

PWM.BAS

This module is part of the original MMBasic library. It is reproduced here with kind permission of Hugh Buckle and Geoff Graham. Be aware it may reference functionality which has changed or is deprecated in the latest versions of MMBasic.

```
5 TW=0:TD=0:TH=0:TM=0:TS=0:TT=0:Timer=0 ' Initialise the elapsed time
variables
10 'initialise I/O pins
15 voltpin1 = 1 'set analog voltage input pin for main bus (can be 1 - 10)
20 voltpin2 = 2 'set analog voltage input pin for 9V rail (can be 1 - 10)
25 voltpin3 = 3 'set analog voltage input pin for 5V rail (can be 1 - 10)
30 pwmpin = 8 'set the pwm output pin (can be 7 - 10)
35 freqpin1 = 14 'set the frequency input pin (can be 11 - 14)
40 freq = 100 'set initial frequency to 100Hz (can be 1 - 5000)
45 duty = 50 'set initial duty cycle to 50% (can be 0 - 100 in increments of
1)
50 offset1 = 0 'use to adjust for any offset in analog input for main bus
55 offset2 = 0 'use to adjust for any offset in analog input for 9V rail
60 offset3 = 0 'use to adjust for any offset in analog input for 5V rail
65 coefficient1 = 20 'use to scale the analog input for main bus (range 0 -
66 (20 * 3.3))
70 coefficient2 = 5 'use to scale the analog input for 9V rail (range 0 -
16.5 (5 * 3.3))
75 coefficient3 = 3 'use to scale the analog input for 5V rail (range 0 -
9.9 (3 * 3.3))
80 tick_avg = 100 'number of analog voltage samples to average each second
85 tick_flag = 0
90 tick_cnt = 0
95 Dim tick_array1(tick_avg)
100 Dim tick_array2(tick_avg)
105 Dim tick_array3(tick_avg)
110 SetPin voltpin3,1 'configure pin for analog input
115 SetPin voltpin2,1 'configure pin for analog input
120 SetPin voltpin1,1 'configure pin for analog input
125 SetPin freqpin1,3 'configure pin for frequency input
130 SetPin pwmpin,10,freq,duty 'initiate PWM output
135 SetTick Cint(1000/tick_avg),1100 'set tick interrupt for tick_avg per
second
140 'initialise print locations
145 locMainBuss = 12
150 loc9VRail = 36
155 loc5VRail = 60
160 locRepFreq = 84
165 locRunFreq = 108
170 locRunDuty = 132
175 locElapsed = 156
180 locCmdString = 180
185 locCmdInput = locCmdString + 12
190 duty_str$ = "Enter duty cycle (0 - 100 for 0% - 100%), 'F' to change
```

```
frequency or 'Q' to quit"
195 freq_str$ = "Enter frequency (1 - 5000 for 1 - 5kHz), 'D' to change duty
cycle or 'Q' to quit"
200 Cls
205 volt1 = offset1 + (Pin(voltpin1) * coefficient)
210 volt2 = offset2 + (Pin(voltpin2) * coefficient)
215 volt3 = offset3 + (Pin(voltpin3) * coefficient)
220 state = 1
225 Locate 0,locMainBuss : ? "Main Buss A V+ =";
230 Locate 96,locMainBuss : ? Format$(volt1,"% .3fV ");
235 Locate 0,loc9VRail : ? "9 Volt Rail V+ =";
240 Locate 96,loc9VRail : ? Format$(volt2,"% .3fV ");
245 Locate 0,loc5VRail : ? "5 Volt Rail V+ =";
250 Locate 96,loc5VRail : ? Format$(volt3,"% .3fV ");
255 Locate 0,locRepFreq : ? "Reported frequency =";
260 Locate 120,locRepFreq : ? Format$(Pin(freqpin1),"% .0fHz ");
265 Locate 0,locRunFreq : ? "Running frequency =";
270 Locate 114,locRunFreq : ? freq; "Hz ";
275 Locate 0,locRunDuty : ? "Mark space ratio =";
280 Locate 108,locRunDuty : ? duty; "% ";
285 Locate 0,locElapsed:GoSub 10000 ' This will print the elapsed time
290 Locate 0,locCmdString : ? duty_str$
295 Do While state > 0
300   GoSub 500
305   If state = 1 Then
310     If number >= 0 And number <= 100 Then
315       duty = number : SetPin pwmpin,10,freq,duty
320     Else
325       GoSub 900 : ? number; " - Invalid duty cycle!" : Pause(1000) : GoSub
900
330     EndIf
335   ElseIf state = 2 Then
340     If number >= 1 And number <= 5000 Then
345       freq = number : SetPin pwmpin,10,freq,duty
350     Else
355       GoSub 900 : ? number; " - Invalid frequency!" : Pause(1000) : GoSub
900
360     EndIf
365   EndIf
370   Locate 96,locMainBuss : ? Format$(volt1,"% .3fV ");
375   Locate 96,loc9VRail : ? Format$(volt2,"% .3fV ")
380   Locate 96,loc5VRail : ? Format$(volt3,"% .3fV ")
385   Locate 120,locRepFreq : ? Format$(Pin(freqpin1),"% .0fHz ");
390   Locate 114,locRunFreq : ? freq; "Hz ";
395   Locate 108,locRunDuty : ? duty; "% ";
400   Locate 0,locElapsed:GoSub 10000 ' This will print the elapsed time
405 Loop
410 SetPin voltpin3,0
415 SetPin voltpin2,0
420 SetPin voltpin1,0
425 SetPin freqpin1,0
```

```
430 SetPin pwmpin,0
435 GoSub 900
459 End
499 ' get number routine
500 GoSub 900
505 Do While 1
510   If tick_flag = 1 Then
515     tick_flag = 0
520     Locate 96,locMainBuss : ? Format$(volt1,"% .3fV ");
525     Locate 96,loc9VRail : ? Format$(volt2,"% .3fV ");
530     Locate 96,loc5VRail : ? Format$(volt3,"% .3fV ");
535     Locate 120,locRepFreq : ? Format$(Pin(freqpin1),"% .0fHz   ");
540     Locate 114,locRunFreq : ? freq; "Hz ";
545     Locate 108,locRunDuty : ? duty; "% ";
550     Locate 0,locElapsed:GoSub 10000 ' This will print the elapsed time
555     Locate Len(numstr$) * 6,locCmdInput
560   EndIf
565   inpkey$ = UCase$(Inkey$)
570   If inpkey$ = "" Then GoTo 510
575   If inpkey$ = "Q" Then state = -1 : Exit
580   If inpkey$ = "D" Then
585     Locate 0,locCmdString
590     ? duty_str$
595     state = 1
600     GoSub 900
605     GoTo 510
610   EndIf
615   If inpkey$ = "F" Then
620     Locate 0,locCmdString
625     ? freq_str$
630     state = 2
635     GoSub 900
640     GoTo 510
645   EndIf
650   If inpkey$ < "0" Or inpkey$ > "9" Then
655     numlen = Len(numstr$)
660     If Asc(inpkey$) = 13 Then
665       If numlen = 0 Then number = -1 Else number = Val(numstr$)
670       Exit
675     ElseIf Asc(inpkey$) = 8 Then
680       If numlen > 0 Then
685         If numlen <> 1 Then numstr$ = Left$(numstr$, numlen - 1) Else
numstr$ = ""
690         csr = (numlen - 1) * 6
695         Locate csr,locCmdInput : ? " "; : Locate csr,locCmdInput
700       EndIf
705     EndIf
710   Else
715     numstr$ = numstr$ + inpkey$
720     ? inpkey$;
725   EndIf
```

```
730 Loop
735 GoSub 900
799 Return
899 ' clear input
900 numstr$ = ""
910 Locate 0,locCmdInput
920 ? Space$(80)
930 Locate 0,locCmdInput
999 Return
1099 ' settick interrupt routine
1100 tick_cnt = tick_cnt + 1
1110 tick_array1(tick_cnt) = offset1 + (Pin(voltpin1) * coefficient1)
1120 tick_array2(tick_cnt) = offset2 + (Pin(voltpin2) * coefficient2)
1130 tick_array3(tick_cnt) = offset3 + (Pin(voltpin3) * coefficient3)
1140 If tick_cnt <> tick_avg Then IReturn
1150 volt1 = 0
1160 volt2 = 0
1170 volt3 = 0
1180 For i = 1 To tick_avg
1190  volt1 = volt1 + tick_array1(i)
1200  volt2 = volt2 + tick_array2(i)
1210  volt3 = volt3 + tick_array3(i)
1220 Next i
1230 volt1 = volt1 / tick_avg
1240 volt2 = volt2 / tick_avg
1250 volt3 = volt3 / tick_avg
1260 tick_cnt = 0
1270 tick_flag = 1
1299 IReturn
9999 ' Elapsed time routine to update the elapsed time and print it
10000 TC=Timer:Timer=0:TT=TT+TC
10010 If TT>=1000 Then TS=TS+(TT\1000):TT=TT Mod 1000
10020 If TS>=60 Then TM=TM+(TS\60):TS=TS Mod 60
10030 If TM>=60 Then TH=TH+(TM\60):TM=TM Mod 60
10040 If TH>=24 Then TD=TD+(TH\24):TH=TH Mod 24
10050 If TD>=7 Then TW=TW+(TD\7):TD=TD Mod 7
10060 pflag=0:?"Elapsed time:";
10070 If TW>0 Then pflag=1:?" weeks,";
10080 If TD>0 Or pflag=1 Then pflag=1:?" days,";
10090 If TH>0 Or pflag=1 Then pflag=1:?" hours,";
10100 If TM>0 Or pflag=1 Then ? TM;" minutes,";
10110 ? Format$(TS," %1.0f.");Format$(TT,"%03.0f seconds ");
10120 Return
```

From:

<https://tbsassist.com/wiki/> - FotS

Permanent link:

https://tbsassist.com/wiki/doku.php?id=mmbasic_original:pwm_bas

Last update: 2024/01/19 09:39



