```
1   ;#########################################################################################################
2   ;## I. S. P. Nagahage & E. A. A Dilrukshi
    ##
3   ;## 2018/05/14
    ##
4   ;## DFROBOT soil moisture prob calibration                                                          ##
5   ;#########################################################################################################
6
7   ;----------------------------I2C Device
    address-------------------------------------------------------
8   I2C_LCD                 equ     4Eh             ;slave address for LCD
9   I2C_ADS1                equ     92h             ;slave address for ADS1 module
10  I2C_ADS2                equ     90h             ;slave address for ADS2 module
11  I2C_SHT                 equ     88h             ;slave address for SHT3X
12
13  ;-----------------------------Port
    declaration---------------------------------------------------------
14  SDA_PIN                 bit     P1.0            ;I2C data line
15  SCL_PIN                 bit     P1.1            ;I2C clock line
16  SINK                    bit     P1.2            ;triggering
17  LED                     bit     P1.3
18  LED1                    bit     P1.5
19
20  ;-----------------------------ESP8266 serial
    communication-----------------------------------------
21  TimerReload             equ     0FDh            ;generate 9600boud rate
22
23
24  ;-----------------------------RAM data storage
    buffers-----------------------------------------------
25
26  BIT_CNT                 data    8h              ; Bank1 register for bit count
27  BYTE_CNT                data    9h              ; Bank1 register for byte count
28  SLV_ADDR                data    0Ah             ; Bank1 register for slave address
29  ADS_PIN                 data    0Bh             ; Select the analog input
30  ADS_SEL                 data    0Ch             ; select ADS module
31
32  OUTONES                 data    0Dh             ;output of ones colum in 2nd compliment to
    ascii conversion
33  OUTENS                  data    0Eh             ;output of 10s
34  OUTHUND                 data    0Fh             ;output of 100s
35  OUTHOUN                 data    10h             ;output of 1000s
36  OUTENTH                 data    11h             ;output of 10000s
37  SIGNOUT                 data    12h             ;plus or minus sign
38
39
40
41  FLAGS                   data    20h             ;location for bit flags in bit addresable area
42  NO_ACK                  bit     FLAGS.0         ;I2C no acknowledge flag
43  BUS_FAULT               bit     FLAGS.1         ;I2C bus fault flag
44  I2C_BUSY                bit     FLAGS.2         ;I2C busy flag
45  SerialComp              bit     FLAGS.3         ;serial complete
46
47  LCD_LATCH               data    21h             ;Enable latch for lcd data in bit addresable
    register
48
49  LCD_BUFF                data    30h             ;data to be written in buffer
50  ALT_RCV                 data    31h             ;BCD data convert and store
51  XMT_DAT                 data    33h             ;transmit buffer in general  memomry
52  RCV_DAT                 data    3Ch             ;recive buffer// aditionally use for misture
    level compare function
53  I2C_DELAY               data    45h             ;reserve last general purpose registers for delay
54  DlayVar                 data    46h             ;tempory erroe checking led
55  TMP_DISP                data    55h             ;tempry display memory
56
57  ;-----------------------------Auxillary RAM data storage
    buffers-----------------------------------------------
58
59  ARAN0                   xdata   00h             ;auxilarry ram starting from 00h, 2 bytes
    required for 2nd compliment
60  ARAN1                   xdata   02h
61  ARAN2                   xdata   04h
62  ARAN3                   xdata   06h
63  ARAN4                   xdata   08h
64  ARAN5                   xdata   0Ah
65  ARAN6                   xdata   0Ch
```

```
66  ARAN7                   xdata   0Eh
67
68
69  MOIST1                  xdata   28h             ;desired moisture level from web input as 2nd
    complimentry 2 bytes
70  MOIST2                  xdata   2Ah
71  MOIST3                  xdata   2Ch
72  MOIST4                  xdata   2Eh
73
74  SHT1T                   xdata   30h
75  SHT2T                   xdata   31h
76  SHT1R                   xdata   32h
77  SHT2R                   xdata   33h
78  ;---------------------------RESET-----------------------------------------------------------------
    ----------
79      org 0
80      ajmp                I2C_RESET
81
82      org                 23h;
83      ajmp                Serial                  ; jump to serial ISR
84
85
86  ;---------------------------I2C
    control---------------------------------------------------------------
87  SEND_STOP:                                      ;I2C stop condition to release the bus
88      clr             SDA_PIN                     ;Get SDA ready for stop
89      acall           Release_SCL_HIGH            ;Set clock for stop
90      mov             I2C_DELAY,#3
91      acall           DELAY                       ;Delay
92      setb            SDA_PIN                     ;Send I2C stop
93      clr             I2C_BUSY                    ;clear I2C busy status
94      ret
95
96  SEND_MSG:                                       ;Buffer @R0=Slvaddr,# of byte to be
    transffered, Data bytes
97      mov             SLV_ADDR,@R0                ;Initialize slave address
98      inc             R0                          ;Next address
99      mov             BYTE_CNT,@R0                ;Initialize BYTE_CNT
100     inc             R0                          ;Next address
101     acall           SEND_DATA                   ;send data
102     ret                                         ;return from subroutine
103
104 MASTER_CONTROLLER:
105     setb            I2C_BUSY                    ;Indicate that I2C frame is in progress
106     clr             NO_ACK                      ;clear error status flag
107     clr             BUS_FAULT
108     jnb             SCL_PIN,FAULT               ;Check for bus
109     jnb             SDA_PIN,FAULT
110     clr             SDA_PIN                     ;Begin I2C start
111     mov             I2C_DELAY,#3
112     acall           DELAY                       ;Delay
113     clr             SCL_PIN                     ;Compleate I2C start
114     mov             I2C_DELAY,#3
115     acall           DELAY                       ;Delay
116     mov             A,SLV_ADDR                  ;get slave address
117     acall           SEND_BYTE                   ;send slave address
118     ret
119
120     FAULT:
121         setb                BUS_FAULT           ;set fault status
122         ret
123
124 SEND_BYTE:                                      ;This subroutine send 1 byte information in A
    and verrify ack
125     mov             BIT_CNT,#8                  ;set bit count value
126     SB_LOOP:
127         rlc             A                       ;send one data bit
128         mov             SDA_PIN,C               ;put the data bit on pin
129         acall           Release_SCL_HIGH        ;Drive scl high
130         mov             I2C_DELAY,#3
131         acall           DELAY                   ;Delay
132
133         clr             SCL_PIN                 ;clear scl
134         mov             I2C_DELAY,#3
135         acall           DELAY                   ;Delay
136         djnz            BIT_CNT,SB_LOOP         ;repeat untill all bits sent
```

```
137
138             setb    SDA_PIN                 ;Relaese data line for acknowledge
139             acall   Release_SCL_HIGH        ;send clock for ackknowledge
140             mov     I2C_DELAY,#4
141             acall   DELAY                   ;delay 4 machine cycles
142             jnb     SDA_PIN,SB_EX           ;check for valid knowledge bit
143             setb    NO_ACK                  ;set status for no acknowledge
144
145         SB_EX:
146             clr             SCL_PIN         ;finish acknowledge bit
147             mov             I2C_DELAY,#3
148             acall           DELAY           ;Delay
149             ret
150
151     SEND_DATA:                              ;transmit multipple data bytes
152         acall   MASTER_CONTROLLER           ;Acquire bus and send slave address
153         jb      NO_ACK,SDEX                 ;Check for slave not responding
154
155         SD_LOOP:
156             mov     A,@R0                   ;Get data byte from the buffer
157             acall   SEND_BYTE               ;send next data byte
158             inc     R0                      ;adavance buffer point
159             jb      NO_ACK,SDEX             ;check for slave not responding
160             djnz    BYTE_CNT,SD_LOOP        ;all bytes sent?
161
162         SDEX:
163             acall   SEND_STOP               ;Done. send I2C stop
164             ret
165
166     TRANSFER:                               ;This subrountine copies data from EPROM
        reference by dptr
167         clr     A
168
169         movc    A,@A+DPTR                   ;moves content of DPTR into A
170         mov     @R1,A                       ;copies A into buffer
171         inc     R1                          ;next address
172         inc     DPTR                        ;next location
173         clr     A                           ;clear ACC
174
175         movc    A,@A+DPTR                   ;moves content of DPTR into A
176         mov     @R1,A                       ;copies A into buffer
177         mov     R0,A                        ;copies A into R0 (# of bytes)
178         inc     R1                          ;next address
179         inc     DPTR                        ;next location
180         clr     A                           ;clears A
181
182         NEXT:
183             movc    A,@A+DPTR               ;moves content of DPTR into A
184             dec     R0                      ;decrease #of remaiining bytes
185             mov     @R1,A                   ; copies A into buffer
186             inc     R1                      ;next address
187             inc     DPTR                    ;next location
188             clr     A                       ;clears A
189             cjne    R0,#0,NEXT              ;compre # of bytes remaing
190             ret
191
192     RECV_MSG:
193         mov     SLV_ADDR,@R1                ;moves SLV_ADDR from buffer R0 points to
194         inc     R1                          ;Next buffer location
195         mov     BYTE_CNT,@R1                ;moves BYTE_CNT value into memory location
196         acall   RCV_DATA                    ;calls receive data subroutine
197         ret                                 ;returns from receive msg subroutine
198
199     RECV_BYTE:
200         mov     BIT_CNT,#8                  ;set bit count
201
202         RB_LOOP:
203             acall   Release_SCL_HIGH        ;read one data bit
204             mov     I2C_DELAY,#3
205             acall   DELAY                   ;Delay
206             mov     C,SDA_PIN               ;get data bit from pin
207             rlc     A                       ;rotate bit into result byte
208             clr     SCL_PIN                 ;clear scl pin
209             mov     I2C_DELAY,#3
210             acall   DELAY                   ;Delay
211             djnz    BIT_CNT,RB_LOOP         ; repeat until all bit receive
```

```
212                 push            ACC                 ;save accumulator
213                 mov             A,BYTE_CNT          ;copies byte count into A
214                 cjne            A,#1,RB_ACK         ;check for last byte of frame
215                 setb            SDA_PIN             ;send no acknowledge on last byte
216                 sjmp            RB_ACLK             ;No ACK onlast byte; jump to RB_ACK
217
218                 RB_ACK:
219                     clr         SDA_PIN             ;send acknowledge bit
220
221                     RB_ACLK:
222                         acall   Release_SCL_HIGH    ;send acknowlwdge clock
223                         pop     ACC                 ;restore accumulator
224                         mov     I2C_DELAY,#3
225                         acall   DELAY               ;Delay
226                         clr     SCL_PIN             ;clear scl pin
227                         setb    SDA_PIN             ;clear acknowledge bit
228                         mov     I2C_DELAY,#4
229                         acall   DELAY               ;Delay
230                         ret                         ;Return from RECV_BYTE
231
232
233   RCV_DATA:
234       inc                   SLV_ADDR              ;set for READ of slave
235       acall                 MASTER_CONTROLLER     ;acquire bus and send slave address
236       jb                    NO_ACK,RDEX           ;check for slave not responding
237
238           RDLoop:
239               acall         RECV_BYTE             ;receive next data byte
240               mov           @R0,A                 ;save data byte in buffer
241               inc           R0                    ;advance buffer point
242               djnz          BYTE_CNT,RDLoop       ;Repeat untill all bytes received
243
244                   RDEX:
245                       acall SEND_STOP             ;done, send an I2C stop
246                       ret                         ;Return from RCV_DATA subroutine
247
248   Release_SCL_HIGH:
249        setb                 SCL_PIN
250        jnb                  SCL_PIN,$
251        ret
252
253   DELAY:
254        djnz                 I2C_DELAY,DELAY       ;delay for I2C bus
255        ret
256
257   ;*************************************LCD in I2C bus
      ********************************************
258
259
260   ;-----------------------------upper nibble
      conversion-----------------------------------------------
261
262   WRITE_2_NIBBLES:
263       mov                   A,LCD_BUFF
264       orl                   LCD_LATCH,#0F0h       ;check the value of upper nibble
265       orl                   A,#0Fh                ;Don't affect bits 0-3
266       anl                   LCD_LATCH,A           ;High nibble to display
267       acall                 LCD_LATCH_E           ;Latching the data
268       mov                   A,LCD_BUFF            ;mask lower nibble to write
269       swap                  A                     ;second nibble
270       orl                   LCD_LATCH,#0F0h       ;Bits 4...7 <- 1
271       orl                   A,#0Fh                ;Don't affect bits 0...3
272       anl                   LCD_LATCH,A           ;Low nibble to display
273       acall                 LCD_LATCH_E           ;Latching the data
274       ret
275
276   ;--------------------------------LCD
      Initialization-------------------------------------------------
277   INIT_LCD:
278       mov                   SLV_ADDR,#I2C_LCD     ;lcd slave address loaded
279       acall                 MASTER_CONTROLLER
280       mov                   LCD_LATCH,#28h        ;RS clear for commnd,R/W clear for write,Clear
      enable
281       acall                 LCD_LATCH_E           ;Latching the data
282       mov                   LCD_BUFF,#28h         ;LCD intializing command
283       acall                 WRITE_2_NIBBLES
```

```
284         mov                     LCD_BUFF,#0Ch       ;Dispaly on and cursor off
285         acall                   WRITE_2_NIBBLES
286         mov                     LCD_BUFF,#06h       ;Increment cursor
287         acall                   WRITE_2_NIBBLES
288         acall                   INTRO
289         acall                   SEND_STOP
290         ret
291
292     ;--------------------------------LCD Latch enable disable
        routing---------------------------------
293     LCD_LATCH_E:
294         setb                    LCD_LATCH.2
295         mov                     A,LCD_LATCH
296         acall                   SEND_BYTE           ;should be edited*******
297         clr                     LCD_LATCH.2
298         mov                     A,LCD_LATCH
299         acall                   SEND_BYTE
300         ret
301
302     ;--------------------------------LCD
        Clear-----------------------------------------------------------
303     CLEAR_LCD:
304         mov                     LCD_BUFF,#01h       ;clear lcd screen
305         acall                   WRITE_COMMAND       ;
306         acall                   LCDDelay
307         ret
308
309     ;--------------------------------LCD Data
        write-----------------------------------------------------
310     WRITE_TEXT:
311         setb                    LCD_LATCH.0         ;RS set bit for data
312         acall                   WRITE_2_NIBBLES     ;
313         ret
314
315     ;--------------------------------LCD Command
        write-----------------------------------------------------
316     WRITE_COMMAND:
317         clr                     LCD_LATCH.0         ;RS set for command
318         acall                   WRITE_2_NIBBLES     ;
319         ret
320
321
322     ;**************************************ADS in I2C bus
        ***********************************************
323     ADS_CONFIG:
324         mov                     R3,#3
325         mov                     R4,#4
326         mov                     ADS_SEL,#I2C_ADS1
327     ADC:
328         mov                     XMT_DAT,ADS_SEL     ;load ads2 module address
329         ajmp                    AN0
330
331
332                 AN0:
333                     cjne                R3,#3,AN1
334                     mov                 ADS_PIN,#11000011b      ;single shot with dissable
        comparator, A0
335                     acall               ADS
336                     dec                 R3
337                     ajmp                ADC
338                     ret
339
340                 AN1:
341                     cjne                R3,#2,AN2
342                     mov                 ADS_PIN,#11010101b      ;single shot with dissable
        comparator, A1
343                     acall               ADS
344                     dec                 R3
345                     ajmp                ADC
346                     ret
347
348                 AN2:
349                     cjne                R3,#1,AN3
350                     mov                 ADS_PIN,#11100101b      ;single shot with dissable
        comparator, A2
351                     acall               ADS
```

```
352                     dec                     R3
353                     ajmp                    ADC
354                     ret
355
356                 AN3:
357                     mov                     ADS_PIN,#11110101b      ;single shot with dissable
    comparator, A3
358                     acall                   ADS
359                     ret
360
361
362 ADS:
363     mov                     XMT_DAT,ADS_SEL         ;ADS module address load
364     mov                     XMT_DAT+1,#3h           ;Write address,address pointer register and
    then configuration register
365     mov                     XMT_DAT+2,#00000001b    ;select configration register
366     mov                     XMT_DAT+3,ADS_PIN       ;single shot with dissable comparator and
    select Analog input
367     mov                     XMT_DAT+4,#10000011b    ;after setting the lower byte of
    configuration register
368     acall                   CONFIGSEND
369           CONREADY:
370                     mov             XMT_DAT+1,#2h
371                     mov             R0,#RCV_DAT
372                     mov             R1,#XMT_DAT
373                     acall           RECV_MSG
374                     acall           SEND_STOP
375                     mov             A,RCV_DAT
376                     clr             C
377                     rlc             A
378                     jnc             CONREADY
379                     clr             C
380
381     mov                     XMT_DAT+1,#1h           ;Write address,address pointer register and
    then configuration register
382     mov                     XMT_DAT+2,#00000000b    ;select conversion register
383     acall                   CONFIGSEND
384     mov                     XMT_DAT+1,#2h
385     mov                     R0,#RCV_DAT
386     mov                     R1,#XMT_DAT
387     acall                   RECV_MSG
388     acall                   SEND_STOP
389     acall                   AARAN0
390     ret
391
392 CONFIGSEND:
393     mov                     R0,#XMT_DAT
394     acall                   SEND_MSG
395     acall                   SEND_STOP
396     mov                     R7,#1h
397     ret
398
399
400 SECOMPL:
401     mov                     dptr,#ARAN0
402     movx                    A,@dptr
403     mov                     RCV_DAT,A
404     inc                     dptr
405     movx                    A,@dptr
406     mov                     RCV_DAT+1,A
407     mov                     A,RCV_DAT               ;
408     rlc                     A
409     jc                      NEG                     ;if carry is there, it shpild be a negative
    number
410     mov                     SIGNOUT,#"+"            ;ascii pluss
411     ajmp                    BIN2BCD
412     ret
413
414                     NEG:
415                         mov         SIGNOUT,#"-" ;ascii minus
416                         clr         C            ;clear the carry
417                         mov         A,RCV_DAT+1  ;move lower byte
418                         cpl         A            ;compliment it
419                         add         A,#1         ;add one
420                         mov         RCV_DAT+1,A  ;result store in RCV_DATA+1 memory space
421                         mov         A,RCV_DAT    ;
```

```
422                             cpl         A
423                             addc        A,#0
424                             mov         RCV_DAT,A
425
426     BIN2BCD:
427         mov                     OUTONES,#00         ;clear memomry
428         mov                     OUTENS,#00
429         mov                     OUTHUND,#00
430         mov                     OUTHOUN,#00
431         mov                     OUTENTH,#00
432
433         mov                     B,#10
434         mov                     A,RCV_DAT+1         ;get low data byte
435         div                     AB                  ;divide it by 10
436         mov                     OUTONES,B           ;save the remainder in 1s colum
437         mov                     B,#10
438         div                     AB                  ;divide quateint by 10
439         mov                     OUTENS,B            ;save the remainder
440         mov                     OUTHUND,A           ;save the last dgit
441         mov                     A,RCV_DAT           ;get the higher byte
442         cjne                    A,#00,HIBYT         ;
443         ajmp                    ASCIWRITE           ;LCD display
444
445     HIBYT:
446         mov                     A,#6                ;if there is a value in higer byte decerse it
        and add
447         add                     A,OUTONES           ;256 to corresponding digits
448         mov                     B,#10
449         div                     AB
450         mov                     OUTONES,B           ;save the 1s colom after adding 6
451         add                     A,#5
452         add                     A,OUTENS
453         mov                     B,#10
454         div                     AB
455         mov                     OUTENS,B            ;save the remainder after adding 5
456         add                     A,#2
457         add                     A,OUTHUND
458         mov                     B,#10               ;devide by 10
459         div                     AB
460         mov                     OUTHUND,B           ;save the remainder after adding 2
461         add                     A,OUTHOUN
462         mov                     OUTHOUN,A
463         djnz                    RCV_DAT,HIBYT       ;decrease and do the process if highr byte is
        still not zero
464         mov                     B,#10               ;if zero, stop the loop and add to 1000 clom
        and 10000 colom
465         mov                     A,OUTHOUN
466         div                     AB
467         mov                     OUTHOUN,B
468         mov                     OUTENTH,A
469
470     ASCIWRITE:
471         mov                     A,OUTONES
472         add                     A,#30h
473         mov                     OUTONES,A
474         mov                     A,OUTENS
475         add                     A,#30h
476         mov                     OUTENS,A
477         mov                     A,OUTHUND
478         add                     A,#30h
479         mov                     OUTHUND,A
480         mov                     A,OUTHOUN
481         add                     A,#30h
482         mov                     OUTHOUN,A
483         mov                     A,OUTENTH
484         add                     A,#30h
485         mov                     OUTENTH,A
486         acall                   SOIL_TMP_HUM
487         mov                     SLV_ADDR,#I2C_LCD   ;lcd slave address loaded
488         acall                   MASTER_CONTROLLER
489         mov                     LCD_BUFF,#0C0h
490         acall                   WRITE_COMMAND
491         mov                     LCD_BUFF,SIGNOUT
492         acall                   WRITE_TEXT
493         mov                     LCD_BUFF,OUTENTH
494         acall                   WRITE_TEXT
```

```
495         mov                     LCD_BUFF,OUTHOUN
496         acall                   WRITE_TEXT
497         mov                     LCD_BUFF,OUTHUND
498         acall                   WRITE_TEXT
499         mov                     LCD_BUFF,OUTENS
500         acall                   WRITE_TEXT
501         mov                     LCD_BUFF,OUTONES
502         acall                   WRITE_TEXT
503         acall                   SEND_STOP
504         ret
505     ;*************************************Auxilary ram for temporarary data storage
        ************************
506     AARAN0:
507         cjne                    R4,#4h,AARAN1
508         mov                     dptr,#ARAN0
509         dec                     R4
510         ajmp                    datamov
511         ret
512
513     AARAN1:
514         cjne                    R4,#3h,AARAN2
515         mov                     dptr,#ARAN1
516         dec                     R4
517         ajmp                    datamov
518         ret
519
520     AARAN2:
521         cjne                    R4,#2h,AARAN3
522         mov                     dptr,#ARAN2
523         dec                     R4
524         ajmp                    datamov
525         ret
526
527     AARAN3:
528         cjne                    R4,#1h,datamovExit
529         mov                     dptr,#ARAN3
530         dec                     R4
531     datamov:
532         mov                     A,RCV_DAT
533         movx                    @dptr,A
534         inc                     dptr
535         mov                     A,RCV_DAT+1
536         movx                    @dptr,A
537         ret
538     datamovExit:
539         ajmp check
540         ret
541
542
543     ;*************************************SHT3X Temp & Humidity
        ****************************************
544
545     SHT_INIT:
546         mov                     XMT_DAT,#I2C_SHT        ;SHT module address load
547         mov                     XMT_DAT+1,#2h           ;have to send two_byte single shot command
548         mov                     XMT_DAT+2,#2Ch          ;MSB
549         mov                     XMT_DAT+3,#6h           ;single shot with dissable comparator and
        select Analog input
550         mov                     R0,#XMT_DAT
551         acall                   SEND_MSG
552         acall                   SEND_STOP
553         mov                     XMT_DAT+1,#5h
554         mov                     R0,#RCV_DAT
555         mov                     R1,#XMT_DAT
556         acall                   RECV_MSG
557         acall                   SEND_STOP
558         mov                     dptr,#SHT1T
559         mov                     A,RCV_DAT
560         movx                    @dptr,A
561         inc                     dptr
562         mov                     A,RCV_DAT+1
563         movx                    @dptr,A
564         inc                     dptr
565         mov                     A,RCV_DAT+3
566         movx                    @dptr,A
567         inc                     dptr
```

```
568         mov                     A,RCV_DAT+4
569         movx                    @dptr,A
570
571
572         mov                     dptr,#SHT1T
573         movx                    A,@dptr
574         mov                     RCV_DAT,A
575         inc                     dptr
576         movx                    A,@dptr
577         mov                     RCV_DAT+1,A
578         mov                     R7,#2h
579         acall                   BIN2BCD
580         acall                   ESP8266
581         acall                   LED2
582         mov                     dptr,#SHT1R
583         movx                    A,@dptr
584         mov                     RCV_DAT,A
585         inc                     dptr
586         movx                    A,@dptr
587         mov                     RCV_DAT+1,A
588         mov                     R7,#3h
589         acall                   BIN2BCD
590         acall                   ESP8266
591         acall                   LED2
592
593         ret
594
595     ;*************************************ESP8266 input ****************************************
596
597     ESP8266:
598         mov                     IE, #90h            ;10010000-enable serial intrupt
599         mov                     DPTR,#smsg1         ;
600         acall                   SBUFLOAD
601         mov                     R0,#ALT_RCV
602         acall                   CRLF
603
604     SerialR:
605         jnb                     RI,$
606         clr                     RI
607         acall                   SINKCHK
608         ajmp                    ATCWJAP
609         ret
610
611     Serial:
612         reti
613
614
615
616     SBUFLOAD:
617         clr                     TI
618         clr                     A                   ;clear Accumulator for any previous data
619         movc                    A,@A+DPTR           ;load the first character in accumulator
620         jz                      sexit               ;go to exit if zero
621         mov                     SBUF,A
622         jnb                     TI,$        ;
623         inc                     DPTR                ;increment data pointer
624         sjmp                    SBUFLOAD            ;jump back to send the next character
625
626             sexit:
627                 ret                                 ;End of routine
628
629     ATCWJAP:
630         mov                     DPTR,#smsg2         ;
631         acall                   SBUFLOAD
632         acall                   COMASUB
633         mov                     DPTR,#smsg3
634         acall                   SBUFLOAD
635         acall                   COMASUB
636         acall                   COMASUB1
637         acall                   COMASUB
638         mov                     DPTR,#smsg4
639         acall                   SBUFLOAD
640         acall                   COMASUB
641         mov                     R0,#ALT_RCV
642         acall                   CRLF
643         acall                   SINKCHK
```

```
644       ATCWJAPR:
645               jnb                     RI,$
646               clr                     RI
647               ;mov                        R0,#ALT_RCV
648               ;acall                  CRLF
649               ;acall                  SINKCHK
650               ajmp                    ATCIPSTRT
651               ret
652
653   CRLF:
654       mov                 SBUF,#0Dh
655       jnb                 TI,$
656       clr                 TI
657       mov                 SBUF,#0Ah
658       jnb                 TI,$
659       clr                 TI
660       ret
661
662   ATCIPSTRT:
663       mov                     R0,#ALT_RCV
664       mov                     DPTR,#smsg5         ;
665       acall                   SBUFLOAD
666       acall                   COMASUB
667       mov                     DPTR,#smsg6
668       acall                   SBUFLOAD
669       acall                   COMASUB
670       acall                   COMASUB1
671       acall                   COMASUB
672       mov                     DPTR,#smsg7
673       acall                   SBUFLOAD
674       acall                   COMASUB
675       acall                   COMASUB1
676       mov                     DPTR,#smsg8
677       acall                   SBUFLOAD
678       mov                     R0,#ALT_RCV
679       acall                   CRLF
680       acall                   SINKCHK
681       ATCIPSTRTR:
682               jnb                     RI,$
683               clr                     RI
684               ajmp                    ATCIPSEND
685               ret
686   ATCIPSEND:
687       mov                     R0,#ALT_RCV
688       mov                     DPTR,#smsg9
689       acall                   SBUFLOAD
690       acall                   CRLF
691       jnb                     RI,$
692       clr                     RI
693       acall                   SELECTR
694       mov                     SBUF,SIGNOUT
695       jnb                     TI,$
696       clr                     TI
697       mov                     SBUF,OUTENTH
698       jnb                     TI,$
699       clr                     TI
700       mov                     SBUF,OUTHOUN
701       jnb                     TI,$
702       clr                     TI
703       mov                     SBUF,OUTHUND
704       jnb                     TI,$
705       clr                     TI
706       mov                     SBUF,OUTENS
707       jnb                     TI,$
708       clr                     TI
709       mov                     SBUF,OUTONES
710       jnb                     TI,$
711       clr                     TI
712       mov                     SBUF,#' '
713       jnb                     TI,$
714       clr                     TI
715       mov                     R0,#ALT_RCV
716       acall                   CRLF
717       acall                   CRLF
718       clr                     RI
719       ATCIPSENDR:
```

```
720                 mov                     DPTR,#smsg13
721                 acall                   SBUFLOAD
722                 acall                   CRLF
723                 jnb                     RI,$
724                 clr                     RI
725                 ret
726
727     SELECTR:
728         cjne                    R6,#3,SELECTR1
729         mov                     DPTR,#smsg10
730         acall                   SBUFLOAD
731         dec                     R6
732         ret
733
734         SELECTR1:
735         cjne                    R6,#2,SELECTR2
736         mov                     DPTR,#smsg11
737         acall                   SBUFLOAD
738         dec                     R6
739         ret
740
741             SELECTR2:
742             mov                     DPTR,#smsg12
743             acall                   SBUFLOAD
744             ret
745
746
747
748     COMASUB:
749         mov                     SBUF,#'"'
750         jnb                     TI,$
751         clr                     TI
752         ret
753
754     COMASUB1:
755         mov                     SBUF,#','
756         jnb                     TI,$
757         clr                     TI
758         ret
759
760     ;_____________________MAIN
        PROGRAM_____________________________________________________________________
761
762     I2C_RESET:
763         mov                     SP,#56H             ;set stack to general porpose register 57H
764         acall                   INIT_LCD            ;lcd initialization
765         mov                     TMOD,#20h           ;timer1, mode 2(auto-reload)
766         mov                     TH1, #0FDh          ;9600 baud rate
767         mov                     SCON,#50h           ;8-bit, 1 stop, REN enable
768         setb                    TR1                 ;starts timer1
769     ADS_set:
770         mov                     R6,#3h
771         mov                     IE, #00h            ;00000000-disable all interupts serial intrupt
772         acall                   ADS_CONFIG
773
774     check:
775         mov                     R7,#01h
776         acall                   SECOMPL
777         acall                   ESP8266
778
779         acall                   SHT_INIT
780
781
782     LED11:
783         setb                    LED
784         acall                   Delay1
785         cpl                     LED
786         acall                   Delay1
787         ajmp                    ADS_set
788         ret
789
790     SINKCHK:
791         setb                    SINK
792         acall                   Delay1
793         cpl                     SINK
794         acall                   Delay1
```

```
795         ret
796
797
798     LED2:
799         setb                    LED1
800         acall                   Delay1
801         cpl                     LED1
802         acall                   Delay1
803         ret
804
805
806     Delay1:                                         ;1-second dlay (1666666 cycles)
807         mov DlayVar,#250
808         mov DlayVar+1,#250
809         mov DlayVar+2,#5
810     DlayLoop:
811         djnz    DlayVar,DlayLoop
812         djnz    DlayVar+1,DlayLoop
813         djnz    DlayVar+2,DlayLoop
814         ret
815
816     LCDDelay:
817         mov     DlayVar+3,#255
818         mov     DlayVar+4,#3
819         DlayLoop1:
820         djnz    DlayVar+3,DlayLoop1
821         djnz    DlayVar+4,DlayLoop1
822         ret
823
824     DISPLAY:
825         clr                     A               ;clear Accumulator for any previous data
826         movc                    A,@A+DPTR       ;load the first character in accumulator
827         jz                      exit            ;go to exit if zero
828         mov                     LCD_BUFF,A
829         acall                   WRITE_TEXT      ;send first char
830         inc                     DPTR            ;increment data pointer
831         sjmp                    DISPLAY         ;jump back to send the next character
832
833             exit:
834                 ret                                             ;End of routine
835
836
837     INTRO:
838         mov                     LCD_BUFF,#081h  ;
839         acall                   WRITE_COMMAND   ;
840         mov                     DPTR,#msg1      ;
841         acall                   DISPLAY         ;
842         mov                     LCD_BUFF,#0C0h  ;
843         acall                   WRITE_COMMAND   ;
844         mov                     DPTR,#msg2      ;
845         acall                   DISPLAY         ;
846         acall                   SEND_STOP
847         ret
848
849
850         ret
851
852     SOIL_TMP_HUM:
853         mov                     SLV_ADDR,#I2C_LCD   ;lcd slave address loaded
854         acall                   MASTER_CONTROLLER
855         acall                   CLEAR_LCD
856         mov                     LCD_BUFF,#080h  ;
857         acall                   WRITE_COMMAND   ;
858
859         SOIL:
860         cjne                    R7,#01h,TMP
861         mov                     DPTR,#msg20     ;
862         acall                   DISPLAY         ;
863         acall                   SEND_STOP
864         ret
865
866         TMP:
867         cjne                    R7,#02h,HUM
868         mov                     DPTR,#msg21     ;
869         acall                   DISPLAY         ;
870         acall                   SEND_STOP
```

```
871         ret
872
873         HUM:
874         mov                     R7,#00h
875         mov                     DPTR,#msg22         ;
876         acall                   DISPLAY             ;
877         acall                   SEND_STOP
878         ret
879
880
881     Error:
882         mov                     SLV_ADDR,#I2C_LCD   ;lcd slave address loaded
883         acall                   MASTER_CONTROLLER
884         mov                     LCD_BUFF,#084h      ;
885         acall                   WRITE_COMMAND       ;
886         mov                     DPTR,#msg3          ;
887         acall                   DISPLAY             ;
888         mov                     LCD_BUFF,#0C0h      ;
889         acall                   WRITE_COMMAND       ;
890         mov                     DPTR,#msg4          ;
891         acall                   DISPLAY             ;
892         acall                   SEND_STOP
893         ret
894
895
896     msg1: db "Watering Robot",00h;
897     msg2: db "Systm Initial...",00h;
898
899     msg3: db "Error.",00h;
900     msg4: db "I2C bus fault...",00h;
901
902     msg20:db "Soil Probe 1: ",00h;
903
904     msg21:db "Temperature: ",00h;
905
906     msg22:db "Air Humidity: ",00h;
907
908     msg: db ":  :       /",00h;
909     msgm: db "System standby",00h;
910
911
912     ;***********************************Serial ESP commands***********************
913
914     smsg1: db "AT",00h;     AT command expecting OK from module
915     smsg2: db "AT+CWJAP=",00h
916     smsg3: db "Buffalo-G-5474",00h
917     smsg4: db "8rve5afib6ff6",00h
918     smsg5: db "AT+CIPSTART=",00h
919     smsg6: db "TCP",00h
920     smsg7: db "184.106.153.149",00h
921     smsg8: db "80",00h
922     smsg9: db "AT+CIPSEND=55",00h
923     smsg10: db "GET /update?k                    =",00h
924     smsg11: db "GET /update?k                    ",00h
925     smsg12: db "GET /update?k                    ",00h
926     smsg13: db "AT+CIPCLOSE",00h
927
928     end
929
930
931
932
933
934
```