed just like a transmitter channel, only they aren't sent to the receiver. They are useful for creating intermediate values for use in other mixers. You'll see them used in the model config being constructed.

## **Inputs**

The inputs include all the controls on the transmitter, as well as some internal values. I'm not going to describe the physical controls in detail, as they vary between transmitter models, and you should already be familiar with them.

One thing all inputs have in common is that they can be inverted. Simply select the input rather than using the `Left` and `Right` buttons to change it, and a `!` will be prepended to the name to indicate that it is inverted.

## ***Sticks***

The two control sticks that dominate an aircraft transmitter. They also have a pair of trims, each of which is two buttons beside or below the stick. The deviationTx labels them as `ELE` (elevator), `AIL` (aileron), `THR` (throttle) and `RUD` (rudder).

## *Knobs and Sliders*

Some of the transmitters also have analog knobs or sliders as inputs. They are labelled `AUX#`, where `#` is a digit from 2 to 7.

## *Trims*

The trims are `TRIMLLD`, where `LL` is the location: `LH` for Left Horizontal, `RV` for Right Vertical, etc, or `L` for the upper Left trim and `R` for the upper Right trim. The `D` is direction the trim works in, either `+` to increase the value or `-` to decrease it.

## *Channels*

Yes, these are the same channels that showed up in the Inputs section. You can use the output of any channel as the input of a mixer. These are actually the previous value of the channel, so you can use the old value as an input for a mixer, allowing you to create things like channels whose default behavior is to hold their old value rather than some fixed value, or to change the value at a regular rate.

In extreme cases, you can configure the model to have more output channels than you actually need and use the extra channels as virtual channels. If you then reconfigure the model to have fewer channels again, these extra output channels will keep their old behavior, but not be transmitted. Besides being inconvenient to work with, this behavior is undocumented and liable to change in the future. Stick with virtual channels unless you run out.

## *Switches*

There are a variety of devices that can be used as switches by deviation Tx. They normally have values of either 100 when on, or -100 when off.

### Physical switches

Like any modern transmitter, the Devo line has a collection of switches available. These have labels like `GEAR#`, `MIX#` and `FMOD#`. The digits indicate a position for the switch, being either 0 and 1 for a two-position switch, and adding 2 for a three-position switch. Each of these has a value of 100 when the switch is in the position named, and -100 otherwise.

This usage is very convenient, but it does mean that `FMOD0` and `!FMODE1` are the same thing, at least on transmitters with a two-position flight mode switch. If there is an `FMOD2` position, it will be -100 for `FMOD0` and 100 for `!FMOD1`.

### Channels as switches

All of the channels are also available as switches. When used as a switch, any value greater than 0 will be on and any value less than or equal to 0 will be off.

# Writing a deviationTx model

## Button switches

Any of the buttons on the transmitter, including those used for navigating the transmitter configuration, can be used as a virtual switch. This requires setting them up as trims for a virtual channel, and then using the virtual channel as a switch. The last section covers this.

## PPM

Finally, you can use the DSC (aka trainer) port on your transmitter for PPM input. This can be used for anything that can output PPM to add controls to your transmitter. Typical examples would be most computerized transmitters acting as buddy box slaves, or FPV goggles with motion tracking equipment. These have labels `PPM#`, for input channels 1 through 8.

## ***Overview of mixes***

And now an overview of mixes on deviationTx. Again, this information is in the deviationTx manual, but I've included my rewrite for your convenience.

Unlike proprietary firmware, where there is usually a limited number of mixes each of which can be attached to a specific channel, in deviationTx every channel has a mix associated with it. These are available in the `Mixer` section of the `Model` page of deviationTx. Each channel has two buttons and a label associated with it, though the label may be missing.

One button has the channel information as a label. The buttons on the first 5 output channels will be labelled `#-XXX`. The `#` is the channel number, and the `XXX` is the normal usage for that channel in the selected protocol. These are `THR`, `RUD`, `AIL`, `ELE` and `GEAR` (gear), with the order depending on the protocol. The other output channels are labelled `Ch#`. The virtual channels are labelled `Virt#`, in this case, the `#` is the virtual channel number.

The second buttons label tells you what kind of mix this channel has. It will be one of `None`, `Simple` or `Complex`. Those are the focus of this document. If the model is a helicopter, you will also see `Cyclic1`, `Cyclic2` and `Cyclic3`. These are normally only used for older collective pitch helicopters without a flight controller, meaning the channel mixing needs to be done in the transmitter. Since you can only select those but not adjust them, they won't be discussed.

Each mix that is not `None` is a sequence of one or more *mixers*. A *mixer* is similar to a mix in proprietary firmware, but much more flexible. Each mixer can have an input associated with it. If the first mixer is not `None` and has an input, that input's name will be the label for this channel on the transmitter `Mixer` page along with the two buttons.

## A mixer

A *mixer* takes the value of an input, modifies it in some way, then mixes it into the output of the previous mixer. Each mixer has at least four different things you can set. Most important is the `Src`, which is the input value for this mixer. It can use any of the things listed in the Inputs section above.

There are also a `Scale` and `Offset` values, which change the value of the mixer by taking a percentage of it, or moving it up or down by a fixed amount. A `Scale` of 100 (or 100%) doesn't change the value at all. A scale of 50% makes it  $\frac{1}{2}$  the value it would otherwise have. A scale of 120% will make it larger by  $\frac{1}{5}$  of the value. And so on. The `Offset` value is simply added to the value. These behaviors may be modified by the selected `Curve`, though.

The `Curve` is a function that is applied to the value of the `Src` to get the initial output value, before the `Scale` and `Offset` are applied to it. When the selection widget turns into a button, you can select that to open a page that lets you change the parameters of the curve. The available curves are:

- 1-1: This just copies the `Src`.
- Fixed: The source is ignored, and the `Scale` value is output.
- Min/Max: This is either `-100` or `100`. The parameter is the input value where the the output changes from `-100` to `100`.
- Zero/Max: This is like `Min/Max`, except the lower value is 0, not `-100`.
- >0: Like `Zero/Max`, except that the `Max` part of the output copies the input instead of being `100`.
- <0: Like >0, except the `Min` part of the curve copies the input and the `Max` part is 0.
- ABSVAL: When the parameter is zero, the output is the absolute value of the `Src`. Moving the parameter moves the point of this V-shaped curve..
- EXPO: This applies an exponential to the curve. The parameters let you set the amount of expo to be applied above and below zero. While the settings are in an unusual place, the idea is the same as it is in proprietary firmware.
- Deadband: This provides a dead band near stick 0. The parameters let you control the size of the dead band above and below zero. Note that it always outputs an absolute value, so you may want to use a `Mux` of `Mult`.
- # Point: This lets you specify 3, 5, 7, 9, 11 or 13 points the output curve will go through. The parameters let you specify each point and enable or disable smoothing.



### A mix

A *mix* is a sequence of mixers. Each is evaluated in order, mixing the value of its `Src` as modified by the `Curve`, `Offset` and `Scale` settings into the output of the previous mixer as specified by the `mux` setting.

`mux` is a new setting. It controls how the value of the previous mixer and the output of this mix are combined. The available settings are:

- `replace`: This ignores the previous mixers output value, and only uses the output of the curve. This is used with the `Switch` settings to for flight modes and similar things.
- `add`: The two values are added together. This is how mixes behave in proprietary firmware.
- `mult`: The two values are multiplied together. Since the values are actually percentages, you need to divide the displayed value by 100 to get the actual displayed value again. I.e. a previous input of 50% and a curve output of 50% are .5 and .5, whose product is .25, so the final output value is 25%.
- `min`: The smaller of the previous mixer value and this mixers value is output by this mixer.
- `max`: The larger of the previous mixer value and this mixers value is output by this mixer.
- `delay`: The curve type should be `Fixed`, and the `Scale` setting scales the delay applied to the output of the previous mixer. This is used for creating a transmitter-controlled changing value.

The output of the previous mixer for the first mixer in a mix is the previous value from that channel. For this reason, you should set the `mux` of the first mixer to `replace` except under very special circumstances. Even when you want to reuse that old value, it's often best to do so explicitly by setting the `Src` of the first mixer to the channel the mix is on and using `replace` for the `mux`. By doing this, you can map the value through a curve before using it, which you can't if you use it implicitly with a `mux` other than `replace`.

The second new setting is `Switch`. If the `Switch` is not `None`, then the selected switch must be on. Otherwise, this mixer is disabled, and it's output will be the same as the output from the previous mixer

Finally, you can set the mixer for `Trim` or `No Trim`. If you have configured trim switches for this channel, their value will be added to the output of the mixer.

### *Reading .ini files*

You may have noticed that there are no pictures of the transmitter settings as are commonly provided in manuals for transmitter software. That's because the different transmitter types have radically different appearances, even though they share the same information. If you want pictures of your transmitter, you should read the deviationTx manual for it.

## Writing a deviation Tx model

They do have one thing in common - the files stored on the flash drive that describe each model. This are what are uploaded to the forums, and are commonly presented in tutorials and discussions there, so you need to understand them. They provide all the information needed to configure the mixers for a channel, so I'm going to use them to illustrate example configurations.

You can access the `model.ini` files by plugging your transmitter into a USB port of a computer and turning on USB mode. You can do that by holding down the `ENT` button when you power on the transmitter - which is recommended if you're going to modify a model. You can also access USB mode from the `Transmitter` menu. The current model will not be reloaded when you exit USB mode, and will be saved when you turn off the transmitter or load another model, so any changes made to it while in USB mode will be lost. The files will be in the `models` directory of the drive.

### Model description

As suggested by the extension, `.ini` files use the Windows ini format. It is divided into sections started by the section name in square brackets, followed by lines of the form `name=value`. However, the first part doesn't show up in a section, and describes the model settings. Here's a sample for the model we're going to configure in a bit:

1. `name=Blade 200 QX`
2. `mixermode=Advanced`
3. `icon=MINIQUAD.BMP`

So this is a model for a `Blade 200 QX`, the mixer will be shown in `Advanced` mode, and it uses the `MINIQUAD` icon. As a reminder, the mixer mode doesn't change what's in this file, just how the user sees the information.

1. `[radio]`
2. `protocol=DSMX`
3. `num_channels=5`
4. `fixed_id=200123`
5. `tx_power=150mW`

The radio section just describes the settings for the radio. Being a Blade model, this uses Blade's DSMX protocol. Being a relatively simple model, it has five channels. That's typical for quality multirotors, as they have 4 channels for attitude control and one to set the flight controllers flight mode.

Finally, we have:

1. `[protocol_opts]`
2. `Telemetry=Off`

which just turns off telemetry on this model, since the receiver doesn't support it.

## First sample mix

A relatively simple mix for channel 1 might look like:

```
1. [channel1]
2. template=expo_dr
3. [mixer]
4. src=AIL
5. dest=Ch1
6. [mixer]
7. src=AIL
8. dest=Ch1
9. switch=AIL DR1
10. scalar=70
```

Line 1 tells us this mix is for channel 1. Line 2 tells us that it's going to be displayed as a `Expo&DR` mixer, which only changes the user interface on the transmitter. Line 3 is the start of the first mixer. Line 4 sets the `Src` for this mixer to the `AIL` input. Line 4 is an artifact of the implementation, and will just be the display value for the channel this mix is on in every mixer. `Curve`, `Scale`, `Offset`, `Switch` and `Trim` are all unspecified, so take on the default values of 1-1, 100, 0, None and 1 (for on) in order.

The second mixer in the mix starts at line 6, and sets the `Switch` with lines 9 and `Scale` with line 10. If the `AIL DR` switch is in the 1 position, this mixer will become active and scale the `Src` by 70%. The `Mux` type is missing, so defaults to `Replace`.

This is a typical dual rate mix, toggling channel 1 between 100% and 70% of the `AIL` input with the `AIL DR` switch.

## Curved sample mixer

A mixer with a curve type might look like:

```
1. [mixer]
2. src=AIL
3. dest=Ch1
4. curvetype=expo
5. points=35,35
```

This adds the `Curve` type of `Expo` at line 4. The `points` setting on line five is the parameters for the curve. In this case, we'll be applying an expo of 35 to both halves of the curve.

## The last mix setting

The last settings you'll see in a mixer are the `mux` type, which is the `mux` setting for this mixer, and `usetrim`, which will always be 0 for `No Trim`, as otherwise it has the default value of `Trim`.

## Writing a deviation Tx model

Beyond the template type for the mix, there are other channel settings which will show up in the channel section. I'll cover those in the next section.

### ***Basic model setup***

In this section, we're going to create a `model.ini` file for a simple five channel aircraft - the Blade 200QX. This is a four channel model with an on-board flight mode set by the fifth channel. That fifth channel gives me a chance to show a simple switch setup. As such, it should work for most four channel aircraft if you just ignore the section about the flight mode control.

The first step is to configure the model. Go to the Model->Setup menu, change the name, select the protocol and the number of channels. Unless you're flying a helicopter, set the model type to Plane or Multi (if it's available and appropriate). Change the icon if you want to. Unless you know the model has telemetry, make sure it is off by selecting the protocol selector and changing it.

After doing that for a Blade 200QX, I have the following in my `model.ini` file:

```
1. name=Blade 200 QX
2. mixermode=Advanced
3. type=multi
4. [radio]
5. protocol=DSMX
6. num_channels=5
7. fixed_id=165827
8. tx_power=150mW
9.
10. [protocol_opts]
11. Telemetry=Off
```

### **Simple mixers**

The default configuration sets up the first four channels as simple mixers for the controls they are normally used for in that protocol. For now, we'll assume that it's right. So the first four channels in the `model.ini` file look like:

```
1. [channel1]
2. template=simple
3. [mixer]
4. src=THR
5. dest=Ch1
6.
7. [channel2]
8. template=simple
9. [mixer]
10. src=AIL
11. dest=Ch2
12.
```

## Writing a deviation Tx model

```
13. [channel3]
14. template=simple
15. [mixer]
16. src=ELE
17. dest=Ch3
18.
19. [channel4]
20. template=simple
21. [mixer]
22. src=RUD
23. dest=Ch4
```

Channel 5 isn't listed, because it's not configured by default, even though there are five channels in the model. Only the first four channels are configured. As you can see, they are each a Simple mix with a 1-1 Curve with a Scale of 100 and an Offset of 0. While it's possible to fly in this configuration, that's not optimal. It's normal to set up as second rate and possibly some expo, so let's do that now.

### **Expo&DR mixers**

I started with my normal 'copter dual rates, using 35% expo on a 100% rate for the high rates, and a linear curve with 70% rates for the low rate. This is only applied to aileron and elevator, and both are toggled by the AIL DR switch. On the Mixer page, select one of those two mixes and set the type to Expo&DR. On the High-Rate mixer set the Curve to EXPO, select that and set the Value to 35. Set the Mid-Rate Switch to AIL DR1, and then set the Scale to 70. Now do the same for the other cyclic channel. The results for channels 2 and 3 are:

```
1. [channel2]
2. template=expo_dr
3. [mixer]
4. src=AIL
5. dest=Ch2
6. curvetype=expo
7. points=35,35
8. [mixer]
9. src=AIL
10. dest=Ch2
11. switch=AIL DR1
12. scalar=70
13.
14. [channel3]
15. template=expo_dr
16. [mixer]
17. src=ELE
18. dest=Ch3
19. curvetype=expo
20. points=35,35
21. [mixer]
```

## Writing a deviation Tx model

```
22.src=ELE
23.dest=Ch3
24.switch=AIL DR1
25.scalar=70
```

The two mixes are pretty much identical, and we've seen it before. So - no surprises so far.

### Setting up flight modes

Now we need to control the flight mode. The 200QX uses the fifth channel for that, which is labeled GEAR. The relevant values are 100/0/-100 for the beginner, intermediate and expert flight modes. To enable that, you can use an Expo&DR mix. The High-Rate Curve is Fixed, with a Scale of -100. The Mid-Rate Switch is FMODE1. It stays Linked with a Scale of 0. For Low-Rate, the Switch is FMODE0 and the Scale is 100. The resulting channel is:

```
1. [channel5]
2. template=expo_dr
3. [mixer]
4. src=AIL
5. dest=Ch5
6. scalar=-100
7. curvetype=fixed
8. [mixer]
9. src=AIL
10.dest=Ch5
11.switch=FMODE1
12.scalar=0
13.curvetype=fixed
14.[mixer]
15.src=AIL
16.dest=Ch5
17.switch=FMODE0
18.curvetype=fixed
```

This should be obvious by now: in flight modes 1 and 2, the resulting mixer is enabled and its output replaces whatever was there. In flight mode 0, neither happens, so the first mixer's output of 100 is used.

However, you can use a switch as a source and do this with two mixers and mostly default values. So change the mix type to Complex with 2 Mixers. Set the first mixer to a Fixed Curve with a Scale of 0. Set the second mixer to a Min/Max Curve with a Source of FMODE0 and a Switch of !FMODE1. To do that, change the Switch selector to FMODE1 then select it to invert the switch. Make sure the Scale is set to 100. If you changed the previous Expo&DR mix, you'll have to fix that. Now the channel looks like:

```
1. [channel5]
2. template=complex
3. [mixer]
```

## Writing a deviation Tx model

```
4. src=AIL
5. dest=Ch5
6. scalar=0
7. usetrim=0
8. curvetype=fixed
9. [mixer]
10.src=FMODE0
11.dest=Ch5
12.switch=!FMODE1
13.usetrim=0
```

Much simpler than before. Now, in flight mode 1, the first mixers output of o is used. Otherwise, the second mixer is enabled, and it's output is used. That is either 100 in flight mode 0, or -100 in any other flight mode. Since the only other flight mode where this mixer is enabled is flight mode 2, that's the only one that gets the -100 value.

These two techniques cover all switch configurations that use a single channel.

### Channel configuration

We're almost ready to fly. We just need to tweak a few more things.

#### *Throttle hold*

There's one last thing that needs to be done to complete a model before I'd fly it. It needs a *Throttle Hold* switch. Since the engine is the primary source of energy in the aircraft, shutting the throttle down will minimize damage to everything involved in a crash. Your aircraft is always involved, but there may be property or even people as well, so killing the engine not only potentially saves you money, but is a good idea for safety reason. On most aircraft that can fly inverted, simply lowering the throttle won't shut the engine down. For that matter, it won't on many multirotors, including the 200 QX. Since you want this action to be as reflexive as possible, putting it on a switch means you do the same thing on every model, so we'll do that.

Setting up a safety switch is easy. Go to the `Mixer` menu, and then select the button with the `THR` label on it. Use the Safety selector to select the switch you want. In this case, we're going to use `Virt1` as a switch, for reasons I'll explain shortly. Then set the `Safe Val` to `-125`. On most models, you should set that value to `-100` initially, and only lower it if that's not enough to properly stop the engines. Some ESC's may assume that the less-than-100 value you start with is actually 100, and adjust their behavior appropriately. The 200QX doesn't have that problem, but I found that at `-100` it doesn't shut itself off as expected. Channel 1 now looks like:

```
1. [channel1]
2. safetysw=Virt1
3. safetyval=-125
4. template=simple
5. [mixer]
```

## Writing a deviation Tx model

6. `src=THR`
7. `dest=Ch1`

It just adds the values we changed from their defaults.

### ***Other channel configuration***

The other things you might want to change on the channel configuration page are `Reverse`, `Fail-safe`, and the two `Scale`'s, though `Reverse` is by far the most common one.

`Reverse` does just what it sounds like - it reverses the channel so that as the output from the mix goes from -100 to 100, the value sent to the receiver goes from 100 to -100. Even on aircraft and radios from the same company, this is sometimes necessary. While you could do this by negating the scale for every mixer in a mix, it's easier to do it once on the channel configuration page. It also means you can work on the mix without having to invert your thinking.

Similarly, the `Scale`'s act as a `Scale` on the output from the entire mix, and you would do it here for much the same reasons. It's similar to the travel setting found on some proprietary transmitters.

`Fail-safe` sets the failsafe value in the receiver. This only works for the Devo protocol. If you're not familiar with it, it's the value the receiver will output on this channel if it loses contact with the transmitter.

### **Other niceties**

If the `THR` Safety switch were a real switch, and not an unconfigured virtual channel, the model would be fine to fly now. So let's fix that channel, and add a few other things that will make flying safer and more fun.

### ***Sticky throttle hold***

Let's start with virtual channel 1, which I chose for the throttle hold switch. We're going to create a *sticky* switch, that only goes off if the throttle output is actually 0. This means that you won't turn off the hold thinking the throttle is off when it's not, only to have the props start spinning - and the aircraft start flying - when you aren't expecting it. This also handles cases where other switches may need to be in the right setting for the throttle output to be zero.

So now on the `Mixer` page, find `Virt1`. Select that, and change the name to something like `T Hold`. Naming the virtual channels makes them a lot easier to deal with. Now back to the `Mixer` page, and edit the mix proper. Set the type to `Complex` and `Mixers` to 3.

Set the first mixer - `Page 1` - so the `Src` is `T Hold`, and the `Curve` is `Min/Max`. This reads the old value of the channel, and outputs -100 if the switch was off, and 100 if it was on. This normalizes the value, as we'll be mangling it pretty badly soon.



## Writing a deviation Tx model

Go to the second mixer - Page 2 - and set the mux to add, the Src to THR, the Curve to 1-1, and the Offset to -1. So, the output from this channel will run from -1 to 199 if the switch was on before, so the switch would only go off if the throttle output is -100 or -99, which is close enough for every model I've tried. If the switch was off, the output will go from -201 to -1, and the switch will stay off no matter what the throttle does.

Finally, the third mixer - Page 3 - is a simple switch setup. Set the switch to RUD DR0, and the Curve to Fixed. The mux should already be replace. This mixer throws out the previous mixer output if the RUD DR switch is in position 0, and passes that previous value through if the switch is in position 1.

Our virtual channel should look like this:

```
1. [virtchan1]
2. name=T Hold
3. template=complex
4. [mixer]
5. src=Virt1
6. dest=Virt1
7. usetrim=0
8. curvetype=min/max
9. points=0
10. [mixer]
11. src=THR
12. dest=Virt1
13. offset=-1
14. usetrim=0
15. muxtype=add
16. [mixer]
17. src=AIL
18. dest=Virt1
19. switch=RUD DR0
20. usetrim=0
21. curvetype=fixed
```

So now RUD DR0 turns on the throttle hold, and RUD DR1 won't turn allow it to be turned off until the throttle output goes to -99 or -100. A sticky throttle hold.

Ok, now it's time to deal with the different transmitters. If you have a Devo 8, 10 or 12, you can use the switches I'm going to use. If you have a Devo 6, you'll have to pick different ones, but it's actually got better names for this configuration. You can, for instance, use HOLD1 for the hold switch. If you have a Devo 7e - well, it can be made to work. If you have extra switches installed, you can use those by changing the names. If not, you'll want to look through the last section - on Trim buttons - to get everything set up.

## *A simple mix*

The 200QX, like many quadcopters, tends to lose lift when it starts moving sideways. This is something that is normally compensated for with a mix in proprietary software. Personally, I don't like doing this, but it's a matter of taste. So I'll show you how to add compensation in the throttle mix. Please note that I **have not** flown with this mix in place, so you'll want to test it yourself!

Go the `Mixer` page, and open the mix for `THR`. Change it from a `Simple` to a `Complex` mixer. If you look through the first mixer, you'll see that it's still the same mixer you had in the simple mix. This is generally the case. Changing the mix type won't change the mixes, at least if they are all still there.

Now set `Mixers` to 3. Go the the second `Page`, and set the `Src` to `AIL`, the `Curve` to `ABSVAL`, the `mux` to `add`, and the `Scale` to 20. The `ABSVAL` `Curve` means the offset of the `AIL` will be used, not the actual value. The `Scale` is a guess for what percentage of the aileron offset we need to add to the throttle.

Now repeat the process on `Page 3` with `ELE` as the `Src`. Your new throttle channel should look like:

```
1. [channel1]
2. safetysw=Virt1
3. safetyval=-100
4. template=complex
5. [mixer]
6. src=THR
7. dest=Ch1
8. [mixer]
9. src=AIL
10.dest=Ch1
11.scalar=20
12.usetrim=0
13.muxtype=add
14.curvetype=absval
15.points=0
16.[mixer]
17.src=ELE
18.dest=Ch1
19.scalar=20
20.usetrim=0
21.muxtype=add
22.curvetype=absval
23.points=0
```

Note that `No Trim` was also on in the new mixers. Also, if you start moving on a diagonal, both aileron and elevator inputs will be added to your throttle, which may not be desirable. But you'll have to tune this yourself.

## Timers

If you've looked at the main page and played around some, you may have figured out that the top number on the left is the throttle output. But the other two haven't moved. Those are timers, and we're going to set them up now.

Since the motors turn whenever throttle hold is off, I want the timers to run when throttle hold is off, as that means the motors are running. For many aircraft, setting the Switch to THR will be a better timer, as it runs when the throttle output is above -100. However, running when it's above -100 also means that the `T_HOLD` channel, which works fine as a switch for the throttle `Safety`, won't work as a switch for a timer. So we need a new channel.

So, back to `Mixer` page, pick another virtual channel, and set the type to `Simple`, the `Src` to `!T_HOLD`, and the `Curve` to `Min/Max`. Give it a name like `DT_HOLD` (for Display Throttle Hold). Note that you can't use `!DT_HOLD` as the switch and not invert the `Src`. This is a bug in the timer code, but should be fixed in the next release.

Go to the `Model` menu, and then the `Timers` page. Set `Timer1` to a countdown timer, `Switch` to `DT_HOLD`, `Reset sw` to `ELE_DR1`, and `Start` to `9:00`. The `Switch` setting will cause the timer to only run when the throttle is not held.

When first setting up a model, you should probably set this `Timer1` to `stopwatch`. After you've flown a few times and gotten a feel for how long the batteries last, change it back to `countdown`. The transmitter alarm will then go off 30 seconds before the timer runs out, and every 10 seconds after that to warn you that it's time to land. Those two times can both be set on the `Transmitter config` page, under `Timer settings`.

Now go down to `Timer2`, and set it to `permanent`. Again, set the switch to `DT_HOLD` or `THR`, whichever you used for `Timer1`. so it runs whenever `Timer1` is running. This timer always counts up, and is saved between flights. It will record the total flight time on the model. If your model needs regularly scheduled maintenance, this is the timer to use for that.

If you now go back to the main page and tick the throttle up, both timers should start running. Here's what my timer config looks like in the `model.in` file:

```
1. [timer1]
2. type=countdown
3. src=Virt1
4. resetsrc=ELE_DR1
5. time=420
6. [timer2]
7. type=permanent
8. src=Virt1
```

9. val=21271496

### ***Channel monitor***

Go to the `Model` menu, and then `Main page config`. If you have a color touch screen, select the plus sign at the top. On either display, find the `Menu 1` item and change its value from `Telemetry monitor` to `Channel monitor`. Now a long-press on the `UP` button will bring up the channel monitor, which is useful for checking your mixer.

### ***Toggle icons***

One - well, three - last things to do, and then we're ready to fly. These really are just display tweaks, and won't affect the flight at all. I have to apologize in advance - here is one of the few places where the different transmitters configuration is radically differently. How you set these up, which ones are on by default, and what icons are available are different depending on the transmitter.

Go to the `Main page config`. If you have a black and with display, find `Toggle1`, and use the source selector next to it to select `T Hold`. Now select the `Toggle1` button, use the `Down` button to select `Pos 1`, and then the `Left` button to select an icon to indicate throttle hold. If you have a color touch screen, select the box labelled `Pos 1`, and then select the wrench icon. Now change the `RUD DR` source to `T Hold`, select `Edit`, select `Pos 1` and then a throttle hold icon.

If there is a toggle using `ELE DR` as a source, change the source to `FMODE`. Otherwise, pick an unused toggle and set its source to `FMODE`. When you edit it, you will see that it has `POS 0`, `POS 1` and `POS 2` unless you're using a Devo 7e. Change the icons for all three to something appropriate for the flight modes that switch enables.

Finally, edit the icons used in the `AIL DR` toggle and set position `I` to a low rate indicator.

Going back to the main page now, you'll see that your selected throttle hold icon goes on when the throttle is held, the low rate indicator goes on when you are in low rates, and the flight mode is displayed on the screen. While I don't recommend looking at the transmitter when you're flying, it's nice to be able to check these things before you take off.

### ***Finding RTF & BNF channel mappings***

The techniques used for basic model setup will work on most kits and RTF and BNF models from the companies that created the protocols. But they don't always work with off-brand RTF and BNF models, since some of those companies don't sell transmitters except as replacement parts for specific models, and will change the protocols and channel mappings seemingly at random. So if you've got one of those, you may have to work out what channels actually do what.

This problem doesn't arise with proprietary firmware - you either fly the manufacturer's models or models that license their protocol, and normally can't change protocols. If you can, it's only to that manufacturer's protocols. And even if you could get an unlicensed model to bind, the firmware isn't

## Writing a deviationTx model

designed to let you change the channel mapping. Since this problem is unique to open source firmware, we'll take a detour to talk about how to handle it when you set up a model.

The best way to find the channel mapping for a model is to search the web for it. rcgroups.com is a good source, but not the only one, so you might search google as well. But if that doesn't turn anything up, you're going to have to experiment with things yourself.

### *Experimenting to find the channels*

This is a simple idea - just try things and see what they do. This should be done in deviationTx's advanced mixer mode, since you can't change the channel source in standard mixer mode. But there are some precursors you'll need to deal with.

First things first: **Take all the propellers off the aircraft!** This is a critical safety step, as it may go to half or even full throttle the first time you bind it.

Some RTF aircraft - especially the toy-grade ones - may require some stick input to bind. Normally, that's going to full throttle and back down. If so, you'll have to try each channel in turn to see if the aircraft binds. Once you get it to bind, You know where the throttle channel is.

### Throttle

If you didn't find the throttle in the bind process, watch the control surfaces when you bind. If the throttle doesn't go on when you bind it, then the throttle is correct.

If the throttle goes on, try raising the throttle stick and see what response you get. If the throttle output changes - especially if it goes down - you've find the throttle. If nothing happens, then one of the other channels is probably the throttle. Try them one at a time until you figure out which is the throttle, and then if necessary rearrange your channels so the throttle stick controls the throttle on your aircraft

### Other control surfaces

The other control surfaces are simple after that. Just try them, and see what responds. If the wrong thing moves, you'll want to rearrange your channels.

### *Rearranging channels*

While doing this experimentation, you may need to rearrange the flight control channels to be different from the default settings. This is easiest to do if you just swap the control with the channel it should be on.

When you look at the mixer settings, the first few channels will have a button with a label that looks like #-?????. The # will be the channel number, and the ????? will be the usual usage for that channel and protocol. There is also a button labelled *Simple*, which is the type of the mix, and a label which initially matches the ????? on the button, which is the current input source for that channel.

## Writing a deviationTx model

If the elevator stick currently controls the aileron, you'll want to swap those. So find the two channels that currently have the labels `ELE` and `AIL` not on a button. Those are the two you want to swap.

To change the input used by a channel, select the `Simple` button. Then find the `Src` selector, and move it to the input you want to control that channel. The four controls are always in the same order, no matter what protocol you use: `AIL`, `ELE`, `THR`, `RUD`. Finally, select `Save` and you're done with that channel.

### *Other controls*

At this point, you've got the basic controls working right, so you need to figure out any other channels that may exist.

They come in four basic flavors.

*Toggles.* Channel high gets one behavior, channel low gets a different one. Like lights on and off.

*Multi-value toggles.* Like toggles, but with more than one behavior and output values. Like flight modes in the 200QX config we just finished.

*Edge-triggered.* Usually on a button. Causes something to happen when the channel goes from high to low or vice versa. Like taking a picture.

*Analog.* Uses a range of values like a flight control. Like a camera gimbals position or speed control.

For toggles and edge-triggered controls, assign a switch as the source for the suspected output channels, and just try them one at a time. If nothing happens, throw the switch the other way to see if it changes. If it's a toggle, you can change the sense of the switch by inverting the source in the mixer - see the section on fine tuning below.

If you want an actual button, you can use the deviationTx grim button switch features. See the last section for how to set these up.

For multi-value toggles, you either need some indication from the aircraft when it changes mode, or documentation on what ranges do what. If you get an indicator, then use an analog input as a `Src` for the suspect channels. Go to the `Channel monitor` page on your controller. Now adjust the analog control, and when you get the indicator you want, note the value for that channel on the monitor. Later, you can set things up on switches as described for the 200QX flight mode controls.

If you don't have any spare analog inputs, you can use an trim buttons to create a full-range analog control. Details are in the last section.

### *WLtoys channel hacks.*

Just a warning. The older WLToys aircraft - anything using the FlySky protocol - use a hack to simulate extra channels. And I mean hack in the sense invoked by the phrase “repairing furniture with an axe”.

The protocol is still a four channel protocol. However, the controller will use values in one of the existing channels well outside the normal range. The flight control board will strip these values off, and treat it as an extra channel.

The WLToys option in the deviationTx implementation of this protocol uses channels 5–8 as switches to control those extra high values - at least as they were in the v9x9 quadcopters. Later aircraft apparently use the same hack, but not quite the same way.

In particular, some of the implemented version create a throttle value way above the normal range, and later WLToys aircraft will try and use that value for the throttle. This is not a good thing.

### *Reversing a channel*

If the throttle starts at full when you bind with the throttle stick down, or a control works backwards, you need to reverse the channel. This is something that also happens with proprietary firmware, so you may be familiar with it.

The best way to reverse a channels output is to select the channel button - the one with the channel number and control on it for the ones we’re talking about. Then select the button that says `Normal` so it says `Reversed`, and select `Ok` and you’re done.

There are other ways to reverse a simple channel, but this is the one that always reverses the output, and won’t require more work with a more advanced configuration

### **Throttle safety**

If you’ve had to change the throttle channel, there’s one last thing to do. When you first load a model, either by turning the controller on or using the model load menu, deviationTx will not try and bind until the throttle is at a safe level. If you’ve changed the throttle this will check the wrong channel, meaning you could bind with the throttle up, and will probably have to clear the warning about the wrong channel every time you use this config. While you can set all of these levels, they aren’t editable on the controller.

So, the first step is to plug your controller into a USB port on your computer. Then turn the controller on while holding the `ENT` button. This is the same thing you did to install the file system when you installed deviationTx on the controller.

## Writing a deviationTx model

In the disk that this puts on your computer, you want to edit the file `models/model#.ini`, the file corresponding to the model number for your aircraft. Any text editor should do, but check the deviationTx forums if you have problems.

Find the `[safety]` section. It should have one line in it, that says `Auto=min`. Change that to be `Ch#=min`, where `#` is the channel number that controls the throttle. This will cause deviationTx to check that channel instead of the default throttle channel for this protocol.

If you are getting warnings about `Ch3` being the wrong value, this means that `Auto` isn't working right for some reason. Changing this to be your throttle channel as above will clear that up.

### *Advanced model setup*

Ok, we've set up a simple model. But so far, we've not done anything that good proprietary firmware can't do. Even modest firmware should be able to do everything but the sticky throttle hold. So now it's time to get serious about this.

### **A panic switch**

The Blade 200 SRX has a *Panic* switch that changes the flight mode to beginner mode, and disables the cyclic inputs. While I'm not sure if that's a good idea, it's easy to implement with deviationTx.

In the `AIL` and `ELE` channel configurations, set `GEAR1` as the `Safety`, and set the `Safe Val` to `0`. For channel five, the `Safety` is again `GEAR1`, but the `Safe Val` is `100`. We'll look at the gear channel in a second. The aileron channel is now:

```
1. [channel2]
2. safetysw=GEAR1
3. template=expo_dr
4. [mixer]
5. src=AIL
6. dest=Ch2
7. curvetype=expo
8. points=35,35
9. [mixer]
10. src=AIL
11. dest=Ch2
12. switch=AIL DR1
13. scalar=70
```

As you can see, all we've done is changed it so the `GEAR1` switch disables the cyclic inputs. Since it also switches the flight mode to beginner, the 200QX will auto-level.



While we're doing this, it might be a good idea to add an icon to the main page to indicate that it's on. So back to the `Main page config`. and get to the list of toggles. Pick an unused toggle, set it to `GEAR`, and then edit it to set an appropriate icon for `Pos 1`.

### Stagility mode

*Stagility* mode was invented by someone using proprietary firmware. They did it by mixing the cyclic controls on a Blade 350QX into the flight mode channel to kick it from Stability to Agility mode - or Intermediate to Expert mode on the 200QX. We can duplicate that, and do it better by not mixing into the flight mode channel, but using the mix as a switch that changes the flight mode channel properly.

**First warning:** "stagility mode" on the 200QX is not supported by Blade or Horizon Hobby, and the flight controller on the 200QX has been known to become disoriented by rapidly changing the flight mode. So use this mode at your own risk.

**Second warning:** The 200QX version 2 firmware in 3d mode has an intermediate flight mode that has stagility-like behavior. Using this mode there is pointless, and not recommended at all.

The first step is to create a channel that will be an on switch with enough cyclic input, and off otherwise. There are a number of options for doing this, but let's set up what I use and then talk about the others.

Pick an unused virtual channel on the `Mixer` page. Change the name to `Stagility`. Now edit the mix. Change it's type to `Complex` (what else?) and set up 3 `Mixers`. The first mixer will have a `Src` of `ALL`, and a `Curve` of `ABSVAL`. The second will have a `Src` of `ELE`, `Curve` of `ABSVAL` and a `mux` of `max`.

So we now have the output of the second mixer being the larger offset of the two cyclic controls. Other alternatives include a `mux` of `add`, or being mathematically accurate and multiplying each cyclic by itself and then adding them. However, taking the `max` works well in practice. This means we now have a value between 0 (no cyclic) and 100 (one or the other at full cyclic).

The third mixer is going to provide the offset to make this so the switch will sometimes go off. If you leave off the third mixer, flight mode 1 would just be expert mode. So we're going to add a `Fixed Curve` with a `Scale` of `-80` and a `mux` of `add`. This means one of your cyclics will have to be 80% of the way to the edge to trigger the switch to expert mode.

The gear channel also needs a minor tweak. Change the first mixer `Curve` to `Min/Max`, the `Src` to `Stagility`, the `Scale` and `Offset` to `-50`. We're repeating the technique used in the second mixer, only this time with the `Stagility` switch. Using `Min/Max` here means the output values will be 0 and -100. Well, after applying the `Scale` and `Offset` to the initial value of -100 and 100, they'll be 0 and 100. So our `Gear` channel with the updated mix and `Safety` now looks like:

## Writing a deviation Tx model

```
1. [channel5]
2. safetysw=GEAR1
3. safetyval=100
4. template=complex
5. [mixer]
6. src=Virt2
7. dest=Ch5
8. scalar=-50
9. offset=-50
10.usetrim=0
11.curvetype=min/max
12.points=0
13. [mixer]
14.src=FMODE0
15.dest=Ch5
16.switch=!FMODE1
17.usetrim=0
```

Much the same as before: if the `FMOD` switch is not in position 1, we output -100 and 100 depending on whether or not it's in position 0. If it's in position 1, instead of just outputting 0, we output either 0 or -100 depending on the sticks and that third mixer in the `Stagility` channel.

### Dialable stagility mode

The downside to all this is that changing the stagility behavior means you have to change the mix, which is clumsy. There ought to be an easier way. And sure enough, there is.

Instead of using a Fixed curve for the third mixer in the `Stagility` channel, let's pick an analog input source. I'm using a `Devo10`, so I can just pick `AUX5`. if you don't have a convenient dial or slider, see the last section on tricks with trim buttons to see how to turn a trim into an analog input.

So, back to the `Mixer` page, and then the third mixer in the `Stagility` channel. Set the `Src` to `AUX5` or whatever you are using for your analog input. Set the `Curve` to 1-1, the `Scale` to 50 and the `Offset` to -55. This means that as `AUX5` goes from -100 to 100, what gets subtracted from the cyclic offset - and hence how much offset is needed to trigger expert mode - will go from 101 to -99. So when `AUX5` is all the way down, you just have stock intermediate mode. Dialed all the way up to 100, then the slightest motion off center will turn on agility mode. And of course, you can get all the behaviors in between. Now that we're done with it, here's the `Stagility` channel:

```
1. [virtchan2]
2. name=Stagility
3. template=complex
4. [mixer]
5. src=ELE
6. dest=Virt2
7. usetrim=0
```

## Writing a deviationTx model

```
8. curvetype=absval
9. points=0
10. [mixer]
11. src=AIL
12. dest=Virt2
13. usetrim=0
14. muxtype=max
15. curvetype=absval
16. points=0
17. [mixer]
18. src=AUX5
19. dest=Virt2
20. scalar=50
21. offset=-51
22. usetrim=0
23. muxtype=add
```

The difference between this and the non-dialable version is only the third mixer, which had a `Fixed` curve. Of course, the reality is that you may want to adjust the `Offset` here to make sure that you actually get a blue light when in flight mode 1 with `AUX5` dialed to +100. Stick calibration, trim, and who knows what might be enough to kick it up to turning on expert mode.

Given how important `AUX5` has become, let's put it on the main page. Unfortunately, you can't do that directly - we need to make it available in an output channel. So to the `Mixer` page, and pick an unused virtual channel. Name it `St Dial` or something similar for "stagillity dial". Set the type to `Simple`, the `Src` to `AUX5`, the `Curve` to `1-1`, the `Scale` to `-50` and the `Offset` to `51`. This makes the output value goe from 101% to 1% as `AUX5` goes from -100% to 100%. It's now how much you have to offset the cyclic sticks to switch to agility mode. To display it, go to the `Main page config page`, and find the unused box. Or add one if you need to, and position it where you want it. I put it underneath the timers and above the rudder trim indicator on my Devo 10. Set the input to `St Dial`, and you're done. I could add the `St Dial` channel here, but I don't think you need it.

One last thing you might do is create a safety for `AUX5`, just to remind you to set it when you load this model. I know I like having that. So plug your transmitter into the USB port on your computer, hold down the `ENT` key, and power it on. In the disk that this puts on your computer, you want to edit the file `models/model#.ini`, the file corresponding to the model number for your aircraft. Any text editor should do, but check the deviationTx forums if you have problems.

Find the `[safety]` section. It should have one line in it, that says `Auto=min`. Below that, add a line that says `AUX5=min`. That will cause deviationTx to check that input as well as the throttle channel when this model is loaded. If it's not in the position where the sticks don't change modes, it reminds you to correct it.

## Initialize the flight controller

Ok, one last tweak that you can't do with most proprietary controllers. Right now, you have to do a stick dance to initialize the flight controller. We're going to automate that, so that turning off throttle hold initializes the flight controller.

First, we need a moving value just to start with. The `Delay mux` setting gives us that. So choose a virtual channel, give it the name `Delay`, and set `Mixers` to 2. The first mixer should have a source of `T Hold` and a `Curve` of `Min/Max`. If you remember, the output of `T Hold` covers a wide range. Using a `Min/Max` curve causes it to go back to -100 to 100.

Now add the delay. Literally. Set the second mixer `Curve` to `Fixed`, and the `mux` to `delay`. Set the `Scale` to 3 to start with. You'll want to experiment with that, though. Here's the resulting `Delay` channel.

```
1. [virtchan4]
2. name=Delay
3. template=complex
4. [mixer]
5. src=!Virt1
6. dest=Virt4
7. usetrim=0
8. curvetype=min/max
9. points=0
10. [mixer]
11. src=AIL
12. dest=Virt4
13. scalar=3
14. usetrim=0
15. muxtype=delay
16. curvetype=fixed
```

We now need to use this moving value to make the rudder channel cycle as required by the 200QX. So change it from a `Simple mix` to a `Complex mix`. Leave the first mixer as is, and change the second one to have a `Src` of `Delay`, a `mux` of `add`, and a 5 `Point Curve`. Select the `Curve` button so you can set the points in the curve. You want them to be 0, 100, 0, -100 and 0 in order. We are going to be adding either the first or last value to the rudder all the time it's not initializing, so they need to be 0. In between, you need to have a 100 and a -100 in that order. I put the extra 0 in the middle because I like symmetry, but it could be at either end as well. I now have:

```
1. Rudder channel:
2. [channel4]
3. template=complex
4. [mixer]
5. src=RUD
6. dest=Ch4
```

## Writing a deviation Tx model

```
7. [mixer]
8. src=Virt4
9. dest=Ch4
10.usetrim=0
11.muxtype=add
12.curvetype=5point
13.points=0,100,0,-100,0
```

You will want to experiment with the `Scale` value on the `Delay` channel. I can set it to 2 and things still work, but they don't if I set it to 1. I also know it works at 5, and presumably 4. I set it to 3 because I want to avoid possible boundary problems from it being 2.

### ***Advanced Flight Mode Settings***

While the 200QX is a great multirotor, it's an older flight controller and only has 3 flight modes, which works very well with the 3-position switches on Devo transmitters. However, modern flight controllers have 5 or more flight modes, and the ability to use a single channel to set them.

Four or five flight modes with two switches are both easy, you just need to set up a three position switch for three flight modes, then add one a mix that's switched on a second switch, and outputs either a fixed value or the value of one of the other switch positions. Here's a five-way switch on `FMODE` and `MIX`. `MIX0` causes `FMODE` to control the output for values of 100, 50, 0, and then `MIX1` outputting 50 and `MIX2` 100.

```
1. [channel5]
2. template=complex
3. [mixer]
4. src=AIL
5. dest=Ch5
6. scalar=50
7. usetrim=0
8. curvetype=fixed
9. [mixer]
10.src=FMODE0
11.dest=Ch5
12.switch=!FMODE1
13.scalar=50
14.offset=50
15.usetrim=0
16.curvetype=min/max
17.points=0
18. [mixer]
19.src=MIX1
20.dest=Ch5
21.switch=!MIX0
22.scalar=25
23.offset=-75
```

## Writing a deviation Tx model

```
24.usetrim=0
25.curvetype=min/max
26.points=0
```

The first two mixers are the same as the previous flight mode selection, except that the scalar and offset move it to the values we want. The last mixer is similar to the second one: when `MIX0` is not on, disable this mix, so we use the value selected by `FMODE`. Otherwise, use the value selected by `MIX1` and `MIX2`, again with a scalar and offset to get the desired values.

This clearly doesn't work with if you need six flight modes. In this case, a 2-position switch and the `Mult mux` type can be used, like so:

```
1. [channel1]
2. template=complex
3. [mixer]
4. src=GEAR1
5. dest=Ch1
6. usetrim=0
7. [mixer]
8. src=AIL
9. dest=Ch1
10.switch=FMODE1
11.scalar=67
12.usetrim=0
13.muxtype=multiply
14.curvetype=fixed
15. [mixer]
16.src=AIL
17.dest=Ch1
18.switch=FMODE2
19.scalar=33
20.usetrim=0
21.muxtype=multiply
22.curvetype=fixed
```

This uses the `GEAR` switch to toggle between positive and negative values based on `FMODE`. For `GEAR1`, `FMODE` goes from 100 to 67 to 33, and `GEAR1` negates those values.

The `add mux` type can be used to do something similar. Set up a three-positions switch to output values of 100, 67, 33, and then use a final mix enabled by the second switch with an `add mux` type, `fixed` curve type, and a scalar of -100.

### ***Tricks with trim buttons***

Normally, trim values are associated with an output channel, and the trim value is added to the channel after the mix value is calculated. But deviation Tx has an incredibly flexible trim support, and can do a

## Writing a deviation Tx model

lot more than that. Not only can trims be used to create a variety of different kinds of switch behaviors, the configuration buttons can also be used as trim buttons. This lets you create controls that aren't available on your transmitter. Of course, these means you get no feedback on their value like you do with a real switch. Toggle icons on the main page can help with that.

Because of their use as normal trim buttons, these extra controls interact oddly with output channels. If you want to use one as a `Src` in some mixer, you have to make sure there is `No Trim` on that mixer, otherwise the trim will get added twice, making the output go from -200 to 200. This means it has to be in a `Complex mix` so you can set `No Trim`.

If you want to use an output channel as the switch instead of a virtual channel as described below, then you have to make sure you have `No Trim` instead of `Trim`, otherwise it'll be added twice. Of course, this also means the mix configuration doesn't really matter, so long as the output is always 0. But this usage isn't documented, so the behavior may change in the future. I again recommend sticking with virtual channels unless you've run out.

As a final note, the navigation buttons only work on the main page. If you're not on the main page, they don't do anything. For the non touch-screen transmitters, this means that the ENT button can't be used, because you use it to start configuration! It works fine on the touch-screen transmitters, though. Similarly, you want to be careful about using the UP and DOWN buttons as momentary switches, as holding them down will cause you to jump to the shortcut page, which will disable changes on it and leave it on.

### A momentary button

None of the Devo transmitters have any kind of momentary button. There's really no good way to simulate them with switches. When configured, these buttons provide a 1 when the button is down, and a 0 otherwise. For the 200QX config, you could use one to reset the timers.

Momentary buttons are also great for edge-triggered controls, like the WLT toys cameras. I use one for changing flight modes on the Blade NanoQX.

So let's create one. Go to the `Mixer` page, and pick an unused virtual channel. Rename it to `Reset`. Set the mix type to `Complex` with `1 Mixers`. Set that mixer to have a `Src` of `Reset`, a `Curve` of `1-1` and a `Scale` of `0`. Make sure the `Trim` is on. Your channel should like like this:

1. `[virtchan5]`
2. `name=Reset`
3. `template=complex`
4. `[mixer]`
5. `src=Virt5`
6. `dest=Virt5`
7. `scalar=0`

## Writing a deviation Tx model

Now got to the Trims page. Choose an unused trim, and set the Input to Reset, and the Trim Step to Momentary. The Momentary setting, like the other switch settings, is to the left of the 0.1 value in the Trim Step selector. Finally, set the Trim + to Exit. In the model.ini file, you get:

```
1. [trim5]
2. src=Virt5
3. pos=Exit
4. step=193
5. value=-100,0,0
```

You now have a switch that will be on when you're holding down the EXT button, and off otherwise.

### A toggle button

If you use the above configuration except set the Trim Step to Toggle, the channel value will toggle between -100 and 100 every time you press that button. They are great for things that you can see the effect of on the aircraft, especially if they are cosmetic in nature. Turning lights on and off, for instance. In the 200QX model, you could use one as the switch for the last mixer in the T Hold channel, but the possibility of bumping it and shutting down the motors while in flight makes me a bit leary of that.

One interesting use of these is for tracking an edge-triggered control's state. You can set up a second virtual channel and a second virtual switch that uses the same button for Trim +, but has a Trim step of Toggle instead of Momentary. If you set up a main page toggle to use this switch, it will change the icon every time you press the button. So you can use it as a flight mode icon on the NanoQX, or to note when video is recording on a camera that uses that button for start and stop.

The channel setup is identical to the momentary button, except you probably want to use a name like Lights. As state, the Trim Step is Toggle. Here's a model.ini showing a pair of trims providing a momentary and toggle on the TRIM\_L+ button on the Devo 10:

```
1. [trim5]
2. src=Virt5
3. pos=TRIM_L+
4. step=193
5. value=-100,0,0
6. [trim6]
7. src=Virt6
8. pos=TRIM_L+
9. step=192
10. value=-100,0,0
```

Note that value settings in the model.ini file. It's the last value of this trim. Like any other trim value, it is saved between sessions. You'll want to watch out for that if you depend on it being in the right state when you first load the model.



## A switch

The toggle switch above has the problem that you need to know the current state in order to know what state you'll get when you push it. If you instead configure it with an `On/Off Trim Step`, you get to use one button to turn it on, and a second to turn it off. This makes a nice way to create an extra two-way switch on a transmitter that doesn't have enough of them. On the 200QX config, one might make a nice panic switch, or you could use one for dual rates.

Of course, it does require two buttons, so I recommend using the ones that are already paired. The throttle trims are available on all transmitters, and unused in most models. Some transmitters have extra pairs of trim buttons. And they all have `Left/Right` and `Up/Down` button pairs.

Again, the configuration is pretty much the same as before, except that `Trim Step` is now `On/Off`, and you have to configure the `Trim -` button as well. So here's what the `model.ini` looks like for an `On/Off` trim pair:

```
1. [trim5]
2. src=Virt5
3. pos=TRIM_L+
4. neg=TRIM_L-
5. step=191
6. value=100,0,0
```

## Adding an analog input

Again, this doesn't add a new capability, but provides an extra control should your transmitter run out. For the 200QX config here, you could use this for the `Stagility` dial.

The channel setup is the same as for the switches above. Either a virtual channel with a `1-1 Curve`, `o Scale` and `Trim`, or an output channel with an output of `o` and `No Trim`. But for these, we are going to use the numeric steps of the trim buttons, not the ones below `.i`. You get 100 steps in each direction. So if you set the `Trim Step` to `i`, the trim value will go from `-100` to `100`. If you want a shorter range, you can set the `Trim step` to less than `i`.

If you want a longer range than `-100` to `100`, that's a bit more work. You can't get it directly, no matter what you set the `Trim step` to. Even using an output channel with `Trim` on the mix won't work. You can use the virtual channel as a `Src` and leave `Trim` on in the mix. But in that case it's probably better to just set the `Scale` instead.

These are still **trim** values. As such, they have features common to all trim values:

- They beep when you set them to zero.
- If you are holding the button, it'll stop changing at zero.
- The `Switch` setting will enable or disable its use.

## Writing a deviation Tx model

- It is saved when you stop using the model to be restored when you next load it.

And here's a `model.ini` example with a I.O step using the `Left` and `Right` buttons, with `MIX` as a switch:

1. `[trim6]`
2. `src=Virt5`
3. `pos=Right`
4. `neg=Left`
5. `step=10`
6. `switch=MIX`
7. `value=27,0,0`

In this case, the step value has a meaning. The `Trim Step` will be one tenth of this value.

### A three position switch

As a special case of the above, you can create a three position switch. Unfortunately, it won't show up with three positions in the various switch and source settings in the configuration. You'll need to use two more virtual channels to get those. For the 200QX, this could provide a flight mode switch on a transmitter that doesn't have a three position switch.

Start by setting up the virtual channel as usual. Let's call it `FMODE0`. As a switch, it will only be true in the `-100` position. The trim configuration is just the analog input as above with `FMODE0` as `Input`, except set the `Trim step` to `100`. It now moves between the three values `-100`, `0` and `100` as you use the up and down buttons.

Now set up a second virtual channel. It's going to be `FMODE1`. It's a `Complex mix`, with `1 Mixers`. That mixer has a `FMODE0` as the `Src`, an `ABSVAL Curve`, `Scale is -1`, and `offset is 1`. `No Trim`, of course. And now the third, as `FMODE2`. Another `Complex mix` with `1 Mixers`. `Src` is again `FMODE0` (make sure you use the virtual channel, not the switch if you already have one!), `Curve is 1-1` and `Scale is -1`. And again, `No Trim`.

These three virtual channels, when used as switches, will follow the trim values as expected. `FMODE2` will be on and the others off when the trim value is `100`. `FMODE1` when it's `0`, and `FMODE0` when it's `-100`.

Here's what you get in `model.ini` with all of them together:

1. `[virtchan5]`
2. `name=FMODE0`
3. `template=complex`
4. `[mixer]`
5. `src=Virt5`

## Writing a deviation Tx model

```
6. dest=Virt5
7. scalar=0
8.
9. [virtchan6]
10.name=FMODE1
11.template=complex
12.[mixer]
13.src=Virt5
14.dest=Virt6
15.scalar=-1
16.offset=1
17.usetrim=0
18.curvetype=absval
19.points=0
20.
21.[virtchan7]
22.name=FMODE2
23.template=complex
24.[mixer]
25.src=Virt5
26.dest=Virt7
27.scalar=-1
28.usetrim=0
29.
30.[trim6]
31.src=Virt5
32.pos=TRIM_L+
33.neg=TRIM_L-
34.step=190
35.value=-100,0,0
```