

```

1 ;=====
2 ; LAB 10 - EEPROM Keypad Logger with 'S' Flash & RB7 Start
3 ; Jacob Horsley - RCET - Fifth Semester
4 ; Git: https://github.com/horsjacoll7/Lab10
5 ;Lab 10 EEPROM
6 ;=====
7 #include <xc.inc>
8 ;-----
9 ; Variables (Bank 0)
10 ;-----
11 PSECT udata_bank0
12 _ADDRESS: DS 1 ; EEPROM address to read from or write to
13 _DATA: DS 1 ; Data to write to EEPROM
14 POSITION: DS 1 ; 0-9 write pointer for tracking stored key count
15 TEMP: DS 1 ; Delay temp + saved row value during key scanning
16 TEMP2: DS 1 ; Delay temp2 for longer delays
17 SAVE_W: DS 1 ; ISR context save for W register
18 SAVE_STATUS: DS 1 ; ISR context save for STATUS register
19 DUMP_GIE_SAVE: DS 1 ; Save GIE during dump operation
20 STATE: DS 1 ; 0 = flash S, 1 = keyscan mode selector
21 STOP: DS 1 ;FOR THE STOPPING OF WRITING to EEPROM
22 ;-----
23 ; Reset & Interrupt vectors
24 ;-----
25 PSECT resetVect, class=CODE, delta=2
26     GOTO Start ; Jump to the start of the program on reset
27 PSECT isrVect, class=CODE, delta=2
28     GOTO INTERRUPT ; Jump to interrupt service routine on interrupt
29 ;-----
30 ; Code section
31 ;-----
32 PSECT code, class=CODE, delta=2
33 ;=====
34 ; INITIALISATION
35 ;=====
36 Start:
37     ;--- Bank 1 -----
38     BSF STATUS,5 ; Select Bank 1
39     BCF STATUS,6 ; Ensure Bank 1 is selected (RP1=0, RP0=1)
40     MOVLW 0xFF ; Load 0xFF into W
41     MOVWF TRISB ; Set PORTB as all inputs
42     CLRF TRISA ; Set PORTA as all outputs
43     CLRF TRISC ; Set PORTC as all outputs
44     MOVLW 0xFF ; Load 0xFF into W
45     MOVWF WPUB ; Enable weak pull-ups on PORTB
46     MOVLW 0x30 ; Load 0x30 into W (for RB5 and RB4 interrupts)
47     MOVWF IOCB ; Enable interrupt-on-change for RB5 and RB4
48     CLRF OPTION_REG ; Clear OPTION_REG (enables pull-ups, sets prescaler)
49     CLRF PSTRCON ; Clear parallel slave port control
50     ;--- Bank 3 -----
51     BSF STATUS,6 ; Select Bank 3 (RP1=1, RP0=1)
52     CLRF ANSEL ; Disable analog inputs on PORTA
53     CLRF ANSELH ; Disable analog inputs on PORTB
54     ;--- Bank 2 -----
55     BCF STATUS,5 ; Select Bank 2 (RP1=1, RP0=0)
56     CLRF CM2CON1 ; Disable comparator module 2
57     ;--- Bank 0 -----
58     BCF STATUS,6 ; Select Bank 0 (RP1=0, RP0=0)
59     CLRF PORTA ; Clear PORTA outputs
60     CLRF PORTB ; Clear PORTB outputs
61     CLRF PORTC ; Clear PORTC outputs
62     CLRF CCP1CON ; Disable CCP1 module
63     CLRF CCP2CON ; Disable CCP2 module
64     CLRF RCSTA ; Disable serial port receiver
65     CLRF SSPCON ; Disable synchronous serial port
66     CLRF T1CON ; Disable Timer1
67     MOVLW 0x88 ; Load 0x88 into W (enable GIE and RBIE)
68     MOVWF INTCON ; Enable global and PORTB change interrupts
69     ;--- Initialise variables -----
70     CLRF _ADDRESS ; Clear EEPROM address
71     CLRF _DATA ; Clear data to write
72     MOVLW 0x0A ; Load 10 into W
73     MOVWF POSITION ; Set initial position to 10 (beyond 0-9 range)
74     CLRF STATE ; start in flash-S mode (STATE=0)
75 ;=====

```

```

76 ; MAIN LOOP
77 ;=====
78 MAINLOOP:
79     BTFSC STATE,0 ; Test if STATE bit 0 is set (keyscan mode)
80     GOTO KEYSKAN_MODE ; If set, go to keyscan mode
81     GOTO FLASH_S_MODE ; Otherwise, go to flash S mode
82 ;=====
83 ; FLASH 'S' MODE - wait for RB7 press
84 ;=====
85 FLASH_S_MODE:
86     MOVLW 0x20 ; space (off) ASCII code
87     MOVWF PORTC ; Display space on 7-segment (turn off)
88     CALL DELAY_LONG ; Call long delay
89     MOVLW 0x53 ; 'S' ASCII code
90     MOVWF PORTC ; Display 'S' on 7-segment
91     CALL DELAY_LONG ; Call long delay
92     BTFSC PORTB,7 ; Test if RB7 is pressed (0=pressed)
93     GOTO FLASH_S_MODE ; If not pressed, continue flashing
94     CALL DELAY ; Debounce delay
95     BTFSC PORTB,7 ; Check again if RB7 is still pressed
96     GOTO FLASH_S_MODE ; If not, continue flashing
97     BSF STATE,0 ; Set STATE to 1 (enter keyscan mode)
98     CLRF POSITION ; Reset position to 0
99     GOTO MAINLOOP ; Return to main loop
100 ;=====
101 ; KEYSKAN MODE
102 ;=====
103 KEYSKAN_MODE:
104     BCF PORTA,5 ; Clear RA5 (possibly for LED or indicator)
105     ;--- Row 3 -----
106     MOVLW 0x06 ; Row 3 select value
107     MOVWF PORTA ; Select row 3 on PORTA
108     MOVWF TEMP ; Save row value in TEMP
109     CALL DELAY ; Delay for settling
110     BTFSS PORTB,3 ; Check column 3 (RB3)
111     GOTO DISP_9 ; If pressed, handle '9'
112     BTFSS PORTB,2 ; Check column 2 (RB2)
113     GOTO DISP_8 ; If pressed, handle '8'
114     BTFSS PORTB,1 ; Check column 1 (RB1)
115     GOTO DISP_7 ; If pressed, handle '7'
116     BTFSS PORTB,0 ; Check column 0 (RB0)
117     GOTO DISP_C ; If pressed, handle 'C'
118
119     ;--- Row 2 -----
120     MOVLW 0x05 ; Row 2 select value
121     MOVWF PORTA ; Select row 2 on PORTA
122     MOVWF TEMP ; Save row value in TEMP
123     CALL DELAY ; Delay for settling
124     BTFSS PORTB,3 ; Check column 3
125     GOTO DISP_6 ; If pressed, handle '6'
126     BTFSS PORTB,2 ; Check column 2
127     GOTO DISP_5 ; If pressed, handle '5'
128     BTFSS PORTB,1 ; Check column 1
129     GOTO DISP_4 ; If pressed, handle '4'
130     BTFSS PORTB,0 ; Check column 0
131     GOTO DISP_B ; If pressed, handle 'B'
132     ;--- Row 1 -----
133     MOVLW 0x03 ; Row 1 select value
134     MOVWF PORTA ; Select row 1 on PORTA
135     MOVWF TEMP ; Save row value in TEMP
136     CALL DELAY ; Delay for settling
137     BTFSS PORTB,3 ; Check column 3
138     GOTO DISP_3 ; If pressed, handle '3'
139     BTFSS PORTB,2 ; Check column 2
140     GOTO DISP_2 ; If pressed, handle '2'
141     BTFSS PORTB,1 ; Check column 1
142     GOTO DISP_1 ; If pressed, handle '1'
143     BTFSS PORTB,0 ; Check column 0
144     GOTO DISP_A ; If pressed, handle 'A'
145
146     BTFSC STOP, 1 ; Check if STOP bit 1 is set
147     GOTO DISP_S ; If set, handle 'S'
148     GOTO KEYSKAN_MODE ; No key pressed, loop back
149 ;=====

```

```

150 ; KEY HANDLERS - One per key
151 ;=====
152 DISP_1:
153     MOVLW 0x31 ; ASCII '1'
154     GOTO HANDLE_KEY ; Go to key handler
155 DISP_2:
156     MOVLW 0x32 ; ASCII '2'
157     GOTO HANDLE_KEY ; Go to key handler
158 DISP_3:
159     MOVLW 0x33 ; ASCII '3'
160     GOTO HANDLE_KEY ; Go to key handler
161 DISP_4:
162     MOVLW 0x34 ; ASCII '4'
163     GOTO HANDLE_KEY ; Go to key handler
164 DISP_5:
165     MOVLW 0x35 ; ASCII '5'
166     GOTO HANDLE_KEY ; Go to key handler
167 DISP_6:
168     MOVLW 0x36 ; ASCII '6'
169     GOTO HANDLE_KEY ; Go to key handler
170 DISP_7:
171     MOVLW 0x37 ; ASCII '7'
172     GOTO HANDLE_KEY ; Go to key handler
173 DISP_8:
174     MOVLW 0x38 ; ASCII '8'
175     GOTO HANDLE_KEY ; Go to key handler
176 DISP_9:
177     MOVLW 0x39 ; ASCII '9'
178     GOTO HANDLE_KEY ; Go to key handler
179
180 DISP_A: MOVLW 0x0A ; Code for 'A'
181         GOTO HANDLE_KEY ; Go to key handler
182 DISP_B: MOVLW 0x0B ; Code for 'B'
183         GOTO HANDLE_KEY ; Go to key handler
184 DISP_C: MOVLW 0x0C ; Code for 'C'
185         GOTO HANDLE_KEY ; Go to key handler
186 DISP_D: MOVLW 0x0D ; Code for 'D' (though not used in scan, included)
187         GOTO HANDLE_KEY ; Go to key handler
188
189 DISP_S:
190     CLRF STOP ; Clear STOP register
191     MOVLW 0x53 ; ASCII 'S'
192     GOTO HANDLE_KEY ; Go to key handler
193 ;=====
194 ; HANDLE_KEY - Write once, wait for release
195 ;=====
196 HANDLE_KEY:
197     MOVWF _DATA ; Store key value in _DATA
198     MOVF POSITION,W ; Load current position
199     MOVWF _ADDRESS ; Set EEPROM address to position
200     CALL WRITE_EEPROM ; Write data to EEPROM
201     CALL READ_EEPROM ; Read back and display on PORTC
202     INCF POSITION,F ; Increment position
203     MOVF POSITION,W ; Load position
204     XORLW 0x0A ; Compare to 10
205     BTFSS STATUS,2 ; If not 10, skip
206     GOTO NO_DUMP2 ; Continue without dump
207     CALL DUMP ; Dump contents if 10 keys stored
208     BCF STATE,0 ; Reset to flash S mode
209     GOTO MAINLOOP ; Return to main loop
210 NO_DUMP2:
211 ; CLRF STOP ; Commented: Clear STOP
212     CALL WAIT_KEY_RELEASE ; Wait for key release (debounce)
213     GOTO MAINLOOP ; Return to main loop
214 ;=====
215 ; WAIT_KEY_RELEASE - Wait until key is released
216 ;=====
217 WAIT_KEY_RELEASE:
218     CALL DELAY ; Initial debounce delay
219 RELEASE_LOOP:
220     MOVF TEMP,W ; Restore saved row
221     MOVWF PORTA ; Re-select row
222     CALL DELAY ; Delay for settling
223     MOVF PORTB,W ; Read PORTB
224     ANDLW 0x0E ; Mask RB1,RB2,RB3 (columns)

```

```

225     XORLW 0x0E ; Check if all high (no press)
226     BTFSS STATUS,2 ; If not all high, loop
227     GOTO RELEASE_LOOP ; Continue waiting
228     CALL DELAY ; Final debounce
229     RETURN ; Return when released
230 ;=====
231 ; WRITE_EEPROM
232 ;=====
233 WRITE_EEPROM:
234     MOVF _ADDRESS,W ; Load address
235     BCF STATUS,5 ; Select Bank 2 for EEADR
236     BSF STATUS,6 ; RP1=1, RP0=0 (Bank 2)
237     MOVWF EEADR ; Set EEPROM address
238     BCF STATUS,5 ; Select Bank 0 for _DATA
239     BCF STATUS,6 ; Bank 0
240     MOVF _DATA,W ; Load data
241     BCF STATUS,5 ; Select Bank 2 for EEDATA
242     BSF STATUS,6 ; Bank 2
243     MOVWF EEDATA ; Set EEPROM data
244     BSF STATUS,5 ; Select Bank 3 for EECON1
245     BSF STATUS,6 ; Bank 3 (RP1=1, RP0=1)
246     BCF EECON1,7 ; Select data EEPROM
247     BSF EECON1,2 ; Enable write
248     BCF INTCON,7 ; Disable global interrupts
249     MOVLW 0x55 ; Write sequence 1
250     MOVWF EECON2 ;
251     MOVLW 0xAA ; Write sequence 2
252     MOVWF EECON2 ;
253     BSF EECON1,1 ; Start write
254     BSF INTCON,7 ; Re-enable global interrupts
255     NOP ; No operation
256 WRITE_POLL:
257     BTFSC EECON1,1 ; Poll for write complete
258     GOTO WRITE_POLL ; Wait if not done
259     BCF EECON1,2 ; Disable write
260     BCF STATUS,5 ; Return to Bank 0
261     BCF STATUS,6 ;
262     RETURN ; Return after write
263 ;=====
264 ; READ_EEPROM
265 ;=====
266 READ_EEPROM:
267     MOVF _ADDRESS,W ; Load address
268     BCF STATUS,5 ; Select Bank 2
269     BSF STATUS,6 ;
270     MOVWF EEADR ; Set EEPROM address
271     BSF STATUS,5 ; Select Bank 3
272     BSF STATUS,6 ;
273     BSF EECON1,0 ; Start read
274     BCF STATUS,5 ; Select Bank 2
275     BSF STATUS,6 ;
276     MOVF EEDATA,W ; Read data into W
277     BCF STATUS,5 ; Select Bank 0
278     BCF STATUS,6 ;
279     MOVWF PORTC ; Output to PORTC (7-segment)
280     RETURN ; Return after read
281 ;=====
282 ; DUMP
283 ;=====
284 DUMP:
285     BSF PORTA,5 ; Set RA5 (possibly indicator LED)
286     CLRF DUMP_GIE_SAVE ; Clear GIE save
287     BTFSC INTCON,7 ; Check if GIE was set
288     BSF DUMP_GIE_SAVE,0 ; Save GIE state
289     BCF INTCON,7 ; Disable global interrupts
290     CLRF _ADDRESS ; Start from address 0
291 DUMP_LOOP:
292     CALL READ_EEPROM ; Read and display on PORTC
293     CALL DELAY_LONG ; Long delay between displays
294     INCF _ADDRESS,F ; Increment address
295     MOVF _ADDRESS,W ; Load address
296     SUBWF POSITION,W ; Compare to position (stored count)
297     BTFSS STATUS,2 ; If not equal, continue
298     GOTO DUMP_LOOP ; Loop until all dumped

```

```

299     BCF INTCON,0 ; Clear RBIF
300     BTFSC DUMP_GIE_SAVE,0 ; Restore GIE if was set
301     BSF INTCON,7 ;
302     BCF PORTA,5 ; Clear RA5
303     RETURN ; Return after dump
304 ;=====
305 ; INTERRUPT (optional)
306 ;=====
307 INTERRUPT:
308     MOVWF SAVE_W ; Save W register
309     SWAPF STATUS,W ; Save STATUS (swap to avoid changing flags)
310     MOVWF SAVE_STATUS ; Store saved STATUS
311
312     ; BTFSS PORTB, 5 ; Commented: Check RB5
313     ; GOTO _DUMP ; If clear, go to dump
314     ; GOTO _RETURN ; Otherwise return
315     ; GOTO _DUMP ; Direct jump to dump (commented)
316     _DUMP:
317     BCF STATUS,5 ; Select Bank 0
318     BCF STATUS,6 ;
319     CALL DUMP ; Call dump routine
320
321     MOVLW 0X0A ; Load 10
322     MOVF _ADDRESS, W ; Move address (redundant?)
323     MOVLW 0X0A ; Load 10
324     MOVF POSITION, W ; Move position (redundant?)
325
326     MOVLW 0XFF ; Load 0xFF
327     MOVWF STOP ; Set STOP to 0xFF
328
329     BCF INTCON,0 ; Clear RBIF
330     GOTO _RETURN ; Go to return
331
332     _RETURN:
333     MOVLW 0X0A ; Load 10 (redundant?)
334     MOVF _ADDRESS, W ; Move address
335     MOVLW 0X0A ; Load 10
336     MOVF POSITION, W ; Move position
337     BCF STATE, 0 ; Reset to flash S mode
338     BCF INTCON, 0 ; Clear RBIF
339     MOVLW 0X53 ; 'S'
340     MOVWF PORTC ; Display 'S'
341     SWAPF SAVE_STATUS,W ; Restore STATUS
342     MOVWF STATUS ;
343     SWAPF SAVE_W,W ; Restore W (swap nibbles)
344     SWAPF SAVE_W,W ;
345     RETFIE ; Return from interrupt
346
347 ;=====
348 ; DELAYS
349 ;=====
350 DELAY:
351     MOVLW 0x80 ; Load counter for short delay
352     MOVWF TEMP ; Store in TEMP
353 DLOOP:
354     DECFSZ TEMP,F ; Decrement and skip if zero
355     GOTO DLOOP ; Loop until zero
356     RETURN ; Return after delay
357 DELAY_LONG:
358     MOVLW 0xFF ; Load outer counter
359     MOVWF TEMP2 ; Store in TEMP2
360 DL_OUTER:
361     MOVLW 0xFF ; Load inner counter
362     MOVWF TEMP ; Store in TEMP
363 DL_INNER:
364     DECFSZ TEMP,F ; Decrement and skip if zero
365     GOTO DL_INNER ; Inner loop
366     DECFSZ TEMP2,F ; Decrement outer and skip if zero
367     GOTO DL_OUTER ; Outer loop
368     RETURN ; Return after long delay
369 END

```