EXPERIMENT 2

SEVEN SEGMENT LED

OBJECTIVE

- Familiarise with seven-segment LEDs
- Interfacing seven-segment LEDs with the KL25Z board
- Creating a rolling display over 4 seven segment LEDs.

THEORY

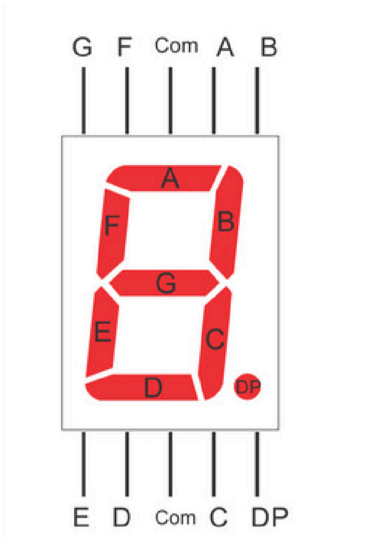

Fig. 1 Seven-segment LED

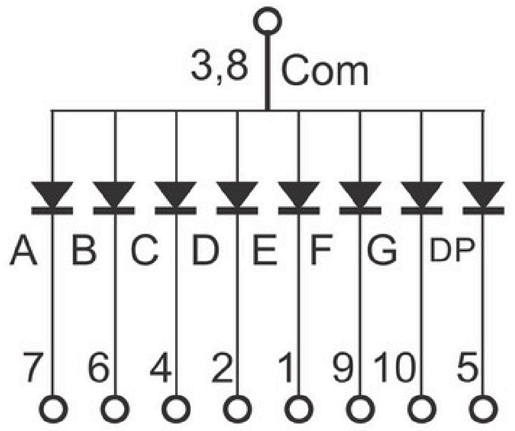

Fig. 2. Internal diagram of common anode SSD

A seven-segment LED or seven-segment display (SSD) is shown in Fig. 1. It consists of seven LEDs A, B, C, D, E, F, and G each of which can be lit up selectively to create various alphanumeric characters. An optional eighth LED helps to display the decimal point (DP). To limit the current flowing through the microcontroller (MCU), resistors have to be connected in series to the LED before connecting to the GPIO pins on the MCU.

The LEDs in the seven-segment display are arranged in a common cathode or anode configuration. In a common anode configuration as shown in Fig. 2, all the anodes are shorted and common to all the LEDs. A positive bias +VDD is applied to the common anode pin and the cathode pin is connected to the GPIO pins of the MCU. Driving a 0 to the cathode pin forward biases the LED and it glows. Driving a 1 (VDD) to the pin removes the bias on the LED and switches the LED off. Contrary to the above operation, a common cathode SSD has all the cathodes shorted and the GPIO pin drives the inverted values on the anode.

Connecting multiple seven-segment LEDs
To use 4 seven-segment LEDs, 8x4=32 connections to the MCU are required. The number of pins required increases proportionately as the number of seven segment digits increases. To reduce the number of pins, the SSDs can be used in a time-multiplexed manner. The circuit diagram to use the 4 SSDs in a time multiplexed manner is shown in Fig. 2.

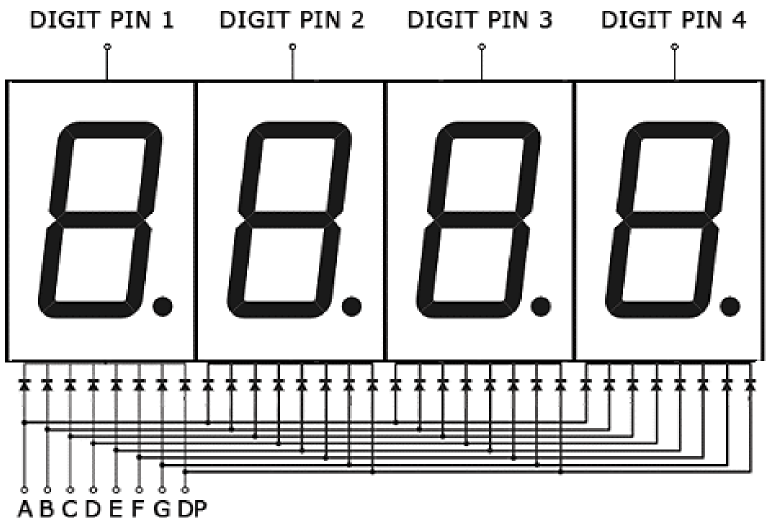

Fig. 3. Connecting 4 seven-segment LEDs for time multiplexing

In this arrangement, the LED data pins A-G and DP of all the digits are connected to the same pins of the MCU. Instead of powering the common anode pin directly from the power supply, it is powered (or enabled) by a control signal from the MCU (Digit pin 1-4). Only one digit is enabled at a time and the corresponding digit is written on the LED data pins. After a small delay, the next digit is enabled and its data is written from the MCU. This is repeated till all digits are displayed. By switching through the digits very fast, it appears that all the digits are simultaneously functioning to humans due to the persistence of vision.

Instead of the MCU directly driving the digit control pin and causing it to be overloaded, it can be driven through a transistor. By doing so the seven-segment LED is powered from the power supply and not directly from the MCU.

Reference: https://www.circuitbasics.com, https://www.circuitstoday.com/

PROCEDURE

1. Calculate the LED current limit resistor to limit LED current at 10 mA and connect the circuit as shown in. Assume that the seven-segment LED is powered from the power supply at +3.3V.
2. Determine the pattern that needs to be driven to the seven-segment display to display numbers 0-9. Assume that we have a common anode display.
3. Connect a seven-segment display to seven GPIO pins on the KL25Z board. Connect the common anode pin (Pin 3,8) to +3.3 V from the power supply. Write C code to verify the seven-segment LED by displaying numbers 0-9 on it.

   Time multiplexed SSD
4. Connect the 4 SSDs as shown in Fig. 3. Connect the SSD data pins to KL25Z through current limiting resistors. Connect the SSD common anode digit pins to the GPIO through a transistor drive circuit.
5. Write C code and verify that the 4 digits are working

   Rolling display
6. To display strings having greater than 4 characters, a rolling type display can be used. The characters move from left to right and repeat itself.
7. Write C code to create a rolling display and verify its functionality.