

Task 1

[illegible]

; 1DT301, Computer Technology I

```
; Lab number: 1
```

```
; Title: Task 1
```

i

```
; Hardware: STK600, CPU ATmega2560
```

;

```
; Function: To light LED 2 using PORTB as output
```

i

```
; Input ports: None
```

i

```
; Output ports: on-board LEDs connected to PORTB.
```

;

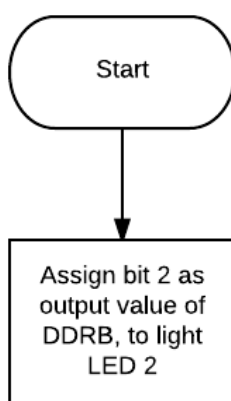
```
; Subroutines: If applicable.
```

;

[illegible]

```
ldi r16, 0x04      ; put decimal value 4 in register
```

```
out 0x04, r16      ; output value of r16 to address 0x04 (DDRB)
```



Task 2

```
; >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
; 1DT301, Computer Technology I

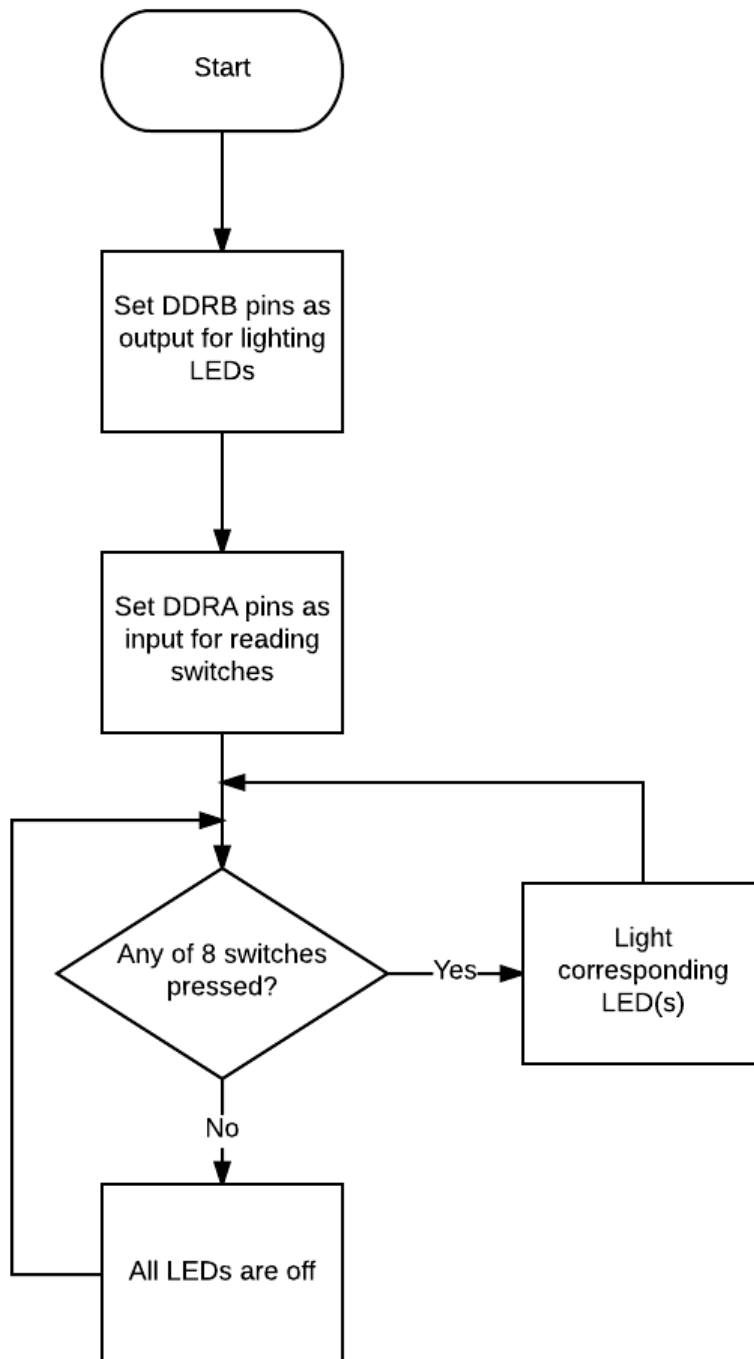
;
; Lab number: 1
; Title: Task 2
;
; Hardware: STK600, CPU ATmega2560
;
; Function: To read the switches and light the corresponding LED
;
; Input ports: Switches connected to port A
;
; Output ports: Leds connected to port B.
;
;
; <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

ldi r16, 0xFF
out DDRB, r16                ; set DDRB as output for lighting LEDs

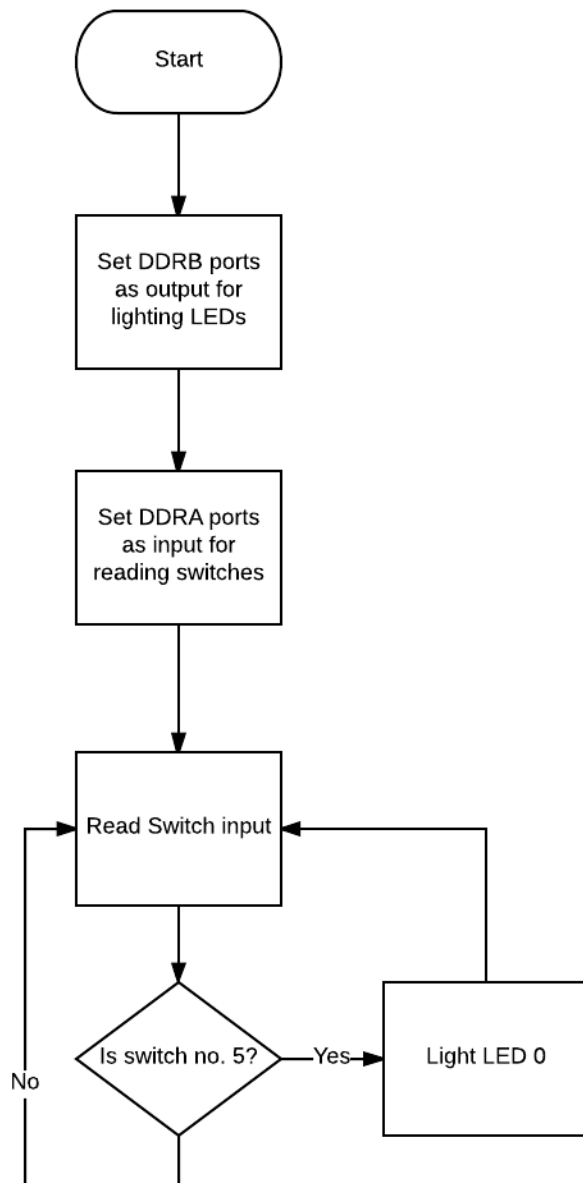
ldi r16, 0x00                ; set DDRA as input for reading switches
out DDRA, r16

loop:                        ; Keep a loop running indefinitely
    in r15, PINA             ; Read switch info from PIN A
    out PORTB, r15           ; Send this info to light LED on Port B.

rjmp loop
```




```
equal: out portB, r18      ; when run, this lights led 0
```



Notes regarding task 4: To light a LED, the bits representing the pins need to be inverted, because the pins are connected to ground on one side and will pull the port pin to ground (0) once they're pressed. This will result in a zero for a pressed button and a 1 for a released one. Once the port pin is low, it will sink current and the LED is lit.

Simplified: because of current and voltage specifications of the board, we have to invert the bits on the port pins to light the LED.

This was a little 'gotcha' while running the code in the simulator.

Task 5

[illegible]

```

delay:                ; 16MHz -> 16000000 cycles = 1s,
                      ; Cycles = 3a + 4ab + 3abc    ->  a(3 + b(4 + 3c))

    ldi r16, 11        ; -> a

delay_1:
    ldi r17, 237       ; -> b

delay_2:
    ldi r18, 255       ; -> c    (a,b,c ~> 0.5s)

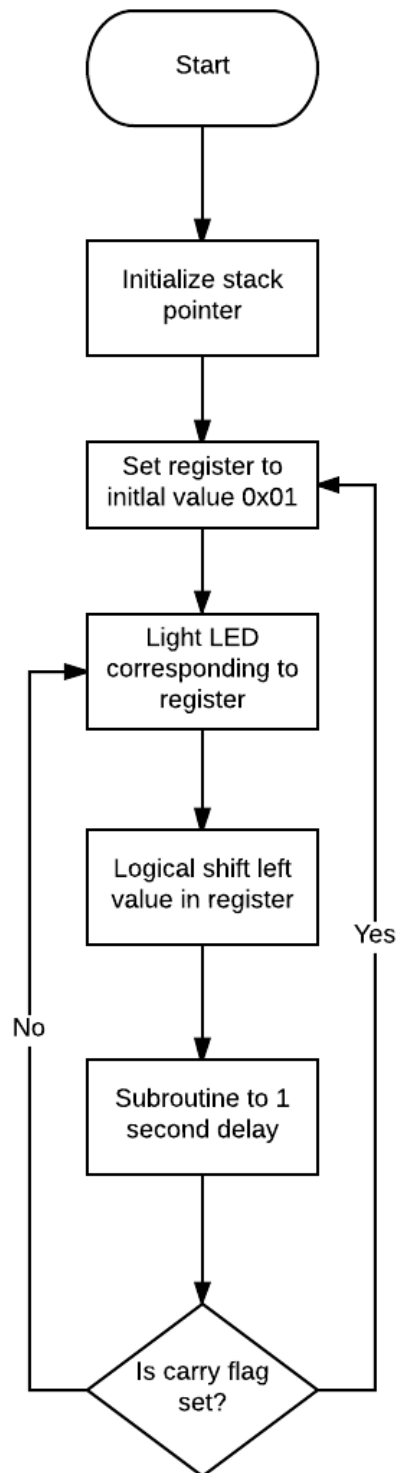
delay_3:
    dec r18            ; decrements value of r18 until 0 because
                      ; it will branch to delay_3 again if not zero.
    brne delay_3       ; 3c - 1          -> d = 3c - 1

    dec r17
    nop               ; delay_2 resets r18 to 255 for 237 times because
                      ; r17 is decremented each time when delay_3 is zero.
    brne delay_2       ; 5b - 1 + bd    -> e = 5b - 1 + 3cb - b

    dec r16            ; the whole process above is repeated 11 times.
    brne delay_1       ; 5a - 1 + ae    -> f = 3a + 5ab + 3abc - ab

    ret               ; f - 1

```



Task 6

[illegible]

```

    brne _add          ; run _add until all LEDs are lit.

_sub:

    out DDRB, r19
    lsr r19
    rcall delay

    brcs _sub          ; branch if carry flag is set in status register

    ldi r20, 0x01
    rjmp _add

delay:                  ; 16MHz -> 16000000 cycles = 1s,
                        ; Cycles = 3a + 4ab + 3abc    ->  a(3 + b(4 + 3c))

    ldi r16, 11        ; -> a

delay_1:
    ldi r17, 237        ; -> b

delay_2:
    ldi r18, 255        ; -> c  (a,b,c ~> 0.5s)

delay_3:
    dec r18             ; decrements value of r18 until 0 because
                        ; it will branch to delay_3 again if not zero.
    brne delay_3        ; 3c - 1          -> d = 3c - 1

    dec r17
    nop                 ; delay_2 resets r18 to 255 for 237 times because
                        ; r17 is decremented each time when delay_3 is zero.
    brne delay_2        ; 5b - 1 + bd    -> e = 5b - 1 + 3cb - b

    dec r16             ; the whole process above is repeated 11 times.
    brne delay_1        ; 5a - 1 + ae    -> f = 3a + 5ab + 3abc - ab

    ret                 ; f - 1

```

