USER MANUAL

PPM ENCODER BOARD

(ATMEGA168)

VERSION 4.2 (Firmware)

Created by Hendrix and Moa

3rd August 2009

Table of Contents

INTRODUCTION

This software is FREE. Use it at your own risk.
This software and application comes without any guarantee of any kind.
You can modify the code and everything else as you please.

The whole project is based around the ATMEL ATMEGA168 AVR processor and all timing is done within interrupts so accuracy and stability is optimized
It is now also possible to select the PPM waveform shift, negative or positive.

Please use the above mentioned processor otherwise you may need to change/modify the source files of the project. The source code knows what chip you use from the device selected in the makefile.

Have in mind that the Paparazzi autopilot is a very complex system and you must pay attention to details

OVERVIEW

You can use the throttle channel or any other channel to put the Paparazzi autopilot into "AUTO2" or “HOME” mode depending on the airframe settings.
This is effected by the PPM Encoder shutting down the PPM signal.

Basically you can connect any Rx channel to any PPM Encoder channel with the following provisions:

The RC receiver's throttle channel must be connected to channel 3 of the PPM Encoder as it is used as a rc signal lost indicator if the setup procedure has not be performed - otherwise any channel can be used if you execute the setup procedure

You can set the point where you want the PPM output to be turned off using the setup procedure else the default is that any servo pulse above ~2025 microseconds will shut down the PPM output thus force the Paparazzi autopilot into the "AUTO2" or “HOME” mode.

Make sure that your throttle channel's servo pulse does not exceed the PPM shut down threshold during normal operation by adjusting the EPA or whatever you call the servo travel limiter.
Only during loss of the RC signal when the receiver enters the failsafe mode should the throttle channel, or any other channel used as a Tx signal lost indication, be allowed to exceed this threshold (say 2100 microseconds).

The PPM board will cut the PPM output when the receiver goes into failsafe and set the throttle channel or any other channel if the setup procedure has been performed, to the extreme position.
This is done in order to simulate a traditional PPM receiver like the ones the Paparazzi designers used.

The extreme channel value used for indicating the signal loss to the PPM Encoder is not passed to the Paparazzi autopilot since the PPM output will be cut off when this extreme servo position is detected.

If the PPM board is set to continue to produce a PPM output, by not setting up the receiver to output this extreme servo position upon Tx signal loss means that your only option is to set the failsafe value of the receiver “MODE” channel to ”AUTO2” so the plane can enter the “AUTO2” mode
This method is **not recommended** for the following reasons:

> If, upon loss of signal, all channels stay legitimate and the “MODE” channel indicates “AUTO2” then Paparazzi will enter “AUTO2” mode. Paparazzi will not be aware of the signal loss condition and will believe the pilot has commanded the AUTO2 mode. Should the signal be reestablished with the Tx switch still in “Manual” mode then Paparazzi will faithfully switch to “MANUAL” mode with no pilot interaction.
>
> Paparazzi, by design, manages the mode switching if it sees a loss of signal but with this configuration, the PPM Encoder hides the loss of signal so Paparazzi cannot manage mode switching safely.
>
> Also be aware that some receivers failsafe due to low battery voltage (e.g.Futaba 7C) which can be reset (for 30 seconds) by selecting minimum throttle. Care must be exercised not to zero the throttle accidentally or purposely without considering the “MODE” switch. position. Again you could be put in command of the aircraft by a method other than using the “MODE” switch.

There are some receivers made by Futaba that output their servo pulses in a peculiar way.
One known Futaba Rx outputs the servo channels in the following order: 7,6,1,2,3,4,5
with channel 6 overlapping with channel 1. That should not cause any problem to the PPM encoder

Sometimes you may notice on some 2.4 Ghz systems that the ppm encoder takes some time (1-3 seconds) to produce a PPM output.
This is because some 2.4 Ghz systems need to perform a hand shaking between the Transmitter and the receiver before they start producing a pulse at their throttle channel.

FIRMWARE VERSIONS
RECEIVER TYPES AND THE PPM ENCODER BEHAVIOR

Version 3.8 firmware / Normal PPM receiver
The PPM Encoder will stop producing any PPM pulses when the Tx signal is lost thus setting the autopilot to "HOME" or "AUTO2" mode depending on the airframe settings.

Version 3.8 firmware / Receiver with failsafe settings on all channels
The user must set the failsafe for the "MODE" channel to 2000 microseconds so the autopilot can enter the "AUTO2" only mode when signal is lost.
This firmware is old and you should upgrade to version 4.00.

Version 3.9 firmware / Normal PPM Receiver
The PPM Encoder will stop producing any PPM pulses thus setting the autopilot the autopilot to "HOME" or "AUTO2" mode depending on the airframe settings.

Version 3.9 firmware / Receiver with failsafe settings on all channels or only the throttle channel.
The user must make sure that the throttle channel will go above 2020 microseconds when the receiver looses the Tx signal but not during normal operation so when the receiver goes in to failsafe (loss of Tx signal) the PPM Encoder will cut the PPM output thus set the autopilot to "HOME" or "AUTO2" mode depending on the airframe settings.

Of course he can leave the throttle channel's failsafe setting to below 2000 microseconds and set the "MODE" channel's failsafe setting to 2000 microseconds thus force the autopilot in to "AUTO2" but this is not recommended.

Version 4.x firmware / Receiver with failsafe settings on all channels or only the throttle channel.
The user must make sure that the throttle channel will go above 2025 microseconds when the receiver looses the Tx signal but not during normal operation so when the receiver goes into failsafe (loss of Tx signal) the PPM Encoder will cut the PPM output thus set the autopilot to "HOME" or "AUTO2" mode depending on the airframe settings.

In this version the user can also use a different than the throttle channel as a Tx lost indicator and also setup the point where the PPM output will by cut off by using a simple setup procedure.

Of course he can leave the throttle channel's failsafe setting to below 2000 microseconds and set the "MODE" channel's failsafe setting to 2000 microseconds thus force the autopilot in to "AUTO2" but this is not recommended.

UNSUITABLE RECEIVERS

A receiver with no failsafe capabilities at least on the throttle channel (ch3) or also does not cut the servo pulses after a Tx signal loss but keeps the last good values is NOT SUITABLE FOR USE WITH THE PPM ENCODER.

IMPORTANT THINGS TO WATCH FOR:

JP1 Jumper

The jumper JP1 has been included to provide some flexibility in the way you power the receiver, servos and PPM Encoder.

Often the power for the receiver, servos and PPM Encoder will come from the autopilot board. In this case JP1 should be in place.

If the receiver and any servos directly connected to the receiver get power from a separate/additional power source and the autopilot +5V is also connected to the PPM Encoder via the PPM connector - THEN YOU MUST REMOVE jumper JP1.
This will isolate the autopilot power supply from the separate receiver power supply.
Failure to remove JP1 will likely damage the autopilot's +5v switching regulator!

	+5V provided from autopilot	+5V Not provided from autopilot
Power supplied from Receiver and Servos	Remove JP1	**JP1 in place**
Power NOT supplied from Receiver and Servos	**JP1 in place**	

For those that might make the PPM Encoder from scratch:

1. Before operating the PPM encoder, remember to **clear** the check mark for the

"CKDIV8" fuse. This is the only fuse that needs to be modified on a new ATMEGA168.
The ATMEGA168 is shipped with the internal oscillator selected as the clock source but with the "CKDIV8" fuse checked it is running at 1Mhz and not the 8Mhz we require.

2. If you decide to also use the bootloader make sure that you also check these three fuses:

   1) BOOTRST, 2)BOOTSZ0, 3)BOOTSZ1.

   Those 3 fuses tell the mcu to move the reset vector to the bootloader address and to reserve all 2048 bytes of bootloader memory space.

Failsafe considerations

1. When a receiver has failsafe mode this means that the servo pulses will not stop when the Tx signal is lost thus the encoder will always see valid servo signals

2. If you use a receiver with failsafe then remember to set the receiver's failsafe value for the throttle to above 2025 microseconds (2050 for example) so the PPM Encoder can simulate a traditional PPM receiver by shutting down the PPM signal.
   You can alter the 2025 microsecond limit by executing the setup procedure.

   Alternatively (not recommended, see above) set the "MODE" channel failsafe value to ~2000 microseconds so the autopilot can go to "AUTO2" mode when the received RC signal is lost or set.

3. If the receiver failsafe of the "MODE" channel is left at 1000 or 1500 microseconds and the throttle channel's failsafe setting does not exceed the 2025 microsecond limit then the aircraft is not protected and you will likely crash and loose the plane for the following reasons.

   a) Paparazzi will stay in Manual mode because the MODE channel still commands this.
   b) PPM Encoder will maintain a good PPM output and so Paparazzi will not enter AUTO2 or HOME mode as the signal is still considered good.

4. If the receiver only provides "hold" on the last good servo signal received it is **not suitable** for use with this encoder as it will prohibit the autopilot entering the "AUTO2" mode or initiate a lost signal condition by ceasing to pass a PPM signal to the autopilot.

   This circuit was designed for use with:

   a) Good quality receivers with full failsafe capabilities those days like the Multiplex IPD, the PCM range of receivers or the newer 2.4GHz ones.
   b) Receivers that can be set to simply turn off the servo pulses when the Tx signal is lost
   c) Simple PPM receivers that cannot maintain good servo pulses when reception is bad.

The Paparazzi “radio.xml" file

1. You must use a "radio.xml" file that contains 8 channels no matter how many channels your receiver has.

2. The PPM Encoder now dictates the number of channels and the PPM timing.

3. The reset_max value inside the radio.xml file should be more than 13000 microseconds.
   Here is an example of a working "radio.xml" file for my Royal Evo 12 and my MC4000 with the "MODE" channel being channel 7:

```
<!DOCTYPE radio SYSTEM "radio.dtd">
<radio name="RE12" data_min="800" data_max="2200" sync_min="5000" sync_max="15000">
<channel ctl="C" function="ROLL" min="950" neutral="1500" max="2050" average="0"/>
<channel ctl="B" function="PITCH" min="950" neutral="1500" max="2050" average="0"/>
<channel ctl="D" function="THROTTLE" min="950" neutral="950" max="2050" average="0"/>
<channel ctl="A" function="YAW" min="950" neutral="1500" max="2050" average="0"/>
<channel ctl="E" function="FLAPS" min="950" neutral="1500" max="2050" average="0"/>
<channel ctl="F" function="GAIN1" min="950" neutral="1500" max="2050" average="1"/>
<channel ctl="G" function="MODE" min="950" neutral="1500" max="2050" average="1"/>
<channel ctl="H" function="GAIN2" min="950" neutral="1500" max="2050" average="1"/>
</radio>
```

If you want the "MODE" channel to be channel 5 just change the order of the lines:

```
<!DOCTYPE radio SYSTEM "radio.dtd">
<radio name="RE12" data_min="800" data_max="2200" sync_min="5000" sync_max="15000">
<channel ctl="C" function="ROLL" min="950" neutral="1500" max="2050" average="0"/>
<channel ctl="B" function="PITCH" min="950" neutral="1500" max="2050" average="0"/>
<channel ctl="D" function="THROTTLE" min="950" neutral="950" max="2050" average="0"/>
<channel ctl="A" function="YAW" min="950" neutral="1500" max="2050" average="0"/>
<channel ctl="G" function="MODE" min="950" neutral="1500" max="2050" average="1"/>
<channel ctl="E" function="FLAPS" min="950" neutral="1500" max="2050" average="0"/>
<channel ctl="F" function="GAIN1" min="950" neutral="1500" max="2050" average="1"/>
<channel ctl="H" function="GAIN2" min="950" neutral="1500" max="2050" average="1"/>
</radio>
```

4. If you want you can change the channel assignment of your Tx or Rx at anytime. Just make sure that the channel functions, as they are connected at the PPM Encoder, match the order in the radio.xml file.

   An example:
   Say you have an eight channel receiver that has channels 1,2,3,4,8 connected to the PPM Encoder with channel 8 being the "MODE" channel and suddenly you want to swap the receiver with a 6 channel one. You now don’t have a receiver channel 8 but the solution is very simple.

   - You must connect channels 1,2,3,4 as before
   - Connect any one of the remaining channels (say Rx channel 5) to channel 8 of the PPM encoder.

- You next assign the "MODE" switch on your transmitter to channel 5.
- You should not change anything in the radio.xml file as servo position 8 on the PPM Encoder was already known to be the MODE channel by Paparazzi.

The PPM Encoder dictates the channel order you see in the "radio.xml" file and not the receiver servo channels. If, as in the above example, channel 5 of the receiver is the "MODE" channel and this channel is connected to servo connector/position 8 of the PPM encoder, then Paparazzi autopilot will see the "MODE" channel as being channel 8 no matter what channel you use at your transmitter or receiver side.

5. The PPM Encoder is built to output a Futaba style PPM signal which differs from that of all other brands including JR. Paparazzi must be set to expect a Futaba style PPM signal.

   - Locate the following line in the airframe file (correct)
     "ap.CFLAGS += -DRADIO_CONTROL -DRADIO_CONTROL_TYPE=RC_FUTABA"

     If your definition reads as follows (incorrect)
     "ap.CFLAGS += -DRADIO_CONTROL -DRADIO_CONTROL_TYPE=RC_JR "
     modify it to be the same as the correct line above with FUTABA in place of JR.

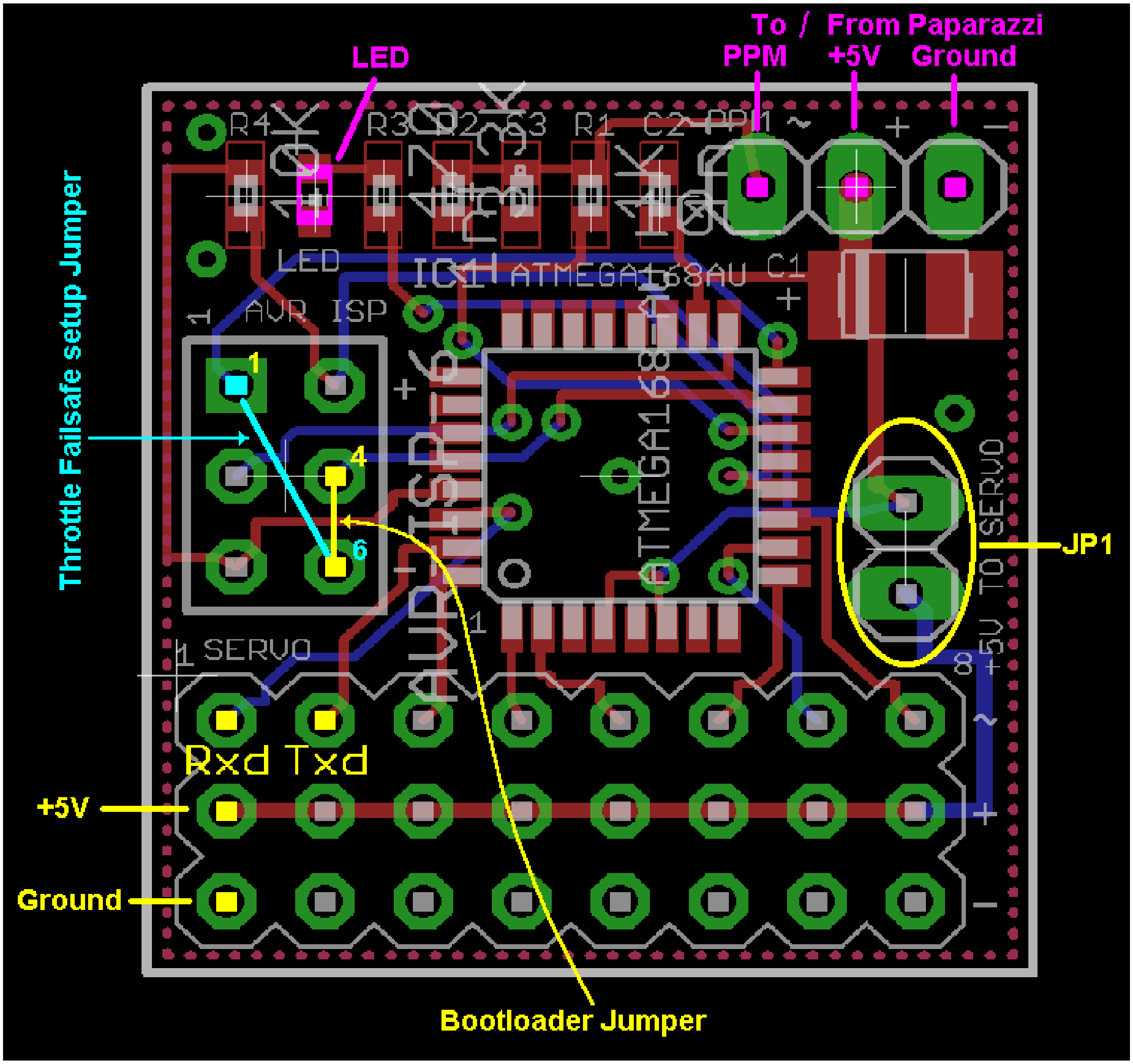
     Alternatively you can set the definition "RC_PPM_OUTPUT_TYPE" to 1.

   It does not matter what make of receiver or transmitter you have since the PPM Encoder controls all parameters of the PPM waveform.

HOW TO TEST THE BOARD

Here we test that the PPM Encoder is basically working.

Refer to the image of the board for the points referred to in this "How to test the Board" procedure". They are highlighted in **Magenta** .



For this test you may initially leave the "MODE" channel of the receiver disconnected from the PPM Encoder. It can be connected later during the procedure.

SAFETY FIRST

- Start with all airplane and Tx power off
- Ensure the JP1 jumper is set correctly as per "IMPORTANT THINGS TO WATCH FOR"

REMOVE THE PROPELLER OR DICONNECT THE MOTOR CABLES !!!

- Connect the encoder to the receiver and autopilot.
  Please verify the integrity of all connections to the autopilot and from the receiver to the PPM Encoder board. The PPM output pin header has the ground near the PCB corner, the positive 5 volts to the middle pin and the PPM output is the pin near the 402 sized components (C2, R1, C3 etc.).

- Power up the airplane with the Tx powered off.
  The LED should be on constantly without blinking.

- Power On the RC transmitter
  The LED should start to blink with a rate 3 times per second (that is 3Hz)
  The LED blinking at 3Hz means that the internal timing is correct, that the board sees all of the incoming receiver servo pulses and that the PPM waveform generator is on and working.

- Confirm that the servos move when you move the sticks.

- Next disconnect a channel from the receiver (just unplug it) and see if the led blinks slower now at 1 time per second (1Hz).
  That means that one or more of the servo signals are lost and that the PPM output is off.

- Now remove power from the plane, connect the "MODE" channel from the RC receiver to the PPM Encoder board if you haven't done it already.

- Ensure you have assigned the Transmitter Mode switch to the appropriate channel.

- Power on the RC transmitter again first and then the airplane.
  The PPM encoder's led should blink fast at a rate of ~3Hz and you should be able to switch between "MANUAL', "AUTO1' and "AUTO2" using the RC transmitter's switch that you have assigned to be the mode change switch.

You can tell in what mode you are in by looking at the GCS screen.

VERY IMPORTANT

The following test ensures the PPM Encoder stops the PPM signal when a failsafe receiver goes into failsafe.
(You can perform this test later if you are going to use the THROTTLE FAILSAFE SETUP procedure)

- Switch to “MANUAL” mode

- Move the throttle stick to:
    a. Minimum if you have a Futaba transmitter
    b. Maximum for all other transmitter brands
    c. To the side you selected during the **“THROTTLE FAILSAFE SETUP”** if you have performed it.

- Adjust the EPA of this throttle stick's side to maximum so the led changes it's blinking rate from 3Hz to 1Hz when you move the throttle stick to the extreme min or max position depending on what brand you have.

- Set this position as the receiver failsafe position for the throttle channel.
  You will be able to see the led changing it's flashing rate when you move the throttle stick.

- **Now return the EPA to normal in order to make sure that this extreme throttle servo pulse cannot occur during normal operation but only when the receiver goes in to failsafe**.

The above steps are required in order to make the PPM Encoder simulate a traditional PPM receiver. The PPM board will cut the PPM output when the receiver goes into failsafe and moves the throttle channel to the extreme position (past 2020 microseconds) exactly like a traditional PPM receiver would.

HOW TO ENTER THE THROTTLE FAILSAFE SETUP

Here we set up the Throttle Failsafe value. The PPM Encoder is being taught a throttle channel pulse width that, beyond which, will indicate that the Rx is in failsafe mode.

Refer to the image of the board for the points referred to in this procedure".
They are highlighted in **Cyan**

If you want to set a channel other than the throttle channel to be the Tx signal lost indicator then you must disconnect all other receiver channels except the desired one from the PPM Encoder and then proceed to complete the setup procedure described below.

If only one channel of the receiver is connected to the PPM Encoder during power up for the setup procedure, then this channel is selected as the Tx signal lost indicator and the PPM shut down threshold can be adjusted for this channel. It can be any channel you want.

If more than one channel of the receiver is connected to the PPM Encoder during power up for the setup procedure then the default channel will be used. This is the Throttle Channel (#3 of the PPM encoder) with a threshold of 2025 microseconds.

This is done in order to enable the user to execute the setup procedure with out disconnecting
the ppm encoder from the system if the default setting of using channel 3 (the throttle channel)
is acceptable.

SAFETY FIRST

- Remove power from the plane
- Ensure the JP1 jumper is set correctly as per "IMPORTANT THINGS TO WATCH FOR"

REMOVE THE PROPELLER OR DICONNECT THE MOTOR CABLES !!!

- Ground pin 1 of the ISP header (Connect Pin 1 to Pin 6 of the ISP header) using a small piece of wire with 2 female pin connectors.

- Stop! ……Do not power up the plane and the PPM board just yet.

- Power up the transmitter

  Make sure that the throttle EPA for the throttle side you select in the next step is not set to maximum because you will need a little extra later on (about 7% more). The actual threshold will be set at 25 microseconds above the recorded value if the value recorded was above 1500 microseconds or 25 microseconds below the recorded value if the value recorded was below 1500 microseconds.

In other words the threshold where the PPM output is cut off will be set 25 microseconds above or below the recorded value depending which side of the throttle stick you have selected.

- Set the throttle stick to the end position you want, say at the min throttle position.
  This position will be set as the extreme value that can happen during normal operation.
  A bit beyond this position is the value you will subsequently set into the Receiver as the failsafe setting

- Then power up the PPM board by connecting the battery to the airplane.
  The LED on the PPM board will light solid and when the setup is finished it will start to flash with 1Hz rate and a duty cycle of:

  - 100ms ON – 900ms OFF if the setup is **successful** and with a rate of **1Hz**
  - 900ms ON – 100ms OFF duty cycle (**almost solid on**) if the **setup failed** (the opposite of when the setup was successful).

  If the setup failed (highly unlikely) then check the connections and repeat the setup procedure after a power cycle.

- If the setup was successful then remove power from the PPM board

- Remove the grounding connector from pin 1 of the ISP header.

- Power up again normally, you should see the led flash at 3Hz as usual.

- Test to see if the LED changes it's blinking frequency to 1Hz by adjusting the transmitter throttle EPA of the side you selected during setup to maximum and then moving the throttle stick to the end point you had selected during the setup procedure.
  The LED should change it's blinking frequency to 1Hz

- Move the throttle stick back towards the center again.
  The LED should change it's blinking frequency back to 3Hz

- Next set this extreme position (the LED should blink with 1 Hz rate) as the receiver failsafe position for the throttle channel.

- Now return the EPA to normal in order and make sure that this extreme throttle servo pulse cannot occur during normal operation. (3Hz blinking at all Throttle positions)

- Finally check that the PPM board changes the led frequency to 1Hz when you switch off the transmitter thus forcing the receiver to enter the failsafe mode.

The above steps are required in order to make the PPM Encoder simulate a traditional PPM receiver. The PPM board will cut the PPM output exactly like a traditional PPM receiver would

when a failsafe capable receiver goes into failsafe and moves the throttle channel to the extreme position.

If you do not perform this setup procedure then the default value will be used which is to cut the PPM output when the throttle channel's servo pulse goes above 2025 microseconds.

You can change the default value defined in the “servo2PPM_settings.h” file to suit your needs but you will need to recompile the source code of the PPM Encoder.

ABOUT THE BOOTLOADER

The whole project is based around the ATMEL AVR processor and specifically the ATMEGA168.
Please use the above mentioned processor otherwise you may need to change/modify the source files of the project.
I have tested the bootloader with other processors and it worked well right out of the box but I can't guarantee anything.
If you want to use the bootloader (recommended so later you can update the firmware without using an AVR programmer) you can use the "app+bootloader.hex" file to flash both the servo2PPM and the bootloader code at once. Alternatively you can first flash the the PPM_encoder_ver_x_x.hex file with the "erase chip before flashing" option of the avrstudio flashing utility turned on and then flash the "bootloader.hex file but this time with the "erase chip before flashing" option turned off
(ie do not erase the chip before you flash the "bootloader.hex" file).

NOTE: Read carefully the later section from the Mega168 datasheet on Boot Lock Bits.

BEFORE ENTERING THE BOOTLOADER DISCONNECT EVERYTHING FROM THE PPM ENCODER BOARD !!!

The places of interest for the "Bootloader procedure" are highlighted in **Yellow** *on the board image*

- Make sure that **everything** (receiver, tiny or TWOG) **is disconnected** from the PPM Encoder board including power

- Place a jumper between pin 4 and pin 6 (ground) of the ISP header.

- Prepare the FTDI cable
  The +5 volts that will power the PPM Encoder board will come from the +5 volts of the FTDI cable so you need to make a cable that will connect the FTDI's +5v,Rxd,Txd and Ground to the PPM encoder's +5v, Rxd, Txd and Ground.

  The +5 volts and Ground of the FTDI cable are connected to any of the PPM Encoder board's servo input +5 volt and ground pins.

  The serial pins for connecting the FTDI cable to the PPM Encoder board are the signal pins of servos 1 and 2 . Servo1=Rxd, Servo2=Txd as depicted on the picture of the board.

  NOTE: these must be connected crossed to the FTDI cable
  ie. Servo 1 signal pin to FTDI Txd and servo 2 signal pin to FTDI Rxd.

- You can then reset the PPM Encoder board as needed by removing the jumper JP1 that is normally used to feed power to the servo header.

- Wait for ~2 seconds and replace the jumper again on JP1 in order to restore power to the board.

In order to upload the file needed, reset the PPM Encoder board by power cycling it making sure that

the connection with the terminal program like "HyperTerminal is already opened and connected at 9600 bps 8 bits, no parity, 1 stop bit, no flow control and carriage return (CR) as terminator of the line.
After power cycling (reset) the PPM Encoder board you should see the prompt "ATH=help >" on your screen.

- Type "ATWF" using capital or small or mixed letters and hit enter.
  Now the "C" character will start to fill the screen which means that you must start the X modem CRC transfer from your terminal program.
- Select the Xmodem type of file transfer, then select the file to be uploaded ("PPM_encoder_ver_x_x.bin") and click the send button or whatever your terminal program uses.
  Be carefull "Xmodem 1K" IS NOT THE SAME!!!
  Usually “Xmodem CRC” is referred as plain "Xmodem".

Then the transfer window of your program will show up telling you the transfer progress and after the transfer is hopefully complete and successful the window will automatically close and you will see the message “update successful” at the terminal 's screen.

The bootloader commands are:

ATWF = write flash.
ATWE = write Eeprom.
ATI = Display version information about the bootloader and the PPM Encoder firmware
ATRB = Read boot lock fuses.
ATRL = Read the low fuses.
ATRH = Read the high fuses.
ATRE = Read the extended fuses.
ATWB = Write or modify the boot lock fuses

For the PPM Encoder firmware version to be displayed using the “ATI” command the PPM Encoder firmware must run at least once in order to register it's version info into the EEPROM.
That means that if you want to be absolutely sure which version you are about to use after a firmware upgrade then

- power off
- remove the jumper from pins 4 and 6 of the ISP header,
- power on again so the PPM Encoder software can run (No need to connect anything to the PPM Encoder board)
- power off the board, install the jumper to pins 6 and 4 and power on again.
  Now the “ATI” command will show something like "Servo2PPM V4.00".

Pay particular attention to the ATWB command as it can render the PPM Encoder useless and then only a flash erase and programming of the board again will make the PPM Encoder work again.

The fuse bytes are presented with binary form for example 00000010 = 2

If you attempt to modify the boot lock fuses you need to issue a command like this:
ATWB11111011 + ENTER

The above command will program the BLB02 fuse.

I recommend to leave the boot lock bits as they are and not play with the “ATWB” command at all as the bootloader code is already protecting the bootloader's flash memory space.

SETTING THE BOOT LOCK BITS

USING THE ATWLxxxxxx BOOTLOADER COMMAND.

BOOT LOCK BIT programming (BLB12, BLB11, BLB02, BLB01, LB2 and LB1)

To set the Boot Loader Lock bits and General Lock bits using the bootloader first examine the table below.

Enter the command ATWLxxxxxx and press enter.
The xxxxxx are the binary presentation of lockbits 5,4,3,2,1 and 0
Bits 7 and 6 are omitted since you cannot program them.

Examples

ATWL11011111
= the boot lock bit 5 will be programmed (BLB12)

ATWL1111101
= the boot lock bit 1 will be programmed (LB2)

ATWL1111110
= the boot lock bit 0 will be programmed (LB1)

So if bits 5..0 in R0 are cleared (zero), the corresponding Boot Lock bit or General Lock bit will be programmed.

“1” means not programmed, “0” means programmed

Be careful with the boot lock bits.

BOOT LOCK PROTECTION MODE TABLES

Below are two tables with the functionality of the boot lock bits of the ATMEGA168.

Boot Lock Bit0 Protection Modes (Application Section)(1)

Table 26-2.

BLB0 Mode	BLB02	BLB01	Protection
1	1	1	No restrictions for SPM or LPM accessing the Application section.
2	1	0	SPM is not allowed to write to the Application section.
3	0	0	SPM is not allowed to write to the Application section, and LPM executing from the Boot Loader section is not allowed to read from the Application section. If Interrupt Vectors are placed in the Boot Loader section, interrupts are disabled while executing from the Application section.
4	0	1	LPM executing from the Boot Loader section is not allowed to read from the Application section. If Interrupt Vectors are placed in the Boot Loader section, interrupts are disabled while executing from the Application section.

Boot Lock Bit1 Protection Modes (Boot Loader Section)(1)

Table 26-3.

BLB1 Mode	BLB12	BLB11	Protection
1	1	1	No restrictions for SPM or LPM accessing the Boot Loader section.
2	1	0	SPM is not allowed to write to the Boot Loader section.
3	0	0	SPM is not allowed to write to the Boot Loader section, and LPM executing from the Application section is not allowed to read from the Boot Loader section. If Interrupt Vectors are placed in the Application section, interrupts are disabled while executing from the Boot Loader section.
4	0	1	LPM executing from the Application section is not allowed to read from the Boot Loader section. If Interrupt Vectors are placed in the Application section, interrupts are disabled while executing from the Boot Loader section.

Note: 1. “1” means un-programmed, “0” means programmed

Have fun.
Chris