

```

1  '=====
2  '
3  '  VERSATILE  ENGINE  SOUND  UNIT
4  '
5  '  PROGRAM 4 OF 4 - SOUND GENERATION (STEAM)
6  '
7  '=====
8  '  VERSION 2.0  14th October 2008           © A.F.Bond  2008
9  '
10 '  (ONLY FOR USE IN CONJUNCTION WITH v2.0 SPEED DEMAND MODULE)
11 '=====
12 '
13 ' This module uses one of the PIC's ADCs to read a speed demand voltage and
14 ' then uses that value to create a variable frequency output signal. The output
15 ' signal comprises repeated cycles of a period of pseudo white noise (to
16 ' simulate the hiss of steam) followed by a period of silence. The repetition
17 ' rate of this cycle is organised to be proportional to the speed demand voltage.
18 '
19 ' To give the engine a characteristic 'beat', a slightly different pitch of
20 ' pseudo-random noise is assigned to each of the cylinders before looping round
21 ' to repeat the sequence.
22 '
23 ' A jumper link is read at program launch and its state determines if the
24 ' cylinder selection mode is entered. If so, then the program succesively plays
25 ' a few seconds of fixed speed sound from the 1,2,3,4,5 or 6 cylinder engine
26 ' variants. Removing the link during one of these periods locks the chip into
27 ' that mode. The cylinder count value is then stored in the PICAXE's EEPROM ie
28 ' non-volatile memory, so that when the link is in the 'engine run' position the
29 ' cylinder count set by the user can be read at power up.
30 '
31 ' NOTE: In the description below be aware that PIN1, PIN2 etc are virtual
32 ' pins and not to be confused with the device's physical pins, which in PIC
33 ' parlance are refered to as 'legs'
34 '
35 '=====
36
37
38
39 'VARIABLE DEFINITIONS
40
41 symbol demand=b0      'speed demand input, read from ADC on "pin4" (leg 3)
42 symbol pause=w4       'length of pause between chuffs
43 symbol chuff=b2       'length of chuff
44 symbol cyl=b4         'number of cylinders in engine
45                       'note numbering begins at 0 as the lookup tables
46                       'have a zero'th element. The length of the tables
47                       'restricts the maximum cylinder count to six
48
49
50 'CONSTANT DEFINITIONS
51
52 symbol neutral_LED=0   'neutral LED driven from "pin0" (leg 7) high=ON
53
54 'experiment with these values to affect character of the engine
55 'values must lie in range 128 to 255, but less than 250 unlikely to be used
56 symbol p1=255         'pitch of pseudo random sound for cylinder 1
57 symbol p2=253         '..... 2
58 symbol p3=254         '..... 3

```

```

59 symbol p4=252      '..... 4
60 symbol p5=254      '..... 5
61 symbol p6=253      '..... 6
62
63
64 'CHIP PROGRAMMING DIRECTIVE (only executed when program is loaded into chip)
65
66 'Pre-load non-zero value into EEPROM storage location 0 (cylinder count)
67 data 0, (2)        'set default cylinder count to 3
68
69
70
71
72
73 '===== PROGRAM STARTS HERE =====
74
75
76 'INITIALISATION
77
78 setfreq m8          'set 8Mhz operation to yield 5uSec resolution
79
80 pause 1000          'allow time for speed input voltage to settle
81
82 init:
83 if pin3=0 then      'check jumper link
84 'CYLINDER SELECT MODE
85 high neutral_LED    'turn on LED
86 sound 1,(120,60)    'warning beep
87
88 for cyl=0 to 5      'step through all cylinder values
89   pause 2000        '1 sec pause between variants (8Mhz doubles values)
90
91   lookup cyl,(350,240,180,144,140,135),pawse 'establish speed for each demo
92   chuff=pawse/12    'duration of steam sound
93   for b7=0 to 7      'run for eight engine cycles
94     for b5= 0 to cyl 'for each cylinder in turn
95       lookup b5,(p1,p2,p3,p4,p5,p6),b6 'get steam pitch for cylinder
96       sound 1,(b6,chuff) 'make the chuff
97       pause pawse    'silence until next chuff
98     next b5
99   next b7
100   if pin3=1 then      'has user selected the current cylinder variant?
101     write 0,cyl      'YES - store value in non-volatile memory
102     sound 1,(120,10) 'acknowledge with a beep
103     goto main        'and jump out of programming mode
104   end if
105   'NO - try next cylinder count
106 next cyl
107 goto init            'no selection made, loop round again
108
109 else
110 'ENGINE RUN MODE
111 read 0, cyl          'get cylinder count previously stored in EEPROM
112 end if
113
114
115
116

```

```

117
118
119 'MAIN LOOP
120 main:
121   readadc 4,demand      'check speed demand status
122
123   'process speed demand value
124
125   pawse=demand*120 max 35000 '120 sets the throttle sensitivity (ie revs range)
126                               'the pawse calculation is limited to 35000 such
127                               'that the next calculation limits the top speed
128                               'to 5000
129
130   pawse=40000-pawse      '40000 is slowest speed
131
132
133   chuff=pawse/1000       'set chuff to fixed proportion of engine cycle
134                           'approx 50%
135
136   if demand>10 then     'check speed demand
137     'RUN
138     low 0                'switch off neutral (ie stopped) LED
139     for b5= 0 to cyl     'loop the required number of cylinders
140       lookup b5,(p1,p2,p3,p4,p5,p6),b6 'get steam pitch for cylinder
141       sound 1,(b6,chuff) 'make the chuff
142       pulsout 2, pawse   'silence until next chuff
143     next b5
144   else
145     'STOP
146     toggle neutral_LED   'flicker neutral LED to signify stopped
147     pause 80             'sets rate of flicker
148   end if
149   goto main
150
151
152
153
154 '=====
155 ' VERSION HISTORY
156 '=====
157
158 'v1.0 1st October 2008
159 'first release
160
161
162 'v2.0 14th October 2008
163 'Logarithmic lookup table function moved into the speed demand module, which
164 'allowed a longer lookup table to be used without code execution time problems
165 'as the speed demand code only has to run at the transmitter frame rate rather
166 'than the engine revolution rate. The resulting throttle response of the
167 'overall system is thereby much smoother.
168 '
169 'The speed demand processing has been reworked to give single numeric values
170 '(for user experimentation) to set the slowest speed and the revs range plus a
171 'top speed limiter which operates if the revs range multiplier is set too high.
172 '
173 'The timing for main engine loop now uses 'pulsout' (to an unused output) to
174 'create the inter-cylinder delays rather than the 'pause' statement. Much

```

```
175 'larger values for an equivalent delay result, meaning that the speed resolution
176 '(ie size of the steps between successive values) is much improved.
177 '
178 '=====
```