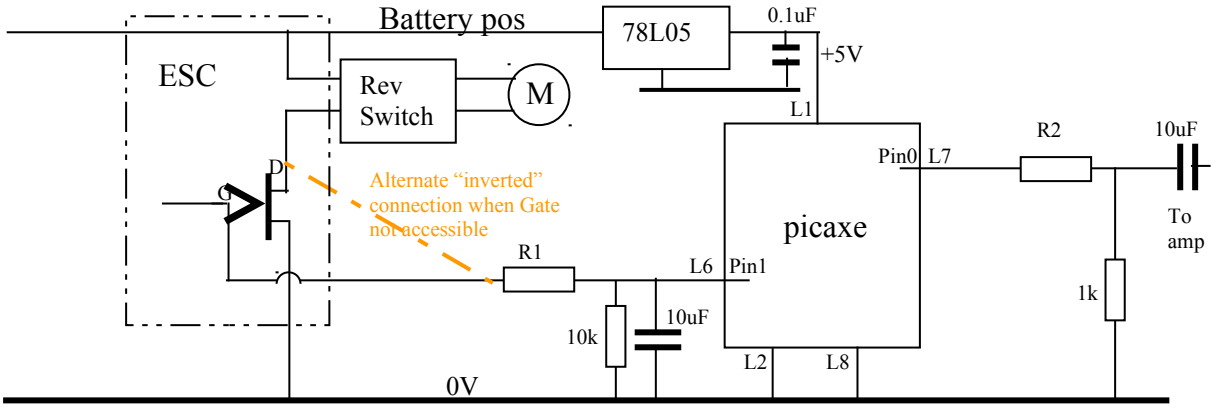
The SANDSTONE & TERMITE RAILWAY
"Linking the Outback to the World"

PICAXE TUTORIALS

3. Sound for battery operated Diesel loco

NOTE: All these programs are for battery operated locos. It should be possible to use them on track power by using a battery to supply the amplifier and Picaxe, and a bridge rectifier from the motor terminals to the input R1 in the diagram. I have not checked this aspect.

Simplest sound – no volume control and the same cct for both diesel and steam



Diesel program:
We produce a diesel throb sound using a shift register to produce a string of pseudo-random numbers at a rate determined by the speed of the loco – ie its motor voltage.
As for the steam chuff, we read in a value for speed (0-1023) on pin1, convert to max 100, then 'clock' (shift) the shift register and wait for a time (pausetime) determined by 'speed' before we repeat.
The shift register is programmed like this – the 4th and 5th bits in an 8 bit word are Exclusively OR-ed and the result put back into bit0. Here's the subroutine to do it:-
SYMBOL pausetime=b4

```
        PAUSE pausetime
        seed=seed*2                 'shift left 1 bit

        bit0= bit5^bit4             'do an EXOR  (^) and put result in bit0
        if bit0=1 THEN GOTO one:
        LOW 0                       'set sound output to low
        GOTO start

one:    HIGH 0                      'set sound output to high
        GOTO start
```

pin0 is set either high or low and is the output sound sent to the amplifier. It's that simple.

As in the steam sound, pausetime is inversely proportional to speed, so it is calculated similarly. Something like:

Pausetime= **60**/speed

Complete program listing of:-

SimplestDiesel.bas

```
'example of the simplest Diesel sound generator
'Greg Hunter            6/4/14
'a voltage on pin1 represents the speed of the loco.
'a disel sound is produced on pin0, by using a shift register to produce
'a pseudo-random sequence, and which must be amplified externally.
'suitable for 08M  and 08M2   at 4 MHz

'pin0 is sound output
'pin1 is speed voltage input 0V=stopped, +5V=max speed
                                'if speed volts are inverted ie 5V=stopped and 0V=max speed, then
                                'enable the line INVspd:

'define variables
        SYMBOL seed=b0     'THE shift register. must use b0 or b1 to allow access to individual
bits
        'b1 not used
        SYMBOL speed=w1
        SYMBOL pausetime=b4

'define constants
        SYMBOL maxspeedcounts=100       'max speed will be represented as this in prog.
        SYMBOL stoppedcounts=5          'define less than this as 'stopped'= about 5% of
maxspeedcounts
        SYMBOL maxspeedIN=255           'see notes at end to calc this value
        SYMBOL speedconstant=100                'adjust for suitable throb rate

'-------------------------------------------------------------------------------------------
        seed=233                'experiment with this - can't be zero!

start:
        READADC10 1, speed
'now adjust for actual input voltage and normalise to a specified maximum (maxspeedcounts)
        speed=speed/4*maxspeedcounts/maxspeedIN MAX maxspeedcounts


'INVspd:  speed=maxspeedcounts-speed     'only use if speed voltage is inverted

'*now speed is always represented by 0 when stopped and maxspeedcounts at max input voltage*

ST10:
        if speed<stoppedcounts then stopped:          'define as stopped go idle
                                                'needed to avoid divide by zero
ST15:
        pausetime=speedconstant/speed  '(in ms)

'other possibilities for calculating pausetime are:
'       pausetime=maxspeedcounts-speed*speedconstant/maxspeedcounts/stoppedcounts+1
```

```
throb:
        PAUSE pausetime

'now do the shift register thing...
        seed=seed*2                     'shift left 1 bit

        bit0= bit5^bit4                 'do an EXOR  (^) and put result in bit0
        if bit0=1 then one:
        LOW 0                           'set sound output to low
        goto start

one:    HIGH 0                          'set sound output to high
        goto start

stopped:
        pausetime=speedconstant/stoppedcounts       'define idle throb rate
        goto throb
'+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

'NOTES:
'the max volts applied to pin1 for the speed MUST NOT EXCEED the picaxe supply voltage
'assuming that is 5V, then a voltage divider on pin1 must reduce the motor volts to
'less than or equal to 5V at the MAXIMUM SUPPLY VOLTAGE !
'It would be prudent to reduce the voltage to a bit less than 5V

'if max volts on pin1 are less than picaxe volts (5V) then you must determine the value
'for maxspeedIN as follows:
'using a multimeter measure the picaxe supply volts (Vp) and
'the voltage on pin1 with max supply voltage (Vm1). Then
'maxspeedIN= Vm1/Vp*256. (round the result UP, and must max of 256.)
'example,  you measure Vp=4.28V and Vm1=2.65V, then enter SYMBOL maxspeedIN=159
‘-----------------------------------------------------------------------------------------
```

Using a LOOKUP table

Another possibility for calculating pausetime is to use a LOOKUP table. This can give a more ‘tailored’ throb for lower speeds. This example uses just 6 ‘notches’ of diesel throb rate, and idle. The LOOKUP table is just like an index, the first term defines how many steps you go across the table to get the result. (starting at zero!)
Eg LOOKUP index, (#0,#1,#2,#3,#4,#5 etc), result
Setting index to 0 will give result= #0, set index to 4 and get result= #4)
If we have ‘index with speed values from 0 to 6, the values #0 etc will be the paustime we need at that speed.

So we need to get index to have values from 0 to 6 when speed goes from 0 to 100. A simple way is to divide the max speed 100 by 16. Then speeds from 0 to 15 will give zero, 16 to 31 give 2 etc till 96 to 100 will give 6. This means we don’t get to ‘notch 1’ until speed 16 which is a bit high! So we can try using index= (speed+11)/16. Then speeds 0 to 4 will give ‘idle’, speeds 5 to 20 will give the 2nd entry (notch1) for pausetime etc up to the 7th entry (notch6) for speeds 85 to 100.

So instead of calculating as above, Pausetime= **60**/speed, we can use:
SYMBOL speedindex=b5

speedindex= speed+11/16
LOOKUP speedindex,(25,12,10,7,4,2,1), pausetime

The value 25 is the pausetime for idle, the value 12 is the pausetime for the 1st accelerating notch and the value 1 is the pausetime for the 6th notch.

If we use a LOOKUP table we can delete speedconstant.
Need a separate program????

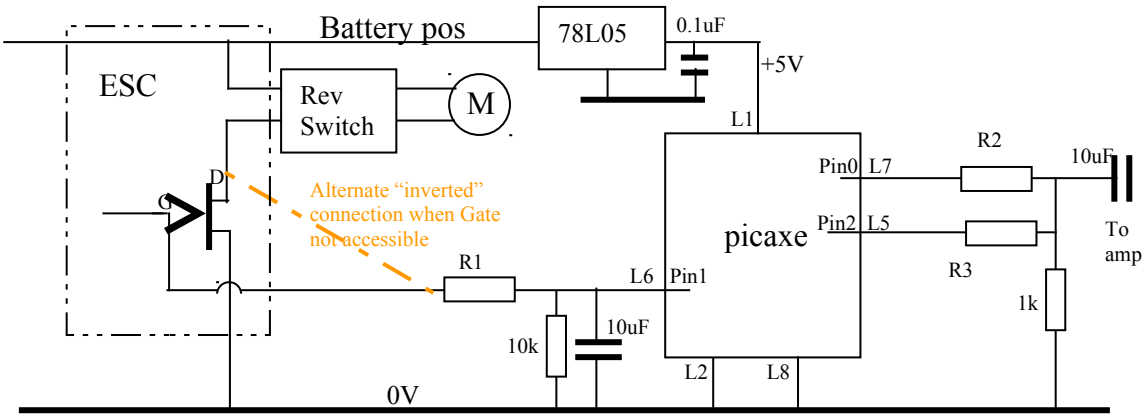
Horn sound

You can make a reasonable single tone horn sound using SOUND pin,(57 ,100). Try experimenting with different values for 57.

Volume control

It would be nice to control the volume, so that for instance, the diesel throbs are quieter when not accelerating.
A simple way would be to control a relay and use its contacts to short out a resistor in series with the speaker. But that's not very elegant – I'd prefer to use the Picaxe directly.
I've used a couple of circuits to do this:

The simplest idea is to send the sound output to two different pins which connect to circuits that provide different gains. It just needs two additional resistors (R2 & R3) and some program additions.



Try using 10k for R2 and 47k for R3. I've found that a quiet level of about ¼ of the loud is about right.

When you want loud you use pin0 as before.
When you want softer volume use pin2

Now how to get the program to decide if we want loud or soft?
I decided I wanted the volume to be soft when speed was decreasing and when loco is stopped.

So I use two new variables called ‘oldspeed1’ and ‘oldspeed2’ to store the speed from the two previous readings. If the ‘oldspeeds’ are less than the latest ‘speed’, then we want it loud. Otherwise we want soft. Because speed is in the range 0-100, oldspeed only needs to be a byte. I found it necessary to use two oldspeeds because you often get a ‘jitter’ of +/-1 in the speed reading.
Then define another variable to hold the pin number we want the sound on (0 or 2).

so define:-
SYMBOL oldspeed1=b10
SYMBOL oldspeed2=b11
SYMBOL soundpin=b8

Then the basic program flow is like this:
Start:
oldspeed2=oldspeed1 ‘to update the oldspeeds
oldspeed1=speed
Then use **READADC10** 1, speed etc as before to get speed 0-100

Now conditions for loud are
1) just starting from stopped, or
2) speed is increasing

IF speed<stoppedcounts **THEN GOTO** stopped
IF oldspeed1=0 and speed>stoppedcounts **THEN GOTO** loud
IF speed>=oldspeed1 AND oldspeed1>oldspeed2 **THEN GOTO** loud
IF speed<oldspeed1 AND oldspeed1<oldspeed2 **THEN GOTO** soft
(if we get here, we just want to leave volume as it was)

ST15: pausetime=speedconstant/speed

Throb: **PAUSE** pausetime
seed=seed*2 'shift left 1 bit

bit0= bit5^bit4 'do an EXOR (^) and put result in bit0
IF bit0=1 **THEN GOTO** one:
LOW soundpin 'set sound output to low
GOTO start

one: **HIGH** soundpin 'set sound output to high
GOTO start

stopped: sets volume soft
soundpin=2
LOW 0 'stop loud vol pin hanging high
speed=0
oldspeed1=0
pausetime=speedconstant/stoppedcounts 'define idle throb rate
GOTO throb

loud:
LOW 2
soundpin=0 ‘set volume loud
GOTO ST15

soft:
```
        LOW 0               'stop loud vol pin hanging high
        soundpin=2          ‘set volume soft
        GOTO ST15
```

Complete listing for:-
SimpleVolumeDIESEL.bas
```
'example of simple Diesel sound generator WITH LOUD/SOFT VOLUME
'Greg Hunter            9/4/14          94 bytes
'a voltage on pin1 represents the speed of the loco.
'a disel sound is produced on pin0, by using a shift register to produce
'a pseudo-random sequence, and which must be amplified externally.
'suitable for 08M  and 08M2   at 4 MHz

'pin0 is LOUD sound output
'pin1 is speed voltage input 0V=stopped, +5V=max speed
                            'if speed volts are inverted ie 5V=stopped and 0V=max speed, then
                            'enable the line INVspd:
'pin2 is SOFT sound output


'define variables
        SYMBOL seed=b0     'THE shift register. must use b0 or b1 to allow access to individual bits
        'b1 not used
        SYMBOL speed=w1     'will be 0-maxspeedcounts in prog after reading in and processing
        SYMBOL pausetime=b4  'determines throb rate
        'b5 possible use in a LOOKUP table
        'b6 not used
        'b7 not used
        SYMBOL soundpin=b8        'pin0 for loud, pin2 for soft
        'b0 not used
        SYMBOL oldspeed1=b10
        SYMBOL oldspeed2=b11

'define constants
        SYMBOL maxspeedcounts=100       'max speed will be represented as this in prog.
        SYMBOL stoppedcounts=5          'define less than this as 'stopped'= about 5% of
                                        maxspeedcounts
        SYMBOL maxspeedIN=255           'see notes at end to calc this value
        SYMBOL speedconstant=100        'adjust for suitable throb rate

'-------------------------------------------------------------------------------------------
        seed=233                'experiment with this - can't be zero!

start:
        oldspeed2=oldspeed1
        oldspeed1=speed

        READADC10 1, speed
'now adjust for actual input voltage and normalise to a specified maximum (maxspeedcounts)
        speed=speed/4*maxspeedcounts/maxspeedIN MAX maxspeedcounts
```

```
'INVspd:     speed=maxspeedcounts-speed      'only use if speed voltage is inverted

'*now speed is always represented by 0 when stopped and maxspeedcounts at max input voltage*

            if speed<stoppedcounts then stopped          'define as stopped go idle
L20:        if OLDspeed1=0 and speed>stoppedcounts then loud
L35:        if speed>=OLDspeed1 AND OLDspeed1>OLDspeed2 then loud
L40:        if speed<OLDspeed1 AND OLDspeed1<OLDspeed2 then soft
'otherwise just leave as it was

ST15:
        pausetime=speedconstant/speed  '(in ms)
throb:
        PAUSE pausetime

'now do the shift register thing...
        seed=seed*2                             'shift left 1 bit

        bit0= bit5^bit4                 'do an EXOR  (^) and put result in bit0
        if bit0=1 then one:
        LOW soundpin                                    'set sound output to low
        goto start

one:    HIGH soundpin                           'set sound output to high
        goto start


stopped:
        soundpin=2
        LOW 0                   'stop loud vol pin hanging high
        speed=0
        oldspeed1=0
        pausetime=speedconstant/stoppedcounts       'define idle throb rate
        goto throb

loud:
        LOW 2
        soundpin=0
        goto ST15
soft:
        LOW 0                   'stop loud vol pin hanging high
        soundpin=2
        goto ST15

'++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

'NOTES:
'the max volts applied to pin1 for the speed MUST NOT EXCEED the picaxe supply voltage
'assuming that is 5V, then a voltage divider on pin1 must reduce the motor volts to
'less than or equal to 5V at the MAXIMUM SUPPLY VOLTAGE !
```

```
'It would be prudent to reduce the voltage to a bit less than 5V

'if max volts on pin1 are less than picaxe volts (5V) then you must determine the value
'for maxspeedIN as follows:
'using a multimeter measure the picaxe supply volts (Vp) and
'the voltage on pin1 with max supply voltage (Vm1). Then
'maxspeedIN= Vm1/Vp*256. (round the result UP, and must max of 256.)
'example,  you measure Vp=4.28V and Vm1=2.65V, then enter SYMBOL maxspeedIN=159
‘++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

An additional refinement could be to reduce the throb to ‘idle’ when the speed is slowing.