Er9x Manual

2015 Edition

Because it was needed!!

This manual covers Er9x firmware used in the 9x and 9XR transmitters

The latest version of this manual is available at:
http://openrcforums.com/forum/viewtopic.php?f=5&t=6473

ER9x_Manual_2015-v01.odt
2015-02-22

Disclaimer

THIS FIRMWARE IS PROVIDED ON AN "AS-IS" BASIS WITHOUT WARRANTY OF ANY KIND AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE DEVELOPER AND/OR AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PERSONAL AND/OR PROPERTY DAMAGE) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS FIRMWARE, EVEN IF THE DEVELOPER AND/OR AUTHOR HAS BEEN ADVISED BY USER OF THE POSSIBILITY OF SUCH POTENTIAL LOSS OR DAMAGE. USER AGREES TO HOLD THE DEVELOPER AND/OR AUTHOR HARMLESS FROM AND AGAINST ANY AND ALL CLAIMS, LOSSES, LIABILITIES AND EXPENSES.

Yeah, and no spitting either! 🙂 Erazz

Notice

Er9x and eePe are free open source software, independently developed. This manual is provided to help you understand and use them.

You can redistribute the software and/or modify it under the terms of the GNU General Public License version 2 as published by the Free Software Foundation. You can also freely quote from, redistribute and/or modify this manual; in doing so please acknowledge the authors. The program and the manuals are distributed in the hope that they will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. In using them, you agree that you accept all responsibility.

For more information go to: http://openrcforums.com/forum/viewforum.php?f=5

Table of Contents

Foreword

From Erazz

Erazz was the original developer of the Er9x firmware. So we should give him the first words in this new version of the manual. Here's the introduction from his 2010 manual:

The IMAX/FLYSKY/TURNIGY/EURGLE 9x is a computerized radio from China. The transmitter is outfitted with a 128x64 pixel monochrome LCD, two 2-axis gimbals, three variable potentiometers (pots), six 2-position switches, one 3-position switch and some funky red trim.

The big thing about this transmitter is the price. At the time of writing radios may cost anywhere from $180 for simple units up to $1000+ for super blinged out bazillion channel super heavyweight monsters. This one costs $60.[1]

Where's the catch then? (you might ask). The catch is in the firmware – FW. The original FW is less-than-perfect. It has bugs, funky navigation and the most annoying beeping I have ever heard. Not good.

However, some neat guy called Thus figured out that the whole radio is made from gimbals, switches, funky trim and a very generic main processing unit that does absolutely everything. He had one of those ah-ha moments where insanity overrides common-sense and decided to completely rewrite the original software and replace it with his own.

At some point I decided that while Thus' FW is great I really wanted more bling for my TX. So in the spirit of open-source, the Thus source was stolen and Er9x was born. (Yeah, I was vain. ER are my initials.)

You might want to check out Thus' code. While you're at it check out RadioClone's code – which also runs on the 9x. While Er9x is based on Thus' code, I have managed to pilfer some of RC's excellent code. His FW is more complex but also more powerful.[2]

Got you all worked up about this? Good. Go now to the kitchen, make yourself a nice cup of coffee. This is a long read. I'll wait here till you're ready. Promise!

Erazz

[1] At one point, the price of the 9x dropped to under $40. Now (2015) it's typically around $70.

[2] Thus' code can be found at http://code.google.com/p/th9x/, while Radio Clone's is at http://radioclone.org/.

From João and Nigel

So Erazz got all this started the way we have it today. Then Mike came along and he's been doing the development ever since. Mike developed and implemented the many advanced features we now have available in Er9x, and he continues to improve the firmware constantly, documenting the changes as he goes.

But over the past couple of years, the manual which Erazz gave us, and which is supposed to bring together everything you need to use Er9x, became seriously out of date. So we've decided to spend some days, eerrr... weeks, eerrrrrr... months, updating the manual, with crucial input from Mike.

Our aim is to provide documentation that reflects the latest officially released version of Er9x. As new versions are released, we plan to update the manual and/or provide supplements. So if you update your firmware, be sure to check for new documentation here.

Because the firmware continues to evolve, if you are using an older version of Er9x you will find that some features mentioned in this manual are not available to you. You will also find that the structure of menus has been revised in recent versions. We don’t think these changes should present a problem for most users, but for easiest and best results we would recommend updating your firmware.

In addition to the official releases, Mike provides test versions of Er9x at this location. Those of you who want to keep up with the latest developments should go there, but be aware that those versions may still have issues.

The plan of this manual is very simple in principle (though a bit more complicated in reality). First, in the Introduction we explain some basic things you need to know about the transmitter and the firmware, including some background on the 9x transmitter and how it has evolved, an introduction to the various versions of the firmware and how Er9x works, and an overview of how to use the firmware. Finally, the Introduction explains the main screens that show you what's going on in the transmitter and can be used to display telemetry data sent back from the model.

Then we go screen by screen and menu by menu through the many settings and controls that give Er9x its amazing power and flexibility. The main section called Radio Setup deals with the global settings that determine how the transmitter works and that affect all the models you program into it. Next is a short section on creating and selecting among the individual model definitions on your transmitter.

The section called Model Setup is the big one. It allows you to program individually each model that you fly (or drive) with this transmitter; in some cases there are options that, for one model, override the global Radio settings.

Finally, in the Annexes we offer examples of programming, notes on the hardware modifications that have been developed to take advantage of Er9x, and a glossary of terms.

Along the way, we've tried to keep at least some of the inimitable humour Erazz brought to the original manual. A bit of humour is healthy and helps to relax while reading. 🙂

We do hope this work helps people enjoy using this great piece of technology in their RC hobby. Er9x demands some effort on the part of its users, but we believe the results are more than worth it.

João and Nigel

Introduction

The 9x Transmitter and Er9x

In the beginning, around 2007, the 9x transmitter was released, a nine-channel radio with an FM module and a long telescoping antenna made by an outfit in China called FlySky RC. It operated on 35 MHz, 40 MHz or 72 MHz, depending on the country. It included firmware that was relatively powerful (but rather buggy) and it could store eight models in memory. Though manufactured by FlySky it was also sold under a variety of other brand names (Eurgle, Imax, etc.). It was great value at well under $100.

Then came 2.4 GHz! By about 2009, modules were available that could be plugged into the 9x to enable it to use the new RF system. As well, HobbyKing had adopted the 9x and branded it Turnigy. The firmware was overhauled, eliminating most of the bugs, and called V2. Originally, the 9x was sold without a module and people installed their own, but by 2010 the Turnigy version included a FlySky 2.4 GHz module and the old FM telescoping antenna was eliminated. It's interesting to note that the standard PPM protocol used with the 2.4 module was limited to eight channels. So that's why, despite the 9x name, in its usual form the transmitter provides only eight channels.

Meanwhile, as you have read above, Erazz, building on the work of others, was hard at work giving the 9x a brain transplant called Er9x. This new open source firmware transformed the transmitter from a cheap and fairly capable controller into a still cheap but now amazingly flexible and powerful radio, taking full advantage of the new 2.4 GHz technology. In 2012, Erazz handed over care and development of the Er9x firmware to Mike, who took up the challenge and further improved it.

Along the way, modules and receivers from FrSky RC (never to be called “FrySky” and not to be confused with FlySky RC, who make the 9x transmitter) gave the 9x outstanding range and reliability. Subsequently, hardware modifications such as the “telemetry mod”, to allow the radio to receive the FrSky telemetry data, as well as haptic feedback (vibration) and a voice module, were developed for the radio by members of the open source community.

One thing that stood in the way of upgrading (flashing) the stock firmware to Er9x was the requirement to install a programmer, which took some delicate soldering on a very small scale. The problem was fixed when the SmartieParts programming board provided a no-solder solution.

By 2012, the 9x was fully developed and in wide use. It was joined by the Turnigy 9XR, which was essentially a reconfigured 9x in a new case.

Despite the arrival of other radios since then, such as the 9XR Pro and Taranis, and the development of the alternative OpenTX firmware as a fork from Er9x, the 9x transmitter with Er9x firmware remains a favourite with many flyers around the world and continues to lead the way in improvements, thanks to Mike's continuing work.

Hardware Variants and Modifications

Apart from the transition from an FM to 2.4 module and the resulting loss of the original telescoping antenna (still visible in the picture on the cover), the 9x has remained virtually unchanged over the years. Currently, there are two main versions, one is black and sold under the name of FlySky and the other is grey and sold by HobbyKing under the Turnigy label. For all practical purposes they are now identical.

Both the Turnigy and the FlySky versions normally include a FlySky/Turnigy 2.4 module permanently wired to an antenna mounted on the top of the transmitter. Note that changing to another module such as the OrangeRX DSM2/DSMX or one of the FrSky modules requires removal of the stock wired-in module. The module required uses the JR form factor.

There may also be versions of the 9x sold by other distributors under a different label, and these may or may not include a module.

Note that receivers from one manufacturer are generally not compatible with modules from another manufacturer, unless specifically designed for compatibility.

In its stock form, the 9x uses an ATmega64 chip (m64) as the central processing unit. This works well but imposes significant limits on both processing and storage. Consequently, some users have replaced the processor of the 9x radio with an ATmega128 or ATmega2561. The former chip doubles the flash and eeprom memory, while the latter also doubles RAM.

Some of the features described in this manual may not be available for the 9x radio with the original ATmega64 processor (though as Mike continues to free up memory by improving the code, more and more features are fitted into the m64 version). Also there are some features like telemetry, voice, haptic (vibration), rotary encoder, etc., that require physical modifications, and/or the addition of new hardware to the TX. Some of these also require the selection of a particular version of the firmware. A brief description of various modifications is provided in Annex B.

In late 2012, HobbyKing released the Turnigy 9XR transmitter. This is essentially a physically redesigned transmitter that is in most respects fully compatible with the 9x. It includes a connector that allows an inexpensive programmer to be plugged in for communication with a computer.

The 9XR comes with a version of Er9x modified by HobbyKing but it can also run the real Er9x firmware. Indeed, we strongly recommended that the stock firmware be replaced because it is not compatible with the *eePe* program (see page 5) and does not to allow backup of firmware or models to a computer. While early versions of the 9XR used the ATmega64 processor, since about spring 2013 the 9XR has come equipped with the ATmega128.

In 2014 the upgraded 9XR Pro was introduced, and in early 2015 it was announced that the 9XR was being discontinued. The Pro has many additional hardware features including a much more powerful processor, voice and haptic functions, and a built-in programming capability. It runs a more advanced version of the firmware called Ersky9x, many of whose features are similar to those in Er9x described in this manual.

Versions of the Firmware

Factors involved in choosing the right version of Er9x for your 9x or 9XR transmitter include:

- Processor: Stock ATmega64 vs. upgraded chips
- Telemetry: whether the Tx has been modified to display FrSky or Jeti telemetry on screen
- Language: Besides English, some versions of Er9x are available in German and Norwegian[3]

The “FrSky” modification re-routes two of the switches, THR and AIL, and allows the ATmega chip to receive telemetry data on the pins where these switches were formerly connected. Depending on

[3] If you'd like to see a version in your own language, ask about creating a language file for it, and offer to do the translation. Everything that's available is done by volunteers.

which processor your radio has on its mainboard, there are a couple of different firmware options associated with this modification.

The currently available Er9x firmware versions are shown in the table below.

Filename	Processor	Templates	Telemetry	Language	Other
Er9x.hex	ATMega64	Yes	No	English	
Er9x-no.hex	ATMega64	Yes	No	Norwegian	
Er9x-de.hex	ATMega64	Yes	No	German	
Er9x-nmea.hex	ATMega64	Yes	FrSky	English	Supports NMEA GPS
Er9x-ardupilot.hex	ATMega64	Yes	FrSky	English	Works with Ardupilot
Er9x-frsky.hex	ATMega64	No	FrSky	English	
Er9x-jeti-hex	ATMega64	No	Jeti	English	
Er9x-128.hex	ATMega128	Yes	FrSky*	English	
Er9x-128-de.hex	ATMega128	Yes	FrSky*	German	
Er9x-2561.hex	ATMega2561	Yes	FrSky*	English	

*These versions of the firmware include an “FrSky mod done” choice in *Radio Setup/Settings/Hardware*.

For the latest release of Er9x, go here. But the the best way to install and manage Er9x firmware is with the eePe program described in the next section.

The eePe Program

The *eePe* program is designed to run on a Windows computer (Macintosh and Linux versions are also available but may not always be up to date).[4] Your 9x/9XR programmer is used by *eePe* to communicate with the transmitter. It manages the process of flashing the transmitter to Er9x firmware and keeping the firmware updated. It also enables you to back up the EEPROM file, single models, and firmware on your computer.

In addition, *eePe* allows you to edit both the general settings of the transmitter and the individual model settings and programming. You can download the EEPROM file from the transmitter to *eePe* and save it to the computer hard drive as an eePe file. You can use *eePe* to edit the individual models within the file and test your changes on the simulator built into *eePe*. When satisfied, you can upload the amended file back to the transmitter. You can also store individual models on the computer as eepm files and copy them into the eePe file at a later date.

In a new development, radios with a voice module (MegaSound9x) installed can now exchange single models between the Tx memory and the voice module's SD Card using the Backup/Restore options. It can also store them to and retrieve them from your computer using *eePe.*

With *eePe,* you can start from scratch to build a new model definition, or to obtain a file on line or from a friend. These models can then be copied to your transmitter by way of the .eepe or .eepm files.

You can download *eePe* here.

[4] Mike relies on other people to compile versions other than Windows. Help would be very much appreciated.

Getting Help on Internet Forums

Help is always just a few clicks away on the internet forums where experienced Er9x users volunteer their knowledge and experience.

Open RC Forums: http://openrcforums.com/forum/index.php

Open RC Forums, Er9x index page: http://openrcforums.com/forum/viewforum.php?f=5

Open RC Forums, 9x transmitter hardware page: http://openrcforums.com/forum/viewforum.php?f=71

Open RC Forums, 9XR transmitter hardware page: http://openrcforums.com/forum/viewforum.php?f=99

RC Groups, 9x transmitter: http://www.rcgroups.com/forums/showthread.php?t=1266162

RC Groups, 9XR transmitter: http://www.rcgroups.com/forums/showthread.php?t=1628785

Supporting Development

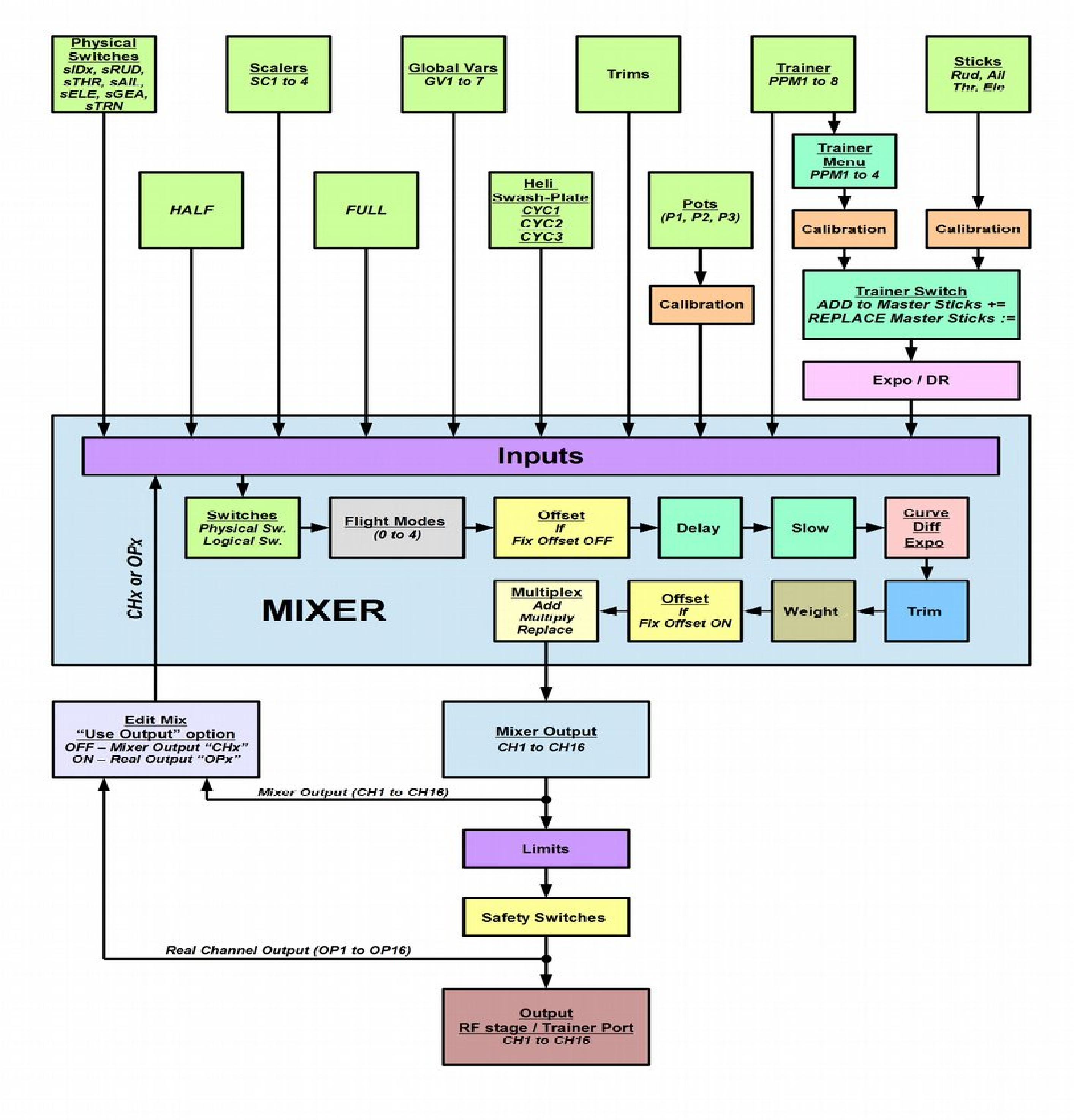
Many people have contributed to the development of Er9x and eePe, a process that is still actively continuing. They don't do it for monetary gain but for the inherent satisfaction of creating something that they and others find useful. That's the spirit of open source software. In other words, they do it for fun. All they expect is that their efforts be appreciated.

Two people deserve our special thanks. The first, of course, is Erazz without whose efforts there simply would not have been Er9x or eePe, and the 9x transmitter might have remained just another cheap radio. Instead it became a demonstration of the astonishing power and flexibility of intelligently designed firmware.

In 2012, Erazz handed the torch to Mike Blandford, who has not only maintained Er9x and eePe but has created a whole set of major improvements and innovations, as reflected in this manual. In addition, Mike has developed Ersky9x, the firmware that is used by the new generation 9XR Pro transmitter and that also supports Taranis. So give a little thought to making a donation to Mike for all he has done, and continues to do, to enhance your enjoyment of our hobby. Donate to Mike here.

To help you get the most from Mike's work, João, with help from Nigel, has taken on responsibility for updating and greatly strengthening the documentation of Er9x. So if you find this manual useful, why not buy them a beer to show it. Donate to the Documentation Team here.

How Er9x Works



As can be seen in the top line of the flow chart, the system receives four types of physical input:

1. Sticks
2. Potentiometers
3. Trims
4. Switches

The analog inputs (sticks and pots) go through a calibration phase. The sticks can also go through Expo and DR (Dual Rate) filters before going to the Mixer.

The Mixer does it all. It processes the inputs by adding them to mixes, scaling and/or offsetting them, applying curves, controlling the mix timing and determining how the mix will interact with other mixes in the same channel. Finally it directs them to the desired Mixer output channel (CH1..CH16). From here the channels can be sent for processing by the Limits and Safety Switches menus, or can be re-directed back into the Mixer as inputs.

After being processed by the Mixer, the individual channels are processed in the Limits menu, ensuring that no output goes too far (possibly damaging servos or control surfaces), as well as taking care of the servo centring (Subtrim). After the limits have been applied, the processing stage is now complete and the channels are now ready to be sent over the air to your model. At this point, if desired, a safety switch can be programmed to lock the output of one or more channels to fixed values. Good examples are Throttle-Cut and Sticky Throttle-Cut.

Finally the output channels can be re-directed back into the Mixer as inputs before they are encoded and sent to the RF module to take that nice little hike through the air to your model. Or they can be sent to the trainer port to serve in a buddy box system linked to another transmitter.

Here is a list showing what happens from the time you move a stick, pot, etc., until your command is sent to the model. Please refer to the diagram on the previous page.

The order of processing of an input is as follows:

1. Obtain the source value, with expo/dual rates applied to sticks.
2. Apply switches or Flight Modes.
3. Add Offset (if "Fix Offset" is not enabled – see page 32).
4. Apply Delay and Slow.
5. Apply Curve/Differential/Expo.
6. Apply Weight.
7. Add Offset (if "Fix Offset" is enabled – recommended).

After all mixes:

8. Apply Subtrim and Limits.
9. Apply Safety Switches.

Working with Er9x

Nomenclature

(Just so we understand each other)

Inputs

Rud – Rudder

Ele – Elevator

Thr – Throttle

Ail – Aileron

P1/P2/P3 – Potentiometers (Pots) – see Layout picture for identification

Switches:[5]

THR – Throttle cut switch. Don't confuse it with “Thr”, meaning the throttle stick. The THR switch is located on the top of the transmitter towards the back left side

RUD – Rudder DR switch

ELE – Elevator DR switch

ID0, ID1, ID2 – Three position switch. ID0 – top, ID1 – mid, ID2 – bottom

AIL – Aileron DR switch

GEA – Gear switch

TRN – Trainer switch. This switch is spring loaded (momentary)

L1 to **L9** and **LA** to **LI** - Logical switches. More on these later.

Every function in this FW is assignable. There are no fixed switch assignments. You can, for example, choose the “TRN” switch to be throttle kill and use the three-position switch to control triple rates. The names are useful, however, since the switches are labelled like that on the Tx.

The “!” sign. Whenever you see “!” before a switch you can read that as “not” or “inverted”. Switches can be normal or inverted. So when choosing the elevator D/R switch “ELE” is normal and “!ELE” denotes inverted (reversed) operation.

Edit and Navigation Buttons

There are six edit buttons on the lower part of the Tx. In this manual they are noted with square brackets, thus [MENU]. Some functions need the button to be pressed and held for a second or so. They are noted as “long” presses like so: [MENU LONG]. Likewise, the [EXIT] button can be pressed short or long.

The four navigation/edit keys are arranged in a cross formation on the left side of the transmitter. [UP] and [DOWN] are clear enough, but the arrangement of the [+] and [–] keys on the 9x is reversed from what one might expect, so in this manual we will refer to these keys as [LEFT] and [RIGHT] to avoid confusion.

As a general rule the [UP]/[DOWN]/[LEFT]/[RIGHT] keys move the cursor appropriately The [MENU] key is used for selection and for editing. The [EXIT] key is used for exiting (surprise). ☺

[5] Physical switches are considered to be OFF when set towards the TOP or BACK of the radio.

Pressing [EXIT] will generally stop the editing of a field or go back to the previous menu. Pressing [EXIT LONG] will exit immediately to the last main screen.

Pressing [MENU LONG] from any main screen will open a pop-up window with the following options:

Model Select

Model Setup

Last Menu

Radio Setup

Statistics

Pressing [MENU] from a telemetry screen will display the following options (see page 13):

Zero Alt.

Zero A1 Offs

Zero A2 Offs

Reset GPS

From any main screen you can press [RIGHT LONG] to enter the Model Select menu. There you can use the [UP/DOWN] keys to choose a model or empty slot and select it by pressing [MENU].

Pressing [LEFT LONG] will enter the *Radio Setup* menu. Once there you can navigate to the required sub-menu using the [LEFT/RIGHT] and [UP/DOWN] keys, and select it by pressing [MENU].

Editing and Saving

As a rule, once a value is changed it is saved. You can turn off your Tx and turn it back on and the values will be saved. The values are saved internally in the processor's EEPROM. However there is a slight delay sometimes, so it's probably a good idea to wait a couple of seconds before turning off. There is no undo functionality. Once something is erased/changed, it's changed for good.

Generally, when a value is highlighted and you cannot move left or right, then pressing [LEFT]/[RIGHT] will change that value.

When moving left or right is possible, you need to press [MENU] to edit that value. Edit-mode is normally displayed by the cursor blinking. Press [LEFT]/[RIGHT] to edit the value in steps of 1. To edit the value in steps of 20, highlight the value then press and hold [MENU] while editing. To exit edit-mode press either [MENU] or [EXIT].

When setting the time value in a Timer, you can use the same procedure and it will change in steps of 1 minute instead of 1 second.

When editing, simultaneously pressing both [LEFT] and [RIGHT] keys inverts the value. Try it, it's cool! (It works less well with the 9XR, which uses a physically different type of directional switch.)

Also, it is possible to navigate and change values with the pots by turning ON Pot Scroll in the *Radio Setup* menu. Pot controls work as follows (and there's also Stick Scroll, which is broadly similar):

P1: Navigate horizontally

P2: Navigate vertically (line change)

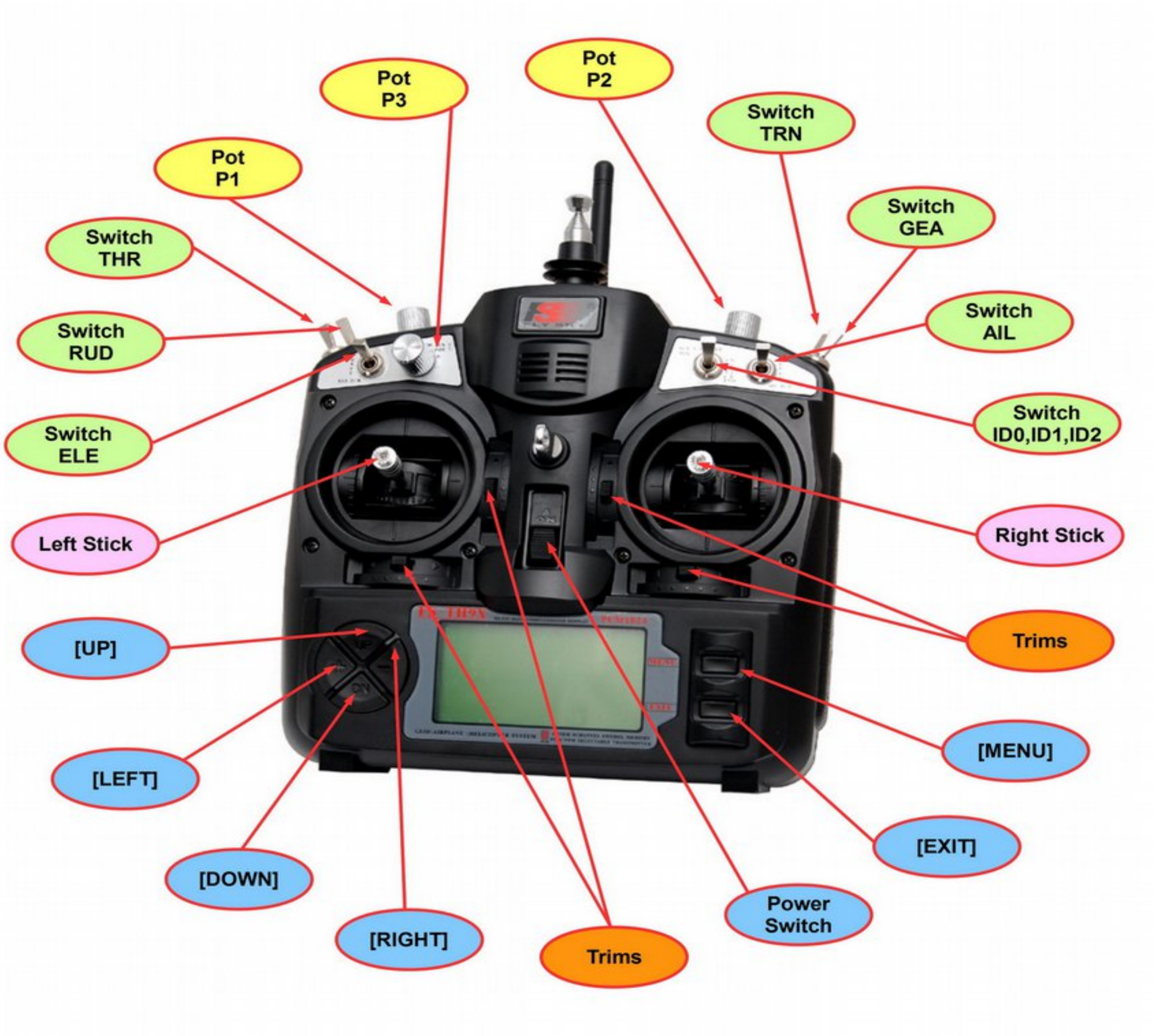
P3: Change values, and in some situations navigate through the options horizontally

Startup – Quick Model Select

On startup holding any of the following keys will quickly load an associated model memory:

[**MENU**]	Model Memory **#1**
[**EXIT**]	Model Memory **#2**
[**DOWN**]	Model Memory **#3**
[**UP**]	Model Memory **#4**
[**RIGHT**]	Model Memory **#5**
[**LEFT**]	Model Memory **#6**

Transmitter Layout



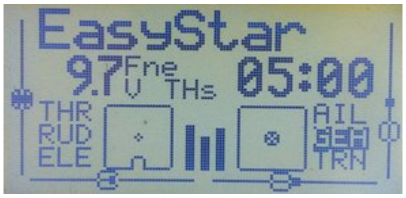
Main Screens

Four of the main screens show the same basic information in the top part of the screen: model name, transmitter battery voltage, timer value, timer switch identity, and trim increment setting. Trim position graphics are shown on the sides and bottom of the screen.

In addition, all versions of Er9x except the standard (non-telemetry) version include Telemetry screens to display data returned from the model.

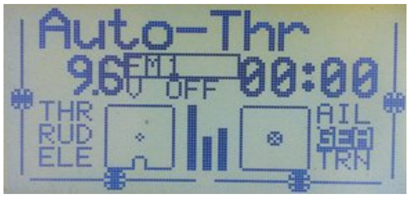
Stick, Pot and Switch Positions Screen

In this main screen, the lower half of the screen shows a graphical representation of stick and potentiometer positions, as well as a list of the physical switches and also all the logical switches (see page 45). When a physical or logical switch is on, it will be highlighted.



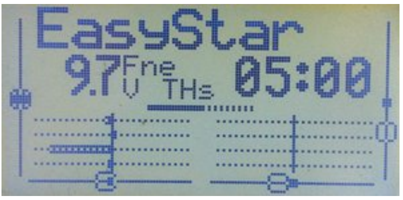
Press [RIGHT] or [LEFT] to cycle through pages showing all the switches and logical switches.

In the upper half of the screen, the transmitter voltage is displayed in large print, along with the trim increment type, here 'Fne' (see page 68). The current value of the Timer is also shown in large print, along with the trigger used to activate it, here ‘THs’ (see page 64). In addition, the active Flight Mode will be displayed if it is other than the default FM0, replacing the trim increment type on the screen, as shown in the picture at right.



Graphic Servo Outputs Screen

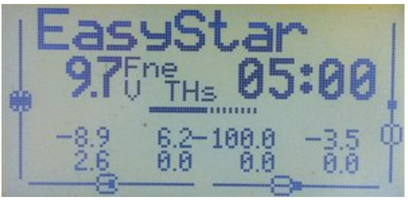
In this view (short press down from the previous screen), the upper part of the screen is identical to the previous screen, while the lower half of the screen shows the channel outputs in bar graph form, reflecting the results of programming for the particular model.



Just above the channel output graphic is a bar showing which of the 16 channels are being displayed (not available on transmitters with ATmega64 processors, as they can only display the first eight channels). A short press of the [RIGHT] or [LEFT] key changes the channel view from the first set of 8 channels (1-8) to the second set (9-16). With the necessary hardware, the transmitter is capable of controlling 16 channels.

Numeric Servo Outputs Screen

In this main screen the lower half shows the same channel outputs as the Graphic Servo Outputs screen but in digital form. The upper part of the screen is identical to the previous screens.

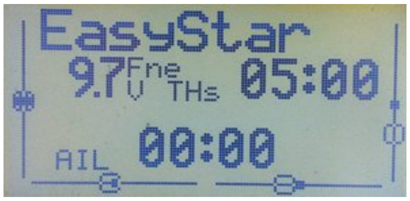


Again, for transmitters using a processor other than the ATmega64, the bar shows which channels are being displayed (here 1-8). A short press of the [RIGHT] or [LEFT] key changes the view between channels 1-8 and channels 9-16.

Warning: It’s easy to give a short [RIGHT/LEFT] press inadvertently and display channels not being used on your transmitter.

Timer 2 Screen

In this main screen the upper half shows the usual information, including Timer 1. The lower half displays Timer 2, and to the left the trigger used to activate it, AIL in the picture. A long press of the [EXIT] key will reset both timers.

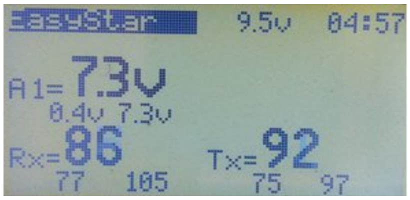


```
EasyStar
9.7Fne    05:00
   V THs
AIL  00:00
```

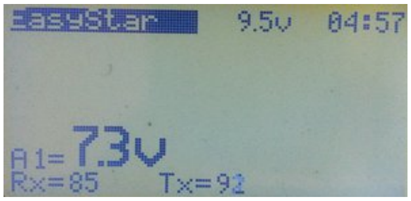
For more information on Timers, see page 64.

Telemetry Screens

These screens are only available with the telemetry versions of the Er9x firmware.



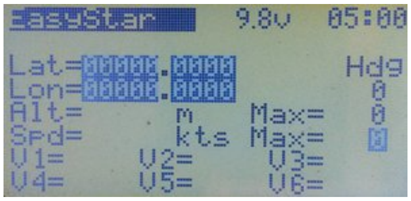
```
EasyStar   9.5v  04:57
A1=7.3v
  0.4v 7.3v
Rx=86      Tx=92
  77  105    75  97
```



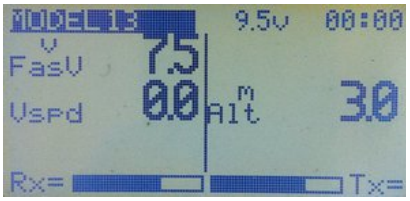
```
EasyStar   9.5v  04:57


A1=7.3v
Rx=85     Tx=92
```

To get to the telemetry screens, you can use short presses of [UP] or [DOWN] or press [DOWN LONG] from any main screen. Use the [RIGHT] or [LEFT] keys to cycle through the four telemetry screens. These include a custom screen which you can adapt to suit your requirements. Consequently, the information displayed may vary from what is shown in the picture.

The telemetry data displayed may include Global Positioning System (GPS) data, voltage, temperature, rate of climb, battery capacity used or other information, depending on what sensors you have installed in your model.



```
EasyStar   9.8v  05:00
Lat=00000.0000    Hdg
Lon=00000.0000      0
Alt=     m   Max=   0
Spd=     kts Max=   0
V1=     V2=     V3=
V4=     V5=     V6=
```



```
MODEL 13     9.5v  00:00
  v
FasV     7.5
            m
Vspd     0.0Alt     3.0

Rx=                  Tx=
```

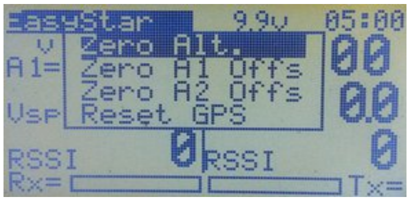
Pressing [MENU] from a telemetry screen will open a pop-up window with the following options:

Zero Alt. - Set altitude to 0.

Zero A1 Offs - If A1 is measuring current, use the present value as an offset, thus setting the displayed value to zero.

Zero A2 Offs - The same as for A1.

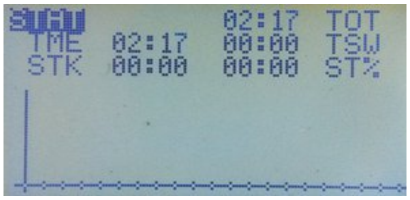
Reset GPS - Reset GPS data.



```
EasyStar    9.9v   05:00
  v Zero Alt.
A1= Zero A1 Offs     00
    Zero A2 Offs
Vsp Reset GPS        0.0
RSSI        0RSSI      0
Rx=                   Tx=
```

To reset a value, scroll down to the field and press [MENU].

Statistics Screens

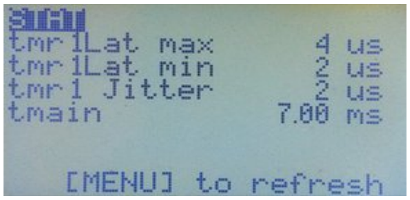


```
STAT            02:17 TOT
 TME  02:17     00:00 TSW
 STK  00:00     00:00 ST%
```

From any of the main screens pressing [UP LONG] will access the two statistics screens. Pressing the [LEFT]/[RIGHT] keys will cycle between them. Pressing [EXIT] will bring you back to the main screens.

On the first STAT screen, TOT shows the total time the Tx has been switched on since last power on. The other values relate to Timer 1 and depend on the trigger(s) used, e.g., STK shows the total time the throttle stick has been above -100%.

The graphic below traces the throttle and it is not available for radios with the ATmega64 chip.

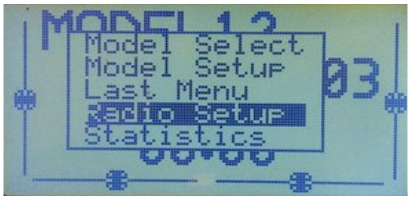


```
STAT
tmr1Lat max        4 us
tmr1Lat min        2 us
tmr1 Jitter        2 us
tmain           7.00 ms

   [MENU] to refresh
```

On the second STAT screen, tmr1LAT max, tmr1LAT min and tmr1 Jitter, relate to the LATency of the code that drives the PPM/PXX signal.

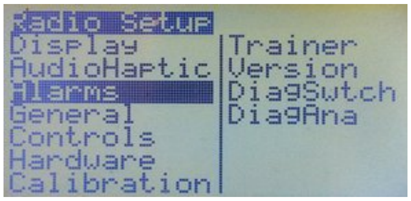
As this is software driven, there is some small jitter compared to the exact time at which the signal should change. These values indicate the amount. As long as the Jitter value is below 5us (μsec), all is OK. You can refresh the values by pressing [MENU].

Radio Setup



This collection of menus determines the various parameters that apply globally to the radio. Settings that apply to individual models are dealt with under *Model Setup* (starting on page 28).

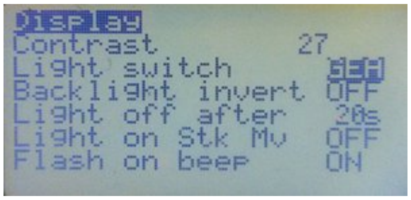
To get to *Radio Setup* from any of the five main screens (telemetry screen included) press [MENU LONG]. This calls up a pop-up menu with several options, as shown at right.



Use the [UP/DOWN] keys to navigate to *Radio Setup* and press [MENU] to enter the *Radio Setu* index menu. Use the directional buttons to highlight the required menu, and press [MENU] to select it.

[EXIT] returns to the main screen.

Display


Display
Contrast 27
Light switch GEA
Backlight invert OFF
Light off after 20s
Light on Stk Mv OFF
Flash on beep ON

Contrast

The contrast of the LCD display screen. The values can be 20..45. The higher the value, the darker the screen.

Light Switch

The display backlight can be set to turn ON via any of the transmitter physical switches (RUD, ID0, ELE, etc.) or any of the Logical Switches (L1, L2,...LA, LB, etc.). It can also be set to full-time ON or OFF by this menu.

Backlight Invert

This is a special option to invert the backlight output, needed ONLY for the rev. 2.2 SmartieParts programming board in the 9x transmitter.

Light Off After

When this is not set to OFF, any key-press will turn on the backlight, which will turn off again after the specified number of seconds. The possible duration ranges from 5 to 600 seconds.

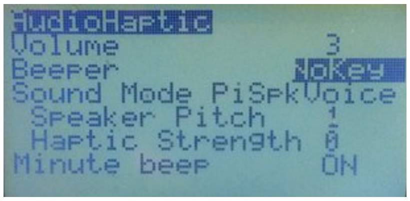
Light on Stick Movement

When this is not set to OFF, movement of the sticks will turn on the backlight; it will turn off after the specified number of seconds. The possible duration ranges from 5 to 600 seconds.

Flash on Beep

Flashes the backlight when some hard coded alarms and warnings sound. For example the timers minute warning, or the radio battery is low.

AudioHaptic



```
AudioHaptic
Volume              3
Beeper          NoKey
Sound Mode PiSpkVoice
 Speaker Pitch      1
 Haptic Strength 0
Minute beep        ON
```

(Some options require a voice module)

Volume

The volume setting for an add-on voice module.

Beeper

(Sets beeping levels)

i. Quiet: No beeping at all. No warning – nada. If the kids are sleeping and you must setup the model in your living room this is the mode to use. Just remember that the Tx will not even warn you when the battery is low. If you're using a LiPo battery to power the TX, watch out! 😉

ii. No keys: The beeps are normal but edit keys are silent.

iii. Normal: Normal beeping.

iv. Long beeps: For those who want to annoy other people.

v. Extra long beeps.... You gotta be joking, right?? 🙂

Sound Mode

(Set to match available stock beeper, piezo speaker and/or voice module)

i. Beeper: The standard beeper.

ii. PiSpkr: Assumes you have the piezo speaker fitted, works OK with stock beeper.

iii. BeeprVoice: Standard beeper with added voice module.

iv. PiSpkVoice: Piezo speaker with the added voice module.

v. MegaSound: This option must be selected to allow model backup/restore to/from the voice module's SD Card to work. The extra backup and restore options will then be available in the Model Select pop-up menu. The voice module must also have an extra serial connection to the radio's main board for this feature to work. For more information see page 90.

Speaker Pitch

Values from 1 to 100 may be set. The pitch varies one tone up for every five value increments.

Haptic Strength

(*Haptic modification needed)*

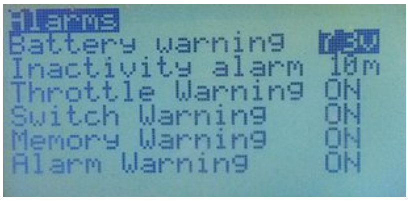
Values from 0 to 5 may be set. This parameter determines the strength of the vibration.

Minute Beep

Beeps every full minute while the timer is running. If a voice module is installed, this activates a voice announcement each minute.

Alarms

Battery Warning



```
Alarms
Battery warning    7.3v
Inactivity alarm   10m
Throttle Warning   ON
Switch Warning     ON
Memory Warning     ON
Alarm Warning      ON
```

When the voltage of the transmitter battery drops below this level the transmitter will warn you to land soon, either with beeps or by speaking (if you have a voice module).

How long the Tx will continue to function depends on the battery and setting. Don't delay your landing as even the best efforts of the developers could not fix the “zero voltage non-functionality” problem of the original firmware. ☺

Inactivity Alarm

This will set up a warning that will play an audio alarm if the Tx is left unattended for the specified amount of time. The default value is 10 minutes. To turn off the Inactivity Alarm, set the value to zero. Values can be from 1 to 250 minutes. To reset the timer (and stop the alarm) simply move one of the sticks. When the Tx is running on USB power, the alarm is inactive.

Throttle Warning

If this is ON and the throttle is not at idle, the Tx will show a warning when it is turned on or the active model is changed. To clear the alert, either press a key or bring the throttle stick to idle. The Tx will not output a signal until the alert is cleared.

Switch Warning

If this is ON when the Tx is turned on, or the active model is changed, the Tx will show a warning if one or more switches are not at their default positions. To clear the alert, either press a key or ensure that all switches are in their default positions. The Tx will not output a signal until the alert is cleared.

Memory Warning

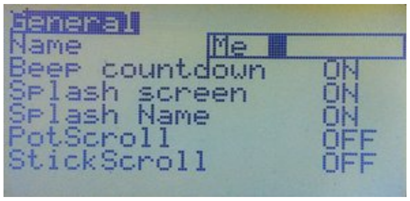
If this is ON, the Tx will show a warning when turned on whenever the available eeprom memory is less than 200 bytes. The Tx will not output a signal until the alert is cleared.

Alarm Warning

This will give you a “heads up” if your beeper is silent. If it is ON and the beeper option in the “AudioHaptic” menu is set to “Quiet”, you will receive an “Alarms Disabled” warning on startup. This feature was added after a programming session left a user flying on silent. It's really useful!

Note: This will not affect voice alarms, only the beeper signal. For example, the sound you hear when operating the trims.

General



```
General
Name            Me
Beep countdown    ON
Splash screen     ON
Splash Name       ON
PotScroll         OFF
StickScroll       OFF
```

Name

The radio owner's name. You... Unless... (hmmm...) This may be displayed on the splash screen.

Beep Countdown

If this is ON, beeps will sound at 30, 20, 10, 5, 4, 3, 2 and 1 seconds before the timer ends. If a voice module is installed, the radio will speak the remaining time.

Splash Screen

When this option is ON, the splash screen is shown when the radio is turned on. The screen can be skipped by pressing any key or moving a stick.

Splash Name

If this is ON, the name set in *Owner Name* above will be shown on the splash screen.

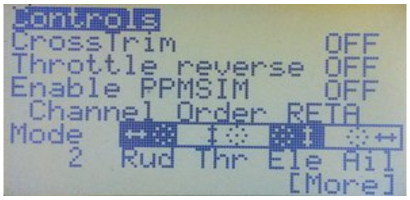
PotScroll

If this is ON, pots P1, P2 and P3 can be used to move between, and edit, menu items.

StickScroll

If this is ON, one of the sticks can be used to move between, and edit, menu items. Which stick is used depends on the selected mode (see below) – for Mode 1 it's the left stick and for Mode 2 the right. Vertical stick movement normally moves between items and sideways movement edits them.

Controls



```
Controls
CrossTrim           OFF
Throttle reverse    OFF
Enable PPMSIM       OFF
 Channel Order RETA
Mode
  2    Rud  Thr  Ele  Ail
                   [More]
```

Cross Trim

Cross Trim swaps the trims so the switches on the left apply to the stick on the right, and those on the right apply to the stick on the left. This can make trim adjustment in flight easier, particularly for Mode 2 fliers, as it allows them to adjust aileron and elevator trim with the left hand while flying with the right. (For another way to assist in the trimming of a new model, see “Trim Switch” under *Model Setup* (page 68).

Throttle Reverse

For the great majority of users, normal throttle operation for powered aircraft has closed throttle (idle) with the stick DOWN (towards the operator when the Tx is in the usual horizontal position) and full throttle with the stick UP (away from the operator). This is the default throttle arrangement for Er9x.

The Throttle Reverse option inverts the operation of the throttle stick. When Throttle Reverse is turned ON, closed throttle is UP (away from the operator) and full throttle is DOWN (towards the operator). Throttle Reverse also inverts the throttle warning on start-up (requiring the throttle stick to be in the UP position) and some other throttle-related functions. This is the preferred direction for some users, particularly when the throttle stick is used for a function such as sail control on a boat, or to emulate the collective lever of a full scale helicopter.

Throttle Reverse here in the *Radio Setup* menu operates globally and affects all models. A similar feature is now also available in the *Model Setup* menu for each individual model. Hence the selection made here can be reversed on a model-by-model basis if required (see page 67).

Note that this option reverses the stick action and is not to be used as a means of reversing the output signal to a throttle servo or electronic speed control (that should be done in the Limits menu).

This feature is for all you wacky people who fly with the throttle backwards. Though I (Erazz) personally don't understand how you fly like that, it's nice to have a choice. 🙂

Warning: Use this feature with extreme caution, especially when dealing with electric models.

Except when a special throttle configuration is required, both global and individual model Throttle Reverse options should be left at OFF.

Enable PPMSIM

When set to ON, this option enables the use of the PPMSIM protocol. This may be required when a 9x transmitter with a FlySky/Turnigy module installed is used with a simulator or as the Slave (buddy box) in a trainer setup. See page 22.

PPMSIM corrects a problem which occurs because the stock FlySky/Turnigy RF module for the 9x transmitter excessively loads down the PPMout signal. This prevents proper operation of the Tx with a simulator or in "buddy box" mode. Before PPMSIM was introduced, it was necessary in such cases either to modify the transmitter or to remove the module each time. PPMSIM overcomes the problem by redirecting the signal to the trainer port via a different pin on the CPU.

PPMSIM should only be enabled when a FlySky/Turnigy or other problematic module is installed AND if the transmitter has not been modified to include a resistor or transistor in the PPM line. In particular, PPMSIM is not required when an FrSky or OrangeRX DSM2/DSMX module is used; these modules can be left in place when using the Tx with a simulator or as a buddy box.

Please note that when PPMSIM is active as the protocol, no PPM output is sent to the RF module; consequently the transmitter cannot directly control a model.

To activate PPSIM for a given model, PPMSIM must first be enabled in this menu and then be selected under *Model Setup*, Protocol (see page 70).

Channel Order

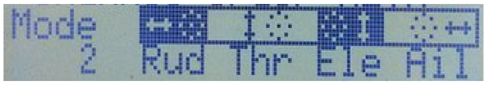
This setting determines the sequence of the first four channels when setting up a new model or applying a template. It is the order in which the outputs will be available from the receiver. The default channel order in Er9x is RETA:

> RETA means RUD = Ch1, ELE = Ch2, THR = Ch3, AIL = Ch4.

Popular channel orders include TAER (used for DSM2/DSMX receivers, which typically provide throttle failsafe only on channel 1) and AETR (used by many manufacturers and often called Futaba order). Er9x provides complete flexibility in selecting channel order.

Note that Channel Order is totally independent of Stick Mode (see below).

Mode *(i.e., Stick Mode)*



```
Mode
   2  Rud Thr Ele Ail
```

Stick Mode determines how the four physical stick axes are assigned to the four primary control inputs (rudder, elevator, throttle, aileron).

Mode 1 uses the left stick for rudder and elevator and the right for throttle and aileron, while Mode 2 has rudder and throttle on the left stick and aileron and elevator on the right (they are sometimes called “throttle right” and “throttle left” modes respectively). Modes 3 and 4 are similar except that rudder is on the right and aileron on the left.

In the picture above, Mode 2 has been selected. In the rectangle, the four stick axes are shown and under each one is the input assigned to it in Mode 2. If you change the mode you will see how the assignments change. Note that this diagram does not show channel order.

Er9x can be set to any of the four stick modes, but the chosen mode must be compatible with the physical setup of the transmitter. If Mode 1 or 3 is selected, the “vertical” axis on the right stick controls throttle and normally has a friction or ratchet mechanism, while the left stick must be spring-centred. If Mode 2 or 4 is selected, the throttle friction mechanism must be on the left stick. Transmitters can usually be purchased as Mode 1 or Mode 2. The 9x is relatively easy to convert between right and left throttle configurations by interchanging the spring and friction mechanisms. For the 9XR, replacement stick assemblies are available in the two physical configurations.

Note that Stick Mode is quite distinct from Channel Order (see above); the two must not be confused. Use Stick Mode to determine how the sticks are configured to control the model: for example, whether you want to use the left or the right stick to control elevator. Use Channel Order to determine which channel will be used to control each servo or other control mechanism in the model: for example, to match the fixed channel order of a BNF Ultramicro model (TAER).

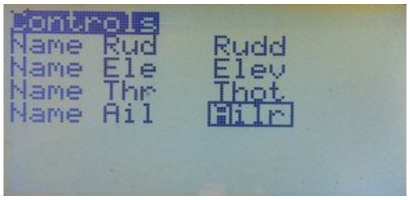
Stick Reverse

If the stock stick assemblies are replaced with a different type (such as Taranis or Aurora 9 gimbals), one or more stick axes may work in directions opposite to the associated trims. The Stick Reverse function is provided purely to correct this situation. To use it, highlight the stick diagram. Pressing [LEFT] or [RIGHT] will then cycle through the 16 different possibilities, with reversed sticks shown on a dark background. The picture above shows “Rud” and “Ele” sticks reversed.

To return to the normal state with no sticks reversed, press [LEFT] repeatedly. Use this feature ONLY to correct sticks that are electrically reversed. Do NOT use it to reverse control directions.

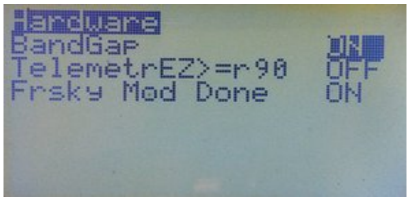
Names of the Four Sticks

On the second page of Controls you can rename the sticks from the default names of Rud, Ele, Thr and Ail. Four characters are allowed for the custom names. For each model you can choose, under *Model Setup, General*, CustomStkNames whether to use the default names or your alternative names (see page 66). This feature is particularly useful for controlling land and water vehicles.



```
Controls
Name Rud    Rudd
Name Ele    Elev
Name Thr    Thot
Name Ail    Ailr
```

Hardware



```
Hardware
BandGap              ON
TelemetrEZ>=r90      OFF
Frsky Mod Done       ON
```

Bandgap

This option allows you to disable the use of the on-chip Bandgap device. This device, which is normally ON, provides an internal reference voltage to ensure accurate measurement of transmitter battery voltage. Without Bandgap, voltages below about 7.2 volts are not reliably reported.

The BG value, shown on the DiagAna screen (see page 24), should be between about 230 and 280. If the value is outside this range, or if it is changing rapidly, then the device is likely to be faulty and Bandgap should be turned OFF. If you do this, however, keep in mind that readings below 7.2 volts may be seriously inaccurate.

TelemetrEZ>= r90

Set this option to “ON” only if you have installed the optional add-on TelemetrEZ board to enable FrSky telemetry, AND have flashed the TelemetrEZ board with r90 or later firmware.

Frsky Mod Done

Set this option to “ON” only if you have done the Frsky telemetry mod in order to be able to display Telemetry information on your radio's screen.

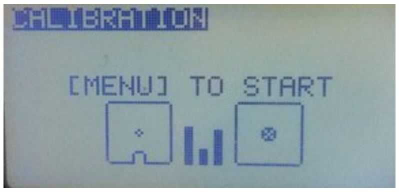
Note: This option is only available on radios with the ATmega128 or Atmega2561 processor; this includes the Turnigy 9XR after about spring 2013 (previously it had an m64 processor).

Calibration

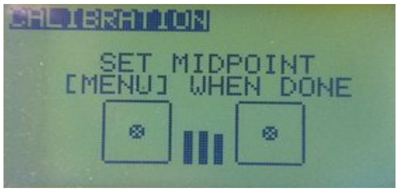
This screen allows you to calibrate the analog inputs A1..A7 (all Sticks and Potentiometers). Calibration is essential when setting up the transmitter.

The calibration method goes like this (see pictures):

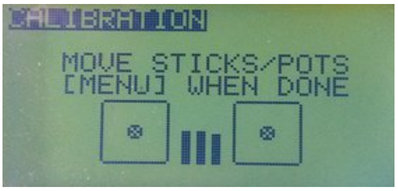
1. Press [MENU]
2. Set Sticks (including Throttle) to centre
3. Press [MENU]
4. Move sticks and pots through full range. Press gently against stops.
5. Press [MENU] – Saves calibration values and takes you back to the start. You can exit at this point.



```
CALIBRATION
[MENU] TO START
```



```
CALIBRATION
SET MIDPOINT
[MENU] WHEN DONE
```

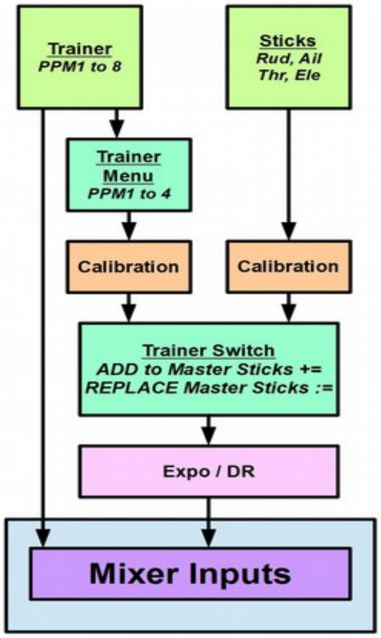


```
CALIBRATION
MOVE STICKS/POTS
[MENU] WHEN DONE
```

Trainer

This menu (see picture on next page) is used when the transmitter is functioning as the Master in a “buddy box” arrangement. The Master transmitter is the one held by the instructor. It is programmed for the model and the model’s receiver is bound to it. The student uses the Slave transmitter, which does not transmit but sends a PPM stream that conveys channels 1 to 8 to the Master transmitter via the trainer port and trainer cable. The Trainer menu is designed to serve the typical trainer setup and deals only with the four stick controls: Rudder, Elevator, Throttle and Aileron. These are carried in the first four channels of the PPM stream and provide all necessary information for basic flight control.

The Trainer menu matches these four channels (PPM1 to PPM4) to the sticks of the Master transmitter, as the diagram at right shows. The channel order of the Slave does not need to be the same as that of the Master, provided the PPMin channels and Master transmitter sticks are correctly matched using this screen. Nor do the stick modes of the two transmitters need to be the same.



The student's transmitter (Slave) does not need to have the same model programming as the instructor's, since only the stick values are used. It just needs a model memory to be set up with a basic four channel model programmed in it. No special settings or mixes are needed for such a basic setup.

The Master and Slave transmitters do not have to be of the same type. Directly compatible transmitters include the various versions of the 9x, as well as the 9XR, 9XR Pro and Taranis. Normally, almost any transmitter that sends a valid PPM signal on its trainer port can be made to work, but some may require different settings, special cables or additional circuitry.

Transfer of control to the student transmitter is done by activating the switch shown in the last column of the Trainer menu. Normally this would be TRN (the momentary toggle switch) for all channels, though sometimes an instructor prefers a different switch, or even chooses to transfer control selectively with separate switches for some channels.

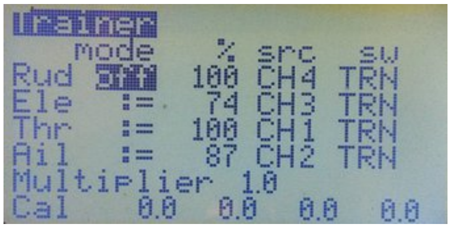
When the Trainer Switch is OFF, the sticks of the Master transmitter control the model. When the Trainer Switch is ON, the Master transmitter gets its inputs from the Student transmitter through the trainer port. These inputs replace, or add to, the Master transmitter stick values, depending on the option selected in the trainer menu. All the processing and mixes on the Master transmitter are applied to these inputs. If, for example, expo, a curve or dual rate is programmed for any of the four primary flight controls on the Master transmitter, it will be applied to the raw trainer inputs from the student transmitter.

For the trainer link to work for the selected model, not only must the Trainer screen in *Radio Setup* be configured in the Master transmitter, but also the Trainer setting in *Model Setup, General* must be set to ON to enable input from the student transmitter to be available (see page 68).

Note that any mixing required to convert the four primary flight controls into servo movements takes place in the Master transmitter. For example, if dual aileron servos on separate channels are used in the model, they will be controlled by the Aileron stick through suitable mixes. No extra mixes are needed for the trainer setup, as turning ON the trainer switch simply uses the corresponding PPMx input to replace or add to the Aileron stick value. Likewise, if the model uses elevon control, the Slave supplies simple Aileron and Elevator values and the necessary mixing takes place in the Master (see page 74 for an explanation and example of elevon mixing).

In addition, all PPMin values (PPM1..PPM8) are available as sources in the mixer, so a model using up to eight channels can be flown in a trainer arrangement. The four additional channels can be used for functions other than the basic flight controls, such as flaps, retractable gear, head tracker and/or camera gimbal control in some FPV setups. See page 87 in Annex A for an explanation of how to do this.

Setting up the Trainer Screen



```
Trainer
    mode    %  src  sw
Rud  OFF   100 CH4 TRN
Ele  :=     74 CH3 TRN
Thr  :=    100 CH1 TRN
Ail  :=     87 CH2 TRN
Multiplier 1.0
Cal    0.0  0.0  0.0  0.0
```

This menu allows the four basic PPMin (trainer) inputs (Rudder, Elevator, Throttle, Aileron) to be configured. The raw inputs from the Trainer transmitter can be selected to Replace (or Add to) the stick inputs on the Master transmitter for training purposes.

The **'mode'** column selects how PPMin is used when the Trainer Switch is activated:

off	**Channel unused**. No input from student stick
+=	**Add** student stick value to instructor stick value (sometimes used for training)
:=	**Replace** instructor stick value with student stick value (normal training setup)

The **'%'** entry applies a weighting to the PPMin value from -100 to +100. Use negative values to reverse the input if necessary. Values closer to 0 reduce the student's control sensitivity.

The **'src'** entry assigns a PPMin channel to each stick shown on the left column. In the picture, the student transmitter is using the Spektrum/JR channel order of TAER.

The '**sw'** entry selects the switch used to activate trainer operation; normally TRN is used for all channels.

Multiplier adjusts all PPMin values by a factor from 0.0 to 5.0. It is normally left at 1. (The multiplier is great for dealing with other transmitters whose makers don't know how to encode PPM. ☺)

Cal*:* To calibrate the Master radio to use the PPMin signals coming from the Slave Tx, first turn on its power switch and select the memory for the trainer model. Then go to *Model Setup, General* and set Trainer to ON.[6]

Now plug the trainer cable into the student transmitter with the power switch in the OFF position. This activates the student transmitter in slave mode.[7]

[6] Note that the Trainer setting in *Model Setup* should only be set to ON when specifically required. Turning it off when not needed avoids any risk that a fault in the trainer port might interfere with control (see page 68).

[7] For the radio to send a PPM signal to the trainer port, the 9x, 9XR, 9XR Pro, Spektrum and many other transmitters must have the power switch in the OFF position and the trainer cable plugged in. Only then can they serve as slave in a trainer setup (or drive a simulator). Others, notably Taranis, require that the transmitter be turned ON and set to "slave" mode. If PPMSIM is activated, however, and the cable is plugged in, the transmitter will send a PPM signal to the trainer port regardless of whether it is ON or OFF (see page 70). In none of these cases will a signal be sent to the RF module.

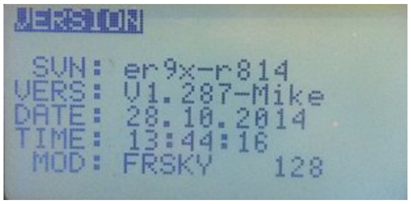
With the Master transmitter still turned ON, plug the trainer cable into its trainer port. The Trainer menu should now show the PPM input values changing when you move the sticks on the Slave transmitter.

Centre the sticks and trims on the Slave transmitter, including the throttle. Now highlight 'Cal' in the Trainer setup menu on the Master transmitter and press the [MENU] button. All of the values to the right of 'Cal' should change to zero.

In order to check that calibration has been successful, you should move the sticks on the Slave (student) transmitter to their fullest extents and make sure that each stick axis has a range of about -100 to 100. This completes calibration of the trainer setup. If necessary, the Multiplier can be used to adjust the input values, but is normally not required.

Test the trainer setup to ensure that control of the primary flight functions transfers correctly to the student when the trainer switch is ON. Check in particular that controls work in the proper directions on both transmitters.

Version



```
VERSION

 SVN: er9x-r814
VERS: V1.287-Mike
DATE: 28.10.2014
TIME: 13:44:16
 MOD: FRSKY     128
```

This screen shows the version information for the current firmware:

SVN: The SVN name of the current revision.

VERS: Version number.

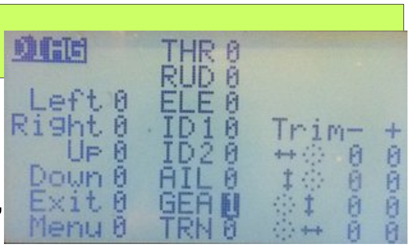
DATE: Compile date for the current firmware.

TIME: Compile time for current firmware.

MOD: Hardware specific version (128 for ATmega128)

Be sure to include the SVN and Mod information if asking for assistance on line.

DiagSwtch

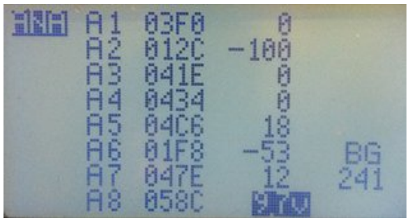


```
DIAG        THR 0
            RUD 0
  Left 0    ELE 0
 Right 0    ID1 0    Trim-  +
    Up 0    ID2 0    ↔       0  0
  Down 0    AIL 0    ↕       0  0
  Exit 0    GEA 1      ↕     0  0
  Menu 0    TRN 0      ↔     0  0
```

This menu will help you visualize the current state of the trims, keys and physical switches and verify their operation.

Each Key/Switch/Trim is represented. When a key or switch is pressed, the value is highlighted and changes from 0 to 1. Very useful for troubleshooting problems with these inputs.

DiagAna



```
ANA A1 03F0    0
    A2 012C -100
    A3 041E    0
    A4 0434    0
    A5 04C6   18
    A6 01F8  -53    BG
    A7 047E   12    241
    A8 058C  9.7v
```

Here the analog inputs are displayed in hexadecimal format to save space (and annoy you at the same time ☺). Values range between 0 and 0x3FF (0..1023).

A1..A4 are the gimbals (sticks).

A5..A7 are the pots (potentiometers).

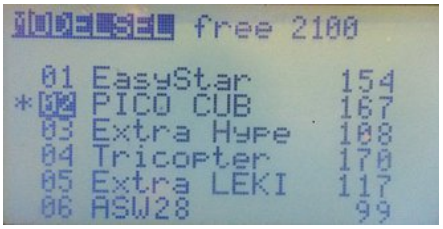
A8 is the voltage of the transmitter battery as reported by the transmitter. This value needs to be calibrated to ensure accuracy of the display and of the low battery warning.

Calibrating TX Battery Voltage: Press [MENU] to enable calibration of A8. Measure the voltage of the battery with a meter while the transmitter is turned on and adjust the A8 value to match by pressing [LEFT]/[RIGHT]. Press [MENU] or [EXIT] to complete the process.

The **BG** value is the Bandgap measurement mentioned earlier. The value should be between about 230 and 280. Readings outside that range may indicate a faulty bandgap device; in that case go to the Hardware menu and turn OFF Bandgap (see page 21).

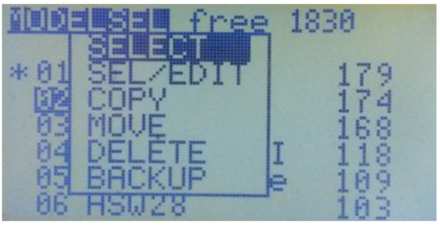
Model Select

To reach the Model Select screen from the Main Screen, press [RIGHT LONG], or press [MENU LONG] and select the option from the pop-up. On the Model Select screen, models stored in the transmitter's EEPROM memory are listed and can be selected (i.e., made active as indicated by an asterisk). They can also be edited, copied, moved, deleted, backed up or restored. At the top of this screen you can see the amount of free space that is still available in the EEPROM.



```
MODELSEL free 2100
 01 EasyStar      154
*02 PICO CUB      167
 03 Extra Hype    108
 04 Tricopter     170
 05 Extra LEKI    117
 06 ASW28          99
```

If a non-active model slot is highlighted and the [MENU] key is pressed, the "SELECT", "SEL/EDIT", "COPY", "MOVE", "DELETE" and "BACKUP" options become available, as shown at right. Move the cursor using the [UP]/[DOWN] keys and pick the option by pressing [MENU].



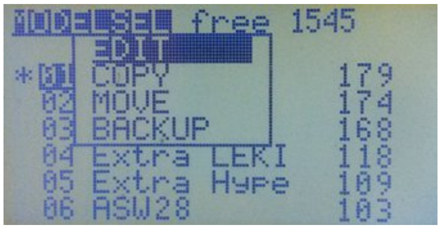
```
MODELSEL free 1830
    SELECT
*01 SEL/EDIT      179
 02 COPY          174
 03 MOVE          168
 04 DELETE   I    118
 05 BACKUP   e    109
 06 ASW28         103
```

SELECT: Selects the model and exits to the last main screen.

SEL/EDIT: Selects the model and enters the edit menu.

Active Model / Select a Model

An asterisk next to the model number indicates that this is the currently active model. Only the active model can be edited. To SELECT a model or memory slot, highlight it, press [MENU], choose SELECT and press [MENU] again to make it the active model.

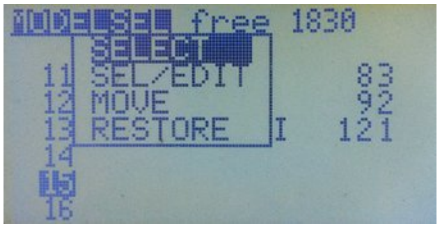


```
MODELSEL free 1545
    EDIT
*01 COPY          179
 02 MOVE          174
 03 BACKUP        168
 04 Extra LEKI    118
 05 Extra Hype    109
 06 ASW28         103
```

If you press [MENU] when the active model is highlighted you get the options "EDIT", "COPY", "MOVE" and "BACKUP" as shown on the picture above. You do not get an option to delete, as a model cannot be deleted while active.

Press short [RIGHT] or [LEFT]to directly enter the *Model Setup* index page for the chosen model. Don't be surprised if you see an ALERT screen telling you that THROTTLE needs to be moved to idle, and/or the switches must be in their default position. For your safety, the throttle stick and switch positions are checked every time you change models, not just when the transmitter is turned on. This important safety feature can be turned off if desired, but doing so is not recommended.

Pressing [MENU] on a blank slot will cause a pop-up menu to come up with the options "SELECT", "SEL/EDIT", "MOVE" and "RESTORE".



```
MODELSEL free 1830
    SELECT
 11 SEL/EDIT       83
 12 MOVE           92
 13 RESTORE  I    121
 14
 15
 16
```

If a blank memory slot is selected as the active model, a basic four-channel model is created by defining standard mixes for channels 1 through 4, using the default channel order set in *Radio Setup, Controls* (see page 19). The new model is initially named MODEL XX, where XX is the memory slot number.

Copy, Move, Delete a Model

Note that the commands in the “Model Select” menu (SELECT, COPY, etc.) apply to the model whose number you have highlighted (which may not be the active model).

To **COPY** a model, highlight it and press [MENU]. Choose COPY and press [MENU] again to confirm that you want to proceed with the duplication. The copy will be created in the first available blank slot.

To **MOVE** a model, highlight it and press [MENU]. Choose MOVE, press [MENU] again. Use the [UP/DOWN] keys to move it to a different memory slot and press [MENU] to drop the model there. You can also move an empty slot and place it in between existing models. Sounds crazy? Yes it does, but it's deliberate. 🙂 Moving a model will not overwrite other models.

To **DELETE** a model, highlight it and press [MENU]. Choose DELETE and press [MENU]. You will be asked to confirm.

Note: You cannot delete the active model. You first need to select another model and make it active, then you can go back to the previous model and delete it.

Backup and Restore a Model

This feature allows you to backup and restore models using the SD Card in a Megasound9x voice module.

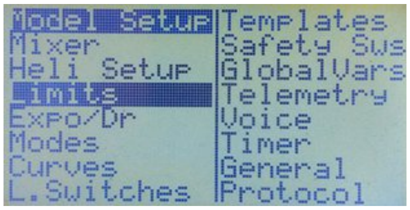
Backup/Restore requires that a Megasound9x voice module be installed in the radio and that two extra connections be made between the voice module and the transmitter main board. These connections are not part of the normal installation of the module. Please click here and read the information in the forum.

For Backup/Restore to be available in the Model Select menu, the option “MegaSound” must be selected in *Radio Setup, AudioHaptic, Sound Mode* (see page 16).

To **BACKUP** a model, highlight it and press [MENU]. Choose BACKUP, press [MENU] again. A progress bar will be shown. When the backup process is finished the message “BACKUP DONE” will be displayed. Press [EXIT] to go back to the Model Select menu.

To **RESTORE** a model, highlight an empty slot and press [MENU]. Choose RESTORE and a list of the models on the SD Card will be displayed. Select a model using the [UP]/[DOWN] keys and press [MENU]. A screen with a progress bar will be shown. When the operation is complete, the message “RESTORED” will be displayed. Press [EXIT] to go back to the Model Select menu.

Model Setup



This page displays the menus that give access to settings specific to the currently active individual model (Model XX). Note that only the menus that are supported by the installed version of the firmware will be shown.

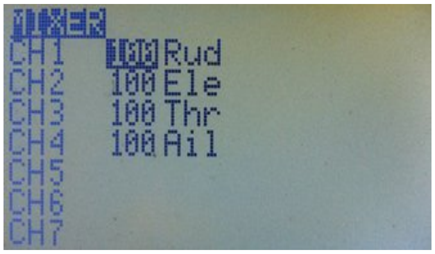
Select a menu using the [UP]/[DOWN] and [LEFT]/[RIGHT] keys, then press [MENU] to open it. The list of menus begins with Mixer, the focal point of programming.

Mixer

The Mixer menu is the heart of Er9x. Without the entries here, nothing happens. Understanding how mixes work is vital to being able to program the transmitter.

The function of the mixer is to take the inputs (Sticks, Pots, Switches, etc.), perform some function(s) on them, and route the results to the output channels. Since selection of inputs is totally free and a wide range of functions is provided, this is a very flexible and powerful system. It's also very quick to boot.

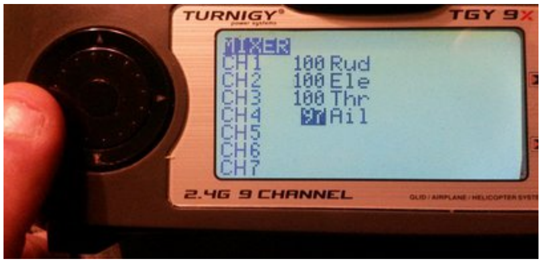
Main Screen



```
MIXER
CH1  100 Rud
CH2  100 Ele
CH3  100 Thr
CH4  100 Ail
CH5
CH6
CH7
```

When you enter the Mixer menu for the first time you will probably see something like the simple set of mixes in the picture at right.

What this is telling you is that, for example, the rudder stick input (Rud) is being routed with a weight of 100% to channel 1 (CH1). Same for the elevator on CH2, throttle on CH3 and aileron on CH4. The sequence of channels of the default model is set by Channel Order in the *Radio Setup, Controls* menu (see page 19). Here the order is RETA.

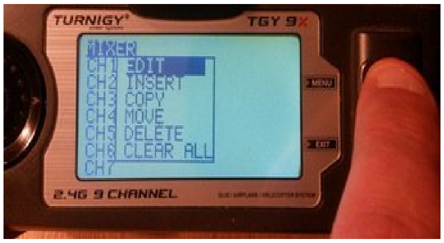


```
MIXER
CH1  100 Rud
CH2  100 Ele
CH3  100 Thr
CH4   97 Ail
CH5
CH6
CH7
```

As you scroll down you'll see that the weight value is being highlighted on channels that have a mix already programmed. Pressing [LEFT]/[RIGHT] will edit the weight value.

If a channel you select has no mix programmed, a box appears around the channel number. An empty channel will simply centre any servo connected to it. The servo will not move at all as the channel has no input.

Pressing [MENU] on an empty channel immediately adds a new mix and goes to edit it. Please see “Edit Mix” below. Pressing [MENU] on an existing mix will pop up a menu of options:



EDIT – Enters the “Edit Mix” menu.

INSERT – Inserts a new mix after the highlighted one.

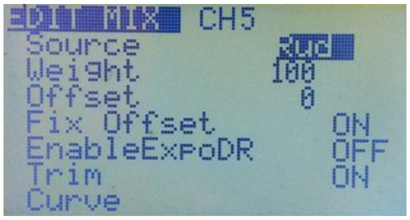
COPY – Copies the highlighted mix and places it on the same channel after the highlighted one.

MOVE – Highlights the whole mix and allows [UP] and [DOWN] to move it. Press [MENU] to return to normal selection mode.

DELETE – Deletes the highlighted mix after displaying a confirmation screen.

CLEAR ALL – Deletes ALL mixes for the model after displaying a confirmation screen.

Edit Mix



```
EDIT MIX  CH5
Source          Rud
Weight          100
Offset            0
Fix Offset       ON
EnableExpoDR    OFF
Trim             ON
Curve
```

We need to take a pause here and look closely at the choices available to us in setting up a mix. This is where you can determine how the source (input) will be transformed into output that drives the servo(s) plugged into the corresponding receiver channel.

It's important to understand that in the Er9x kind of programming, the only way anything can be controlled is through an explicit mix. In many other radios, most of the mixes are built-in but hidden; here they are fully visible and editable, giving the radio enormous flexibility and power, but requiring that, except in the case of the default mixes for a new model, you create the mixes yourself.

Here are the available options for each mix:

Source

This is the input for the mix. It can be any of the following:

Stick or potentiometer

(Rud, Ele, Thr, Ail, P1, P2, P3).

HALF

Normally controlled by a switch. Output is either 0 if the switch is OFF or Weight if the switch is ON. For example, if weight is 100, output is either 0 or 100; if weight is -80, output is 0 or -80.

FULL

Normally controlled by a switch. Output is either -Weight if the switch is OFF or +Weight if the switch is ON. For example, if weight is 100, output is -100 or 100.

HALF and FULL can be a little confusing. Look in the examples section for, well, examples 🙂

CYC1, CYC2, CYC3

The outputs of the heli swash-plate mix for a flybar-type helicopter using CCPM. Once swash mixing is turned on (Heli Setup, page 39) these become active as mix inputs reflecting the results.

Generally CYC1 controls the fore/aft output and the other two do the rolling. On the 120X mode CYC1 is the odd one out.

PPM1..PPM8

These represent the PPM input from the trainer port. They can be used to configure a trainer (buddy box) system or simply to extend the radio with more functions (like head tracking for you FPV fans). Note that the Trainer menu under *Radio Setup* (page 22) deals only with CH1 – CH4, so PPM input offers a more flexible way to receive inputs from another transmitter or device (see page 87 for examples).

CH1..CH16

These are the mixer outputs of all 16 channels, not including limits, safety switches, etc.; they are available from the Input list when the option “Use Output” is OFF (see page 34). Normally higher channels (beyond those transmitted) can be used as virtual channels and their mixer output can be used as source for another mix. This capability can be used to chain mixes for more complex behaviour.

OP1..OP16

These are the real outputs of all 16 channels; they are available from the list when the option “Use Output” is ON (see page 34). Normally higher channels (beyond those transmitted) can be used as virtual channels and their real output can be used as source for the mixer. The "Use Output" option can provide the real output from any of the 16 channels to be used anywhere in the programming you can select a channel as the source.

Switches

sIDx sTHR, sRUD, sELE, sAIL, sGEA, sTRN – Any of the various physical switches can be used as the source for a mix, thus providing a simple way to program a switch to control a function such as Gear or Flaps.

Within the Source field of the Edit Mix screen, the first switch listed is sIDx, meaning the three-position switch (ID0, ID1, ID2). The two-position switches are then listed as sTHR, sRUD, sELE, sAIL, sGEA, sTRN. The “s” indicates that the switch is being used as a source.

Used as the mix source, a two-position switch will assume a value of *-/+ weight* depending on its state, while the three-position switch will also provide a middle position output. For example:

CH5 100% sIDx

This mix will produce outputs of -100, 0, 100 in response to the three-position switch.

CH6 100% sTRN

Gives 100 when the momentary switch is pressed and returns to -100 when released.

There are several ways to adjust the results of such a mix. The operation can be reversed by using a negative value for weight. As well, an offset or curve can be used to set non-symmetrical values for the output. Alternatively, in the Curve field of the Edit Mix screen, a Differential (Diff) curve, can be used to produce non-symmetrical outputs. For example:

CH5 80% sAIL Diff(-50%)

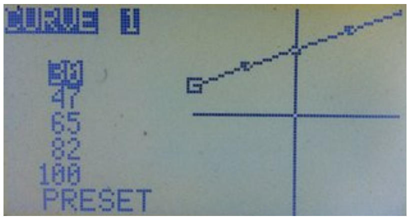
The output of this mix will be -80% when sAIL is OFF, and 40% when sAIL is ON.

An alternative way to modify the outputs is to use the Limits menu. When a mix using a switch as source is the only mix in the channel, the end points can be controlled by changing the individual limits values, while for the three-position switch (IDx) the middle point can be adjusted by changing the Subtrim value. Also in the Limits menu, the direction can be reversed by inverting the channel.

In Er9x, multiple mixes can be combined on a single channel. Here is an example that provides triple rates using two mixes. The first is controlled by a pot (P1) and the second uses the three-position switch (IDx) as source:

CH5 100% P1

* 100% sIDx Curve (c1) (Note Multiply mix – see page 36)



CURVE 1

30
47
65
82
100
PRESET

With the curve (c1) programmed as in the picture these mixes will scale the weight of P1 using the three-position switch. When the switch is on ID0 the resulting weight will be 30%; on ID1 it will be 65%; and on ID2 it will be 100%.

The following example shows how the AIL switch can be used as the source (sAIL) in a mix that provides spoileron action with a pair of aileron servos on separate channels:

CH01 +100% Ail

+10% sAIL Offset 10% Slow(u10:d10)

CH05 -100% Ail

+10% sAIL Offset 10% Slow(u10:d10)

For each channel, the first mix provides normal aileron action, while the second uses sAIL to raise the ailerons slowly. If combined with lowered flaps, this provides CROW braking.

Finally, with one switch entered in the Source field, a second switch can be used in the Switch field to turn the mix on or off. When ON, the output will be the value of the source switch. When OFF, the mix output will be 0%, regardless of the weight or position of the source switch. For example:

CH5 100% sELE Switch (RUD)

When the RUD switch is ON, the mix output will be the usual -100 or 100, depending on the ELE switch position. When OFF, the output will be 0%, regardless of ELE position.

GV1..GV7

The values of the Global Variables (GVARs) (see page 54).

THIS

A source that represents the combination of all mixes for the channel to this point. Well what does it mean? Tricky, but here's an explanation:

In Er9x, if we apply *“Slow down/up”* to a single mix on a channel, it works as it should. Example:

CH5 100% FULL switch (ID1) slow (u3 : d3)

This mix will work without any problems.

But the following won't work:

CH15 -100% HALF

R   +50% FULL switch (ID1) slow (u3 : d3)

R +100% FULL switch (ID2) slow (u3 : d3)

When we flick the switch the channel will not slow down but jump when coming back towards -100%. This is due to how the mixer works. If we have more than one mix with “slow” on the same channel, it won't work properly. Sometimes flexibility comes at a cost.

Anyway, we can solve this problem by using a higher unused channel as a virtual channel for handling the switches, and then apply it as source for CH5. Then apply the slow to CH5.

The following will work properly:

CH5 +100% CH16 slow (u3 : d3)

CH16 -100% HALF

R +50% FULL switch (ID1)

R +100% FULL switch (ID2)

We removed the “slow” from the mixes with switches and applied it to a single mix on CH5.

The THIS input will solve the problem in a different way, without having to use a virtual channel. You can keep all your mixes together and create a mix AFTER all of them with THIS as its source. Then apply the “slow” to this last mix. It has the same effect as having all the mixes with the switches on a virtual channel. The difference is, you keep all your mixes together. Here is how :

CH15 -100% HALF

R +50% FULL switch (ID1)

R +100% FULL switch (ID2)

R +100% THIS slow (u3 : d3)

SC1..4

Scalers provide an adjusted input value by applying an offset and/or scaling factor. (see page 56).

Weight

This value multiplies the value from the input. It can range from -125% to +125%. Default is 100%. GVARs can also be used as weight. To use a GVAR press [MENU LONG].

Offset

This value is added to the value from the input. It can range from -125% to 125%. Default is 0%. GVARs can also be used as offset. To use a GVAR press [MENU LONG].

Fix Offset

Fix Offset is ON by default. It corrects an issue with previous versions of the firmware, which added offset values early in the mixing process and consequently had outcomes that were hard to calculate. Fix Offset instead adds offset values at the end of the mixing process, where their impact is clear and predictable. Fix Offset should only be turned OFF when using a model definition created under an earlier version of the firmware.

What it Does

When Fix Offset is ON, offset is added at the end of Mixer processing, as follows:

1. Obtain the source value, with expo/dual rates applied to sticks.
2. Apply Delay and Slow.
3. Apply Curve/Differential/Expo.
4. Multiply by Weight.
5. Add Offset.

The Fix Offset approach ensures that the offset value is not subjected to mixer processes that might change its value; hence it makes the mix results easy to predict.

When Fix Offset is OFF, the offset value is added immediately after the source value is defined (i.e., after step 1 above). The offset value is thus affected by subsequent processing and the results are often hard (if not impossible) to calculate. This is how previous versions of Er9x worked. The ability to turn OFF Fix Offset is therefore provided to allow existing model definitions to be used without modification.

When programming new model definitions it is strongly recommended that the new approach be used, i.e., that Fix Offset be left turned ON. Indeed, even when modifying an existing model definition, it is preferable to use Fix Offset and adjust the programming accordingly.

Understanding "Fix Offset"

NOTE: The following examples assume that all values for Dual Rates, expo, etc., are default, just in case you decide to use other inputs to the mix, like the sticks for example.

Consider the following mix:

CH6 50% P1 offset 50%

If the offset is not taken into account, a weight of 50% means that the output value of the mix will travel from -50% to 50% when the input P1 (pot) is moved through its full range. The offset of 50% simply adds 50 to the value, giving a final mix output range of 0% to 100%.

To put it slightly differently:

P1 fully counter clockwise gives an output of -50%.

Adding 50% offset gives a final mix output value of -50 + 50 = 0%.

Moving P1 fully clockwise gives a value of 50 + 50 = 100%.

So, with the mix above, moving the pot through its full range will give mix output values between 0% and 100%, centred on 50%.

Let's take another example:

CH5 20% P1 offset 10%

If there were no offset, the 20% mix would output values between -20% and 20%.

P1 fully counter clockwise gives an output of -20%. Now add 10% offset:

Adding 10% offset gives a final mix output value of -20 + 10 = -10%.

Moving P1 fully clockwise gives a value of 20 + 10 = 30%.

So, moving the pot through its full range will give us Mix output values between -10% and 30%, centred on 10%.

In essence, Fix Offset ON ensures that the Mix output is simply shifted by the amount set in the offset field.

Now try it yourself with other values for Weight and Offset. You can also use other inputs (like the sticks) if you get fed up with moving the little pot. ☺

Turn on your radio with a servo attached to the receiver and you will see that changing the value of the offset shifts the centre of the servo travel by the offset amount. As well, you will observe that the servo travels the value of *-weight* to one side of the new centre and *+weight* to the other when you move the input, in this case P1, to the extremes.

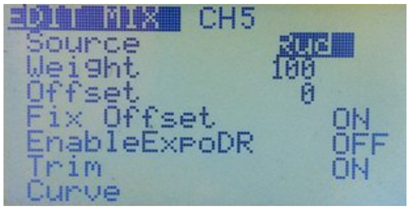
Then try calculating the result of the examples above with Fix Offset turned OFF, that is with the offset applied to the P1 input. You will find (a) it’s hard to do even for these simple examples and (b) the results are totally different from what you might expect! Imagine how difficult it would be to predict the result of a Mix output when other complex processing is applied after the offset.

Please note that Mix Output value and Channel Output value are quite distinct. The Mix Output is the final value of a single mix; it can be added to, multiplied by, or used to replace other mixes in the same channel. The Channel Output is the result of all the mixes applied to that same channel.

To fully understand offset, it's a good idea to program the above examples in your radio, if possible with a receiver and a servo connected, so you can see what the different fields of the mixer do. Of course you can also use the eePe program on your computer for this purpose (see page 5).

EnableExpoDR

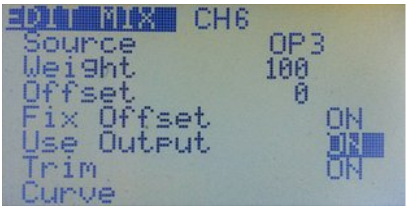
This option is only visible when a stick is selected as source for the mix, and it is ON by default. When OFF, any Expo and Rates applied previously to the input (Rud in the picture) will be ignored and the raw value of the input will be used instead.



```
EDIT MIX  CH5
Source          Rud
Weight          100
Offset            0
Fix Offset       ON
EnableExpoDR    OFF
Trim             ON
Curve
```

Use Output

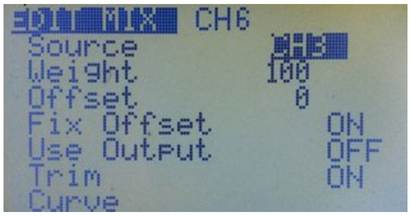
The purpose of this option is to provide access to the real output of a channel (channel 3 in the pictures) so it can be used as source for a mix in the channel currently being edited (channel 6 in the pictures). This channel (6) can in turn be used as the source for a GVAR, scaler, logical switch or in any other situation where a channel can be selected as the source.



```
EDIT MIX  CH6
Source          OP3
Weight          100
Offset            0
Fix Offset       ON
Use Output       ON
Trim             ON
Curve
```

When this option is ON, the real output of the source channel, including the results of any safety switch and of the various settings in the Limits menu (which are applied after the mixer), is available to be used as input for the mix. The input appears as “OPx”, where “x” is the source channel number. In the picture, this is the real output of channel 3 and is identified as “OP3”. This mix input is identical to the channel value sent over the air to the receiver (see the diagram on page 7).

When this option is OFF (the default), only the mixer output of the source channel is available to be used as input to the mix; any applicable safety switch and limits menu settings are ignored, as they are applied after the mixer. The input in this case appears as “CHx”, where “x” is the source channel number. In the picture, the input (the mixer output of channel 3) can be seen as “CH3”.



```
EDIT MIX  CH6
Source          CH3
Weight          100
Offset            0
Fix Offset       ON
Use Output      OFF
Trim             ON
Curve
```

Understanding “Use Output”

To understand how this option works, try the following example:

Set up channel 3 for throttle and add a safety switch using the THR switch as a throttle lock:

CH3: +100% Thr with the safety switch **CH3 S THR -100**

Now go to the main graphic servo outputs screen (“bars”) and check that the throttle channel responds normally when the THR switch is OFF (throttle is unlocked). Turning ON the THR switch should lock the output at full low (-100) regardless of stick position.

Next set up mixes on, say, channels 7 and 8 with channel 3 as the source. You will see that in the Mix Edit screen for these channels the option “Use Output” appears. Leave it at OFF (the default) for channel 7 and turn it ON for channel 8:

CH7: +100% CH3 with "Use Output" OFF (i.e., safety switches/limits NOT included)

CH8: +100% OP3 with "Use Output" ON (i.e., safety switches/limits included).

With the safety switch (THR) in the OFF (unlocked) position, the bars on the graphic output screen will show that when you move the throttle stick, the outputs for channels 3, 7 and 8 all change together.

If you now turn the THR switch ON, channel 3 will be locked at low throttle (-100), but channel 7 will continue to respond to stick inputs because the safety switch is ignored (“Use Output” is OFF).

By contrast, channel 8 will now be locked at low throttle regardless of stick position because it is a true copy of channel 3, and therefore is affected by the safety switch (“Use Output” is ON). Any settings in the Limits menu would also affect channel 8 but not channel 7.

In general, the "Use Output" option can provide the real output from any of the 16 channels to be used anywhere in the programming you can select a channel as the source. In the example above you could select channel 8 as the source for a GVAR, scaler, logical switch or even another mix.

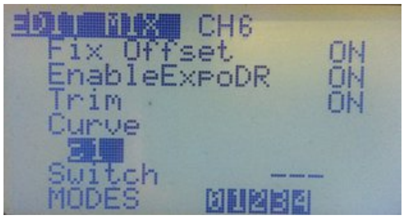
Note also that the “Use Output” option is not available on mixes that have a stick (Rud, Ele, Thr, Ail) as the source; instead, those mixes provide the “EnableExpoDR” option (see page 34).

Trim

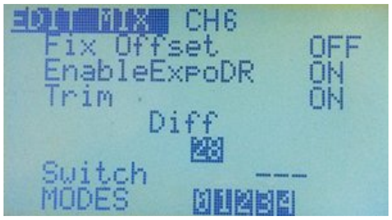
When this is ON, any trim value for the channel will be carried through the mix. When OFF, any trim value is ignored.

Curve

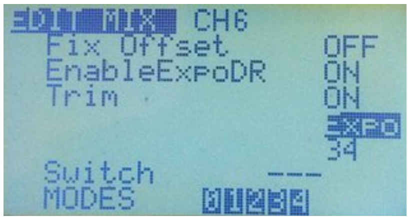
Here you may select a Curve or apply Diff (Differential) or Expo (Exponential) to the mix.



```
EDIT MIX  CH6
 Fix Offset        ON
 EnableExpoDR      ON
 Trim              ON
 Curve
   c1
 Switch           ---
 MODES       01234
```



```
EDIT MIX  CH6
 Fix Offset        OFF
 EnableExpoDR      ON
 Trim              ON
          Diff
           28
 Switch           ---
 MODES       01234
```



```
EDIT MIX  CH6
 Fix Offset        OFF
 EnableExpoDR      ON
 Trim              ON
                   Expo
                   34
 Switch           ---
 MODES       01234
```

The pictures above show each of these three options. Use the [LEFT] and [RIGHT] keys to select the one you want to apply to the mix. Pressing the [DOWN] key highlights the value under the words “Curve”, “Diff” or “Expo”. The [LEFT] and [RIGHT] keys adjust the values. The range of possible values for Expo and Diff is 0 to +/- 100.

For a curve you have:

x>0 – The value of the source is carried through only if it is positive (greater than zero). If the source value is less than zero, a zero value is applied.

X<0 – Same but for negative values.

|x| – The value is passed as an absolute value. The output value will be always positive regardless of the source being negative or positive. An input of either -50 or +50 will produce an output value of +50.

f>0 – If the source is positive then the output value is "+weight" otherwise it is 0.

f<0 – If source is negative then the output value is "-weight" otherwise it is 0.

|f| – Output is either "+weight" or "-weight" depending on the sign of the source.

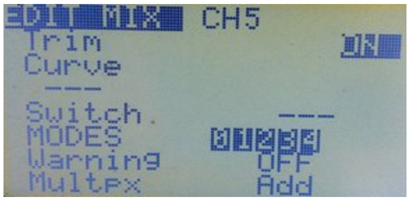
c1..c16 – Custom Curves. These are defined in the "Curves" menu (see page 44). You can press [MENU] to edit the curve directly. When you have finished editing the curve, short press the [EXIT] key to return to the EDIT MIX menu.

Switch

Here you select the switch to control the mix. If a switch is not selected then the mix is ON by default. Options are the physical switches, the logical switches and the inverse of all of them.

Modes (Flight Modes)

This option allows you to select in which Flight Mode(s) this particular mix is active; by default a mix is active in all Flight Modes. See page 43 for an explanation of flight modes.



```
EDIT MIX  CH5
Trim              ON
Curve
 ---
Switch.        ---
MODES        01234
Warning        OFF
Multpx         Add
```

Warning

Here you can select an audible signal that will sound whenever a mix is turned on (only works when a switch is defined). You have an option of three different warning sounds. The sound clusters will play repeatedly.

Multpx (multiplex)

This value defines how the mix affects the channel. The three options are: Add, Multiply and Replace.

Add – This is the default – the mix is added to the previous values in the same channel.

You can add a fixed value to the channel, like an offset:

```
CH2  +90%   Ele
     +10%   FULL
```

When the Ele channel is in the middle, the output will be 0 + 10 = 10% (just like an offset). When at full throw, the outputs are:

-90 + 10 = -80% and 10 + 90 = 100%

Now we ADD a variable value to the first mix. Two examples. The first one is the familiar Rudder-Aileron mix. We will ADD a bit of Aileron to the Rudder

```
CH1  +100% Rud
     +10%  Ail  switch (AIL)
```

Another example. The well known Elevon mix used on flying wings:

CH1 +50% Ele

+50% Ail

CH2 +50% Ele

-50% Ail

Multiply – Use this to multiply the previous values in the same channel.

A multiply mix is recognized by an asterisk **" * "** before the weight value. What is a multiply mix? A multiply mix multiplies percentages, thus:

80% is 80/100

60% is 60/100

So if they are multiplied we get:

(80*60)/(100*100) = 4800/10000 = 48/100 = 48%.

Basically a multiply mix applies a percentage to the value of the mixes above it on the same channel. For example:

CH1 +100% Rud

* +20% FULL

The output of the channel here will be 20% of the weight from the first mix and goes from -20% to +20%. Imagine that we change the weight to 80%. The output will now go from -16% to 16%.

Here's another example of a multiply mix using an "offset" to change the output of a channel from 0% movement to full throw in response to the position of Pot P1:

CH1 +100% Rud

* +50% P1 offset 50% (Multiplies previous line by 0% to 100%, depending on P1)

So, the total output from CH1 when moving the Rud stick will be scaled depending on the position of P1, from 0% weight (no movement, servo centred) to full throw which is 100%, the value of the weight of the mix above.

Note: Make sure Fix Offset is ON in the Edit Mix menu or the offset calculation will be wrong (see page 32).

If you have eePe installed on your computer you can just play with these values and confirm what was explained above. You can also change the values and see the output of the channel change accordingly. If you still didn't install eePe, well, what are you waiting for? ☺

Replace – This option is used in conjunction with a switch. When the switch is ON, a replace mix overrides all the previous mixes on the same channel.

A Replace mix is recognized by an "R" on the left hand side before the weight value. Here is an example of a replace mix involving a safety switch on the throttle channel:

CH3 +100% Thr

R -100% FULL !THR

As we know, the "!" indicates the reversed operation of the switch; that means the mix will be ON when the switch is OFF. So in this case, the channel output will be -100% because the replace mix is overriding the previous mix, which has the throttle stick as source. When the switch is OFF, you can move the stick as much as you like, your plane ain't goin' nowhere. 🙂

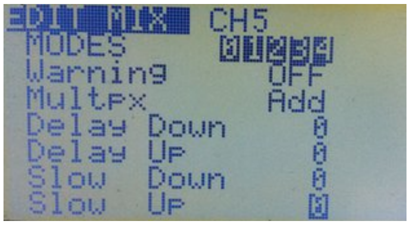
When you turn the switch ON, the mix will turn OFF, and the Thr stick will control the channel again. Cool..!!!

Delay Down/Up

Use this option to delay application of the mix, usually to delay the action of a switch. When the switch is turned on or off the mixer will wait the specified number of seconds before changing the value.

Slow Down/Up

Use this option to slow the rate of change in the channel. When not zero these values will determine the maximum speed with which the value can change. The specified value is the number of seconds to go between -100% and 100%. When you select Slow it only applies to that mix, NOT to any others on the same channel.

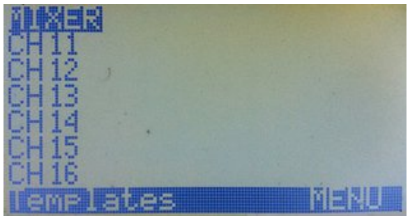


```
EDIT MIX  CH5
MODES        01234
Warning        OFF
Multpx         Add
Delay Down       0
Delay Up         0
Slow  Down       0
Slow  Up         0
```

If you want a channel output to always move slowly, but with multiple mixes, one easy way is to use another channel for all the mixes, then use the output of that channel as the SOURCE of the real output with “Slow” selected. Yet another possibility, if you really wish to keep all your mixes together is to use a Replace mix after all the others, with THIS as source (see page 31). This mix will then handle the slow Down/Up function. See also the example on page 71.

Templates

Templates are starting points for model setups. When selected with [MENU LONG] they will either ***add to*** or ***replace*** the current model's mixes and settings with standard ones for a particular usage scenario. Best applied to a newly created model, they can be tweaked to achieve the desired result, or simply used to get an idea of what's required for the particular model type.



```
MIXER
CH11
CH12
CH13
CH14
CH15
CH16
Templates                 MENU
```

The Templates menu is found at the bottom of the list of channels in the Edit Mix menu (most quickly reached by pressing [UP] from the CH1 entry). It is also accessible as a separate menu item within the *Model Setup* menu, except in the telemetry version of Er9x for the ATmega64 processor (Er9x-frsky.hex), which does not include templates.

Nine templates are provided. For details of what they do and how they work, see page 49.

01 Simple 4-CH (basic four channel model)

02 T-Cut (simple throttle cut switch)

03 Sticky T-Cut (sticky throttle cut switch)

04 V-Tail (rudder-elevator mix)

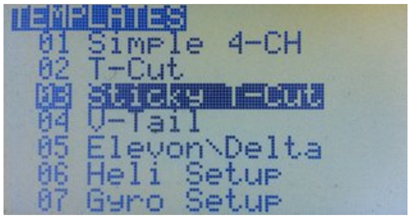
05 Elevon\Delta (aileron-elevator mix)

06 Heli Setup (for CCPM helicopters)

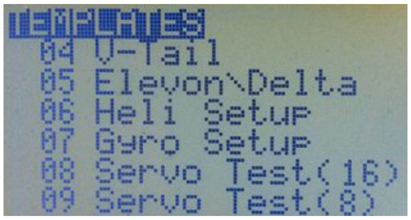
07 Gyro Setup (for tail rotor)

08 Servo Test (16) (oscillating output on channel 16)

09 Servo Test (8) (oscillating output on channel 8)



```
TEMPLATES
 01 Simple 4-CH
 02 T-Cut
 03 Sticky T-Cut
 04 V-Tail
 05 Elevon\Delta
 06 Heli Setup
 07 Gyro Setup
```

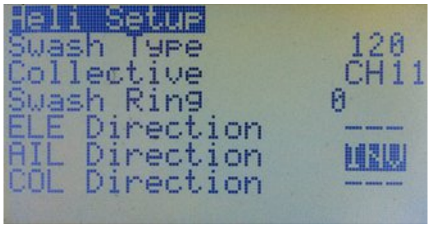


```
TEMPLATES
 04 V-Tail
 05 Elevon\Delta
 06 Heli Setup
 07 Gyro Setup
 08 Servo Test(16)
 09 Servo Test(8)
```

Heli Setup

This menu adds specialized functions to the mixer capability to support CCPM (Cyclic-Collective Pitch Mixing). It allows the swash plate type to be selected and the control authority of cyclical inputs to be limited through the Swash Ring setting.

Note that many modern rotary wing models, whether flybarless helicopters or multirotor machines, use an on-board flight controller that cannot accept CCPM. Likewise, coaxial helicopters typically require only simple fixed wing type inputs for the rotors (aileron, elevator, rudder) and use throttle to control climb and descent. For all such models, you can ignore this section.



```
Heli Setup
Swash Type           120
Collective          CH11
Swash Ring         0
ELE Direction        ---
AIL Direction        INV
COL Direction        ---
```

Swash Type

This defines the type of swash plate used by the heli:

120: “Standard” 120 degree swash plate. The “pitch” servo is towards the front/back.

120X: Same 120 degree swash plate but turned 90° so the pitch servo is on one side.

140: Less common 140 degree swash plate – the “pitch” servo is towards the front/back.

90: Basically a simple 90 degree setup where a single servo operates the collective pitch directly and discrete aileron and elevator servos tilt the swash. May be used for fixed pitch helicopters.

Collective

This defines the source of the collective input. The idea is that you can create a mix in a virtual channel (above those sent to the receiver) incorporating all the required curves and switches, then simply use it as the input here to mix with the other inputs.

Swash Ring

As the name implies, this feature limits stick outputs, just as a physical swash ring limits stick movements. Notice that this feature only works on AIL and ELE, regardless of stick mode selected.

ELE/AIL/COL Direction

These allow the direction of the cyclical and collective input functions to be inverted. Use them to make the controls move the correct way when setting up your heli.

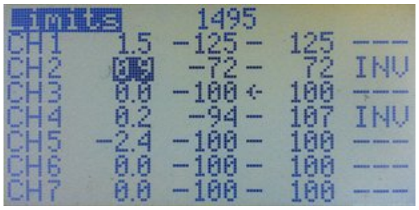
The outputs of this Heli Setup menu are CYC1, CYC2 and CYC3. These need to be assigned on the MIXER menu to the channels that will drive the swash servos. The settings made here have no effect unless you use CYC1, CYC2 and CYC3 as sources in the mixer.

As noted above, a multirotor or flybarless helicopter which uses an onboard flight controller (FC) will NOT use the CCPM configuration.

For more information on setup please consult forums devoted to the relevant type of helicopter.

Limits

Limits operate on the Mixer output channels. In the *Limits* menu (see picture at right) you can set the centre point (Subtrim) and the limits (both minimum and maximum), as well as reversing the output of the channel (INV for invert). Use the [MENU] key to activate a field and then use the [RIGHT] or [LEFT] keys to increase or decrease the value. Use [EXIT] to finish editing.



```
Limits          1495
CH1      1.5  -125 -  125 ---
CH2      0.0   -72 -   72 INV
CH3      0.0  -100 <- 100 ---
CH4      0.2   -94 -  107 INV
CH5     -2.4  -100 -  100 ---
CH6      0.0  -100 -  100 ---
CH7      0.0  -100 -  100 ---
```

Each channel here corresponds to a channel in the receiver. The limits you set on a line apply to that channel only.

Channel Output

At the top of the screen, in the middle, is a numeric value that corresponds to the pulse width output of the selected channel in microseconds (often written “usec” or correctly “µs”). The range of -100% to 100% will display as 988 to 2012 microseconds, with midpoint at 1500 microseconds.

In Er9x the limits can be extended to -/+ 125% by enabling the “E.Limits” (Extended Limits) option in *Model Setup, General* – see below. The pulse width will then range from 860 to 2140µs.

(Traditional transmitters generally use a significantly smaller 100% range of about 1100 to 1900µs, and allow extended throw of up to 150%, or 900 to 2100µs. When setting up a model, this difference must be considered: to achieve the equivalent of traditional 100% throw, set Er9x limits to 80%.)

Columns

Channel Number (first column)

Ranges from CH1 to CH16. Identifies channels as defined in the Mixer.

SubT – Subtrim (second column)

The values of SubT set the centre point of the channel and can range from -100 to 100. Increments are 0.1, giving fine resolution.

You can use the stick to set the centre point (Subtrim). While the SubT value is highlighted, hold the stick so the control surface is centred. Press [MENU LONG] and the position will be recorded. It's the magic of Er9x again ☺

Limits – Min/Max *(third and fourth columns)*

Limits values set the endpoints of the channel. Each limit can range between -100% and +100% (or -125% and +125% if extended limits are enabled – see below).

An arrow indicates which side of the midpoint is active; it changes as you move the stick or other input, indicating the limit value to adjust for that direction. With the Tx and Rx ON, you can move the stick to one end of its throw, then adjust the appropriate limit value to give the desired control surface movement.

If the “E-Limits” (extended limits) option is ON in the *Model Setup, General* menu, then it's possible to set a range between -125% and +125%. Since 100% on the Er9x firmware already corresponds to 125% on most other transmitters, extended limits should not normally be required.

Warning: When adjusting limits, please test to ensure that you don't cause unwanted mixing between channels or exceed servo mechanical limits.

The limits act both as gains and as absolute limits. They reduce output proportionally to ensure that servos do not move beyond the limits you set here.

Turning the AUTO LIMITS option ON in the *Model Setup, General* menu will allow Subtrim to override the Limits setting by a maximum amount of 10, in order to ensure that servo throw remains the same on both sides of centre. The modified values are displayed within a box.

Be careful when using this feature as the control limit values are no longer absolute and a servo could exceed its mechanical limit.

INV *Invert (fifth column)*

This reverses the output of the channel. Choices are "---" (i.e., normal) or "INV" (i.e., reversed).

Copy Trim

At the very bottom of the LIMITS menu is the option: COPY TRIM [MENU]. Press [UP] to get there quickly.

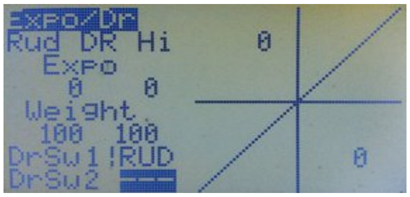
After you have trimmed your model to fly straight and level by using the trim buttons, and landed safely, you can select this command line and press the [MENU] button. The radio will beep, the trims will be converted into Subtrim values, and the trim positions shown on the Main Screens will all be returned to centre.

Note that "InstaTrim" (controlled by the "Trim Sw" selected under *Model Setup, General*) offers another approach to trimming by converting stick positions to Subtrim in flight (see page 68). The two methods are complementary.

Expo/DR

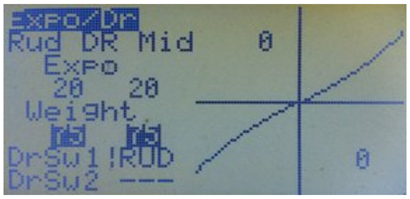
This screen allows you to enter and edit Expo and D/R (Dual Rate) values for the three main controls (Rud/Ele/Ail), as well as Throttle, though the latter is less common. As we will see below, "dual rate" doesn't tell the full story, as THREE different rates may be set, but for now let's deal with two.

The pictures at right show the effect of Expo and D/R. The first picture shows that with zero expo and 100% rate (Weight) the response of the rudder (y-axis) to stick input (x-axis) is a straight line at 45°. In other words, the servo will move through its full normal throw in direct proportion to the movement of the stick.



```
Expo/Dr
Rud DR Hi       0
  Expo
   0    0
 Weight
  100  100
DrSw1 !RUD          0
DrSw2  ---
```

The second picture shows how 20% expo changes the control response to a shallow S-curve, flatter (less sensitive) around the middle (neutral) and steeper towards the ends. At the same time, reducing the rate (Weight) to 75% makes the whole curve less steep. The total effect is to give smoother control, especially around neutral, at the expense of total control travel.



```
Expo/Dr
Rud DR Mid      0
  Expo
   20   20
 Weight
   75   75
DrSw1 !RUD          0
DrSw2  ---
```

In this example, the rudder D/R switch (RUD) is used to select between Hi and Mid rates (though any other switch could have been used). Here the switch action has been reversed by choosing the NOT state of the switch (!RUD). This involves simply pressing [LEFT] repeatedly rather than [RIGHT] to select the switch. The second switch option (DrSw2) is not used for a two-way expo/DR setup.

In the same manner, expo/DR setups can be created for elevator and aileron. The [LEFT/RIGHT] buttons are used to select between controls for editing.

Switching between Hi and Mid rate can either be assigned to a separate switch for each control (Rud, Ele, Ail), or all three setups can be controlled by a single switch (say ELE). For the latter, simply select ELE for all three of the controls.

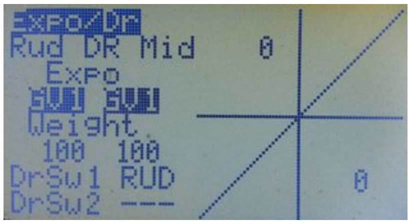
When adjusting Expo and Weight values, the numbers for the two control directions will normally move together, provided the stick is centred. To adjust them separately, hold the stick in the desired control direction. An example of such differential settings is that for balanced response a particular model may require more DOWN elevator than UP; accordingly, UP might, for example, be set to weight 80 with 25% expo while DOWN is set to 100 and with 0% expo.

To adjust Expo or Weight in steps of 1, highlight the relevant field and press [MENU]. The numbers will blink. To edit, use the [UP]/[DOWN], [RIGHT]/[LEFT] keys. Press [MENU]/[EXIT] when done.

To adjust Expo or Weight in steps of 20, highlight the relevant field and press and hold [MENU]. The numbers will blink. To edit, use the directional keys while continuing to press [MENU].

Using Global Variables

The usual fixed values for Expo and/or Weight can be replaced by Global Variables (GVARs). To do this, highlight the field and press [MENU] for around 3 seconds. Choose from GV1 to GV5 by using [UP]/[DOWN], [RIGHT]/[LEFT]. The result will be similar to the picture at right (provided the stick is in the middle during the process).

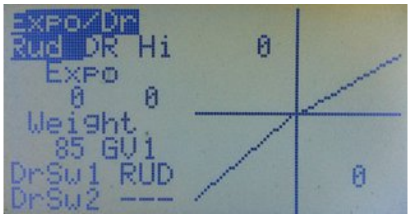


```
Expo/Dr
Rud DR Mid   0
  Expo
 GV1 GV1
 Weight
  100  100
DrSw1 RUD      0
DrSw2 ---
```

If the stick is held to one side while pressing and holding [MENU], only one side of the control throw will be affected. This is shown in the second picture at right, where Weight on one side remains as a fixed number, while the value on the other side is replaced by GV1.

Next go to the GVARs menu and set the source for GV1 to pot P1, for example. Once the GVAR source is defined, you will be able to see in the graphic representation how varying the value affects the shape of the control response curve.

To get back to fixed values, select the field(s) and press/hold [MENU].



```
Expo/Dr
Rud DR Hi    0
  Expo
   0   0
 Weight
  85 GV1
DrSw1 RUD      0
DrSw2 ---
```

If Expo and Weight are set to the same GVARs for rudder, elevator and aileron, then adjusting a GVAR value will change the corresponding setting for all of them. This can be done in flight and is very useful when experimenting with expo and rate settings. More on GVARs later (see page 54).

Triple Rates

As noted above, you are not limited to two expo/rate combinations. For each control you can set two switches, allowing you to select from three combinations. The main switch (DrSw1) changes between Hi and Mid values, with OFF (up) giving Hi rate. If DrSw1 is ON (down), the second switch (DrSw2) can select either between Mid or Low values. In other words:

Hi is DrSw1 OFF (DrSw2 doesn't matter)

Mid is DrSw1 ON, DrSw2 OFF

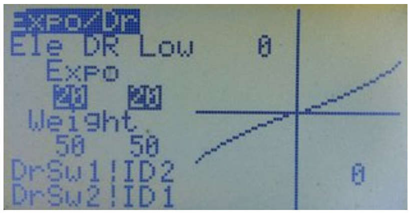
Lo is DrSw1 ON, DrSw2 ON

Note that “Hi”, “Mid” and “Low” are only names, you can set any rate you like for each of them.

The pictures below show a typical triple rate configuration. In this case, the three position switch is used instead of two separate two position switches. The positions are labelled ID0, ID1 and ID2.

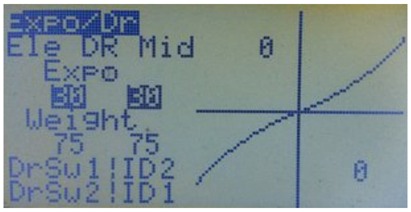
For DrSw1 enter “!ID2” (remember, exclamation mark means “not”) and for the DrSw2 enter “!ID1”.

The logic is that since DrSw1 is <u>not</u> ID2 it must be either ID0 or ID1. Likewise since DrSw2 is <u>not</u> ID1 it must be ID2 or ID0. Confused? Don't worry, it works.



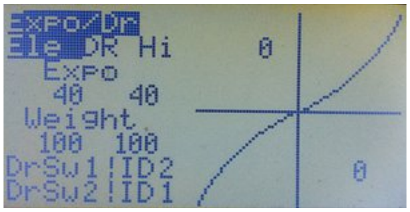
```
Expo/Dr
Ele DR Low     0
  Expo
   20   20
 Weight
   50   50
DrSw1 !ID2       0
DrSw2 !ID1
```

3 position switch is OFF (ID0)



```
Expo/Dr
Ele DR Mid     0
  Expo
   30   30
 Weight
   75   75
DrSw1 !ID2       0
DrSw2 !ID1
```

3 position switch is in the middle (ID1)



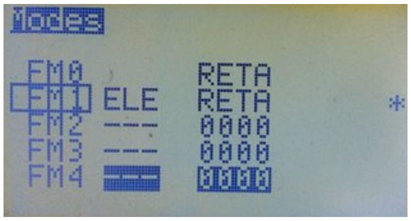
```
Expo/Dr
Ele DR Hi      0
  Expo
   40   40
 Weight
  100  100
DrSw1 !ID2       0
DrSw2 !ID1
```

3 position switch is on (ID2)

Modes *(Flight Modes)*

Five flight modes (FM) are available: the default mode FM0, plus FM1 to FM04. Each flight mode has a selectable activation switch (physical or logical), a trim selection array, and Fade-in/Fade-out parameters for smooth transitions between modes.

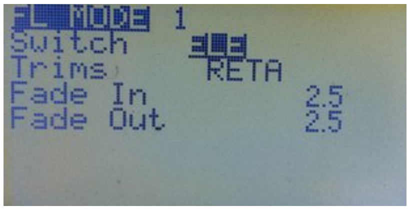
Flight Modes can also be used to determine which mixes are turned on for each mode. The Edit Mix menu (see below) allows the mode(s) in which a mix is active to be selected; by default a mix is active in all Flight Modes.



```
Modes

FM0          RETA
FM1  ELE     RETA      *
FM2  ---     0000
FM3  ---     0000
FM4  ---     0000
```

Perhaps the most common reason for using Flight Modes, rather than just programming the required features directly, is to allow different trim settings for specific model configurations, for example when flaps are deployed. Flight Mode 0 (FM0) is the default and always has its own trim settings. FM0 is active if no other flight mode is active.

Any other mode showing RETA (or some of those letters) has its own separate trim settings for those channels. Modes showing numbers share trims with another FM and thus use the same trim values as that mode. In the first picture, FM0 and FM1 have their own trims, while the other modes all use FM0 trims.



```
FL MODE 1
Switch       ELE
Trims         RETA
Fade In          2.5
Fade Out         2.5
```

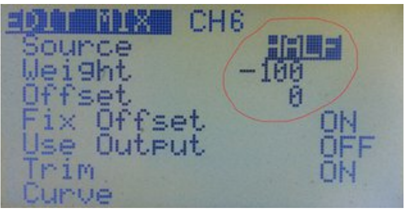
If more than one flight mode is enabled, then the lower numbered mode will be active.

A box is drawn around the name of the currently active flight mode. To change a flight mode setting, press [MENU] when the values of the mode you wish to edit are highlighted. This will display the editing screen. The name of the active flight mode will also be displayed in the all main screens, except for the telemetry ones.

The following example should help you understand the basics of Flight Modes. We will program flaps on channel 6 and control the mix by Flight Mode 1 (FM1) using the ELE switch.

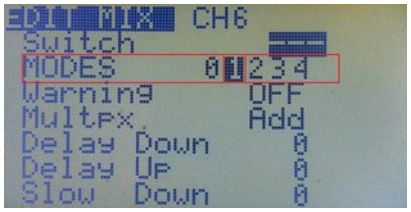
Start by creating a mix on channel 6 as shown at right:

CH6 -100% HALF



```
EDIT MIX  CH6
Source           HALF
Weight          -100
Offset             0
Fix Offset          ON
Use Output          OFF
Trim                ON
Curve
```

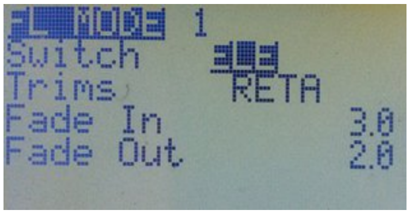
Now scroll down to the MODES option and unselect all flight modes except FM1. This means that this mix will be active ONLY when FM1 is active. In this case, because HALF is selected as source, the mix output will be -100% when the flight mode is active and 0% when it is not. This mix is now controlled exclusively by Flight Mode 1.



```
EDIT MIX  CH6
Switch           ---
MODES        01234
Warning         OFF
Multpx          Add
Delay Down        0
Delay Up          0
Slow  Down        0
```

Next go to the flight modes menu and edit FM1 by highlighting it and pressing [MENU]. Select ELE as switch and RETA as trims (i.e., the mode has its own trims). Also set values for Fade In and Fade Out to give a smooth transition between the two modes, FM0 (default-normal flying) and FM1 (Flaps down).

When you turn the ELE switch ON, the flaps will take 3 seconds to go to their down position; when you turn the switch OFF, the flaps will take 2 seconds to go up.

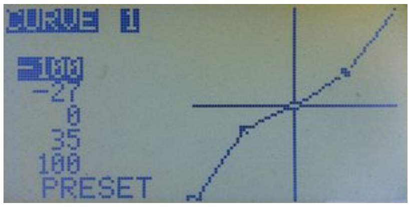


```
FL MODE 1
Switch       ELE
Trims       RETA
Fade In        3.0
Fade Out       2.0
```

When FM1 is active you can also trim the elevator to compensate for the effect of lowering for flaps. These trim settings belong only to this mode due to the “own trim” setting.

Curves

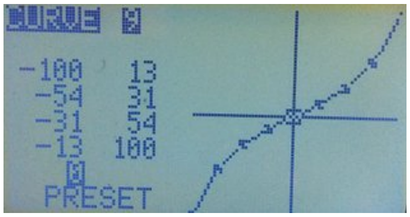
Curves tell the servo how to move in response to stick movement. In Er9x there are eight 5-point curves and eight 9-point curves. Curves CV1 to CV8 are 5-point, while CV9 to CV16 are 9-point.



```
CURVE 1
-100
 -27
   0
  35
 100
PRESET
```

A **5pt curve** can be edited at positions: -100%, -50%, 0%, 50%, 100%.

A **9pt curve** can be edited at positions: -100%, -75%, -50%, -25%, 0%, 25%, 50%, 75%, 100%.

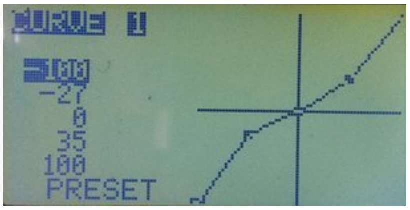


```
CURVE 9
-100   13
 -54   31
 -31   54
 -13  100
   0
PRESET
```

Scroll down to the curve you wish to modify and press [MENU] or [RIGHT]. This will bring up the screen for editing the curve:

You will see 5 or 9 editable points with the values initially set at zero. Scroll through the points with [UP/DOWN]. Change the value with [LEFT/RIGHT].

If you select the “PRESET” entry and press [MENU] to make it blink, pressing [LEFT] or [RIGHT] will populate the values with a linear curve (i.e., straight line) that is a good starting point. [LEFT] gives lines sloping up from left to right. [RIGHT] gives lines sloping the other way. Pressing the key repeatedly changes the slope.



```
CURVE 1
-100
 -27
   0
  35
 100
PRESET
```

The individual points can be edited by highlighting, pressing [MENU] and then using [RIGHT/LEFT] to adjust the value. Press [EXIT] to navigate to the next point.

The graph will show the shape of the curve as you edit the points.

Logical Switches

Logical Switches, previously called Custom Switches, produce an ON or OFF output based on a logical operation to compare values and/or assess logical conditions (e.g., which switches are ON). As virtual switches, they can be used in programming anywhere a switch can be selected. They will turn on when the logical condition being tested is verified (true).

The following examples will help to clarify the concept:

Example 1

A logical switch can be used to detect the position of an input such as the throttle stick. Suppose you want logical switch L1 to turn ON if the throttle stick value is above -96% (i.e., not at low throttle).

This is what it looks like:

```
L1  v>val   Thr   -96
```

"L1" is the switch number and "v>val" dictates the condition to be met for the switch to turn ON. In this case the source or variable (v) should be greater than (>) the value (val) you set, -96%. When the stick is above that value, the logical switch L1 will turn ON.

Example 2

Now, imagine that you want a mix in your programming to be ON only if two switches, RUD and GEA, are both ON. You need to program a logical switch as follows:

```
L5   AND   RUD   GEA
```

That means that the logical switch L5 will be ON only when both RUD <u>and</u> GEA switches are ON. If one or both of them are OFF, L5 will be OFF.

Example 3

Suppose you want to be able to ensure that a model does not exceed a certain altitude, say 400 feet. You want both automatic throttle reduction and an audible warning when the limit is reached. You also want to be able to turn off the limit mechanism with a switch on the transmitter.

Of course to do this you will need an altitude sensor and telemetry to send the information back to the transmitter. We'll assume you are using FrSky telemetry equipment, have done the telemetry mod to your transmitter and have installed the FrSky version of Er9x. (And, incidentally, that you have set telemetry to report in Imperial units – feet.)

With all this done, you can set up the necessary programming. First is a logical switch to respond when your aircraft is above 400ft, say L2:

```
L2   v>val  alt   400
```

Now we need to add the condition that an enabling switch, say ELE, must be ON. So we create a second logical switch, L3 that can be used to require that <u>both</u> the craft be above 400ft <u>and</u> the ELE switch be ON if action is to be taken to limit altitude:

```
L3   AND   L2   ELE
```

This means that L3 will be ON only when <u>both</u> L2 (craft over 400ft) and the ELE switch (enabling altitude limitation) are ON.

Now we use L3 to control a mix that stops the motor, like this:

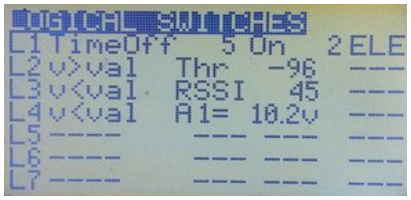
```
CH3   100%   Thr
  R    -100%   FULL   switch (L3)
```

This Replace mix in the throttle channel will stop the motor when the aircraft is above 400 feet AND the ELE switch is ON.

Other examples of automatic throttle control based on altitude are provided in the Annex A (page 71).

The Logical Switches Menu

Switch Number (Column 1) ─ The first column identifies the switch. There are 12 Logical Switches available in Er9x for the ATmega 64 processor, identified as L1..L9 and LA.. LC. With the m128 and m2561 processors, 18 Logical Switches are available: L1..L9 and LA..LI.



```
LOGICAL SWITCHES
L1 TimeOff     5 On    2 ELE
L2 v>val   Thr   -96   ---
L3 v<val   RSSI   45   ---
L4 v<val   A1=  10.2v  ---
L5 -----   ---  ---    ---
L6 -----   ---  ---    ---
L7 -----   ---  ---    ---
```

Operation/Condition (Column 2) ─ This column shows the operation or condition to be applied by the switch. The various arithmetic, logical, differential and timing operations are detailed below. Once the defined condition(s) is/are met, the logical switch will turn ON.

Source/ Value (Columns 3 and 4) ─ The use of these two columns varies according to the type of operation or condition involved. For arithmetic functions, Column 3 specifies the source of the variable and Column 4 gives the value (val) to which it is to be related. For logical conditions, the columns specify the switches involved. For evaluations the two variables (sources) to be compared are identified here. For timing, they specify duration ON and OFF.

Extra AND Function (Column 5) ─ If a switch is identified in this column, it acts as a further condition that must be met for the logical switch to turn ON. In other words, this switch functions as an "AND" condition in addition to the main condition of the logical switch.

Consider Example 3 above. It uses two logical switches, one to detect when altitude exceeds 400 feet, and a second to ensure that action will only be taken when (1) the model is above 400 feet AND (2) the ELE switch is ON. Using the extra AND function we can do the same using only one logical switch:

```
L2  v>val  alt  400  ELE
```

This sets up L2 to be used in the mix to stop the motor.

```
CH3   100%   Thr
  R    -100%  FULL   switch (L2)
```

Types of Logical Switch

There are several types of Logical Switch:

1. Regular Arithmetic Operations

These operations compare a variable/source (v) with a constant value (val):

v>val, v<val, |v|>val, |v|<val

In addition logical switches LD to LI (not available for the m64 processor) provide the condition v1~=val, which means that variable "v1" is approximately equal to the value "val".

The source of the variable, selected in Column 3 of the menu, can be a Stick, Pot, PPM input, Channel, Telemetry value, GVAR or Scaler.

The value, entered in Column 4 of the menu, is the test point for the condition. The possible range of values depends on which source is selected as the variable.

The following are examples of regular arithmetic operations:

L1 v>val Thr 0 (LS will be ON if throttle is above the midpoint – 0%)

L1 |v|<val Thr 10 (LS will be ON if throttle stick is within +/-10% of the midpoint)

L2 v<val A1 10.2V (LS will be ON when the voltage of a battery connected to A1 via a sensor is less than 10.2v.)

L3 v>val Alt 250m (LS will be ON when the altitude is greater than 250 meters.)

2. Logical Conditions

The operation assesses the state of the two switches identified in Columns 3 and 4 of the Logical Switches menu. The logical conditions which can be applied are:

AND, OR, XOR (exclusive OR).

In addition, two functions, Latch and F-Flop, are available (discussed below).

Available sources are the physical and logical switches, as well as the inverse of each of them.

The following are examples of operations involving logical conditions:

L2 OR RUD AIL (L2 will be always ON except when both switches are OFF.)

L2 AND ELE THR (L2 will turn ON if both ELE and THR switches are ON. Otherwise it will be OFF.)

L2 AND ELE !THR (L2 will turn ON only when the ELE switch is ON and the THR switch is OFF. Note the "!" on the THR switch, which means the inverse of the switch.)

L2 XOR RUD AIL (L2 will be ON when only one of the switches is ON; it will be OFF when both switches are either ON or OFF.)

3. Evaluations

Evaluations compare variables from two sources. There are four such operations:

v1==v2, v1!=v2, v1>v2, v1<v2.

The source of each variable can be a Stick, Pot, PPM input, Channel, Telemetry value, GVAR or Scaler.

The two sources to be compared, v1 and v2, are specified in Columns 3 and 4 of the Logical Switches menu.

For example:

L3 v1<v2 CH1 Rud (L3 is ON only if CH1 value is less than Rud value.)

L1 v1!=v2 Ele Rud (L1 is ON only if Ele stick value is different from Rud stick value)

4. Extra AND Switch

Logical Switches also offer an extra AND condition. If a switch is selected in the last column, it must be ON, in addition to the already programmed logical condition, for the logical switch to become active.

The extra AND condition can only be THR, RUD, ELE, ID0, ID1, ID2, AIL, GEA or L1..L7.

For example:

L4 OR THR !ID2 L1 (L4 is ON if either THR is ON or ID2 is OFF AND if L1 is ON.)

Timers can be controlled by this extra AND condition (see later item).

5*. Latch*

The Latch function is a type of logical switch that is useful where you wish to capture and report the value of an event (e.g., time reached, height reached, brief dip in battery voltage) and then to reset the value until a new event happens. The Latch function can be used anywhere a switch would work: to control a mix for example. Examples of Latch function applications are included in Annex A (see pages 77 and 81)

Two switches are used as input to the Latch function. Turning ON the first sets the logical switch output to ON. When the first switch is ON, the second switch has no effect, but when the first is OFF, turning ON the second will reset the logical switch to OFF. To restart the cycle, the second switch is turned OFF. For example:

L3 Latch TRN RUD

Logical switch L3 turns ON when the TRN switch is pressed and OFF if the RUD switch is turned ON (but only if the TRN switch has been released to OFF).

6. F-Flop (Flip-Flop)

The F-Flop function “remembers” the input state and provides it as the output. Again, two switches are used for input, one as the “trigger” and the other to provide the “data”. So, for example, if the momentary TRN switch is used as the trigger, pulling it ON sets the output of the logical switch to the current state of the second input.

One example of the use of F-Flop is “Toggle” mode, in which each activation of the trainer switch causes the logical switch to change state once between ON and OFF.

Here's how it is set up. The “state” of logical switch 4 (L4) changes every time the trainer switch is pulled, as the second input is defined as opposite to the current state:

L4 F-Flop TRN !L4

When you flick the trainer switch (the trigger), L4 will assume the state of the second switch, which is !L4 (its own inverse state). Every time you flick the switch, L4 will change state. Nice trick. 🙂

7. Timer

The timer function allows you to specify the time for which the switch is OFF and the time for which the switch is ON. The units are seconds.

The default value is 1 when you select the timer. By pressing the [RIGHT] key you increment the value in steps of 1 second. The maximum value is 100 seconds. By pressing the [LEFT] key, you increment the value in steps of 0.1 second to a maximum value of 5.0 seconds.

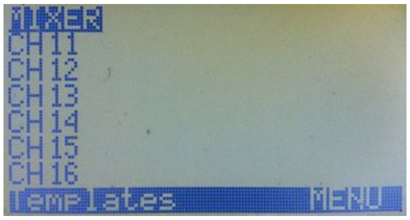
A timer can be subject to an AND condition specified in the last column. For example:

L4 TimeOff 5 On 0.5 THR

(L4 is ON for 0.5 seconds, then OFF for 5 seconds, but only if THR is ON. If the THR switch is OFF, the timer is stopped and the L4 switch remains OFF.)

Templates

Templates are starting points for model setups. When selected with [MENU LONG] they will either ***add to*** or ***replace*** the current model's mixes and settings with standard ones for a particular usage scenario. Best applied to a newly created model, they can be tweaked to achieve the desired result, or simply used to get an idea of what's required for the particular model type.



Templates are not available in the telemetry version of Er9x for the ATmega64 processor (Er9x-frsky.hex).

Note that the Templates menu is also available in the Edit Mix menu at the bottom of the list of channels, as shown at right.

There are nine templates:

01 Simple 4-CH

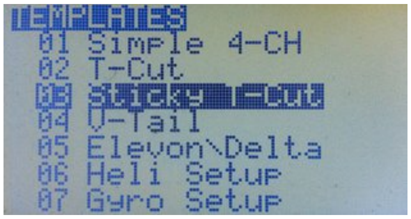
Creates a basic four channel fixed wing model with 100% weighting on each of the channels. Channel order is determined by the setting in *Radio Setup, Controls*. Note that applying this template overwrites any mixes already set for any channels in the mixer. The mixes produced by this template are identical to those used when a new model is created.

02 T-Cut

Inserts on the throttle channel a simple (S-type) safety switch controlled by the THR switch. When activated, this replaces the throttle channel output with a value of -100. For an explanation of Safety Switches, see page 51.

CH3 S THR -100

03 Sticky T-Cut



Inserts on the throttle channel a sticky (X-type) safety switch controlled by the THR switch. When activated, this replaces the throttle channel output with a value of -100. To regain control, the throttle stick must be in the low position.

CH3 X THR -100

04 V-Tail

Inserts mixes for both rudder and elevator input into the two channels used for the tail servos. When the elevator stick is pulled UP, both surfaces move up. When the rudder stick is moved RIGHT, the left surface goes up/right and the right surface goes down/right. Note that this mix may overwrite existing mixes.

05 Elevon\Delta

Similar to V-Tail but inserts mixes for aileron and elevator into the two channels that control the elevon servos. When the elevator stick is moved UP, both surfaces move up. When the aileron stick is moved RIGHT, the left surface goes down and the right surface goes up. Note that this mix may overwrite existing mixes. For more information on elevon and V-tail mix adjustment, see page 74.

06 Heli Setup

Sets up basic CCPM mixes for a collective pitch flybar-type helicopter with the common 120° swash arrangement (see Heli Setup, page 39). The three swash servo output channels are CYC1, CYC2, CYC3. Flight Modes are controlled by the 3-position switch. Curves c1, c2, c3 are used for throttle on CH5, while c4, c5, c6 are used for pitch on CH 11 (in this case, a “virtual” channel not sent to the receiver). Note that this mix may overwrite existing mixes.

07 Gyro Setup

Creates an adjustable output on CH6 controlled by potentiometer P2 and the Gear switch to set the gyro gain for a helicopter tail rotor. When the Gear switch is in the OFF position, moving P2 clockwise causes the output to range from 0 to 100; when the Gear switch is in the ON position, the output ranges from 0 to -100.

08 Servo Test (16)

Generates an output on channel 16 that moves slowly from -100 to +100 and back repeatedly. Action in each direction takes 7 seconds with a pause of 2 seconds between movements.

A logical switch (LB) is used as the timer:

LB Time OFF 9 ON 9

Note that you could also add an extra AND switch at the end of the logical switch line to turn the Timer on and off.

A mix on channel 16 controls the magnitude and speed of the output:

CH16 100% FULL switch (LB) slow (d7:u7)

To apply this output to a channel that has a servo connected to it, just set up a mix with CH16 as the source and a switch to turn the mix ON and OFF, like this:

CH6 100% CH16 switch (GEA)

If this mix is not the only one in the channel, then make it a Replace mix:

CH1 100% Rud
R   100% CH16 switch (GEA) (when Gear switch is ON, the servo will oscillate slowly)

If you want a fast switch-type movement, you can just use the logical switch from the template (LB) directly in the switch field of a mix:

CH7 100% FULL switch (LB)

The mix will switch channel 7 between -100% and 100% every 9 seconds. To change the interval, adjust the time values in logical switch LB.

You can use these inputs on as many channels as required to test servos. You can also use this template to have the servos move when performing a range check on your radio.

09 Servo Test (8)

The same as Template 08 but creates the mix on channel 8 instead of channel 16.

Warning: *To use either of the Servo Test templates to range test your equipment, it is recommended that you create a new model for that purpose. DO NOT use such a template on an existing model – if you flip the wrong switch in flight, you could have an interesting 9 seconds or so.* 🙂

Note: *The built-in templates cannot be edited or replaced. As a starting point for new models, you may wish to create generic models and store them either on the computer (using eePe), or in unused model memories of the transmitter. You can then copy, rename and adapt a model as required.*

Safety Switches

(Including Audio Alarms and Voice Switches)

The concept of the basic Safety Switch is that when activated, it overwrites (replaces) the existing output for a particular channel.

For example, you might want to work on your electric model with the battery connected and not have to worry that a programming error could start the motor at full throttle. You could set a “throttle lock” Safety Switch that replaces the output from the throttle channel with a fixed low throttle value.

Note, however, that such a switch does not make your radio foolproof! You must still be careful at all times and remove propellers and rotor blades from your models when working on them. But it can add another level of protection to reduce the risks associated with this hobby.

Safety Switch entries that are not required for the safety function may be used to control audio, haptic vibration (if available), and voice output (if the voice modification has been carried out).

There are four types of Safety Switch. When are activated by a physical switch such as THR they have the following results:

S – (Simple Safety): Replaces the existing value for a channel with a fixed “safe” value. Frequently used as a “throttle cut” switch, generally with a value of -100 on the throttle channel.

> Useful also when setting failsafe on certain receivers. For each affected channel, just program a safety switch to set your desired servo position. Use the same physical switch to control all these safety switches. When you turn the switch ON, all the channels will move to the correct position and you can press the button or do whatever else is needed to have the receiver record the failsafe settings. Examples of receivers for which this is useful include “D” series FrSky and Spektrum receivers that provide “Preset Failsafe.”

X – (Sticky Safety): Replaces the existing value for a channel with a fixed “safe” value and requires that the input (usually throttle stick) be in the “safe” position to resume control.

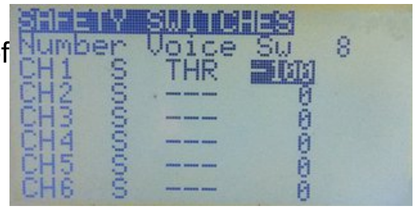
A – (Audio): Plays a sound or haptic vibration chosen from a list.

V – Voice): Plays a voice file.

Simple Safety Switches

To use the simple **S-type** safety switch, suppose you want to set a safety switch to lock the Electronic Speed Control (ESC) of an electric model at zero throttle. If Channel 1 is throttle, and full low throttle is -100, then what you need is:

```
CH1   S   THR   -100
```



```
SAFETY SWITCHES
Number Voice Sw   8
CH1  S   THR   -100
CH2  S   ---      0
CH3  S   ---      0
CH4  S   ---      0
CH5  S   ---      0
CH6  S   ---      0
```

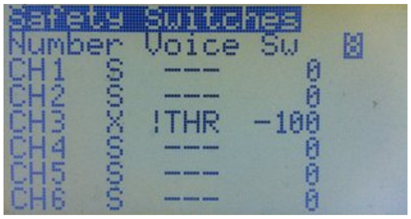
This means that when the THR switch is ON the safety switch ensures that the value of Channel 1 is always -100, regardless of the position of the throttle stick. When the THR switch is OFF the channel is again controlled by the throttle stick. In the picture, the first column is the channel to which the safety switch applies, "S" identifies this as a simple safety switch and "THR" identifies the physical switch that activates it.

Warning! When a safety switch is active, the output of the channel is replaced by the defined safety switch value (here -100) and the throttle stays closed. The value of the channel before it is processed by the safety switches, however, continues to be determined by the throttle stick position and/or any other mixes programmed in it. If another channel uses Thr as the source for a mix, it will not be protected by the safety switch.

If, for example, you were to have a mix to add down elevator as the throttle setting increases, using Thr as the source, cutting off the motor at full throttle with the safety switch would leave the elevator still in its down position. The solution is to turn on "Use Output" in the throttle-to-elevator mix and use the real throttle channel output (OPx) as source for that mix (see page 34).

It is also essential to recognize that a simple safety switch on the throttle channel does not fully protect you from unexpected motor starts. If you turn off the switch with the throttle stick anywhere other than in the safe (throttle closed) position, the motor will immediately start. To avoid this risk, you can program a "sticky" safety switch instead (see below).

Sticky Safety Switches



```
Safety Switches
Number Voice Sw   8
CH1  S   ---      0
CH2  S   ---      0
CH3  X  !THR   -100
CH4  S   ---      0
CH5  S   ---      0
CH6  S   ---      0
```

The **X-type** safety switch works exactly like the basic safety switch, except that it is "sticky". Before the channel can function normally again, the throttle stick must be returned to full low (below about -98%). In other words, the X-type safety switch performs the same function as the "Sticky T-Cut template" in earlier versions of Er9x and is much easier to program.

A sticky switch is generally desirable because it prevents the motor from starting unexpectedly if the throttle stick happens to be anywhere other than full low when the safety switch is turned off.

Note: *The following audio and voice alarms are still functional for the time being, but the use of the new "Voice Alarms" menu is highly recommended (see page 61). The alarms within the Safety Switches menu are retained only for compatibility with older model programming.*

Audio Alarms

The **A-type** safety switch (for Audio) plays, at 4-second intervals, a sound or haptic vibration chosen from the following list:

Warn1, Warn2, Cheep, Ring, SciFi, Robot, Chirp, Tada, Siren, AlmClk, Ratata, Tick, Haptc1, Haptc2, Haptc3.

Here's an example:

CH2 A AIL Ring

This will cause a double ring sound every four seconds whenever the Aileron switch is ON.

Voice Alerts

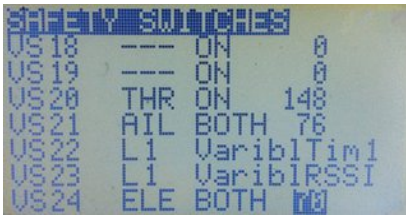
The **V-type** safety switch will play a voice file every 4 seconds when the switch is ON. For example:

CH6 V ID2 137

This will say “Flaps second position” when the three-position switch is full down. For full control of voice announcements, use a Voice Switch, as explained below.

Following the 16 Safety Switches (CH1 to 16) are, by default, eight Voice Switches (VS17 to VS24) that trigger the playing of a particular voice file and determine the conditions and manner in which it is played. If more than eight Voice Switches are required, Safety Switches can be converted into Voice Switches. At the top of the screen the value of “Number Voice Sw” can be adjusted to increase the default eight up to a maximum of all 24 switches.

The following picture shows the lower portion of the list, where Safety Switches give way to Voice Switches. If the number of voice Switches were increased to, for example, 10, then CH15 and CH16 would become VS15 and VS16.



```
SAFETY SWITCHES
VS18   ---  ON       0
VS19   ---  ON       0
VS20   THR  ON     148
VS21   AIL  BOTH    76
VS22   L1   VariblTim1
VS23   L1   VariblRSSI
VS24   ELE  BOTH    70
```

In this screen, the first column identifies the Voice Switch number.

The second column may contain any physical or virtual switch such as RUD, ELE, ID0, ID1, ID2, L1, L2, L3,...etc.

The third column may contain ON, OFF, BOTH, 15 sec, 30 sec, 60 sec, and Varibl.

ON: when the switch is activated, the item in the fourth column will be announced ***ONCE***.

OFF: when the switch is deactivated, the item in the fourth column will be announced ***ONCE***.

BOTH: both ON and OFF voice switches can be activated from one command line, provided the two voice files are consecutively numbered and are consecutively ON and OFF files.

Examples of Audio and Voice Alerts

Consider the following example of a Voice Switch used with the physical Gear switch:

VS26 GEA BOTH 76

This means that when the GEA switch is turned ON the transmitter plays File 76 “Gear up” and when the switch is turned OFF it plays File 77 “Gear down”.

When “Varibl” is selected in the third column, the voice will play a telemetry, GVAR, or scaler value specified in the next column.

The fourth column may contain a voice file number, a telemetry value, Timer 1 or Timer 2 value, or the value of a Global Variable (see page 54).

The A- and V-type Safety Switches can be used to program a telemetry battery alarm using a logical switch (see page 45):

First set a Logical Switch to detect the voltage of the battery by means of a sensor plugged into the A1 analog port of an FrSky receiver.

L1 v<val A1= 10.1V (this will turn ON if the voltage drops under 10.1v)

Then we can use the output of the logical switch to trigger the alarm in several ways:

First option uses an Audio alarm, which works even if the radio lacks a voice module. Choose an unused channel (normally a high number such as 13) in the Safety Switch menu and set:

CH13 A L1 warn1 (A is an audio safety switch and warn 1 is an audio alarm)

Second option uses a Voice switch that plays a voice file every 4 seconds if the logical switch stays on. Choose an unused channel (normally a high number such as 13) in Safety Switch menu and set:

CH13 V L1 31 (V is a voice safety switch and 31 is the voice file number)

Third option uses a Voice switch to play a voice file only ONCE every time the logical switch turns ON:

VS18 L1 ON 31

Global Variables and Scalers

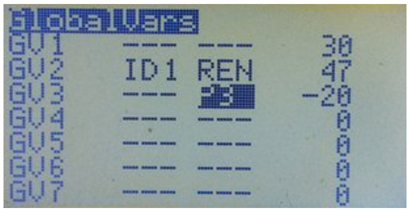
Global Variables

Global variables (GVARs) can replace the usual numbers for any Weight, Offset, Differential or Expo setting. Their main use is to allow the group adjustment of several parameters that should have the same value. In other words, a single global variable (such as GV1) can be inserted in several fields, allowing all of them to be adjusted simultaneously by changing the one GVAR value.

To use a GVAR in any of the fields mentioned above, just highlight the field value and press [MENU LONG] for about 3 seconds. The value will be replaced by "GV1". Then press [RIGHT] or [LEFT] to select the desired GVAR number. To change the value back to numbers press [MENU LONG] again.

GVARs can also be used as source for a mix, Scaler, Logical Switch, etc. They can be selected from the usual list of sources.

Seven GVARs are available to be configured in the "GlobalVars" menu. The picture shows the GVARS screen for a radio with ATmega 128 or ATmega 2561 processor. Note that the second column is not available on radios with an ATmega64 processor.



```
GlobalVars
GV1  ----  ----   30
GV2  ID1   REN    47
GV3  ----  P3    -20
GV4  ----  ----    0
GV5  ----  ----    0
GV6  ----  ----    0
GV7  ----  ----    0
```

In Column 3 any of the inputs shown in the table below can be selected, in which case the GVAR value shown in Column 4 will reflect the current value of the input. If Column 3 is left as "**---**", a constant (fixed value) for the GVAR can be entered in Column 4.

If Column 2 is available, any physical or logical switch (or its inverse) can be selected and used to control an input specified in Column 3. When the switch is ON, the input determines the GVAR value; when the switch is OFF, or if no switch is selected, the GVAR keeps its current value. A switch in Column 2 has no effect if Column 3 is left as "**---**".

Possible control inputs to be selected in column 3 are shown in the Table 1, below:

Table 1: Global Variable Inputs

Rtm	Rudder TriM	**Thr**	Throttle stick
Etm	Elevator TriM	**Ail**	Aileron stick
Ttm	Throttle TriM	**P1**	Pot P1
Atm	Aileron TriM	**P2**	Pot P2
REN	Rotary Encoder	**P3**	Pot P3
Rud	Rudder stick	**C1..C16**	Mixer Outputs (Channels 1 to 16)
Ele	Elevator stick		

The range of a GVAR varies with context. When replacing weight and/or offset values in a mix, when replacing Expo and/or Dual rate values and when used as source for a mix, the GVAR value is restricted to -100 to +100. When used as source for a Scaler or as variable in a Logical Switch, it can range from -125 to 125.

As shown below, in some places all seven GVARs may be used, while in other places only GVARs GV1 to GV5 are available.

Uses of GVARs

The source of a mix (GVARs 1-7)

When a GVAR is used as the source for a mix, its value (-125 to +125) is scaled to provide -100 to +100 input. One application is to use a trim switch to control a mix. For example, on many electric models and gliders the throttle trim is not used. If trim is disabled in the stick mix, the trim switch is available for use as the input to a GVAR, and that GVAR in turn may be used as the source for a mix.

Weight and offset of a mix (GVARs 1-5)

The weight and/or offset values for a mix may be replaced by GVARs. These GVARs can be adjusted in flight (by using pots as the sources, for example).

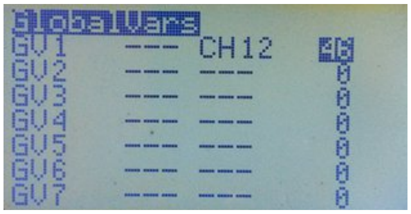
Expo and Dual Rate (GVARs 1-5)

The expo and/or dual rate values for one or more of the primary controls may be replaced by GVARs. This facilitates the fine tuning of control response by allowing the values to be adjusted in flight.

Examples of GVAR Use

1. Aileron differential on a glider with four surfaces that respond to aileron function

When trying to find the sweet spot for the differential value, instead of having to edit the differential value in the four mixers separately, all can use the same global variable. For example, GV1 can be selected by an extra LONG press of the [MENU] key on the Differential field for each control surface mix. Then adjusting GV1 using a pot and a virtual channel (say CH12), we can vary all four differentials simultaneously.



```
GlobalVars
GV1      ---  CH12   46
GV2      ---  ---     0
GV3      ---  ---     0
GV4      ---  ---     0
GV5      ---  ---     0
GV6      ---  ---     0
GV7      ---  ---     0
```

Here is the mix for the virtual channel:

CH12 -25% P3 offset 25% (Moving P3 clockwise will give a mix output from 50% to 0%.)

Now use CH12 as source for GV1. See picture at right.

With GV1 as source for the differential field of all four control surfaces, P3 can be used to set a differential value between 50% and 0%.

2. Using global variables in the Expo/DR menu

To provide adjustable expo, highlight the Expo value, then long press the [MENU] key for about three seconds. Select from GV1 through GV5 – say GV1. Then in the GlobalVars screen define the source of the GVAR as one of the potentiometers, say P3. Go back to the Expo/DR screen. As you adjust the potentiometer, the shape of the Expo curve will change.

Note that because the input from the pot ranges from -100 to +100, the expo value will do likewise. Normally, however, only positive expo values are used, in order to soften control response (flatten the curve) around neutral. It is advisable, therefore, to use the pot as input to a mix on one of the "virtual" channels (above those being transmitted) and to adjust that mix to produce a range from 0 to +100 as the pot is turned through its full range of motion. For example:

CH12 50% P3 offset 50% (Moving P3 clockwise gives a mix output from 0% to 100%)

Now use CH12 as source to GVAR1 (GV1) and use GV1 to adjust the Expo value.

If a GVAR is used for Weight in the D/R setting, only positive input values in the range 0 to 100 are used. Thus only half the pot range will change the weight, while the other half of the movement has no effect. Furthermore, it is risky to have any control that can set DR weight to zero (or even a very low value) on a primary control, as this would result in loss of control. Consequently, as in the above example, a pot should be used as input to a mix on a virtual channel, and the output of that mix should be restricted to a safe range of control rate for the model, such as 50% to 100%.

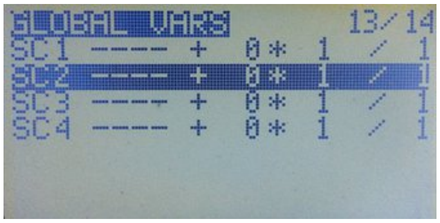
For example:

CH12 25% P3 offset 75% (Moving P3 clockwise gives a mix output from 50% to 100%)

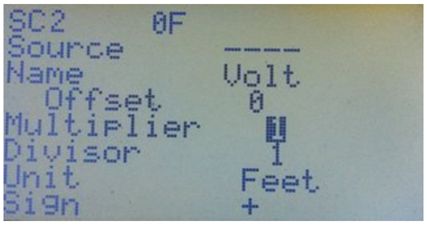
Now use CH12 as source to GVAR1 (GV1) and use GV1 to adjust the Dual Rate value.

Scalers

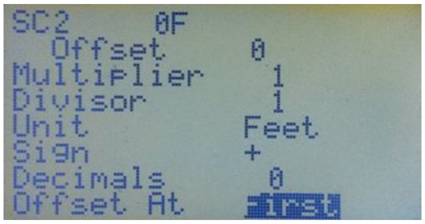
A scaler is a function that takes an input value and applies an offset and/or scaling factors (multiplication/division by fixed ratios). The result may be used for display on the custom telemetry screen, for voice output, as the source for a mix, or as a comparison value for a logical switch.



```
GLOBAL VARS          13/14
SC1 ---- +  0* 1  / 1
SC2 ---- +  0* 1  / 1
SC3 ---- +  0* 1  / 1
SC4 ---- +  0* 1  / 1
```



```
SC2      0F
Source          ----
Name            Volt
  Offset         0
Multiplier       1
Divisor          1
Unit            Feet
Sign             +
```



```
SC2      0F
  Offset        0
Multiplier       1
Divisor          1
Unit           Feet
Sign           +
Decimals        0
Offset At     First
```

For telemetry display purposes each scaler may be given a four character name. For both telemetry display and voice usage, the units may be selected, together with the number of implicit decimal places.

The parameters for a scaler are:

***Source* –** Stick/pot value, trainer input, channel output or telemetry value.

***Name* –** Four-character, user-edited name.

***Offset* –** -32,000 to +32,000.

***Multiplier* –** 1 to 256.

***Divisor* –** 1 to 256.

***Unit* –** Select from Feet, Volts, Deg_C, Deg_F, mAh, Amps, Metre, Watts.

***Sign* –** Select the sign of the result. **"+"** or **"-"**

***Decimals* –** Number of decimal places 0, 1 or 2.

***Offset At* –** Define whether the offset is applied First, or after the multiplier and divisor.

Telemetry

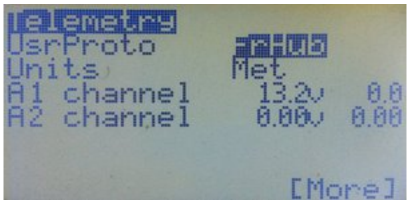
Telemetry is only available via the transmitter if you have modified the radio by re-routing the THR and AIL switches to another two pins on the processor as explained here. To avoid the need for the rather delicate soldering involved in that process, you can use the TelemetrEZ device from SmartieParts (where the solderless programmer board for the 9x comes from).

If your radio has the original ATmega 64 chip (as used in the stock 9x), you must flash the FrSky version of Er9x (Er9x-frsky.hex). If your radio has the ATmega128 or ATmega 2561, you must flash the corresponding Er9x version (Er9x-128.hex/Er9x-2561.hex) and set the option "Frsky Mod Done" to ON; you will find this setting in *Radio Setup, Hardware* (see page 21).

The Telemetry menu is found under the *Model Setup* index menu and consists of four screens.

Telemetry (Screen 1)

The first screen, shown at right, provides for selecting the telemetry protocol and units, as well as setting up analog inputs.



```
Telemetry
UsrProto          FrHub
Units             Met
A1 channel         13.2v   0.0
A2 channel         0.00v  0.00

                        [More]
```

The first item on the screen is "UsrProto", where you can use [MENU] to toggle between "FrHub" (*FrSky Telemetry Hub*) and "WSHhi" (*Winged Shadow How Hi*) telemetry protocols. The Winged Shadow unit has apparently been discontinued but many are still in use.

Next you can select either Metric or Imperial units using the same method. The picture shows "Met" for metric units.

"A1" and "A2" on this screen refer to the two analog inputs available on FrSky D-series receivers. These inputs can accept a maximum of 3.3v. To measure higher voltages, therefore, a voltage divider consisting of a pair of resistors must be used. For example, a divider with 4:1 ratio will measure voltages up to 13.2v (4 x 3.3v).

The “A1” input is used in most D-type receivers to measure receiver voltage by way of a built-in 4:1 divider. Others, like the D8R-II Plus, have an internal divider to measure the receiver voltage that can be connected to A1 using a jumper. When no jumper is in place, another voltage source can be measured using an external voltage divider.

The “A2” port does not provide an internal voltage divider. Consequently, to measure voltages above 3.3v, an external divider must be used as a voltage sensor. For example, to measure the voltage of a 4-cell LiPo battery (maximum 16.8v fully charged), a 6:1 divider can be used; this will accept voltages up to 6 x 3.3v = 19.8v. With such a divider, 16.8v will be divided by 6, becoming 2.8v as input to the receiver telemetry.

Warning: Do not connect a battery (or other power source) that supplies more than 3.3v directly to the analog ports A1 and A2 or they may be destroyed.

Note that the FrSky X8R receivers only have only one analog input, “A1”, which is connected internally to measure the voltage of the receiver power supply. A converter is available separately to enable analog voltage inputs (“A2”) to the digital S.Port of the X8R.

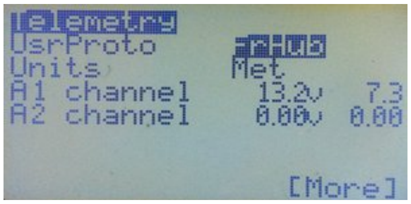
The X4R receiver has an external analog port (“A2”), and the X6R has a built in battery voltage sensor that can be connected to the “A1” port using a jumper, just like with the D8R-II Plus.

From the X6R manual:

> “With external analog telemetry port (AD1): “When pins of A1 and X (Middle Pin) are connected by the jumper, A1 will change from External Analog Telemetry Port to Internal Built-in battery voltage sensor.”

Within the telemetry circuits of the receiver, the value of the analog input (between 0 and 3.3v) is converted to a number between 0 and 255. This is encoded digitally and sent to the transmitter, where, by use of the divider ratio, it is converted back to a number representing the voltage.

In the screen shot at right, the numbers immediately to the right of “A1 channel” and “A2 channel” represent the measurement Range, which depends on the divider ratio of the voltage sensor. For the 4:1 ratio used to measure receiver voltage, this value should be set to 13.2 (3.3 x 4). For a 6:1 ratio sensor, the range would be set to 19.8 (3.3 x 6). The range is the maximum allowed voltage for that particular voltage divider. Higher than 19.8V and the voltage at the analog port will be over the maximum 3.3V, possibly destroying the receiver.



```
Telemetry
UsrProto          FrHub
Units             Met
A1 channel        13.2v   7.3
A2 channel        0.00v  0.00

                      [More]
```

The number in the right hand column represents the final Telemetry read-out (in the picture, 7.3V). If it does not agree exactly with the actual value measured with a voltmeter, the range value can be adjusted slightly. The usual cause of inaccuracies in the reading is the tolerance of the resistors used in the voltage divider.

The lower-case ‘v’ following the Range value represents one of four scaling factors used in converting the telemetry number (0-255) into a suitable read-out value. The ‘v’ setting corresponds to a read-out range of 0 to 25.5 volts, while a capital ‘V’ setting gives 0 to 51 volts (at reduced resolution).

Set to ‘A’ for current measurement, the telemetry number is converted to a range of 0 to 65 Amps.

Finally, the ‘-‘ setting displays the unconverted telemetry number (0-255), which can be used for parameters other than voltage or current, such as temperature or RPM, depending on the sensor.

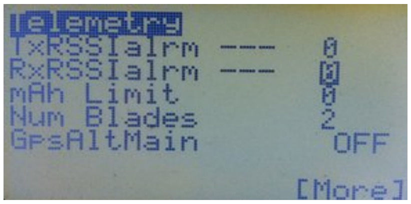
The “A2” read-out works in the same fashion. The Range setting will depend on the sensor being used. For the FBVS-01 voltage sensor, using the default ratio of 6:1, the range setting would be 19.8v. Fuel level, GPS position, Altitude, Current and RPM may also be measured using FrSky sensors which plug into a hub. The hub produces a stream of signal outputs suitable for the serial port on some D-series receivers.

When using third party sensors, the scaling factor of the voltage divider depends on the type of sensor installed and the magnitude of the values being measured. Refer to the instructions which come with the sensor for details on how to install and calibrate it.

Voice (if available) or audio warnings can be programmed for “A1” and “A2” values in the Voice Alarms menu (see page 61).

Telemetry (Screen 2)

This page, shown at right, continues the telemetry settings.



```
telemetry
TxRSSIalrm ---   0
RxRSSIalrm ---   0
mAh Limit        0
Num Blades       2
GpsAltMain      OFF
                [More]
```

RSSI Alarms

The “TxRSSIalrm” and “RxRSSIalrm” settings are those stored on a FrSky DJT or DHT module. They are read from the module for display. If they are changed in the menu, then the module is updated with the new values. They are therefore common to all models. Yellow causes a single beep, Orange causes a double beep and Red causes a triple beep from the module. This only works if you have a hardware modification providing a full bi-directional serial interface to the module.

mAh Limit

This setting causes an audio alarm (selectable) to sound when the used capacity of your flight battery exceeds the set value. (This feature may be removed in a future release. It is now recommended to use the Voice Alarm menu to program an audio, haptic or voice alarm).

Num Blades

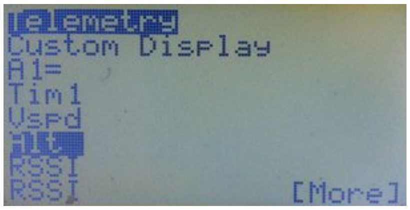
Setting for the RPM sensor. Number of blades, 1 to 127.

GpsAltMain

This displays the Global Positioning System (GPS) determination of altitude and may be turned ON or OFF. If ON, the barometric altitude reading is replaced by GPS altitude (this feature may be removed in a future release of the firmware).

Telemetry (Screen 3) – Custom Display

Short presses of the [DOWN] key will take you to the custom telemetry display screen.



```
telemetry
Custom Display
A1=
Tim1
Vspd
Alt
RSSI
RSSI            [More]
```

Up to six telemetry sources can be displayed on a single screen in the order you select. Highlight the empty field, and using the [LEFT] or [RIGHT] keys select from the options available.

The possible telemetry sources are shown in Table 3 below.

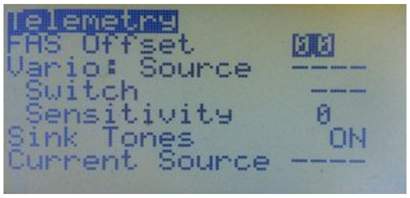
Table 3: Telemetry Sources

A1 -	'D' series Rx A1	**Batt -**	Tx battery
A2 -	'D' series Rx A2	**Amps -**	FAS current (FrSky Ampere Sensor)
RSSI -	RSSI	**Mah -**	FAS Capacity (mAh)
TSSI -	'D' series Tx RSSI, 'X' series Tx SWR	**Ctot -**	Sum of cell voltages
Tim1 -	Timer 1	**FasV -**	FAS voltage
Tim2 -	Timer 2	**AccX -**	Acceleration X
Alt -	Barometric Altitude	**AccY -**	Acceleration Y
Galt -	GPS Altitude	**AccZ -**	Acceleration Z
Gspd -	GPSspeed	**Vspd -**	Barometric Vertical speed
T1 -	Temperature 1	**Gvr1..7 -**	Global Variables 1 to 7
T2 -	Temperature 2	**Fwat -**	Watts from FAS (V * I)
RPM -	RPM	**RxV -**	'X' series receiver voltage
FUEL -	FUEL	**Hdg -**	GPS Heading
Mah1 -	mAh from A1 used as current sensor	**A3 -**	SPort A3
Mah2 -	mAh from A2 used as current sensor	**A4 -**	SPort A4
Cvlt -	Cell Voltage (lowest)	**SC1..4 -**	Scaler 1 to 4

Telemetry (Screen 4)

FAS Offset

FAS stands for FrSky Ampere Sensor. These units come in several sizes. Some FAS show a small reading when no current is really flowing; the FAS Offset field may be used to 'zero-out' this error.



```
Telemetry
FAS Offset          0.0
Vario: Source      ----
 Switch             ---
 Sensitivity          0
Sink Tones           ON
Current Source     ----
```

Vario

The S-port enabled FrSky variometer sensor is also configured here using the following settings:

Source – The Source may be either “vspd”, the value passed directly over telemetry, or the “A2” port if using the analog value.

Switch – Vario tones may be enabled by means of a switch, the specific switch is selected by the Switch entry. The value includes “ON” to allow the vario tones to be permanently enabled. The vario tones are short bleeps that sound more frequently the higher the climb or sink rate.

Sensitivity – The sensitivity for sounding the tones may also be set. The higher the value, the less the sensitivity. Typical values are 2 for “vspd” and 40 for “A2”.

Sink Tones

Option to turn OFF the sink tones. Some might find the sink tones annoying. I do. 🙂

Current Source

Possible Current Sources are A1/A2, FAS, and Scalers 1 to 4.

A Note on the openXsensor Project

openXsensor (oXs) is an open source project based on an Arduino platform. It can connect to various sensors and transmit the data via FrSky telemetry back to the transmitter. oXs enables you to build sensors to match your particular needs. You can find the oXs project page here.

oXs is compatible with both the FrSky D-series receivers (HUB protocol) and the newer X-series Rx (with Smart Port protocol).

Some of the measurements supported by oXs are:

Altimeter/Variometer (similar to FrSky Variometer high precision sensor)

Uses MS5611 pressure sensor. A second barometric sensor can also be used.

Air Speed

Can be used by oXs to compensate vertical speed measurements (dTE)

Voltage (up to six measurements)

Battery voltage (like FrSky FBVS-01), LiPo cells (like FrSky FLVS-01 or FLVSS)

RPM (for brushless motor, optical sensor, magnetic sensor)

Current (similar to FrSky FAS or FCS)

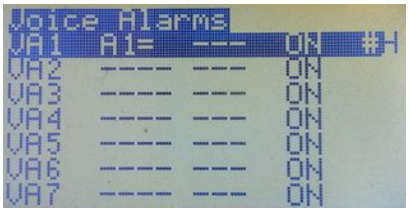
Can calculate the consumed current. oXs can use any sensor outputting voltage.

Temperature

Voice Alarms

This feature requires that a suitable voice module with SD Card be installed in the radio. Haptic (vibration) alarms require a further modification.

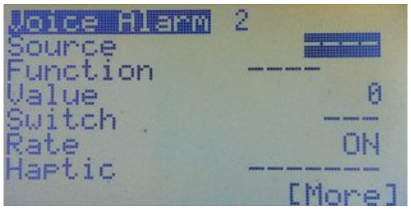
Voice Alarms provide a single powerful menu that allows you to program all your voice, audio and haptic (vibration) alarms. They can be used to signal a wide variety of events and conditions, from warning of low battery or high altitude, to announcing “Gear up!” or providing a spoken read-out of telemetry values, to making the transmitter beep or vibrate when it's time to land.



```
Voice Alarms
VA1   A1=   ----   ON    #H
VA2   ----- ----   ON
VA3   ----- ----   ON
VA4   ----- ----   ON
VA5   ----- ----   ON
VA6   ----- ----   ON
VA7   ----- ----   ON
```

Voice Alarms supersede all the older means of setting alarms.

Previously, to create an alarm it was necessary to program a logical switch to detect, say, a telemetry event like a battery pack voltage, then set a condition to trigger the switch, and finally use the output in the Safety Switches menu to create an audio, voice and/or haptic alarm. It worked well but was cumbersome.



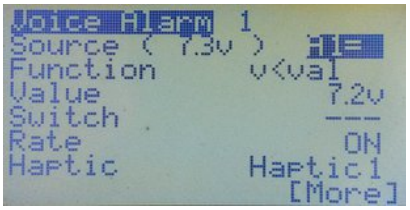
```
Voice Alarm  2
Source              ----
Function       ----
Value                  0
Switch               ---
Rate                  ON
Haptic         --------
                  [More]
```

Er9x provides eight of the new Voice Alarms: VA1 to VA8.

The pictures show how a voice alarm looks when you start a new one. The various fields not only replace the separate Logical Switches, Safety Switches and Voice Switches required previously, but they provide additional options for greater flexibility.

Source

Here you can select from all the sources: Sticks, Potentiometers, Channels, PPM inputs (Trainer Port), Scalers, Telemetry, GVARs, etc. These, combined with a function, can be used to generate alarms, or just let the nice lady that lives inside your radio tell you the source values. We'll see some examples later. If a source is selected, its current value will be displayed to the right of the word “Source”. In the picture, it's “(7.3v)”.



```
Voice Alarm  1
Source ( 7.3v )   A1=
Function      v<val
Value          7.2v
Switch          ---
Rate             ON
Haptic      Haptic1
            [More]
```

Function

Here we set a logical condition that compares the source value with another value in order to verify if the condition is true or not. It is, in effect, a logical switch. For example:

> Imagine we have analog Port 1 (A1=) as source, and we want to play an alarm if the voltage drops below a certain level. We select the function “v<val”. This means that IF the voltage on A1 (v) is smaller than (<) the value we set (val), here 7.2v, then the condition will be met (TRUE). The “function switch” will turn on, triggering the playing of a voice or audio file and/or vibration.

Value

This is the set value to which the source value is compared by the logical function. If the logical condition is verified, the “function switch” will turn on, triggering voice, audio and/or vibration events.

Switch

A physical or logical switch can be used to play a source value, like telemetry, GVAR, or stick position, or to play voice files when turning switches ON or OFF. This field is normally left at “---” for alarms.

> Examples: Flaps up, Flaps down, Gear up, Gear down, etc.

You can select any of the physical switches and the logical switches as well as their inverse.

Rate

This field allows us to set how the Voice/Audio/Haptic alarm will be played.

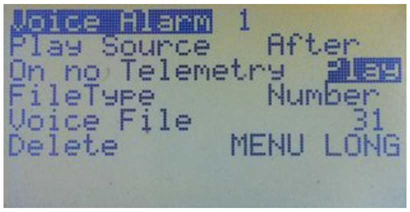
> **On –** The event/file will be played once when either the physical/logical switch is ON, or the “function switch” is ON (logical condition is TRUE), depending on what is being used.
>
> **Off –** The source, file, or alarm will be played once when either the physical/logical switch is OFF or the “function switch” is OFF (logical condition is FALSE), depending on what is being used.
>
> **Both –** This plays one file once when the switch is turned ON, and the consecutive file once when the switch is turned OFF.
>
> > For example, suppose you have two files. Flaps1 and Flaps2. You select Flaps 1.
> >
> > When you turn the switch ON, Flaps 1 will be played. The second file (Flaps 2), which must be the consecutive file number on the SD Card, will be played when you turn the switch OFF.
>
> **1,2 ,3 ,4... –** This repeats a file or event every “n” seconds, from 1 to 30 seconds.

Haptic *(requires hardware modification)*

This option causes the Tx to vibrate when voice or audio alarms or events are sounded. Select one of the three preset vibration patterns: Haptic1, Haptic2, Haptic3.

Play Source

As the name says, this option will play the source value, in the example used earlier the "A1" telemetry voltage, either before or after the selected voice file/audio.



```
Voice Alarm 1
Play Source      After
On no Telemetry   Play
FileType        Number
Voice File          31
Delete       MENU LONG
```

Before we had the Voice Alarms menu, two voice switches were required to say, for example:

"Flight pack voltage" and then the telemetry value "Seven point three".

Now with the Play Source option, we can do the whole announcement in one menu. The options for Play Source are:

Before – the source value will be played before the file.

After – the source value will be played after the selected voice file.

No – The source won't be played. Only a voice file, audio or haptic will be played.

On no Telemetry

When this option is set to PLAY, the radio will announce "No Telemetry" if a telemetry event (source) is triggered but the Tx is not receiving a valid telemetry signal.

Some people find this annoying so you have the alternative of MUTE. You decide. 🙂

File Type

Here you select the type of alarm by pressing the [LEFT/RIGHT] keys. The options are "Number" to play a voice file, "Audio" for a warning sound file, or "------" for no audio or voice.

Voice File

If the File Type is set to Number, you will be able to select the voice file number in this field by pressing [LEFT/RIGHT]. Pressing [MENU LONG] when the file number is highlighted will cause the radio to play the voice file (so you can check that you've got the right one).

If the File Type is set to audio you can select some pre-programmed sounds. These are generated internally by the Er9x firmware. The options are:

Warn1, Warn2, Cheap, Ring, SciFi, Robot, Chirp, Tada, Crickt, Siren, AlmClk, Ratata and Tick.

Delete

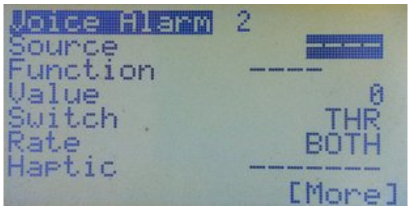
Press [MENU LONG] to delete the voice alarm. All fields will reset to their default values.

Examples of Voice Alarms

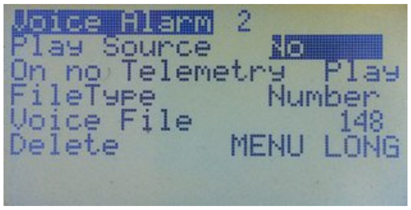
Example 1 – Normal Voice Switch

This example shows how to program a normal voice switch using a two-position switch, THR, to play two files. One when the switch is turned ON, and the other when it is turned OFF. For this, we will use the BOTH option. That means that the first file will be played when we turn the switch ON, and the consecutive file number on the SD Card plays when we turn the switch OFF.

The first file is the one we select when using the BOTH option. In this case File number 0148 (Throttle enabled). This will play when we turn the switch ON. The next one, 0149 (Throttle disabled), will be played when we turn the switch OFF.



```
Voice Alarm 2
Source          ----
Function        ----
Value              0
Switch           THR
Rate            BOTH
Haptic      --------
              [More]
```

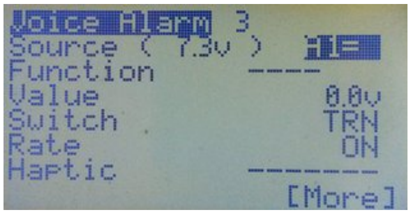


```
Voice Alarm 2
Play Source       No
On no Telemetry Play
FileType      Number
Voice File       148
Delete     MENU LONG
```

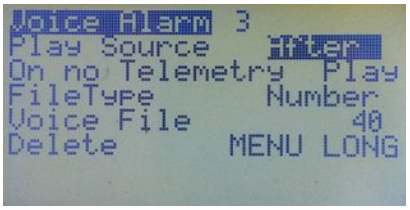
Example 2 – Play a Source Value (Telemetry) after a Voice File

Here, when the switch “TRN” is turned ON, the source value “A1=” will be played once (Rate is set to “ON”), AFTER the Voice file number 0040 (Pack Voltage). If the voltage on A1= is for example 11.5V, the Tx will say: “Pack Voltage, Eleven Point Five Volts”

Experiment with different settings and values to understand what every option does.



```
Voice Alarm 3
Source ( 7.3v )  A1=
Function        ----
Value           0.0v
Switch           TRN
Rate              ON
Haptic      --------
              [More]
```

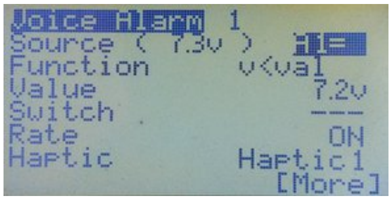


```
Voice Alarm 3
Play Source    After
On no Telemetry Play
FileType      Number
Voice File        40
Delete     MENU LONG
```

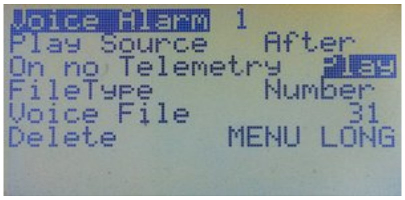
Example 3 – Voice Alarm with Haptic (Vibration)

This requires that a haptic module be installed in the transmitter.

Here we have a voltage alarm. We want the TX to warn us if the voltage drops under 7.2v by playing a voice alarm followed by a voltage report (Play the Source Value).



```
Voice Alarm 1
Source ( 7.3v )  A1=
Function       v<val
Value           7.2v
Switch           ---
Rate              ON
Haptic       Haptic1
              [More]
```



```
Voice Alarm 1
Play Source    After
On no Telemetry Play
FileType      Number
Voice File        31
Delete     MENU LONG
```

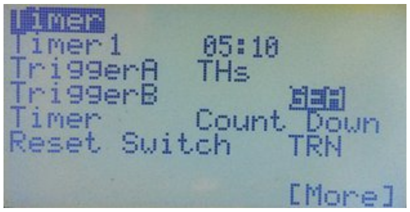
The voice file number 0031 (flight battery low) will play once (Rate is set to “ON”), if the source (A1=) drops below 7.2V (v<val = A1<7.2V), followed (AFTER) by the voltage report (Play Source). The radio will vibrate at the same time (Haptic1).

The complete phrase will be:“Flight Battery Low, Seven Point Two Volts”. Try it.

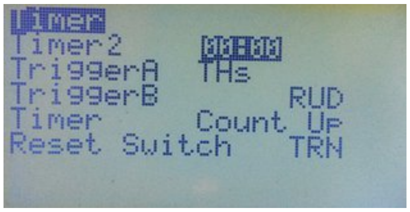
Timer

Er9x provides two fully programmable timers that can count either up or down. By default, the timers are set to OFF and are not operational. Timer 1 is on the first page of this menu, Timer 2 on page 2.

To set the time in steps of one second, highlight the field and press [MENU]. The field will blink. Use the [UP]/[DOWN]/[RIGHT]/[LEFT] keys to edit the time.



```
Timer
Timer1       05:10
TriggerA     THs
TriggerB       OFF
Timer   Count Down
Reset Switch   TRN
            [More]
```



```
Timer
Timer2         00:00
TriggerA       THs
TriggerB            RUD
Timer          Count Up
Reset Switch        TRN
```

To set the time in steps of one minute highlight the field and press and hold the [MENU] key while setting the time.

Press [MENU]/[EXIT] when done.

Two triggers are available to start and stop each timer.

Trigger A

Options for Trigger A are OFF, ON, THs, TH%, or c1% to c16%.

OFF – Timer is OFF.

ON – Timer is on, always counting.

THs, TH% – Select one of these to activate the timer based on the throttle stick position. When THs is selected, the timer starts whenever the stick is not at -100. The TH% option is the same except the timer speed is determined by stick position. When the stick is at -100, the timer is stopped. When at +100 the timer runs at normal speed. When midway the timer's speed reflects the stick position.

c1% to c16% – These provide the proportional output of any channel from 1 to 16. When the channel is at -100, the timer is stopped. When at +100 the timer runs at normal speed. When midway the timer's speed reflects the channel position.

Trigger B

Trigger B acts as an "AND" switch to “Trigger A” (see above). That means you need two conditions for the Timer to start counting. If a Trigger B switch is defined, then it needs to be ON for Trigger A to work.

Options for Trigger B are any physical switch, any logical switch, any momentary switch and the inverse of any of them.

Switches – You can specify any physical switch or logical switch (see page 45) so whenever that switch is activated the timer counts. For example, if you want the timer to count when the ELE switch is on, set Trigger A to ON and Trigger B to ELE.

To select a physical switch you can, instead of using the keys, just highlight the switch field and physically move the switch to the required position; the firmware will detect the action.

Momentary switches - A switch plus “m” (like TRNm) means “momentary”. Moving the switch once to the ON position and back turns the timer ON. Moving it ON and OFF again turns the timer OFF. Try it with the TRNm switch. Flick the switch to toggle the timer ON and OFF.

Difficult I know, take a sip from your coffee, relax a bit, take a break, it's easy once you try it. 🙂

Timer

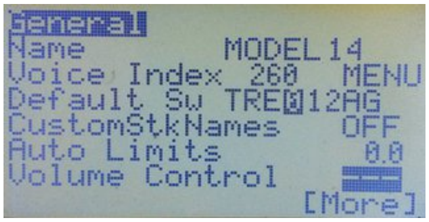
Here you can choose whether the timer counts up or down. Press [MENU] or [RIGHT]/[LEFT] to toggle between “Count Up” and “Count Down”

Timer Reset Switch

This resets the timer to its set value. Options are any physical switch, any logical switch, any momentary switch and the inverse of any of them.

For example, imagine you set the TRN switch as the timer reset switch, and the timer is set to count down from 10:00 (ten minutes) with the throttle stick as trigger (THs). At any point if you hit the TRN switch, the timer will be reset to 10:00. If it is counting when you reset it, it will resume counting from the beginning.

General



```
General
Name          MODEL14
Voice Index  260   MENU
Default Sw  TRE012AG
CustomStkNames    OFF
Auto Limits        0.0
Volume Control    ---
                [More]
```

Name *(Model Name)*

This item works like Owner Name in the Radio General Settings menu. To edit, scroll down until the first letter or space in the name is highlighted. Press [MENU] and the highlighted character will start flashing. You can now edit it using the [LEFT] and [RIGHT] keys. The following are shown in sequence:

> Blank space, upper case letters, lower case letters, numerals, plus punctuation – underscore, short dash and period (_, -, .).

When you have the correct letter or numeral in place, press [MENU] or [EXIT] to regain the ability to move left and right. Once you have finished, press [DOWN] to get to the next menu item.

Voice Index (Only works if a voice module is installed)

Here you specify the voice file that will be announced when the model is selected. Highlighting the number and pressing [MENU] will cause the corresponding voice file to be played. To edit it just press [LEFT] or [RIGHT] to change the file number. The range of numbers for the models voice files is **260** to **309**. If you don't want a name to be announced, just assign a number that doesn't have a voice file associated with it (probably a high number).

Default Sw (Default Switch Positions)

Sets the default switch positions at start-up. Highlight the field, set the switches in the desired positions, and then press the Menu key. The switches that default in the ON position will then be highlighted.

CustomStkNames

This enables the use on this model of the custom stick names you have defined in the *Radio Setup, Controls* menu (see page 20). Use the [MENU] key to toggle this feature ON or OFF.

Auto Limits

This setting applies to the Limits menu (page 40) by allowing Subtrim to override the Limits setting so that servo throw remains the same on both sides of centre. The value here determines the maximum amount the limits can be exceeded (e.g., a setting of 10.0 allows a limit set at 100% to go as high as plus/minus 110%).

Be careful when using this feature as the control limit values are no longer absolute and could allow a servo to exceed a mechanical limit.

Volume Control *(Only works if a voice module is installed)*

Sets a specific control to adjust the audio volume, typically one of the three pots or a Global Variable.

E. Limits *(Extended Limits)*

Allows control limits to go to plus or minus 125% instead of the normal 100% maximum. Note that 100% on the Er9x firmware already corresponds to approximately 125% on most other transmitters, so extended limits should not normally be required.

Warning: Please test to make sure this does not cause unwanted mixing between channels or exceed servo mechanical limits.

Throttle Default

When this option is set to "End", the throttle-closed stick position is defined as full DOWN, unless the throttle is reversed (see the next section and page 18). This is the default for Er9x and the normal setting for aircraft.

When set to "Centre", the throttle-closed position becomes the middle of the stick range, allowing stick movement in one direction for forward control and in the other for reverse or braking, as may be required for land or water vehicles. A centre-off setup generally uses a spring-centred throttle stick.

With either setting of this option, the Throttle Cut and Sticky Throttle Cut feature in Safety Switches (page 51) and Templates (page 49) work correctly, as does the Throttle Warning alarm (page 17).

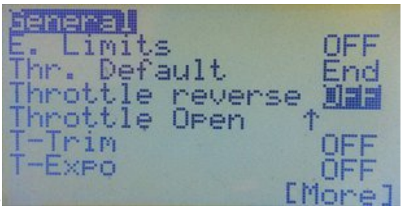
Throttle Reverse

When turned ON, the Throttle Reverse option here in *Model Setup* inverts the operation of the throttle stick for a single model, relative to the global Throttle Reverse setting in *Radio Setup* (see page 18). Whenever this option is changed, a Throttle Alert warning will be shown unless the stick happens to be in the new throttle closed position.

Note that the throttle stick direction for an individual model reflects the interaction of the two Throttle Reverse settings (global and individual). If one is ON and the other OFF, throttle will be reversed. If both are the same (either ON or OFF) the throttle action will not be reversed; it will operate according to the Er9x default.

This option reverses the throttle stick action and is NOT to be used as a means of reversing the output signal to a throttle servo or electronic speed control (that should be done using Invert in the Limits menu).

To make clear which way the throttle is operating for the current model, a **direction indicator** is provided in the next line. This has a vertical arrow which points in the direction of HIGH or OPEN throttle.

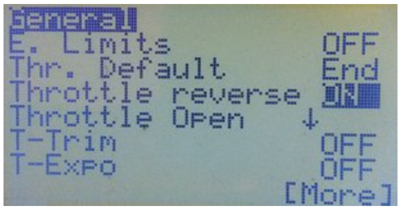


```
General
E. Limits              OFF
Thr. Default           End
Throttle reverse       OFF
Throttle Open       ↑
T-Trim                 OFF
T-Expo                 OFF
                    [More]
```

The top picture at right shows normal (default) throttle direction with open throttle in the UP (away from operator) position. The lower picture shows reverse throttle, with open throttle DOWN (towards operator).

The arrow is purely an indicator and cannot be changed directly.

Warning: Use this feature with extreme caution, especially with electric models. Except when the throttle stick action needs to be reversed, both global and individual model Throttle Reverse options should be OFF.



```
General
E. Limits              OFF
Thr. Default           End
Throttle reverse       ON
Throttle Open       ↓
T-Trim                 OFF
T-Expo                 OFF
                    [More]
```

T-Trim *(Throttle Trim)*

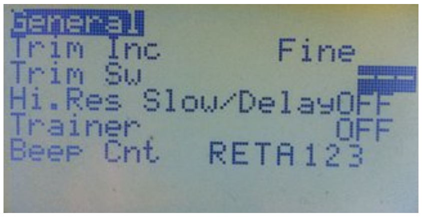
This is a useful feature for fuel-powered models. When turned ON, (1) the centre detent for the throttle trim is removed, and (2) throttle trim affects only the “low” side of the throttle range. That means you can use the trim for setting idle while full throttle remains unchanged. When OFF, throttle trim behaves like trim on other primary controls.

T-Expo

When ON, this option causes throttle stick expo to apply from zero to full throttle, instead of having a centre point like the expo setting on the other stick axes.

Trim Inc *(Trim increments)*

This setting determines how the active trims behave.



```
General
Trim Inc      Fine
Trim Sw        ---
Hi.Res Slow/DelayOFF
Trainer        OFF
Beep Cnt  RETA123
```

Exp – Exponential – trims are fine near the centre and get progressively coarser further out.

ExFine – Extra fine – 1 step per click.

Fine – 2 steps per click.

Medium – 4 steps per click.

Coarse – 8 steps per click.

Trim Switch *(“InstaTrim”)*

When this switch is activated, the current position of the sticks is copied to the Subtrim settings (Limits menu); hence the sticks must immediately be neutralized. Choose a switch that is easily reached while holding the sticks but unlikely to be flipped accidentally.

This function is very useful for first flights, as it avoids the need to take your hands off the stick to press the trim buttons. Simply hold the plane level with the sticks and flick the selected Trim Switch. Immediately centre the sticks and the plane should be roughly trimmed. Fine trimming can still be done with the regular trim buttons, if necessary.

The function is triggered by moving the switch from its default position. If a regular switch is used, the switch must first be moved back to its default position before InstaTrim can again be triggered.

NOTE: It is advisable to disable this function once initial trimming is completed as it can have seriously negative results if triggered accidentally!

Hi.Res Slow/Delay *(High Resolution Slow/Delay)*

When selected this option increases the resolution of time settings for both “Delay” and “Slow” functions in the Mixer. If OFF, Delay and Slow functions can be set between 0 and 15 seconds in steps of 1 second. If Hi.Res is ON, then they can be set between 0.0 and 3.0 seconds in steps of 0.2 seconds. Press [MENU] to toggle this option ON or OFF

Trainer

When ON this enables a model to receive trainer inputs from a connected Slave (student) transmitter, providing that all the configurations were correctly made in the Trainer Setup menu. The option is for use when the transmitter is serving as Master in a buddy box arrangement. See page 22 for an explanation of the Trainer function.

Unless specifically required for training purposes, this option should be left turned OFF.

Beep Cnt *(Beep Centre)*

Causes inputs to beep when centred. RETA123 corresponds to: Rud, Ele, Thr, Ail, P1, P2, P3 inputs. This is useful for locating the centre of the pots without looking (e.g., while flying!). Press [MENU] to turn ON centre beeping for an input. It will then become highlighted.

Protocol

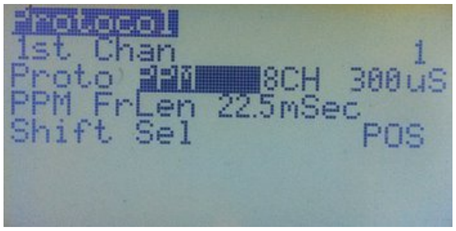
The contents of the Protocol menu vary according to the “Proto” setting. Choices are PPM, PXX, DSM2, PPM16 and PPMSIM (the latter only when active in the *Radio Setup, Controls* menu – see page 19).

1st Chan

This is the first channel in the signal sequence. It is usually left at 1, but other start numbers may be needed for a flight controller which accepts raw PPM. Note that this setting has no effect when used with the DSM2 protocol.

PPM

This is the protocol used for many plug-in RF modules (including the OrangeRX DSM2/DSMX, FlySky, FrSky DJT and Spektrum DM9). Choices are:



```
Protocol
1st Chan               1
Proto PPM    8CH   300uS
PPM FrLen 22.5mSec
Shift Sel            POS
```

Number of Channels: Can be set from 4 to 16 in increments of 2. This is important when using the TX with some flight simulator dongles that can't accept more than 4 or 6 channels. Also, with receivers having 6 or fewer channels, such as the FrSky D4R-II, setting the PPM number of channels to match the receiver allows for a faster PPM signal with less latency.

Pulse Spacing: The default spacing of 300uSec is suitable for most modules, but may be adjusted to eliminate jitter.(The Spektrum DM9 requires 400uSec for accurate centring.)

PPM FrLen (Frame Length): The length of each frame within the PPM train, in milliseconds. For a setup with 8 or fewer channels it it can normally be left at the default 22.5 ms. You can use the following formula to calculate the minimum frame length:

Number of Ch * Max Pulse Length (2140uS with E-limits enabled) + Sync Pulse (5ms)

Example: 10 Ch * 2.14ms + 5ms (Sync) = 26.4ms (set to 26.5ms)

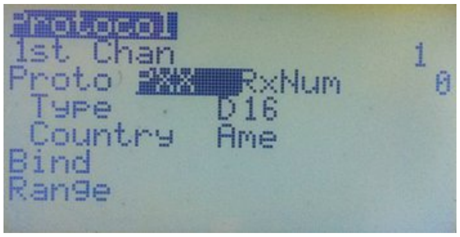
Below are recommended PPM frame lengths for various numbers of channels:

No of Channels	Frame Length (ms)	No of Channels	Frame Length (ms)
4	14	12	31
6	18	14	35
8 (Default)	22.5	16	39.5
10	26.5		

Shift Sel: Selects POSitive or NEGative pulses. Some modules and simulators adjust automatically (e.g., Frsky DJT/DHT and OrangeRX DSM2/DSMX), while others have specific requirements (Spektrum DM9 module requires NEG).

PXX

This setting should be selected for the FrSky XJT module. Options for this protocol are:



```
Protocol
1st Chan              1
Proto PXX   RxNum     0
 Type       D16
 Country    Ame
Bind
Range
```

RxNum - Receiver match. This gives the receiver a number (0...124) to prevent you from flying with the wrong model selected in the radio.

Bind - Puts the XJT module in Bind Mode.

Range - Puts the XJT in Range Check Mode (much reduced power).

Type - Selects the type of receiver. Choose D16 for the X series, D8 for the D series, and LRP for the long range receivers.

Country - *S*ets the RF module power to match legal requirements in some countries.

Failsafe - Sets receiver failsafe from the transmitter (applies only to X-series receivers).

For more information, please go to the FrSky website and download the manuals.

DSM2

This setting is used where an internal module extracted from a Spektrum transmitter is "hacked" into the transmitter. Instructions for doing this can be found here. Options for this protocol are:

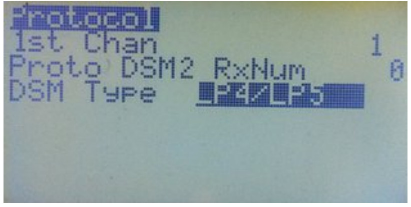
RxNum – Allows use of the Spektrum ModelMatch feature to prevent you from flying with the wrong model selected in the radio. Gives the receiver a number (0...124).

DSM Type – Choose according to the type of “hack” module:

LP4/LP5 short range module from an RTF ultra micro model or helicopter;

DSM2only for a DSM2 module from a DSM2 transmitter such as the HP6DSM or early DX5e or DX6i;

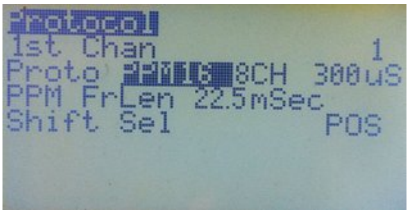
DSM2/DSMX for a DSMX module from a DX4e, DX5e or DX6i.



```
Protocol
1st Chan              1
Proto DSM2  RxNum     0
DSM Type    LP4/LP5
```

PPM16

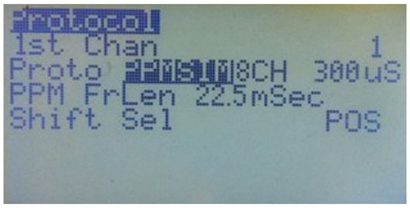
This setting sends channels 9 to 16 to the trainer port. If a second RF module is connected to the trainer port, the transmitter can output up to 16 channels. Cool!!



```
Protocol
1st Chan              1
Proto PPM16 8CH  300uS
PPM FrLen 22.5mSec
Shift Sel           POS
```

PPMSIM

For use only when a 9x transmitter is to be used with a simulator or as the buddy box in a trainer setup AND the transmitter has a FlySky/ Turnigy module installed AND the PPM line to the module has not been modified (see page 19).



```
Protocol
1st Chan              1
Proto PPMSIM 8CH 300uS
PPM FrLen 22.5mSec
Shift Sel           POS
```

Annexes

Annex A: Programming Examples

Unless you have a friend to help you, the only way to learn programming is just to do it, using the manual as a guide. But examples and explanations can help a lot. So here are examples ranging from the very simplest programming of a stick output, switch or pot, through basic mixes that cause one channel to affect another, to some quite subtle combinations of mixes and logical switches that can manage actions such as coordinating gear retract with the opening and closing of doors.

Several of the following examples are based on the work of Pat MacKenzie, “MikeB” and “Gyro Gearloose”. Thanks to Reinhard (“ReSt”) for collecting them and sharing with us. We have tried to give credit where appropriate. But blame us, not them, if we got things wrong in our explanations.

Note that some examples make use of features only available in recent versions of Er9x. One of these is the sticky throttle-cut setting in “Safety Switches” (see page 52). Another is the “Fix Offset” option that makes the results of offsets much more predictable (see page 32). It's a good idea to upgrade to the latest version of Er9x, though most things can be done with older versions.

ATTENTION: Use the examples below at your own risk. These examples are not recipes to be copied blindly. You will need to adapt them to the specific needs of your model. Don't expect them to work right away in your application.

It is imperative that you remove props and rotors from your models (or unplug motors) while setting up and testing programming, especially as many of the examples involve throttle. But then, if you fly electric models you should know that anyway – it's one of the basic rules. You have been warned. 🙂

List of Examples

1. Simple Mixes
2. Using the Basic Mix Options (Multiplex)
3. Simple Throttle Cut
4. Elevon and V-tail Mix
5. Taileron Mix
6. Down Elevator with Increasing Throttle
7. Automatic Glow Driver
8. Automatic Throttle Control based on Altitude
9. Variable Steering Depending on Throttle Position
10. Shift a Curve Up or Down with a Pot
11. Flight Battery Voltage Warning
12. Adjusting Throttle Idle with a Pot
13. Inverse Proportional Channels
14. One-Shot Timer Switch
15. Differential Thrust for a Twin
16. Automatic Landing Gear and Canopy
17. Sequenced Landing Gear and Doors (simple)
18. Sequenced Landing Gear (doors closed with gear extended
19. Flaps Working as Ailerons
20. Setting up Er9x to Emulate a Spektrum Transmitter
21. Using Channels PPM1 to PPM8 in a Trainer Setup

1. Simple Mixes

Full Throw Mix

This first mix uses FULL as source. It will be at -100% when the ELE switch is OFF and 100% when the ELE switch is turned ON.

```
    CH5  100%  FULL   switch (ELE)
```

Note that you could achieve the same result by using the switch as source (leave Switch as "---"):

```
    CH5  100% sELE
```

Half Throw Mix

This mix uses HALF as source. The channel will be at -100% when the switch is OFF and at 0% when the switch is ON. This reverse operation is due to "!" before the switch name, which means we are using the inverse of the switch.

```
    CH6  -100%  HALF  switch (!THR)
```

Mix with Pot as Source

This simple mix on channel 8 uses a potentiometer (P3) as source. Because the weight is 80%, when moving the pot through its full range the mix will output values between -80% and 80%.

```
CH8  80%  P3
```

Simple Throttle Cut Mix

This mix provides a simple throttle-cut on channel 3. The first line produces output from -100% to 100% in response to the throttle stick. The second replaces the first one when the THR switch is turned on, thus locking the throttle channel at -100% (low). Note the "R" multiplex setting.

```
    CH3  100%  Thr
      R  -100%   HALF  switch (THR)
```

Three-Position Mix

This mix uses the three-position switch as source. When the switch is on ID0 (up), the channel will be at -100%, when on ID1 (middle) it will be at 0%, and when the ID2 position (down) is selected the channel will be at 100%. Because the Slow setting is used, the output will change up over 3 seconds and down over 2 seconds when the switch setting is changed.

```
    CH7  100%  sIDx  slow (u3 : d2)
```

Here is a variation on the same mix. This one adds offset to set the value for ID1 (middle), and weight and differential to set the values for both endpoints (ID0 and ID2). When the three-position switch is on ID0 (up), the channel will be at -80%, when on ID1(middle), the channel will be at +20%, and when on ID2 it will be at +90%.

As you can see, the weight, offset and differential can be used to set the value for the three positions of the switch.

```
    CH10  100%  sIDx  offset (20%)  Diff (-30)  slow (u3 : d2)
```

2. Using the Basic Mix Options (Multiplex)

The Multiplex option in the Edit Mix window provides a choice of three types of mix: add, multiply and replace. See page 36 for a full explanation.

An **ADD** mix is added to the result of all mixer lines above it.

A **MULTIPLY** mix scales the result of all mixer lines above it.

A **REPLACE** mix, as the name says, replaces all mixer lines above it. If several replace mixes are active, the last one replaces all previous mixes.

Most of the time you use an **ADD** mix, as shown in this example. Here CH 1 is the Aileron channel and CH4 is Rudder:

CH1 100% Ail (Gives aileron control in response to Aileron stick)

CH4 100% Rud (Gives rudder control in response to Rudder stick)

+ 10% Ail Switch (RUD) (If the switch is ON, 10% of Aileron stick control is added to the Rudder channel)

Using this combination of mixes, Aileron and Rudder are controlled independently by their respective sticks over the full range from -100% to 100%. In addition, when the Aileron stick is moved, the rudder also moves in the same direction, but by a much smaller amount. In this example, 10% of aileron control will be added to the Rudder channel. The Rudder stick has no effect on the ailerons, as these mixes are programmed in the Rudder channel and control the rudder servo only.

This could be used to help coordinate turns for a model that needs rudder input to avoid skidding. Very useful for lazy people that don't want to learn how to use the rudder. 🙂

If we change the second line for the Rudder channel to a **REPLACE** mix, we get:

CH1 100% Ail (Gives aileron control in response to Aileron stick)

CH4 100% Rud (Gives rudder control in response to Rudder stick)

R 100% Ail Switch (RUD) (If the switch is ON, Aileron stick alone controls rudder)

Again the first mix on CH4 controls the rudder servo with the Rudder stick, but only when the RUD switch is OFF. With the RUD switch ON, the Rudder stick does nothing and the Aileron stick has full control of the rudder servo.

If we now use a **MULTIPLY** mix, we can set up something like the following on the Rudder channel:

CH1 100% Ail (Gives aileron control in response to Aileron stick)

CH4 100% Ail Switch (RUD) (Gives rudder control in response to Aileron stick)

* 5% P1 Offset 5% (“Fix Offset” option ON) (Adjusts weight of previous mix, depending on P1)

+ 100% Rud (Gives full rudder control in response to Rudder stick)

This combination of mixes again allows Aileron and Rudder to be controlled by their respective sticks over the full range from -100% to 100%. In addition, when the Aileron stick is moved, the rudder also moves in the same direction, but by a much smaller and adjustable amount. In this example, between 0 and 10% of aileron control, depending on the setting of P1, will be added to the Rudder channel. The Rudder stick has no effect on the ailerons, as the relevant mixes are programmed in the Rudder channel and control the rudder servo only.

Such a variable mix can be used to fine-tune the amount of rudder added to avoid skidding in turns.

3. Simple Throttle Cut

We mentioned this mix earlier, and the same result can be achieved using a Safety Switch. But creating it yourself via the mixer is a very useful exercise, so here are step-by-step instructions.

1. Go to the MIXER menu.

2. Start off with the default four mixes (Rud, Ele, Thr, Ail) in whatever channel order you prefer.

3. Scroll down to your throttle channel. Let's say CH3.

4. Press [MENU] to display the pop-up window with more options, and then select “INSERT”. A new mix will be inserted on CH3 and the “Edit Mix” menu will come up.

5. Within Edit Mix, change source to “HALF” and Weight to “-100”.

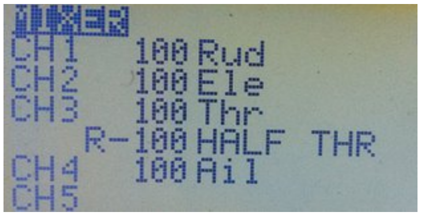
6. Scroll down and set Switch to “THR”.

7. Scroll down further until you reach Multpx. Change the value to “Replace”.

8. Press [EXIT]. You should now see something like in the picture below:

What it tells you is that on CH3 you have two mixes defined.

The first takes the value of the throttle stick. The second depends on the THR switch. When the switch is OFF the mix is ignored, so the only value will be the value of the throttle stick. When the switch is ON that value is replaced by -100%, and the channel is locked at low throttle. The throttle stick no longer controls the channel.



```
MIXER
CH1   100 Rud
CH2   100 Ele
CH3   100 Thr
    R-100 HALF THR
CH4   100 Ail
CH5
```

4. Elevon and V-tail Mix

This type of mix is used where two control surfaces must be able to move either together or in opposite directions. In the case of an elevon setup, also called a “delta wing mix”, the two inputs are elevator and aileron. When elevator input is applied, the two surfaces move together to pitch the nose of the model up or down; when aileron is applied they move in opposite directions to produce roll.
V tail controls are similar, except that the inputs are elevator and rudder.

To be capable of elevon or V-tail control, each control surface must have its own servo and the servos must be on separate channels (not a Y-cable). Each channel requires two mixes, one for the elevator stick input and the other for aileron stick input. Where the elevon servos are plugged into channel 2 (right elevon) and channel 3 (left elevon), the basic arrangement looks like this:

CH2 50 Ele
+ -50 Ail (Note negative weight on one Ail mix)
CH3 50 Ele
+ 50 Ail

It's easy to see that if the stick is moved to give up elevator, both servos receive the same input, while if aileron is applied, the negative weight on one channel means that they receive opposing inputs. (Note that it might be necessary to reverse the directions, depending on the mechanical setup of servo arms and pushrods.)

The reason for not using 100% weighting is that if, for example, full up elevator and full left aileron were applied simultaneously, the two inputs would add up to 200%. This is not good! It means that somewhere about half stick movement, the limits settings would begin to clip the channel outputs and limit servo movement. The solution is to ensure that the combined weight value of the two mixes in a channel does not exceed 100%.

Note that the weighting of the two inputs may need to be different, depending on the model (e.g., tailless models often require much less elevator than aileron). Hence, an elevon program might have 70% weighting for the aileron mixes and 30% for the elevator mixes (70% + 30% = 100%).

Nigel

5. Taileron Mix

This program adds to aileron authority by allowing separate elevator halves to “twist” like ailerons while still providing normal elevator control when moved together. “Tailerons” are a variation on the elevon/V-tail mix explained above.

The model must have individual servos on separate channels for the two elevator halves. We will assume that CH2 is Left Elevator and CH7 is Right Elevator. Ailerons can be on a single radio channel, CH1 in this case.

We will use the three-position switch to turn the “taileron” function on and off:

ID0: Normal flight characteristics (no aileron action on elevators)

ID1: Normal flight characteristics plus additional function (e.g., flaps)

ID2: Tailerons engaged (elevators give aileron action for enhanced manoeuvrability)

The mixes required to achieve this action are as follows:

```
CH1  -100%  Ail    (Normal Aileron control)
CH2  +100%  Ele  (Normal Elevator control)
        -100%  Ail  Switch (ID2) (Adds Aileron input when switch is in ID2 position)
CH7  +100%  Ele  (Normal Elevator control)
       +100%  Ail  Switch (ID2)  (Adds Aileron input when switch is in ID2 position)
```

Note that positive and negative weightings depend on the mechanical setup of the servos and elevators. As explained above, it may be necessary to reduce the weights well below 100% to avoid the channel output being clipped by the limits settings and introducing deadband at the end of the stick travel.

Thanks to Gyro Gearloose

6. Down Elevator with Increasing Throttle

This program is used to compensate the tendency of some models to climb at higher power settings by adding a small amount of down elevator as throttle is increased.

For this example, let's assume that down elevator uses the positive side of the weight value. Then for a weight of 100%, down elevator will range from 0 to 100, and up elevator from 0 to -100.

So to add down elevator proportionally to the throttle position you need a mix that works on the positive side of the range and that has Thr as source:

CH2 +100% Ele

+5% Thr offset (+5%) switch (ELE)

The offset makes sure the Thr mix adds only positive values between 0 and 10% to the elevator channel value, depending on the throttle stick position.

A weight of 5% produces a movement from -5% to 5% with the full movement of the throttle stick.

An offset of 5% will move the centre point by 5%. Thus 0%+5% = 5% (new centre).

The result when the throttle is low will be -weight + offset (-5+5 = 0). When throttle is at full, the mix will give +weight + offset (5+5 = 10).

So, the second mix will "add" 0% to the first mix when throttle is low and 10% when the throttle is at full, increasing proportionally through the full throttle range.

The ELE switch enables/disables the mix.

Version Active Only When Above Mid-throttle

Here the objective is to add a small amount of down elevator proportionally to the throttle stick but only in the top half of throttle stick movement. The mix is similar to the previous one but uses a curve to set the range over which it operates:

CH2 +100% Ele

10% Thr switch (ELE) Curve (x>0)

The curve (x>0) means that the mix will only respond to positive values from the throttle stick. For the negative values (below the mid point) the output of the mix will be 0 and it will not affect the elevator channel. From the mid point upwards the value of the mix will increase to 10% at full throttle.

Examples will show what's happening:

If throttle is below the mid point there will be no effect on the elevator value.

If elevator value is +10 (some down) and you move the throttle stick up to say 3/4 of full throttle, the mix value will be 5%, so 10 + 5 = 15% down.

If elevator is at centre (0%) and you go full throttle, the throttle mix value (10%) will be added to the elevator channel, giving 10% down.

Advice – Create this mix on your transmitter or in eePe and experiment with different values. If you do it on the radio with a live model, please **remove the prop or unplug the motor!**

7. Automatic Glow Driver

Suppose you have a glow plug driver (a NiCd cell and electronic switch) which you want to turn on whenever the throttle stick position is below 10% in order to improve the idle of the engine. For this you need a mix to drive the switch and a logical switch to control the mix. Assume the glow driver switch is on CH8 and requires a positive value to turn it on.

Throttle output varies from -100 to 100, a range of 200, so 10% throttle occurs at -80.

1. Go to the Logical Switch menu and highlight "L1"
2. Select the condition as "v<val".
3. Select the source as "Thr".

4. Select the “value” as -80

The resulting logical switch looks like:

L1 v<val -80

Now you need a mix to use that switch. In the mixer menu select CH8 for glow driver operation. Select FULL as the source and Switch as L1.

CH8 100% FULL switch (L1)

Now whenever the throttle stick position goes below 10% (i.e., Thr value goes below -80), CH8 will go from -100% to 100%, turning the glow driver ON.

8. Automatic Throttle Control based on Altitude

The aim here is to control the throttle automatically so you don't have to be constantly adjusting it. Just leave the throttle stick at a comfortable climb setting and forget about it. When the model reaches the preset maximum altitude, the motor will stop. Leave the throttle at the climb setting, and the motor will restart when the model descends to the preset minimum altitude. (This example assumes that you have done the telemetry hardware mod to the transmitter and have the telemetry version of Er9x installed.)

Let's use a pot instead of altitude for now, so you can test it first on the radio or in eePe. Then we will replace the pot with the telemetry source “Alt” (Altitude).

Program two logical switches to detect two positions of P1, Let's say, 80% and -80%

L1 v>val P1 80 (switch will turn ON when P1 is above 80%)

L2 v<val P1 -80 (switch will turn ON when P1 is below -80%)

Now program a logical switch as Latch using both logical switches above:

L3 Latch L1 L2

L1 sets the switch (turns it ON) and L2 will reset it (turn it OFF).

Now turn the pot completely anti-clockwise and start moving it clockwise. When P1 reaches 80%, L3 will turn ON due to L1 becoming true. Now start moving the pot counter clockwise. When under 80% L1 will go OFF but L3 will stay ON. Continue to move the pot and when it reaches -80% L2 will turn ON and consequently reset the Latch, turning L3 OFF.

Now, let's create a couple mixes controlled by the latch function of L3:

CH3 100% Thr

R -100% FULL switch (L3)

When the pot (P1) is above 80%, L1 will set the latch function, turning L3 ON and activating the replace mix. This pulls the channel to -100% (i.e., throttle OFF), even with the Thr stick fully up. Now turn P1 fully counter clockwise; when it reaches -80%, L2 will turn ON and will reset the latch function, turning L3 OFF. The replace mix controlled by L3 will be disabled, giving control back to the throttle stick.

Now replace P1 with altitude settings in L1 and L2:

L1 v>val 80m (meters)

L2 v<val 10m (meters)

Yes, you've guessed it, You have an automatic throttle control based on altitude. When the model reaches 80 meters, the motor will stop automatically, and the model start gliding down; when it flies below 10 meters, the motor will start automatically and the cycle repeats. Cool? 😉

Enhanced Version

This is the same Auto-Throttle control example as above but with the addition of an enable/disable switch and also smooth transition of the motor when starting/stopping. To achieve this we will use a flight mode (FM1).

Program the two logical switches so one will detect the altitude at which the motor should automatically turn OFF, and the other the altitude at which the motor should be able to start again. For this you should keep the throttle stick at the required setting for climbing.

L1 v>val 80m (meters)

L2 v<val 10m (meters)

Now program the logical switch using the latch function:

L3 Latch L1 L2 GEA (The GEA switch will enable the auto-throttle when turned ON)

Next we will create the mixes. The difference here is that we won't use L3 to control the Replace mix as in the example above. We will use it to control flight mode 1 (FM1) instead.

CH3 100% Thr (activate this mix on flight mode 0 only.)

R -100% FULL (activate this mix on flight mode 1 only.)

To select the modes in which the mixes are active, go to the “Edit Mix” menu, scroll down to “MODES”. The mix will be active on the highlighted modes.

Next go to the Flight Modes menu, select FM1 and edit it as shown:

Switch: L3

Trims: RETA

This gives the flight mode its own trim settings, so you can adjust the trims for climb mode and they will be independent from the trims that are active when the motor is OFF.

To allow for a smooth transition when the motor stops, and when control returns to the throttle stick:

Fade In: 3 seconds

Fade Out: 3 seconds

How to use this program:

With the throttle stick at the bottom (motor stopped) turn the GEA switch ON to activate the “auto-throttle control”. Then advance the throttle, launch the model and climb. The motor should stop automatically when the model reaches 80 meters. Now glide down with the throttle stick still at the climbing setting. When the model reaches 10 meters, the motor should ramp up to climb power.

To deactivate this, just bring the Thr stick back to minimum and turn the GEA switch OFF. You will now have normal throttle control.

João

9. Variable Steering Depending on Throttle Position

The purpose of this program is to reduce the sensitivity of nosewheel steering as the throttle setting increases (just as some cars reduce power steering boost with speed).

Suppose we want a steering rate of 100% when the throttle is closed, reducing to zero rate (no nosewheel steering) at full throttle. When the throttle is opened almost all the way, the steering will be very gentle.

What is required is a Multiply mix added to the nosewheel servo channel (CH6):

CH6 100% Rud

* -50% Thr Offset(-50%) (Note: Fix offset option is on)

Thanks to MikeB

10. Shift a Curve Up or Down with a Pot

This mix enables adjustment of the middle range of the throttle without affecting the ends.

Set up a curve (e.g. c2): 0, 100, 100, 100, 0.

On an unused channel, e.g. CH16, we set up the following mixes:

CH16: +100% Thr Curve(c2)

* 50% P1 Offset(50%) (Note the multiply multiplex setting)

Now on the main throttle channel (CH3) we add:

CH3: +10% CH16

This will add 0 to 10% to your throttle output controlled by P1, in the middle range, but the addition will be steadily reduced to zero as the stick is moved towards either end of the throttle range.

Thanks to MikeB

11. Flight Battery Voltage Warning

The purpose of this program is to have the radio detect and issue a warning when the voltage of the LiPo flight pack battery drops below a certain level for at least 3 seconds. The reason for this delay in reporting is that under heavy load, the battery experiences a significant voltage drop; if the instantaneous voltage is used, you may get a “Low Battery” warning every time you go to full throttle, well before you need to think about landing.

The program assumes you have Er9x-frsky installed and are using FrSky telemetry with the battery pack sensor connected to A2. It requires the following logical switches:

L1 v<val A2 10.5 (detects when battery voltage drops below 10.5v)

L2 TimeOff 1 ON 3 AND switch L1 (turns ON for 3 seconds, OFF for 1 second if L1 is ON)

L3 AND !L2 L1 (trigger for “Low Battery” alarm)

Here's how it works: While L1 is OFF (voltage on A2 >= 10.5), all other logical switches are OFF. When the voltage drops below the preset level (10.5V), L1 turns ON, allowing L2 to start counting. L2 turns ON for 3 seconds then OFF for 1 second for as long as L1 is ON.

L3 looks for the inverse of L2 and will therefore turn ON 3 seconds after L1 turned ON, as long as L1 is still ON. It will stay ON for 1 second then turn OFF for another 3 seconds.

Use L3 to trigger your "Low Battery" alarm.

Note: The optimum setting for the trigger voltage will vary depending on the C-rating and age of the battery, as well as the full throttle current. The suggested value of 10.5v is typical for a 3s battery and represents 3.5v per cell.

Testing the Program

Here's a way to test the program without having telemetry set up and working. We can use use a scaler to simulate a voltage, using one of the pots (P1) as source.

Set Scaler 1 (SC1) as follows (it's located near the bottom of the Global Variables menu):

Source: P1 (digital values range from -1024 to 1024)	Unit: volts
Offset: 93	Sign: +
Multiplier: 1	Decimals: 1
Divisor: 11	Offset At: Last

This setup will give values between 0 and 18.6 when moving P1 from one end to the other.

Here's how it works: Imagine P1 is at a position that gives a value of 512 (about 3/4 of rotation clockwise). The multiplier is 1 so the value stays at 512. Then the divisor is applied: 512/11 = 46.5 (round this to 46). Now the offset is applied (because it was set to “Last”): 46 + 93 = 139. The decimal setting of 1 is next applied to give 13.9. As the units are set to volts, this will become 13.9V.

The number at the top of the edit screen (next to SC1) will show the output and vary with P1.

To use this test setup, change the first logical switch in the example above to use the scaler output:

L1 v<val SC1 10.5 (A2 replaced by SC1)

Thanks to MikeB

12. Adjusting Throttle Idle with a Pot

This example works exactly like the “Throttle trim” feature in Er9x, except that it uses the pot P1 instead of a trim switch. It adjusts the idle setting without affecting the full throttle setting.

Assuming CH3 is the throttle channel, the required lines are as follows:

CH03 -50% Thr Offset (50%) (defines half throttle range)

* +40% P1 Offset (40%) (multiply mix introduces P1)

+100% Thr (adds on the basic throttle stick mix)

Thanks to MikeB

13. Inverse Proportional Channels

This is an example of how the value of one channel can have an inverse proportional impact on another. In this case, when throttle is at low the aileron movement will range between -100% and 100%; when at full throttle the aileron movement will range from -20% to 20%. Here aileron uses CH1.

CH01: +100% Ail (Basic aileron stick mix)

* -40% Thr Offset(60%) (Fix Offset ON) (Multiply mix reduces response as Thr increases)

Thanks to MikeB

14. One-Shot Timer Switch

This example illustrates the use of a Latch logical switch (see page 48). Briefly pressing the momentary TRN switch causes the affected channel (here CH16) to go from its normal -100% value to +100%, wait 2 seconds, and return to -100%. The logical switches required are:

L5 Latch TRN L6 (Latch triggered by TRN switch, reset by L6)

L6 v>val CH16 99 (Detects +100 value to reset L5)

The switches control the following mix:

CH16 100% FULL Switch (L5) Delay (u2:d0) (L5 triggers FULL change for 2 seconds)

How it works: When you press the TRN switch you trigger the Latch logical switch (L5) and the mix goes to +100%. Logical switch L6 detects when the same mix is above 99% and resets L5, turning it off. The mix will then return to -100% after a 2 second delay as a result of: (u2:d0).

Thanks to MikeB

15. Differential Thrust for a Twin

The purpose of this program is to enable the rudder channel on a twin-motor model to adjust the power to the motors so that differential thrust assists turning. This is particularly helpful for a seaplane while taxiing (but should be used very cautiously in the air!). This example could be easily adapted to equalize thrust where there are minor differences between the two motors, ESCs or props.

The model must use separate channels to control the ESCs (electronic speed controls) for the two motors (a Y-cable cannot be used). Remember to set failsafe to low throttle on both channels. Safety above all!

The three-position switch is used to turn differential thrust on and off:

ID0: Normal flight characteristics (throttles work in unison, no thrust differential)

ID1: Normal flight characteristics (e.g., reserved for control of flaps, no thrust differential)

ID2: Differential Thrust engaged

In this example we use CH3 for the left throttle and CH8 for the right throttle. Rudder is on CH1.

The required mixes are as follows:

CH1 100% Rud (basic rudder stick mix)

CH3 100% Thr (basic throttle stick mix for left motor)

+ 50% Rud Switch (ID2) (adds rudder input to throttle when ID2 is ON)

CH8 100% Thr (basic throttle stick mix for right motor)

+ -50% Rud Switch (ID2) (adds inverted rudder input to throttle when ID2 is ON)

For safety, a single kill switch is required that locks both motors at low throttle. The following safety switches will provide sticky throttle kill on channels 3 and 8:

CH3 X !THR -100

CH8 X !THR -100

Thanks to Gyro Gearloose

16. Automatic Landing Gear and Canopy

Gyro Gearloose says:

> Retractable landing gear is great. It's sexy, and it actually makes your plane faster. But a good portion of people can't reach or remember to flip the right switch at the right time. I thought it would be even sexier if retractable gear (and in this case, an electric canopy) was fully automatic, based on the throttle input. Here, we assume that if the motor is running at any appreciable speed, we're taxiing for takeoff and should close the canopy. We also assume that if the motor is running at a good speed for more than 10 seconds, we've either gotten off the ground or crashed into something, so we should lift the landing gear in flight. Cut the throttle, and we're coming in to land again. Sit on the ground with the throttle off for a while and we can impress people by automatically letting the 'pilot' out of the plane. And of course, since Murphy lives at airstrips, we want an override to force gear down and canopy closed RIGHT NOW when something goes horribly wrong.

In this example, CH3 is throttle, CH5 is gear and CH8 is canopy. Negative values represent lowered gear and open canopy.

Logical Switch 1 (L1) tracks whether the throttle is open at all (above -95) by monitoring the throttle stick.

```
L1  v<val  THR  -95
```

In the mixer we detect the real output of the throttle channel by using a high channel (such as CH10) with CH3 as source. The option “Use Output” is turned ON to ensure that any safety switch action will be included (see page 34).

```
CH10  100%  OP3 (“Use Output” option ON)
```

We now use CH10 to trigger Logical Switch 2 (L2). This tracks whether the throttle is at flight power (above -50) by monitoring the real output of the throttle channel (as opposed to just the throttle stick). It will thus detect if the “Sticky Throttle-Kill” is ON:

```
L2  v<val  CH10  -50
```

We now have what we need to control the gear (CH5) using L2, so we set up the following:

```
CH5  -100%  FULL  Switch (L2)  Delay (u10:d0)  Slow (u2:d2)
```

The gear is delayed 10 seconds after flyable throttle is reached before retracting, but to allow a quick landing, it is lowered immediately when the throttle drops below the flyable level. A Slow setting of 2 seconds is used for raising and lowering the gear.

The canopy (CH8) uses a similar mix but controlled by L1 and with different timing:

```
CH8  -100%  FULL  Switch (L1)  Delay (u0:d15)  Slow (u2:d2)
```

The canopy closes any time the throttle is open at all (above -95), and waits 15 seconds before it will open again if the throttle is closed. So, don't dead-stick for longer than this during a landing, unless you want your pilot windburned. 🙂 It is set to take 2 seconds to move between open and closed.

For safety, the canopy obeys the throttle stick position, while the gear obeys the actual channel output – so having TK engaged during setup and bumping the stick will NOT fold the gear. If TK is not engaged, then we expect you intended to fly and it's OK for the gear to go up 10 seconds after the throttle is opened to a flyable level.

The programming includes a Sticky Throttle Kill (TK) feature:

CH3 X !THR -100

Finally the GEA switch acts as our bailout safety – when it's OFF the system is in “automatic mode”, while when it's ON we get “failsafe” mode with gear-down and canopy-closed.

CH5 S GEA -100

CH8 S GEA 100

Thanks to Gyro Gearloose

17. Sequenced Landing Gear and Doors (simple)

Gyro Gearloose says:

> After the complexity of the automatic gear above, this is much easier. Here, we assume you have retractable landing gear and gear-doors with separate servos, and you want them to open in a logical fashion instead of ramming the gear against the door while it's still opening. Or perhaps you have bomb doors that need to open before you release the bomb. Either way, this is a simple demonstration of the delay functionality applied in an overlapping fashion to two channels... gear retracts before doors close, then doors open before gear deploys.

In this example, CH5 controls gear, while CH7 controls the doors that need to open first/close last. GEA switch actives the sequences.

CH5 100% FULL Switch (GEA) Delay (u3:d0) Slow (u2:d2)

CH7 100% FULL Switch (GEA) Delay (u0:d3) Slow (u2:d2)

Starting with gear down and doors open, switching GEA to ON triggers the landing gear to retract over a period of 2 seconds. The doors start to close 1 second later and take a further 2 seconds to complete their movement. When GEA is switched OFF, the doors open and the gear extends.

Thanks to Gyro Gearloose

18. Sequenced Landing Gear (doors closed with gear extended)

Version 1

This program does basically the same as the previous simple example, but closes the gear bay doors not only when the gear is up (as usual) but also when it's down. Not many aircraft close the doors with the gear down, but I think it's cool. 🙂

The example uses five logical switches to manage the action. Flick the trainer switch once, the doors open, the gear moves up, the doors close. Flick the trainer switch again and the reverse happens. As before, the retractable gear is controlled by CH5 and the gear bay doors by CH7. Negative values are used for gear down and for doors closed.

The following mixes control gear retraction and door opening with appropriate delays and speed of operation.

CH5 100% FULL Switch (L1) Delay (u5:d5) Slow (u5:d5) (Retract control)

CH7 100% FULL Switch (L5) Delay (u8:d0) Slow (u3:d3) (Door opening control)

The following logical switches serve to make sure it's safe to initiate gear retract:

```
L2  v<val  CH7  -98
L7  AND  L2  TRN
```

These two logical switches working together will detect when CH7 is low (doors closed) and the TRN switch is pressed. If CH7 is not low, it means the doors aren't completely closed and the sequence hasn't finished, so even if you press the TRN switch, nothing will happen. L7 is then used to start a new sequence. With this you can only start another sequence when the previous has finished. This is to avoid things getting out of sync and destroying the mechanics, servos, etc.

Now we need a way to control gear retract:

```
L1  F-Flop  L7  !L1
```

This switch is a F-Flop (Flip-Flop) that will turn ON when L7 is on, and OFF when L7 is ON again. This controls the mix on CH5, the retractable gear channel.

```
L6  v>val  CH7  99 (Turns on when doors are open)
L5  Latch  L7  L6
```

L6 and L5 form a “One-Shot Timer” switch. That means when you trigger it, it will go to 100%, and return to -100% automatically after a delay value you set. See page 81.

L6 detects when CH7 is at +100%.

L5 is a Latch switch and controls CH7 (gear bay doors).

Here's how it works. L7 will set the latch switch on when you press the TRN switch. The doors will start opening as CH7 moves towards +100%. When it gets there, L6 condition will be verified and will turn the latch switch (L5) OFF. Then the return sequence starts by waiting until the retract stop moving (Delay), and then moving slowly (Slow) back to -100%, closing the doors again. This finishes the sequence.

It's a good idea to have a separate “Panic” switch in case you need to open the doors and get the gear down quickly. Using the AIL switch for this purpose, the following safety switches will do the job:

```
CH5  S  AIL  -100%
CH7  S  AIL   100%
```

João

Version 2

Here is another way to achieve the same sequence of actions as the previous example. Starting with the gear down and doors closed, when “gear up” is commanded the doors open, the gear retracts and the doors close again. And the reverse happens when the time comes to land. Three channels and two logical switches are used.

Channel 16 serves as the 'master request' specifying which position (UP or DOWN) the gear should be in. Channel 13 controls the gear retraction and extension, while Channel 14 controls the opening and closing of the doors.

Any time the values of Channels 13 (gear position) and 16 (requested gear position) differ, Logical Switch A (LA) causes Channel 14 to open the doors to allow the gear position to be changed. Logical

Switch B (LB) detects whether the doors are open to allow the gear to retract or extend. Unless the door is open channel 13 remains wherever it is. When Channels 13 and 16 match, Channel 14 closes the doors.

Logical switches:

```
LA  v1!=v2  CH13  CH16
LB  v>val  CH14  98
```

Mixes:

```
CH13  +100%  CH16  Switch (LB)  slow (u4:d4)
    R   +100%  CH13  Switch (!LB)
CH14  +100%  FULL  Switch (LA)  Slow (u2:d2)
CH16  +100%  FULL  Switch (GEA)
```

Thanks to MikeB

19. Flaps Working as Ailerons

This programming enables flaps to be used in three different ways. First they can act in unison with ailerons with no flap action; in effect this creates full length ailerons that produce less drag than smaller ailerons alone moving to steeper angles. Second, the flaps can be set to operate separately from the ailerons in the conventional fashion. Third, the flaps can both move together in response to a pot or differentially in response to aileron input.

To function in these different ways, the two flaps must be controlled independently. This requires that each have a separate servo and be on a separate channel; a Y-cable cannot be used. In this application aileron servos can be either on separate channels or a Y-cable.

In the following example, CH1 is used for aileron control, CH6 for left flap, and CH7 for right flap.

The three modes of operation are controlled by the three-position switch:

ID0 – Flaps work in unison with ailerons to act as full-length ailerons; no flap action

ID1 – Flaps function as flaps, no aileron action; P3 sets flap position

ID2 – Flaps provide combination of flap and aileron action; P3 sets flap position

Here is one way to do the necessary mixes, with flaps on channels 6 and 7. Note that the aileron mixes use the multiplex setting of Replace:

```
CH1 +100% Ail
CH6  R  +100%   CH1  Switch (!ID1) (moves flaps with aileron channel when ID1 is not selected)
           +100%   P3  Switch (!ID0) (moves flaps with P3 when ID0 is not selected)
CH7  R  -100% CH1  Switch (!ID1) (moves flaps with aileron channel when ID1 is not selected)
           +100%  P3  Switch (!ID0) (moves flaps with P3 when ID0 is not selected)
```

When ID2 is selected, both !ID0 and !ID1 are satisfied (i.e., both ID0 and ID1 are OFF). Hence both the aileron input (CH1) mix and the flap input (P3) mix are ON.

Another way of achieving the same result but using one more mix by putting P3 on another channel. In this case, channel 9:

CH1 +100% Ail

CH6 R +100% CH1 Switch (!ID1)

+100% CH9 Switch (!ID0)

CH7 R -100% CH1 Switch (!ID1)

+100% CH9 Switch (!ID0)

CH9 +100% P3

One more step is needed. As you might have noticed if you programmed this example on the transmitter or in eePe, when you move P3 the flaps move down when P3 is between -100% and 0%, but move up when P3 is between 0% and 100%. We want flaps to go down with P3.

We need to modify the mix on channel 9 so it uses only negative values throughout the full range of P3. When P3 is fully clockwise the output to the flap servos is 0% and the flaps are up. When P3 is moved to the other end, the output is -100% and the flaps move full down.

CH9 +50% P3 Offset (-50%)

Thanks to Gyro Gearloose

20. Setting up Er9x to Emulate a Spektrum Transmitter

It's easy to set up Er9x on a 9x or 9XR radio to fly models that are designed to be flown with the Spektrum DX4e or DX5e, as well as more capable transmitters like the DX6. These “BNF” or “PNP” models are sold under brands such as E-Flite, ParkZone, HobbyZone and Blade. Specifically our aim here is to emulate a DX5e radio with three-position switch on channel 5 and momentary two-position switch on channel 6.

Once you have created such a definition on your transmitter (or in *eePe* on your computer), all you have to do to setup a new model is to copy and rename the model memory. Keep the original definition intact so you can use it again for another aircraft (or go back to it if you make modifications that don’t work out). It's also a good starting point for developing your own model setups.

To make your transmitter work like a Spektrum DX5e you need to create a new model to serve as a template – you might call it “DX5Emulate”. Here are the requirements:

Channel Order: TAER. If this isn't your default channel order in *Radio Setup, Controls*, you can set it just for this model by changing the individual channel mixes in the Edit Mixer screen.

Disable Trim in the Throttle Channel: You don’t want it for electric models. Go to *Mixer, Edit Mix*, highlight CH1, select *Edit* and disable the “Trim” option by setting it to OFF.

Dual Rate for Ail, Ele, Rud. Use 100% for high and 75% for low to match the DX5e. Add moderate (e.g., 20%) expo if desired. To match DX5e, use the AIL switch to control all channels.

Default mixes for channels 1-4. Leave at 100% weight. Use Limits to control throw.

Channel 5: Three-position switch. This provides full mode control for SAFE technology models (Beginner, Intermediate, Experienced). Can also use two-position switch if you prefer.

CH5 100 sIDx

Channel 6: Momentary switch. This uses the trainer switch (TRN) as “panic button” for SAFE technology models, flight mode switch for NanoQX, etc.

CH6   100 sTRN

Limits: Set to +/-80% (except low throttle). This is required because 100% in Er9x is equivalent to 125% on a Spektrum radio. If Er9x limits are left at 100%, linear servos may be damaged and modes may not work correctly.

Limits: Leave low throttle only at -100%. This is equivalent to low throttle with full down trim on Spektrum and ensures that the ESC will arm properly. Other limits should be 80%.

Limits: Reverse (invert) channels 2 and 4. To match the Spektrum direction for aileron and rudder. For some models CH5 may also need to be reversed for proper operation.

In addition to these basic changes, you may want to add the following:

Timer: Set to 5:00, Countdown, THs (throttle) as trigger.

Sticky Throttle Lock. Overrides CH1 to -100. Disables throttle stick, requires stick in low to reset.

Safety Switch 1: Ch1 X THR -100 (Using “Thr Cut” switch).

Nigel

21. Using Channels PPM1 to PPM8 in a Trainer Setup

As explained under *Radio Setup* (page 22), in a trainer arrangement a PPM stream carrying channels 1 to 8 is sent from the Slave Tx to the Master Tx through the trainer cable. From this stream, the Er9x Trainer screen on the Master Tx deals only with the first four (PPM1 to PPM4), which represent the basic stick controls: Rudder, Elevator, Throttle and Aileron.

To include any of channels 5 to 8 (PPM5 to PPM8) in the trainer arrangement requires a basic mix in the Slave Tx for each channel we intend to use. Here are some examples of possible Er9x mixes:

CH5 100% HALF Switch (GEA) (Produces outputs of 0, 100 in response to the GEAR switch)

CH6 100% sIDx (Produces outputs of -100, 0, 100 in response to the three-position switch)

CH7 100% sELE (Produces outputs of -100, 100 in response to the ELE switch)

CH8 100% P1 (Produces an output ranging from -100 to 100 in response to the P1 pot)

In the Master transmitter, for each channel we want the Slave to control, there must be a second mix to replace the original mix when the trainer switch is ON. This replace mix uses the appropriate PPMx input from the Slave as source, and must mimic the original mix, with the same throws, offset, curves, etc., so the channel behaves exactly the same whether the trainer switch is ON or OFF).

Student Control of Flaps

Suppose, for example, the trainer model has flaps that are operated on channel 5 of the Master transmitter by the three-position switch. Flight testing has shown that 80% weight is required for optimum flap action. (Please note that the details used are just examples and don't represent any real flap setup. For example, flap movement could be controlled by adjusting limits, rather than mix weight.)

The mix therefore is as follows:

CH5 80% sIDx (Three-position switched mix to operate flaps)

For the student to control the flaps, a switched mix is required on an available channel of the Slave transmitter. In the Master transmitter flaps are controlled by the three-position switch IDx, so we need a basic mix controlled by a three-position switch on the Slave (Student) Tx for this purpose; but in this case we want to use channel 6:

CH6 100% sIDx (Basic three-position switched mix to generate PPM6)

Finally, we go back to the Master transmitter and add a second mix to channel 5. This mix replaces the previous one when the trainer switch is turned ON. In so doing it shifts control from the flap switch on the Master to the corresponding switch on the Slave. The mixes on the Master now look like this:

CH5 80% sIDx (Three-position switched mix to operate flaps on channel 5.)[8]

R   80% PPM6 (TRN) (Replace mix that uses PPM6 from trainer port to control flaps when trainer switch is ON)

With this additional mix, control of channel 5 on the Master, along with the four primary controls, is transferred to the student when the instructor presses the trainer switch.

Flaperon on an EasyStar

The trainer setup on one of my models, a very old MPX EasyStar first edition with ailerons cut on the wings. First edition didn't have ailerons originally. It uses “flaperon” programming, so when flap effect is required, both ailerons move down while maintaining aileron control.

Ailerons are programmed on channels 4 for left aileron and 5 for right aileron. The main flap mix is on channel 9 and is added to the aileron channels using a mix for each with CH9 as source. Switch sIDx (three-position switch) is used to control the flaps. ID0 gives flaps-up, while ID1 and ID2 select different flaps-down positions. Curve c1 sets the value of CH9 for all three positions of the switch.

CH4 100% Ail Diff (-65) (Left aileron mix with differential, source Aileron stick)

-100% CH9

CH5 100% Ail Diff (65) (Right aileron mix with differential, source Aileron stick)

+100% CH9

CH9 100% sIDx Curve (c1) Slow (u4:d2) (Applies curve and slow)

Curve1: 0, -9, -20, -32, -45

Now we want to program a trainer setup that includes student control of flap movement. This setup is a bit more complicated than the examples before because of the curve and slow. Note that the four primary controls are configured in the Trainer Setup menu (not shown here).

To allow the student to control the flaps we need to program a basic mix in the student (Slave) transmitter, for example on CH6, using a three-position switch (sIDx).

CH6 100% sIDx

[8] Why does the mix on the Slave use 100% weight when 80% weight is required to operate the flaps? Because we want a single trainer model on the Slave transmitter to be usable for all trainer applications. So we set the weighting of its mixes to 100% and make any necessary adjustments for the particular model in the Master transmitter. For the same reason, Slow programming is done on the Master not the Slave.

Then we program the following mixes on CH9 on the Master Tx. CH4 and CH5 remain unchanged.

CH9 100% sIDx

R 100% PPM6 switch (TRN) (Replaces the first mix with the PPM6 value when TRN is ON)

R 100% THIS Curve (c1) Slow (u4:d2) The curve and slow are handled by this mix.

For more information on the THIS function, see page 31 of the manual.

Other Uses of PPM5 to PPM8

The PPMx sources allow you to give the student control not only of the flaps but also of other functions, such as retractable landing gear.

A different type of application of the PPM channels would be to use the second transmitter not for the flight controls of the aircraft but for an auxiliary function such as camera gimbals. The aileron and elevator sticks on the Slave transmitter could drive, say, channels 4 and 2 respectively (assuming an RETA setup); this could be transferred via PPM4 and PPM2 to the Master transmitter, and thence to the aircraft to control pan on channel 7 and tilt on channel 8. Thus the pilot could fly the aircraft with the Master transmitter while a second person operated the camera using the Slave.

Mixes on Slave Tx:

CH2 100% Ele (Tilt stick input)

CH4 100% Ail (Pan stick input)

Mixes on Master Tx:

CH7 75% PPM4 (Pan control)

CH8 90% PPM2 (Tilt control)

Another possibility is to have the camera gimbal controlled by two pots on the Master Tx but replace them with the trainer inputs if needed.

Here are typical mixes for the Master transmitter:

CH7 75% P1 (Pan pot input)

R 75% PPM4 Switch (RUD) (Replace mix shifts control to Slave Tx when RUD switch is ON)

CH8 90% P2 (Tilt pot input)

R 90% PPM2 Switch (RUD) (Replace mix shifts control to Slave Tx when RUD switch is ON)

Adjust the weight values in the Master transmitter to match your equipment. You can also do it in the “Limits” menu (see page 40). Use Limits for endpoints and Subtrim to adjust centring.

For a head tracker a very similar setup can be used. The difference is that the head tracker produces the PPM signal instead of a transmitter.

João

Annex B: Hardware Modifications

Programmers

A programmer is required if the transmitter is to communicate with a computer in order to upgrade the firmware, use the eePe program and backup files.

The SmartieParts Solderless Programmer Board is the most popular way to add a programmer to the 9x transmitter. As the name implies, it includes the programmer and can be installed without soldering. It normally remains in the transmitter permanently.

http://www.smartieparts.com/shop/index.php?main_page=product_info&products_id=331

Alternatively, a connector can be soldered into the transmitter to enable an external programmer to be plugged in when required. For more information, see:

http://openrcforums.com/wiki/index.php/Flashing_your_9x#What_hardware_or_mods_are_required_to_flash.3F

http://openrcforums.com/wiki/index.php/How_to_flash_your_9x

Note that the 9XR transmitter already provides a programmer connector.

A good source of programmers for use with Er9x is: http://9xrprogrammer.com/

Screen Backlight

Backlight can be added to the 9x screen with this inexpensive component from HobbyKing:

http://www.hobbyking.com/hobbyking/store/__17129__Turnigy_9X_LCD_Backlight_Kit_White_DIY_.html

How to do it:

http://openrcforums.com/wiki/index.php/9x_Full_Mod_Backlight

Backlighting is already installed in the 9XR transmitter.

Alternative Stick Assemblies

The stick assemblies (gimbals) used in the 9x transmitter can be replaced with those available for the FrSky Taranis. This thread discusses the issues involved:

http://openrcforums.com/forum/viewtopic.php?f=95&t=4281&hilit=sticks

The 9XR transmitter uses a different stick design and neither the 9x nor the FrSky gimbals will fit. Replacement stick assemblies for the 9XR are available from HobbyKing and are very inexpensive. Note that the 9XR stick assemblies are available in both mode 1 (throttle right) and mode 2 (throttle left) configurations. While the stick assemblies can be converted from one mode to the other, it's easier to buy new ones if you want to change modes.

Speaker & Haptic

This mod replaces the stock 9x beeper with a piezo speaker allowing tones to be played instead of beeps. It also adds haptic feedback by adding a vibration motor.

http://openrcforums.com/wiki/index.php/9x_Full_Mod_Speaker_&_Haptic

http://openrcforums.com/wiki/index.php/Speaker_Mod_with_Haptic_Feedback

Voice Module

This mod adds voice alerts to the 9x transmitter using a custom circuit board with an ATmega microcontroller that plays voice files contained on a micro-SD card.

http://openrcforums.com/wiki/index.php/9xVoice_Mod

Also, there was a commercial version of the Megasound9x voice module that unfortunately was discontinued. The seller has indicated that if there is enough demand he may produce them again. Here is the link to his internet page.

http://rc.emiter.hu/rc-miskolc/index.php/megasound-9x-installacioja

Update as of February 2015: We expect new developments soon. The idea is to use serial communication between the voice module and the ATmega chip on the 9x mainboard instead of connecting it to the LCD lines. A couple of trim switches might need to be rerouted. There is also an effort to build a voice module using an Arduino pro-mini 5V, 16Mhz, an SD Card module for Arduino, and an amp module. All these parts can be found very cheap on eBay. This could mean a very cheap and easy way to build a voice module, probably with very little fine soldering. Joao is working on it with support from Mike. Keep your fingers crossed.

Telemetry (FrSky)

This mod adds the ability to display FrSky telemetry values on the 9x transmitter when using firmware that supports it, like er9x or open9x. THIS IS NOT A SIMPLE MOD! It involves cutting circuit board traces & soldering fine wires directly onto processor pins. If you're not experienced with circuit boards or don't have a SMALL tipped soldering iron, consider the 9x TelemetrEZ board instead and save yourself a lot of trouble.

http://openrcforums.com/wiki/index.php/9x_Full_Mod_Telemetry

While the tutorial on the link above is excellent explaining how to reroute the AIL and THR switches, we do recommend the telemetry mod using the “Level converter”:

http://openrcforums.com/wiki/index.php/How_to_modify_your_9x_to_support_telemetry#Overview

Rotary Encoder

Er9x has support for a rotary encoder. There are a couple of different possibilities, the favourite one being a 7-way rotary encoder that includes the four directional keys. For more information:

http://openrcforums.com/forum/viewtopic.php?f=7&t=1084

DSM2/DSMX Module

This mod converts a FlySky module to a six channel Spektrum DSM2/DSMX compatible module, using the internals from an inexpensive Spektrum DX4e transmitter. The module can then be used in a 9x or 9XR transmitter running Er9x.

http://openrcforums.com/wiki/index.php/9x_Full_Mod_DSM2/X

http://openrcforums.com/wiki/index.php/How_to_modify_your_9x_to_the_DSM_protocol

A variety of other approaches to building a “hack” module can be found on the Internet.

Annex C: Glossary of Terms for Er9x

This Glossary is intended to help understand the Er9x firmware used by the FlySky/Turnigy TH9x and Turnigy 9XR transmitters. Acronyms, abbreviations and specific terms are organised by the names of the screens where they are first used.

Radio Setup

BMP: Bitmap image file. In raster image file format (128 x 64 pixels, 2-bit monochrome), used for the splash screen. Can be transferred to the radio by using the eePe computer program to embed the image in the firmware and then uploading firmware to the radio.

Sounds: monophonic sound track recordings sampled at up to 22kHz and saved in Waveform Audio (.wav) file format. For use with the Megasound voice module.

Trainer Multiplier: A value used to adjust the stick outputs of the student’s transmitter (Slave) to meet the requirements of the instructor’s transmitter (Master).

SVN: Subversion Number, part of a record keeping process used in software development, to keep track of multiple source code revisions over time.

VERS: The firmware version number stored in the transmitter’s EEPROM non-volatile memory.

MOD: Which modification to the firmware is installed.

Haptic Strength: The strength of the tactile feedback vibration used for alarms.

Model Setup

Voice Index: Number of the voice file on the SD card used to make an announcement when the model is selected or the radio turned on. Applies to radios with Voice modification only.

Trim Inc: Increment by which trim settings change for each press of the trim button.

Trim Sw: Switch used to trigger the “InstaTrim” feature, thereby converting stick deflections to subtrim values in the Limits screen and resetting trims to neutral.

Timer: Can count up or down and can be started and stopped by a physical or virtual switch.

THs, **TH%:** Timer functions that use the throttle stick as the source. **THs** causes the timer to run when the stick is off idle. TH% adjusts the timer rate according to the throttle percentage.

c1% to c16%: Timer functions that use the mixer channels 1 to 16 as the source; **cx%** adjusts the timer rate (speed) according to the channel value. When the channel is at -100, the timer is stopped. When at +100 the timer runs at normal speed. When midway the timer's speed reflects the channel position.

T-Trim: Throttle Trim, which applies to the lower half of the stick throw only. This is used for fuel powered models only. Electric models normally do not use throttle trim.

T-Expo: Applies expo to the throttle curve across the whole range, rather than to each side of the centre point as the other controls require.

Throttle Default: When this option is set to “End”, the throttle-closed stick position is defined as full DOWN, unless the throttle is reversed. When set to “Centre”, the throttle-closed position becomes the middle of the stick range.

Throttle Reverse: When turned ON, the Throttle Reverse option in *Model Setup* inverts the operation of the throttle stick for a single model, relative to the global Throttle Reverse setting in *Radio Setup*.

Auto Limits: Amount by which Subtrim settings can over-ride the limits in order to keep the throw equal on both sides of centre.

Minute call: Timed voice enunciation of the timer count via loud speaker or audio jack.

RF: Radio Frequency transmission circuit used to send signals over the air between transmitter and receiver.

Failsafe mode: Transmitter mode which allows the receiver's failsafe settings to be changed from the transmitter. Functions only with certain RF module/receiver combinations.

PPM: Pulse Position Modulation, the usual way of sending control signals by radio frequency (RF) transmission for radio control purposes. Each assembly of pulses for one cycle of data constitutes the PPM frame.

PPM frame: Assembly of pulses for one cycle of data in PPM transmission. The length is measured in milliseconds. The frame length, pulse spacing and polarity can be set in the transmitter to match the requirements of the module and receiver.

Heli Setup

Swash Type: Swash-plate design configuration in degrees: 90, 120, 120X (same as 120, but with the swash plate turned 90 degrees) or 140.

Collective Source: Virtual channel to be used on the MIXER page for pitch curve(s).

Swash Ring: Sets limits on swash servo travel. Emulates the action of a physical swash ring, a circular metal plate which restricts stick travel to the same distance in all directions.

Long cyc. Direction: Longitudinal Cyclical Direction refers to “Ele” action, which can be Normal (---) or Inverse (INV).

Lateral cyc. Direction: Lateral Cyclical Direction refers to “Ail” action, which can be Normal (---) or Inverse (INV).

Coll. Pitch Direction: Collective Pitch direction, which can be Normal (**---**) or Inverse (INV).

Flight Modes

FM0: Default flight mode.

FM1-FM4: Additional flight modes that can be assigned to unique switch positions/ combinations.

Expo/DR *(Exponential and Dual Rate)*

Rud, **Ele**, **Thr**, **Ail:** The four gimbal stick axes, which can independently be assigned: Weighting (gain/rate) and Exponential. Ultimately, these stick inputs are assigned to output channels in the **Mixer** menu.

Weight: Percentage gain or rate used to determine the amount of servo throw.

Expo: Exponential. Mathematical exponential function applied to stick travel each side of centre. In most transmitters, including those using Er9x, positive expo reduces stick sensitivity around neutral (note: in Futaba transmitters, this same action is called negative expo).

Mixes

Source: An input to be used in the mix line calculation. A pick list of possible inputs is provided.

Weight: Source value multiplier, percentage, applied before calculating the mix line output value.

Offset: A value added to the Source value before calculating the mix line output value.

Fix Offset: A value added to the Source value after calculating the mix line output value.

Trim: Applies stick Trim values in a mix. Always **OFF** when Source is not assigned to a stick.

HALF: A value of 0 to + 100 or 0 to -100.

FULL: A value of -100 to +100.

sIDx: The three-position switch as source for a mix with three values: -Weight, 0% and +Weight.

sTHR, sRUD, sELE, sAIL, sGEA, sTRN: A two-position switch can be used as source for a mix with two values: -/+ Weight.

CH1..CH16: These are the mixer outputs of all 16 channels; they are available from the list when the option “Use Output” is OFF. They can also be selected anywhere a Channel can be used as source.

OP1..OP16: These are the real outputs of all 16 channels; they are available from the list when the option “Use Output” is ON.

THIS: the combination of all mixes for this channel down to this point in processing.

Multpx: Multiplexing, a mathematical relationship with the line above. Options are **Add** (e.g., 10% + 50% = 60%), **Multiply** (e.g., 10% * 50% = 5%), or **Replace**. These are symbolically represented respectively by **+=**, ***=** and **:=**.

Curve: Applies conditional logic or a custom curve to the **Source** in computing the mix line output value. Other choices are **Diff** for differential and **Expo** for exponential.

Modes: Enables or disables the mix line in specific flight modes.

Delay: Time in seconds before the mix line output begins to be implemented.

Slow: Specifies a transition speed, time in seconds for the mix line output to go from -100 to +100.

CYC1, **CYC2**, **CYC3:** Helicopter cyclical mixer outputs.

PPM1, PPM2, ….. , PPM7, PPM8: Trainer port input channels.

GV1 - GV7: Global Variable 1 through Global Variable 7 used as an input.

Scaler: A function that applies an offset and/or scaling factors to an input.

Limits

Min, Max: Travel limits in percent.

(---): Servo travel direction normal.

(INV): Servo travel direction inverse.

Curves

Preset: Straight line curves pre-programmed into firmware.

Logical Switches

L1...L9 - LA...LI: logic switches, programmable to compare values as true or false (active when true, inactive when false). *(Transmitters with Atmega64 processor have 12 switches, others 18)*

TimeOff, TimeOn: A timed pulse trigger function that sets the time in seconds the trigger is OFF/ON.

Comparison Functions

v>val, v<val, |v|>val, |v|<val, v~=val *(Note: v~=val is only available in switches LD...LI)*

AND, OR, XOR (exclusive or)

v1==v2, v1!=v2, v1>v2, v1<v2, Latch, F-Flop

Timer

Comparison Operators: Arithmetic

> greater than

< less than

|v| absolute value of

= equal to

!= not equal to (different)

Comparison Operators: Boolean Logic

AND: both true.

OR: either one is true.

XOR: either one is true, but not both.

Assignable as Variables

Stick Inputs: Rud, Ele, Thr, Ail.

Knobs: P1, P2, P3.

FULL: Special case variable with value from -100 to +100.

HALF: Special case variable with value from 0 to +100 or 0 to -100.

CYC 1, CYC 2, CYC 3: Inputs from Helicopter cyclic mix.

PPM1 through to **PPM8:** Trainer port input signals.

CH1 through **CH16:** The Mixer outputs of all 16 channels, not including limits and safety switches, can be used as inputs to the mixer.

OP1 through **OP16**: Real Outputs of all 16 channels used as inputs to the mixer when the "Use Output" option is ON in the “Edit Mix” menu. These include settings from the “Limits” menu and “Safety Switches”.

Safety Switches

CH1 S: Channel 1 Safety Switch (may also be set to type X for sticky safety switch, type A for audio alarm, or type V for voice alarm).

VS 15: Voice Switch 15 (safety switch converted to voice switch).

Telemetry *(All telemetry readings are from sensors mounted in the aircraft unless otherwise noted.)*

UsrProto: Specifies either Winged Shadow or FrSky Hub protocol.

oXs (openXsensor): Open source project that provides sensor data to a frsky telemetry system. Works with the old Hub protocol (D), as well as with the new S-Port protocol (X). Most of the telemetry sources listed below can also be used when using the oXs connected to a frsky receiver.

A1, **A2:** analog input channels on FrSky telemetry receivers (3.3v maximum).

RSSI: Received Signal Strength Indication; strength at the receiver of the control signal from the transmitter.

TSSI: Transmitter Signal Strength Indication: Strength at the transmitter of the telemetry signal from the receiver.

Tim1, Tim2: Timer 1 and Timer 2 set in the *Model Setup* menu.

Alt: Altitude reading above ground level from the barometric sensor

Galt: Altitude reading above ground level from the GPS sensor

Gspd: Ground Speed from the GPS sensor.

T1, T2: Temperature sensor readings.

RPM: Revolutions Per Minute.

Fuel: Fuel tank capacity.

mAh, mAh1, mAh2: Battery energy consumption.

Cvlt: Voltage of lowest cell in LiPo pack.

Volt Alarm: Low voltage alarm for lowest cell in LiPo pack.

Batt: Transmitter battery voltage.

Amps: Current.

Ctot: Total value from the voltage sensor.

FasV: Pack voltage from an FrSky FAS40 or FAS100 current sensor.

FAS Offset: Correction factor applied to the FrSky FAS40 or FAS100 current sensor .

AccX, **AccY**, **AccZ:** Acceleration values in Gs for X,Y, and Z axes.

Fwat: Instantaneous power delivered from battery, calculated from current sensor voltage and amperage values.

RxV: Receiver voltage.

Hdg: Heading from the GPS sensor.

Num Blades: Number of propeller blades for RPM sensor.

GpsAltMain: Altitude source for the altitude alarm.

Vario Source: Telemetry source from which the variometer readings are received. Possible sources in Er9x are "Vspd" e "A2"

Sink Tones: Low frequency tones signalling that the aircraft is losing altitude.

AltAlarm: Altitude alarm.

Vspd: Vertical speed (rate of climb or descent).

Global Variables

GV1 GV7: Variable values that can be assigned to any weight, offset, differential, or exponential setting used in **Expo/DR**, **Custom Functions**, **Custom Switches**, and **Mixer** menus.