LAB #2

Kizito NKURIKIYEYEZU

January 22, 2023

1 BACKGROUND

In this exercise, you will design and build an electronic dice similar to this one[1].

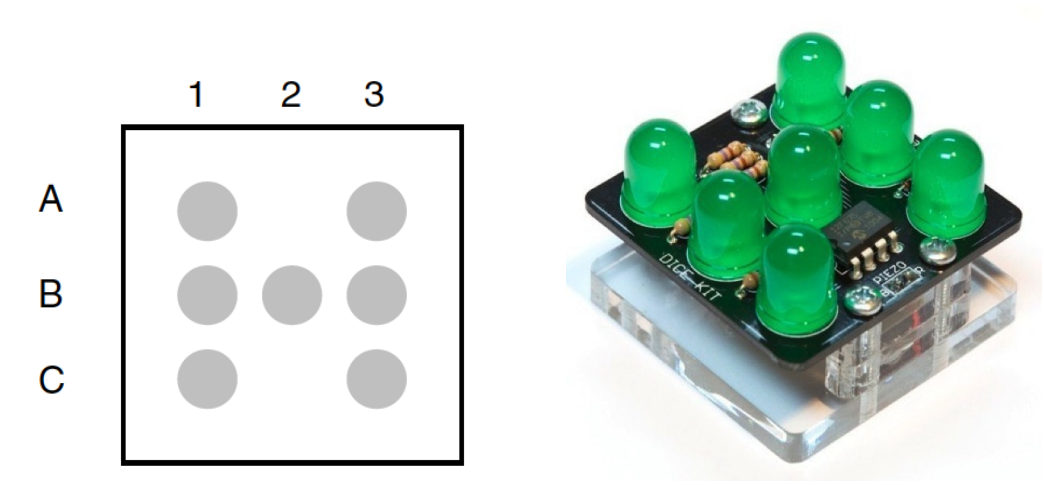
The decide will functions as follow:

- The device consist of ATtiny2313, seven LEDs, and two switches
- The device should have an input switch, and upon pressing the switch, a number (selected randomly by the controller) between 1 and 6 should be displayed. As shown in Figure 1, Figure 3 and Figure 2 the LEDs are organized such that when they turn ON, they indicate numbers as on a real dice.
- The device has two switches:
  - RESET_SWITCH —when the reset switch is pressed, the device turn off all LED
  - ROLL_SWITCH —when the roll switch is pressed, the display should go off momentarily before displaying a random number. This gives feedback to the user that the switch press was recognized by the processor. In the absence of this blanking feature, if the user presses a switch and the next number happens to be the same as the last one, the user may not recognize that.
- The dice should be very compact and small. Thus, use a small microcontroller. For example, the ATtiny2313 will suffice.
- You should follow the industry accepted embedded coding standard[2]
- The final circuit of the device is shown in Figure 4. Calculate the values of resistors that are required and draw, test and virtually prototype the device in proteus.

2 PROGRAM PSEUDOCODE

The operation of the dice is described in Algorithm 1. At the beginning of the program PORTC pins are configured as outputs and bit 0 of PORTB (RB0) is configured as input. The program

[1] https://www.spikenzielabs.com/learn/dicekit.html
[2] https://barrgroup.com/sites/default/files/barr_c_coding_standard_2018.pdf



(a) Dice LED arrangment **(b)** Real-world implementation

FIGURE 1. Output LED arrangement for the dice.

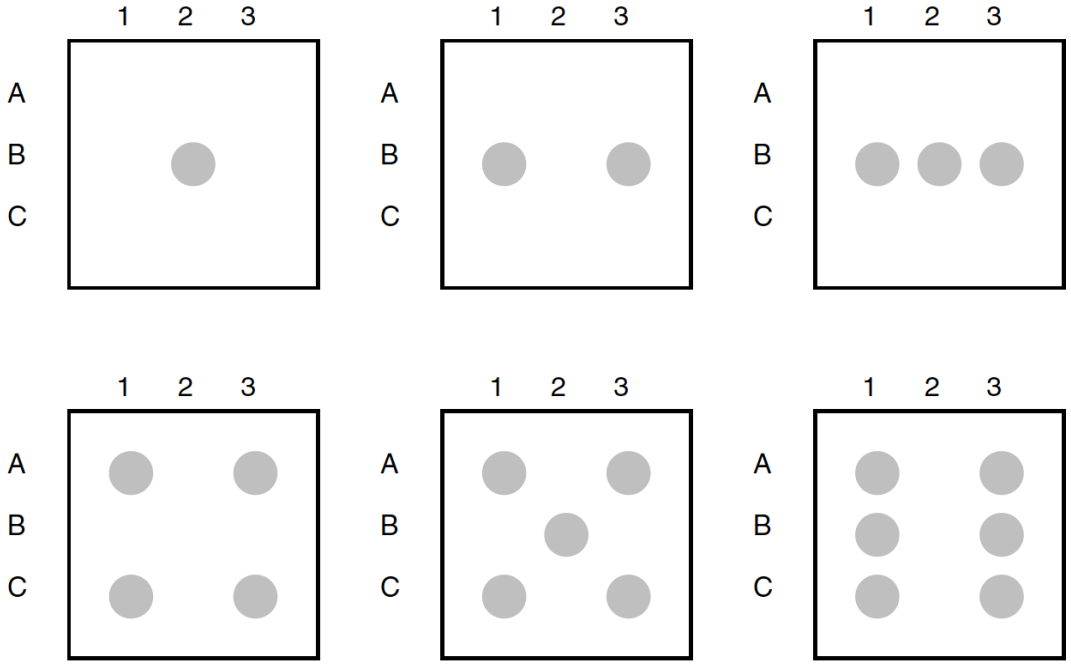


FIGURE 2. LEDs light up in this fashion for the numbers 1 to 6.

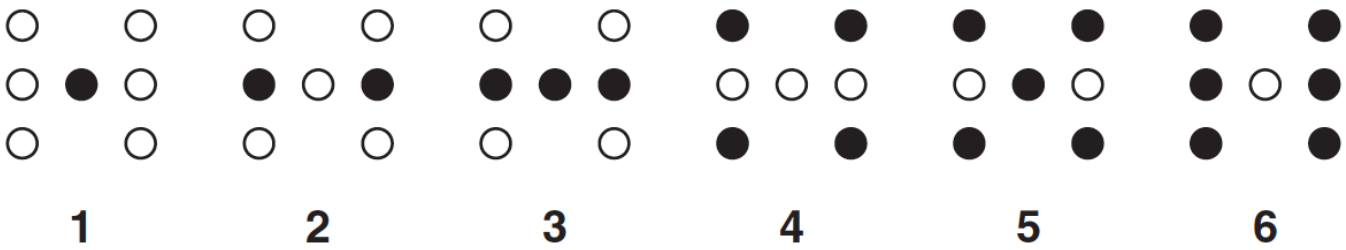


FIGURE 3. LED layout arrangement and the corresponding digits

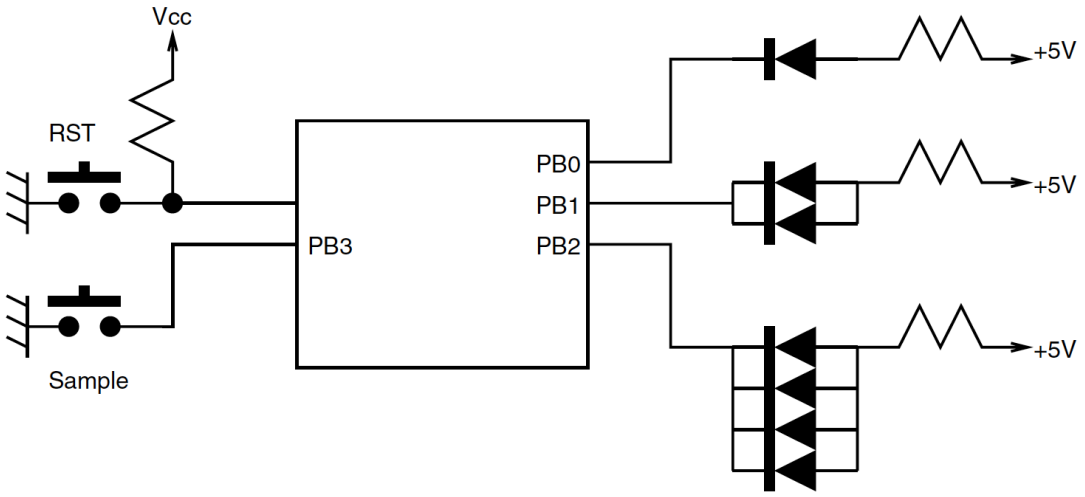


FIGURE 4. Block diagram for the electronic dice using an ATtiny2313 MCU

then executes in a loop continuously and increments a variable between 1 and 6. The state of the push-button switch is checked and when the switch is pressed (switch output at logic 0), the current number is sent to the LEDs.Asimple array is used to find out the LEDs to be turned ON corresponding to the dice number.

Algorithm 1: Program Pseudocode to control the dice

Input:
- RESET_SWITCH —when the reset switch is pressed, the device turn off all LED
- ROLL_SWITCH —when the roll switch is pressed, a new random number is displayed

Initialization
- All pins connected to the switches are initialized as inputs
- All pins connected to the LEDs are initialized as outputs
- Turn off all LEDs

end

Output: A random number between 1 and 6 displayed on the LED

Loop
- **if** *RESET_SWITCH==PRESSED* **then**
  - Turn OFF all LEDs
- **if** *ROLL_SWITCH==PRESSED* **then**
  - Wait for one second
  - Generate a new random number
  - Turn ON the LEDs corresponding to the random number

EndLoop

You can use the AVR Libc to generate the random number. Please refer to its random function, which is described here.

3 Lab submission

- I will check your work (Proteus simulation and the source code) at the beginning of the next lecturer.
- You can do the lab in a group of no more than 3 students
- The lab is handed out on January 22, 2023 and is due on **January 30, 2023**.
- Because the course will only lasts for 11 weeks, this is a hard deadline and **late submissions are not accepted after the deadline**.
- If you're not available in the lab, you will get a zero for the assignment (unless you can provide a documented evidence for your absence)