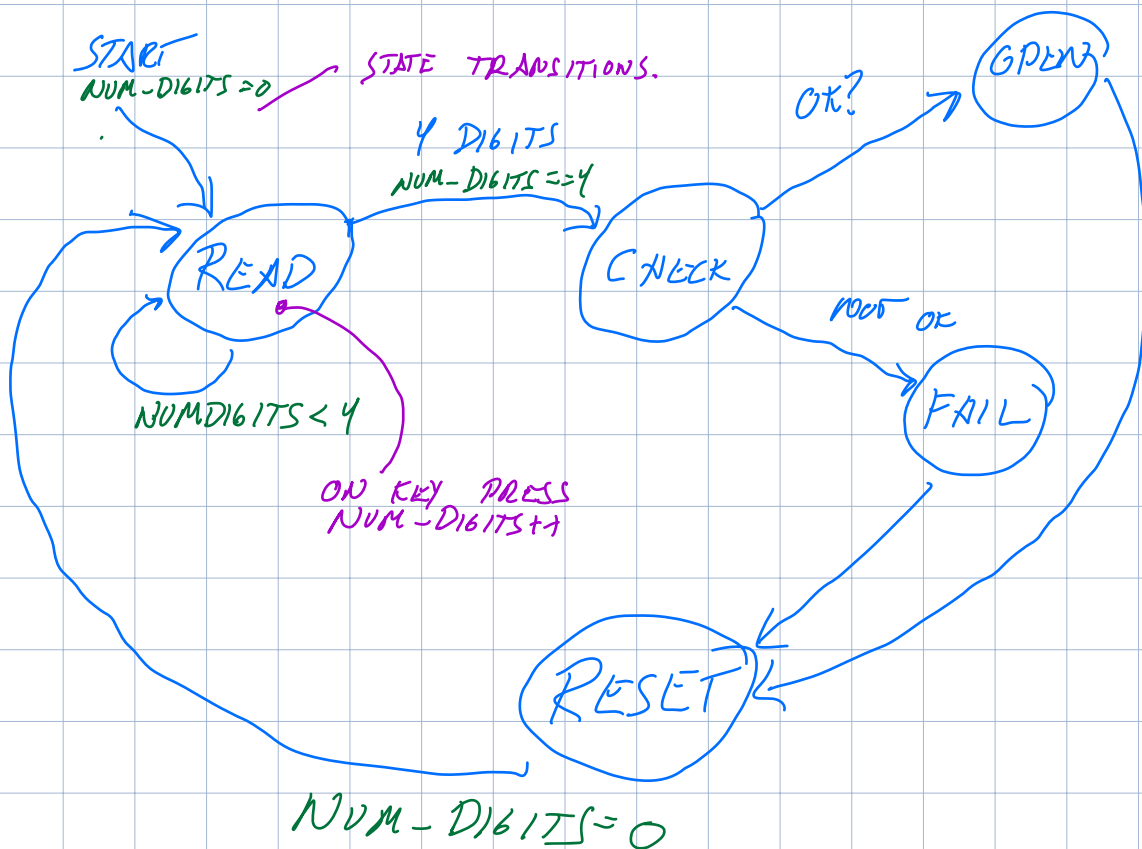


STATE MACHINE EXAMPLE: COMBINATION LOCK:

- PRE-SET COMBINATION (4 DIGITS)
- READ 4 KEYS FROM KEYPAD
 - IF YES: OPEN ☺
 - IF NO: FAIL ☹



```
1 // State machine example: Combination lock
2
3 #include <msp430.h>
4
5 #include "peripherals.h"
6 #include "utils/test_runner.h"
7 #include "leds.h"
8
9 #define CODE_LENGTH 4
10
11 // The combination we want the user to enter
12 char real_code[CODE_LENGTH] = {'1', '2', '3', '4'};
13
14 enum lock_state { // "Enumerated type"
15     READ = 0,
16     CHECK = 1,
17     OPEN = 2,
18     FAIL = 3,
19     RESET = 4,
20 };
21
22
23 // Function Prototypes
24 void swDelay(char numLoops);
25
26
27 // Main
28 void main(void)
29 {
30     char key;
31     enum lock_state state = READ; // Initial state is READ
32
33     char num_digits = 0;
34     char input_code[CODE_LENGTH];
35
36     WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer
37
38     // *** System initialization ***
39     Lecture_configLeds();
40     configDisplay();
41     configKeypad();
42
43     Graphics_clearDisplay(&g_sContext); // Clear the display
44     Graphics_flushBuffer(&g_sContext); // Refresh the display
45
46
47     // ... Continued on the next page ...
48
```

```
49
50 while (1)    // Forever loop
51 {
52     key = KeypadGetKey(); // Read a key from the keypad
53                           // (returns an ASCII character)
54
55     switch(state){
56     case READ:
57         Graphics_drawStringCentered(&g_sContext, "Enter key",
58                                     AUTO_STRING_LENGTH, 48, 15,
59                                     TRANSPARENT_TEXT);
60         Graphics_flushBuffer(&g_sContext); // Send changes to display
61
62         if (key != 0) { // Each time a key is pressed
63             //
64             input_code[num_digits] = key;
65             num_digits++;
66             setLeds(num_digits);
67
68             BuzzerOn();
69             swDelay(1);
70             BuzzerOff();
71         }
72
73         if (num_digits >= CODE_LENGTH) {
74             state = CHECK;
75         } else {
76             state = READ;
77         }
78         break;
79     case CHECK: {
80         int i;
81         int ok = 1;
82
83         for(i = 0; i < CODE_LENGTH; i++) {
84             if (input_code[i] != real_code[i]) {
85                 ok = 0;
86             }
87         }
88
89         if (ok) {
90             state = OPEN;
91         } else {
92             state = FAIL;
93         }
94         break;
95     }
96 }
```

```
107     case OPEN:
108         Graphics_clearDisplay(&g_sContext);
109         Graphics_drawStringCentered(&g_sContext, "YAY",
110                                     AUTO_STRING_LENGTH,
111                                     48, 15, TRANSPARENT_TEXT);
112         Graphics_flushBuffer(&g_sContext);
113         swDelay(1);
114         state = RESET; // Go back to the beginning
115         break;
116     case FAIL:
117         Graphics_clearDisplay(&g_sContext);
118         Graphics_drawStringCentered(&g_sContext, "BOO",
119                                     AUTO_STRING_LENGTH,
120                                     48, 15, TRANSPARENT_TEXT);
121         Graphics_flushBuffer(&g_sContext);
122         state = RESET;
123         break;
124     case RESET:
125         // Reset all program state
126         num_digits = 0;
127         Graphics_clearDisplay(&g_sContext);
128         Graphics_flushBuffer(&g_sContext);
129
130         state = READ;
131         break;
132 }
133
134 } // end while (1)
135 }
136
137
138
139 /// ... Other demo functions omitted ...
140
```