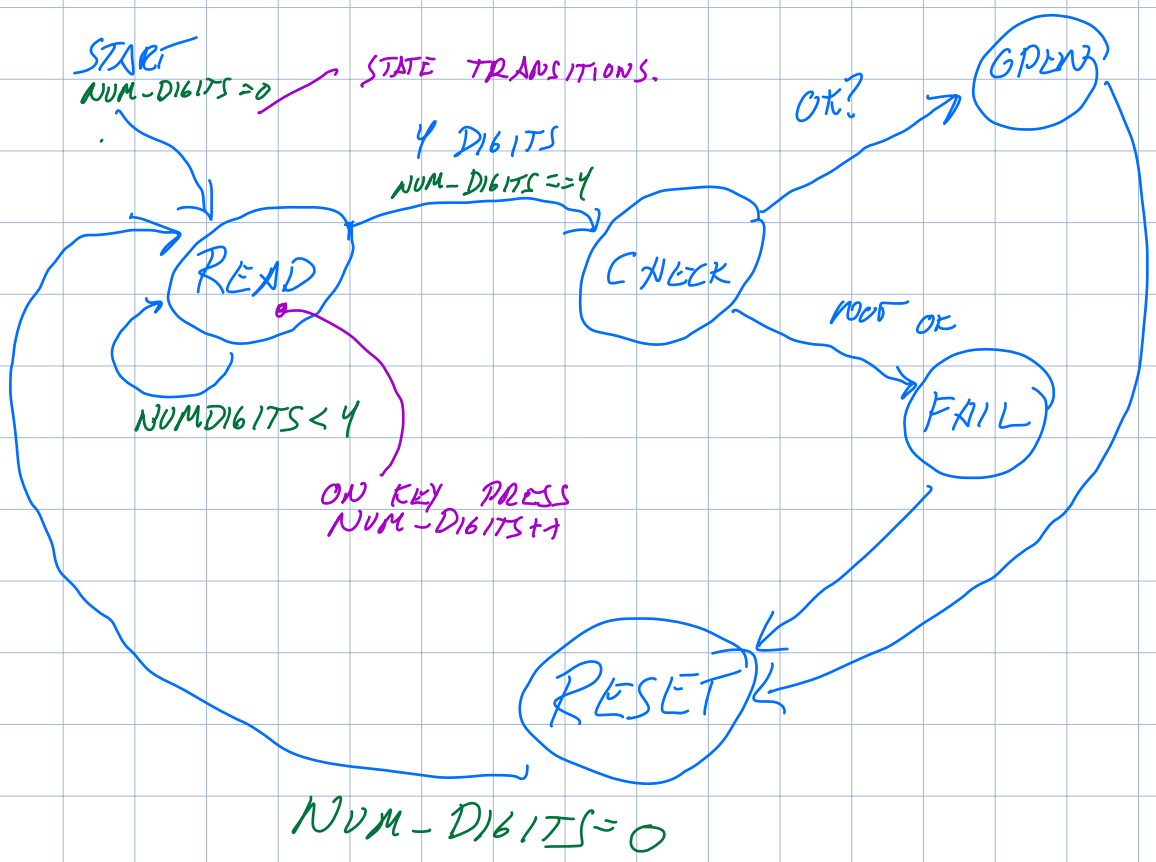
STATE MACHINE EXAMPLE: COMBINATION LOCK:

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



```c
// State machine example:  Combination lock

#include <msp430.h>

#include "peripherals.h"
#include "utils/test_runner.h"
#include "leds.h"

#define CODE_LENGTH 4

// The combination we want the user to enter
char real_code[CODE_LENGTH] = {'1', '2', '3', '4'};

enum lock_state { // "Enumerated type"
    READ = 0,
    CHECK = 1,
    OPEN = 2,
    FAIL = 3,
    RESET = 4,
};


// Function Prototypes
void swDelay(char numLoops);


// Main
void main(void)
{
    char key;
    enum lock_state state = READ; // Initial state is READ

    char num_digits = 0;
    char input_code[CODE_LENGTH];

    WDTCTL = WDTPW | WDTHOLD;    // Stop watchdog timer

    // *** System initialization ***
    Lecture_configLeds();
    configDisplay();
    configKeypad();

    Graphics_clearDisplay(&g_sContext); // Clear the display
    Graphics_flushBuffer(&g_sContext); // Refresh the display


   // ... Continued on the next page ...

```

```c

    while (1)    // Forever loop
    {
        key = KeypadGetKey(); // Read a key from the keypad
                              // (returns an ASCII character)

        switch(state){
        case READ:
            Graphics_drawStringCentered(&g_sContext, "Enter key",
                                        AUTO_STRING_LENGTH, 48, 15,
                                        TRANSPARENT_TEXT);
            Graphics_flushBuffer(&g_sContext); // Send changes to display

            if (key != 0) { // Each time a key is pressed
                //
                input_code[num_digits] = key;
                num_digits++;
                setLeds(num_digits);

                BuzzerOn();
                swDelay(1);
                BuzzerOff();
            }

            if (num_digits >= CODE_LENGTH) {
                state = CHECK;
            } else {
                state = READ;
            }
            break;
        case CHECK: {
            int i;
            int ok = 1;

            for(i = 0; i < CODE_LENGTH; i++) {
                if (input_code[i] != real_code[i]) {
                    ok = 0;
                }
            }

            if (ok) {
                state = OPEN;
            } else {
                state = FAIL;
            }
            break;
        }

```

```c
        case OPEN:
            Graphics_clearDisplay(&g_sContext);
            Graphics_drawStringCentered(&g_sContext, "YAY",
                                        AUTO_STRING_LENGTH,
                                        48, 15, TRANSPARENT_TEXT);
            Graphics_flushBuffer(&g_sContext);
            swDelay(1);
            state = RESET; // Go back to the beginning
            break;
        case FAIL:
            Graphics_clearDisplay(&g_sContext);
            Graphics_drawStringCentered(&g_sContext, "BOO",
                                        AUTO_STRING_LENGTH,
                                        48, 15, TRANSPARENT_TEXT);
            Graphics_flushBuffer(&g_sContext);
            state = RESET;
            break;
        case RESET:
            // Reset all program state
            num_digits = 0;
            Graphics_clearDisplay(&g_sContext);
            Graphics_flushBuffer(&g_sContext);

            state = READ;
            break;
        }


    }  // end while (1)
}


/// ... Other demo functions omitted ...

```