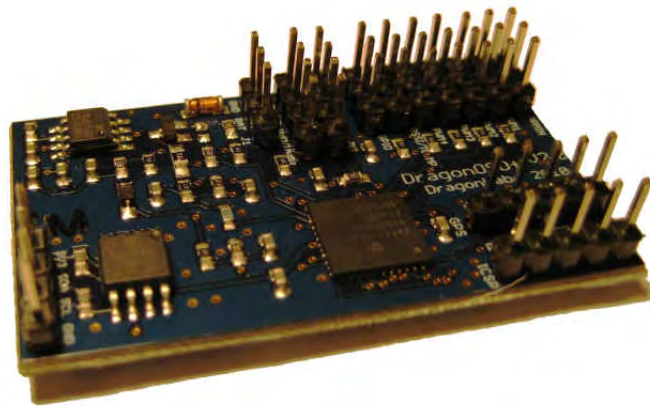


DRAGON0SD+ v2.0

Instruction Guide



DragonOSD+ v2.0 Instruction Guide by Chris Raynak (www.rctoolbox.no-ip.com)

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without written permission from the author or DragonLabs, except for the inclusion of brief quotations in a review. Copyright © 2010 DragonLabs

Table of Contents

<u>Section</u>	<u>Page</u>
Overview -----	4
Required Accessories-----	5
Optional Accessories-----	7
DOSD+ Board Connections -----	8
Connecting Devices to the DOSD+ -----	17
Configuring the DOSD+ -----	21
On-Screen Menu -----	21
Initial Configuration via Serial Terminal-----	24
Serial Terminal Commands-----	26
Alternative command set-----	28
Installing the DOSD+ -----	29
DOSD+ Startup -----	30
Display Elements-----	34
Configuration Menu-----	37
Main Menu -----	39
OSD Settings Menu -----	41
GPS and Alert Settings Menu-----	46
Autopilot Settings Menu -----	48
Calibration Menu -----	51
Custom Settings Menu-----	53
Using the on-board position logging function-----	56
Setting the on-board failsafe -----	57
DOSD+ Autopilot configuration -----	58
DOSD+ calibration procedures -----	60
RSSI Calibration-----	60
Current and mAh Calibration -----	61
V1 and V2 Calibration-----	61
Powering your OSD -----	63
FTDI to DOSD+ Adaptor Fabrication-----	64
Getting the PPM Stream Data Out of Your Receiver-----	70
Finding the PPM signal -----	70
Building a PPM Buffer-----	70
Identify your PPM Signal Lead and Provide a Connection Point-----	72
Assembling your Buffer / Isolator-----	73
PPM Buffer Wiring Block Diagram -----	85
Using an 8 Channel DIY Drones PPM Encoder Board -----	87
How The Encoder Works-----	88
Encoder Requirements -----	88
Deciding on a Servo Jumper Configuration-----	91
Encoder Block Connection Diagrams -----	93
Testing The Encoder -----	95
Configuring the DOSD+ for Your Radio -----	96
Using the Serial Port -----	97
DOSD+ Firmware Upgrading-----	98
Supported I2C Peripherals-----	105
Block Connection Diagram of Typical DOSD+ System -----	107
Disclaimer of Warranty / Limitation of Liability -----	108

Overview

Congratulations on your purchase of the DragonOSD+ V2.0 by DragonLabs. We suggest you take the time and read over these set up instructions before you begin. The DragonOSD+ (DOSD+) is an exciting leap forward in UAV technology that we are sure you will enjoy.

The DragonOSD+ has many features and options that allow you to customize the behavior of the OSD. With the DOSD+ you have such features as:

Flight logging – GPS tracks can be downloaded and viewed in Google Earth

Autopilot – can be set to fly the plane home automatically

Home indicator – will always indicate the direction back to launch site

mAh meter – will give an indication of mAh consumed

Blank screen – you can turn off all on screen display for video purposes

Of course, the OSD will also perform all the other standard functions such as the display of heading, altitude, time elapsed, battery voltage(s) and so on.

You probably won't need to fiddle too much with the options if you are just using the OSD in the most basic form, but if you want to get the most out of the OSD and to use the more advanced features, it is imperative for you to understand how to properly configure the OSD and understand something about how the device works.

Required Accessories

The following accessories are recommended for your DOSD+

- A suitable power supply
- A serial-TTL to USB converter cable
- A suitable GPS Module

Power supply:

The bare minimum hardware you need to go along with your other FPV hardware to make your DOSD+ function is 2S1P Lipo battery pack. The board will accept from about 4.6V up to 20V (although we do not recommend using more than about 16V). You can also use other means to power your board, and these options are listed in the "Powering Your OSD" section. This bare minimum configuration will activate the board and display any voltage and sensor inputs you may have attached to it. But where is the fun in that? To take full advantage of all the DOSD+ can offer you, we suggest these additional components be used:

Serial – TTL to USB Converter Cable:

Serial Port

The ICSP programming port on the DOSD+ doubles as the serial port. Most of the time you won't even need to think about the programming port so just treat it as a serial port. This is a TTL level port so you will need to use a serial-TTL to USB converter to connect it to your PC. Some possible converters can be found on e-bay or SparkFun:

http://shop.ebay.com/i.html?_kw=usb%20ttl&_fcid=180&_localstpos=&_sc=1&_sop=15&_sticky=1&_stpos=&_gbr=1
http://www.sparkfun.com/commerce/product_info.php?products_id=9115
http://store.diydrones.com/product_p/ttl-232r-3v3.htm

If you already have a USB converter from a manufacturers (RVOSD, Flytron, etc.) those will also work. You will need to make sure that the pin-outs of the connectors match up with that of the OSD header. See the section "FTDI to DOSD+ Adaptor Fabrication" for suggestions on how to make up an adapter if you use the FTDI cable.

GPS Module:

A GPS module will allow you the DOSD+ to help you navigate in the air and enable many more features. Currently there are only three viable choices of GPS families on the market that offer update rates of greater than 1Hz and still reasonably affordable. The three families are MediaTek (MTK), Antaris (u-blox) and SkyTrax (Venus). The most common of these are the MTK based modules, which include models such as LS20033, LS20031, EB-85A, and other. Many companies offer GPS'es based on the MTK chipset. They were the first to offer economical 5Hz modules and have recently updated their units to 10Hz capability.

<http://www.rcgroups.com/forums/showthread.php?t=1212682&highlight=locosys>
http://www.sparkfun.com/commerce/product_info.php?products_id=8266
http://www.sparkfun.com/commerce/product_info.php?products_id=8975
<http://www.dpcav.com/xcart/product.php?productid=16502&cat=250&page=1>

The next chipset, the Antaris, comes from Atmel, and offered up to 4Hz capability although it

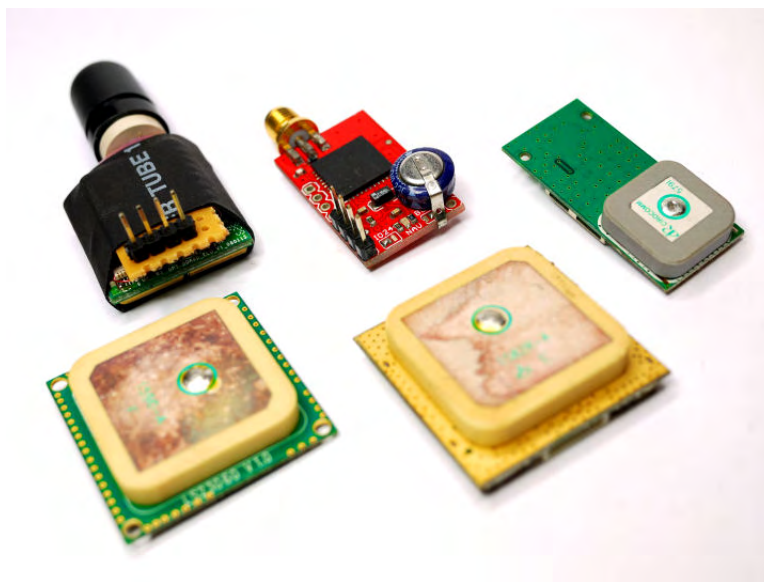
has been mentioned that some of the newer ones might even handle up to 10Hz. The u-blox modules uses this chipset and has been noted for its good accuracy as compared to the MTK chipset. GPS modules such as the GS407 would be an example of such a product.

http://www.sparkfun.com/commerce/product_info.php?products_id=9436
http://www.sparkfun.com/commerce/product_info.php?products_id=9566
<http://store.diydrones.com/ProductDetails.asp?ProductCode=SPK-GPS-GS407>

The SkyTrax offering uses the Venus chipset and is capable of 10Hz update rates. This is a rather new module and it is uncertain at this point if there will be many modules based on this chipset. Sparkfun does, however, offer two boards that use the chipset and they appear to work quite well when paired with an active antenna.

http://www.sparkfun.com/commerce/product_info.php?products_id=9758
http://www.sparkfun.com/commerce/product_info.php?products_id=9133
http://www.sparkfun.com/commerce/product_info.php?products_id=9171

Unfortunately, the previously very popular SiRF III chipset still does not support more than 1Hz update rates and so they would not be particularly suitable for use with GPS based autopilots. This would include modules such as the EM-406.



Optional Accessories

The following accessories will enhance your DOSD+ experience even more:

- A PPM data stream compatible Rx or PPM Encoder for your Rx.
- A current sensor
- A RSSI input buffer
- I2C Peripherals

PPM data stream compatible Rx or PPM Encoder for your Rx:

The DOSD+ expects a PPM-stream as the input from the receiver. In other words, unless you have a PPM-stream output from your receiver (not a regular servo PWM output) you will not be able to access the on-screen menu and will be limited to configuring the OSD via the serial terminal. The advantage of the PPM-stream scheme is that you only need one signal cable from the receiver to the DOSD+. If you have a receiver that outputs a PPM-stream (such as the DragonLink or Thomas's LRS) then it is just a matter of plugging in the cable from the PPM output on the receiver to the PPM input port on the DOSD+. If you have an analog receiver, you will need to either use a PPM Encoder or modify the receiver you plan to use to extract the PPM-stream.

The DOSD+ can be used in a basic mode as is. Just plug in current sensor (and GPS if you like) and you're done. However, there are a host of features that you may want to access as you grow in your FPV experience. The DOSD+ is ready to grow with you when you're ready. To do this, you have two options. The first is to use the PC Commander configuration software tool. With this simple utility, you can configure most of the DOSD+ features while on the ground. The second is to use a receiver with a PPM output. The advantage of taking this slightly more complicated approach is that you will be able to access the DOSD+ on screen menus and change features in at the field and while in flight "on the fly". Some receivers (such as the Dragon Link) have a PPM output. For other receivers take a look at the section "Getting the PPM Data Stream Out of Your Receiver".

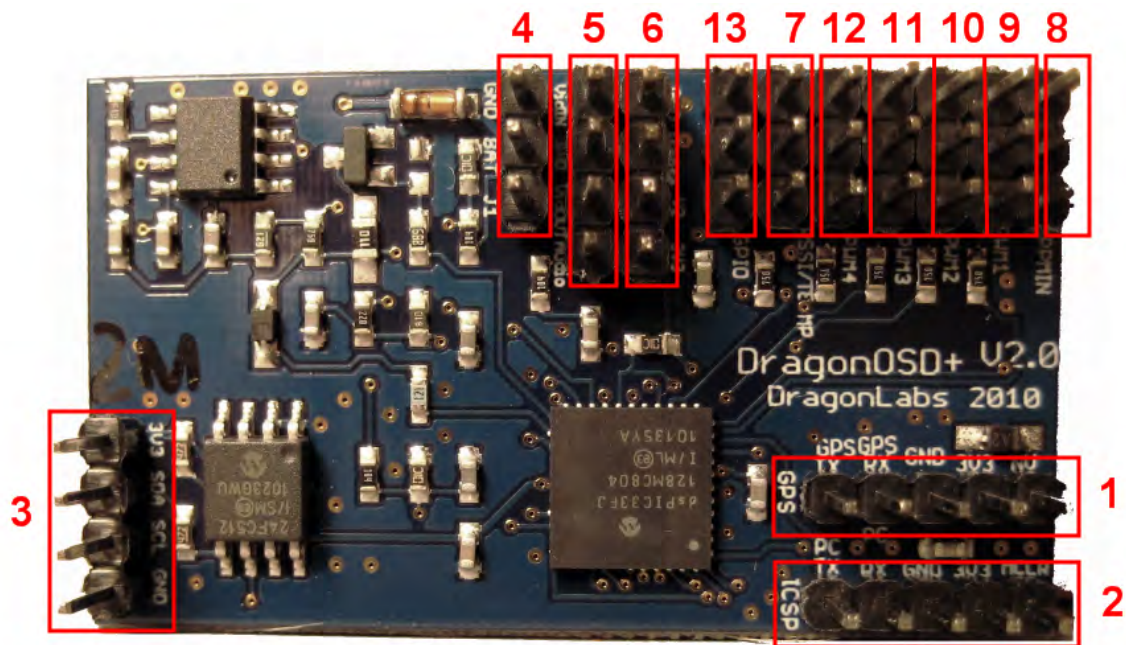
I2C Peripherals:

The I2C Peripheral connector can be used to attach various types of sensors to the DOSD+. This includes barometric altitude sensors, true air speed sensor, tilt compensated magnetic compass, temperature sensor, IMU, and so on. When supported peripherals are attached, the DOSD+ will automatically detect and recognize them at boot-up and will make use of them (subject to user's menu settings). These peripherals can be daisy-chained so that several of them can sit on the same bus and operate at the same time. As we go along, we hope to add more supported peripherals to our list. See "Supported I2C Peripherals" for a list of currently supported devices.

DOSD+ Board Connections

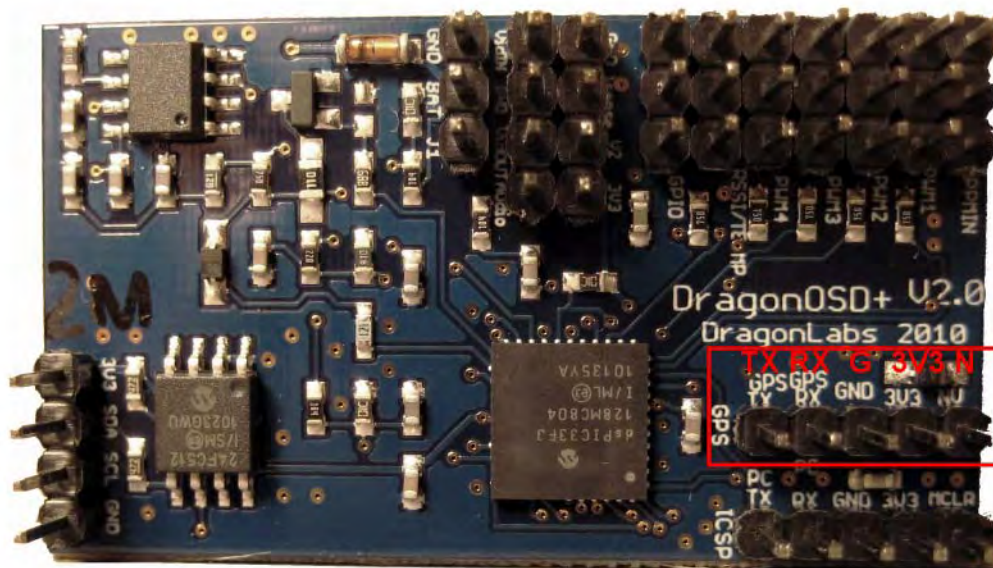
The DOSD+ board has 10 sets of connectors that are used (1 is unused). These are:-

1. GPS
2. Serial/ICSP programming
3. I2C Peripheral
4. Power
5. Video/Audio input/output
6. Current/Voltage sensor
7. RSSI/Temperature/Analog sensor (unity buffered)
8. PPM stream input (unity buffered)
9. PWM Servo output 1
10. PWM Servo output 2
11. PWM Servo output 3
12. PWM Servo output 4
13. GPIO –currently not used.



1. GPS Connection:

The GPS modules will need to be connected to the DOSD+ using suitable cable connections. **Do NOT assume that the default connector pin-outs will match the pin-out required by the OSD board.** There are 4-primary connections (+3.3V, Gnd, RX, and TX). Note that the DOSD+ only supplies 3.3V so the GPS modules need to be able to operate on 3.3V (almost all the newer ones and the listed ones do.)

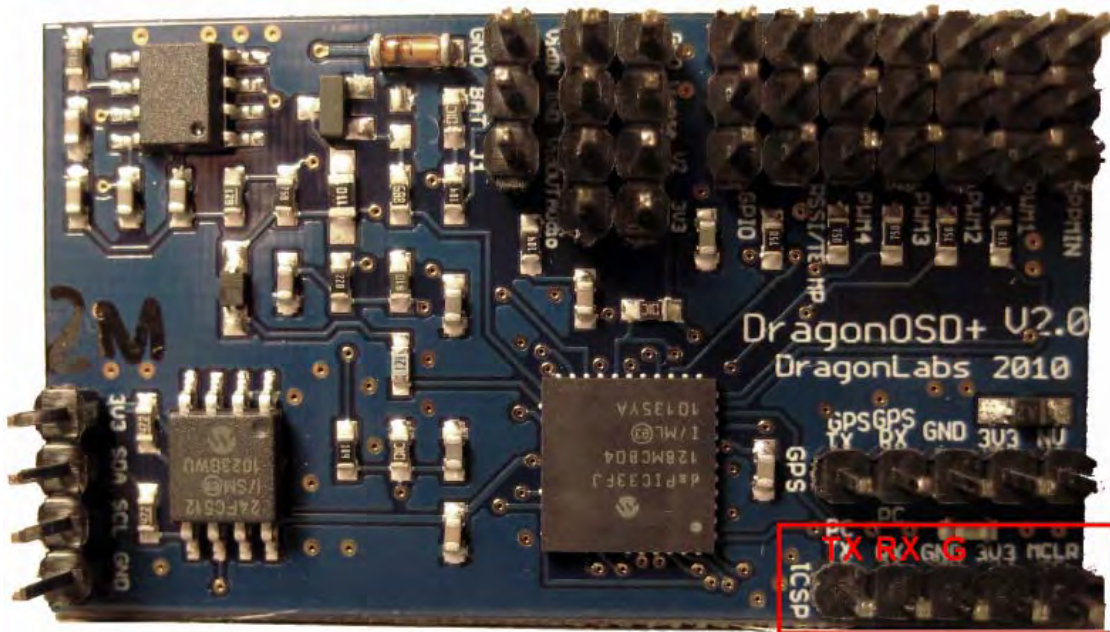


The extra pin (last on the right in the photo) is a diode blocked line to charge a supercap (if there is one) and is sometimes unneeded, since it is usually not required, except maybe for the LS20033 GPS.

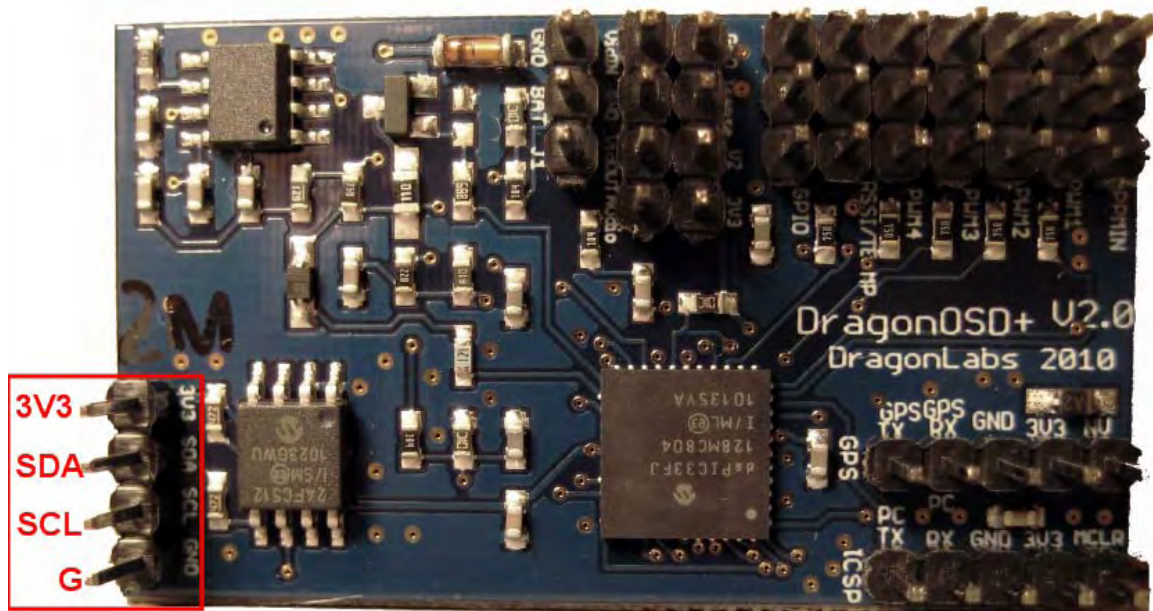
The DOSD+ will typically attempt to auto-detect the module baud rate, and then the module type. If this succeeds, normal operation will continue. However, if the auto-detect phase fails, it will fall back on the GPS choice selected in the configuration menu. Under normal circumstances though, this should not fail and if it does, it probably indicates a more complex problem at hand. At the point of writing, the DOSD+ will not recognize the SiRF chipset to discourage its' use.

2. Serial/ICSP Programming Connection:

Some functions can only be set via the serial port - including the setting of the callsign to be displayed and the downloading of flight logs. The firmware can also be updated via the same serial port so new special programmer is required to update the firmware (although if something should go wrong - you may need a programmer to fix it). Please see the section "Using the Serial Port" for further information.

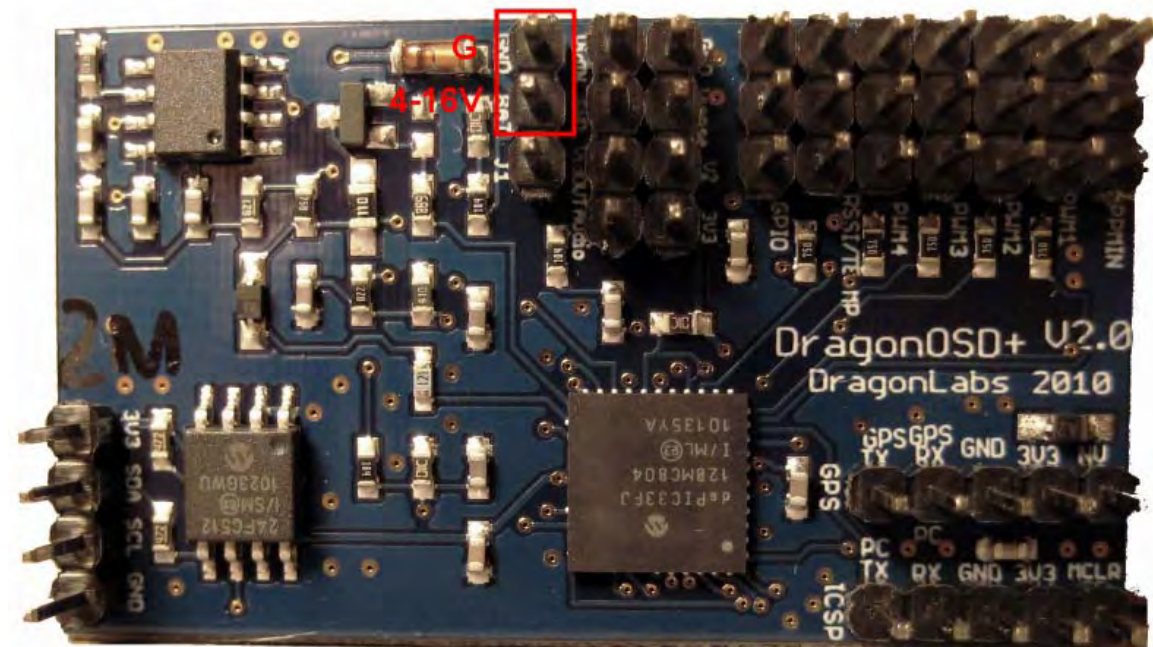


3. I2C Peripheral Connection:



4. Power Connection:

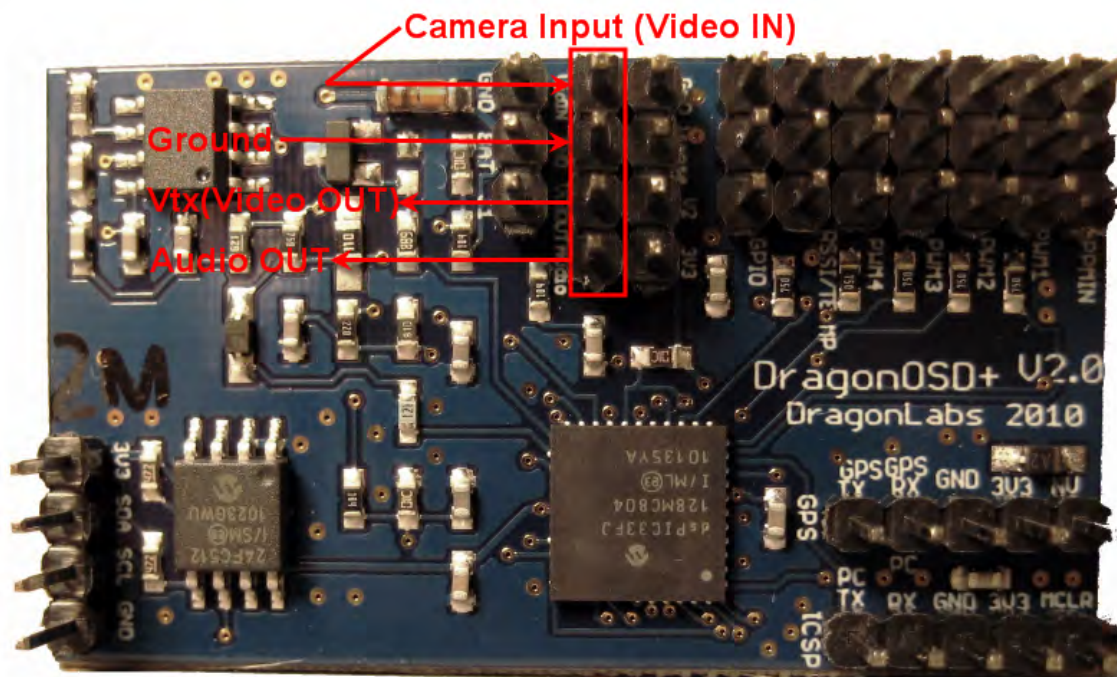
Only two pins are used for the power connection. Ground is closest to the edge of the board. This input is protected by a diode against accidental reversal of power, nevertheless it goes without saying that this is not recommended at all.



5. Video/Audio input/output Connection:

All four pins are used for the Video/Audio connections it is important that you properly connect your camera, otherwise the video output may exhibit strange results. Starting from the edge of the board the connections are:

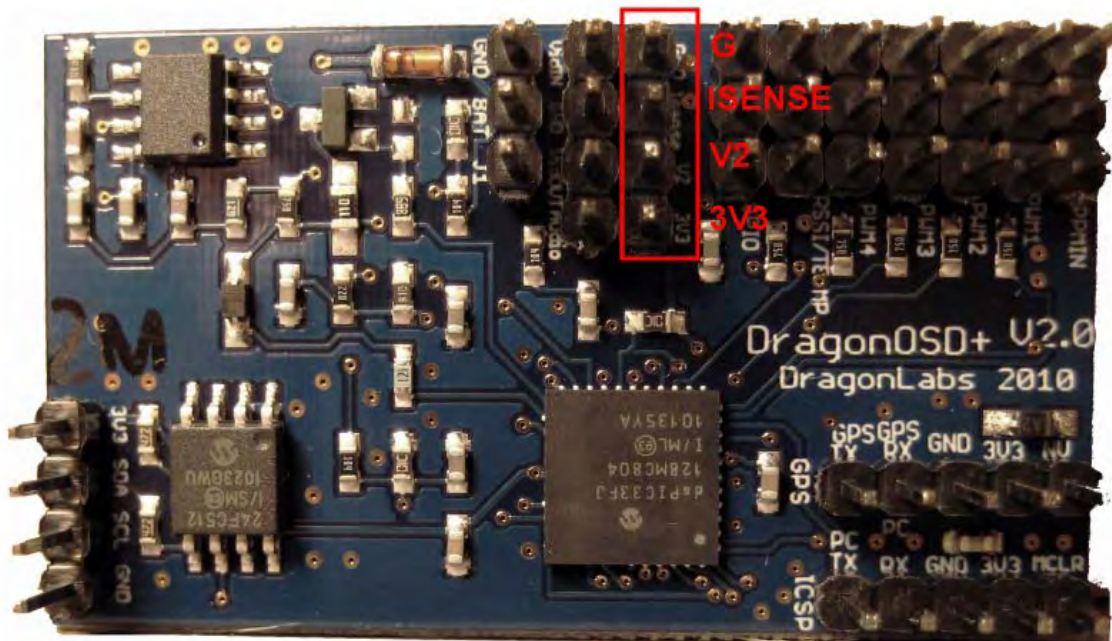
- Input from camera into the DOSD+ (VIDEO IN)
- Ground
- VIDEO OUT of the DOSD+ (this goes to your Vtx Transmitter)
- AUDIO OUT (this goes to your Vtx Transmitter also)



6. Current/Voltage sensor Connection:

This connection can be used to monitor your flight pack voltage and current. To measure the current, you need a sensor attached to your flight pack and the DOSD+ should be calibrated. See "DOSD+ Calibration Procedures " section for further information. Starting at the edge of the board the connections are:

- Ground
- ISENSE (input from current sensor)
- V2 (Voltage 2 input)
- 3v3 (3.3v output not normally used)

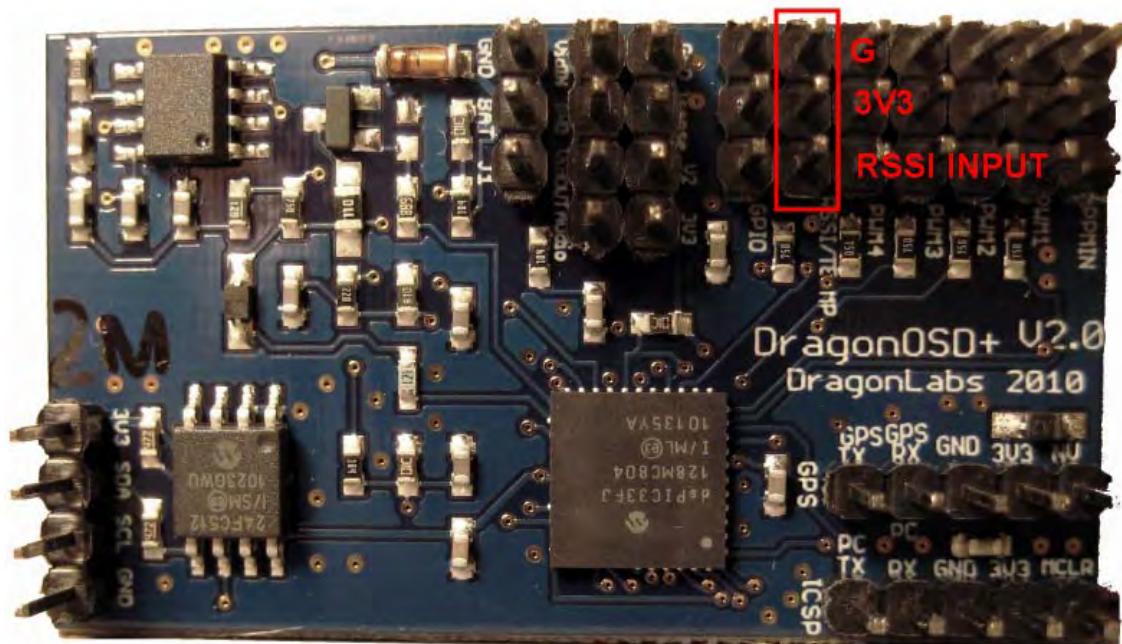


7. RSSI/Temperature/Analog sensor Connection:

The RSSI input is also shared with the analog temperature sensor. Note however, that when used as an RSSI input, you **ONLY USE THE FIRST AND THIRD PINS of the header. DO NOT CONNECT ANYTHING TO THE MIDDLE PIN.** The middle pin carries 3.3V and is meant to power the analog temperature sensor and should not be connected to any other power source. Starting at the edge of the board the connections are:

- Ground
- 3V3(3.3v output for analog temperature sensor)
- RSSI (Receiver Signal Strength Input)

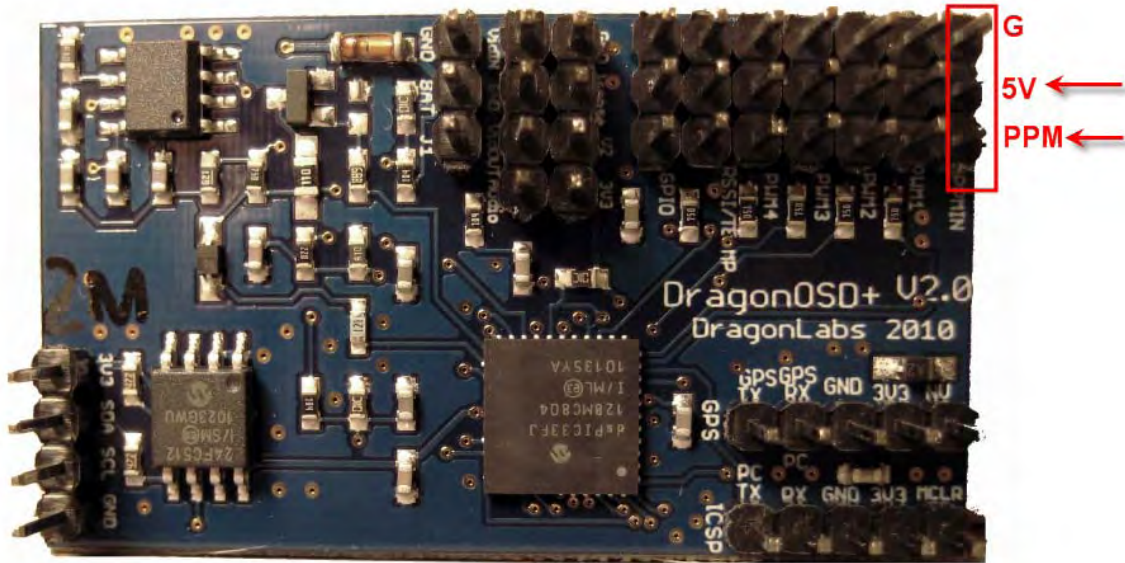
On the DragonLink receiver, the RSSI output is isolated from the receiver so it has an output to connect to the DOSD+ for RSSI monitoring. The RSSI input connection on the DOSD+ V2.0 board is buffered, so no additional drain is placed on your receiver.



8. PPM Stream Input Connection:

Input connection from Dragon Link, PPM encoder board, or PPM receiver output. Starting at the edge of the board the connections are:

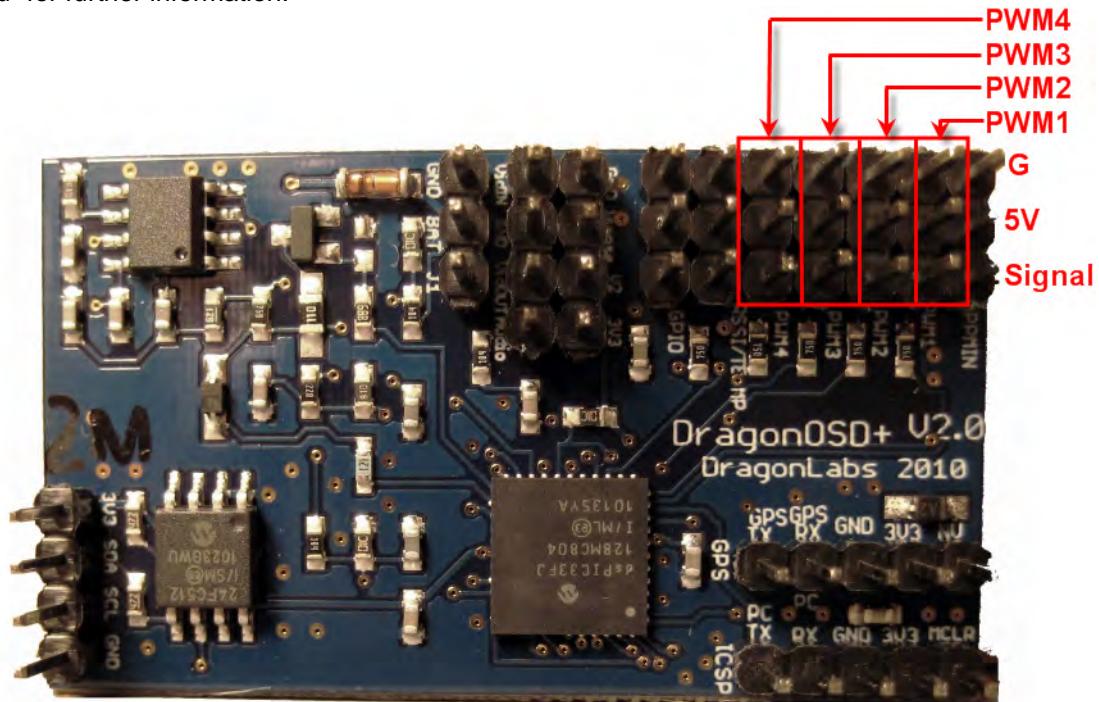
- Ground
- 5V (5 volt input from encoder or other device TO DOSD+)
- PPM (PPM Signal input to DOSD+)



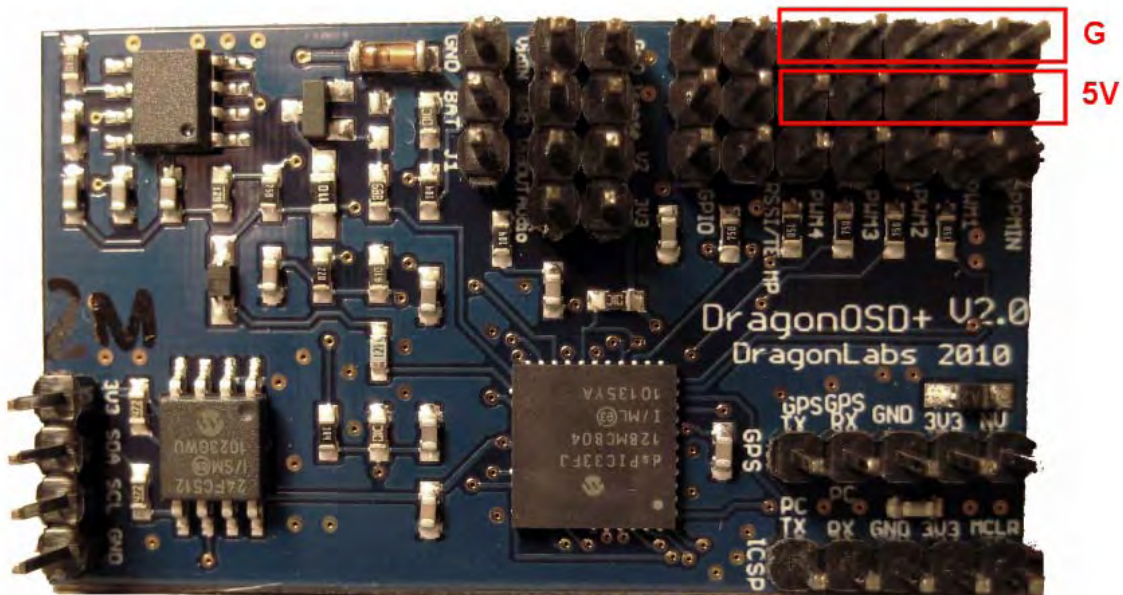
While not required, the 5V input from your PPM Encoder or other device can be used to supply power to the servos connected to the PWM1, PWM2, PWM 3 and PWM 4 outputs from the DOSD+. Please see the section “Getting the PPM Stream Data Out of Your Receiver” for further information. The PPM input is unity buffered on V2.0 of the DOSD+ so it should not impact the operation of your existing Rx, if you tap the PPM line.

9.10. PWM Servo output 1 and PWM Servo output 2 Connections:

The PWM1,2,3 and PWM4 servo outputs are where you would normally connect your elevator, rudder, throttle and aileron servos. The exact function of each is determined by you when you program your DOSD+ for flight. Please see the section “Configuration Menu, Custom Settings Menu” for further information.



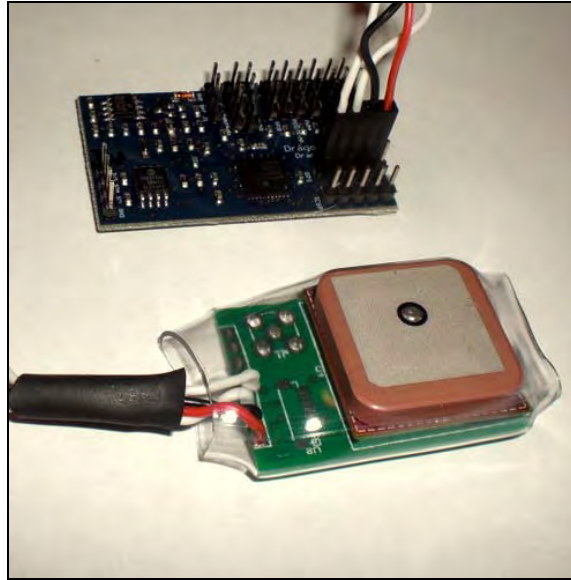
If you plan on using these outputs, you need to supply 5 volts, either at the PPM 5 volt input, or at one of the 5V servo terminals.



Connecting Devices to the DOSD+

Connecting the GPS module

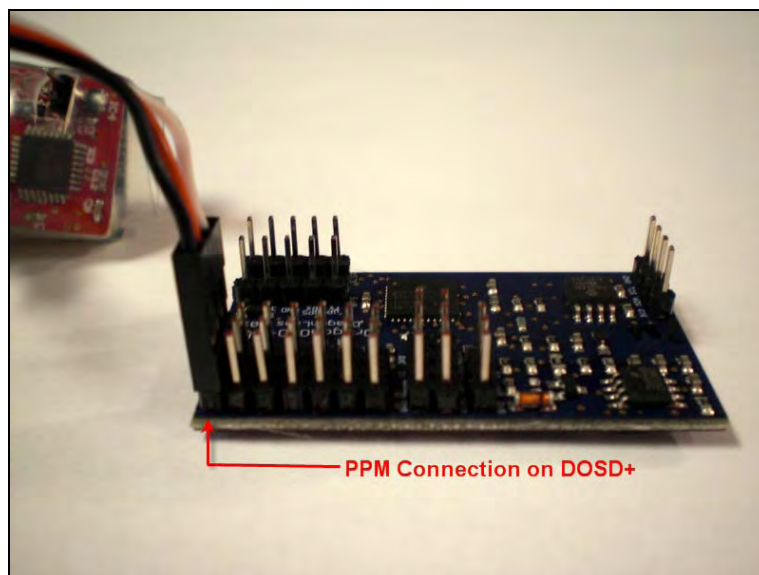
The GPS modules will need to be connected to the DOSD+ using suitable cable connections. **Do NOT assume that the default connector pin-outs will match the pin-out required by the OSD board.** There are 4-primary connections (+3.3V, Gnd, RX, and TX). Note that the DOSD+ only supplies 3.3V so the GPS modules need to be able to operate on 3.3V (almost all the newer ones and the listed ones do.)

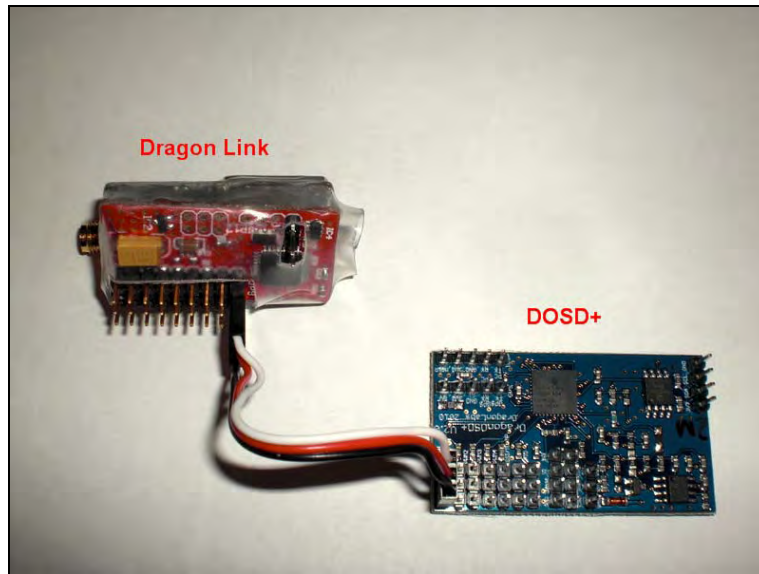


The DOSD+ will typically attempt to auto-detect the module baud rate, and then the module type. If this succeeds, normal operation will continue. However, if the auto-detect phase fails, it will fall back on the GPS choice selected in the configuration menu. Under normal circumstances though, this should not fail and if it does, it probably indicates a more complex problem at hand. At the point of writing, the DOSD+ will not recognize the SiRF chipset to discourage its' use.

Connecting the Receiver and Servos

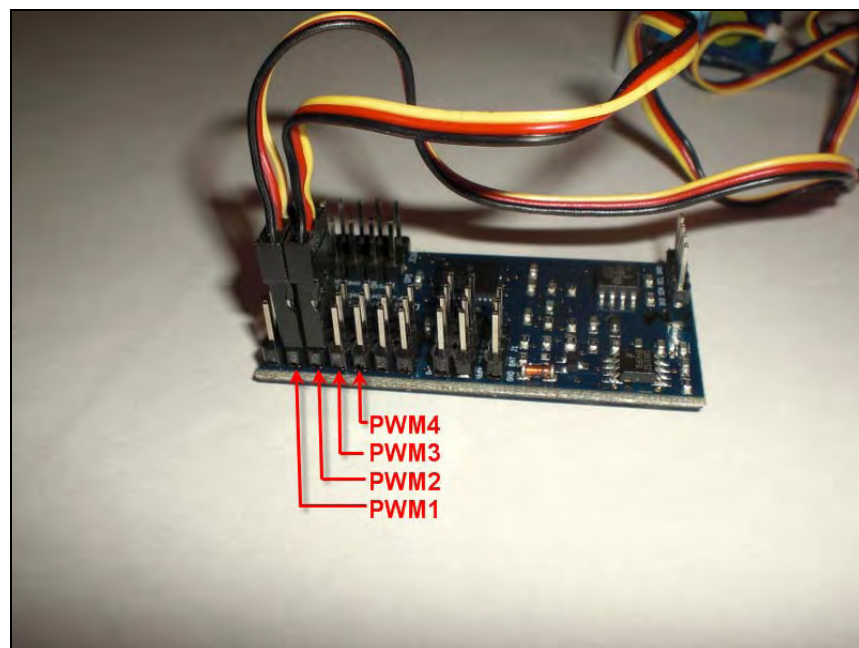
If you have a Dragon Link receiver, connect RX channel 9 from the Dragon Link to the PPM input on the DOSD+.





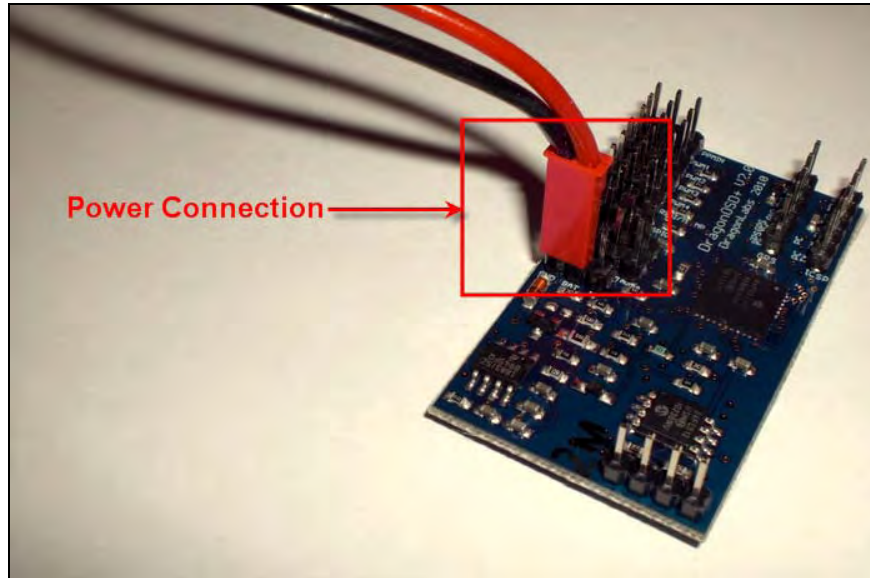
If you don't have a Dragon Link receiver, you need to connect your receiver's PPM output here, or use an encoder device to create a PPM output from your receiver. Please see "Getting the PPM Stream Data Out of Your Receiver" section for further information.

Your elevator and rudder servos should connect to the PWM1 and PWM2 outputs as shown. Additional PWM3 and 4 outputs can be used for other controls.



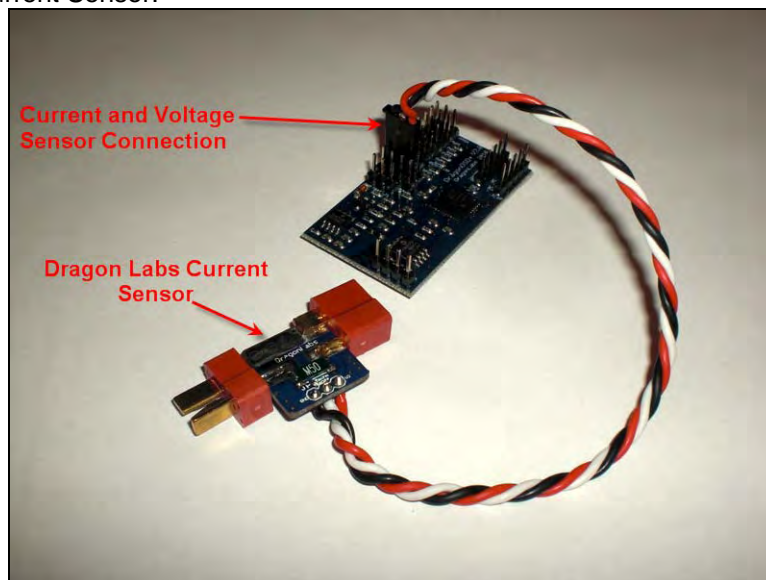
Connecting the Power Supply

Only two pins are used for the power connection. Ground is closest to the edge of the board. You can power the DOSD+ with a separate battery, or by way of other various options. Please see the section “Powering Your OSD” for further information.



Connecting Sensors and other Peripheral Devices

Now is also a good time to connect any other sensors and devices to your DOSD+, such as the DragonLabs Current Sensor:



Please refer to the connection section and the documentation which came with your sensors to verify proper hookup.

Configuring the DOSD+

There are two methods of configuring the DOSD+ - either via the on-screen menu or via the serial port.

On-Screen Menu

As previously stated, **the DragonOSD+ expects a PPM-stream as the input from the receiver**. In other words, unless you have a PPM-stream output from your receiver (not a regular servo PWM output) you will not be able to access the on-screen menu and will be limited to configuring the OSD via the serial terminal. If you have an analog receiver, you will need to refer to the section "Getting the PPM Data Stream Out of Your Receiver" about how to extract the data PPM stream.

The PPM-stream carries the control information for multiple channels (usually up to 8, but can be as low as 5 or as high as 12). What you need to understand at this point is that different radios have different conventions for their channel assignments. For example, with J.R. radios - channel-1 is usually the throttle channel, whereas Futaba radios use channel-3 for the same. These conventions usually affect only the first four channels.

The DOSD+, when first powered up, does not know what kind of radio - and therefore what kind of channel assignment - is in use. As such it needs to have some way of identifying the radio. If the radio type has not been specified by any other means, the usual startup radio-mode is the "undefined" mode. In this mode, it will attempt to guess the radio type based on the PPM-stream it is receiving. If the receiver is not active at this point, or there is no PPM-stream available, the OSD will be unable to guess the radio type that is being used. In this scenario then, it will fall back to the default assignments which correspond to a J.R. radio. Note that many receivers will not output anything unless it is receiving a signal from the radio.

Assuming that the receiver is active, and there is a PPM-stream going into the DOSD+ when it is powered up, it will attempt to analyze the received PPM-frame. It is looking to see which channel might be the throttle channel and for this to work, your throttle channel needs to be at the zero-throttle position or close to it. If it finds such a channel among the first four channels, it will assume that it is the throttle channel and assign the other channels according. If the auto-detection of radio type fails, you will probably not be able to easily configure the radio via the on-screen menu until you manually assign the radio type via the serial terminal.

To summarize:-

- 1. When first powered up, radio type is undefined**
- 2. When radio type is undefined, the OSD will attempt to guess the radio type from the PPM-stream**
- 3. Therefore the PPM-stream needs to be present when the DragonOSD+ is powered up**
- 4. The throttle stick needs to be at the zero position while the other usual sticks at the centered position**

Assuming all these conditions are fulfilled, the radio type should be correctly detected. Right now, four conventions are recognized - J.R., Futaba, Multiplex and Sanwa. The detected radio type is not automatically saved until you perform a save to EEPROM from the main menu. Beyond this, you can also manually assign the channels you want to use for each of the control functions (discussed later.)

To configure the DOSD+ via the on-screen menu, you will need three channels to be correctly assigned. These are:-

1. aileron channel
2. elevator channel
3. mode-control channel

The aileron and the elevator channels should be automatically detected in the above procedure. The default assignment for the mode-control channel is channel-5. This is the channel that is used to switch the DOSD+ into any one of the four possible modes:

- 1. Normal flight mode**
- 2. Menu mode**
- 3. Auto-pilot mode**
- 4. No OSD display mode**

Note: It is quite possible that the value of the control channel as you start up (before you start configuring the radio) is such that the OSD might go into any one of these modes - including mode 4 where nothing is displayed. You just need to get it configured for proper operation.

As a rough guide - the servo PWM settings are as follows:-

MENU - above 1.9ms (>90%)
NORMAL - between 1.6ms to 1.9ms (60% to 90%)
AUTOPILLOT - between 1.2ms to 1.6ms (20% to 60%)
NODISPLAY - less than 1.2ms (<20%)

Now, to begin configure the DOSD+ via the on-screen menu, you will need to make the channel-5 output such that it is at a level that triggers the menu mode. An easy way to do this will be to assign a slider or knob device to channel-5. This way you can just move the slider or knob until the menu shows up. Bear in mind that if you have adjusted your end points for the channel, you may need to move them back to the original positions so that your slider or knob can traverse the entire servo range. The preferred setup is to assign a 3-way switch device to the mode-control channel. A typical set it up would be so that the three positions of the switch correspond to the autopilot, normal and menu mode respectively. This allows the most convenient access to each mode.

If you have done the above correctly, you should now be able to activate the menu and once inside, you will be able to use the aileron and elevator sticks to navigate the menu and adjust the settings. However, it is possible that you find the orientation of one or both of the sticks reversed in relation to how it moves the selection on the menu. In this case, you will need to go into the **"Custom Settings"** screen. Select that on the main menu with the elevator stick and move the aileron stick to the right (or left if reversed) and you will be inside that menu.

You will see two selections called **"AIL Menu direction"** and **"ELE Menu direction"** near the bottom of the list. You can go to these selections and reverse the orientation if you so wish. Bear in mind that it can get a bit tricky if you are reversing the aileron direction because the change takes effect instantaneously. Once you have done this, you should be able to normally navigate around the various menu pages and change the settings.

The stick controls are proportional in the menu. This means that if you push the aileron stick further right or further left, the rate of change will increase accordingly. This will help if you need to make a big change in the settings. If you have dual-rates active on the sticks, this will limit the range of the sticks electronically and you may not be able to get the full proportional range to achieve the fastest rate of change. So if you find that your settings are changing too slowly even when the sticks are at the limit, you should check if you have DR active and if the end-points for that stick has been reduced. Many other settings can affect the limit of the stick - such as differential ailerons, which would give you more throw in one direction than the other.

In the **"Custom Settings"** menu, you will find that you can re-assign all the channels in use as well, including the control channel (CTRL channel). Bear in mind that the change here is instantaneous so if you make a change from 5 to 6, for example. It will immediately use channel-6 as the control channel and if channel-6 isn't already configured to activate the menu option, you

may be kicked out of the menu into whatever mode channel-6 current corresponds to. Also, if you happen to overshoot the channel and accidentally go to 7, you will be stuck as well.

Fortunately, all these settings are not saved even though they come into effect for the current OSD session (which ends when you power down the board). So you can play with all the settings and when you are satisfied with it, go back to the main menu and select "**Save to EEPROM**" and move your stick right until you see the word "**SAVED**" appear on the line. These settings, including the radio mode and various assignments are now saved as default and will be used in all subsequent sessions. Once the radio mode has been saved, it will no longer attempt to auto-detect this in the future so that even if the PPM-stream is not present or the receiver is not active when the DragonOSD+ is powered up, it will still use the saved settings.

Channel-5 is the default control channel because with some Futaba receivers, the internal PPM-stream only contains five channels. If, however, you want to use a different channel and are unable, for one reason or another, to use channel-5 - you should configure the mode-control channel via the serial terminal. Likewise, if the radio type detection fails or if you have an unconventional channel assignment, you can also do that via the serial terminal.

Initial Configuration via Serial Terminal

To configure the DOSD+ via the serial terminal, you will need to connect the serial port to a PC/laptop with suitable terminal software (such as HyperTerm or TeraTerm). The connection is made through a UART/serial-to-TTL device. Please see the section “FTDI to DOSD+ Adaptor Fabrication” about connecting your DOSD+ board’s serial connection.

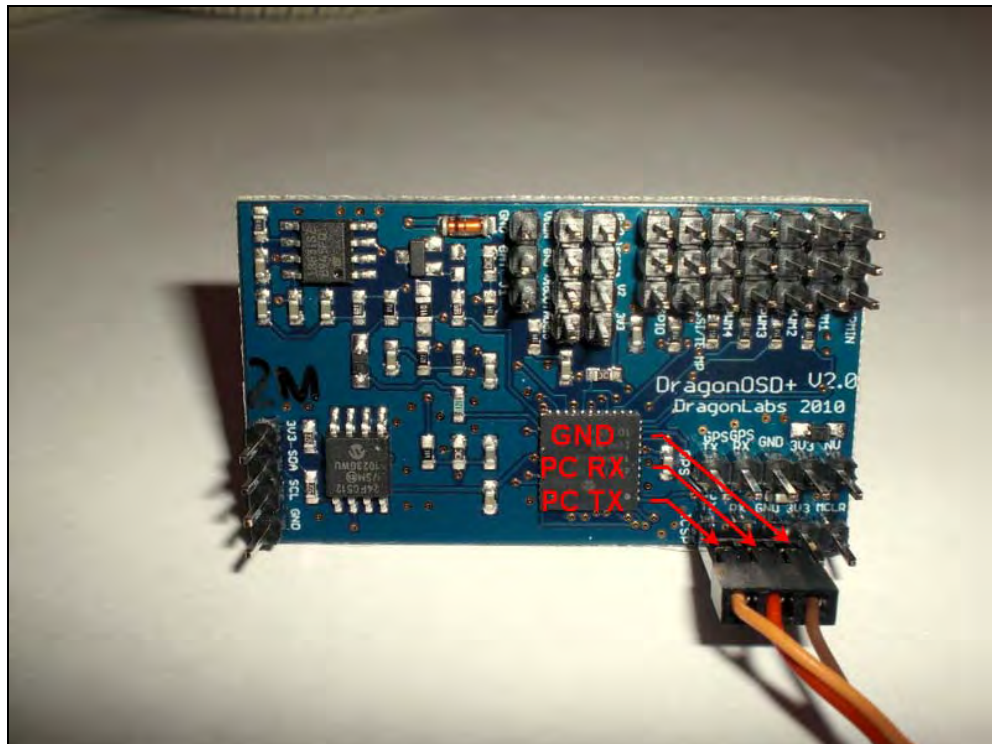
You should also have the serial terminal software open and the correct COM port selected. The serial port should be set up for 38400 baud, 8-bits, No parity, and 1-stop bit (8N1 in short.)

(The baud rate has been changed to 115200 from firmware version 7.5Beta11 and upwards.)

When you have this connected correctly, you should see a copyright and version message printed on the screen when the DragonOSD+ is powered up. You should also see a ">" cursor at which you can type various commands.

If you don't see that, some basic things to check are:

0. Is the USB converter correctly installed and recognized as a COM port?
1. Is the correct COM port being selected?
2. Make sure no other terminal program (HyperTerm or TeraTerm) is running when updating)
3. Are the serial settings correct? (38400 baud, 8N1)
4. Are the RX and TX lines reversed or crossed?
5. Is the serial port being connected to the correct header pins?
6. Is the OSD powered up?



Note that the serial connection consists of 3-wires connected to the ICSP port. The three wires are marked out in the photo above. Do not connect anything to the remaining two pins.

At this point in the initial configuration, there are several commands of particular significance. You can assign the radio type manually by typing:-

SET RADIOMODE FUTABA

If you are using a Futaba radio. Other options are JR, SANWA and MPX. You can also assign the individual channels with the command:

SET CTRLCHAN 7

If you want the mode-control channel to be channel-7. Other assignable channels are AILCHAN, AIL2CHAN, ELECHAN, THROTTLECHAN. These assignments determine how the DOSD+ will interpret the incoming channels. The actual channel usage is determined on the radio itself.

Another setting that can only be done in via the serial terminal is setting of the callsign. The callsign can be no longer than 10-characters and will be displayed for one-minute at the top of every ten-minute interval if you have enabled the callsign display option. You assign the callsign as follows by typing:

SET CALLSIGN DRAGONOSD

Your callsign will become "DRAGONOSD". Anything longer than 10-characters will be ignored.

As with the on-screen menu, these settings are put into effect immediately but are not automatically saved. To do that, when you are satisfied with your changes, type:

SAVE

and you're done. If for some reason you botch up the settings and want to revert to the original default state, just type:

DEFAULTS

and cycle the power to the DOSD+. The DOSD+ will come up in a default settings state. You can also accomplish the same effect using the "Boot from defaults" selection in the main menu. To activate this, you will need to go through the arming sequence of moving the aileron stick all the way to the right, then all the way to the left, and then all the way to the right again. After that, you will need to restart the board for the defaults to come into effect.

Serial Terminal Commands

The following commands are accepted using the serial terminal (W.E.F. version 5.8 onwards):

DUMP
XDUMP
STATUS
SHOWLOGS
RESET
CLS
SAVE
READEEPROM
DEFAULTS
HELP
SET
QUERY

SET

This accepts the following arguments:-

RADIOMODE

CTRLCHAN
AILCHAN
AIL2CHAN
ELECHAN
RUDDCHAN
THROTTLECHAN
FLAPCHAN
PANCHAN
TILTCHAN
GAINCHAN
ELEVONMIX

ALTLADDER
SPDLADDER
DISPV1
DISPCALLSIGN
GLIDESCOPE
VSI
BAROTEMP
DISPTIME
DISPGDIST
VARIO

METRIC
ELEDIR
RUDDIR
COMPASSSTYLE
HORIZONSTYLE
LONLATSTYLE
OSDMODE
GPMODE

LOGPERIOD

VALIDHDOP
MINVALID

DISTCUTOFF
ALTRATIO
GLIDESLOPE

SAFEZONEX
SAFEZONEY

VMULT1
VMULT2
VMULT3
IMULT
MAHMULT
WHMULT
CURROFFSET
RSSIHIGH
RSSILOW

CALLSIGN

APROTGAIN
APROCGAIN
ROTLIMIT
ROCLIMIT
ROTSTEPGAIN
ROCSTEPGAIN
MAXHDGCHG
MAXALTCHG
SLOWLIMIT
HITDIST
STABGAIN

MAHWARNING
V1WARNING
V2WARNING
STALLSPEED

QUERY

This accepts all the arguments for SET, and the following:-

HOME
POSITION
TRUETEMP
VIDEOMODE
TIME
ETIME
STIME
HOMEDIST
ABSDIST
CURRENTWP
ROT
ROC
ROLL

PITCH
ALT
GALT
PALT
GROUNDSPEED
AIRSPEED

HAVEIMU
HAVEBARO
HAVEGPS
HAVECAS

SATS
PPM
V1
V2
V3
CURRENT
MAH
WH

Alternative command set

The following commands are supported:-

\$S <varname> - set variable
\$Q <varname> - query variable
\$E - enumerate settable variables
\$C - enumerate non-settable variables
\$W - save all settings to EEPROM
\$V - return version number
\$D - reset all values to defaults (permanent and immediate change)
\$X - reset OSD
\$M - reset flight logs (will erase everything immediately)
\$Z - clears the OSD screen (and redraws where possible)
\$JW <wpnum> - save waypoint data to waypoint number wpnum

There are two types of variables - those that can be set and queried, and those that be queried but cannot be changed.

Installing the DOSD+

For proper operation of the DOSD+, there are several guidelines that one should follow. This is in order to ensure minimal interference to the more sensitive electronics. In general:

- Keep the video transmitter well clear of the GPS module
- Keep the video transmitter well clear of the R/C receiver and it's antenna
- Keep the DOSD+ as far as possible from the GPS module
- Keep the camera away from the R/C receiver and it's antenna
- Keep the DragonOSD away from the R/C receiver if possible

Based on this:

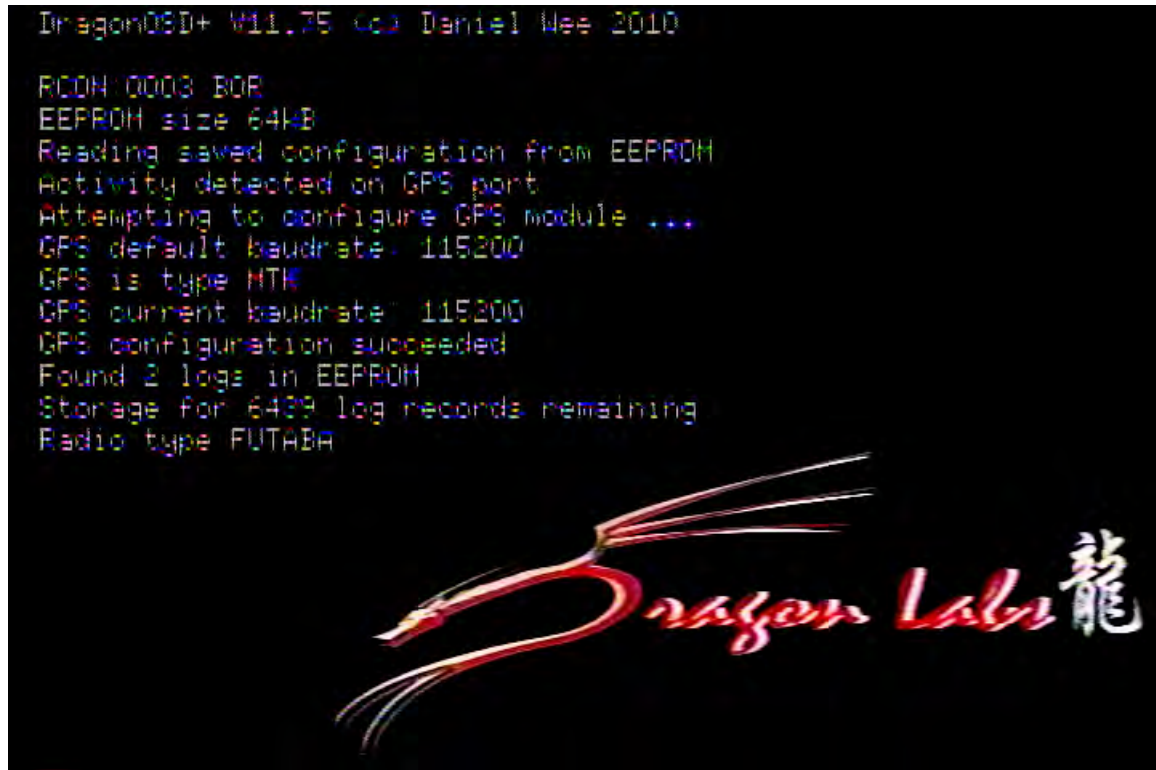
- The GPS module can be close to the R/C receiver and it's antenna
- The video transmitter can be close to the camera
- The camera can be close to the DOSD+
- The ESC can be close to the camera and DOSD+

One should also take care to:

- Use twisted cables where possible
- Use clip-on chokes for groups of cables
- Keep the servo and control cables as a group, away from the receiver
- Keep all cables as short as possible
- Keep ESC cables away from the video cables if not shielded

DOSD+ Startup

Upon powering up the DOSD+, it will attempt to detect any devices you have connected to it and display this information for a short time on the screen. It will also display the radio and GPS information as well.



Once this is done, the words "HOME not set" will appear in the middle of the screen. If you find that nothing is happening, you may want to check that the GPS cable is properly connected with the correct orientation, and then try to re-connect the power to the DOSD+ board.



Once the GPS starts sending data you should see the satellites acquisition screen.



The bars at the center of the screen indicate the signal strength of the satellites that the GPS module is seeing. Immediately below each bar is the unique number of that satellite. Just because several bars are present, does not necessarily mean that the data has been locked in and at the top of the screen it may still say "0 SATS".

Above the signal strength display are the words "HOME not set". Once the satellites are locked and the data is accurate enough – the home position countdown will begin. If the satellite data becomes inaccurate, the countdown will stop until it improves. During this time the home position will be set so the autopilot is able to return the aircraft, and the home indicator points the correct way. The countdown requires at least 3 satellite locks. *You should not move the aircraft during this time.*

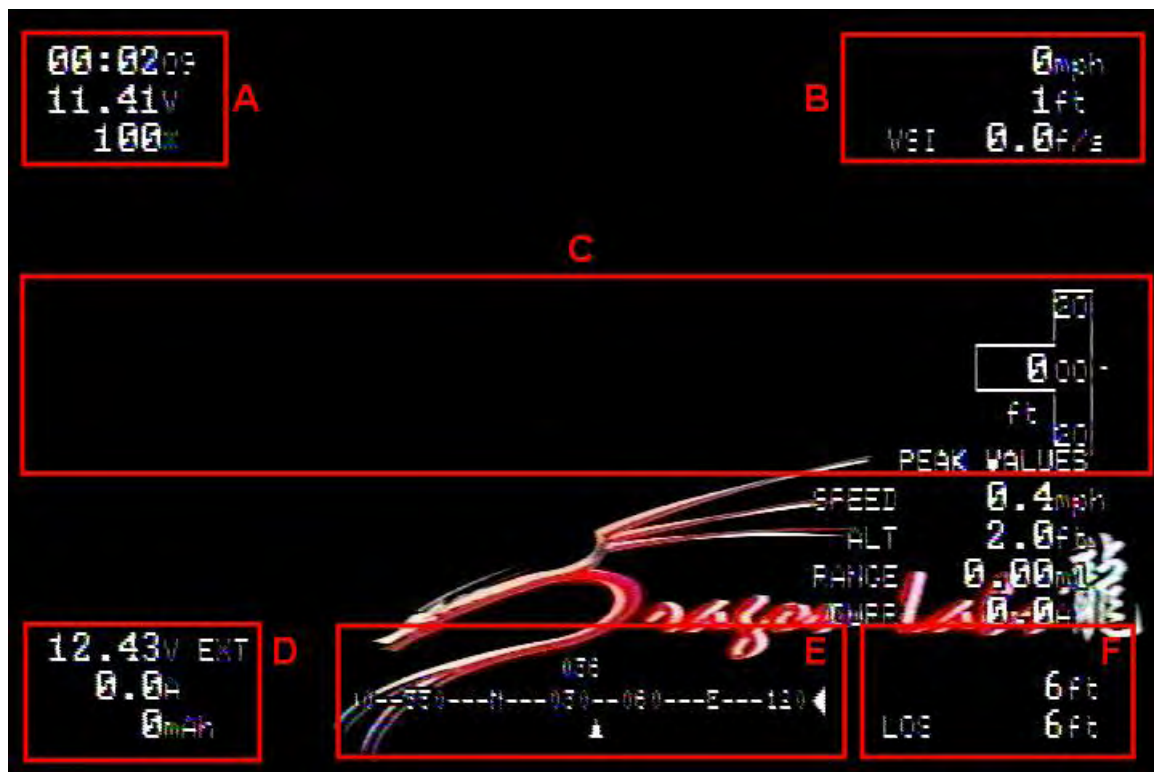
While waiting for satellite locks, it is not unusual to see the GPS coordinates and the altitude fluctuate. The altitude displayed at this time is exactly what the GPS reports, the altitude above sea level. Once HOME is set, this becomes the altitude relative to the HOME position/altitude. Once the home position has been recorded, the screen below will be shown. Depending on the altitude fluctuation (can be up to a few meters) the GPS



coordinates, number of satellites locked in, and the HDOP display may be automatically turned off. This is determined by the LOWALTRATIO menu setting. If the relative altitude divided by the ground distance from HOME is above LOWALTRATIO, the display will be turned off automatically. If the plane should lose too much altitude, the display will automatically come back up. This is so if the plane is about to crash, the GPS coordinates will be displayed and captured by any recording device that may be recording the flight, and which can then be used to help find the downed plane.

Another thing that you will notice is that on the right side of the screen, a group of values will be displayed under the heading "PEAK VALUES". This group will show up when HOME is set and the ground speed falls below about 0.5kph. It will display the highest speed, altitude (absolute above sea level), distance (line of sight) and current drawn (if the current sensor is connected), that were recorded. Once the plane starts moving, this display will disappear.

Display Elements



This section discusses the primary display elements. The way these elements are arranged is based on the screen mode you select. The first to be discussed is DOSD Mode 1. It has in it 6 major groups.

Group A

This group consists of three elements, the upper being the elapsed time. The count-up begins as soon as HOME is set and will continue until the device is powered down. The element below that is the battery voltage that is supplying the DOSD+ board. If the voltage should drop below the battery warning threshold, this element will begin to flash as a way of warning the user of impending loss of power. The last element in this group represents the Rx signal strength (RSSI).

Group B

This group also consists of 3 elements. The upper element, is the ground speed as supplied by the GPS module. The element below that is the aircraft's altitude. The last element in this group is the VSI (Vertical Speed Indicator) which shows the rate of decent or climb.

When the altitude to ground distance ratio falls below LOWALTRATIO as set in the configuration menu, this lower element will begin to flash as a way of warning the user. On the right of this display will be seen an up or down arrow indicating the vertical direction of the plane. This will usually become more frequently visible for large changes of altitude rather than for smaller changes.

Group C

This group also consists of up to 3 elements, depending on the options chosen and the hardware available. The left element (not shown) is the glideslope indicator. The center element (not

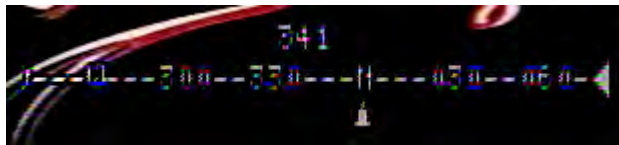
shown) is the IMU artificial horizon. The element on the right is the altitude ladder indicating the aircraft's current altitude.

Group D

This group will only be displayed if the current sensor is connected, and a secondary external battery is connected to the DOSD+. The first element is the voltage of the secondary battery. Typically this would be your flight pack powering the motor of your aircraft. There is an "EXT" suffix to indicate that it is the external battery voltage that is being measured. The second element is the amount of current being drawn, and below that is the mAh consumed thus far. There is also a warning threshold associated with the mAh reading. When the reading exceeds the MAHWARNING value (which can be changed in the configuration menu), it will start flashing to warn the pilot of impending loss of power.

Group E

Group E is where the compass will appear. The style of compass displayed varies depending on your preference. For example, when the compass is set to mode 1, it will appear this way:



Mode 1:

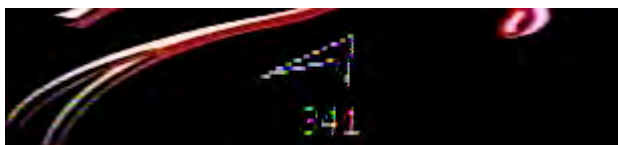
The first element is the number in the middle ("341" in the picture) which represents the GPS track. Note that the track is not necessarily the same as the heading in that it represents which way your plane is moving, and not which way it is pointing. As such, you may be pointed towards what you think is your launch point, but the track could be off due to side slip as a result of a crosswind, for example. Below that is the compass reticule which also indicates the GPS track and will scroll to show which way you are turning. This reticule represents the forward 180 degrees of the GPS track heading. Below the reticule, you can sometimes see an icon. This represents the heading for the launch point. It will only be visible if the launch point is somewhere in the front 180 degrees of the track heading.

If the launch point goes outside of that 180 degrees, then the arrow at the left (or right as it may be) end of the reticule will become outward pointing rather than inward pointing. That will indicate the direction of the launch point but that it is behind rather than in front. When the track heading is directly towards the launch point, the heading indicator will be highlighted to make it more obvious.

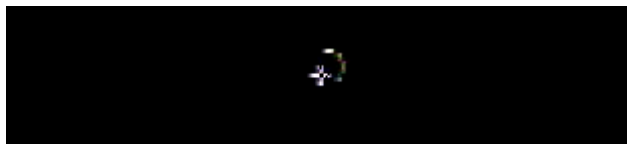
The other compass style modes (modes 2, 3, 4, 5 and 6) are also shown. You can see how these differ from mode 1.



Mode 2:



Mode 3:



Mode 4:



Mode 5:



Mode 6:

Group F

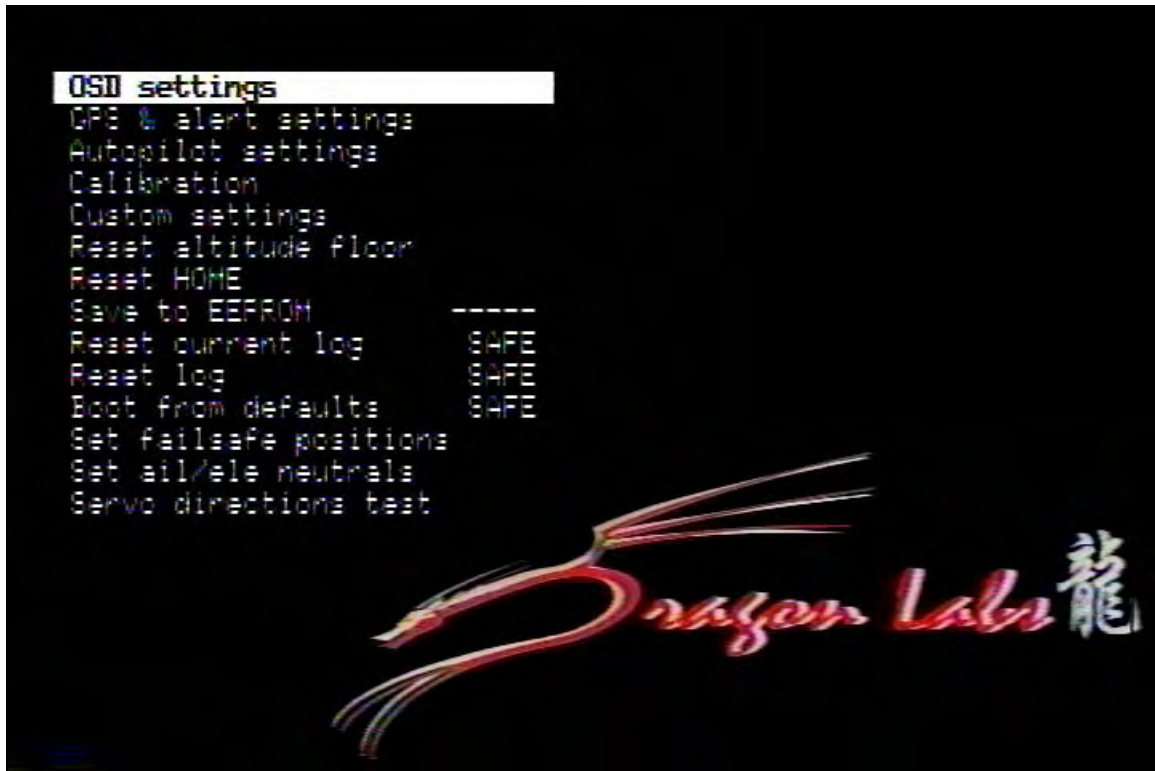
This group consists of two elements, the upper one being the ground distance while the lower one is the absolute or line-of-sight (LOS) distance.

DOSD Mode 2 is arranged slightly different:



Most of the elements are grouped at the top of the screen. For consistency, the groups described above have been shown here as well (A though F). This mode also has the addition of a “radar” icon of sorts appearing at the center to of this photo. It displays the direction to home.

Configuration Menu

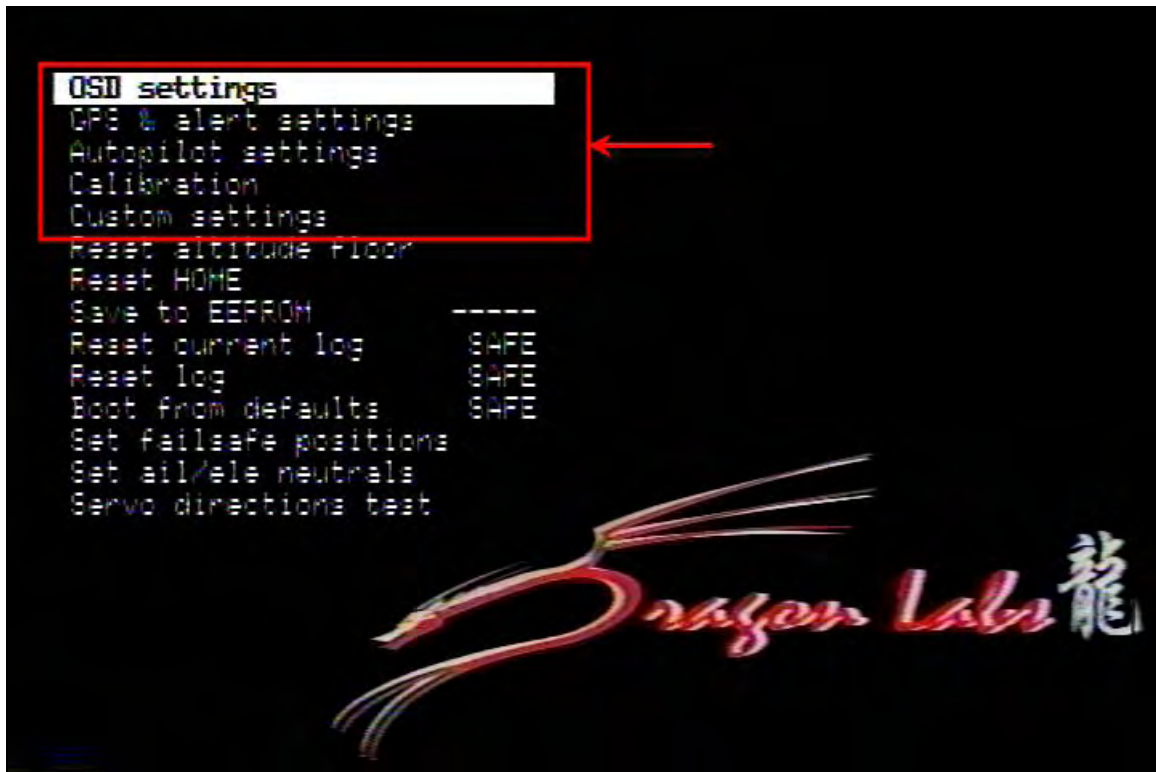


The configuration menu allows various parameters and options to be set for the DragonOSD. In order to use the configuration menu, the Aileron and Elevator inputs must be connected to the appropriate channel on the receiver. Additionally the control channel must also be connected to a spare channel. Setting the control channel's PWM output as follows will produce the following results:

MENU - above 1.9ms (>90%)
NORMAL - between 1.6ms to 1.9ms (60% to 90%)
AUTOPILOT - between 1.2ms to 1.6ms (20% to 60%)
NODISPLAY - less than 1.2ms (<20%)

Once the configuration menu has been called up, you can move up and down through the items using the elevator stick (up or down). To change a setting, move the aileron positions (left or right). If a stick is held in a position other than neutral, it will repeat the incrementing/decrementing action until it reaches the maximum/minimum limit. If the direction of the movement does not correspond to the stick direction, you may have to change the ELE menu direction and/or AIL menu direction setting until it behaves correctly. Doing this will usually also ensure that the servo directions are correctly configured. All changes take place immediately but are not saved unless explicitly commanded by the user. Note also that when you re-enter the configuration menu, you will still be at the same setting as when you exited the configuration menu last.

This will assist in making quick adjustments without having to move the selection to the desired item all over again. *Note also that any changes you make are immediately active the moment you change it, but they are not saved unless you explicitly select the SAVE CONFIG option. Doing so will make the changes survive power cycling.*



The DOSD+ has six separate menus. They are:

- 1) The Main menu (shown above)
- 2) OSD Settings menu
- 3) GPS & alert settings menu
- 4) Autopilot settings menu
- 5) Calibration menu
- 6) Custom settings menu

Each one will be discussed, starting with the Main Menu

Main Menu



OSD Settings: Accesses the OSD settings menu.

GPS & Alert Settings: Accesses the GPS & Alert Settings menu.

Autopilot Settings: Accesses the autopilot settings menu.

Calibration: Accesses the calibration menu.

Custom Settings: Accesses the custom settings menu.

Reset altitude floor: Uses the planes current altitude as the zero feet (meter) point.

Reset Home: Resets the home position to the current aircraft location.

Save to EEPROM: Use this function to permanently save any changes you make to the DOSD+. If any changes have been made that has not been saved, you will see 5 dashes "-----" displayed. Once you change this to "DONE" the data will be saved.

Reset Current Log: Clears the current log of all values.

Reset Log: Clears the specified log number of values.

Boot From Defaults: Resets all DOSD+ values to their default state.

Set failsafe position: Sets the failsafe operating positions. Use the following procedure for setting fail safes:

1. Trim your plane so that it is flying straight and level with your hands off, ideally in a low wind condition.
2. With the aileron and elevator in neutral, activate the menu.
3. Go down to the last option on the root menu and move the stick firmly to the right. You will hear a confirmation beep if you have audio connected. Otherwise it's silent.

Failsafe is now set. If the PPM stream should become corrupt or disappear for any reason, after being present at boot-up, these failsafe settings will be adopted. If no fail safes have been saved, it will adopt a neutral position (middle) for all the channels except for the throttle channel which defaults to about 3% (as a safety precaution).

When failsafe is active, the autopilot will be activated if GPS is attached and home is already set. If home is not set, it will just fly normally - the amount of stabilization will depend on whether the gain channel is set and the position of that channel when failsafe is set. So if you have the gain channel set to channel 8, for example, and the knob set to 50% when setting the failsafe, those will be the values used to control the stabilizer when failsafe becomes active.

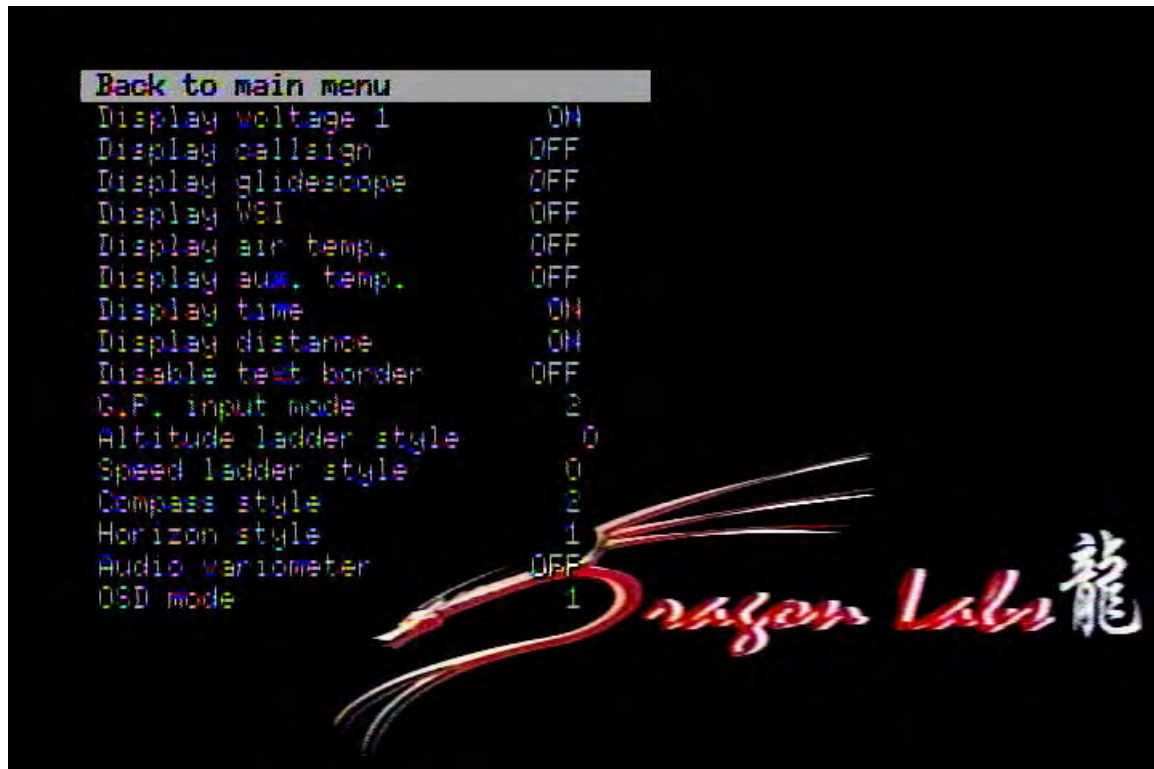
Note that if you are using an advanced receiver which has its own failsafe built into the PPM stream (such as the Dragon Link), the DOSD+ will never know that the receiver has gone into failsafe and will not indicate as such. In that case, you will need to set the failsafe on the radio itself, which takes precedence over the DOSD+'s failsafe. It is still a good idea to set the failsafe though, just in case.

For those of you who do not use the PPM input functions, the DOSD+ will not go into autopilot in the absence of a PPM signal unless you introduce a valid PPM signal at some point during or after booting up. If those are not present, the DOSD+ will assume that the PPM input is not being used and will not activate the failsafe function.

Set Ail/ele Neutrals: Sets the current aileron and elevator positions as the "neutral" for the DOSD+.

Servo Directions Test: Test used to determine if your servo direction is set correctly from the DOSD+ Menu. To run the test, hold the aileron stick to the right for several seconds until your servos move, let go of the stick and see if the ailerons and elevator are set correctly, if not reverse them in the DOSD+ custom settings menu. This will determine if your autopilot will operate the ailerons and elevators correctly.

OSD Settings Menu



Back to Main Menu: Returns you to the previous main system menu.

Display Voltage 1: (default ON) Determines weather or not you wish to display the voltage input to the DOSD+. This value is displayed in the upper left of the screen when enabled.

Display Callsign: (default OFF) Determines weather or not you wish to display the callsign you have assigned to the DOSD. When enabled, the callsign will be displayed in the upper left of the screen for one-minute at the top of every ten-minute interval if you have enabled the callsign display option.

Display Glidescope: (default OFF) Turns Glidscope display on or off. The Glidescope when turned on appears on the left side of the screen. The glidescope shows the optimal approach slope for landing. It tells you if you are too high or too low.

Display VSI: (default OFF) Turns the VSI (Vertical Speed Indicator) on or off. It is used to inform you of the instantaneous rate of descent or climb. When turned on it appears at the upper right side of the screen.

Display Air Temp: (default OFF) Turns the air temperature display on or off. The air temp readings will only appears if you have the display options enabled AND if you have the actual sensor attached.

Display Aux Temp: (default OFF) Turns the aux air temp on or off. The aux temp only appears if you have the display options enabled AND if you have the actual sensor attached.

Display Time: (default ON) Turns elapsed time display on or off. Indicates the elapsed time that the DOSD+ has been powered. Appears at the upper left of the screen when turn on.

Display Distance: (default ON) Turns distance from launch point on or off. Appears at the lower right side of the screen when enabled. The distance value at the bottom with the LOS is the line of sight distance, hence LOS. The other one displays ground distance, which does not take altitude into account.

Disable Text Border: (default OFF) Turns boarder on or off around the text characters of the onscreen menues and displayed elements. This can help you in situations where you need to make your displayed characters more bold.

G.P. Input Mode: (default 0) GP (General Purpose) input can be used to read various analog sensors, such as an analog temperature sensor. Currently it has the following functions:

Mode 0 – display no input.

Mode 1 – display temperature (if you have an analog temperature sensor connected)

Mode 2 - display RSSI in percentages

Mode 3 - display RSSI in dBm.

When enabled, the value will be displayed at the upper left of the OSD display. Generally you would use mode 2. Mode 3 only really works if you happen to have a DragonLink receiver attached.

Altitude Ladder Style: (default 0) Turns the graphical altitude ladder on or off on the right side of the display and sets the style displayed. A value of 0 disables the ladder. When enabled the current altitude of the aircraft is displayed. Styles will appear as follows:

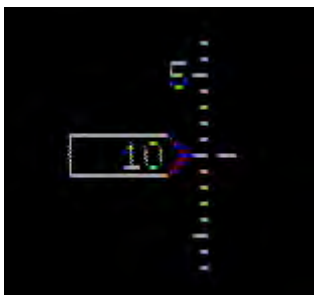


Style 1:



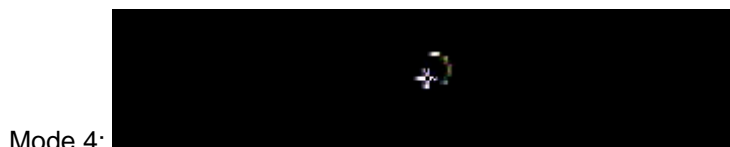
Style 2:

Speed Ladder Style: (default 0) Turns the graphical speed ladder on if desired when set to 1. The speed ladder will appear at the left side of the screen and appear this way:



Compass Style: (default 0) Sets the compass style displayed at the bottom center of the screen if a GPS or other navigation aid is present. Note mode 4 appears in the center of the screen. Currently it has the following settings:

Mode 0 – display no compass



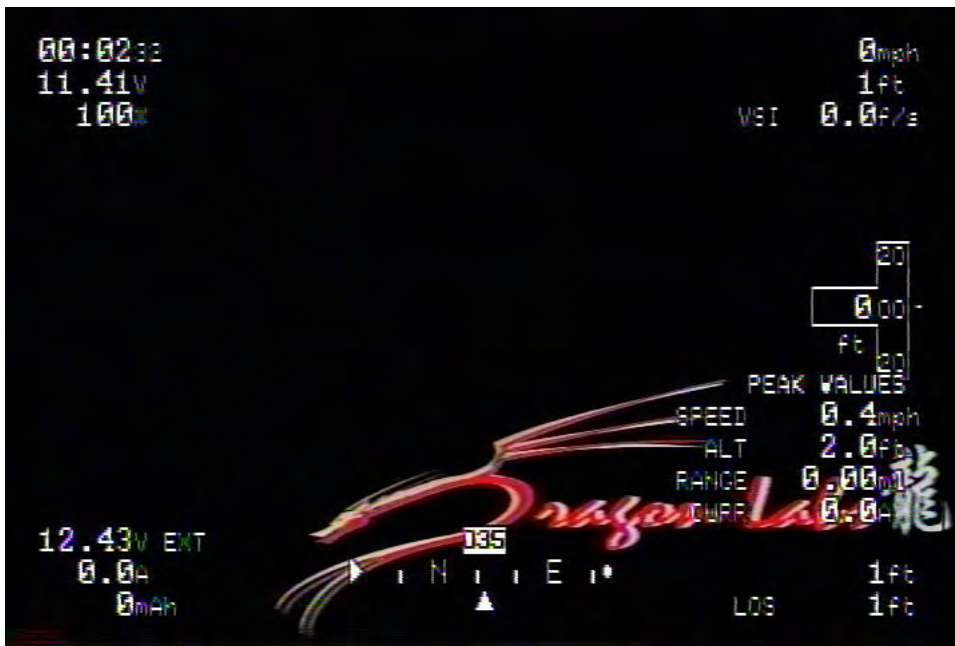
Horizon Style: (default 0) Turns the artificial horizon on or off if the IMU is present. When enabled the horizon will appear as a line at the center of the display indicating the aircrafts orientation from “level” (tilt left, right, pitched up or down).

Audio Variometer: (default OFF) When enabled and if you have a barometer attached to the DOSD+ you will hear rising and falling audio tones which represent thermals you can fly through to assist your aircraft in flight, just as glider pilots do. You will need the audio line connected to your Vtx as well.

OSD Mode: (default 1) The OSD Mode value controls the display layout. Modes are currently defined as follows:

Mode 0: No display

Mode 1: Standard Control layout



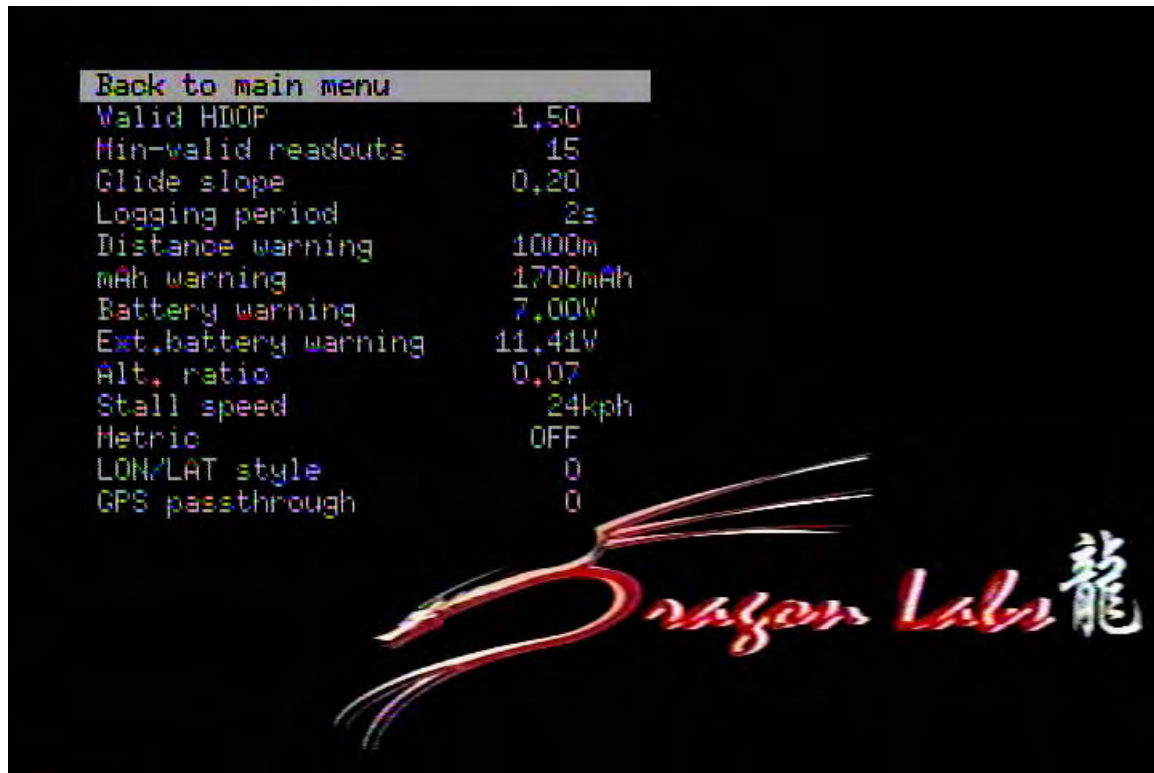
Mode 2: Alternate Control layout



Mode 3: Diagnostic Screen



GPS and Alert Settings Menu



Back to Main Menu: Returns you to the previous main system menu.

Valid HDOP: (default 1.50) The HDOP (horizontal dilution of precision) is the maximum value acceptable before the HOME SET countdown begins. The lower this value is, the tighter the GPS reception criteria is (demands greater accuracy). Setting this too low will lead to difficulty in setting the home location. Setting it too high leads to inaccurate home locations.

Min-valid readouts: (default 15) The number of good HDOP readings required before the GPS reading is considered valid for home location. The greater the value, the more stable the GPS reading must be.

Glide slope: (default 0.20): The glideslope setting works in conjunction with the glidescope instrument. This determines the actual slope angle which you want the glidescope to indicate. So 0.2 means that the ratio of altitude to distance is 1:5 and this is used for the glidescope. It will indicate if you are above or below this glideslope setting.

Logging Period: (default 0) Period of time (in seconds) OSD will use to log positional information. A value of 5, for example, means that a position will be saved every 5-seconds, if the speed is above about 0.5km/h. There are a total of nearly 6000 logging points so at 5-seconds, a clean log will give you over 8-hours worth of logging. Logging will stop once the plane's ground speed gets below 0.5 km/h and resume as you go over it.

When you cycle the power, the last log will be closed and a new one started. In this way, you can have multiple logs in memory until the memory is all used up, at which point logging will stop. You can either reset the current log (since power was applied) or delete all logs for a clean slate. Before you do that, though, you can list all the logs and download the ones of interest to you in the form of a KML file which Google Earth can open.

Issuing the command SHOWLOGS from the serial terminal will list all the available logs, if any. You can then use DUMP to dump the log in question using command:
DUMP 4

where 4 is the log number listed using SHOWLOGS.

The output should be cut and pasted into a separate text file with a KML extension.

Distance Warning: (default 1500m) Determines the Line Of Sight distance beyond which the LOS distance indicator will begin to flash.

Mah Warning: (default 1700mah) This is the mAh threshold above which the mAh display will begin to flash. mAh is derived from the current sensor input. mA is the unit for current, and h is for hour so mAh is the product of current (from the current sensor) and time.

Battery Warning: (default 7.00 Volts) This is the voltage threshold below which the Battery voltage display will begin to flash.

External Battery Warning: (default 7.00 Volts) This is the voltage threshold below which the external battery voltage display will begin to flash.

Alt. Ratio (default 0.07) This is the ratio of the altitude to the ground distance below which the altitude indicator will begin to flash and the GPS coordinates be displayed automatically.

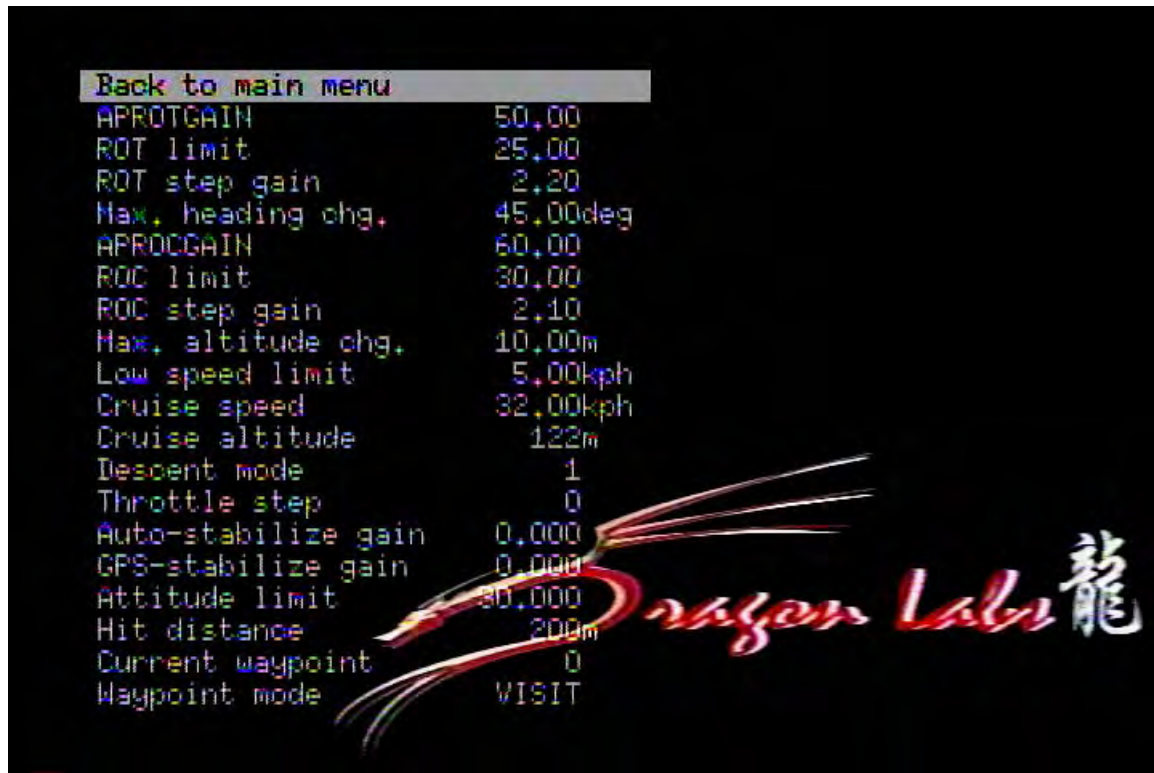
Stall Speed (default 24kph) This is the airspeed threshold below which the airspeed indicator will begin to flash.

Metric (default ON) Sets unit of measure value either to Metric (if enabled Kph) or English (mph) if disabled.

LON/LAT Style: (default 0): Changes the display styles available for the Longitude and Latitude values.

GPS Passthrough (default 0) GPS passthrough, when set to 1 or 2, will pass the NMEA strings from the GPS to the serial port. This will allow you to send the data to another device that needs it - for example Ardupilot or similar. When set to 1, it will pass through RMC and GGA strings. When in mode 2, only the GGA string is passed through. If set in simulation mode, it will also send out the simulated GPS position through the serial port, allowing you to work in conjunction with GooPs to feed the data to Google Earth, for example.

Autopilot Settings Menu



Back to Main Menu: Returns you to the previous main system menu.

APROTGAIN: (default 68.00) The autopilot gain that determines how sensitive the loop is to heading errors. Change this value gradually and with care. It is preferably to adjust ROT step gain first before adjusting this. Changes may be difficult to perceive.

ROT limit (default 30.00) The maximum permissible rate of turn that the autopilot will use while correcting for heading.

ROT step gain: (default 2.00) The gain used by the heading correction loop for driving the aileron/rudder servo. By setting this to a very low value such as 1, you effectively disable the heading control component of the autopilot. Too great a value here can lead to excessive aileron/rudder movement and severe oscillation. Change this value gradually while observing behavior.

Max. heading chg. (default 45.00 deg) Determines how aggressively the autopilot will make its' heading corrections. Change gradually and with care. Too great a value can lead to the plane rolling over!

APPROCGAIN (default 50.00)) The autopilot gain that determines how sensitive the loop is to altitude errors. Change this value gradually and with care. It is preferably to adjust ROC step gain first before adjusting this. Changes may be difficult to perceive.

ROC limit (default 10.00) The maximum permissible rate of climb that the autopilot will use in trying to achieve the desired cruise altitude. This value should be adjusted with care.

ROC step gain: (default 2.00) The gain used by the altitude setting loop for driving the elevator servo. By setting this to a very low value such as 1, you effectively disable the altitude control component of the autopilot. Too great a value here can lead to excessive elevator movement and severe porpoising. Change this value gradually while observing behavior.

Max. altitude chg.: (default 10.00m) Determines how aggressively the autopilot will make its' altitude corrections. Change gradually and with care.

Low speed limit: (default 5.00kph) The speed below which the autopilot will not attempt to climb to the cruise altitude.

Cruise Speed: (default 33.00kph) Sets the desired cruising speed the autopilot will use.

Cruise Altitude: (default 100m) Determines the desired cruising altitude for the autopilot. You may or may not always achieve this depending on how much energy the plane has. All being well, the autopilot will reach this altitude and try to stay as close to it as possible while flying back to the launch point.

Decent mode: (default 1) Determines decent mode autopilot will follow when home is reached. Modes are as follows:

Mode 0: No descent

Mode 1: If within "hit distance" from home, descend slowly to cruise altitude. Otherwise do not actively descend.

Mode 2: Slow descent to cruise altitude

Mode 3: Active descent to cruise altitude while maintaining cruise speed or higher

mode 4: Active descent

Mode 1 is like Mode 3 in that they both attempt to maintain cruise speed.

Throttle step (default 0) Throttle step is the minimum step size used when adjusting throttle when throttle management is used. A larger value means the autopilot is going to make throttle changes faster, but with coarser steps. A smaller value will allow finer adjustments but also result in slower response.

Auto-stabilize gain (default 0.000) Auto stabilization gain applies only to IMU stabilization. The effects of IMU and GPS stabilization are cumulative so you will have to play with the gains to determine how much of each you want. Has no function if you do not have an IMU.

GPS-stabilize gain (default 0.000) This setting helps stabilize your plane when the autopilot is off. GPS stabilization is disabled during autopilot since the autopilot should know what to do based on GPS input already. As such, you can only test this when not in autopilot mode. When active (i.e. GPS stabilizer gain is more than 0.0) it is always functioning, even when you're flying, but only when you are above about 2km/h ground speed.

Normally to test the GPS stabilizer, just increase the gain to something like 2.0. This will attempt to maintain the heading roughly (not in a strict way). So if you let go of the stick, you should see the plane eventually level. Having a 10Hz GPS would make this work better than a lower rate one since you will get more corrections.

Attitude limit: (default 30.000) Attitude limit is the angle of roll and pitch beyond which the IMU stabilizer (if IMU is present) will override autopilot input. So for example the autopilot commands a turn that results in a 50-degree bank, but attitude limit is set to 30-degrees. The IMU stabilizer will limit the bank to 30-degrees. Same applies for pitch.

Hit distance: (default 200m) Distance from target waypoint value autopilot uses to determine if a specific waypoint has been reached.

Current Waypoint: (default 0) When non-zero the said waypoint will be loaded into memory and autopilot activation will send the plane towards that waypoint. Upon reaching within Hit distance (meters) of this way point the autopilot will act according to what Waypoint defines.

Waypoint Mode: (default VISIT) Global waypoint setting which determines how autopilot will act. Modes are as follows:

CIRCLE – circle about waypoint

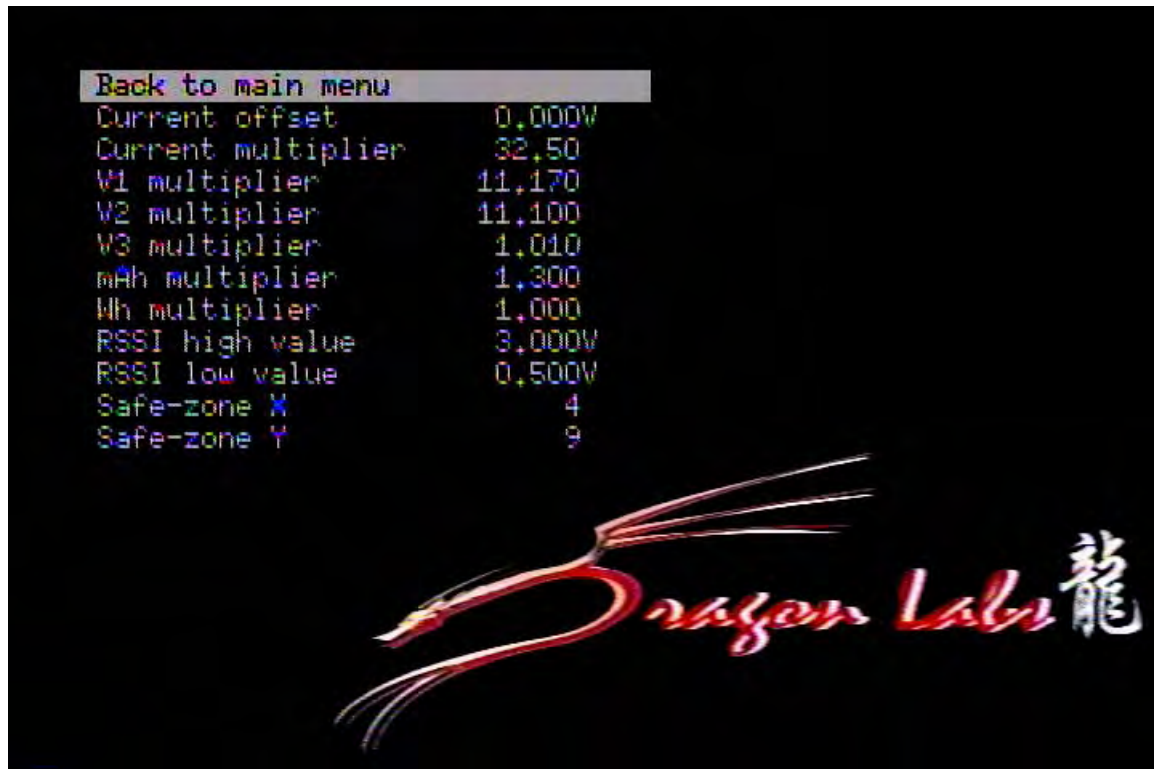
VISIT - reach waypoint then come right home

LOITER - do not change anything, loiter about waypoint

NEXT - load next active waypoint and head there

Notes: Waypoint mode is the global waypoint setting which is used for handling all waypoints except when it is set to "CUSTOM". In case of custom, the code will use the waypoint mode setting for the individual waypoints instead.

Calibration Menu



Back to Main Menu: Returns you to the previous main system menu.

Current offset: (default 0.000V) This is the coefficient used to compute the current readout. You can increase or decrease this as needed to get the amperage reading as close as possible under nominal current draw conditions.

Current multiplier: (default 32.50)

V1 multiplier: (default 11.17) Sets the V1 (DOSD+ input voltage) multiplier

V2 multiplier: (default 11.17) Sets the V2 (current sensor voltage) multiplier

V3 multiplier: (default 11.17) Sets the voltage multiplier for the GP input (for analog temperature or RSSI).

mAh multiplier: (default 1.354) Sets the mAH multiplier for the mah display.

Wh multiplier: (default 1.000) Not currently used.

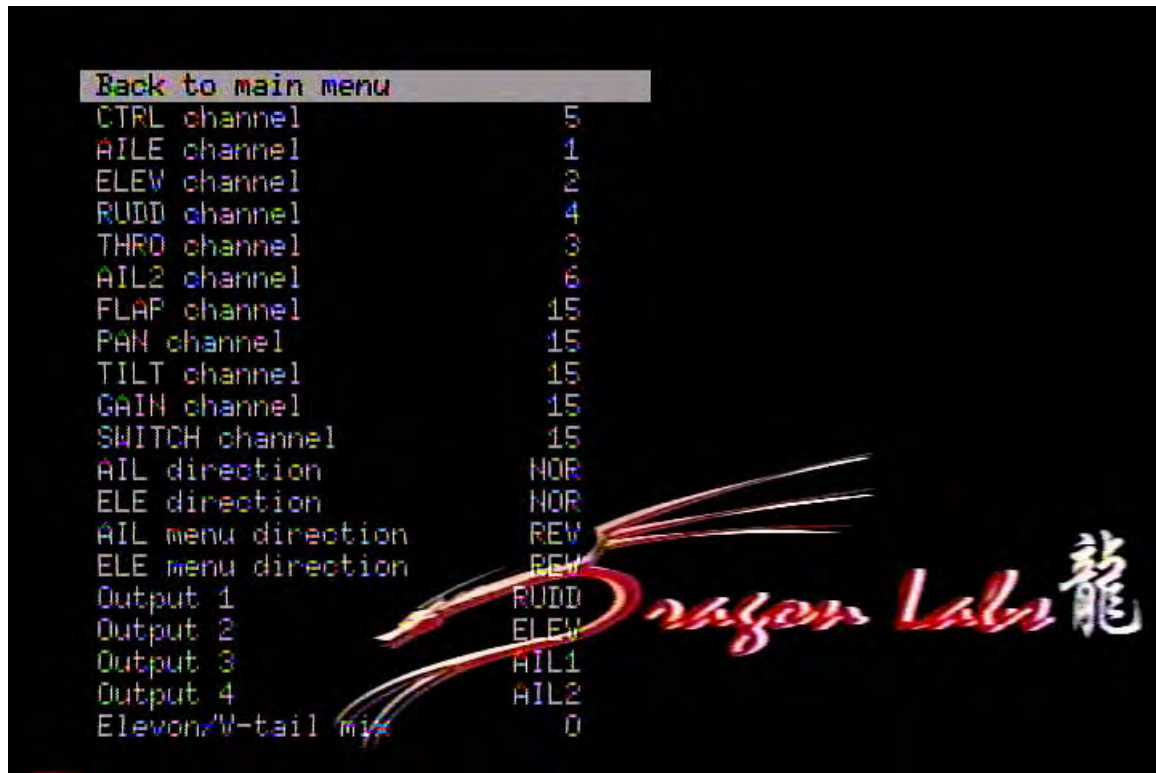
RSSI high value: (default 2.230V) Sets the voltage high value when your signal is at 100% strength.

RSSI low value: (default 0.500V) Sets the voltage low value when your signal is at its worst.

Safe-zone X: (default 0) This setting allows you to move the display components away from the left edge thus forming a "safe zone" where the displayed text will not be obscured or cut off.

Safe-zone Y: (default 0) This setting allows you to move the display components away from the top edge thus forming a “safe zone” where the displayed text will not be obscured or cut off.

Custom Settings Menu



Back to Main Menu: Returns you to the previous main system menu.

CTRL channel: (default 5) Defines the control channel used by the OSD which controls the following settings:

1. Normal flight mode
2. Menu mode
3. Auto-pilot mode
4. No OSD display mode

AILE channel: (default 1) Defines the channel used to make selections and change values within the DOSD+ menu system.

ELEV channel: (default 2) Defines the channel used to move up and down within the DOSD+ menu system.

RUDD channel: (default 4) Defines what channel the rudder is connected to on your Rx.

THRO channel: (default 3) Defines what channel the throttle is connected to on your Rx.

AIL2 channel: (default 6) Defines what channel the second aileron is connected to on your Rx.

FLAP channel: (default 15) Defines what channel the flaps are connected to on your Rx.

PAN channel: (default 15) Defines what channel the camera pan servo is connected to on your Rx.

TILT channel: (default 15) Defines what channel the camera tilt servo is connected to on your Rx.

GAIN channel: (default 15) Defines what channel the gain function is assigned to on your Rx. The gain channel is the “global control” which determines the overall sensitivity for flight stabilization. This channel assignment is critical for “dialing in” your best stabilization. It is recommended this channel be assigned to a control on your Tx, such as a knob, so you can properly set the amount of stabilization which impacts your aircraft. Assigning this channel to a switch instead will turn stabilization on or off entirely.

SWITCH channel: (default 15) Defines what channel the switch function is assigned to on your Rx.

AIL direction: (default NOR) Defines the direction the aileron will move. Valid values are NOR (normal) and REV (reverse)

ELE direction: (default NOR) Defines the direction the elevator will move. Valid values are NOR (normal) and REV (reverse)

AIL menu direction: (default NOR) Defines the direction the aileron stick controls movement within the menu. Valid values are NOR (normal) and REV (reverse)

ELE menu direction: (default NOR) Defines the direction the elevator stick controls movement within the menu. Valid values are NOR (normal) and REV (reverse)

Output 1: (default RUD) Defines what function the PWM1 connection on the DOSD+ is being used for. Valid values are:

THROTTLE
AIL1
AIL2
RUDDER
ELEVATOR
ELEVON1
ELEVON2

Output 2: (default ELEV) Defines what function the PWM2 connection on the DOSD+ is being used for. Valid values are:

THROTTLE
AIL1
AIL2
RUDDER
ELEVATOR
ELEVON1
ELEVON2

Output 3: (default AIL1) Defines what function the PWM3 connection on the DOSD+ is being used for. Valid values are:

THROTTLE
AIL1
AIL2
RUDDER
ELEVATOR

ELEVON1
ELEVON2

Output 4: (default AIL2) Defines what function the PWM4 connection on the DOSD+ is being used for. Valid values are:

THROTTLE
AIL1
AIL2
RUDDER
ELEVATOR
ELEVON1
ELEVON2

Eleven/V-tail mix (default 0): Eleven mix determines how the aileron and elevator channels are mixed to produce the EVN1 and EVN2, as well as VTL1 and VTL2 outputs. These are the eleven and vtail mixes respectively.

As an example of how complicated this is to explain, when:-

eleven mix = 0, EVN1 = aileron + elevator, EVN2 = aileron - elevator
eleven mix = 1, EVN1 = aileron - elevator, EVN2 = -aileron + elevator
eleven mix = 2, EVN1 = -aileron + elevator, EVN2 = aileron + elevator
eleven mix = 3, EVN1 = -aileron + elevator, EVN2 = -aileron - elevator

The same mixing applies to VTL as well except that the rudder is used instead of ailerons.

Using the on-board position logging function

The DOSD+ has a position logging function. You will need to assign a value to the Logging period option which is in seconds. A value of 5, for example, means that a position will be saved every 5-seconds, if the speed is above about 0.5km/h. There is a total of nearly 6000 logging points so at 5-seconds, a clean log will give you over 8-hours worth of logging. Logging will stop once the plane's ground speed gets below 0.5 km/h and resume as you go over it.

When you cycle the power, the last log will be closed and a new one started. In this way, you can have multiple logs in memory until the memory is all used up, at which point logging will stop. You can either reset the current log (since power was applied) or delete all logs for a clean slate. Before you do that, though, you can list all the logs and download the ones of interest to you in the form of a KML file which Google Earth can open.

Issuing the command SHOWLOGS from the serial terminal should list all the available logs, if any. You can then use DUMP to dump the log in question using:

DUMP 4

where 4 is the log number listed using **SHOWLOGS**.

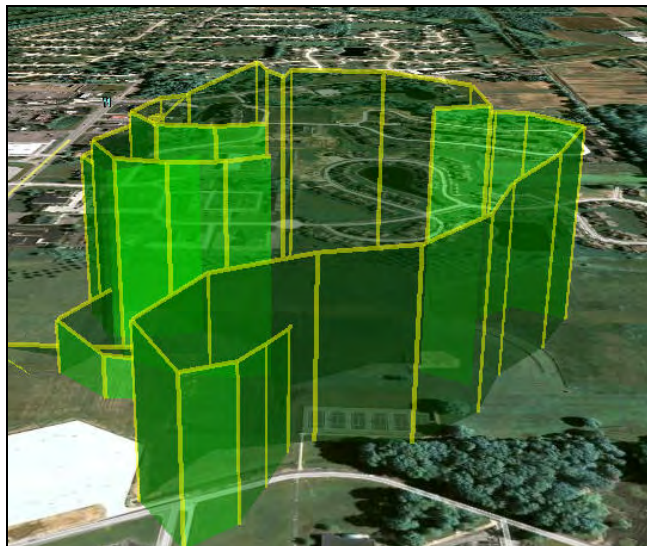
The output that appears should be cut and pasted by you into a separate text file with a KML extension. The easiest way to do this is to:

- 1) Open the serial terminal.
- 2) Issue the DUMP command to retrieve the desired log (Example DUMP 4), and copy everything you see to the clipboard.
- 3) Open up Microsoft NotePad and paste the clipboard contents into it. Then save the file with extension KML (Example flight 7.kml).

If done correctly, and you have Google Earth installed the file icon will appear this way:



Double click on the file, and Google Earth will open up and you will be able to see your flight!



Setting the on-board failsafe

The procedure for setting the failsafe is as follows:-

1. Trim your plane so that it is flying straight and level with your hands off, ideally in a low wind condition.
2. With the aileron and elevator in neutral, activate the menu.
3. Go down to the last option on the root menu and move the stick firmly to the right. You will hear a confirmation beep if you have audio connected. Otherwise it's silent.

Failsafe is now set. If the PPM stream should become corrupt or disappear for any reason, after being present at boot-up, these failsafe settings will be adopted. If no failsafes have been saved, it will adopt a neutral position (middle) for all the channels except for the throttle channel which defaults to about 3% (as a safety precaution).

When failsafe is active, the autopilot will be activated if GPS is attached and home is already set. If home is not set, it will just fly normally - the amount of stabilization will depend on whether the gain channel is set and the position of that channel when failsafe is set. So if you have the gain channel set to channel 8, for example, and the knob set to 50% when setting the failsafe, those will be the values used to control the stabilizer when failsafe becomes active.

Note that if you are using an advanced receiver which has its own failsafe built into the PPM stream (such as the Dragon Link), the DOSD+ will never know that the receiver has gone into failsafe and will not indicate as such. In that case, you will need to set the failsafe on the radio itself, which takes precedence over the DOSD+'s failsafe. It is still a good idea to set the failsafe though, just in case.

For those of you who do not use the PPM input functions, the DOSD+ will not go into autopilot in the absence of a PPM signal unless you introduce a valid PPM signal at some point during or after booting up. If those are not present, the DOSD+ will assume that the PPM input is not being used and will not activate the failsafe function.

DOSD+ Autopilot configuration

Note: Unless you have GPS home position set - the autopilot will not function under any circumstances.

The keys to successful setting up of the autopilot function are:-

1. Test a bit at a time.
2. Test one part at a time (i.e. turning to home, then climbing to cruise altitude).
3. Being familiar with the menu settings and able to access them quickly.
4. Do as much testing on the ground as possible.

To do this - the first thing you'd do would be to:-

1. Ensure that you have the Elev. direction and Rudd. direction set correctly in the menu options such that you can navigate the menu in the correct orientation (i.e. moving the elevator stick down brings the menu selection down, moving the aileron stick right increases the parameter value.)
2. Test the autopilot without the stabilizer or IMU working first.
3. Trim out your plane so that it flies more or less straight and level before starting your test.
4. Configure your radio so you can activate or de-activate autopilot mode with a switch.
5. Ensure that you are getting good satellite readings and a solid position fix (HDOP less than 1.3 would be good)
6. We will adjust the turns first so set ROCSTEPGAIN to 0.0 for this test. Also set the Auto and GPS stabilizer gain to 0.0. Remember that changes need to be saved if you want them to persist through a power-cycle.
7. In flight, with the default autopilot settings - fly in a direction away from home. Once you are flying more or less straight (and not in too windy conditions), activate the autopilot. Verify that the autopilot label is printed on screen. Now, without any stick inputs, observe if the plane is attempting to turn in the correct direction.
8. If the autopilot is turning in the right direction, de-activate the autopilot and fly in an opposite direction. Repeat #7 above and see if the autopilot will turn correctly in the opposite direction. Remember that when the autopilot is manually engaged (i.e. with the switch), you still retain control over the plane so if you get concerned about the way the plane is going, you can override it with your stick inputs.
9. If the plane is turning the wrong way in #7 or #8, re-do the test to verify it. If it is confirmed, you will need to reverse the Rudd. direction in the menu. After that, try again. Note that reversing this setting will also reverse the way the aileron stick behaves in the menu navigation.
10. Once you have the plane turning in the correct direction, you can now assess if the plane is turning too slow, or too hard. If it is turning too hard, it may start to oscillate hard - i.e. turning too hard and then correcting too hard. If this is the case, you may want to reduce APROTGAIN and possibly reduce ROTSTEPGAIN. Normally if it is close to where I want it, I tend to adjust ROTSTEPGAIN. It is preferable to keep ROTSTEPGAIN no more than about 8 but even that could be too much depending on the particular servo and setup. If the plane is turning too weakly, you can increase APROTGAIN and/or ROTSTEPGAIN. Increasing ROTSTEPGAIN too much will result in exaggerated snake-like oscillation movement.

Once you are happy with your settings - turn on the autopilot and see if it brings the plane back properly from about 300m to 500m out (making sure that your radio link can go that far in the first place).

This concludes the adjustment for the turning autopilot. The climbing autopilot will attempt to bring the plane to the cruise altitude which you may want to change in the menu. Other than that, the procedure is pretty much the same except that you are dealing with the Elev. direction, the APROCGAIN and the ROCSTEPGAIN. Instead of turning left or right, you want to observe if the plane is attempting to climb or dive depending on whether you are above or below the cruise altitude. You will need to increase ROCSTEPGAIN to around 3.0 to start with and go from there.

There are other settings for the autopilot but you want to start with these. If the turn direction is correct, the stabilizer should also be working correctly. For that you will need to set the maximum allowable gain, and, optionally, a gain control channel.

Note that you could, technically, perform all these tests on the ground while walking around the field with the plane and observing the movement of the control surfaces. If you choose to do so, you may want to use a very large value for ROTSTEPGAIN so that the servo movements are move easily visible, for when trying to ascertain if the autopilot is turning the right way. Also note that you need to be moving constantly or the GPS will not put out a valid heading which the autopilot needs.

DOSD+ calibration procedures

RSSI Calibration

For those wishing to calibrate their RSSI, the procedure is as follows:-

1. Make sure that RSSI is connected and you have some functionality
2. Turn the transmitter off
3. Adjust the RSSILOW parameter until the RSSI reading just reaches 0%
4. Turn the transmitter on (near the receiver)
5. Adjust the RSSIHIGH parameter until the RSSI reading just reaches 100%
6. Save the settings to EEPROM when you are satisfied with the results

Some notes about this:-

1. There will be some fluctuation of the RSSI reading so it's not unusual to see the readings jump between 0% to 7% on the low side for example. The high side values tend to be more stable but then again, do not be surprised at some fluctuation.

2. The RSSI reading is actually a voltage reading so that you can actually measure the RSSI line with a DMM to get an idea of what the voltages are. This way you can enter the voltages directly rather than take a trial and error approach to find the threshold readings. Having said that, it is probably still necessary to tweak it a bit to get the readings exactly right because the calibration of the board voltage readings may not be as accurate as that of your meter (or vice versa).

3. When using the menu to make this adjustment, it is necessary to exit the menu to look at the resultant readings. When using the serial terminal to make this adjustment, you should be able to see the change immediately. The serial commands take the form of:-

SET RSSILOW 1.0
SET RSSIHIGH 2.8
SAVE

4. All of the above assumes that you have already activated the RSSI percentage display. This would be the G.P. Input mode 2, which can also be set using the serial terminal command:-

SET GP MODE 2

Some of you may notice that there is a GP MODE 3. I do not recommend using this (dBm display) because you will need to calibrate this very accurately for it to make any real sense at all.

5. It should be fairly common knowledge at this point that some receivers will require a buffer for the RSSI line. The reason for this is three fold:-

- a) the RSSI voltage is used for internal AGC (gain control) purposes of the receiver
- b) it was not really meant to drive an external ADC (such as found on the OSD)
- c) because of a & b, the RSSI voltage may be pulled too low and cause the receiver to malfunction

Some receivers such as the DragonLink do not have this problem but many of the other receivers do. If you need such a buffer, you can try the following link:-

<http://www.dpcav.com/xcart/product.php?printable=Y&productid=16500&cat=0&page=14&js=n>

6. Different receivers will require different calibrations, even if they're the same model and make.

7. For those of you who are planning to fly long range, I would suggest not going below about 30% to be on the safe side, unless you are equipped with a functioning RTH system.

Current and mAh Calibration

To start with, if you are using the SFE Atto-sensor, your Current Offset value needs to be at 0.0. Also make sure that the resistor indicated in the link below, has been removed:-

<download/file.php?id=709>

If you are using an Allegro based sensor, you may need to set the Current Offset to whatever the output voltage of the sensor is when there is no current flowing through it.

The current reading should be calibrated before use. Ideally, you'd connect a Wattmeter in series and power up the motor and watch the current reading on the Wattmeter and compare that to the OSD reading. Then you would adjust the Current multiplier (IMULT) such that the two match up.

Suppose that the wattmeter reading is IW and the OSD reading is IO, and the current multiplier is M1 during the test. The adjusted value for the multiplier should be M2 where:-

$$M2 = IW / IO * M1$$

You should also calibrate the mAh reading. This is done by flying a fully charged pack and noting the final mAh reading. Then charge the battery with a charger that reports the amount of mAh going into the battery. Note the differences and then adjust the mAh multiplier such that they agree.

Suppose that the mAh reading on the charger for a full top-up charge is WT and the mAh reading indicated at the end of the flight on the OSD is WO, and the mAh multiplier is presently M1, then adjusted multiplier should be M2 where:-

$$M2 = WT / WO * M1$$

V1 and V2 Calibration

Because of component variations about the tolerance values, the voltage readings may not be as accurate as it could be. If you want the highest accuracy for the voltages read, you should calibrate them. In order to do so, you will need an accurate digital multi-meter (DMM).

Let us start with V1, which is the supply voltage for the DOSD+ board. Whatever voltage present on these pins will be read as V1. Note that if you are powering the board using the current sensor board and jumpering J1, then V1 and V2 are connected and the same voltage is being read by both channels. Nevertheless, both channels need to be calibrated individually.

Hook up the power to the DOSD+ and see what the reading at the top left corner is (assuming that you have not disabled the display). Let us call this reading VOSD. While still connected, use the DMM in DC-voltage mode and measure the voltage across the battery terminals. Call this reading VDMM. Assuming that the V1 multiplier is presently M1. The new adjusted value for the V1 multiplier should be:-

$$M2 = (VDMM / VOSD) * M1$$

The multiplier for V2 is determined in the same way. Connect the current sensor to the DOSD+ board and then connect a battery to it. Note the voltage reading at the bottom left group and call that VOSD. Measure the voltage across the battery (which is connected to the current sensor)

terminals using the DMM in DC-voltage mode, and call that VDMM. Where the present V2 multiplier is M1, the new adjusted V2 multiplier is given as M2 using the above formula.

Be sure to save your changes.

Powering your OSD

The DOSD+ has an on-board switching power supply that is very efficient and heavily filtered so as to prevent the possibility of supply noise getting in or out of the lines. Being a switching power supply means that the regulator will stay cool all the time, without wasting power in heat dissipation, thus extending your battery life. It also allows for a wide range of power supply voltage to be used - from about 4.6V up to 20V (although we do not recommend using more than about 16V).

If you do not want to use a secondary battery to power the OSD board, you may power it off the battery attached to the current sensor. To do this, you should place a jumper across J1 and the middle-pin of the power header. This will effectively connect V2 on the current sensor header to the supply pin (middle-pin) of the power header. In this case, both the voltages displayed on the OSD (internal and external power) will be the same and the user may choose to disable one of the displayed voltages via the menu option.

Because the DragonOSD+ is able to run at 5V without modification, the user may also power the OSD from the receiver's 5V line. The easiest way to do this would be to connect a jumper from one of the 5V lines to the supply pin. The wire may be soldered under the board. This will only work if the PPM input header is being connected to the receiver and the 5V from the receiver is being passed on through that cable to the OSD board. Given that the OSD only draws a scant 55mA, this arrangement would work without issue.

FTDI to DOSD+ Adaptor Fabrication



The purpose of this section is to describe how to fabricate an adaptor for FTDI Chip's "**TTL-232R-3V3 USB to TTL Serial Converter Cable**". This adaptor will allow you to connect the FTDI cable to the DOSD+ to enable serial communications.

Requirements

Materials

- Light Duty 7" Servo/Battery Wire with Male JR/Hitec Plug 28 AWG or similar (<http://www.bphobbies.com/view.asp?id=V895018&pid=W897953>)
 - (3) Female Gold Terminals (http://www.hansenhobbies.com/products/connectors/pt1inconnectors/pt1in_ft/)
 - (1) Gold Single Row Female to Male Adapter Header - 40 Pin (http://www.hansenhobbies.com/products/connectors/pt1inconnectors/pt1in_srfma40/)
 - (1).1" 1x2 Connector Housing (http://www.hansenhobbies.com/products/connectors/pt1inconnectors/pt1in_1x2/)
 - (1) .1" 1x1 Connector Housing (http://www.hansenhobbies.com/products/connectors/pt1inconnectors/pt1in_1x1/)
- OR
- Light Duty 7" Servo/Battery Wire with Male JR/Hitec Plug 28 AWG or similar (<http://www.bphobbies.com/view.asp?id=V895018&pid=W897953>)
 - (3) Male Gold Terminals (http://www.hansenhobbies.com/products/connectors/pt1inconnectors/pt1in_mt/)
 - (1).1" 1x5 Connector Housing (http://www.hansenhobbies.com/products/connectors/pt1inconnectors/pt1in_1x5/)

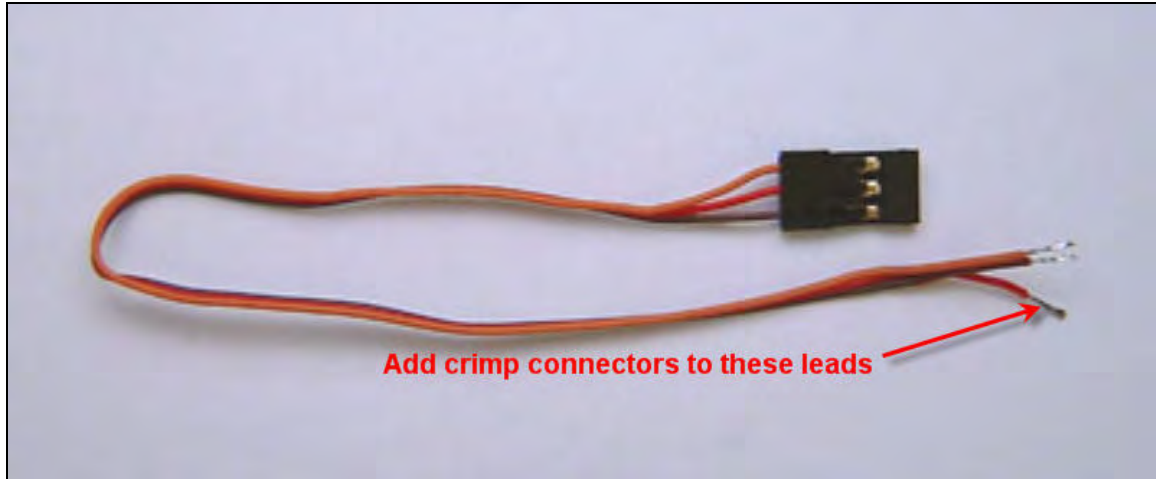
Required Tools

Tools

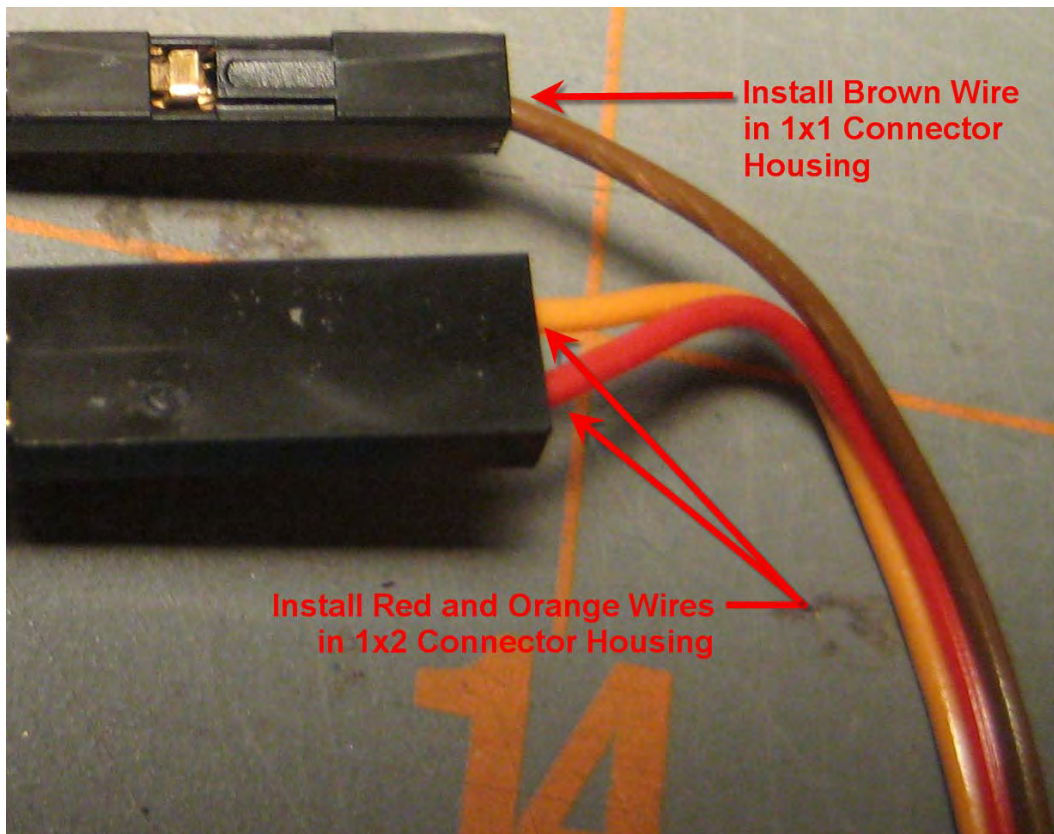
- Crimper
- Wire strippers
- Cutter Pliers

Assembling the Adaptor:

1. Remove the Light Duty 7" Servo/Battery Wire with Male JR/Hitec Plug from it's packaging. You will need to crimp on (3) female gold terminal connectors on to the end of each wire.

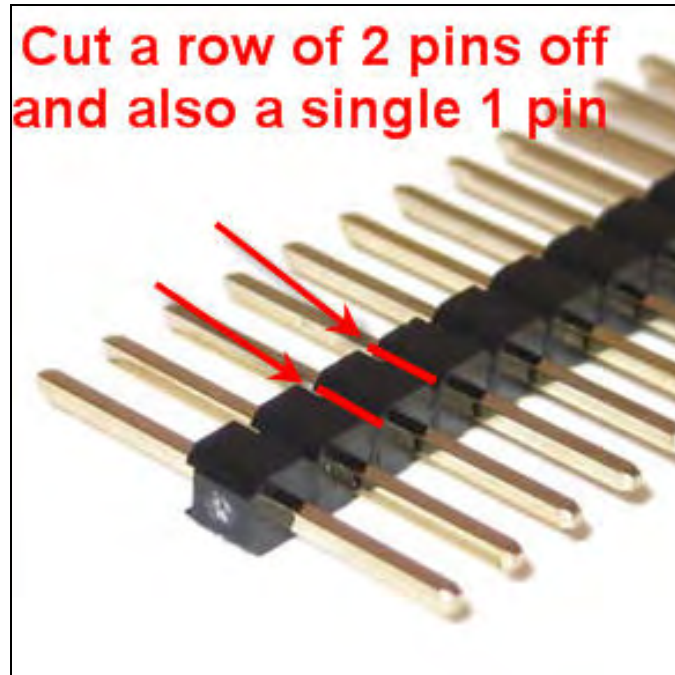


2. Install the connector with the brown wire in the 1 x 1 connector housing. Install the red and orange wires in the 1 x 2 connector housing.

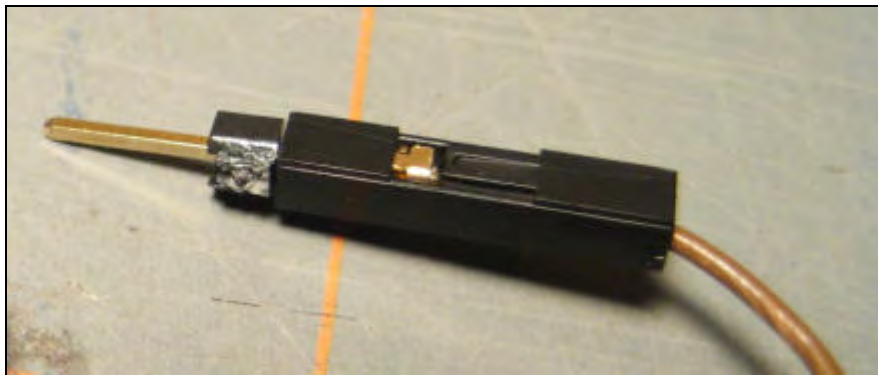


If you purchased a servo cable with Futaba color codes instead of JR you will need to install the black wire in the 1 x 1 connector housing and the red and white wires in the 1 x 2 connector housing.

3. Take the 40 Pin Gold Single Row Female to Male Adapter Header and cut a row consisting of two pins off of it. Then cut a single pin row. The rows are easy to separate using cutter pliers.

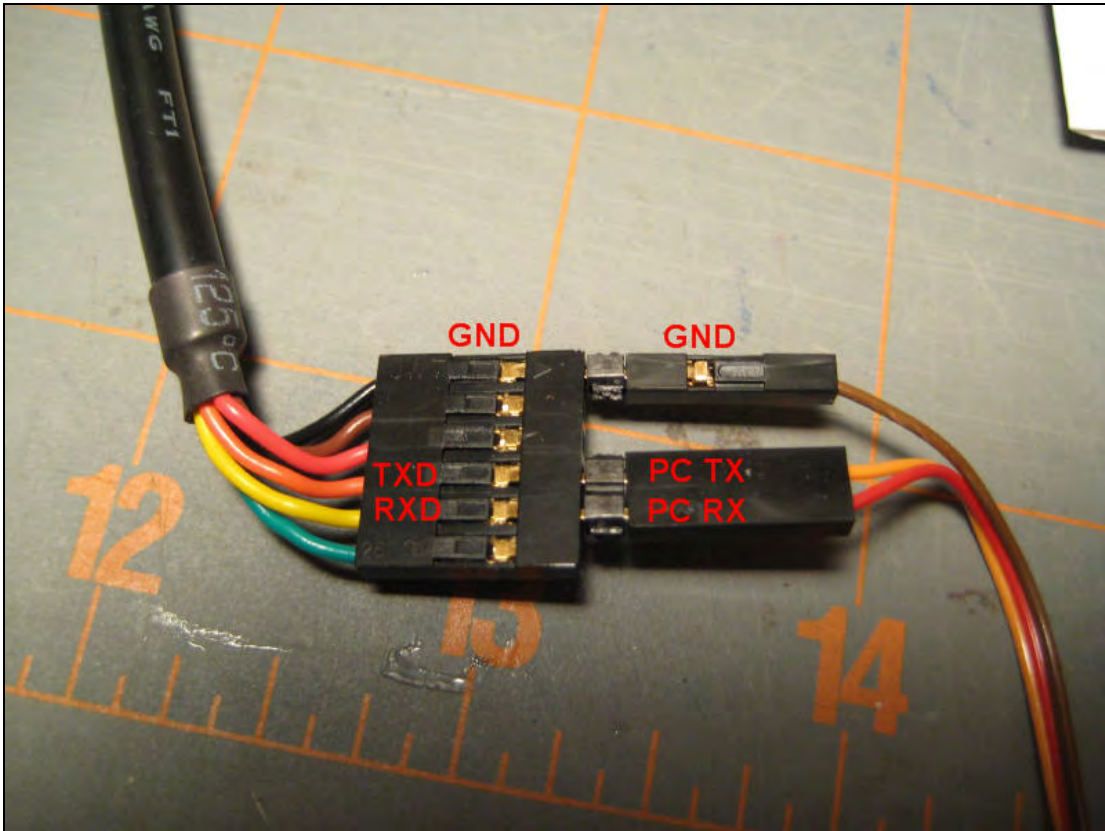


4. Insert the single row pin into the 1 x 1 connector housing. Repeat this with the 1 x 2 housing and two pin row.

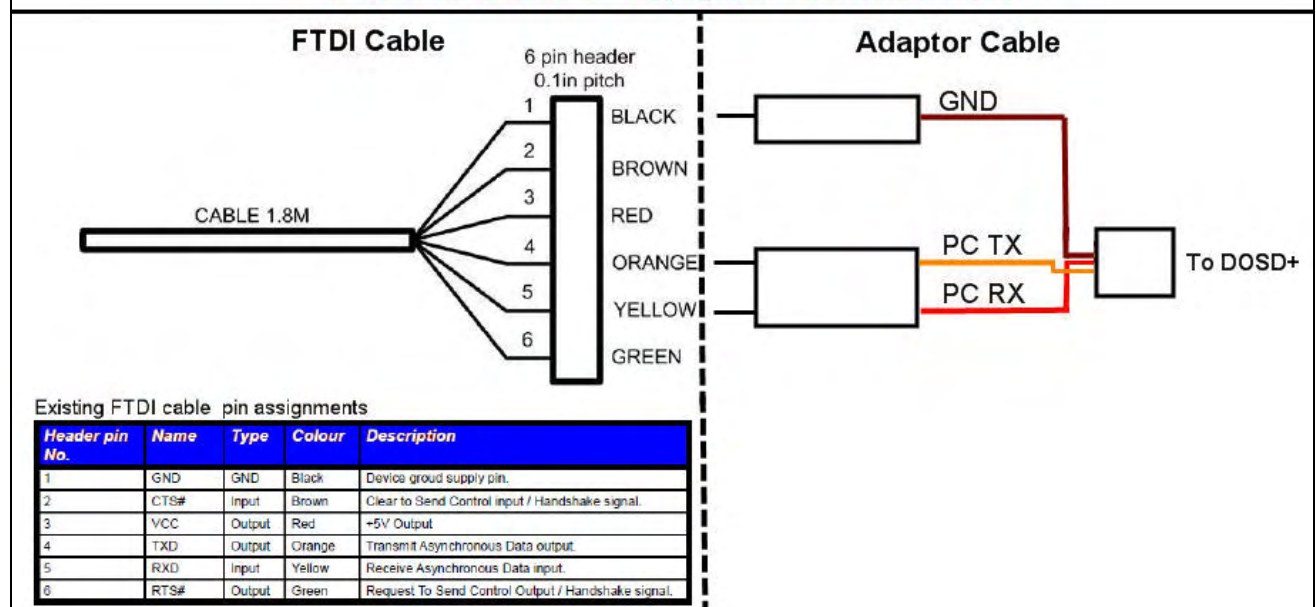


Connecting the Adaptor to The FTDI Cable

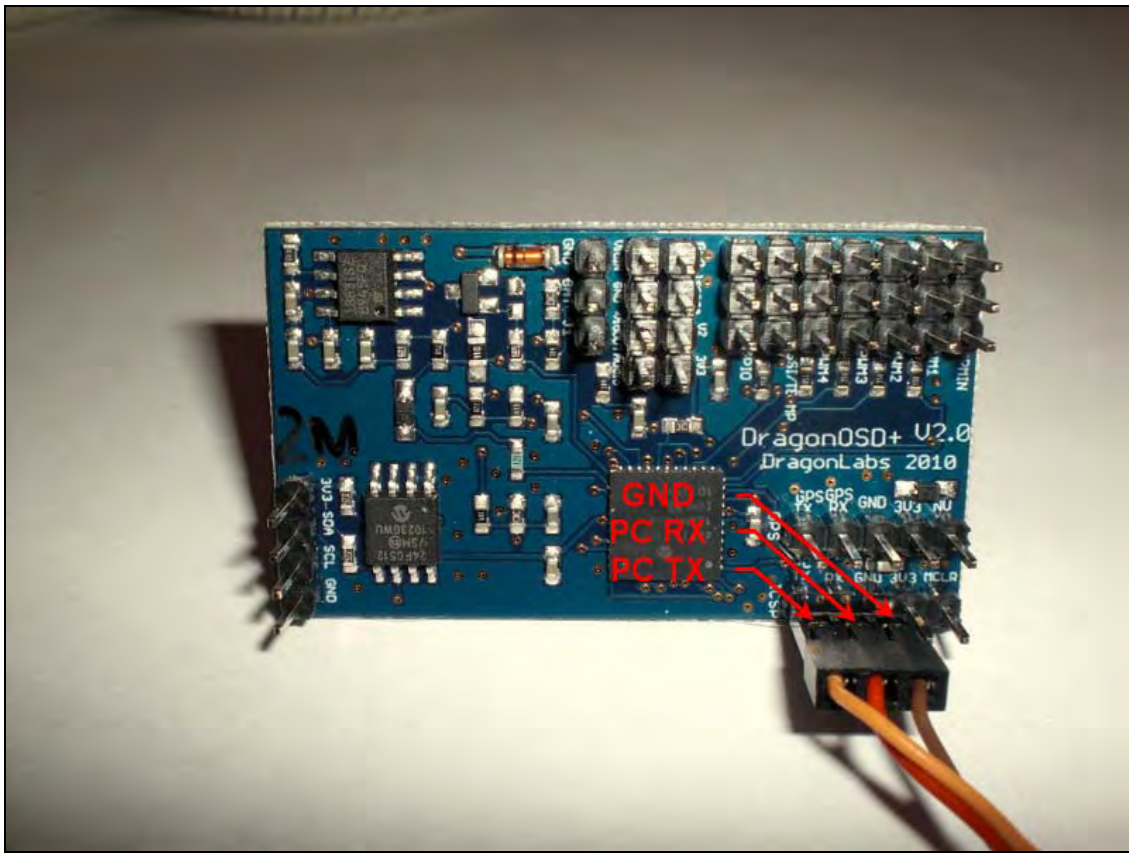
5. Plug the 1 x 1 connector and 1 x 2 connector into the FTDI cable as shown. Pay attention to the pin assignments and verify everything is properly connected.



FTDI to DOSD+ Adaptor Connections



6. Connect the male plug to the DOSD+ as shown. You should now be able to use your terminal software to communicate with the DOSD+.



Getting the PPM Stream Data Out of Your Receiver

The DOSD+ requires a PPM stream from your receiver. When a PPM-stream is needed from a radio, usually some kind of modification needs to be made. Most PPM radios will have no problems with this since there will be the PPM-stream available (such as in the Dragon Link) somewhere in the radio so what you basically need to do is to open the radio up and find the data stream and hook a wire up to it to, possibly, an unused servo channel (after cutting the trace for that channel first, of course).

For PCM radios, it's a bit more of a toss-up since some PCM radios do have the PPM data stream available on board although in some cases it may be limited (in channels - FFAST radios have a 5-channels PPM-stream available internally). In other cases the internal design may be a pure digital to servo implementation, in which case there will be no PPM stream available anywhere. For these, a channel multiplexer, active or passive would be required.

To that end, a number of approaches are presented here for you. Which route you choose will be up to you based on the hardware you have to work with.

Finding the PPM signal

The first task, especially with a PCM style radio, is to identify the PPM signal connection point. The following link will assist you with this:

http://paparazzi.enac.fr/wiki/Other_Hardware#Recommended_35.2F40Mhz_R.2FC_Receivers

Some receivers need to have the PPM signal buffered because the signal is not strong enough to be used as an output. If this is the case a simple PPM Buffer can be built. If you do not wish to do this there are commercially available PPM encoders. One of these encoders is listed following this section.

Building a PPM Buffer

A PPM buffer should not normally be needed, as the DOSD+ V2.0 has a built in unity buffer on the input. However in cases where you want to add an additional buffer this section describes how to fabricate a PPM Buffer / Isolator. This buffer can be used in instances, where your 72mhz Rx is not capable of supplying an acceptable PPM signal to the DOSD+ because of line loading.

The buffer consists of a LM358 operational amplifier set up as a voltage follower which isolates the signal. As the input PPM signal changes, the output will change by an equal amount. Since it has a high input impedance, it makes an excellent isolation amplifier, allowing your Rx to function normally.

Materials

- (1) LM358 Low Power Dual Operational Amplifier
(<http://www.mouser.com/?Keyword=LM358AN&FS=True>)
- (1) 8-Pin Retention Contact socket (optional)
(<http://www.radioshack.com/product/index.jsp?productId=2062604>)
- (5) Female Gold Terminals
(http://www.hansenhobbies.com/products/connectors/pt1inconnectors/pt1in_ft/)
- (1) Gold Single Row Female to Male Adapter Header - 40 Pin

- (http://www.hansenhobbies.com/products/connectors/pt1inconnectors/pt1in_srfma40/)
- (1).1" 1x3 Connector Housing
(http://www.hansenhobbies.com/products/connectors/pt1inconnectors/pt1in_1x3/)
- (1).1" 1x2 Connector Housing
(http://www.hansenhobbies.com/products/connectors/pt1inconnectors/pt1in_1x2/)
- (1) .1" 1x1 Connector Housing
(http://www.hansenhobbies.com/products/connectors/pt1inconnectors/pt1in_1x1/)
- (1) Toroid Ferrite Core EMI/RFI Suppressor, 10mms
(<http://www.dpcav.com/xcart/product.php?productid=16499>)
- 3 FT of 32 AWG, 2 conductor wire or similar
(<http://www.dpcav.com/xcart/product.php?productid=16503>)
- Shrink tubing

Required Tools

Tools

- Crimper
- Wire strippers
- Cutter Pliers
- Soldering Iron
- Solder

Identify your PPM Signal Lead and Provide a Connection Point:

Before you begin building your PPM Buffer / Isolator, you need to identify where your PPM signal lead is on your particular Rx. This document does not cover the procedure on how to identify your particular PPM signal. However there are a number of web sites which may help you do this:

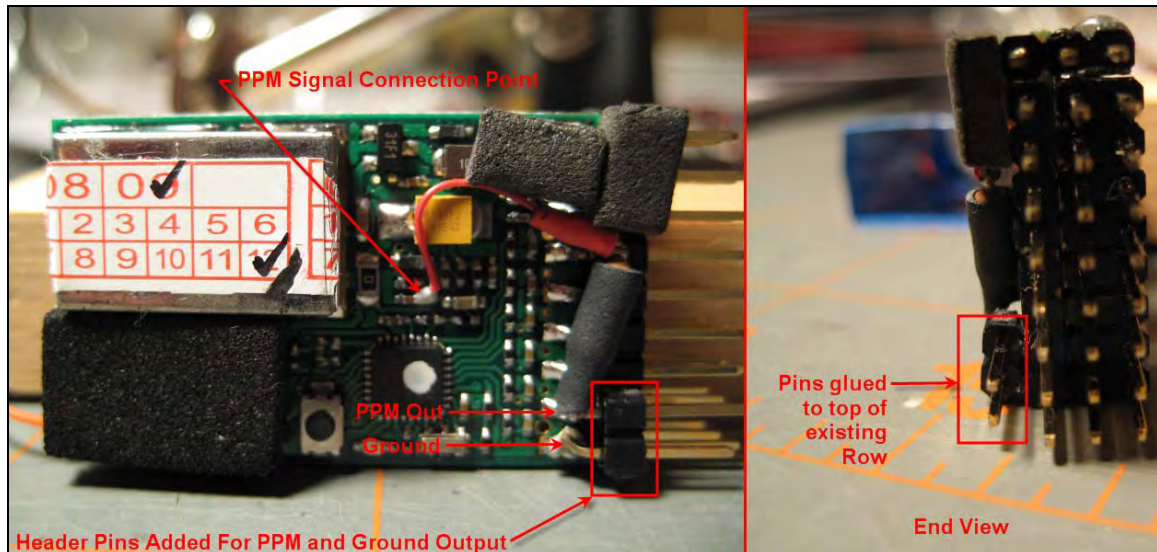
<http://www.forum.tsebi.com/viewtopic.php?f=7&t=37>

http://paparazzi.enac.fr/wiki/Other_Hardware#Recommended_35.2F40Mhz_R.2FC_Receive

Once the Rx PPM connection point has been identified, you need to route that signal lead out of your Rx, along with a ground connection. The easiest way to do this is to use a set of header connector pins as a termination point:

(http://www.hansenhobbies.com/products/connectors/pt1inconnectors/pt1in_srfma40/)

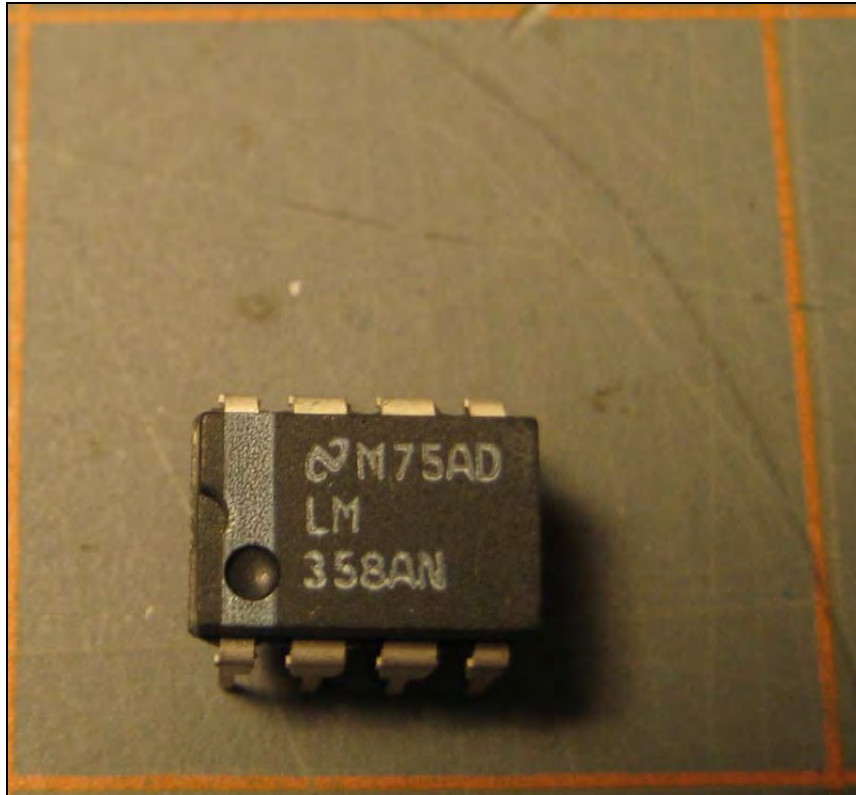
That way, you can use a connector to attach the PPM Buffer / Isolator. Shown below is an example of this type of arrangement, using a Corona 8 channel Rx:



Once you have routed the PPM signal lead out of the Rx, you can start work on your buffer. NOTE: Some Rx's require a resistor in series with the PPM signal lead. Typically this resistor will have a value between 10K and 100K. If you connect your PPM signal lead to the buffer and experience problems with your Rx, try adding a resistor in this range.

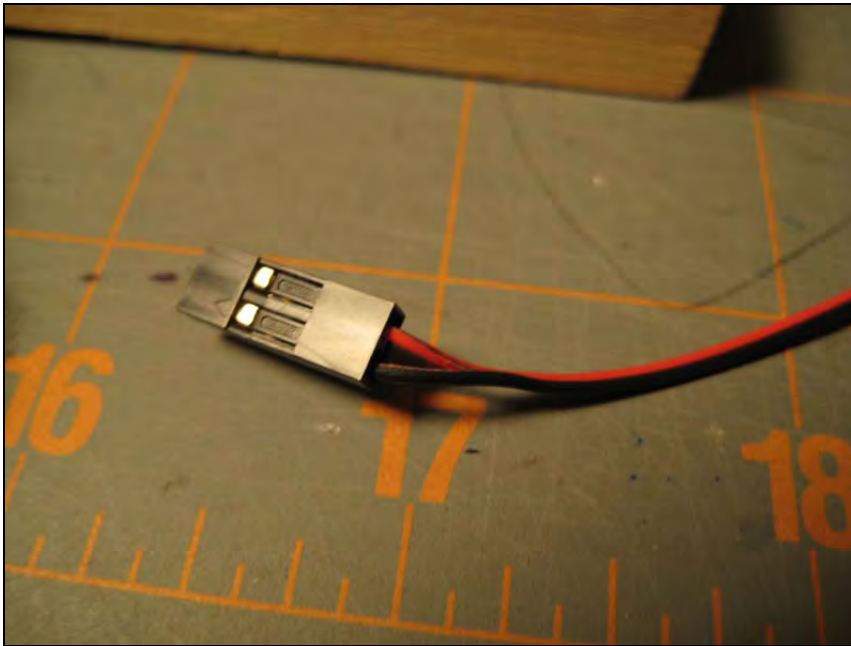
Assembling your Buffer / Isolator:

The PPM Buffer / Isolator uses a LM358 Low Power Dual Operational Amplifier:



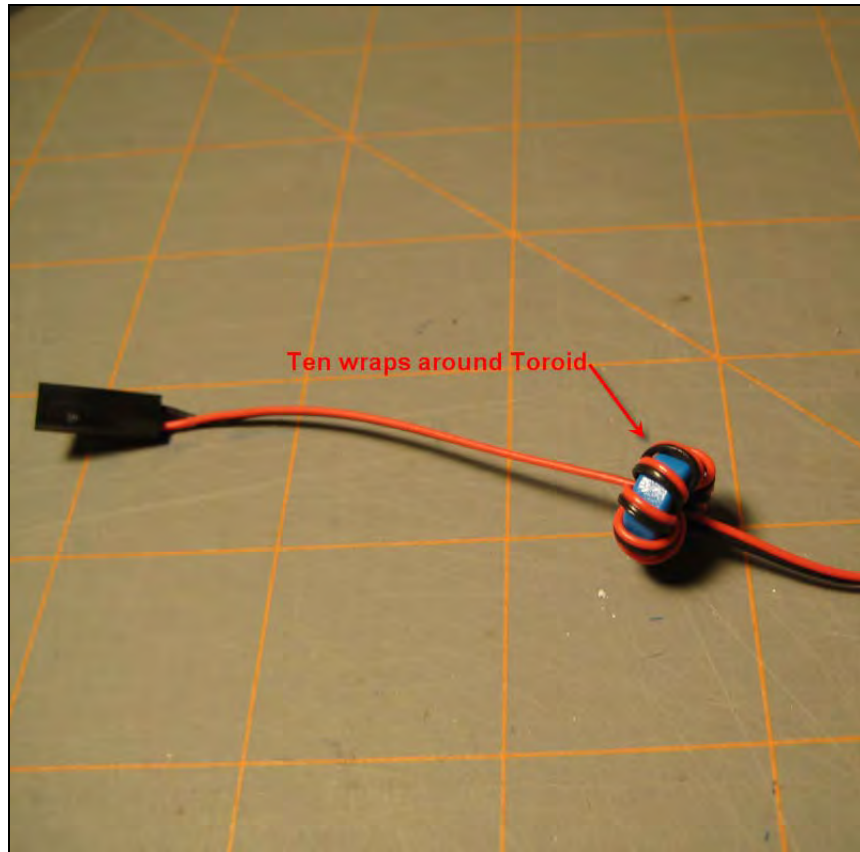
This 8 pin chip is available from a variety of sources and does the job well. If you prefer, you can solder your connections directly to the chip itself, rather than place the chip in an 8 pin socket. But the socket provides you an added layer of protection, by avoiding damage to the chip while soldering.

1. Take your 32 AWG, 2 conductor wire and crimp on (2) female gold terminal connectors on one end. Once this is done, install the wires in a 1 x 2 connector housing, as shown below.

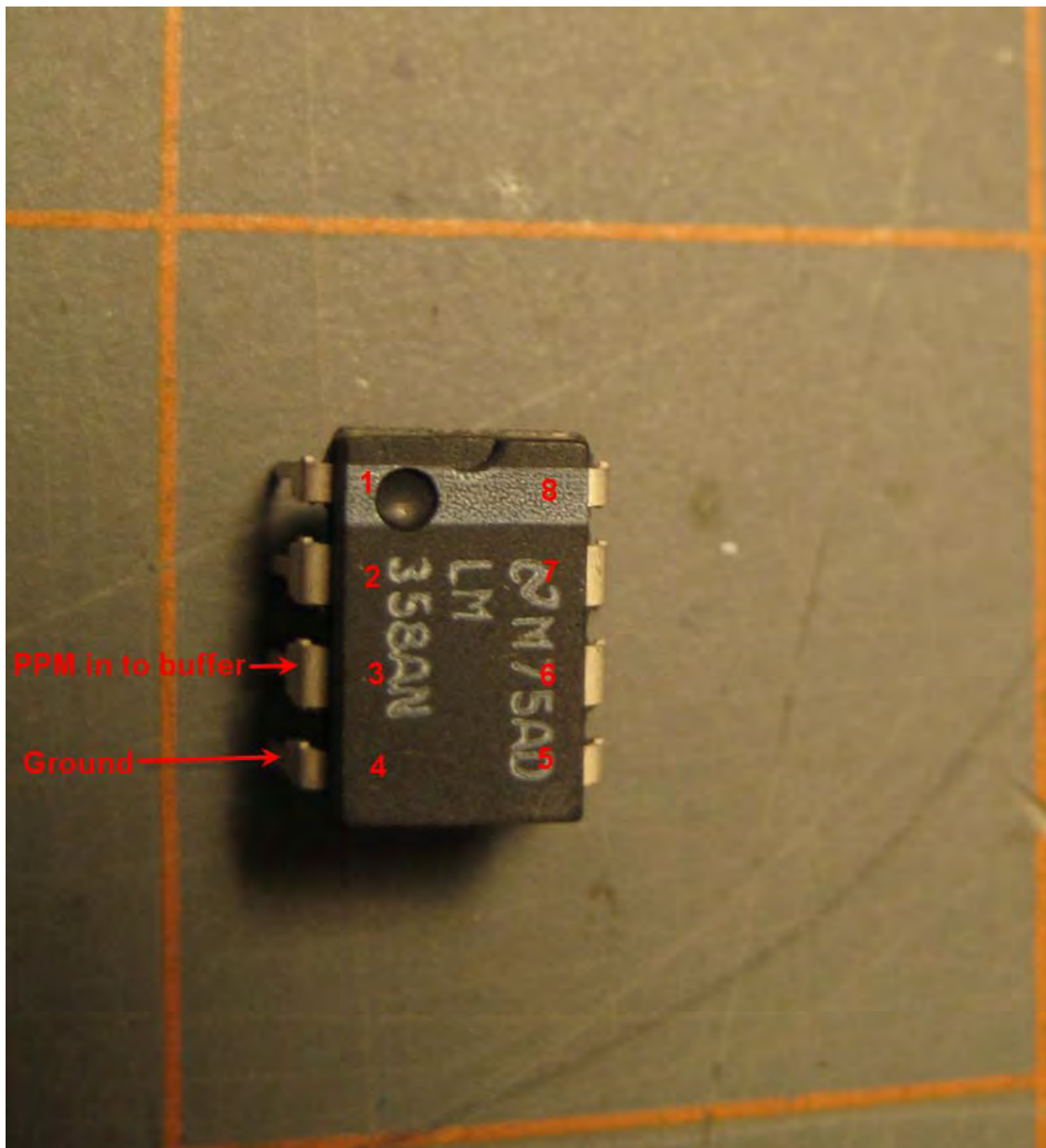


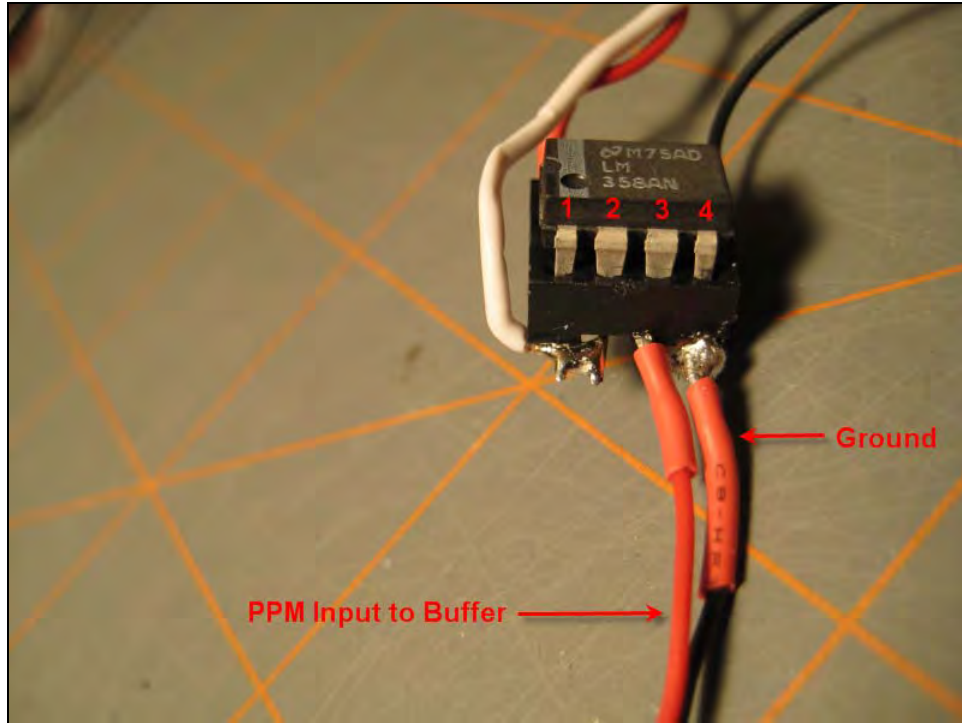
Using the hookup wire shown, the red wire will be the PPM input to the buffer, and the black wire will be ground.

2. About 1 to 2 inches from the connector place a toroid ring and wrap the cable around it approximately ten times. The toroid helps keep noise out of your Rx.



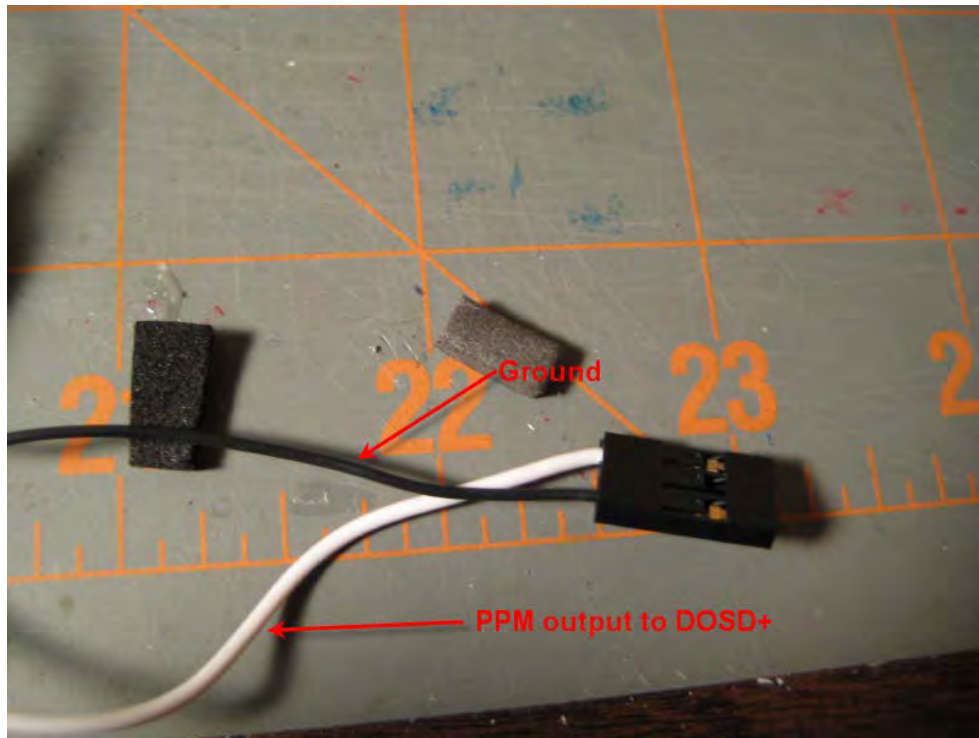
You will be connecting the PPM signal lead and ground connections to pins 3 and 4 of the socket or chip. The ground connection will also have an additional wire connected to pin 4 that goes to the DOSD+.





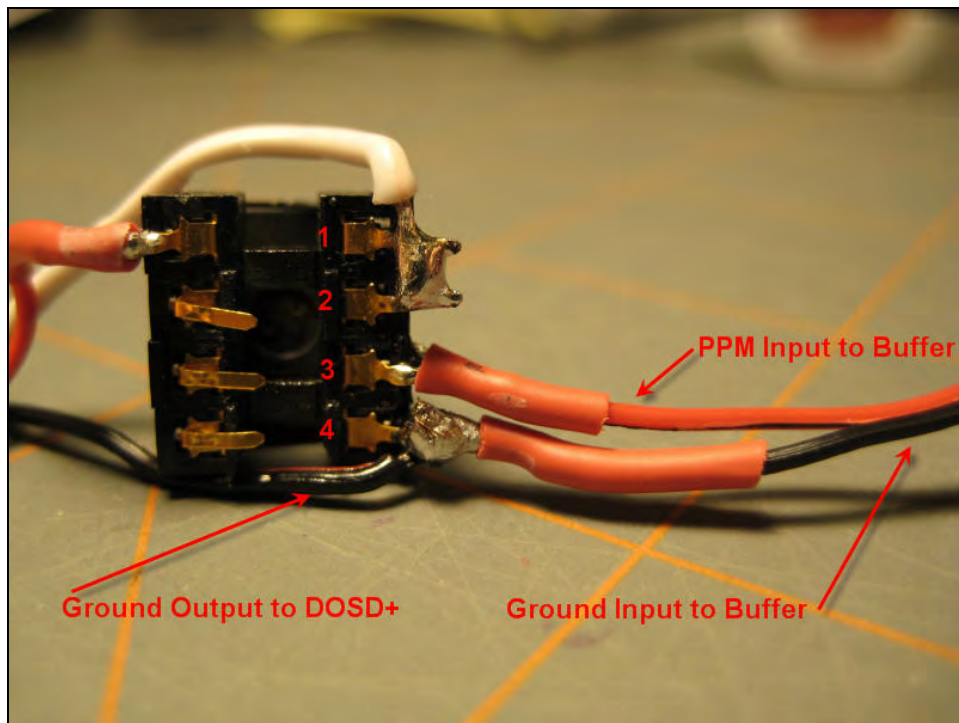
Solder the red PPM input lead to pin 3 of the socket or chip. Do not solder the ground lead yet to pin 4.

3. Take some more of your 32 AWG, 2 conductor wire and crimp on (2) female gold terminal connectors on one end. Once this is done, install the wires in a 1 x 3 connector housing, as shown below. The white wire terminal connector should be plugged into the first position of the housing and the black wire terminal connector should be plugged into the third position of the housing. The middle position is not used.

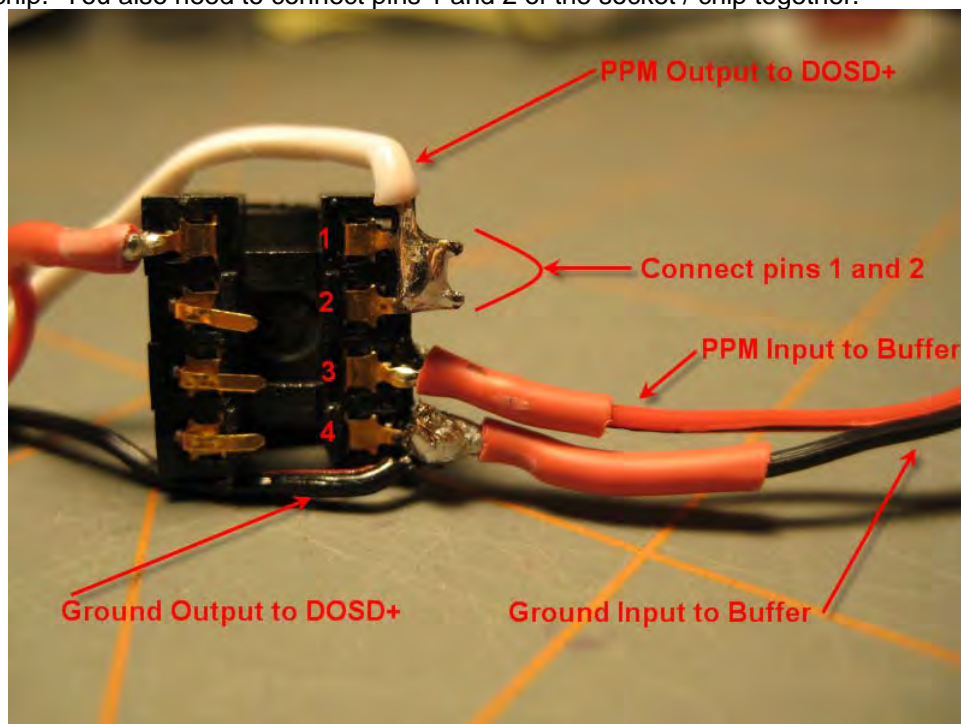


This will become your PPM input connection to the DOSD+.

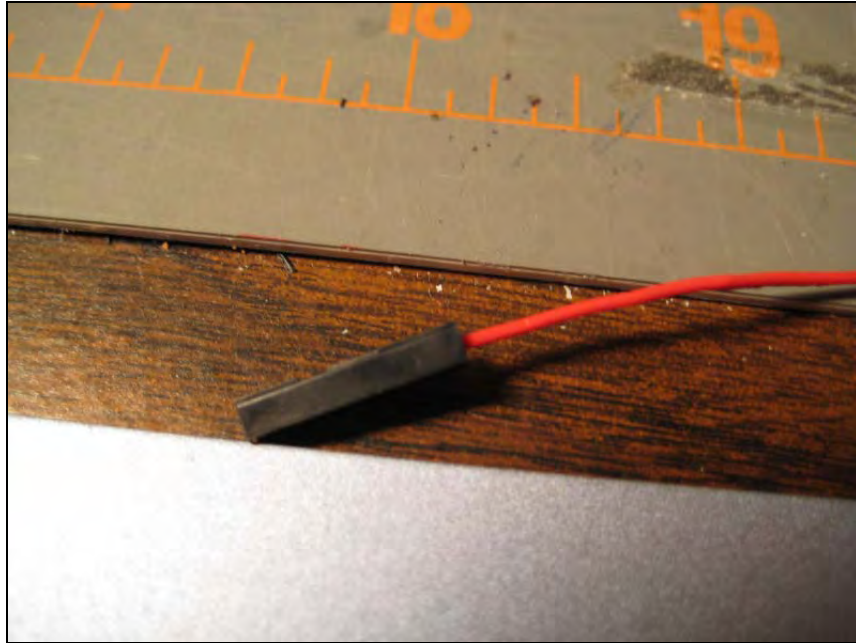
4. Solder both black ground leads to pin 4 of the socket / chip. One of the black leads is from the toroid wrapped wire. The other is from the 3 position connector that goes to the DOSD+.



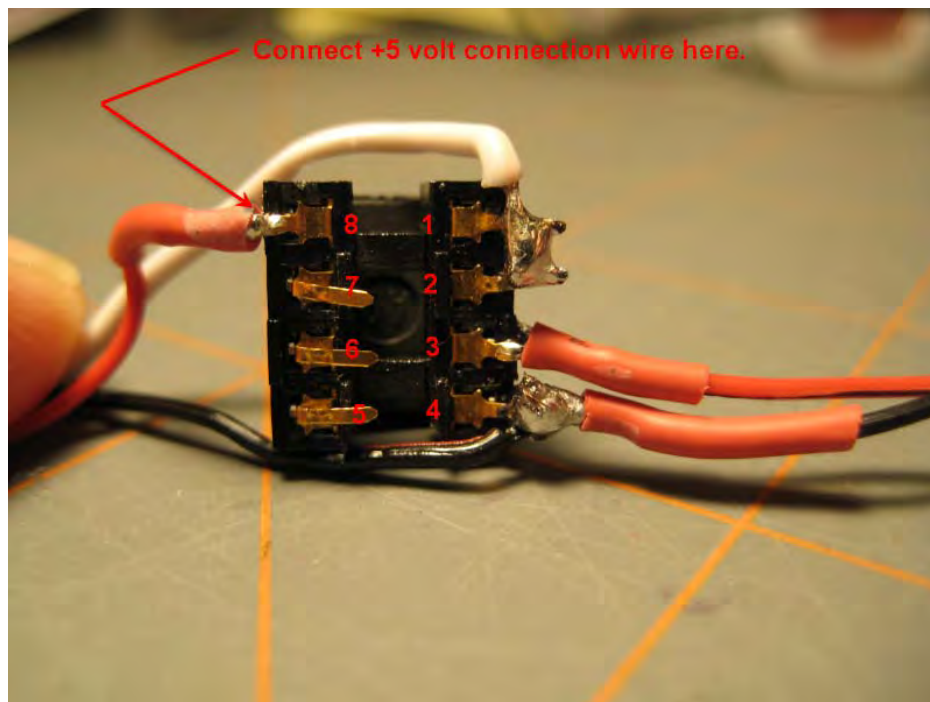
5. Now connect the white PPM output wire which goes to the DOSD+ to pin 1 of the socket /chip. You also need to connect pins 1 and 2 of the socket / chip together.



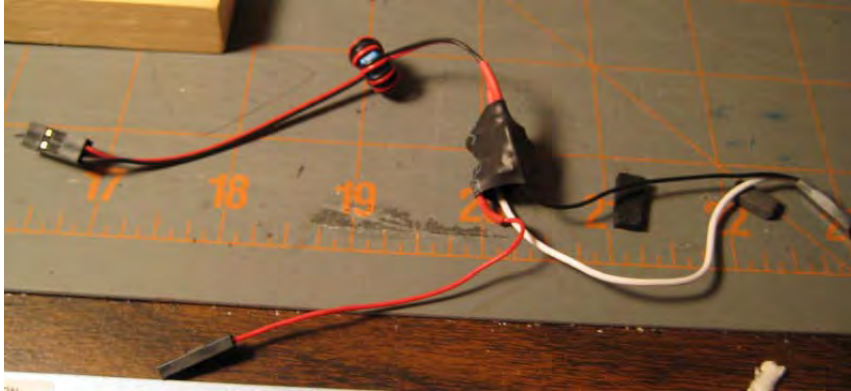
6. Take a single conductor of your 32 AWG, 2 conductor wire and crimp on (1) female gold terminal connector on one end. Once this is done, install the wires in a 1 x 1 connector housing, as shown below. This wire will supply +5 volts to your buffer. The best place to obtain this is from an unused channel position on your receiver.



7. Now connect the other end of the wire to pin 8 of the socket /chip.

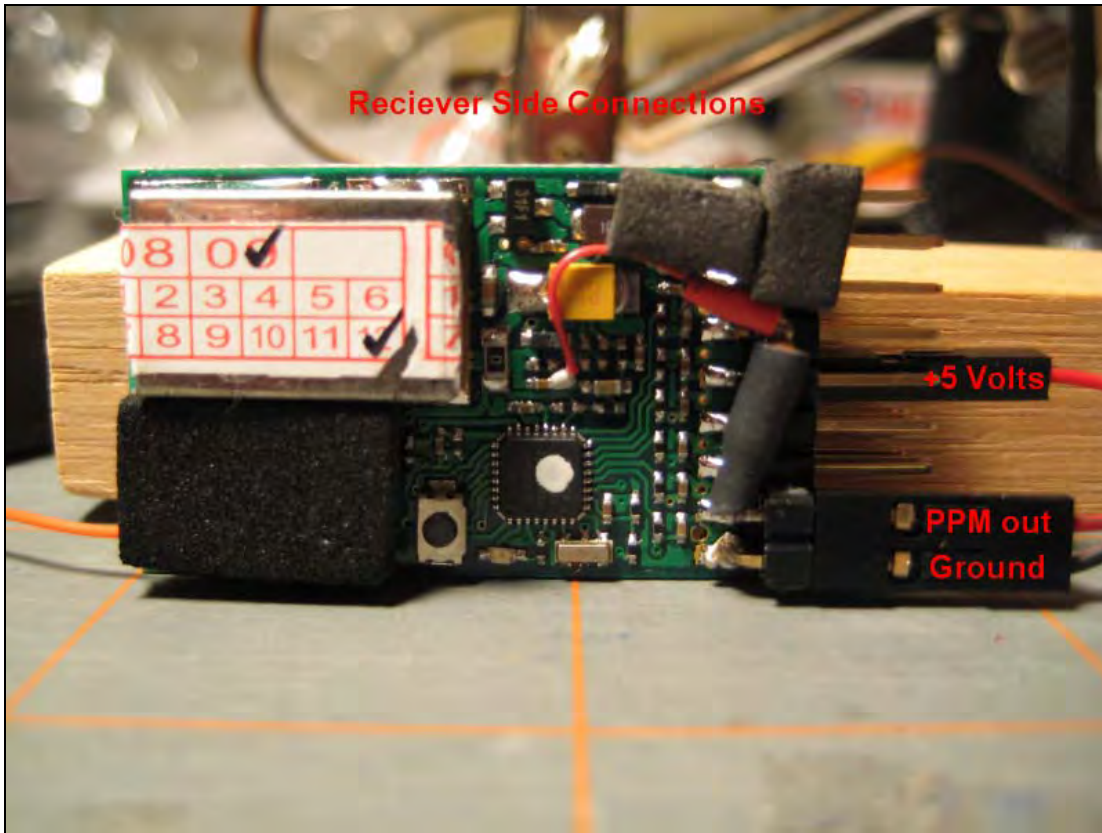


8. Apply some heat shrink tubing around the socket / chip assembly to protect the connections and you are done.

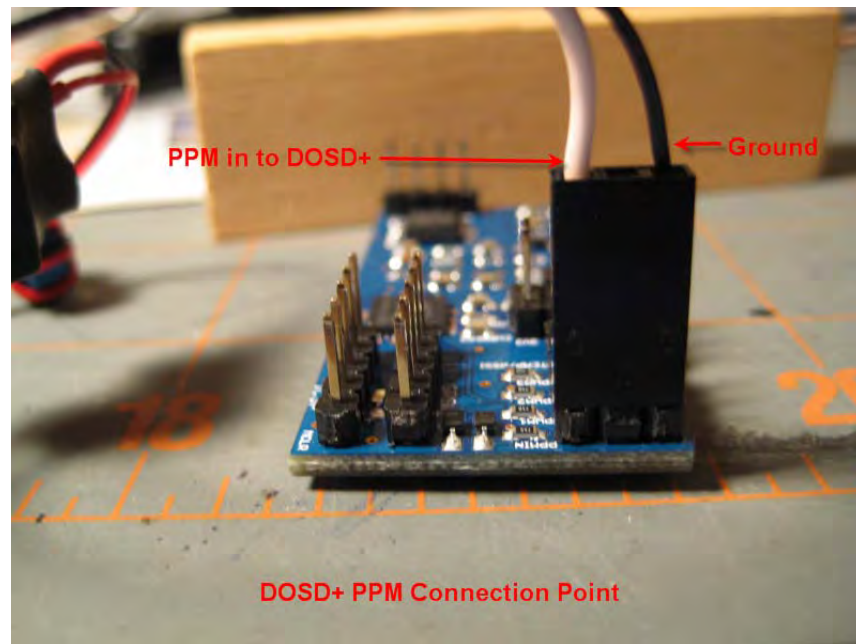


Connecting The Buffer / Isolator to the DOSD+ and Receiver:

Connect the PPM output and ground connection connector to your Rx as shown below. Also connect the +5 volt power to a spare unused channel position's +5 volt terminal:



The other connector gets attached to the DOSD+ as shown.



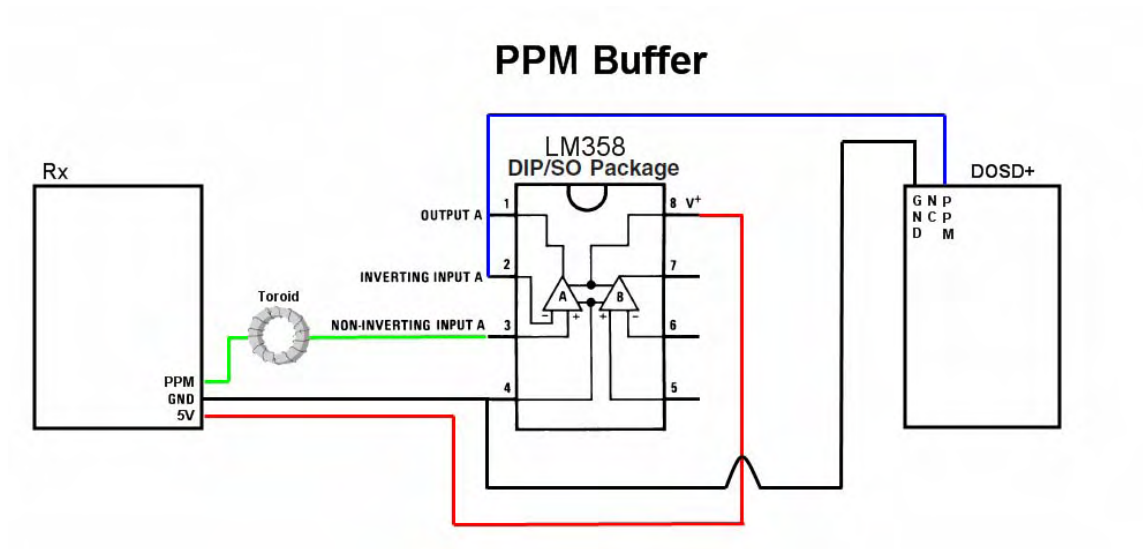
The buffer can now be used with your DOSD+.

Final Notes about Using the PPM Signal from your Rx:

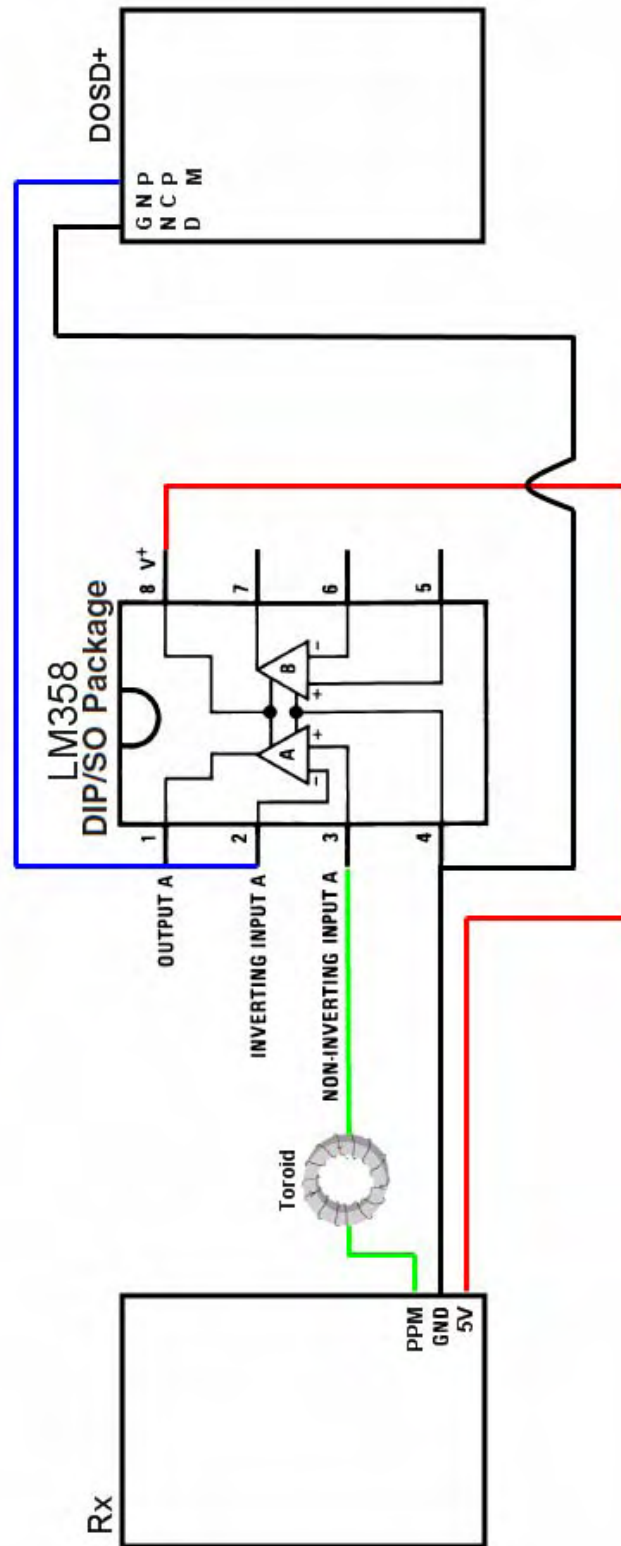
Some receivers do not produce a stable PPM signal when your transmitter is switched off, or when the receiver goes into “fail safe” mode. You should verify the operation of your particular receiver and take this into account when planning on how to use your particular system. You should never rely on the RTH function to work properly where the PPM signal is not stable during fail safe as unpredictable results will occur.

Wiring Block Diagram:

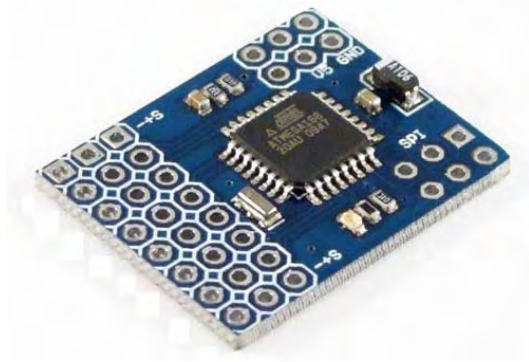
The following block diagram illustrates the connections between your receiver, buffer, and DOSD+. Use this diagram to verify your connections and when troubleshooting.



PPM Buffer



Using an 8 Channel DIY Drones PPM Encoder Board



This section describes how to install and use the DIY Drones 8 channel PPM encoder board with the DOSD+. The encoder can be used in instances, where your 72mhz Rx is not capable of supplying an acceptable PPM signal, or if you do not wish to modify your existing Rx.

Requirements

Materials

- (1) 8 Channel PPM Encoder Version 2
(http://store.diydrones.com/product_p/br-ppme.htm)
- (2) 3x4 Right Angle Pin Headers
(http://store.diydrones.com/product_p/pr-0001-04.htm)
- (4) or more Servo Extension Cable 5cm Female-Female (see installation notes)
(http://store.diydrones.com/product_p/pr-0003-03-5cm.htm)
- (1) or more Servo Y Connector cables (see installation notes)
(<http://www.bphobbies.com/view.asp?id=V834691&pid=T090448>)
- (1) Gold Single Row Straight Header - 40 Pin
(http://www.hansenhobbies.com/products/connectors/pt1inconnectors/pt1in_srsh40/)

OR

- (24) Female Gold Terminals
(http://www.hansenhobbies.com/products/connectors/pt1inconnectors/pt1in_ft/)
- (8) or more .1" 1x3 Connector Housing
(http://www.hansenhobbies.com/products/connectors/pt1inconnectors/pt1in_1x3/)
- 3 FT of servo wire
(<http://www.hansenhobbies.com/products/connectors/servowire/>)
- Shrink tubing

Required Tools

Tools

- Crimper
- Wire strippers
- Cutter Pliers
- Soldering Iron
- Solder

How The Encoder Works:

Before you begin assembling your PPM encoder board and integrating it with your DOSD+, it will be necessary for you to choose a wiring configuration which works best for you. Choosing a wiring configuration determines a number of things such as:

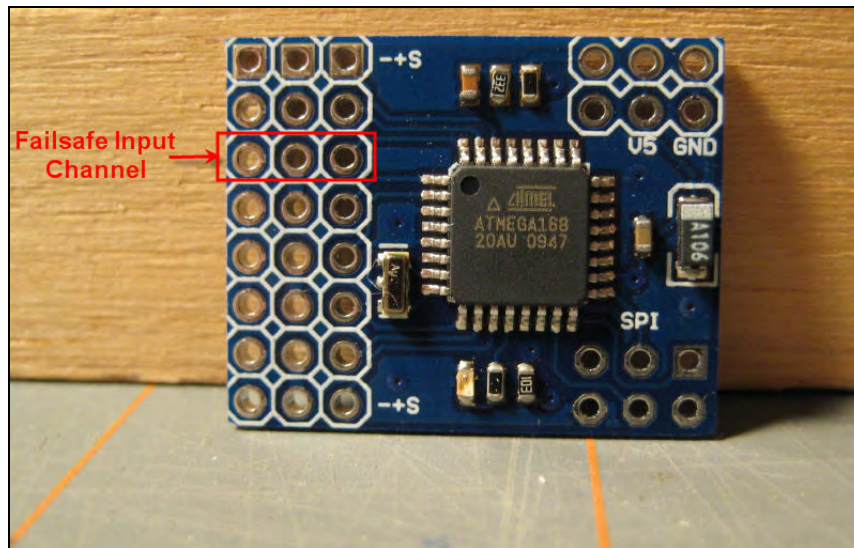
- The hardware you will need
- The space required in your aircraft (the number of wires you will be putting in)
- The complexity level of your installation
- The skills and tools required to achieve success

To choose a configuration that best fits your needs; you first need to understand what the encoder does, and how it works with your receiver. It is suggested that you read the entire procedure first before you start.

The DIY Drones 8 channel PPM encoder board works by taking each servo PWM output signal and combining them into one PPM data stream output. This PPM data output can then be read by the DOSD+. So to achieve this, you need to run the servo outputs from your Rx to the encoder, instead of to each servo. The encoder comes preloaded with the firmware you need, so no programming is needed.

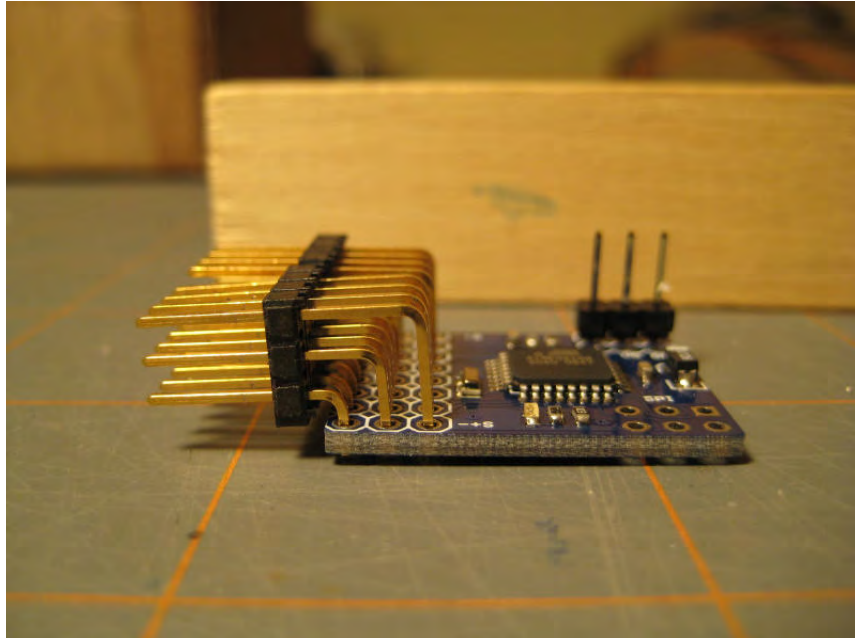
Encoder Requirements:

In order to get the encoder to work, you need to supply power and a “fail safe” input. The power will be provided by the jumpers you will run from any servo channel to the encoder. The “fail safe” input is the 3rd channel input on the encoder. It requires a valid servo channel output to make the encoder work. **Without this input, the encoder board will not produce any PPM output unless you change the firmware.**

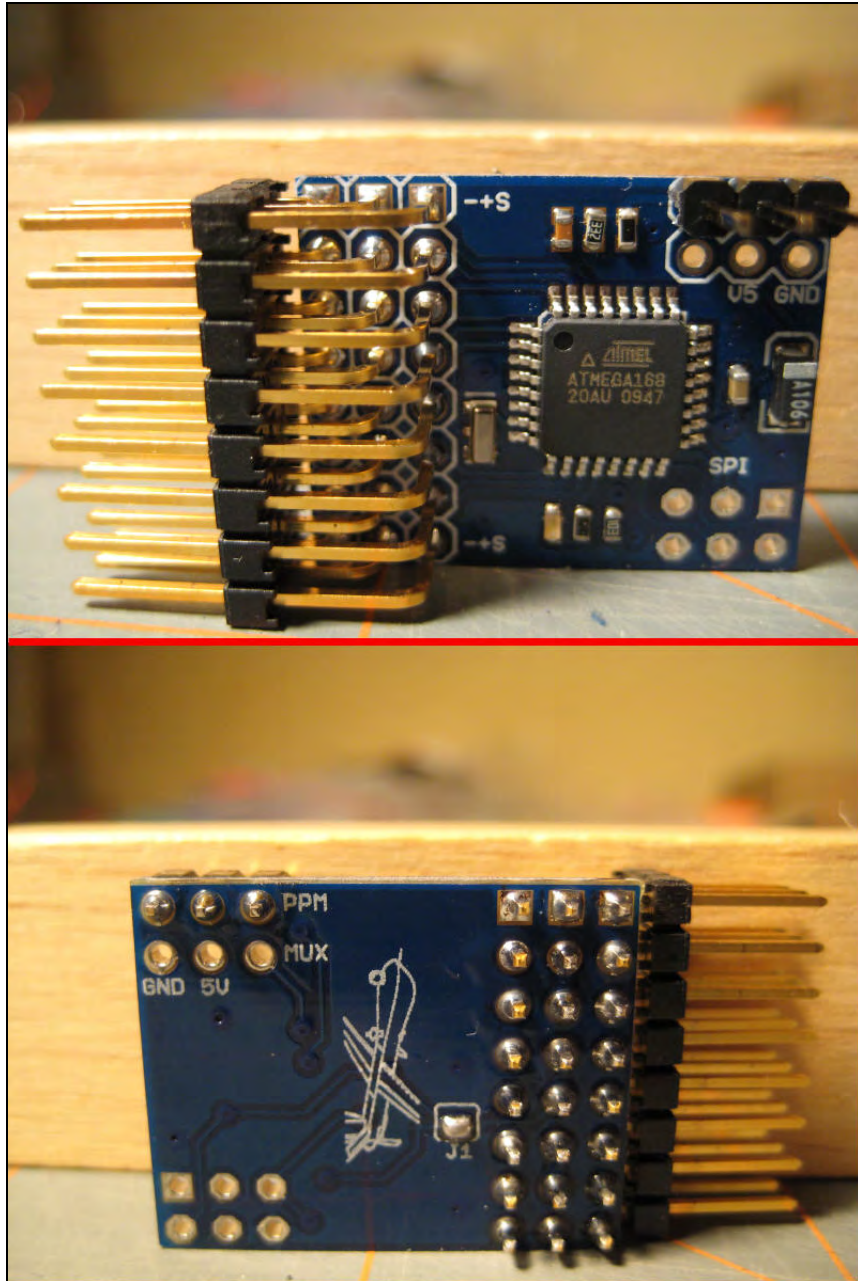


Adding Connection Pins To Your Encoder:

The encoder ships from DIY Drones without any type of connector pins. For a cleaner installation of your cables, it is suggested you get (2) 3x4 right angle pin headers for the servo cable inputs. This will allow you to route the servo cables from your Rx over to the encoder. However, soldering this type of connector requires a high degree of skill, a very small tipped soldering iron, and a steady hand. The board also requires a 3 pin single row straight header for the PPM connection. This can be obtained by cutting a 3 pin portion off of the recommended "gold single row straight header - 40 Pin" strip. As you can see by the image below, the right angle header is an assembly and cannot easily be separated for individual row installation.



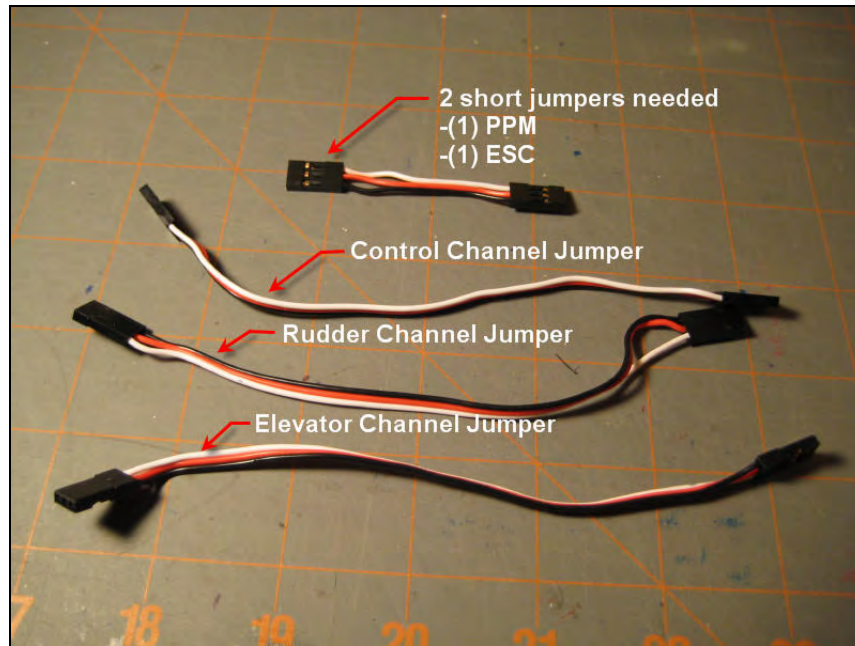
If you are not comfortable using a right angle header, then an alternately, it is suggested you cut 8 more rows of 3 pins from the straight 50 pin header and solder one row in at a time for each channel. This will allow you to “work your way down” the board when soldering. For this installation guide, the right angle headers were used. Once you are done, your board should look like this:



Once your connectors are attached, you are ready to decide on how many servo jumpers you need.

Deciding on a Servo Jumper Configuration:

To decide how many servo jumpers you will need, you need to decide how much PPM information you want to provide to the DOSD. At bare minimum, you need to connect 3 servo channels for the DOSD, (Rudder, Elevator and a control channel, usually channel 5) plus one additional for the encoder failsafe, channel 3 (the ESC) (unless you are doing some sort of jumpering over at the encoder board).



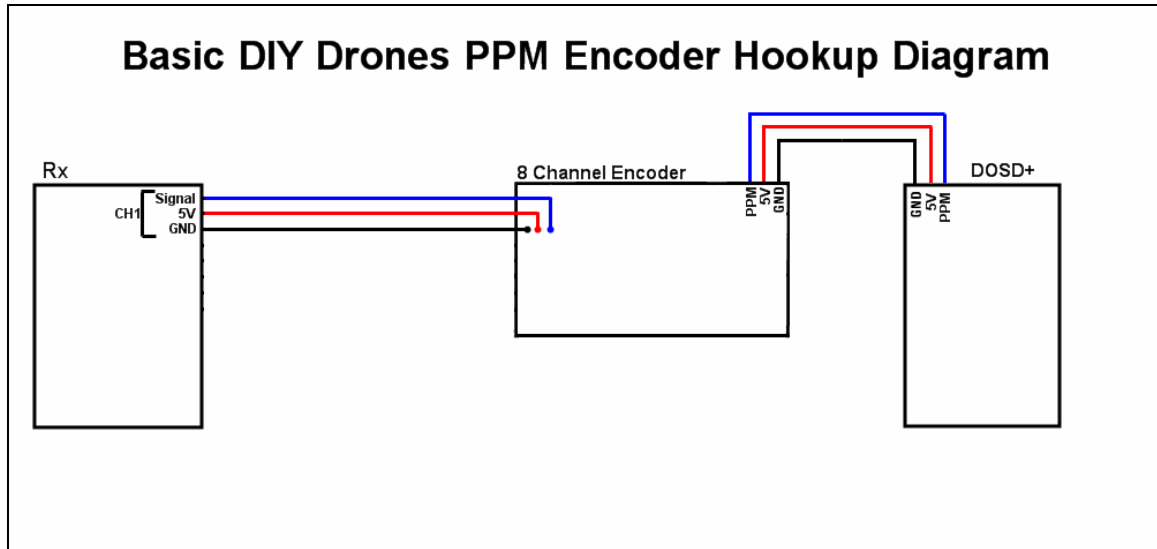
You will also need an additional short jumper for the PPM output out of the encoder. Also a small "Y" servo connector will be needed for the ESC.



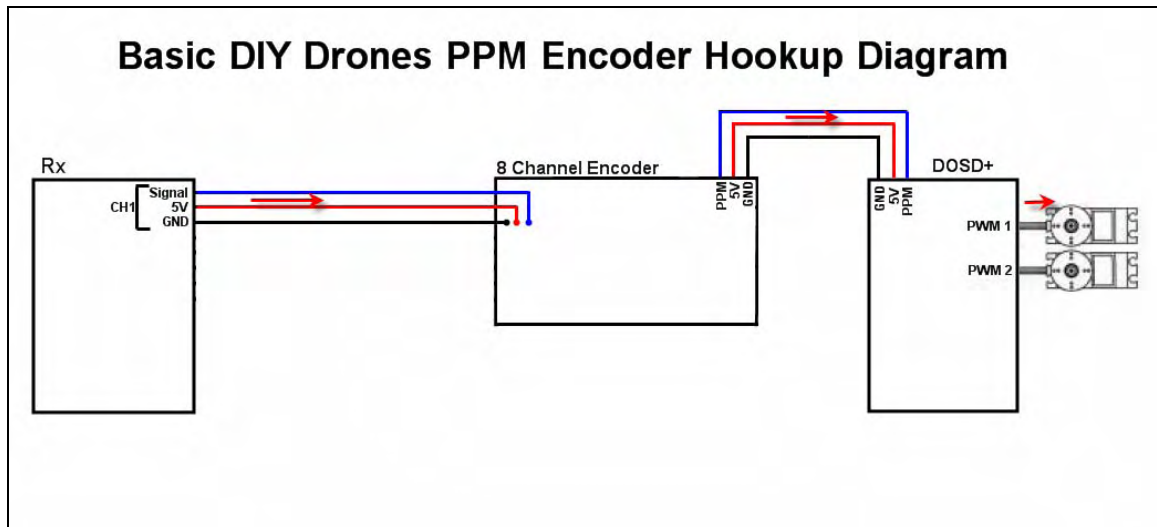
If you want to provide more information in the PPM data stream you need to connect more servo outputs to the encoder. This means more “Y” splitters if the output leads need to run to actual servos.

Block Connection Diagrams:

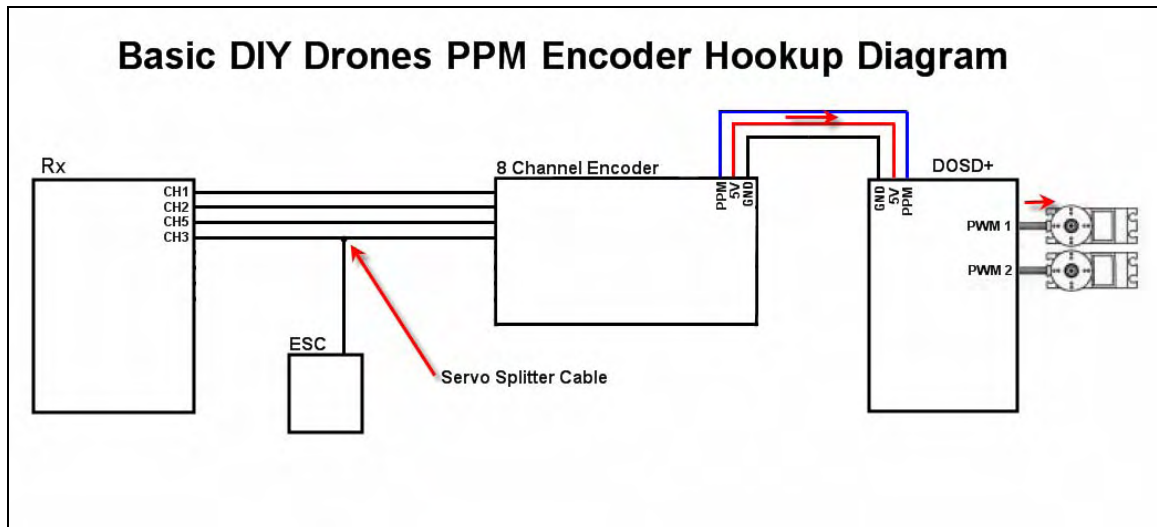
In its most simplistic form, each channel of your receiver is connected the encoder as shown below:



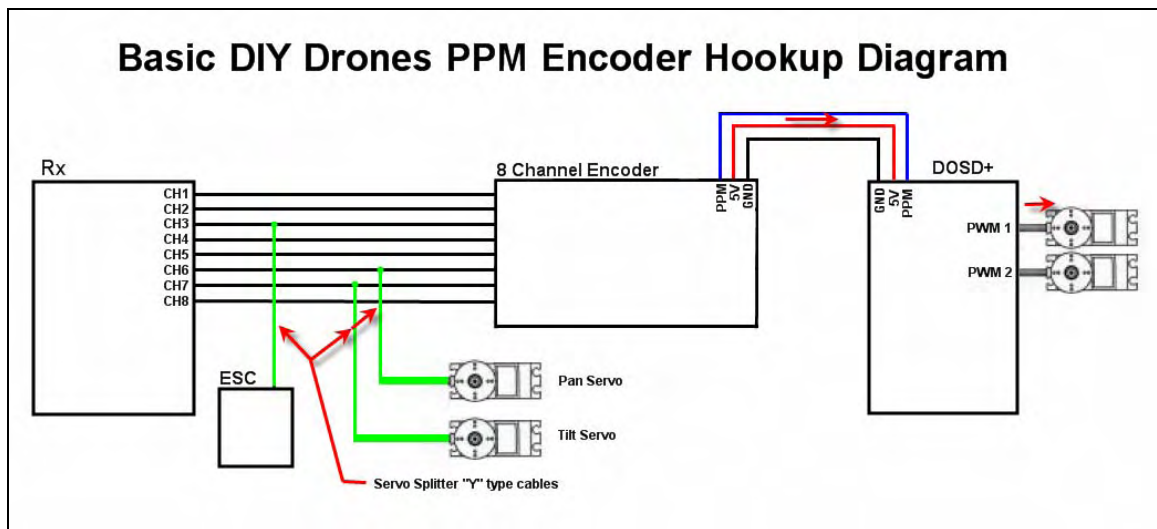
The encoder's PPM output connects to the DOSD+ by way of a short jumper. This jumper also provides 5 volts and ground to the rows of pins on the DOSD where the servos connect. This provides power to your servos so that they will operate when connected to the DOSD+ PWM1 and PWM2 outputs.



For a “basic” setup, as previously mentioned, you need to connect 3 servo channels for the DOSD, (Rudder, Elevator and a control channel, usually channel 5) plus one additional for the encoder failsafe, channel 3 (the ESC) (unless you are doing some sort of jumpering over at the encoder board):



This would provide the basic control functionality that you would need. However if you wish, you can run additional channels from your Rx to the encoder (up to 8) to provide additional information to the DOSD+ (such as pan and tilt information). Keep in mind that if you do this, you may need to split off the servo outputs with additional splitters to the servos, which requires additional room in your aircraft.



Testing The Encoder:

Once you have decided on a wiring configuration, it is a good idea to test your encoder with the Rx. To do this, connect the cabling between Rx, Encoder and ESC and power up. If you have your Tx turned off, the encoder blue LED will probably be on and not blinking. Once you turn the Tx on, the LED should start to blink indicating that an active PPM data stream is being produced. After verifying this, you can connect the PPM cable from the DOSD to the Encoder.



Configuring the DOSD+ for Your Radio:

Normally when powered up, the DOSD+ will attempt to identify your radio type and set the channels used accordingly. However depending on how you cabled your encoder it might not be correctly recognized. To ensure that your DOSD+ will see your inputs properly you can use the Terminal Commands with the Tera Term program to set up the DOSD+.

If you have cabled your encoder for a “basic” setup (aileron on channel 1, elevator on channel 2, ESC on channel 3, rudder on channel 4 and control channel 5), you would use the following “SET” terminal commands to configure your DOSD+:

```
Set ailchan 1  
Set elechan 2  
Set throttlechan 3  
Set ruddchan 4  
Set ctrlchan 5
```

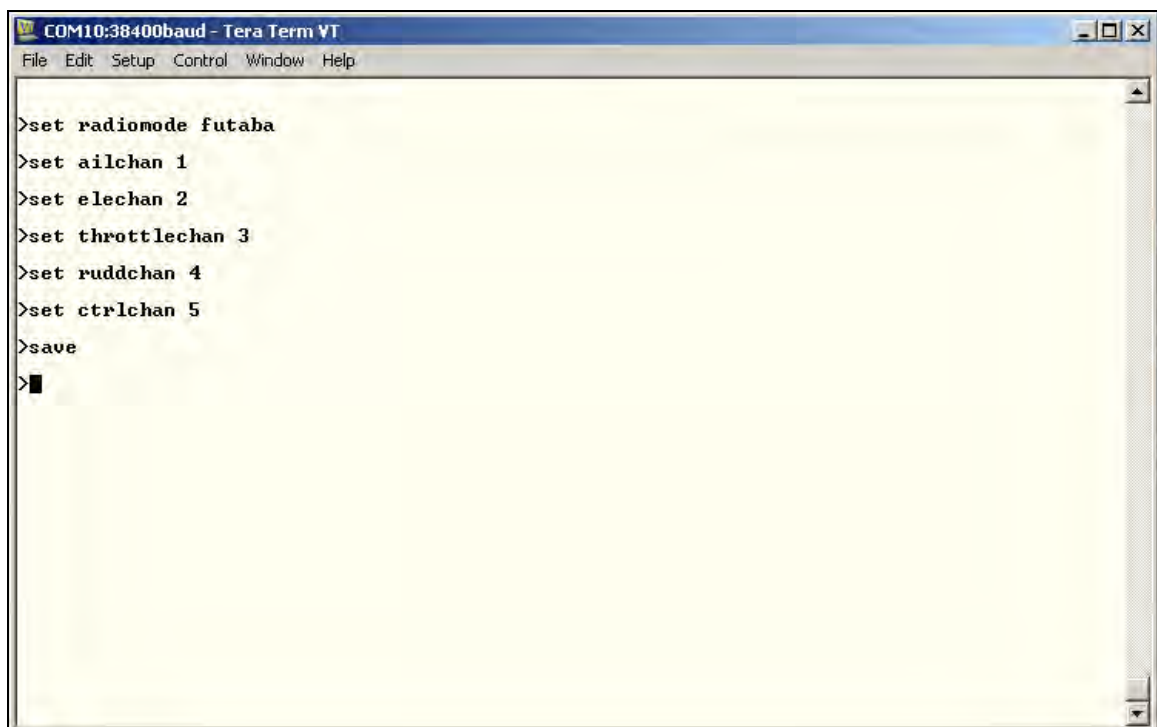
You would also use the “SET RADIOMODE” command to tell the DOSD+ what brand transmitter you are using. So for example if you are using a Futaba transmitter the command you would use would be:

```
Set radiomode futaba
```

Finally, you need to save your settings into the DOSD+ memory with the “SAVE” command like this:

```
Save
```

The below screen shot depicts this:



The screenshot shows a terminal window titled "COM10:38400baud - Tera Term VT". The window has a menu bar with "File", "Edit", "Setup", "Control", "Window", and "Help". The terminal text is as follows:

```
>set radiomode futaba  
>set ailchan 1  
>set elechan 2  
>set throttlechan 3  
>set ruddchan 4  
>set ctrlchan 5  
>save  
>█
```

Using the Serial Port

Once you have connected your PC to the serial port, you can use a program such as Hyper Terminal or TeraTerm Pro to issue serial terminal commands to the DOSD+. Tera Term Pro is a free application that can be download using the following link:

<http://en.sourceforge.jp/projects/ttssh2/releases/>

The serial port settings are 38400, 8N1. (The baud rate has been changed to 115200 from version 7.5Beta11 and upwards.) You will know that things are working if you see a copyright message show up on the terminal once the DragonOSD+ is powered up. If you don't see that, some basic things to check are:-

0. Is the USB converter correctly installed and recognized as a COM port?
1. Is the correct COM port being selected?
2. Make sure no other terminal program (HyperTerm or TeraTerm is running when updating)
3. Are the serial settings correct? (38400 baud, 8N1)
4. Are the RX and TX lines reversed or crossed?
5. Is the serial port being connected to the correct header pins?
6. Is the OSD powered up?

Once you have configured the port in this way you can issue any of the valid DOSD+ commands. Please see "Configuring The DOSD+" for valid serial port commands.

You can also use DragonLabs PC Commander available at this link:

<http://forum.tsebi.com/viewtopic.php?f=7&t=93>

DOSD+ Firmware Upgrading

The DOSD+ has a built-in bootloader that allows firmware upgrades to be performed over a normal TTL-serial connection. In order to do this, you will need a serial -to-TTL converter (usually USB) and hook that up to the OSD board correctly. Additionally, you will need to software to upload the firmware via the bootloader, which also needs to be configured properly. Lastly, you will need the firmware file itself. Note that if you are running this in Windows 7 (and possibly Vista) it may be necessary to run this in Administrator mode if it crashes upon exit in normal mode. Either that or you should install it in a directory other than the Program Files directory.

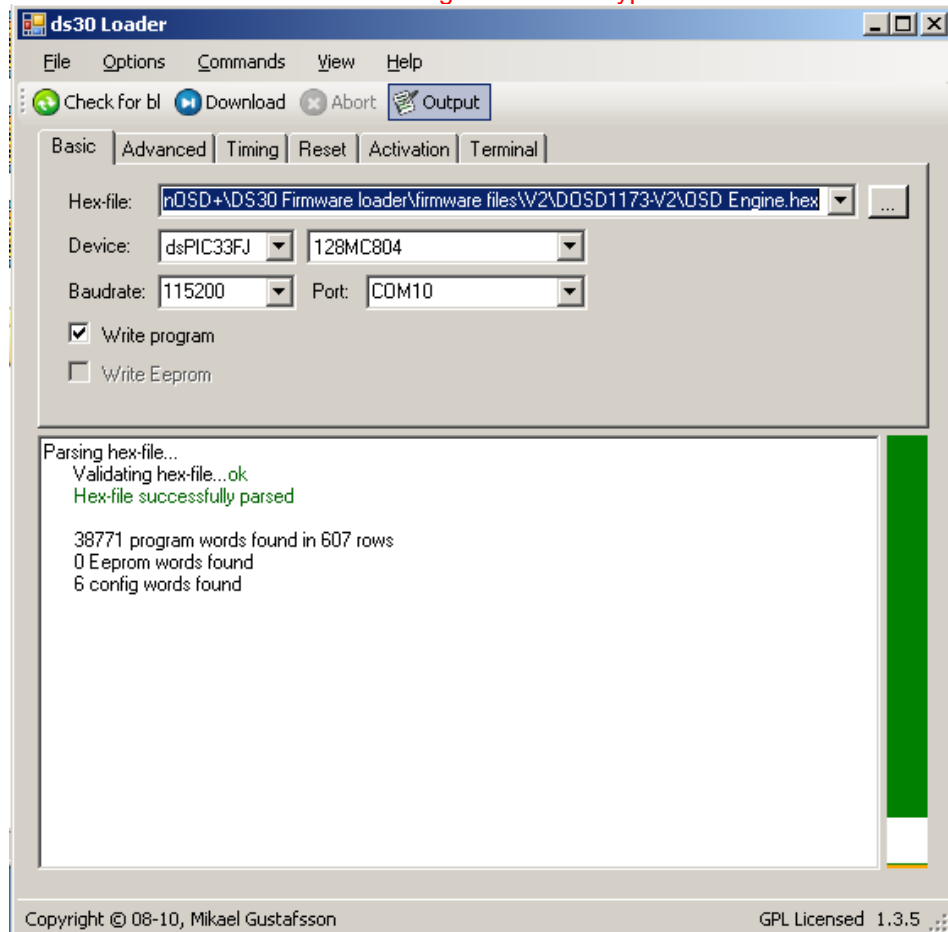
The uploader software can be downloaded from the following link:-

<http://mrmackey.no-ip.org/elektronik/ds30loader/downloads.php>

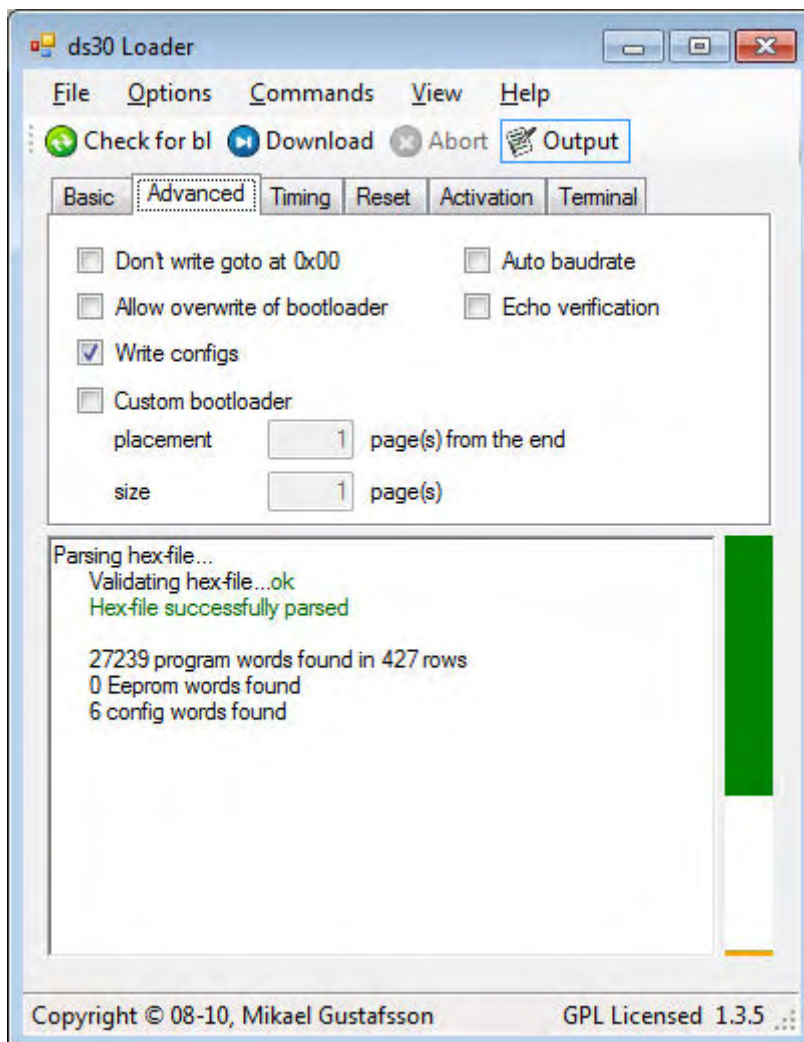
Get the latest Windows installer (assuming you are using Windows), or the version that suits your operating platform (Linux and Mac also supported). [At the time of this writing, the latest version is 1.3.5 in file ds30_loader_100206.exe] After you have installed it, connect your serial-TTL converter which will show up as a COMx port. **Open up the "ds30 Loader GUI", change the "View" to "Advanced Mode"**, and configure it as follows (using the appropriate COMx port and leaving out the Hex-file entry for the moment

Basic Tab Settings:

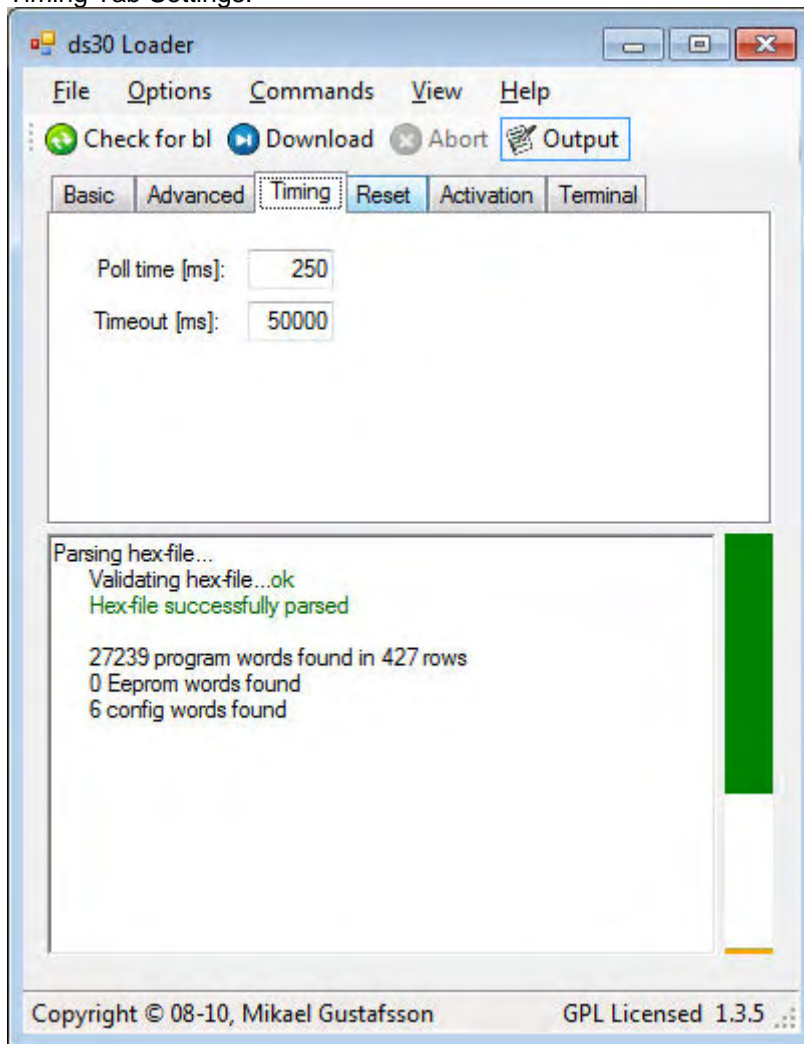
(Note to version 1 users: You need to change the Device type to dsPIC33FJ 128MC804)



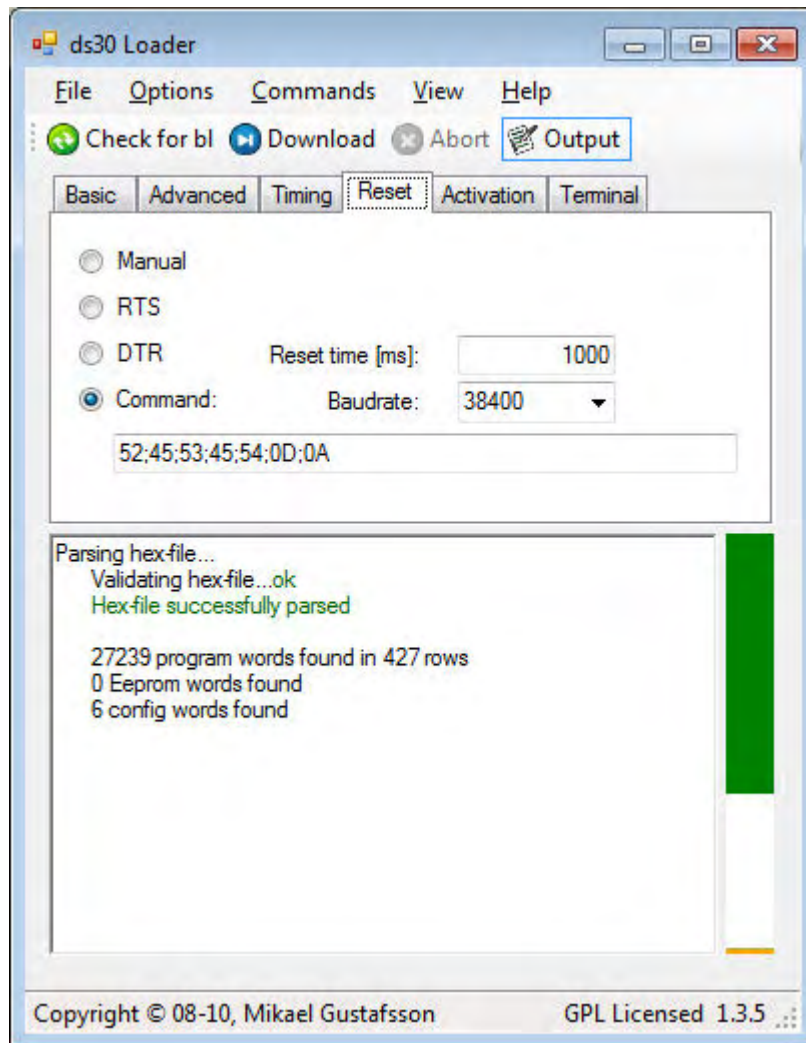
Advanced Tab Settings:



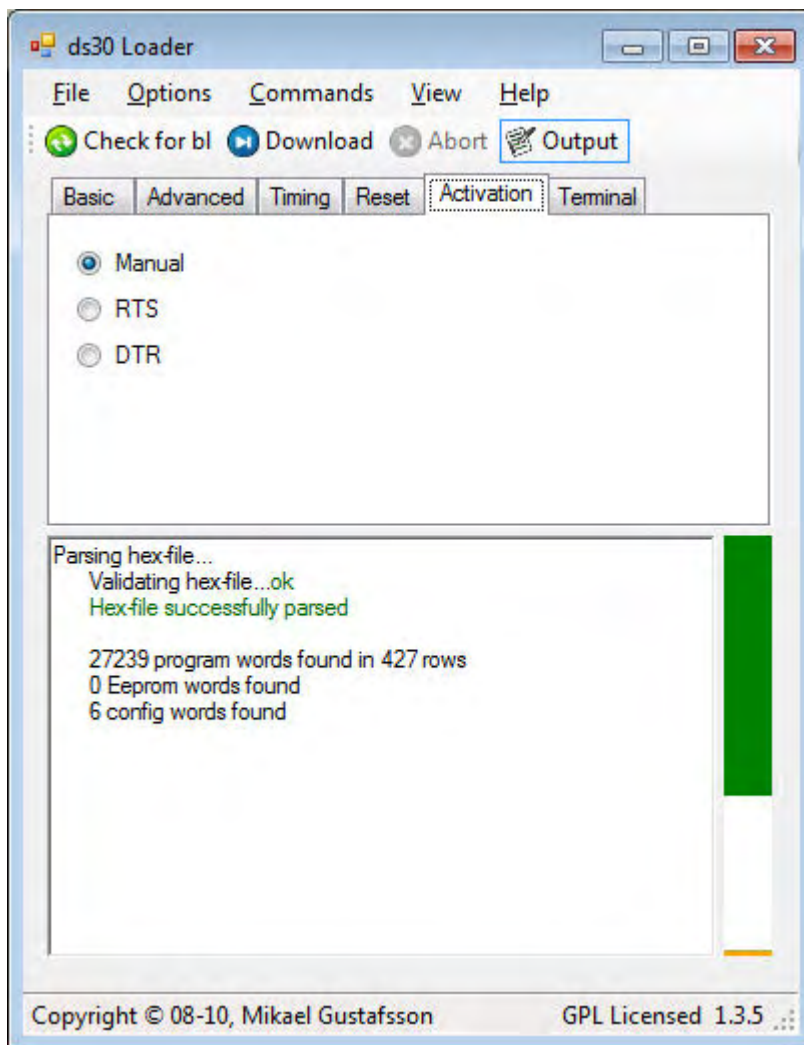
Timing Tab Settings:



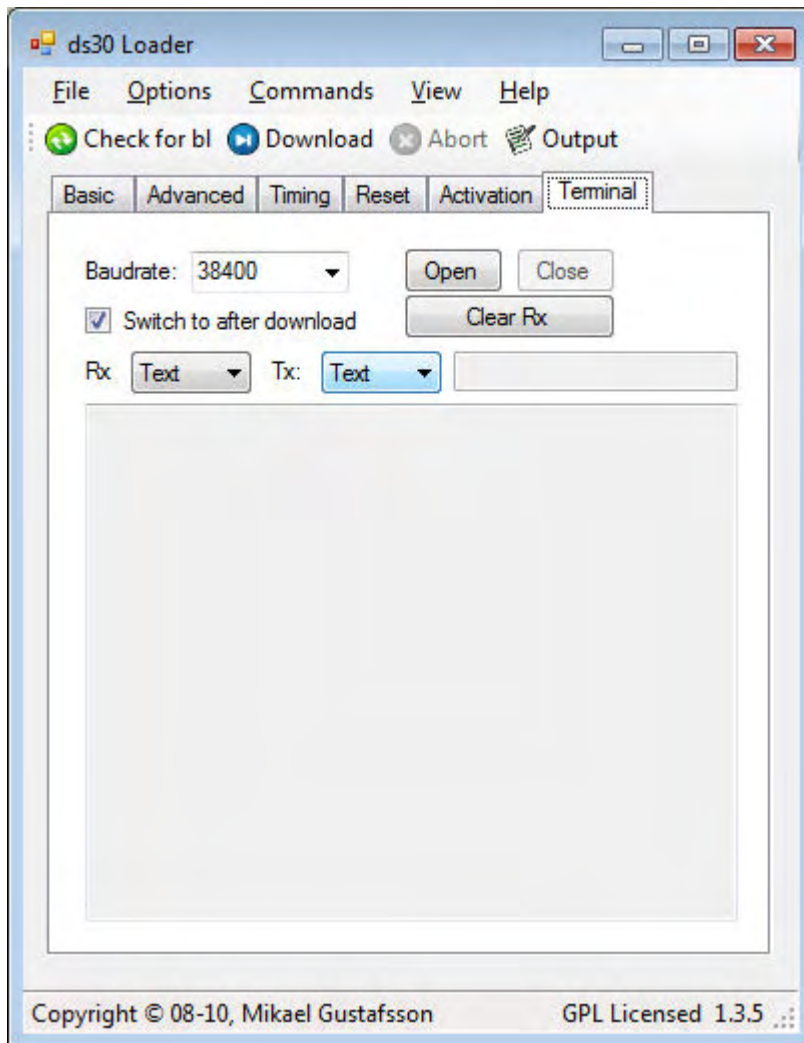
Reset Tab Settings:



Activation tab settings:



Terminal Tab Settings:



Once you have configured the program as shown, you should download the latest firmware file for the DOSD+. The firmware is available at this link:

<http://forum.tsebi.com/viewtopic.php?f=7&t=67>

The firmware file is attached. Download that to a directory of your choice and on the Basic tab, point to that firmware file in the "Hex-file" selector.

Note: The baudrates on the RESET and TERMINAL tabs should be 115200 instead of 38400 from version 7.5BETA14 onwards.

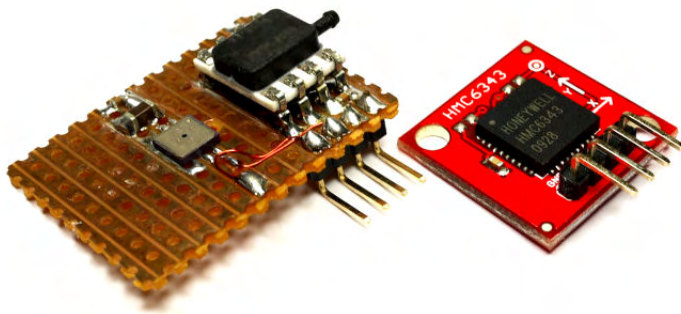
Now, you normally want to click on the "Download" button on the top of the dialog window, and then power up the OSD for the upgrade to work. So make sure everything is connected. Click "Download", then power up OSD board. The green progress bar at the top right of the window should indicate how far the upgrade has proceeded. Having said that, if the system is configured and working properly, you should be able to click "Download" anytime after the OSD has booted up properly.

Bear in mind that only one application can make use of a COM port at any one time. So if you have a terminal program open to the COM port in question, the firmware uploader will be unable to use that same port and thus uploading will fail. As such, it is best to close all other terminal programs before firing up the uploader utility.

Supported I2C Peripherals

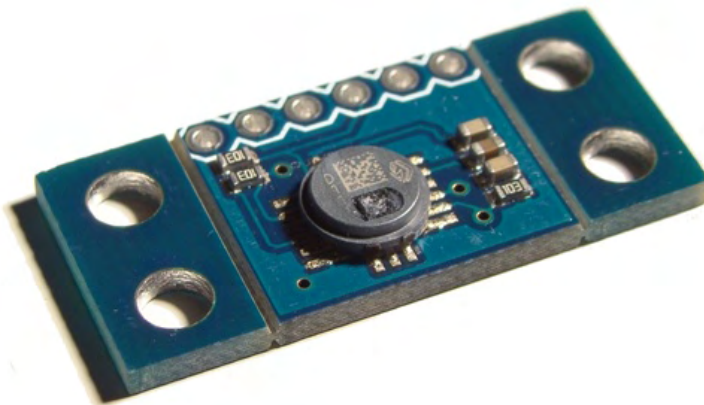
The I2C Peripheral header at the end of the board can be used to attach various types of sensors to the DOSD+. This includes barometric altitude sensors, true air speed sensor, tilt compensated magnetic compass, temperature sensor, IMU, and so on. When supported peripherals are attached, the DOSD+ will automatically detect and recognize them at boot-up and will make use of them (subject to user's menu settings). These peripherals can be daisy-chained so that several of them can sit on the same bus and operate at the same time. As we go along, we hope to add more supported peripherals to our list.

True air-speed, barometric altitude, temperature combo and tilt-compensated magnetic compass



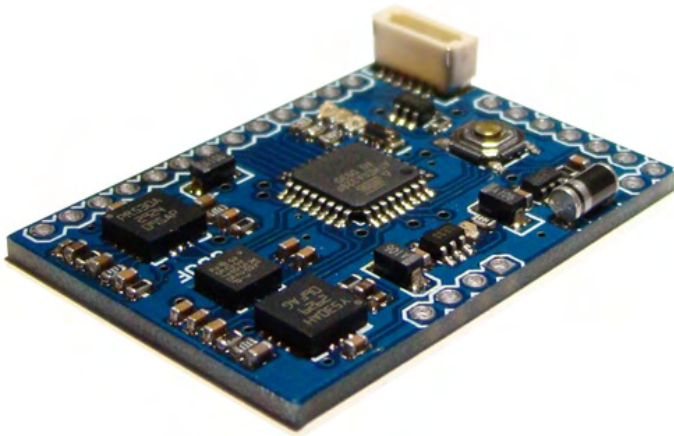
http://www.sparkfun.com/commerce/product_info.php?products_id=8656
http://www.sparkfun.com/commerce/product_info.php?products_id=9694

High-resolution barometric and temperature sensor



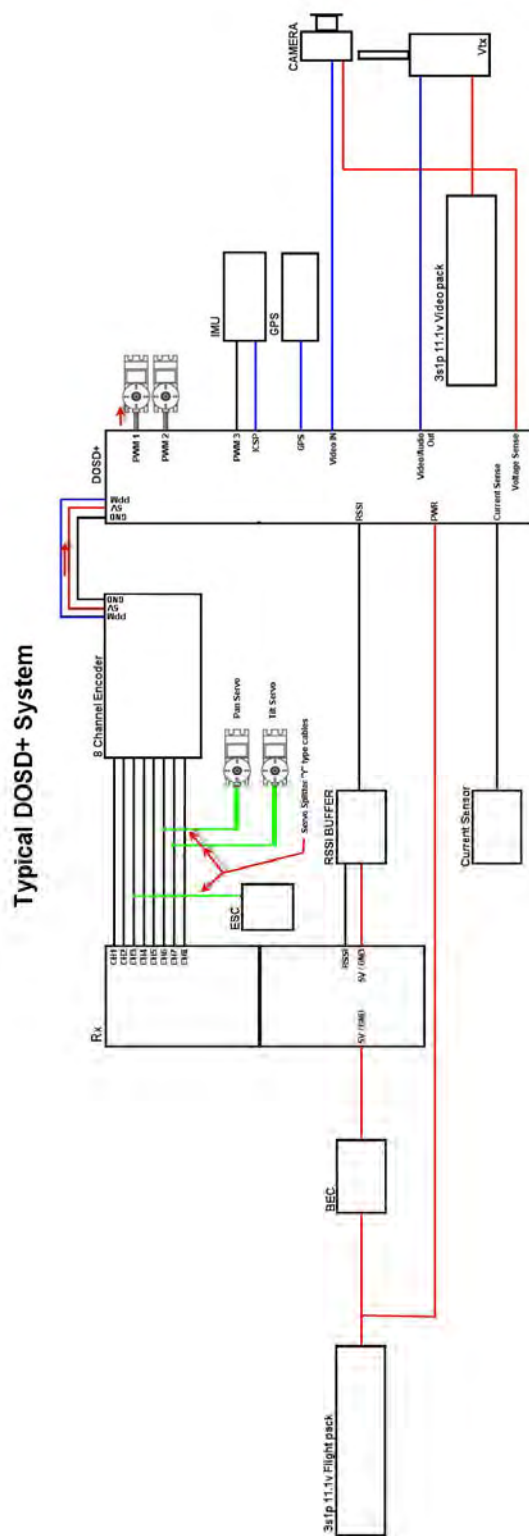
http://store.diydrones.com/product_p/br-scp1000d11-01.htm

ArduIMU+ V2



<https://store.diydrones.com/ProductDetails.asp?ProductCode=KT-ArduIMU-20>

Block Connection Diagram of Typical DOSD+ System:



Disclaimer of Warranty

You understand and agree that your use of this documented procedure is entirely at your own risk and is provided "As Is" and "As Available". Dragon Labs does not make any express or implied warranties, endorsements or representations whatsoever as to the operation of, or usability of this information, content, materials, and/or products. This shall include, but not be limited to, implied warranties of merchantability and fitness for a particular purpose and non-infringement, and warranties that access to or use of the service will be uninterrupted or error-free or that defects in the document will be corrected.

Limitation of Liability

You understand and agree that Dragon Labs and any of its subsidiaries or affiliates shall in no event be liable for any direct, indirect, incidental, consequential, or exemplary damages. This shall include, but not be limited to damages for loss of profits, business interruption, business reputation or goodwill, loss of programs or information or other intangible loss arising out of the use of or the inability to use the service, or information, or any permanent or temporary cessation of such service or access to information, or the deletion or corruption of any content or information, or the failure to store any content or information. The above limitation shall apply whether or not Dragon Labs has been advised of or should have been aware of the possibility of such damages. In jurisdictions where the exclusion or limitation of liability for consequential or incidental damages is not allowed the liability of Dragon Labs is limited to the greatest extent permitted by law.