Single Channel Encoder

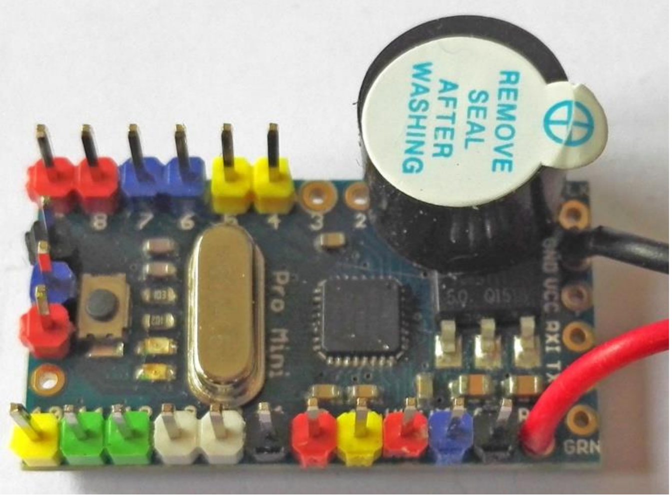


Figure 1. Single Channel Encoder based on Arduino Pro Mini

Introduction

The single channel encoder is designed to convert vintage single channel transmitters or new build transmitters to 2.4GHz. It's based on an Arduino Pro Mini and is one of the easiest encoders to build, as beside the power in, there is just the header pins and sounder to solder onto the Arduino circuit board.

It is fully configurable and offers enough features to be suitable for the majority of conversions, either with a push button or with a 3 position switch like a MacGregor Codamac.

A Short History of the Development

I decided to build a reproduction OS Pixie MK1 transmitter and needed a compact encoder for it. I wanted to do my own encoder, so chose to base it on an Arduino Pro Mini as it was the smallest Arduino commonly available at the time.

The encoder is designed for the way I like to use a transmitter, so it has servo reverse, servo travel, channel order to suit FrSky or OrangeRx modules, but most importantly a countdown timer, as I like to plan my landing before the motor stops!

The encoder is configured by fitting jumper links to the header pins. These are soldered on the Arduino "facing upwards" and 10 pins are "paired up", one as an input with the "pull up" enabled and the other pin could be programmed as an output with 0 volts. If a "jumper link" is fitted you would read a "0", if the jumper link is not fitted you read a "1". This is compact and it saves having to fit any configuration switches.

Most of the code is re-used from my multi-encoder and uses my own compound timing measurements from one of my Elmic Compact escapements (0.8 secs for a full escapement rotation). This may differ to other encoders that have a slightly faster rotation, but your thumb should tune in quite quickly.

Contents

Specification

- Based on an Arduino Pro Mini 5V 16MHz powered by a 2S Lipo.
- No separate circuit board required, just the Pro Mini with the headers soldered on facing upwards and the sounder soldered directly onto the Arduino.
- Very compact, only 35mm x 22mm (1.35" x 0.87") including the buzzer or 35mm x 18mm without (buzzer mounted remotely)
- Compound or Sequential operation.
- Compound 2 or 3 position blip throttle, with 1, 2 or 3 short beeps to indicate its position.
- Compound kick up (or down) elevator.
- Optional sequential beeps to indicate next direction (one beep for right next, two for left)
- Optional Countdown Timer Alarm from 30 secs to 7 minutes with reminder beeps at 3, 2 and 1 minute and 30, 20 and 5 seconds
- Optional "Pulsed Throttle" five seconds from the end of the countdown timer. The throttle is rapidly switched on-off-on-off-on giving an easy to notice alarm. It is really only suitable for electric power, due to the speed of switching.
- Optional Motor Cut after timer elapsed (default for sequential, but also available or compound)
- Motor start sequence for electric motors, so they don't inadvertently start
- Adjustable servo throw (end point adjust/travel volume) and servo reverse for all channels
- Throttle has adjustable closed (tick-over), mid and full throttle positions
- Throttle is ramped up and down between positions over 2 seconds (servo slow)
- Sub-trim for Rudder and Elevator
- Three position toggle switch (spring centre) for programming that can also be used for "automated compound" (right before left to mimic a MacGregor Codamac, Futaba FT-5E or Staveley Tonelock) or 2 channel reed operation (similar to a Graupner Varioton)
- The push button is transmitted (on channel 4) so that an original escapement can be used with a suitable switching device in the aircraft
- Either RETS (rudder, elevator, throttle and switch) or TRES output order to suit different 2.4GHz modules
- Inactivity alarm (if the single channel or 3 position switch is not operated within 3 minutes)
- Optional low battery voltage alarm (requires extra circuitry)

Choosing Which Arduino Pro Mini to Use

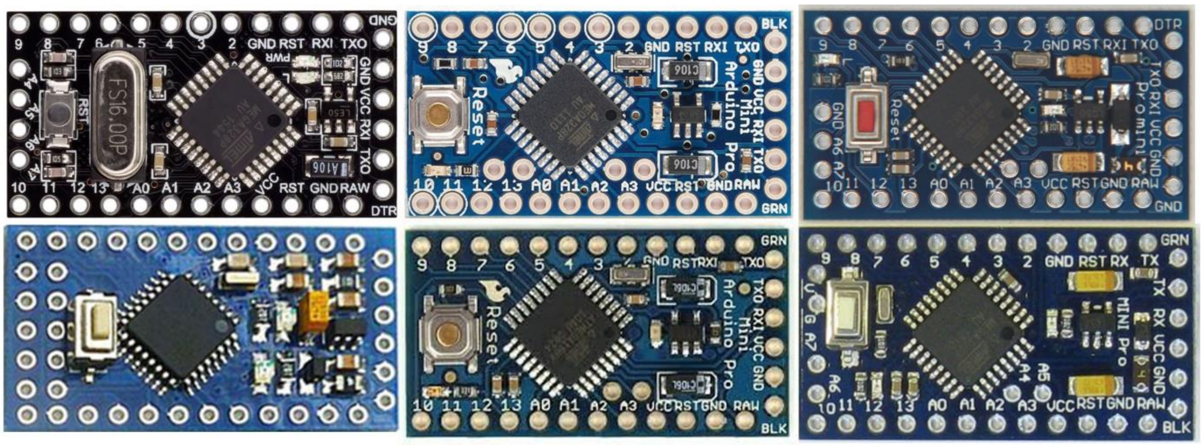

GND
9 8 7 6 5 4 3 2 GND RST RXI TXO
PWR
FS16.00P
RST
A4 A5 A6 A7
GND VCC RXI TXO
10 11 12 13 A0 A1 A2 A3 VCC RST GND RAW
DTR
BLK
9 8 7 6 5 4 3 2 GND RST RXI TXO
Reset
GND VCC RXI TXO
Arduino Pro Mini
10 11 12 13 A0 A1 A2 A3 VCC RST GND RAW
GRN
DTR
9 8 7 6 5 4 3 2 GND RST RXI TXO
Reset
GND A6 A7
TXO RXI VCC GND
Pro mini
10 11 12 13 A0 A1 A2 A3 VCC RST GND RAW
GND
GRN
9 8 7 6 5 4 3 2 GND RST RXI TXO
Reset
TXO RXI VCC GND
Arduino Pro Mini
10 11 12 13 A0 A1 A2 A3 VCC RST GND RAW
BLK
GRN
9 8 7 6 5 4 3 2 GND RST RX TX
A6 A7
A4 A5
TX RX VCC GND
MINI Pro
10 11 12 13 A0 A1 A2 A3 VCC RST GND RAW
BLK

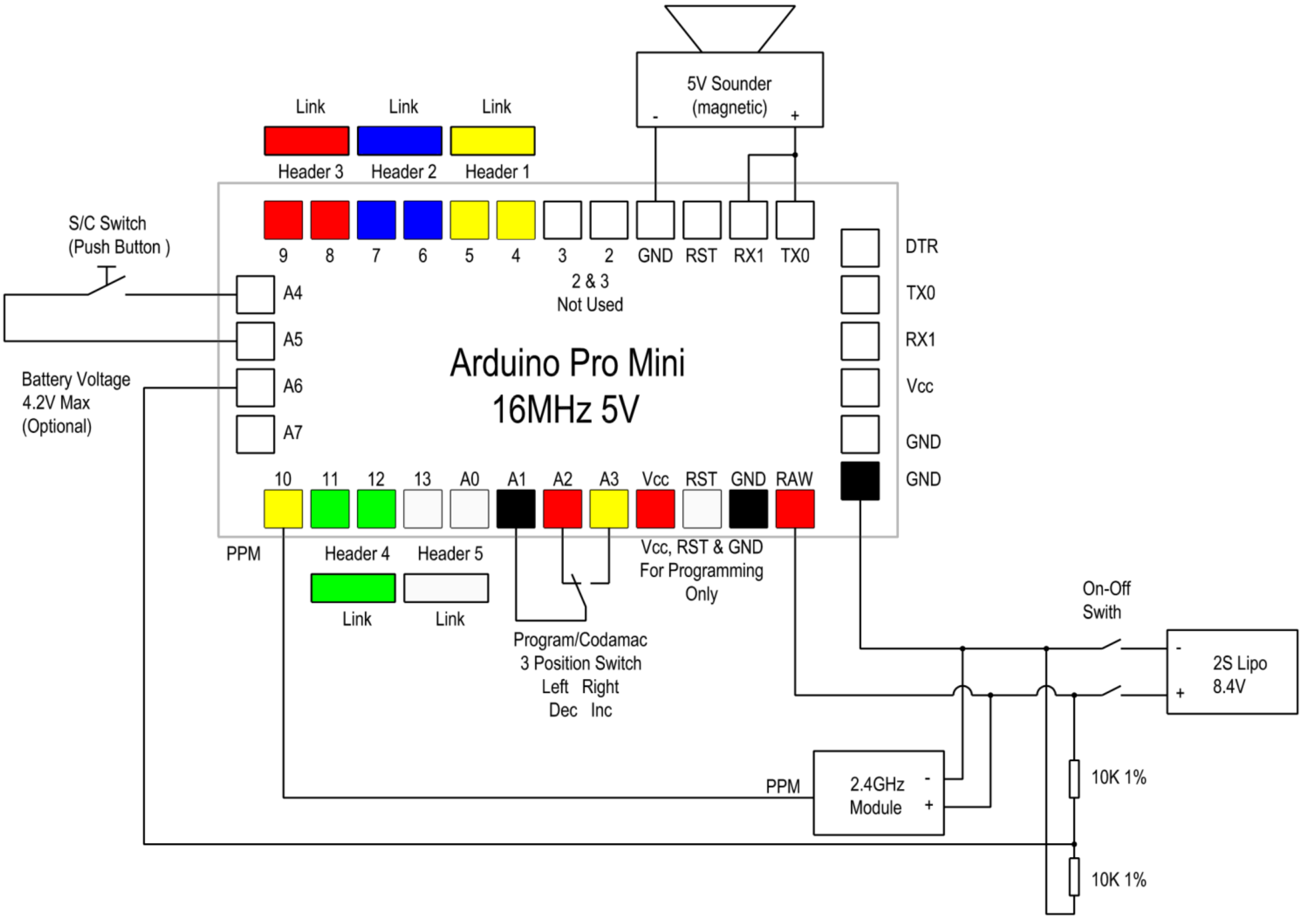
Figure 2. A Variety of Arduino Pro Mini with Different Board Layouts

We need to use an Arduino Pro Mini 5V 16MHz with A4 and A5 and if the low battery alarm is being used it needs A6 and A7 as well. The ATmega328P version is easier for most to use, as ones that say they have a ATmega168 actually have the modern ATmega168P controller which isn’t supported in the Arduino IDE by default. You can add the ATmega168P with the” boards manager”, but that is beyond the scope of these instructions.

There are lots of different layout variations with the Arduino Pro Mini, above shows six variations, but I have at least another six. The original version used an ATmega168 and didn’t have A4 or A5 (or A6 and A7 for that matter). Most versions are based on the ATmega328P or the ATmega168P and have at least A4 and A5 available and many have A6 and A7 as well. But these additional pins are located in different positions depending on the manufacturer layout. Some Pro Mini use a resonator, others use a crystal. The crystal does give slightly better stability, but in all honesty you won’t notice much difference with a single channel encoder. Also the TX0, RX1, Vcc, GND and DTR pins on the end are reversed on some boards, half with the DTR nearest the side with pins 0 – 9, the other half with it nearest RAW.

All the Arduino Pro Minis shown above are suitable. My preferred general layout is the one top left, as they have a crystal and the A4, A5, A6 and A7 pins are on the end (not in the middle of the board) where they are more easily accessible. My favourite for development is the one shown bottom left as it has the ISP (in system programmer) 6 pin header available, rather than having to make up a special lead to program them. But the most numerous versions appear to be the ones in the centre with the additional analogue inputs located “inside” the other pins.

Building the Encoder



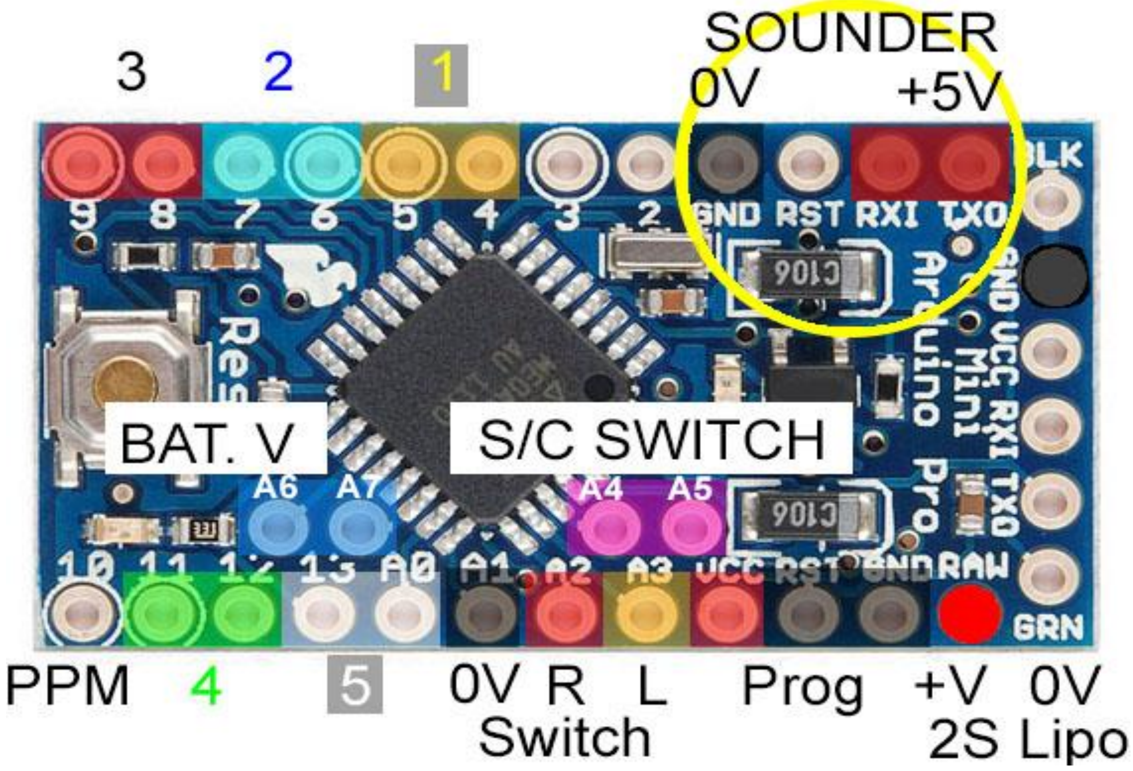


Figure 3. Typical Arduino Pro Mini with Pins Coloured to Show the Different Header Pins and Connections

In the schematic diagram and highlighted photo, two different types of layouts are shown, you will have to choose which matches you Pro Mini.

The Arduino header pins are soldered facing upwards and are used to configure the encoder by using header jumpers on the pins, together with a three position spring centred switch and the single channel pushbutton. For ease of identification I have used different colour header pins for the different pairs of header pins, though this isn't necessary and the ones that usually get supplied with the Arduino can be used. Through the rest of the instructions the following colours designate the following headers:

Header/Jumper 1	4 and 5	Yellow
Header/Jumper 2	6 and 7	Blue
Header/Jumper 3	8 and 9	Red
Header/Jumper 4	11 and 12	Green
Header/Jumper 5	13 and A0	White

The single channel pushbutton is connected to A4 and A5. It can be either hard wired into the Arduino or connected via 0.1" headers and either a servo plug or a JST connector with 0.1" centres.

The 3 Position Switch is connected to A1 (com), A2 (right/inc), A3 (left/dec). It is used for programming and can also be used for direction control if used with a MacGregor Codamac style transmitter. If the encoder is used with a traditional single channel transmitter, the three position switch can be connected with 0.1" headers and a servo lead so that it is removable when not in use. If it is going to be wired to a Codamac type switched joystick, then it can be hard wired or connected with a servo connector.

The PPM output is on pin 10 and can be hard wired or connected with a header and a suitable connector, such as a contact removed from a servo plug and covered with heat-shrink sleeving.

The low battery voltage is optional, indeed it has only been fitted to two of my prototype encoders to ensure it works OK. It isn't enabled by default, so if it is used it will have to be enabled in Program Mode. It requires two 10K 1% resistors in series as a potential divider when using a 2S lipo. The low battery level has been set at 7.2V (approx. 20% battery left) and it has been found that if 1% tolerance 10K resistors are used for the divider it should not need any calibration. You can however calibrate it for a different alarm voltage or if using wider tolerance resistors. There isn't room to mount the resistors on the Arduino, so one of the prototypes had the resistors wired to the back of the on-off switch, on the other they were mounted on a small piece of strip-board and mounted remotely with a 12V sounder.

The buzzer/sounder has its own section on the following page.

To mount the encoder in a transmitter, use strong double sided foam adhesive tape, such as servo mounting tape. If it is being mounted in a metal enclosure, it is best to first mount it on a thin plastic sheet, to ensure the solder joints on the bottom of the Arduino could not short out.

It has been tested with a FrSky DHT, an OrangeRx hack module and a Spektrum DX8, though it should work with most other encoders. The Arduino sketch/program would need minor alterations to work with a Spektrum DM9 however as it has different timings to most other 2.4GHz modules.

The Buzzer/Sounder

There are two options for the buzzer to be used, a 5V magnetic buzzer (often called an active buzzer/sounder) can be soldered directly onto the Arduino board or a 12V magnetic buzzer mounted remotely.

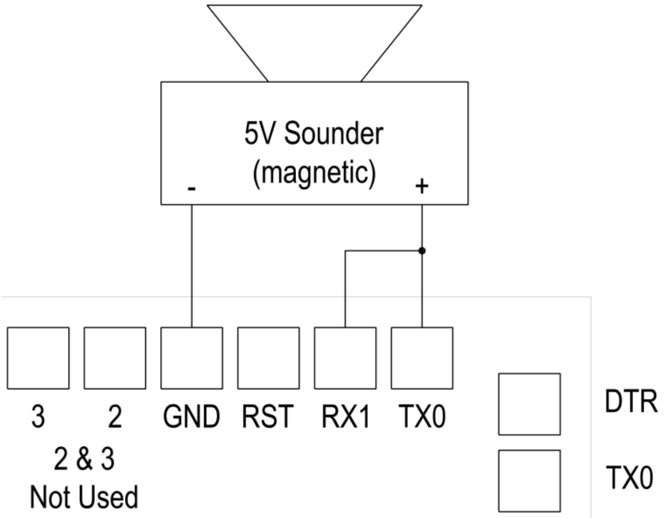


Figure 4. Connecting a 5V Buzzer

The 5V buzzer + connects to TX0 and is linked with RX1 and the 0V going onto the 0V next to pin2.

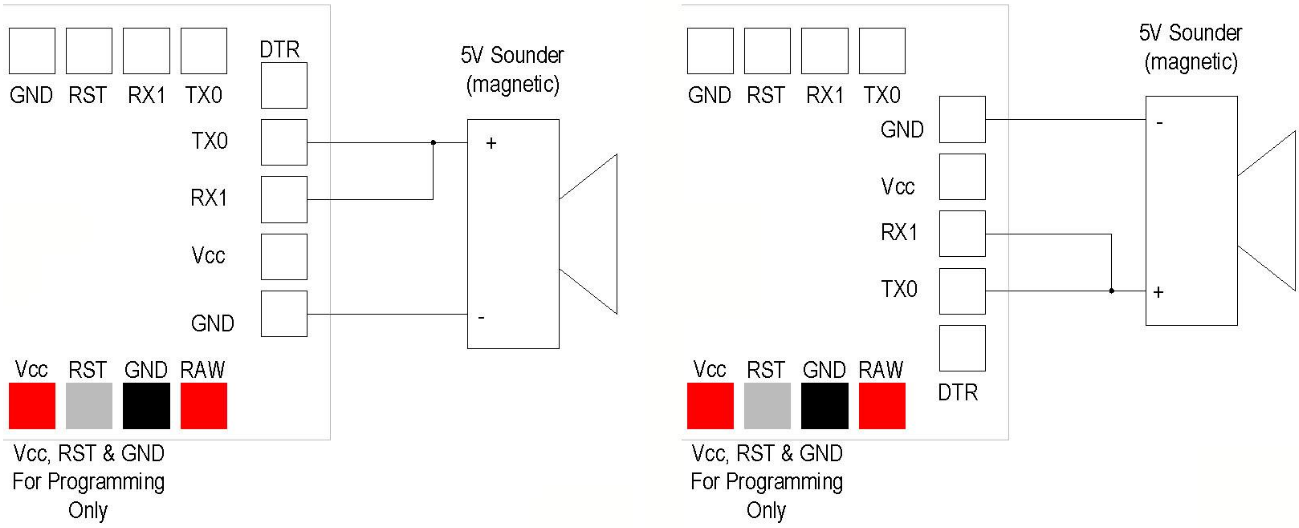


Figure 5. 5V Buzzer Connected to the End Connectors, with the two common layouts shown.

The TX0, RX1 and 0V pins are duplicated on the end of an Arduino Pro Mini, so you can fit a 5V sounder in the end position if prefered. Note that Arduino Pro Mini come with two different layouts of the end pins and on some boards come with the”end” GND pin at the bottom next to RAW and others it is next to TX0 at the top, both versions are shown above.

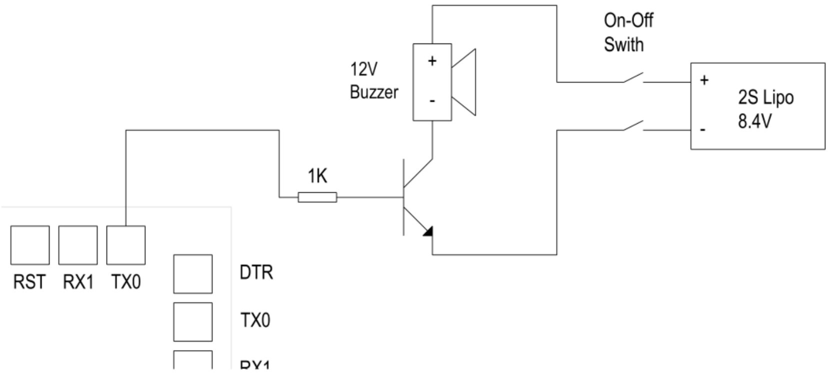


Figure 6. Connecting a 12V Buzzer

Alternatively a 12V magnetic buzzer can be fitted remotely and powered off the 2S lipo being switched by a transistor controlled by TX0 (there is no need to link TX0 and RX1 in this case). Any small npn transistor can be used, I used a 2N3904 as I bought (by mistake) a bulk lot of a 100 a few years back. There are pros and cons to the on-board and remotely mounted sounders.

Many people recommend not to switch a typical 5V buzzer directly by an Arduino output as the sounder takes 30mA current (near the maximum current an Arduino output can switch) and is slightly inductive, so putting a further stain the Arduino output. To alleviate this, two Arduino outputs are connected together to drive the sounder and are switched on/off simultaneously together (by writing directly to the port register) and in this way they share the current. On my scope the output looks OK and is far better than using a single output to directly switch a 5V sounder.

To test this, I left an encoder running for 6 days (using a power supply set to 8.4V) and was continuously beeping on/off (inactivity alarm). It was in a padded box to stop the worst of the noise, but after 6 days I got fed up with the noise in the workshop. I then repeated the test with another Arduino over 4 days, that has been used for regular flying and flight simulator practice over 18 months and the sounder is still working perfectly after hours of use. This small sample may not be representative, but it’s enough to convince me it’s OK to use.

But if the optional “Low Battery Voltage Alarm” is going to be fitted, then you will need a separate circuit board for the two 10K resistors as a potential divider and it is then sensible to fit a 12V buzzer and transistor to the circuit board and supply it directly with the 8.4V lipo battery supply.

Both buzzer options work well and whichever configuration is used, there is no alteration needed to the Arduino program. If space is tight, use the 5V buzzer on board, if there is plenty of space and the low battery voltage alarm is being used or you don’t like the idea of using a 5V sounder direct, fit a separate 12V buzzer.

Note the buzzer must not be a piezo or passive sounder, they will not work.

The sounder is used for a number of alerts and alarms, the countdown timer, sequential beeps to indicate the next rudder direction, compound throttle position, motor starting or timer starting, inactivity alarm and low battery alarm. During normal flying it can get confusing if there is more than one sounder alert and it is suggested that if say you are flying sequential with sequential beeps to indicate the next direction, then disable the countdown timer sounder and rely on the “pulse throttle” alert to indicate when the timer has elapsed.

Programming the Arduino

Arduino Pro Mini's are normally programmed via a FTDI USB to Serial adapter (or similar CH340 device) via the connectors at the end of the board (GND, Vcc, RX1, TX0 and DTR). This method has a number of disadvantages for an encoder.

- It takes far longer to boot up, typically a few seconds, as it has to wait a pre-determined time to see if there is any PC connected trying to upload a new program. This extra time can be enough to stop some 2.4GHz modules from working
- The bootloader communicates via the on-board UART on pins 0 and 1, so these pins would not be available for the sounder (the only pins where there is room to fit the sounder on-board).
- The bootloader take up an extra 2K of the 32K flash memory, though in this application it is not a problem as there is ample flash memory.

The only advantage of the FTDI USB to serial programming method is that it can be used for serial debugging so that various program variables can be monitored, indeed in the early stages of development this facility was used. But we are not debugging this encoder, I hope!!

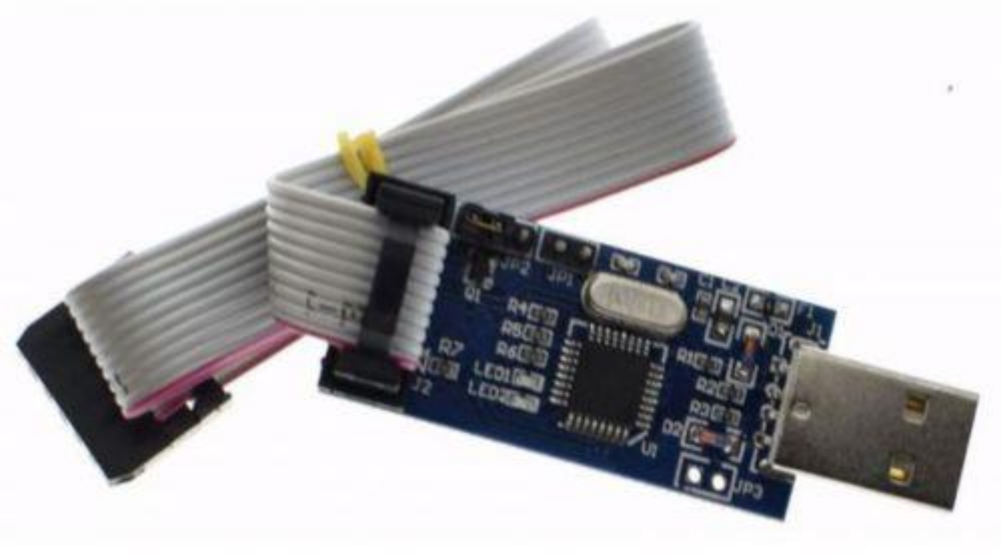


Figure 7. A Typical ISP Programmer, an USBasp

For our application an ISP (in system programmer) programmer is needed, something like an USBASP or an USBTinyISP, though an Arduino Uno can be programmed to be used as an ISP (just Google it for information). The ISP programs the on-board ATmega328P (or ATmega168) directly, not using a bootloader, so giving near instant boot times. And it frees up the D0 and D1 pins to be used by the sounder (the only place on the Arduino that there is room to fit the sounder without fouling other pins). And for larger programs it frees up the whole 32,678 bytes of flash memory. It is assumed below that an USBASP is being used, but it equally applies to any other ISP.

Making a Programming Lead

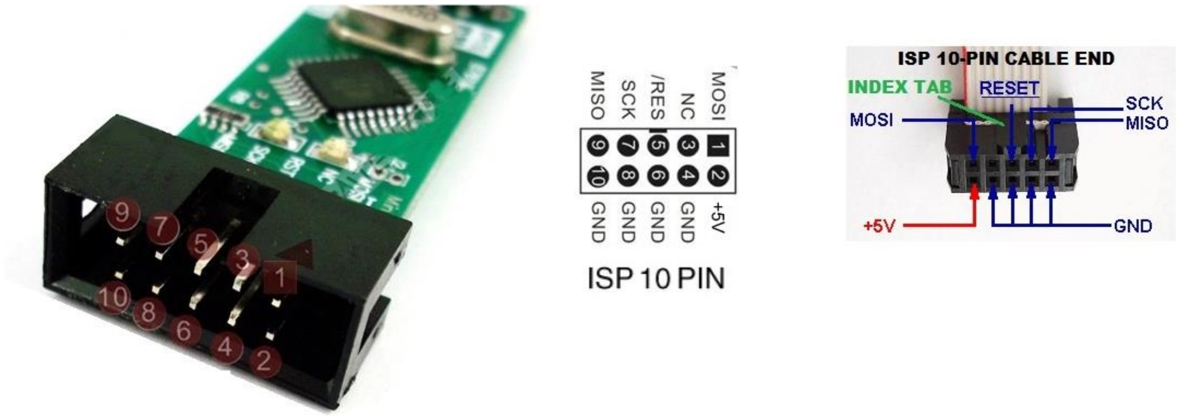


Figure 8. USBasp Pinout

An ISP programmer is normally connected to an Arduino via a 6-pin connector by a 3x2 header. Unfortunately the majority of Arduino Pro Minis don't have a 6-pin ISP programming connector, so we have to make a lead to suit. There are many ways this can be made, but I will detail the lead I have used for a few years without problem.

The 10 way ribbon cable will usually have a red stripe on one end designating pin 1. This ribbon cable is cut in half and the bare ends are separated for approximately 2.5cm (1" in old money) and a short length stripped back. Next two servo leads are connected to the ribbon cable, I used the leads off two old servo's that were no longer serviceable, but you could just as easily use new servo leads. In the photo's on the next page I have found a servo lead with JR colours (brown, red, yellow) and Futaba (black, red, white), so the two leads are easily distinguishable.

The next step is not essential, but I use a small screw driver to remove the red and white leads from one of the servo leads and swap them around. This is used for the GND, RST and Vcc connection so that the black coincide with GND and red with Vcc (see photo's on the next page).

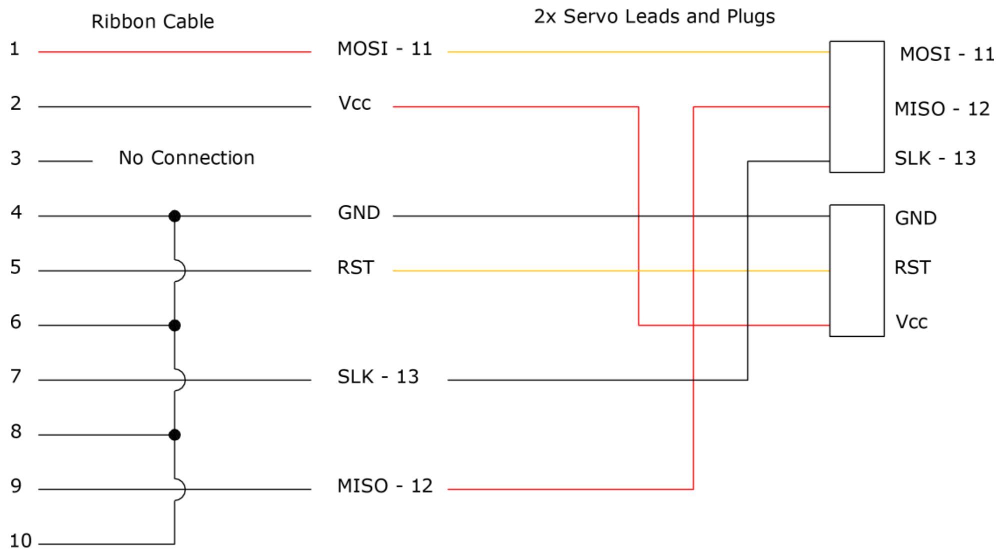


Figure 9. ISP Program Lead for an Arduino Pro Mini using two Servo Leads

Next solder the servo leads to the ribbon cable as shown in Figure 9, so you finish with a lead that can be plugged into the ISP programmer with the 10 pin connector and into the Arduino Pro Mini with the two servo leads. Note that you need to connect up core 4, 6, 8 and 10 of the ribbon cable to the GND connection.

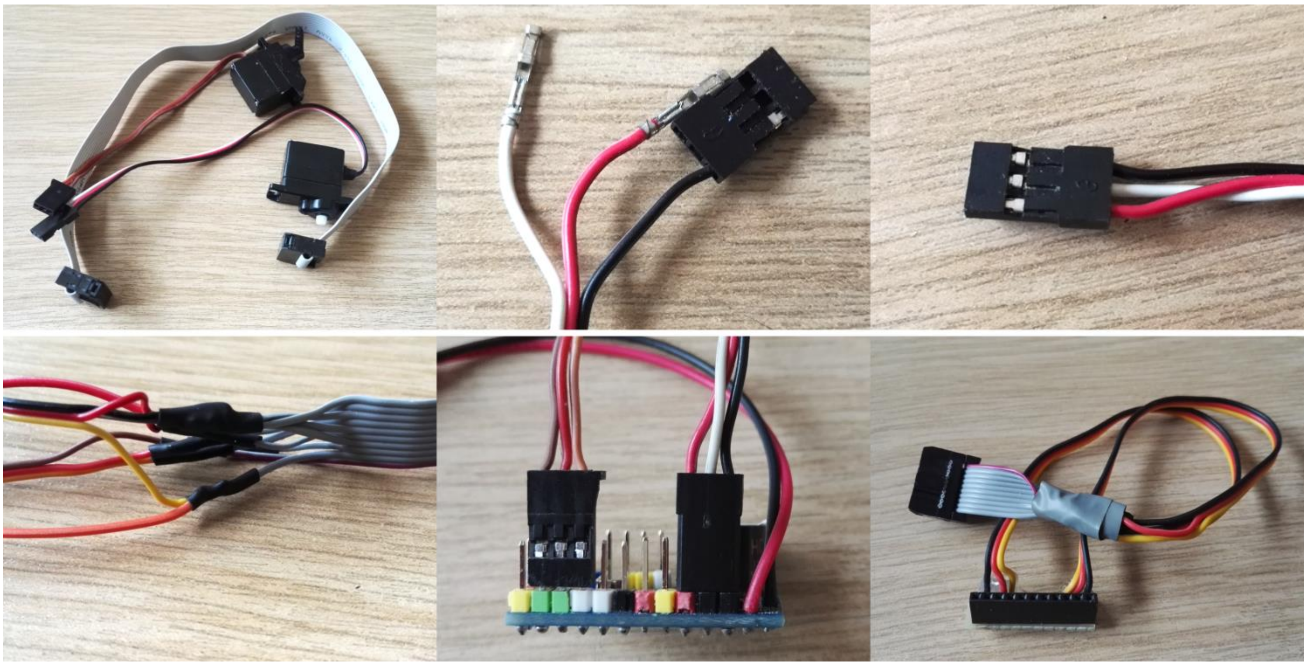


Figure 10. Stages in Making a Programming Lead for an Arduino Pro Mini

Figure 10 shows the stages in making a programming lead for a Arduino Pro Mini, from top left clockwise: The ribbon cable that come with an USBasp and two donor servo's, One servo lead swap the red and white wires (so the red and black line up with Vcc and GND), solder the servo leads to the ribbon cable as in Figure 9, how the servo leads/program cable connect to a Arduino Pro Mini, an alternative programming lead made from a 0.1" socket.

Programming the Arduino

Open the Arduino IDE and then open the encoder sketch/program. Select an Arduino Pro Mini and 5V 16MHz from the Tools -> Boards. Next select the type of ISP you have from "Tools -> Programmer" (USBasp in my case). Next select "Sketch -> Upload Using Programmer" this compiles the sketch/program and uploads it to the Arduino. Once complete remove the programmer cables and the encoder is ready to use.

The first time the Arduino runs, it detects that the EEPROM is blank and it automatically loads and save the default values, so it can be used without programming, as long as the default values suit your requirements.

Operating Modes

There are two different modes of operation, Run Mode used for flying and Program Mode used to set up or configure the encoder's settings. When you power up normally you will be in Run Mode and get two short beeps as confirmation. If you hold down the S/C switch when powering up you will be in Program Mode and get a single beep (0.8 secs) as confirmation.

Run Mode

In Run Mode, the Compound or Sequential operation, together with the duration of the timer are selected by fitting jumper links to the various header pins. The position of the jumper links are read at power up and if jumper links are either fitted or removed after powered up, their action will be ignored until the next time the encoder is powered up.

Compound Mode

Compound Mode is selected when there is **no** jumper link fitted to header 4 (green header in the photos).

This mode emulates the typical compound escapement such as the Elmic Compact where one press gives right, two presses give left, three presses give kick up elevator and a quick blip changes the throttle position (the kick up elevator and throttle positions can be adjusted or disabled in Program Mode if required).

The throttle can be 2 or 3 position and all throttle positions are adjustable in program mode.

The default throttle min and max values – throttle fully closed and throttle fully open are suitable for the vast majority of electric ESC, so don't adjust them if flying electric.

To get 2 position throttle adjust mid throttle to minimum (or maximum) in Program Mode and this gives:-
Electric: Closed (Stopped) -> Full -> Closed (it starts at closed throttle)
I/C: Tick-Over -> Full -> Tick Over (it starts at tick over)

Alternatively the mid throttle can be adjusted between the fully closed and open positions to give Compound 3 position throttle. Its order is:-
Electric: Stopped -> Mid -> Full -> Mid -> Stopped (it starts at closed throttle)
I/C: Tick-Over -> Mid -> Full -> Mid -> Tick Over (it starts at tick over)

Throttle beeps. When the blip throttle is operated and the engine speed changes the sounder beeps to indicate what throttle position you have selected. One beep for closed/tick-over, two beeps for mid-throttle and three beeps for full throttle, for both 2 and 3 position throttles

Kick up elevator is available. To make it kick down, go into Program Mode and adjust the elevator servo throw to minimum to reverse the servo. To disable kick up elevator, set elevator servo movement to near centre. The kick-up elevator escapement ”stop” is also removed, so pressing the S/C switch 3 times does not stop the “escapement” .

Sequential Mode

This mode emulates the typical sequential escapement such as the Elmic Conquest where one press gives right and when released back to neutral, the next press give left and releasing back to neutral, then right, neutral, left , neutral and so on.

Sequential operation is selected by fitting a jumper link to header 4 (green header in the photos).

Sequential beeps are available where the sounder beeps to indicate the next direction, one beep indicating right rudder next, two beeps indicating left next. This can be disabled in the settings in Program Mode.

Throttle Control

Motor starting and throttle control is different for "Electric" and "I/C" modes and **f**or Compound and Sequential, so there are four different combinations (Comp - Elec, Comp - I/C, Sequ - Elec and Seq - I/C).

But why have Electric and I/C modes? With electric powered models, you need a start sequence that will ensure that the motor does not start inadvertently. When the motor does start, the timer starts at the same time. But with models with I/C engines, you may want to start and adjust the engine, then start the timer just before launching.

Electric Mode is selected when there is **no** link fitted to header 5 (white header in the photos). I/C mode is selected by fitting a jumper link to header 5.

Starting an Electric Motor with Compound and Sequential Modes.

- Press the S/C switch (push button) for 3 seconds and the sounder starts beeping rapidly then after a further 2 seconds it beeps continuously for another second before the motor starts.
- Releasing the switch anytime during the start sequence cancels the motor start
- In both compound and sequential mode, the throttle starts at full throttle
- If the countdown timer, pulse when timer is finished or motor run timer (cut when timer finished) are enabled the timer starts running as soon as the motor is started (in Sequential mode the motor run timer set as default)
- The time set is common for countdown, motor run and motor pulse. It starts when the motor starts, but pauses in compound mode when the throttle is in the closed position (with the motor stopped).

I/C Motor

- When Compound Mode and I/C mode are selected together, the blip throttle can be controlled immediately on power on.
- When Sequential Mode and I/C mode are selected together, the throttle starts at the mid throttle setting and goes to full throttle once the timer is started. Remember that mid throttle is adjustable from minimum to maximum servo throw, so can be set from tick-over to max throttle. This was done so engines can be started at part throttle without them screaming at full rpm. WARNING never use these setting (sequential and I/C) if used with an electric motor as it may immediately start.
- To start the timer, press and hold the button for 3 seconds until there is a 1 second beep
- The count down timer runs continuously (unlike the electric mode, where it pauses when the throttle is closed)
- The Motor Run timer is available for both compound and sequential modes, so that short motor runs are possible without measuring out small quantities of fuel.

Timer

There is a single timer linked with the throttle that has three actions, Countdown Timer, Run Timer and “Pulse Throttle” 5 seconds before the timer elapses. All can be turned on or off in Program Mode settings.

The timer also acts differently with electric and I/C modes. With Electric Mode the timer only runs when the motor is running, so if the throttle is closed and the motor is stopped (compound mode 2 or 3 position throttle) then the timer stops until the motor is started again. But with an I/C engine it will run continuously once started.

Countdown Timer

Once the Timer is started it gives a short reminder beep at 6, 5 and 4 minutes remaining, then beeps at 3, 2 and 1 minute, one beep for each minute (ie 3 beeps at three minutes), three short beeps at 30 seconds, 2 short beeps at 20 seconds and continuous beeps from 5 seconds to elapsed when there is a long beep.

With Electric Mode once the timer has elapsed and the throttle is still open, there will be regular short beeps every 5 seconds to remind you that the timer has elapsed and the motor is still running. There is no further warning in I/C mode.

Run Timer

Once the timer has elapsed, the Run Timer closes the throttle. It is forced on when Sequential and Electric Modes have been set, but it can be enabled for any other mode.

“Pulse Throttle” 5 seconds before Elapses

If the countdown timer has been disabled, such as when using sequential beeps (so the next direction beeps aren’t confused with the countdown timer) a warning that the timer is about to elapse can be given by the “Pulse Throttle” 5 seconds before the timer elapses.

The throttle is rapidly switched on-off-on-off-on giving an easy to notice alarm. It is really only suitable for electric power, due to the speed of switching, an I/C engine may not respond quick enough. Alo it works well with most esc (electronic speed controller) designed for “quads” where a rapid response is needed, but doesn’t work as well (if at all) with up-market esc with a ramped response such as Castle Creations. It is defiantly a case of cheaper can be better.

Adjusting the Timer Duration

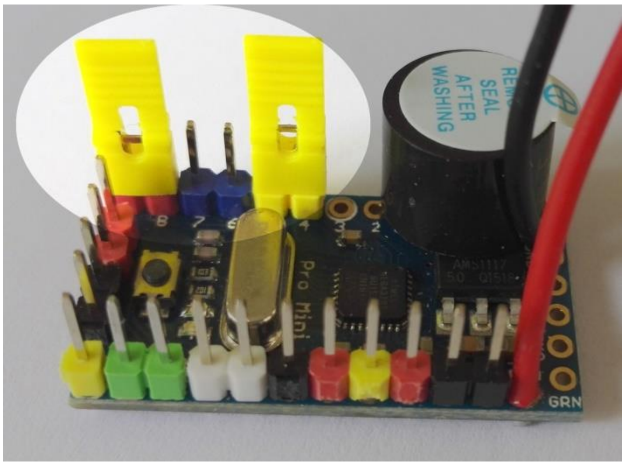
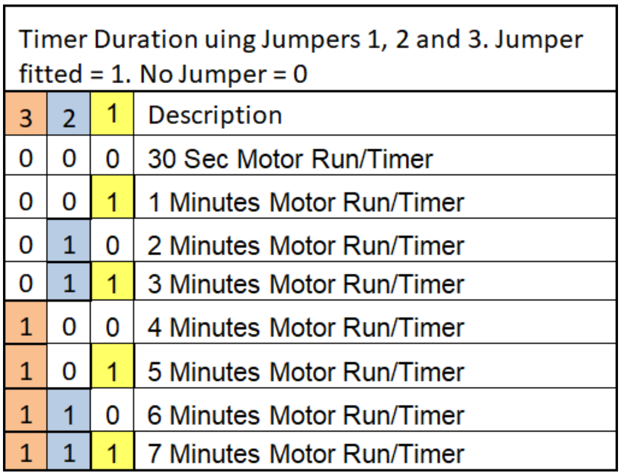


Figure 12. Timer Set to 5 Minutes with Headers 1 & 3



Timer Duration uing Jumpers 1, 2 and 3. Jumper fitted = 1. No Jumper = 0

3	2	1	Description
0	0	0	30 Sec Motor Run/Timer
0	0	1	1 Minutes Motor Run/Timer
0	1	0	2 Minutes Motor Run/Timer
0	1	1	3 Minutes Motor Run/Timer
1	0	0	4 Minutes Motor Run/Timer
1	0	1	5 Minutes Motor Run/Timer
1	1	0	6 Minutes Motor Run/Timer
1	1	1	7 Minutes Motor Run/Timer

Figure 11. Setting the Timer Duration with Headers 1, 2 and 3

The Timer duration is set by fitting jumper links to header 1, 2 and 3 to give a time as above in Figure 12. The timer is set to 5 minutes in Figure 11 (jumper link fitted to header 1 and 3).

Reed and Codamac Emulation

The three position switch (sprung to centre) used for programming can also be used on transmitters with a 3 position switch joystick such as a MacGregor Codamac, Futaba FR23 (FT-5E) or Staveley Tonelock) that automatically timed the switch keying to give reliable right and left with an escapement. An emulation is provided called Codamac Mode.

Alternatively it can emulate 2 channel reed operation for sets such as a Graupner Varioton, ED Black Knight or various REP’s. At the moment the reed emulation gives instant right and left and there is no “servo slow” to slow the servo to the original slow speed. This hasn’t found to be an issue with the generally slower flying rudder only models (or rudder and throttle).

Program Mode

- Program mode is entered by holding the S/C push button when powering on, you will get one longer beep (0.4sec) at power on to confirm (normal mode has 2 short beeps)
- It is used to adjust the various settings such as servo direction/travel and trim and also to turn various options on or off.
- To select which setting is to be adjusted or the option to be turned on or off, plug jumpers into the appropriate headers on the Arduino. The jumper can be moved with the Arduino powered up, so various settings can be altered in one session.
- In the table below, a "1" means fit a jumper link and a 0 means leave open
  eg to adjust the "Throttle closed" servo output, fit jumper links to headers 1 and 3 (the same headers fitted as Figure 11).

Jumpers 1, 2, 3, 4. Prog Jumper fitted = 1. No Jumper = 0. Press S/C switch 2 secs to save to EEPROM							
4	3	2	1	P	Description	Adjustment Range	Default
0	0	0	0	0	Rudder Neutral/Sub-Trim	+/-12.7% in 0.1% Steps	0
0	0	0	1	1	Elevator Neutral/Sub-Trim	+/-12.7% in 0.1% Steps	0
0	0	1	0	2	Rudder Travel and reverse	+/-100% (right rudder is output)	+100%
0	0	1	1	3	Elevator Travel and Reverse	+/-100% (up elevator is output)	+100%
0	1	0	0	4	Switch Travel and reverse	+/-100% (S/C operated is output)	+100%
0	1	0	1	5	Throttle Closed	0 - 100%	0%
0	1	1	0	6	Throttle Mid	0 - 100%	+50%
0	1	1	1	7	Throttle Full	0 - 100%	+100%
1	0	0	0	8	Count Down Timer Sounder	Inc = On : Dec = Off	On
1	0	0	1	9	Motor Run/Cut Timer	Inc = On : Dec = Off	Off
1	0	1	0	10	Pulse Throttle Before Timer Elapsed	Inc = On : Dec = Off	Off
1	0	1	1	11	Channel Order	Inc = TRES : Dec = RETS	RETS
1	1	0	0	12	3 Pos Switch Mode	Inc = Codamac : Dec = Reeds	Reeds
1	1	0	1	13	Sequential Bleeps	Inc = On : Dec = Off	On
1	1	1	0	14	Set low battery voltage	Inc = Set Voltage: Dec = Off	Off
1	1	1	1	15	Reset All	Press S/C switch for 4 secs	

Programming – General

- The servo travels/direction and sub-trims are increased/decreased using the three position switch in 0.1% steps for sub-trim and 1% steps for travel.
- At the maximum value you get a beep to indicate you are at maximum, at minimum you get two short beeps and if there is a centre (+/- values) you get a short beep to indicate the centre.
- When the 3 position switch is operated it “auto increases” after 1 second of pressing, so a short press gives a small step, continuously holding the switch will rapidly increase/decrease the setting.
- The other options are turned on or off using the three position switch and then written to EEPROM when the single channel switch is pressed for 2 seconds and is confirmed with a beep.
- If settings are adjusted and then the power is switched off without the single channel switch being pressed, the new settings will be lost.
- Therefore if you do not wish to keep a new setting, turn the power off without pressing the single channel switch

Programs 0 and 1 – Rudder and Elevator Trim

- The rudder and elevator sub-trim can be increased/decreased to get the servo neutral or to adjust the model trim if there is no easy method of adjustment on the model
- The sub-trim is applied at full servo travel as well, so will also affect the servo travel end point
- Don’t forget to press the S/C switch for 2 seconds until a beep is heard to save the settings

Programs 2, 3 and 4 – Rudder, Elevator and “Switch” Travel/Reverse

- Rudder Travel/ Reverse outputs full right rudder. Use the three position switch to decrease/increase the end travel.
- If the rudder output is reversed, then decrease the rudder movement past centre and continue until the rudder is at the desired right position.
- For Elevator Travel/ Reverse outputs full up elevator. If it is reversed or you wish to have kick-down elevator, then decrease the elevator movement past centre and continue until the elevator is at the desired position.
- If you wish to disable kick-up elevator, decrease the servo travel until it is neutral (no elevator movement) and you will get a short beep to confirm it is in the centre (0% movement). You may need to temporarily fit a servo to see the movement.
- For “Switch” Travel/Reverse, it output the servo position for when the switch is pressed. If the output is reversed, then decrease the switch servo output movement past centre and continue until it is at the desired position.
- Don’t forget to press the S/C switch for 2 seconds until a beep is heard to save the settings

Programs 5, 6 and 7 – Throttle Closed, Mid and Full

- The throttle closed, mid and full servo positions are each individually adjustable between fully anti-clockwise to fully clockwise.
- If the model is electric powered do not adjust the throttle closed or the throttle open as the default values suit 99% of esc (electronic speed controllers) on the market.
- For an I/C model, if the throttle servo is reversed, then adjust the throttle closed and the throttle open settings to reverse the servo operation. Don't forget to reset the 2.4GHz module failsafe.
- If compound 2 position throttle is required, adjust the throttle mid to minimum (decrease the value until 2 short beeps).
- If Sequential and I/C modes will be used then the initial throttle position is set by the throttle mid. So you can have Sequential and I/C mode start with the throttle near tick over or full throttle, whichever is easier to start and handle.
- Don't forget to press the S/C switch for 2 seconds until a beep is heard to save the settings

Program 8, 9 and 10 – Countdown Timer, Motor Run and Pulse Throttle Before Timer Elapsed

- These timer options can be individually turned on or off, you can have them all on, all off or just the selected ones on.
- To turn them on press the 3 position switch to the increase position, confirmed by a single beep
- To turn them off press the 3 position switch to decrease position, confirmed by 2 beeps
- Don't forget to press the S/C switch for 2 seconds until a beep is heard to save the settings

Program 11 – Channel Output Order

- Channel output order can be selected between RETS - Rudder, Elevator, Throttle and Switch (FrSky) or TRES – Throttle, Rudder, Elevator and Switch (Spektrum and OrangeRX)
- To select TRES press the 3 position switch to the increase position, confirmed by a single beep and to select RETS press the 3 position switch to the decrease position, confirmed by 2 beeps
- Don't forget to press the S/C switch for 2 seconds until a beep is heard to save the settings

Program 12 - Three Position Switch Mode

- The same switch that is used for programming can also be used for rudder control, either to emulate a MacGregor Codamac or alternatively a two channel reed set such as a Graupner Variophon.
- To select Codamac mode press the 3 position switch to the increase position, confirmed by a single beep and to select Reeds mode press the 3 position switch to the decrease position, confirmed by 2 beeps
- Don't forget to press the S/C switch for 2 seconds until a beep is heard to save the settings

Program 13 – Sequential Beeps

- When in Sequential Mode you can have beeps to indicate what the next switch press will give, one beep for right next press and two beeps for left next press.
- To enable sequential beeps, press the 3 position switch to the increase position, confirmed by a single beep and to disable press the 3 position switch to the decrease position, confirmed by 2 beeps
- Don't forget to press the S/C switch for 2 seconds until a beep is heard to save the settings

Program 14 – Low Battery Voltage Alarm

- The encoder can alarm when there is low battery voltage and is set at 7.2V into a divider using 2x 10K 1% resistors to give 3.6V to A7.
- To enable the Low Battery Voltage Alarm, press the 3 position switch to the increase position, confirmed by a single beep and to disable it press the 3 position switch to the decrease position, confirmed by 2 beeps
- If a custom alarm voltage is required or it needs to be calibrated as lower tolerance resistors have been used press the 3 position switch to the increase position, confirmed by a single beep and continue to hold for a further 3 seconds when there will be a longer beep. The new alarm voltage has been set.
- Don't forget to press the S/C switch for 2 seconds until a beep is heard to save the settings

Program 15 – Reset All Settings to Default

- If you wish to return all the settings to the default values (if say it is being used for a new model) then press and hold the S/C switch for 4 seconds until it beep to confirm that all default settings have been saved to EEPROM.

Components

Most of the components should be easy to find, indeed everything could be found on eBay, but not necessarily the best price or in some cases quality. I use RS Components and Farnell for most of my components, but have given Rapid Electronics (https://www.rapidonline.com) components as they are more user friendly for non-trade customers.

There are literally hundreds of vendors for Arduino Pro Mini. If you want an original Sparkfun version then Proto-Pic can be recommended. However I have used clone Arduino Pro Mini from eBay and Banggood with complete success. And the clones are often a fraction of the price of the originals, but the old adage of "you get what you pay for" should be remembered as the component quality of the clones is unknown, whereas the Sparkfun version is a reputable company. "You pay your money and take your choice" applies here.

Sparkfun Arduino Pro Mini 5V 16MHz	Proto-Pic DEV-11113 (www.proto-pic.co.uk)
Arduino Pro Mini 5V 16MHz Clone	eBay, Banggood etc
Jumpers with a tab/handle	Rapid 22-0695
5V Sounder	Rapid 35-0055
12V Sounder	Rapid 35-0056
3 Position Switch (sprung to centre)	Rapid 75-0086
Servo Leads	Spares Box!