DESIGN OF KEYPAD & LCD AS NON-BLOCKING FUNCTIONS

Kareem Magdy Albolaqi

Sprint

8

Keypad problem description

PROBLEM:

1- there is a delay for debouncing that block the program

2- if long press occurred, it will block the program

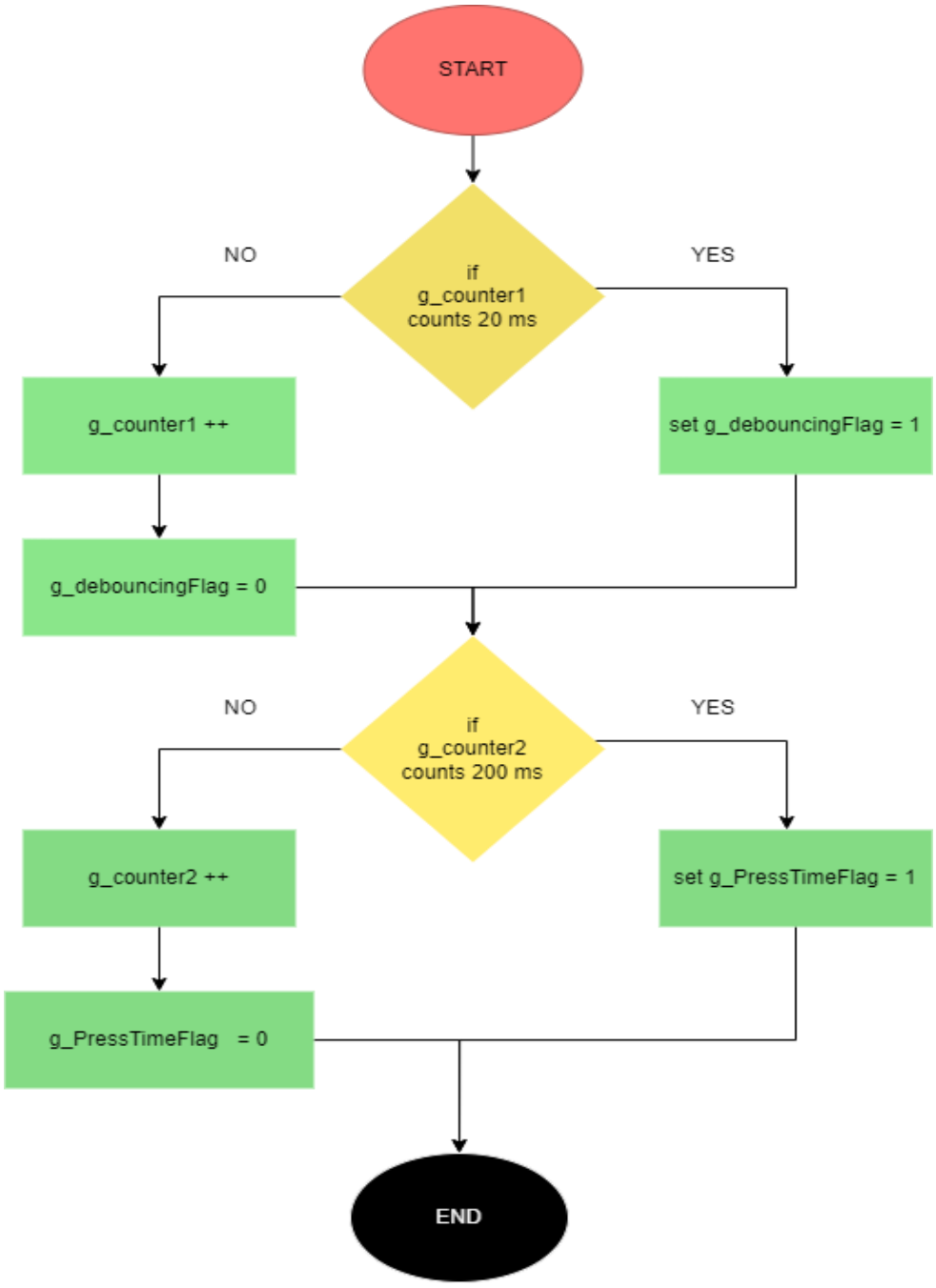
SOLUTION:

1. using timer overflow interrupt to count 20 MS that used to avoid debouncing

2. using timer overflow interrupt to make the keypad function returns its value every 200 MS

KEYPAD FLOWCHART

KEYPAD_delay()

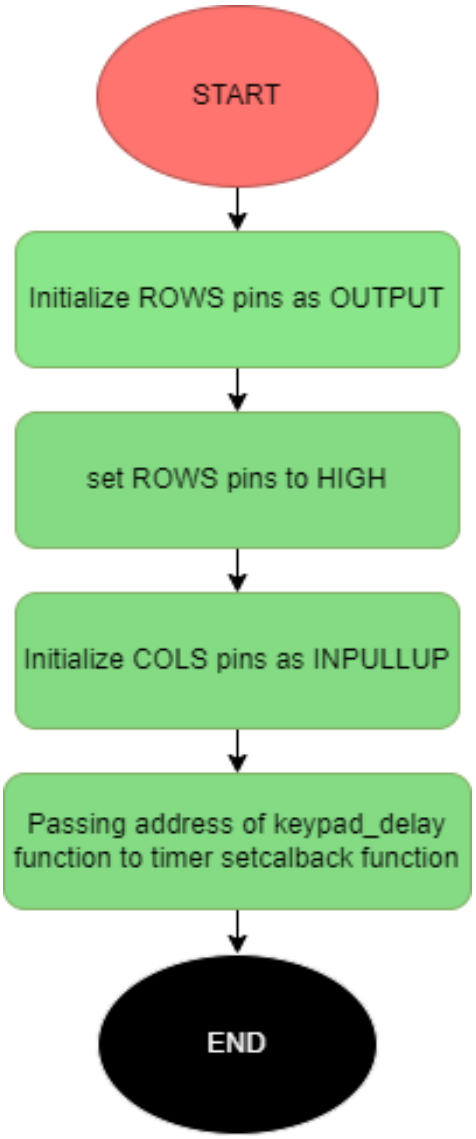


description

- This function will be passed to the timer setcallback function
- Is used to do two things:
  1. 20 ms delay for debouncing and when timer finished counting 20 ms it will rise global flag (g_debouncingFlag)
  2. 200 ms delay used before each return of the value and when timer finished it will rise a global flag (g_pressTimeFlag)

KEYPAD FLOWCHART

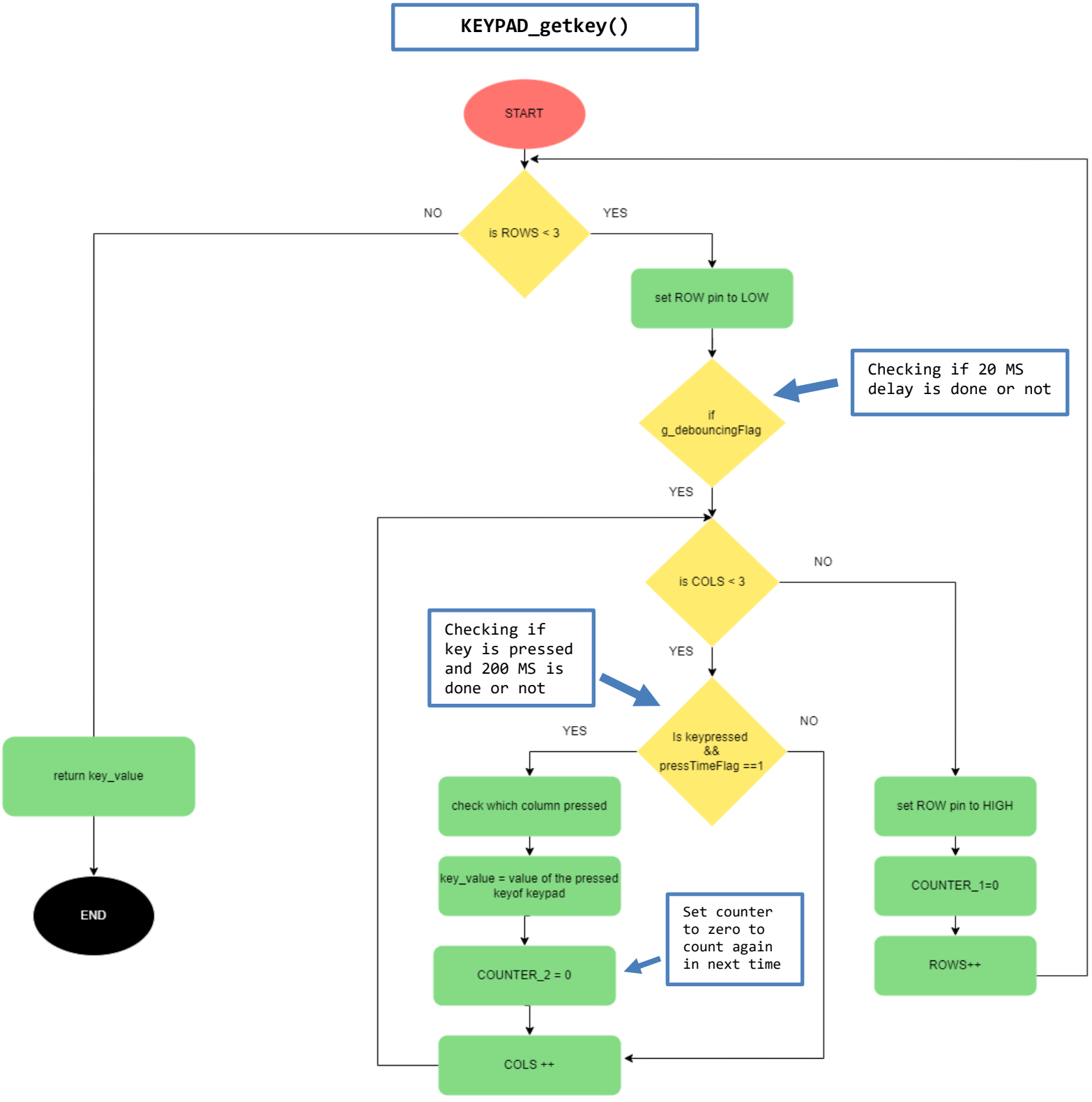
KEYPAD_init()



description

- This function is used to initialize rows of keypad as OUTPUT and set them HIGH
- Also, initialize columns of keypad as INPUT PULLUP
- Then , passing KEYPAD_delay( ) function to the timer setcallback function

KEYPAD FLOWCHART



KEYPAD FLOWCHART

KEYPAD_getkey()

description

- First , row pin of keypad will set to LOW
- Then, check if the debouncing delay (20 MS) is done or not
- If it not done it will continue the program
- If it is done then we will check two thing
    1. columns of keypad if anyone of them is pressed or not
    2. If the press time delay(200 MS) is passed or not , this time is used to returning the same pressed key many times in one click and avoid blocking the whole program
- If key is pressed and press time is done then we will save the pressed key to return it and set counter of the press time delay and counter of debouncing delay to zero to repeat counting in the next time

LCD problem description

PROBLEM:

- Implementation of lcd need to use some delays and that will block our program

SOLUTION:

- using timer overflow interrupt to count seconds required for implementation

LCD FLOWCHART

```
void LCD_8_Bit_init(void){

    for (uint8_t data_pins = 0 ; data_pins < 8 ; data_pins++)
    {
        DIO_init(lcd__dio_data_port,data_pins,OUTPUT);
    }

    for (uint8_t cmd_pins = 0 ; cmd_pins < 3 ; cmd_pins++)
    {
        DIO_init(lcd_dio_cmd_port,cmd_pins,OUTPUT);
    }
     _delay_ms(20);

     LCD_8_Bit_Sen Command(LCD_16_2_INIT);
     LCD_8_Bit_Send ommand(DISPLAY_ON_CURSOR_ON);
     LCD_8_Bit_SendCo mand(INCREMENT_CURSOR);
     LCD_8_Bit_SendComm nd(CLEAR_DISPLAY);
     LCD_8_Bit_SendComma (CURSOR_HOME_POSITION);

}
```

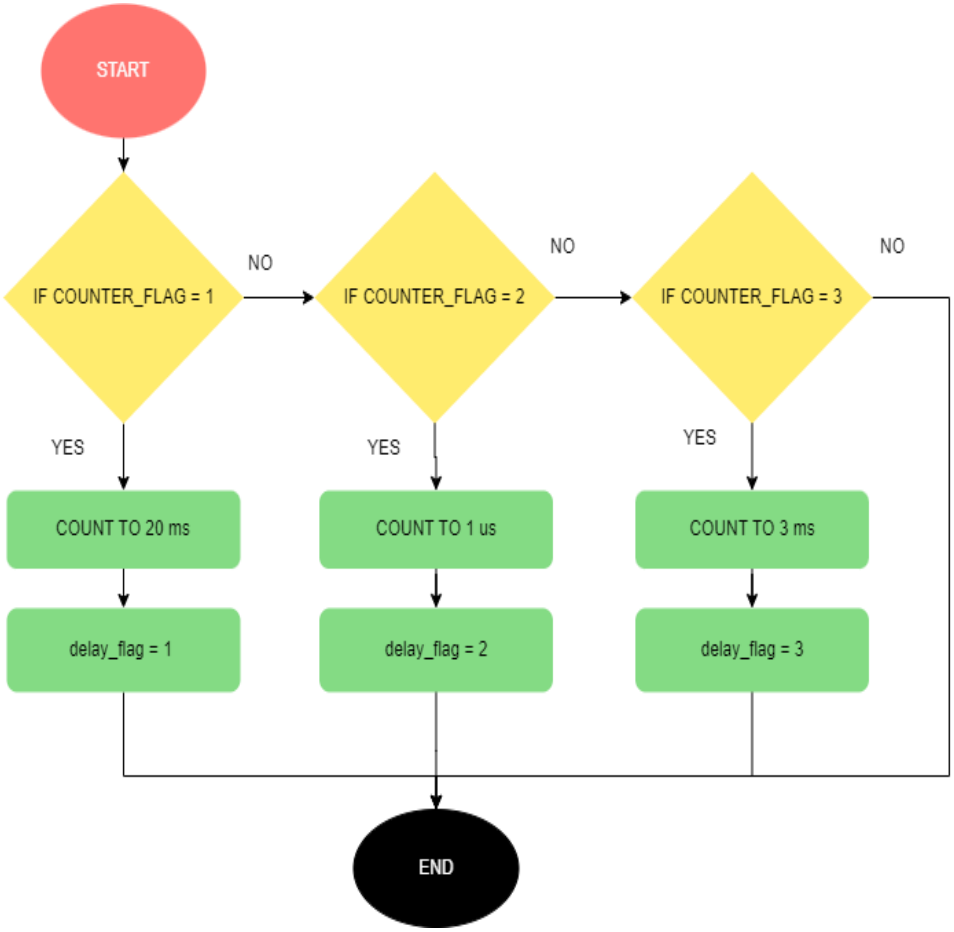
```
void LCD_8_Bit_SendCommand(uint8_t a_cmd){
    LCD_REG_DATA_PORT = a_cmd;
    DIO_write(lcd_dio_cmd_port,RS,LOW);
    DIO_write(lcd_dio_cmd_port,RW,LOW);
    DIO_write(lcd_dio_cmd_port,EN,HIGH);
    _delay_us(1);
    DIO_write(lcd_dio_cmd_port,EN,LOW);
    _delay_ms(3);


}
```

- WE HAVE TO USE 3 DELAYS TO IMPLMENT LCD INIT AND SEND COMMAND SO WE WILL USE FLAG WITH 3 STATE, A STATE FOR EACH DELAY
- WHEN WE SET THIS FLAG TO EQUAL 1 SO THE TIMER WILL COUNT 20 MS, AND IF WE SET IT TO EQUAL 2 TIMER WILL COUNT 1 MICROSECONDS, AND IF WE SET IT TO EQUAL 3 TIMER WILL COUNT 3 MS

LCD FLOWCHART

LCD_delay()

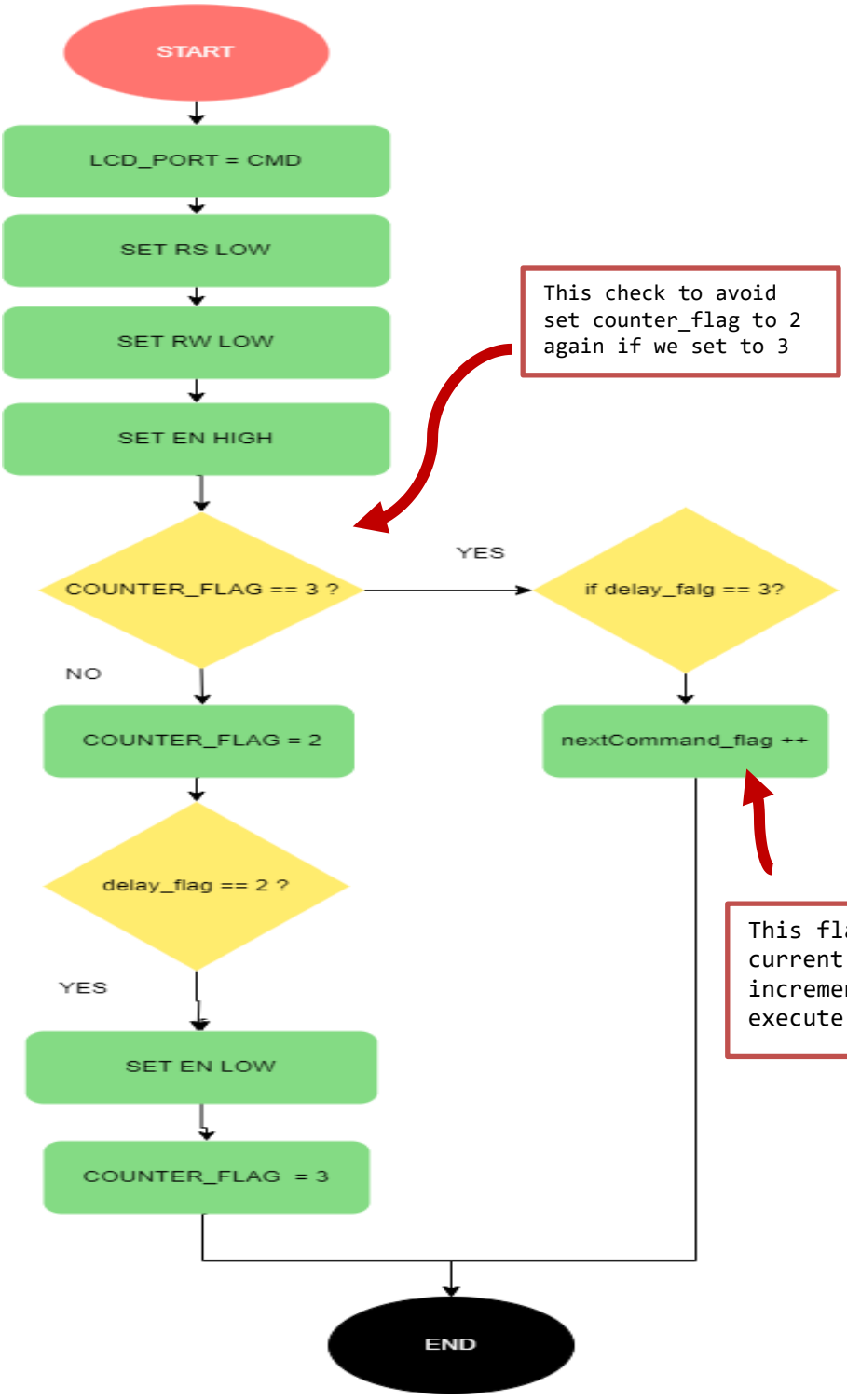


description

- This function will be passed to the timer setcallback function
- IF we set COUNTER_FLAG = 1 , timer will count 20ms then set delay_flag to 1 to indicate that the timer counting is done
- IF we set COUNTER_FLAG = 2 , timer will count 1microsec then set delay_flag to 2 to indicate that the timer counting is done
- IF we set COUNTER_FLAG = 3 , timer will count 1microsec then set delay_flag to 3 to indicate that the timer counting is done

LCD FLOWCHART

LCD_sendCommand

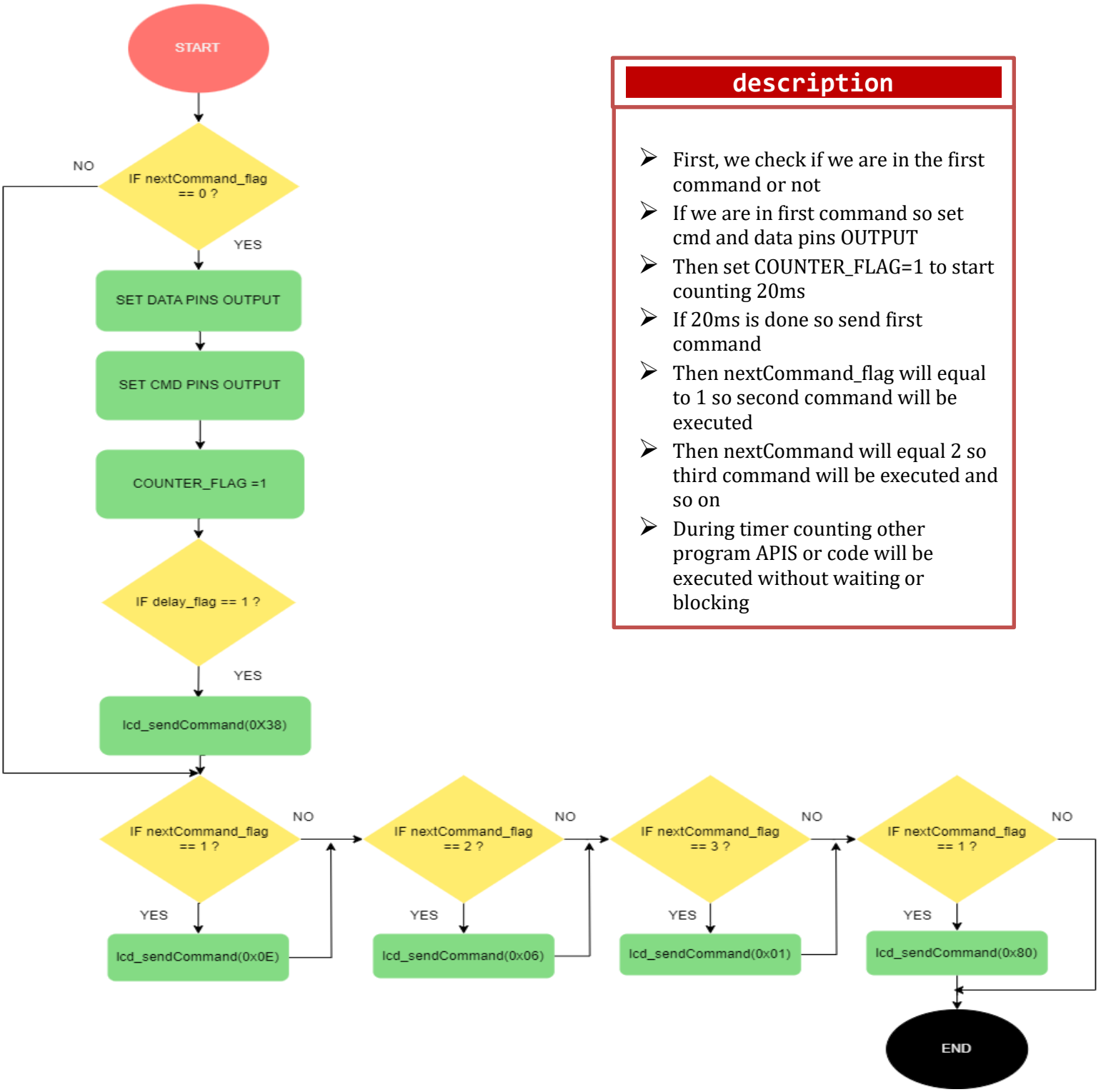


description

- This function is used to send commands to lcd by writing command in lcd data pins with set RS to low and the set enable to HIGH
- Then, we set COUNTER_FLAG = 2 to make timer indicate that we want to count 1 microsec and when it finishes counting we it will set the delay_flag = 2 so we check it if it equal 2 or not
- If delay_flag equals 2 then we set enable pin to LOW and set COUNTER_FLAG to 3 to count 2 ms
- If timer finish counting then delay_flag will equal 3 and increment nextCommand_flag by one to execute the next command

LCD FLOWCHART

LCD_init()



description

- First, we check if we are in the first command or not
- If we are in first command so set cmd and data pins OUTPUT
- Then set COUNTER_FLAG=1 to start counting 20ms
- If 20ms is done so send first command
- Then nextCommand_flag will equal to 1 so second command will be executed
- Then nextCommand will equal 2 so third command will be executed and so on
- During timer counting other program APIS or code will be executed without waiting or blocking

END