

The program emulates a finite state machine. Each state displays a different LED color: red, blue, green, and purple. In the initial/start state, all LEDs are off. State change occurs after a button is pressed (when a button goes down and then up, it is considered “pressed”).

When the code is loaded to the microcontroller, both LEDs, LED1 and LED2 are off. When BUTTON2 is pressed, LED2 turns RED. When the button is pressed again, LED2 turns BLUE. When BUTTON2 is pressed again, LED2 turns GREEN. Pressed two more times, LED2 turns PURPLE, and then RED, and the sequence of LED2 colors repeats after each button press.

LED1 has similar toggle functionality, but its logic is simple. BUTTON1 turns LED1, which is RED, off when it is on, and on when it is off.

The buttons are responsive at the “same time”. In actuality, the code related to BUTTON1 and BUTTON2 are executed asynchronously. Meaning, if the buttons are pressed at the same time, one instruction will execute first, and then the next set. In other words, they are executed sequentially. It is an illusion that their respective code is executed at the same time. The clock of the microcontroller is 48MHz, thus instructions are executed quickly enough that the human eye processes them as simultaneous events.

In conclusion, the program meets project specifications. An “A” on the project would be grand. Thank you.