```
;Driver for JY-LKM1638
;Allows control of 7 segment display and LEDs.  Will read all the switch values into
;variables
;
;DS18B20                      ---> pin 3 08M2
;Output to relay              ---> pin 2 08M2 this is an input pin, but uses 'pullup' to act as output
;LKM1638 Input pin 3 (CLK)    ---> pin 7 08M2
;LKM1638 Input pin 4 (DIO)    ---> pin 6 08M2
;LKM1638 Input pin 5 (STB0)   ---> pin 5 08M2
;
;The LKM1638 uses 3 pins to control 8 seven segment displays, 8 bicolour LEDs and read
;the values from 8 pushbuttons.
;This program also reads temperature (to 0.1deg) from a DS18B20 and outputs to a relay to control a heater
;All from an 08M2!
;
;The LKM1638 module has four basic command types
;1) Write to the display in fixed address format
;2) Write to the display in auto increment address format
;3) Set the display brightness
;4) Read the pushbuttons
;
;This code should be used in conjunction with the TM1638 driver chip datasheet for the LKM1638, here
;https://docs.google.com/file/d/0B84N2SrJaybwZTgxYjM4ZmEtY2EyZi00YjVjLWIzOTctYTlhMjJkM2MxMTBl/edit
;
;The Main program loop is used as a thermostat
;Permanent temperatur readout on the LH 4 digits
;The RH 4 digits displaying either maximum temp, minimum temp or thermostat set value
;The function of the RH digits is controlled by the pushbuttons as described below
;The display brightness is also controlled by a pushbutton and PICAXE reset by another
;Maximum, minimum, set and brightness levels are retained in eeprom
:
;To use the JY-LKM1638 module for other applications, use the subroutines provided and write your code
;to call them as required
;
;Subroutines
;
;Convsettemp:
;Passed     settemp
;Returns    huns, tens, units deci, sign
;Purpose    Converts settemp variable into digits for 7seg display
;Calls      None
;
;Display:
;Passed     bank, sign, tens, units, deci
;Returns    None
;Purpose    Display 3 digits and a sign on the 7seg displays as two banks of 4 digits
;           LH 4 digits are bank 0, RH 4 digits are bank 8
;Calls      writemode, lookupchar, writedisplay, displaybrit
;
;Clearleds:
;Passed     None
;Returns    None
;Purpose    Clears all the LEDs
;Calls      writemode, writedisplay, displaybrit
;
;Clearchar:
;Passed     None
;Returns    None
;Purpose    Clears all the 7seg displays and the LEDs
;Calls      writemode, sendchar, displaybrit
;
;Sendchar:
;Passed     dataio
;Returns    None
;Purpose    Sends the bits of dataio serially to the LKM1638
;Calls      None
;
;Getkeys:
;Passed     None
;Returns    s1, s2 s3, s4, s5, s6, s7, s8 (also contained in keys)
;Purpose    Gets the individual switch values
;Calls      sendchar
;
;Lookupchar:
;Passed     char
;Returns    dispvalue
;Purpose    Takes the char variable and converts it into the correct code to display on 7seg display
;           e.g. char value 8 displays '8', char value 17 displays '-'
;           The number of characters in the lookup can be increased if requred for symbols etc.
;Calls      None
;
;Gethrtemp
;Passed     hrtemp
;Returns    deci, units, tens, huns, sign
;Purpose    Takes hrtemp and convers into individual digits for display. Correctely handling negative
;           values.  (Note: hrtemp value is different upon return)
;Calls      None
;
;Writedisplay:
;Passed     dispaddr, dispvalue
;Returns    None
;Purpose    Writes value to individual 7seg display or LED. For 7 seg, dispaddr are 8 even addresses from 0
;           to 14 and dispvalue is obtained from Lookupchar subroutine.
;           For LEDs dispaddr are 8 odd addresses from 1 to 15 and dispvalue is 0=0ff, 1=red, 2=green.
;Calls      sendchar, displaybrit
;
;Displaybrit:
;Passed     dispbrit
;Returns    None
;Purpose    The last byte in protocol to set display brightness.  From $88 min to $8f max. Min normally ok
;Calls      sendchar
;
'Writemode
;Passed     autoaddr
;Returns    None
;Purpose    Puts display into autocrement mode, where all the 7 seg and LEDs can be written to just by sending
;           data bytes.  Used in Clearchar subroutine to blank all displays
;Calls      sendchar
;
;
#picaxe 08m2
#no_data                ;Minimum(0,1), Maximum(2,3), set(4), dispbrit(5)  stored in eeprom so do not overwrite
```

```
dirsc = %00010111       ;c4, c2, c1, c0 as output
symbol clock      = c.0 ;Clock output pin
symbol dio        = c.1 ;Data input output pin
symbol strobe     = c.2 ;Strobe output pin
symbol tempio     = c.4 ;DS18B20 io
symbol s1         = bit16 ;b2 - switch 1 - Reset program
symbol s2         = bit17 ;b2 - switch 2
symbol s3         = bit18 ;b2 - switch 3 - Cycle the display brightness
symbol s4         = bit19 ;b2 - switch 4 - Decrement set temperature, limited to 0 deg
symbol s5         = bit20 ;b2 - switch 5 - Increment set temperature, limited to 63.5 deg
symbol s6         = bit21 ;b2 - switch 6 - Enter set temperature mode, press again to exit
symbol s7         = bit22 ;b2 - switch 7 - Display min temp or bold to reset min temp
symbol s8         = bit23 ;b2 - switch 8 - Display max temp or hold to reset max temp
symbol dataio     = b0 ;w0 and bit 0 to bit 7 - byte to be sent out serially to LKM1638
symbol pad        = b1 ;w0 and bit 8 to bit 15 - used as a counter byte for the serial data
symbol iobuf      = w0 ;b0, b1 - combines dataio and pad to send the serial bits
symbol keys       = b2 ;w1, bit16 to bit 23 - byte containing the switch values
;symbol           = b3 ;w1 - spare
symbol dispaddr   = b4 ;w2 - offset from display start address (odd = LED, even = 7seg)
symbol huns       = b5 ;w2 - hundreds digit
symbol tens       = b6 ;w3 - tens digit
symbol units      = b7 ;w3 - units digit
symbol deci       = b8 ;w4 - lsbyte of actual temperature
symbol whole      = b9 ;w4 - msbyte of actual temperature
symbol hrtemp     = w4 ;b8,b9 - word of actual temperature
symbol sign       = b10;w5 - sign digit
symbol settemp    = b11;w5 - bit 0=0.5deg, bits 1-8=0 to 127 deg
symbol char       = b12;w6 - lookup offset to find the 7 seg code to display
symbol bank       = b13;w6 - bank 0 is the LH 4 7seg digits, bank 8 is the RH 4 7seg displays
symbol dispvalue  = b14;w7 - the code to send to the LEDs or 7seg digit (for LEDs 0=off, 1=red, 2= green)
symbol dispbrit   = b15;w7 - used to control the brightness of the display
;symbol           = w8, b16,b17 - spare
symbol tmpry      = b18;w9 - reusable temporary byte0
symbol tmpry1     = b19;w9 - reusable temporary byte1
symbol tmpryw     = w9 ;b18, b19 - reusable temporary word
symbol maximum    = w10;b20,b21 - maximum temperature variable
symbol minimum    = w11;b22,b23 - minimum termperature variable
symbol maxmin     = b24;w12 - contol byte for LED colours
symbol delaycnt   = b25;w12 - counter used to test for long key press
;symbol           = w13, b26, b27 - spare
;
;Constants
symbol heaton     = $08;Heater output pin. Use 'pullup' command into FET to
symbol heatoff    = $00;turn input pin c.3 into output
symbol fixaddr    = $c0;Start address of LEDs and 7 seg displays
symbol autoaddr   = $40;Address to define the autoincrement display mode
symbol readmode   = $42;Address to read keys
;
;-----------------------------------------------------
init:
setfreq m32             ;To improve response from keys
high strobe             ;Ensure strobe is initially high
call clearchar          ;Clear all characters
read 0, word minimum    ;get stored min temperature
read 2, word maximum    ;get stored max temperature
read 4, settemp         ;get stored set temperature
read 5, dispbrit        ;get stored display brightness
;If no min value in eeprom then set minimum to a value to ensure it is populated on first reading
;If no max value in eeprom then leave at zero to ensure it is popluated on first reading
if maximum = 0 then : endif               ;Ensure max and min variables are
if minimum = 0 then : let minimum = 46080 : endif     ;initially populated from the current temperature
if dispbrit = 0 then : let dispbrit = $88 : endif     ;if no stored value set to minimum brightness
maxmin = 0              ;Initial value to identify no keys yet pressed
;-----------------------------------------------------
main:
tmpryw = 0                    ;Reset temporary word
setfreq m4                    ;necessary for readtemp12 command
readtemp12 tempio,hrtemp      ;Read raw value into hrtemp, lower 11 bits have temp, upper 5 have sign
setfreq m32                   ;reset freq
hrtemp = hrtemp + 880 * 16    ;Add equivalent of 55deg to temp reading so readings will be from 0 to 180
                              ;rather than -55 to 125 (MSB then holds whole degrees + 55) Easier for comparison
if hrtemp > maximum then let maximum = hrtemp: write 2, word maximum: endif   ;Set max value & write to eeprom
if hrtemp < minimum then let minimum = hrtemp: write 0, word minimum: endif   ;Set min value & write to eeprom
;
call getkeys            ;Read the keys state
;
if s1 = 1 then reset : endif  ;Reset if S1 is pressed
;
;Check if any keys pressed
select case keys
      ;
      case $80                            ;S8 pressed, show maximum
            delaycnt = delaycnt max 254 +1;increment if key held down, limit at 255
            select case delaycnt
                  case 1                  ;First time key pressed
                        maxmin = 2        ;Display maximum temp
                        call clearleds
                        dispaddr = 15     ;'Max' LED is 15
                        dispvalue = 2     ;Turn 'Max' LED green
                        call writedisplay
                  case 5                  ;Key held for >~4 secs
                        maximum = 0       ;Reset max temp to extreme
            endselect
      ;
      case $40                            ;S7 pressed, show minimum
            delaycnt = delaycnt max 254 +1;increment if key held down, limit at 255
            select case delaycnt
                  case 1                  ;First time key pressed
                        maxmin = 1        ;Display minimum temp
                        call clearleds
                        dispaddr = 13     ;'Max' LED is 13
                        dispvalue = 2     ;Turn 'Min' LED green
                        call writedisplay
                  case 5                  ;Key held for >~4 secs
                        minimum = 46080   ;Reset max temp to extreme
            endselect
      ;
      case $20                            ;S6 pressed, show/set temperature
            delaycnt = delaycnt max 254 +1;increment if key held down, limit at 255
            select case delaycnt
                  case 1                  ;First time key pressed
                        call clearleds    ;Clear LEDs
```

```
                        dispaddr = 11     ;'Set' LED is 11
                        dispvalue = 2     ;Turn 'Set' LED green
                        call writedisplay
                        maxmin = 3        ;Display 'Set' temperature
                  case 5                  ;Key held for >~4 secs
                        call clearleds    ;'Set' loop so turn 'Set' LED red
                        dispaddr = 11
                        dispvalue =1
                        call writedisplay
                        do until keys <> $20 : call getkeys :loop ;Wait for S6 to be released
                        do until keys = $20 ;Loop here until 'Set' key is pressed again
                              call getkeys
                              select case keys
                                    case $10    ;Increase value
                                          settemp = settemp max 126 + 1 ;add 0.5 deg to set value, limit max to 63.5 deg
                                          write 4, settemp  ;Update eeprom
                                    case  $8    ;Decrease value
                                          settemp = settemp min 1 - 1   ;subtract 0.5 deg from set value, all bits 0 = 0
deg
                                          write 4, settemp  ;Update eeprom
                              endselect
                              call convsettemp  ;Convert value for display
                              call display      ;Display on 7 segments
                        loop
                        call clearleds    ;Clear LEDs
                        dispaddr = 11     ;'Set' LED is 11
                        dispvalue = 2     ;Turn 'Set' LED green
                        call writedisplay
            endselect
      ;
      case $4     ;S3 brightness
            dispbrit = dispbrit max $8f + 1
            if dispbrit = $90 then : dispbrit = $88 : endif ;Max brighness $8f, so cycle back to min
            write 5, dispbrit ;Save display brightness
      ;
      else                    ;No key pressed so continue and update previous value
            delaycnt = 0      ;Key is released so reset counter to 0
endselect
;
;Test actual tempertature against set temperature
tmpry1 = settemp
tmpryw = tmpryw /2      ;Shift 0.5 deg bit into msbit of lower byte
tmpry1 = tmpry1 + 55    ;Add 55 deg to msbyte
if hrtemp < tmpryw then ;temperature is less than 'set' temperature
      pullup heaton           ;Turn heater on, using c.3 as output with pullup command and a FET
else
      pullup heatoff          ;Turn heater off, c.3 is pulled low by external pull down resistor
endif
;
;Write actual temperature to first bank of digits
bank = 0
call gethrtemp    ;Convert temperature into correct units, decimal & sign
call display
;
;Dispaly maximum, minimum temperature, 'Set' temperture or blank display on 2nd bank of digits
bank = 8                ;Second block of 4 digits
if maxmin = 3 then      ;display the 'Set' temperature
      call convsettemp
      call display
else
      lookup maxmin,(0,minimum, maximum),hrtemp
      call gethrtemp          ;Convert temperature into correct units, decimal & sign
      call display
endif
;
goto main
end
;-----------------------------------------------------
convsettemp:                        ;Converts settemp value for display
      huns = 0                      ;Not setting above 99 deg so always 0
      tens = settemp/2//100/10      ;/2 removes decimal //100/10 gets the tens only
      units = settemp/2//100//10    ;/2 removes decimal //100//10 gets the units only
      deci = settemp and 1 * 5      ;Mask off LSB, if 1 then decimal =5, if 0 then decimal = 0
      sign = 18                     ;Sign always positive for setting
return
;-----------------------------------------------------
display:                    ;Displays data on the 7 seg displays, using 2 blocks of 4 digits
      call writemode          ;Put display in write mode
      ;
      dispaddr = bank + 0     ;Set sign write address
      if huns = 1 then : sign = 1 : else : sign = sign : endif    ;Display '1' hundred or sign
      char = sign             ;Lookup the display code for sign
      call lookupchar
      call writedisplay
      ;
      dispaddr = bank + 2     ;Set tens write address
      if tens = 0 and huns = 0 then : tens = 18 : endif ;suppress leading zeros (18 is space)
      char = tens             ;Lookup the display code for tens digit
      call lookupchar
      call writedisplay
      ;
      dispaddr = bank + 4     ;Set units write address
      char = units            ;Lookup the display code for units digit
      call lookupchar
      dispvalue = dispvalue OR $80  ;Append decimal point to this digit
      call writedisplay
      ;
      dispaddr = bank + 6     ;Set decimal write address
      char = deci             ;Lookup the display code for decimal digit
      call lookupchar
      call writedisplay
      ;
      call displaybrit        ;Set brightness
return
;-----------------------------------------------------
clearleds:              ;Clear LEDs only
      call writemode    ;Display in write mode
      ;
      for dispaddr = 1 to 15 step 2 ;Increment LED addresses (odd)
            dispvalue = 0     ;0 turns LED off
            call writedisplay
      next
```

```
336       call displaybrit  ;set brightness
337 return
338 ;-----------------------------------------------------
339 clearchar:              ;Clear LEDs and 7 seg displays
340       call writemode    ;Display in write mode (default is also auto increment)
341       ;
342       low strobe        ; Strobe low
343       dataio = fixaddr  ; Set start address
344       call sendchar
345       for tmpry = 1 to 16     ;Write to all 16 addresses for LEDs and 7 seg displays
346             dataio = 0        ;Zero blanks the  display
347             call sendchar
348       next
349       high strobe       ; Strobe high, keep low to end of data
350       call displaybrit  ;set brightness
351 return
352 ;-----------------------------------------------------
353 sendchar:   ;Fundamental routine to send all characters to LKM1638 module serially
354       pad         = $ff ;Set counter
355       high clock        ;Ensure clock is high for pulseout
356       do
357         pinc.1 = bit0   ;Make c.1 the value in bit0
358         iobuf = iobuf/2 ;Shift right
359         pulsout clock,1 '1.25us (@ 32MHz) clock pulse
360       loop until pad = 0
361 return
362 ;-----------------------------------------------------
363 getkeys:    ;Reads the input keys in and places them in bits16 to bits23
364       dataio      = readmode  ; Data mode read
365       low strobe
366       call sendchar
367       input c.1               ;set c.1 as input
368       high clock              ;Ensure clock is high for pulseout
369       for tmpry = 1 to 16     ;Read in bits 0-15
370             bit0 = pinc.1     ;Make bit0 the value on c.1. Need to use c.1 as it is both in & out
371             iobuf = iobuf*2   ;Shift bit left
372             pulsout clock,1   ;Clock pulse to read next bit
373       next
374       s6 = bit3               ;Move 1st word switch values out of buffer
375       s2 = bit7
376       s5 = bit11
377       s1 = bit15
378       for tmpry = 1 to 16     ;Read in bits 16-31
379             bit0 = pinc.1     ;Make bit0 the value on b.0. Need to use c.1 as it is both in & out
380             iobuf = iobuf*2   ;Shift bit left
381             pulsout clock,1   ;Clock pulse to read next bit
382       next
383       s8 = bit3               ;Move 2nd word switch values out of buffer
384       s4 = bit7
385       s7 = bit11
386       s3 = bit15
387       output c.1              ;Return c.1 to output
388       high strobe
389 return
390 ;-----------------------------------------------------
391 lookupchar: ;Looks up the code to display the digit in 'char' on the 7 seg display
392       ;character  ( 0 , 1, 2 , 3 , 4 , 5 , 6 , 7, 8 , 9 , A , b , C , d , E , F , . , - ,  )
393       lookup char,($3f,$6,$5b,$4f,$66,$6d,$7d,$7,$7f,$6f,$77,$7c,$39,$5e,$79,$71,$80,$40,$0),dispvalue
394 return
395 ;-----------------------------------------------------
396 gethrtemp:
397       if hrtemp <> 0 then           ;0 is dummy value used to force display of dashes
398             whole = whole - 55
399             sign = 18               ;space
400             if whole >  127 then    ;top bit of msb set means negative
401                   hrtemp = - hrtemp ;2's complement equivalent to (NOT hrtemp +1)
402                   sign = 17         ;minus sign
403             endif
404             deci = deci / 16 * 10 + 8 / 16 ;8 is required to round decimal correctly
405             units = whole //100 //10
406             tens = whole //100 /10
407             huns = whole /100
408       else
409             tens = 18   ;Force all values to blanks
410             units = 18
411             deci = 18
412             sign = 18
413       endif
414 return
415 ;-----------------------------------------------------
416 writedisplay:
417       dataio = fixaddr + dispaddr   ;Increment from start address
418       low strobe
419       call sendchar
420       dataio = dispvalue             ;For LEDs 0=off, 1=red, 2= green
421       call sendchar
422       high strobe
423       call displaybrit
424 return
425 ;-----------------------------------------------------
426 displaybrit:
427       dataio      = dispbrit        ;Display control on, brightness level
428       low strobe                    ;Strobe low
429       call sendchar
430       high strobe                   ;Strobe high
431 return
432 ;-----------------------------------------------------
433 writemode:
434       dataio      = autoaddr  ;Data mode auto increment
435       low strobe              ;Strobe low
436       call sendchar
437       high strobe             ;Strobe high
438 return
439
```