Linnéuniversitetet
Kalmar Växjö

Lab Report

Lab 2

Arduino UNO, REV 3 – ATmega 328p

Author(s): David Mozart, Marcus Thornemo Larsson
Supervisor: Cristian Babes
Semester: HT19
Course Code: 1DT301

Table Of Content

Task I

Description

The program switches between Johnson- and ring-counter each time the switch is pressed.

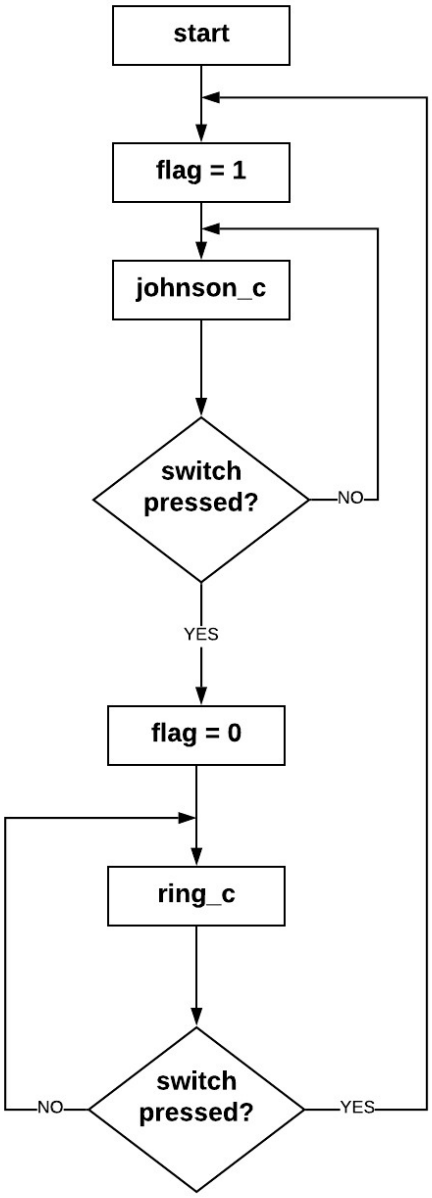
Assembly

```
ldi r20, HIGH(RAMEND)
OUT SPH, R20
ldi R20, low(RAMEND)
out SPL, R20

ldi r16, 0xff
out DDRB, r16
clr r16
out DDRD, r16

ldi r16, 0b0000_0000
ldi r17, 0b0000_0001
out PORTB, r16
ldi r23, 0xff

main:
            rcall johnson_counter_inc

ring_counter:
            out PORTB, r17
            rcall delay
            cpi r17, 0b0010_0000
            breq reset_ring_counter
            lsl r17
            rjmp ring_counter

reset_ring_counter:
            ldi r17, 0b0000_0001
            rjmp ring_counter


johnson_counter_inc:
            add r16, r17
            out PORTB, r16
            cpi r17, 0b0100_0000
            breq johnson_counter_dec
            lsl r17
            rcall delay
            rjmp johnson_counter_inc
```

```
johnson_counter_dec:
            lsr r17
            sub r16, r17
            out PORTB, r16
            rcall delay
            cpi r17, 0b0000_0001
            breq johnson_counter_inc
            rjmp johnson_counter_dec


switch_pressed:
            clr r24
            in r24, PIND
            andi r24, 0b0000_1000
            cpi r24, 0b0000_1000
            breq toggle_flag
            reti
toggle_flag:
            cpi r23, 0x00
            breq set_high
            rcall set_low
            reti

set_low:
            rcall switch_delay
            ldi r23, 0x00
            ldi r16, 0b0000_0000
            ldi r17, 0b0000_0001
            jmp ring_counter

set_high:
            rcall switch_delay
            ldi r23, 0xff
            ldi r16, 0b0000_0000
            ldi r17, 0b0000_0001
            jmp johnson_counter_inc


delay:
            ldi r18, 4
            ldi r19, 5
            ldi r20, 50
L1:
            rcall switch_pressed
            dec r20
            brne L1
            rcall switch_pressed
            dec r19
            brne L1
            rcall switch_pressed
            dec r18
            brne L1
reti
```

```
switch_delay:
    ldi  r18, 21
    ldi  r19, 75
    ldi  r20, 191
L2: dec  r20
    brne L2
    dec  r19
    brne L2
    dec  r18
    brne L2
reti
```

Flowchart


start
flag = 1
johnson_c
switch pressed?
NO
YES
flag = 0
ring_c
switch pressed?
NO
YES

Task II

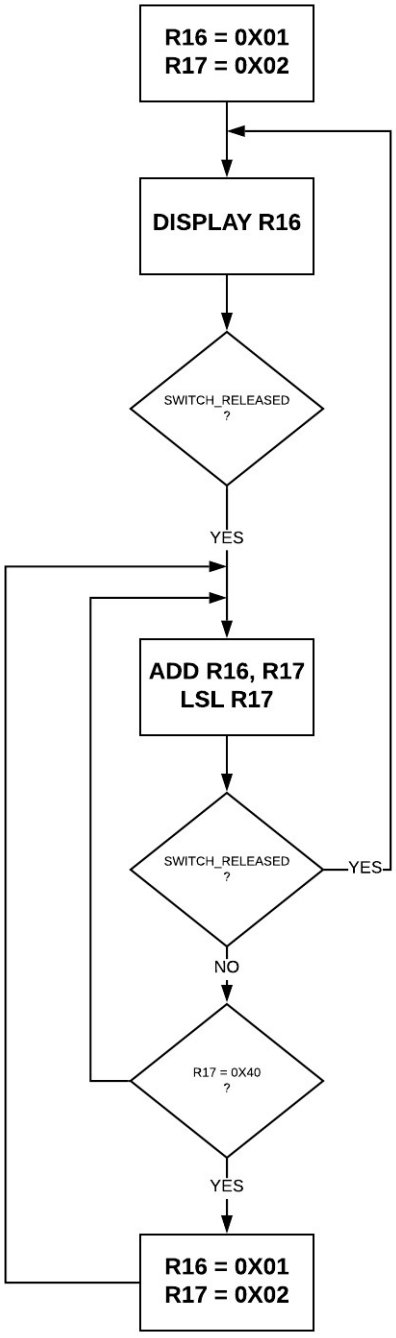
Description

The program displays, through the LEDs, different dice results [1...6] each time the switch is released.

Assembly

```
ldi r20, HIGH(RAMEND)
OUT SPH, R20
ldi R20, low(RAMEND)
out SPL, R20

ldi r16, 0xff
out DDRB, r16
clr r16
out DDRD, r16

ldi r16, 0b0000_0001
ldi r17, 0b0000_0010
ldi r18, 0x00
out PORTB, r16

main:
            out PORTB, r16
            in r18, PIND
            andi r18, 0b0000_1000
            cpi r18, 0b0000_1000
            breq switch_pressed
            rjmp main

switch_pressed:
            ldi r19, 0x00
            out PORTB, r19
            in r18, PIND
            andi r18, 0b0000_1000
            cpi r18, 0b0000_1000
            breq randomize
            rcall main
randomize:
            add r16, r17
            lsl r17
            cpi r17, 0b1000_0000
            breq reset
            rjmp switch_pressed

reset:
            ldi r16, 0b0000_0001
            ldi r17, 0b0000_0010
            rjmp switch_pressed
```

Flowchart


R16 = 0X01
R17 = 0X02
DISPLAY R16
SWITCH_RELEASED ?
YES
ADD R16, R17
LSL R17
SWITCH_RELEASED ?
YES
NO
R17 = 0X40 ?
YES
R16 = 0X01
R17 = 0X02

Task III

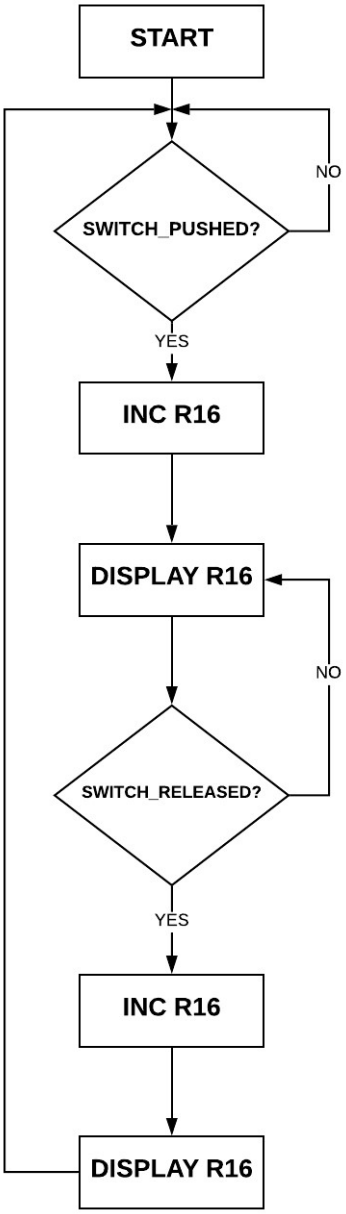
Description

The program displays the number of times (in binary, 8-bits) the switch has been pressed and released.

Assembly

```
ldi r16, 0xff
out DDRB, r16
clr r16

ldi r17, 0x00
out DDRD, r17

start:
            in r17, PIND
            andi r17, 0b0000_1000
            cpi r17, 0b0000_1000
            breq pushed
            rjmp start

pushed:
            inc r16
            out PORTB, r16
            rjmp loop

loop:
            in r17, PIND
            andi r17, 0b0000_1000
            cpi r17, 0b0000_0000
            breq released
            rjmp loop

released:
            inc r16
            out PORTB, r16
            rjmp start
```

Flowchart


START
SWITCH_PUSHED?
NO
YES
INC R16
DISPLAY R16
SWITCH_RELEASED?
NO
YES
INC R16
DISPLAY R16

Task IV

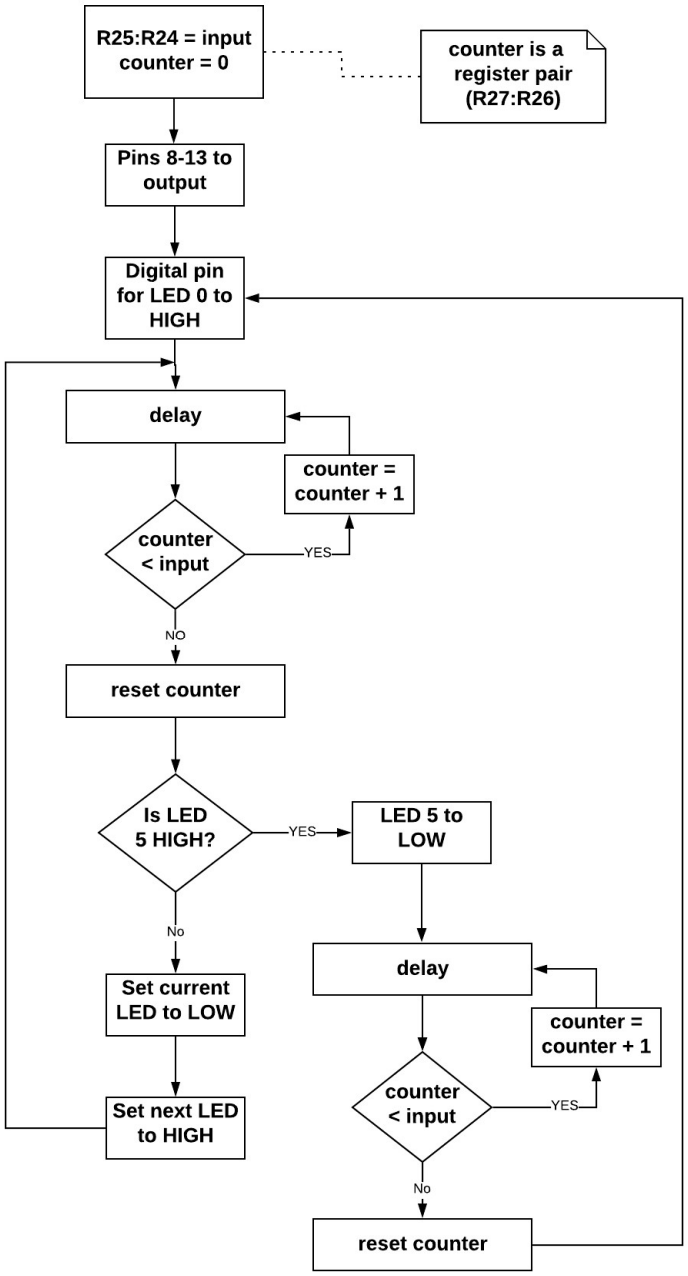
Description

This program takes a number-of-milliseconds as input and delays the transition between each LED in the ring counter according to that.

Assembly

```
ldi r20, HIGH(RAMEND)
OUT SPH, R20
ldi R20, low(RAMEND)
out SPL, R20

ldi r16, 0xff
out DDRB, r16
ldi r16, 0b0000_0001

ldi r25, 0x00
ldi r24, 0x05

ldi r27, 0x00
ldi r26, 0x00

ring_counter:
    out PORTB, r16
    cpi r16, 0b0100_0000
    brge reset
    lsl r16
    rcall wait_milliseconds
    rjmp ring_counter

reset:
    ldi r16, 0b0000_0001
    rjmp ring_counter

wait_milliseconds:
    cp r26, r24
    cpc r27, r25
    brne delay
    clr r26
    clr r27
reti

delay:
    adiw r27:r26, 1
    ldi  r18, 21
    ldi  r19, 199
L1: dec  r19
    brne L1
    dec  r18
    brne L1
rjmp wait_milliseconds
```

Flowchart


R25:R24 = input
counter = 0
counter is a register pair (R27:R26)
Pins 8-13 to output
Digital pin for LED 0 to HIGH
delay
counter = counter + 1
counter < input
YES
NO
reset counter
Is LED 5 HIGH?
YES
LED 5 to LOW
No
Set current LED to LOW
Set next LED to HIGH
delay
counter = counter + 1
counter < input
YES
No
reset counter