Shortcuts

SRCMBC HOME

Hints and Tips

ALAN'S ELECTRICS
SOUND UNITS
Marine Engine

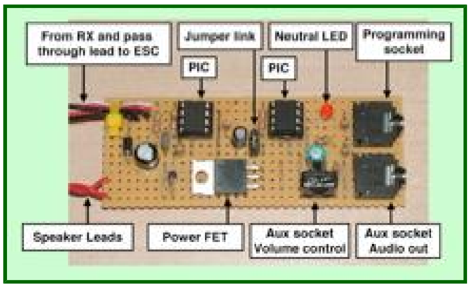
_ _ _ _ ? ...

Hosted by 1&1

Versatile Marine Engine Sound Unit

Update (June 2009)
There is demonstration of the unit on the SRCMBC Channel on YouTube (search for "Alan Bond"), and you can also hear (and see) it in action in the movies of "Phoenix" (petrol) or "Marignon" (diesel) on the same SRCMBC Channel.

Overview
This is a general purpose engine sound module that can produce a variety of different engine sounds. In all cases the hardware module remains the same and the type of engine sound and it's character are varied by firmware (ie the fitted software) alone.



The three software builds currently available implement multi-cylinder petrol, diesel or steam engines, the number of cylinders being a user selectable option by the use of a jumper link. In all cases the engine sound varies smoothly and proportionally with the throttle demand. The petrol and diesel engine variants have an idling time-out of around one minute after which they run down and stop. Opening the throttle again causes the engine to re-start. The steam engine variant stops immediately the throttle is closed.

All units may be built 'as is' using the released software, or the more adventurous of you can experiment by tweaking the values in the code to change the exhaust pulse (or steam hiss) character of each cylinder, the tickover speed, the rev range, the idle time-out period etc. The code has been specifically written to make such experimentation easy and has been commented accordingly.

I am keen that you experiment with the code yourselves and will be only too delighted to be put in the shade by your efforts - so please share your results for the benefit of us all, by e-mailing them to me.

With club members in mind, this project has been tagged as a 'marine engine' sound unit , but clearly this module could also be used for steam or diesel locomotives and maybe even tanks etc.

So What Does It Actually Sound Like?
Below you can access some short sound samples in mp3 format that illustrate the results obtained. Please bear in mind this is a 'fun' attempt to synthesize an 'engine-like' sound and thus it may not fit the bill for some of the more 'serious' modellers out there. They should consider using sound units that employ recordings of actual engines (which in turn cost 'serious' amounts of money).

4 cylinder petrol engine 3 cylinder diesel engine 4 cylinder steam engine

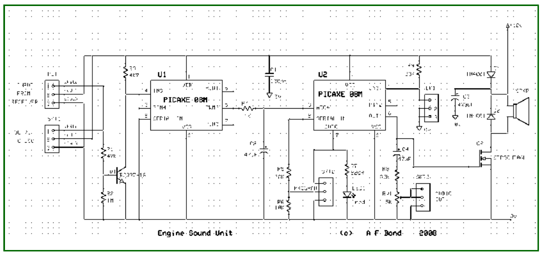
For the purposes of the above demonstration files, the idling time-out (approximately one minute) has been reduced to a few seconds. My thanks go to Ray Hellicar for collaborating with me to optimize the sound parameter values and the engine starting 'noise'.

Programmable Interface Controllers (PICs)
The high degree of reconfigurability of this module is made possible by using chips called PICs (programmable interface controllers) -aka microcontrollers.

The particular variant of PIC I am using is called the 'PICAXE' which is programmed in BASIC. As well as being easy to understand and use, this particular dialect of BASIC has many specialised commands intended for radio control and robotics applications which renders it ideal for this project. A good starting point to see the entire range of PICAXE chips, their support hardware and accessories - not to mention downloading the free programming editor software - is on the TECHNOBOTS site.

Arrangements have been made with TECHNOBOTS to supply all the specialised components required for this unit (including the option for supply of pre-programmed PICAXE chips) as a complete 'kit'. The circuit diagram and assembly drawings are available for download from the TECHNOBOTS site, where the software (source code) may also be found in the .bas format (ie suitable to load directly into the programming editor) for those wishing to program the chips themselves and/or tweak the code.

Hardware



As the circuit shows, two PICAXE chips are used. The first chip interfaces to the receiver and processes the throttle demand, whilst the second chip produces the actual engine sound via a power FET output stage.

The input signal from the receiver is converted (if necessary) to a full range 5v signal required to interface with the PIC (U1) by a transistor buffer stage (Q1, R1, R2, R3) which has the consequence of inverting the polarity of the signal, but this is dealt with by the software. This gives compatibilty with receivers which have insufficient output voltage swing to operate the PIC. The software in this chip converts the incoming radio control signal into a 4Khz pulse-width modulated (PWM) signal from output PWM2. This pulse train is low pass filtered by R4, C2 to produce a smoothly varying dc control voltage from zero to 5v.

This control voltage is fed into the second PIC (U2) which is responsible for sound generation. The signal is converted from the analogue to the digital domain by one of the PIC's A to D converters (ADC4) for further processing. At first sight it may seem strange to have converted the receiver signal from digital to analogue and then back to digital again, but the sound generation PIC (U2) is so busy it cannot process the receiver input AND generate the sounds, but it *does* have time to glance at a dc control voltage on one of its input pins.

The engine sound signal is output from pin6 and fed to an audio out socket SKT3 via C4, R8 and volume control RV1 - this caters for integrating the unit output into an existing boat amplifier system. For stand-alone systems, the sound signal can drive a speaker via power FET Q2, which must be a logic level enhancement type as only a 5v pk to pk gate signal is available from the PIC (U2). D2 protects Q2 against any high voltage spikes caused by driving the inductive load presented by the speaker. D1 allows the speaker to be driven from the 5v supply direct, but becomes back-biased, thereby protecting the 5v supply, if a 7.2v, 9.6v or 12v supply is attached to increase the volume. C3 smooths the speaker supply as large current pulses are taken when the FET switches - however the low duty cycle of the sound signal results in a modest average current consumption.

LK1 is a two position jumper link which either connects IN3 to 0v or 5v and is read by the PIC to select software options. The receiver input is passed out unchanged to the boat's ESC on the flying socket SKT1, thereby obviating the need for a 'Y' lead.

Software

The published software has been extensively commented so it is proposed to just give an overview of its operation in general. If any points are still obscure please e-mail me. Listings of the software are available here as pdf files, or a ZIP file of all the source code in bas format may be downloaded from the TECHNOBOTS site.

The software in U1 ("speed demand.bas") converts the incoming r/c signal into a 4Khz pulse-width modulated (PWM) signal from output PWM2. A bi-directional throttle command is assumed with neutral/stop based on a 1.5mSec pulse. The software therefore produces an increasing speed demand output for deviations either side of neutral. To cater for the logarithmic nature of sound and the (fixed) code execution time that becomes increasingly significant as the engine speed increases, the speed demand is quantised into 25 distinct values and a quasi-logarithmic lookup table is used to assign values to the PWM command to ensure a linear sounding throttle response. In practice the 25 steps of speed sound smoothly continuous (some commercial units only use 8 steps!). The "pwmout" command (only available for PIN2) is utilised as once invoked it operates as a background task, independent of the program, allowing the PIC to now monitor the receiver input.

The PWM signal is then low pass filtered to produce a smoothly varying dc control voltage which is then passed to the second chip.

U2 is loaded with the chosen petrol, diesel or steam sound software. It takes the dc control voltage input and produces an engine note (ie revs) that varies in direct proportion to that control voltage.

To assist users with setting their tx throttle trim, when the zero speed 'step' is selected a LED is turned on. In the case of the petrol/diesel variants, a time-out counter is also started whenever the throttle demand is zero, and of course it is reset should the throttle be re-opened prior to the timeout.

The actual exhaust pulse noise is implemented using the PICAXE's 'sound' command, which can be set to output various pseudo random digital pulse trains, from a rumble to a hiss. It can also vary the duration of the sounds, so that short staccato pulses of rumble can be used for internal combustion type engines, or long pulses of hiss for the steam sound.

The character of the sound (especially the perceived 'beat') is varied by assigning different noise pitch values to each cylinder. For example, a repeated loop of three different sounds can be used to create a three cylinder internal combustion engine or a three cylinder marine steam engine. A single cylinder engine can be simulated by reducing the sound loop to a single element. Alternatively the loop can be expanded to embrace four or even six cylinders.

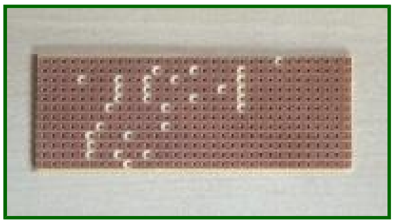
Jumper link LK1 determines if the cylinder selection mode is to be entered. If the jumper is in position at power-up the program issues a 'programming' beep and then plays short samples of the engine with increaing numbers of cylinders and you remove the jumper when you hear the one you want to use. A comfirmatory beep is issued and the chip is then locked into that mode. The chosen number of cylinders is stored in the PICAXE's (non-volatile) EEPROM so at the next power-up if the jumper is not fitted (normal condition) the program reads the value stored in EEPROM and commences operation.

It will be noted that the petrol and diesel programs are in fact identical other than the sound data values embedded in them. The steam program, whilst neccesarily different, is however broadly similar in concept.

The Build

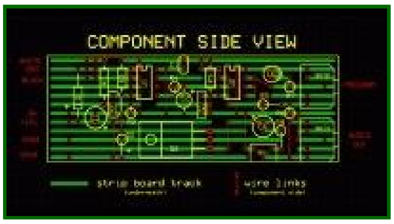
The drawings below show a strip-board layout which has been designed with the less experienced constructor in mind. A personal computer serial interface has been included so as to give a means of uploading the published programs into the two PIC chips. The layout also assigns the serial interfaced PIC chip to the sound generation task, as that way it is easy to experiment with loading different values into the sound commands or the idle, revs range and timeout constants without having to remove the chip.

track cutting guide

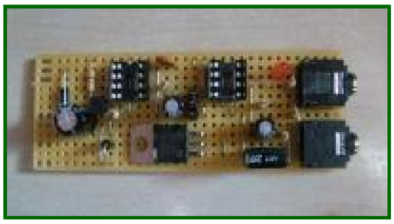


the cut stripboard

The strip-board should be cut to size and then the tracks cut in the indicated positions. A 3 or 4 mm drill bit makes a good track cutter - just place the tip in the appropriate hole and twist it back forth between your fingers. Lowest profile parts should be inserted first so that means the wire links. Note that to save space, there are wire links beneath U1 and U2, so they *must* be inserted before the chip sockets. Then add the components in order of ascending height. Finally check for solder splashes between adjacent tracks, the most common fault in strip-board assemblies. The flying lead plug and socket (PL1 and SKT1) are made from cutting a servo extension lead in two and I chose to anchor these leads to the board by a small tie-wrap.



parts layout



built unit

If you have bought pre-programmed PICAXE chips then they may be fitted (ensure the correct function is in the correct socket), otherwise the following steps must be followed if you wish to program blank PICAXE chips yourself. To do so, you are assumed to have some computing ability and be able to resolve issues like assigning the PC's COM port to get things working. If in doubt go for the pre-programmed chips option.

Firstly you must have downloaded and installed the free programming editor software and secondly downloaded the zip file of all the software modules - all available free from the TECHNOBOTS site.

Now start up the programming editor and load the program "speed demand.bas". Fit a blank PICAXE chip in the U2 position (NOT its final location, but it needs to connected to the serial interface socket for programming before moving to position U1). Fit the programming lead from your PC's serial interface socket to the 3.5mm stereo socket SKT2. Power the unit and from the programming editor menu select PICAXE/RUN. If all is well you should see the progress bar indicating the chip loading in progress. When complete, remove the power and put the chip you have just programmed in position U1. Then repeat the procedure with a new chip for "sound generation (xxxxxx).bas".

For those club members who feel unable to attempt the precision assembly and soldering or chip programming, I might be conned into making a unit for you at cost, for some assistance in return with glamorizing the temporary cardboard decking and superstructures for which my models are famed, or maybe I could be tempted by some article from your junk-box.

The Installation
I must point out that I have no previous experience of installing sound systems in boats, so all I can do is give some guidance based on my own experiments. Those of you who have already fitted commercial sound units, and maybe have been doing so for years, are invited to share your experience with us on the club website.

It can't be stressed too much that the final quality of the sound is highly dependent on the size, and type of speaker used and the 'character' of that sound is modified - for better or worse - by the speaker's mounting and location in the boat. For example a bench test using my music-centre speaker gives awesome results, but any speaker I have that is capable of fitting into my boats is, by comparison, a serious compromise!

For the diesel version, the more bass your speaker can create, the better - it will also benefit enormously from fitting a resonating tube approximately the diameter of the cone on top of the speaker and/or coupling the speaker to the hull on a large area cross-deck. Bench experiments with the resonating tube will also show a large change in sound 'character' as the open end of the tube is gradually occluded. But be sure to check how these results are changed when the speaker and tube assembly are mounted in the boat with the superstructure in place.

It has been suggested that there may be some mileage in experimenting with the satellite speakers from computer surround-sound systems, or MP3 docking stations which reportedly give a punchy bass even without their sub-woofer.

If you chose to fit a regular speaker, then it should be noted that paper coned speakers generally have a much lower resonant frequency than the mylar coned (splash-proof) types and if you can get one dubbed as a 'woofer' or 'bass' unit with a long-throw rubber cone surround so much the better. Unfortunately the latter are generally very heavy, as well as bulky, though in some cases you may be able to remove ballast to compensate for that.

The steam sounds are of much higher frequency and good results have been obtained with a mylar coned speaker from TECHNOBOTS part no. 2400-010. A (larger, heavier but more powerful) speaker suitable for the petrol and diesel units is part no. 2400-015.

Sound Experimentation / Code Tweaking
This is best carried out with the final speaker installation in the model itself - if necessary, temporarily extend the speaker wires to get the circuit within range of your download cable. As with all experimentation it is best to make small changes at first until you become more acquanted with the module.

Character changes relate to the exhaust pitch and duration as well as the number of cylinders. In my experiments, increasing the number of cylinders beyond four produced very little extra character, but the cylinder lookup tables support up to six cylinder operation. My experiments with single cylinder variants of the petrol or diesel engine produced a less than convincing sound and I have chosen not to embarrass myself by supporting these variants in the released source code, but they are very easy to implement should you wish to experiment yourselves. I am however currently working on another approach to this problem - watch this space.

Exhaust pitch values must lie in the range 128 to 255 otherwise constant frequency tones are produced rather than random noise (the former are intended for the PICAXE to play tunes). The lower numbers are best for the diesel character and the higher values more suited to petrol engines.

Exhaust durations are best kept fairly short - dependent upon the pitch involved, values above 6 or so tend to produce a bad 'squawk'. The cumulative (time) value of these exhaust durations will have a minor affect on the engine operating speeds.

With the petrol and diesel variants, the tickover speed, and the revs range (ie the throttle sensitivity from tickover upwards) may be set by adjusting the appropriate constant equate values in the code. Likewise the startup speed and revs range of the steam program.

As noted in the (petrol/diesel) code comments, the 'starter' sound could undoubtedly be improved by keen experimenters, or omitted altogether.

I wish you lots of fun experimenting - and please keep me informed of your masterpieces, so I can share them with others.

The Future
My long term aim is to build sound units utilising real recordings of boat engines, naval guns, ship's horns, sirens etc so I don't plan to put too much more effort into these engine sound units, but I think you'll find they compare favourably with commercial units that employ similar sound synthesis techniques. However, if sufficient interest is displayed, I have an idea for a de-luxe version of the unit where engine parameters can be adjusted by potentiometers in real-time rather than by code tweaking - over to you!

I look forward to discussing all or any of the above with you, at the pondside, by e-mail, or by using my Feedback form.

Page maintained by Alan Bond Last modified: 13:57, 26 June 2009