ESET 369 LAB REPORT

Student name	Kyle Rex and Blake Jackson
Group number	40
Lab Section	501
Lab session date	January 28, 2025
Lab instructor	Xin Zhuang

INTRODUCTION

The purpose of this lab is to utilize C/C++ coding to control the LED outputs on the BH boards. With proper software configuration on the launchpad and proper physical configuration on the BH board, the output should simulate different LED outputs. This will familiarize the user with launchpad GPIO configuration as well as the configuration of the BH board. System A involved creating a blinking LED and System B involved creating a specific LED sequence.

SYSTEM A

The purpose of System A is to correctly configure the launchpad and board to enable LED_A, one of the 8 LEDs on the BH board used for this lab, to blink on and off indefinitely. The blinking duration is required to be between 0.5 to 1 second. The first step in addressing this problem is making the physical board connections between the BH board and the MSP430FR5994 microcontroller. This can be done by connecting nine specific pins together using male and female jumper wires, checking that all the DIP switches on the BH board are in the correct configuration, and ensuring the microcontroller itself is properly seated on the BH board. Eight of the pins control LEDs and one controls a button. The next step in solving this problem is to ensure that the GPIO pins are not locked. This can be accomplished by using "PM5CTL0 &=~LOCKLPM5". Next, define the direction of port 3 and set it as an output. Once that is complete a running loop can be configured to infinitely keep the LED on flickering on and off. To do this, make a statement within a while loop. Finally, add the body of the loop which can be done by using a XOR bitwise operation. Always add a delay after to ensure that the ports have time to read the signal.

The result of this code sequence should be an actively blinking LED_A. This LED should flash until an interrupt is present. As previously stated, the flashing sequence of the LED should be between 0.5 and 1 second. This can be done using a delay function which should be set to approximately 500000. As shown in Table 1, the LED took around 9 seconds to blink a total of 10 times therefore achieving the goal of the system of having a time period of 0.5 to 1 second between each blink.

Table 1: Blinks per Second Chart

	Time
10 Blinks	9 Seconds

SYSTEM B

Continuing with the same physical setup as System A, System B involved correctly configuring the launchpad and board to display an eight LED sequence indefinitely while a button is being pressed. When the button is not pressed all eight of the LEDs should be turned off. The duration of the single LED sequence must be around 2 seconds. This requires the transition to the next LED pattern to be reasonably quick between 0.2 and 0.3 seconds. The sequence that occurs while the button is pressed is meant to recreate an effect that is shown on a car from the Knight Rider Show (NBC Classics). This sequence can be seen in Figure 1.

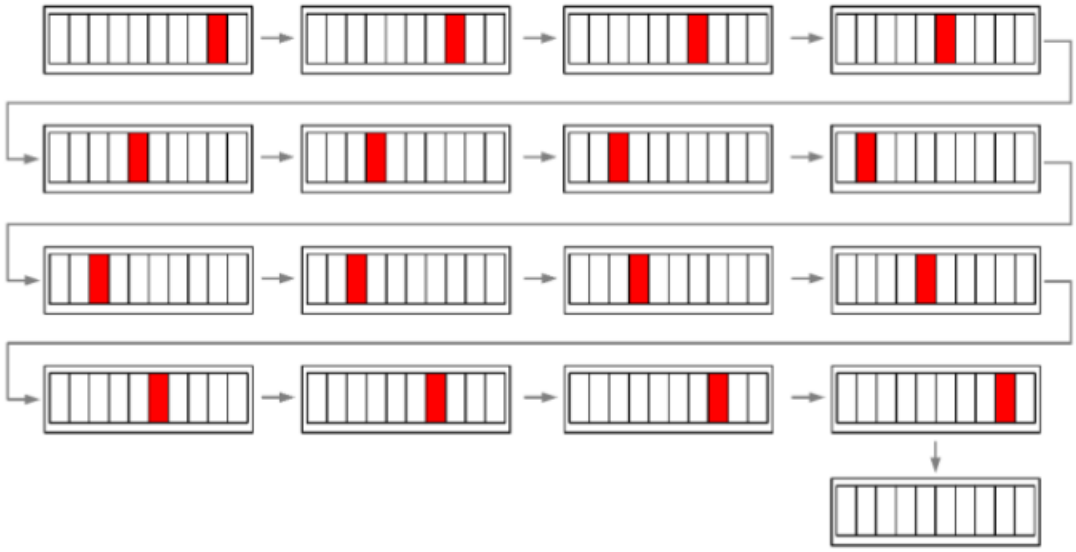

Figure 1: LED Sequence

This can be done by utilizing the same LED commands used in System A with the addition of the button functions and commands. A pull up button configuration is used for this lab using “P6OUT |= BIT0;”. Once the button is initialized a running loop can be configured to infinitely keep the LED sequence going by using a while loop that has an internal if function that requires the button to be set too low. This allows the sequence to only occur while the button is being pressed. Creating the body of the while loop involved initializing each LED, turning them on and off, and using a delay function a total of sixteen times (to replicate the exact 16 step sequence shown in Figure 1). As previously stated, the flashing sequence of the LEDs should be between 0.2 and 0.3 seconds. This can be done using a delay function which should be set to approximately 150000. This code sequence results in displaying the Knight Rider effect therefore achieving the goal of the system.

CONCLUSION

This lab successfully demonstrated the use of C/C++ programming to control LED outputs on the BH board using the MSP430FR5994 microcontroller. System A achieved a correctly configured blinking LED with a time interval between 0.5 and 1 second, meeting the lab's requirements. System B implemented an eight-LED sequential pattern, replicating the Knight

Rider effect while a button was pressed. The successful execution of both systems reinforced key concepts in GPIO configuration, bitwise operations, and timing control using delay functions. Overall, this lab provided valuable hands-on experience with embedded system programming, further solidifying an understanding of microcontroller-based LED control.

REFERENCES

NBC Classics. (n.d.). *Knight Rider - Original Show Intro*. YouTube. https://www.youtube.com/watch?v=oNyXYPhnUIs&t=19s