Yash Sangale | Electronics Engineer

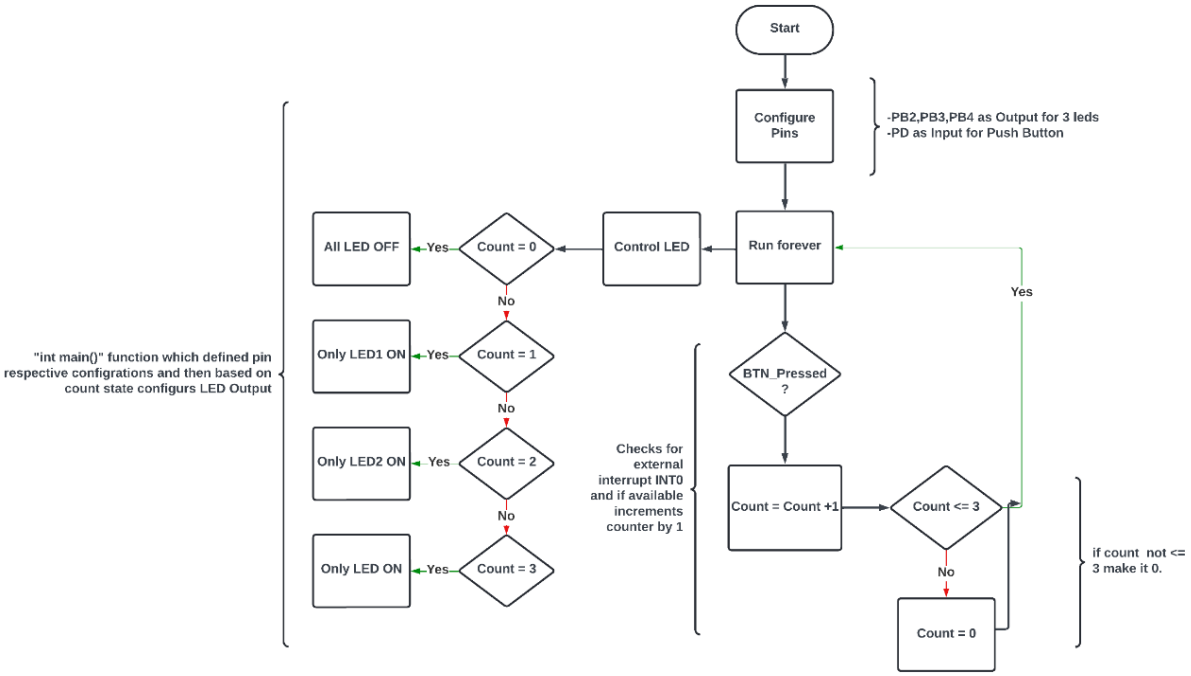
+91 80009-25523 | sangale.yash1307@gmail.com | www.linkedin.com/in/yash-sangale

Task:

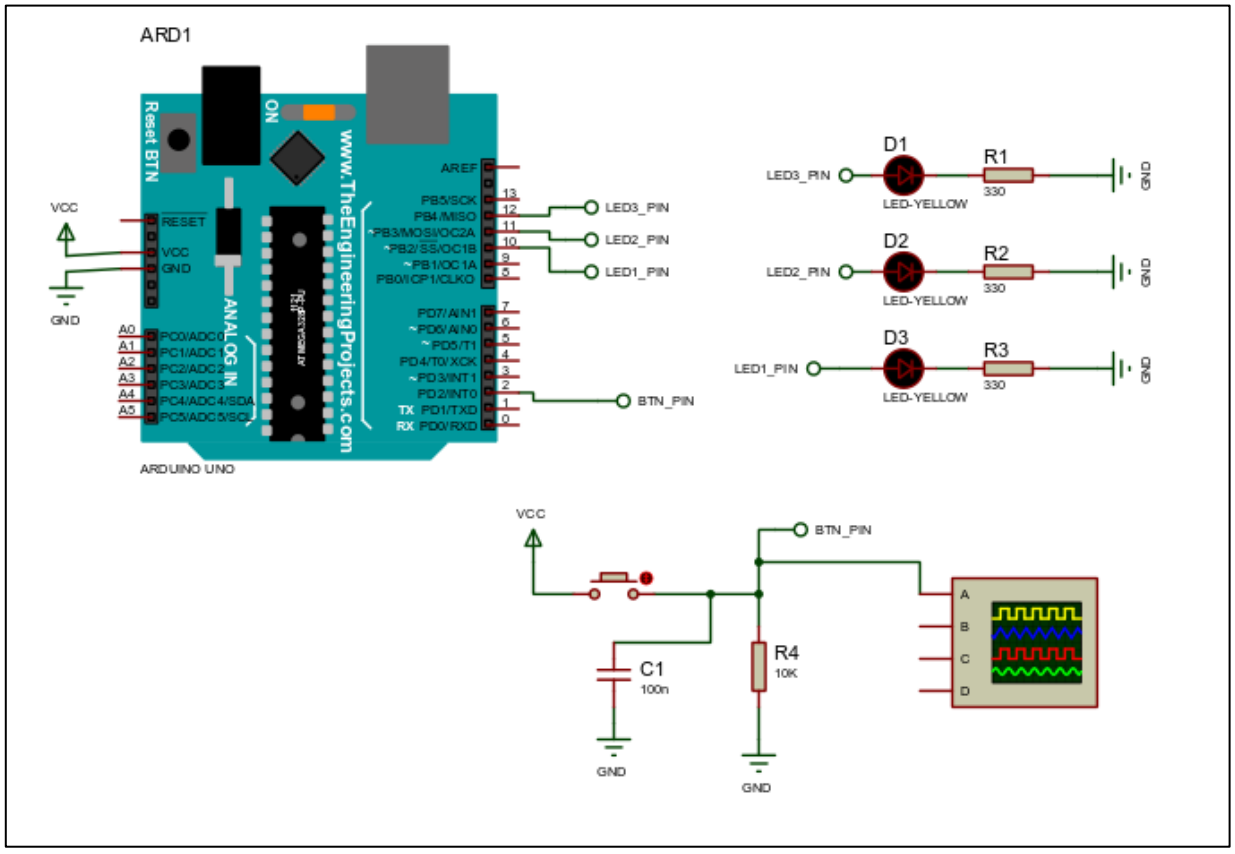
Connect 3 LED and a Push Button with Arduino (Uno/Nano/Mega/Pro Mini). The set of 3 LEDs will Glow one by one when the button is pressed.

Theory:

Code Flow:



Schematic:



Code:

```
#include <avr/io.h>
```

Github Link: https://github.com/Yash-Sangale/BTN_PRESS_LED_GLOW

```
#include <avr/interrupt.h>

#define LED1 PB4     // D12
#define LED2 PB3     // D11
#define LED3 PB2     // D10
#define BTN_PIN PD2  // D2

/************ Function Declarations **************/

void pin_config();                 // Function to configure pins
void LED_OFF();                    // Function to turn off all LEDs
void LED1_ON();                    // Function to turn on LED1
void LED2_ON();                    // Function to turn on LED2
void LED3_ON();                    // Function to turn on LED3
void control_led(uint8_t count);  // Function to control LEDs based on button press
count
bool btn_pressed();                // Function to detect valid button press

/************ Global Variable Declarations **************/

volatile uint8_t btn_press_count = 0;  // Stores button press counts
const unsigned long d_delay = 65;      // 50 milliseconds delay for avoiding
debounce effect
unsigned long last_d_time = 0;

int main() {
  pin_config();  // Configure pins
  while (1) {
    control_led(btn_press_count);  // Control LEDs based on button press count
  }
  return 0;
}

void pin_config() {
  DDRB = 0b00011100;         // Define LED pins as output
  DDRD &= ~(1 << BTN_PIN);  // Set BTN_PIN as input
  PORTD &= ~(1 << BTN_PIN);  // Enable pull-up resistor for BTN_PIN
  EIMSK |= (1 << INT0);     // Enable INT0 interrupt
  EICRA |= 0b0011;            // Set rising edge of INT0 generates an interrupt
request
  SREG |= (1 << 7);         // Enable global interrupts
}

// Function to turn off all LEDs
void LED_OFF() {
  PORTB &= ~(1 << LED1);  // LED1 OFF
  PORTB &= ~(1 << LED2);  // LED2 OFF
  PORTB &= ~(1 << LED3);  // LED3 OFF
}

// Only LED1 ON
void LED1_ON() {
  PORTB |= (1 << LED1);   // LED1 ON
  PORTB &= ~(1 << LED2);  // LED2 OFF
  PORTB &= ~(1 << LED3);  // LED3 OFF
}

// Only LED2 ON
```

Github Link: https://github.com/Yash-Sangale/BTN_PRESS_LED_GLOW

```
void LED2_ON() {
  PORTB &= ~(1 << LED1);  // LED1 OFF
  PORTB |= (1 << LED2);   // LED2 ON
  PORTB &= ~(1 << LED3);  // LED3 OFF
}

// Only LED3 ON
void LED3_ON() {
  PORTB &= ~(1 << LED1);  // LED1 OFF
  PORTB &= ~(1 << LED2);  // LED2 OFF
  PORTB |= (1 << LED3);   // LED3 ON
}

// Controls LED Operation based on button press count which is handled by External
Interrupt INT0
void control_led(uint8_t count) {
  switch (count) {
    case 0: LED_OFF(); break;
    case 1: LED1_ON(); break;
    case 2: LED2_ON(); break;
    case 3: LED3_ON(); break;
  }
}

// External Interrupt which increments Button Press on press detection
ISR(INT0_vect) {
  if (btn_pressed()) {

    btn_press_count++;
  }
  btn_press_count = (btn_press_count >= 4) ? 0 : btn_press_count;
  EIFR |= (1 << INTF0);  // Clear INT0 interrupt flag
}

// Function to detect valid button press avoiding Debounce Effect
bool btn_pressed() {
  if (bit_is_set(PIND, BTN_PIN)) {
    _delay_ms(d_delay);  // Debouncing delay
    if (bit_is_set(PIND, BTN_PIN)) {
      return true;
    }
  }
  return false;
}
```

Github Link: https://github.com/Yash-Sangale/BTN_PRESS_LED_GLOW