Embedded Systems Lab Experiment 8

Pulse Width Modulation using ATMega 32 microcontroller

To program ATMega 32 microcontroller to accept multiple analogue inputs and mux among them and using the digital value to control the led brightness by PWM technique
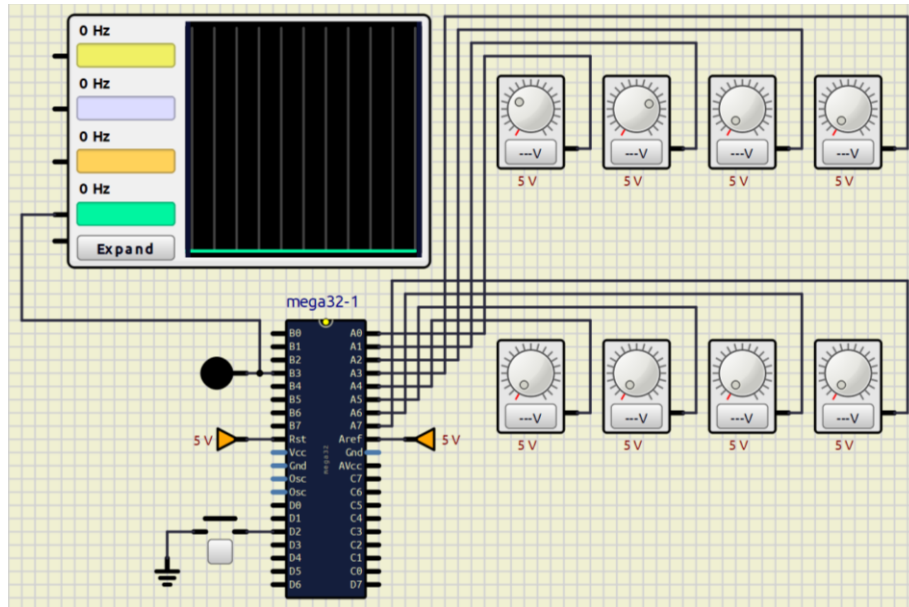
Jitbitan Baroi
3-17-2024

•) <u>Objective</u>: To program an At Mega 32 µC to accept multiple analog inputs and mux among them and using that to the digital value to control the led brightness.

•) <u>Apparatus reqd</u>.:

Name	Specification	Quantity
1. µC	At Mega 32	1
2. variable voltage sources	~0 - 5V	8
3. LED	0 -	1
4. push button	—	1

Schematic used for mux multiples analog inputs at Pin A and controlling the brightness of the LED using one analog input at a time and the PWM signal is shown in the oscilloscope
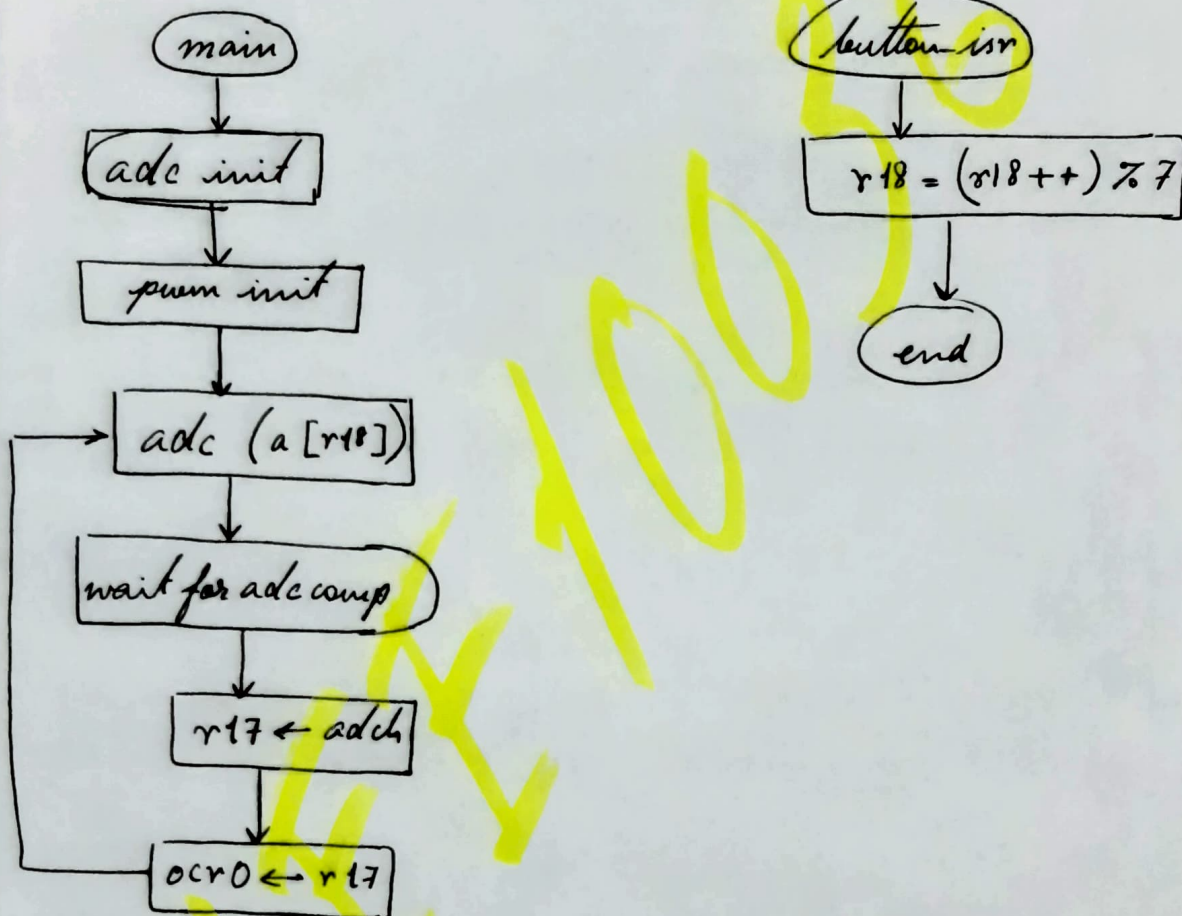
- Flow of the program:

```
        main
          │
          ▼
     ┌──────────┐
     │ adc init │
     └──────────┘
          │
          ▼
     ┌──────────┐
     │ pwm init │
     └──────────┘
          │
          ▼
 ┌─►┌────────────┐
 │  │ adc (a[r18])│
 │  └────────────┘
 │        │
 │        ▼
 │  ┌────────────────────┐
 │  │ wait for adc comp  │
 │  └────────────────────┘
 │        │
 │        ▼
 │  ┌────────────┐
 │  │ r17 ← adch │
 │  └────────────┘
 │        │
 │        ▼
 │  ┌──────────────┐
 └──│ ocr0 ← r17   │
    └──────────────┘
```

```
      button isr
          │
          ▼
  ┌──────────────────┐
  │ r18 = (r18++) % 7│
  └──────────────────┘
          │
          ▼
        end
```

• Code:

```
. include "m32def.inc"          →
. org 0                         →
    jmp start                   →  reset intrr
. org 2                         →
    jmp int_isr_0               →  int 0 intrr
start:                          →
    // setting sph & spl.       →
    ldi r16, 0x00               →
    out ddra, r16               →  ⎫  port a and portd as ip
    out dddrd, r16              →  ⎭
    ldi r16, 0xFF               →
    out ddrb, r16               →  port b as output
    out portd, r16              →
    sei.                        →  enable intrr.
    ldi r16, 0x40               →  ⎫
    out gicr, r16               →  ⎬  gicr ← 0x40
    ldi r16, 0x03               →  ⎫
    out mcucr, r16              →  ⎬  mcucr ← 0x03
    ldi r30, 0x07               →
```

```
ldi r18, 0x00        →
ldi r19, 0x20        →
ldi r16, 0x69        →
out tccr0, r16       →  enable timer
ldi r20, 0x20        →
main: out adadmux, r20  →  admux ← r20
ldi r16, 0x87        →  }
out adcsra, r16      →  } set enable adc.
ldi r16, 0xc7        →  }
out adcsra, r16      →  } start adc.
l1: in r16, adcsra   →  }
andi r16, 0x10       →  } wait for adc complete.
breq l1              →  }
in r17, adch         →  adch → r17
out ocr0, r17        →  ocr0 ← r17.
jmp main             →
out isr0: inc r18    →  r18++
andi r18, 0x07       →  r18 ← r18 % 7
mov r20, r18         →  r20 ← r18.
ori r20, 0x20        →  r20 ← r20 or 0x20
reti                    return
```
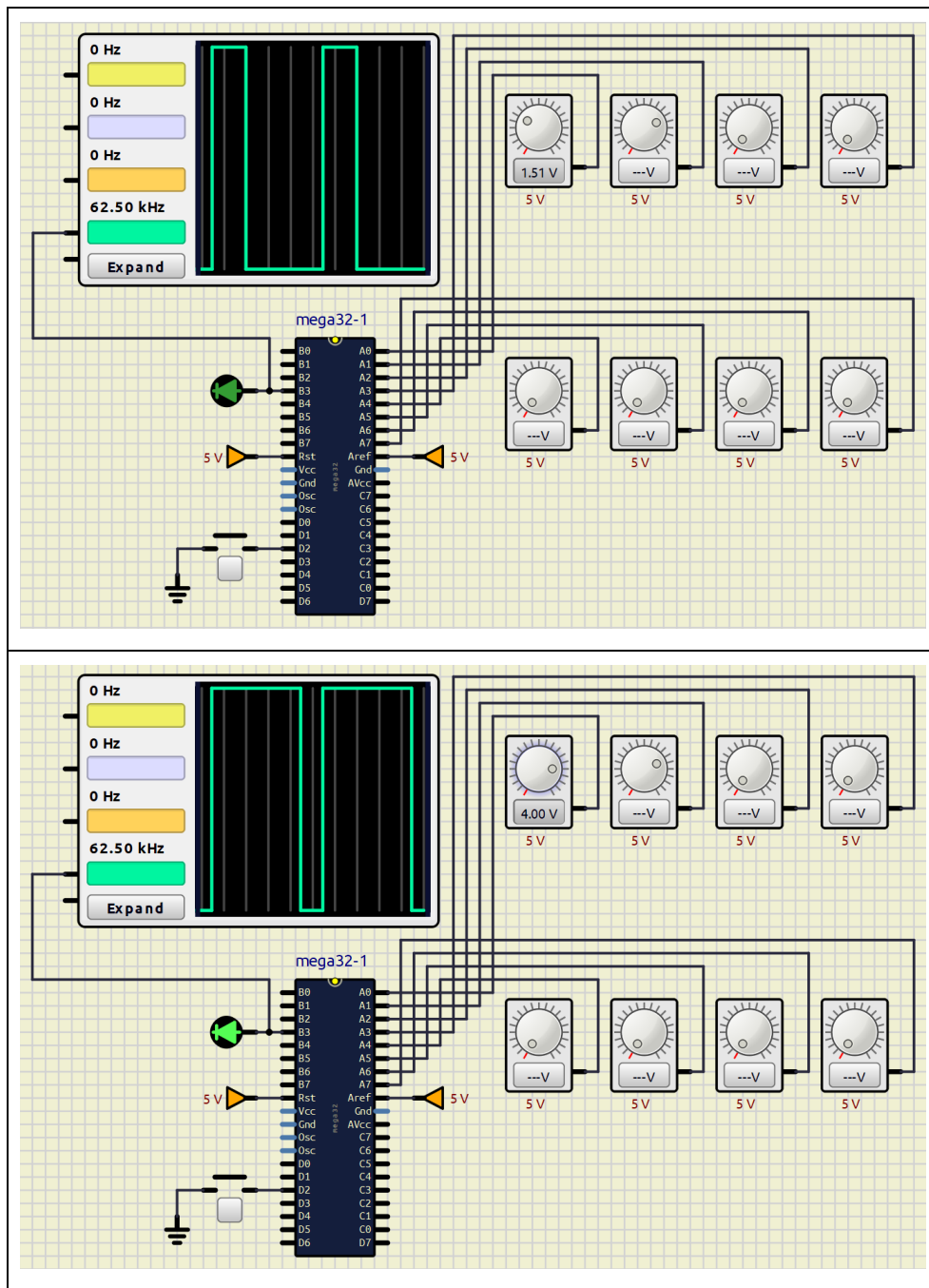
Two different analog voltage levels from variable voltage source 1 are given and the corresponding brightness of the LED as well as the PWN signal is shown
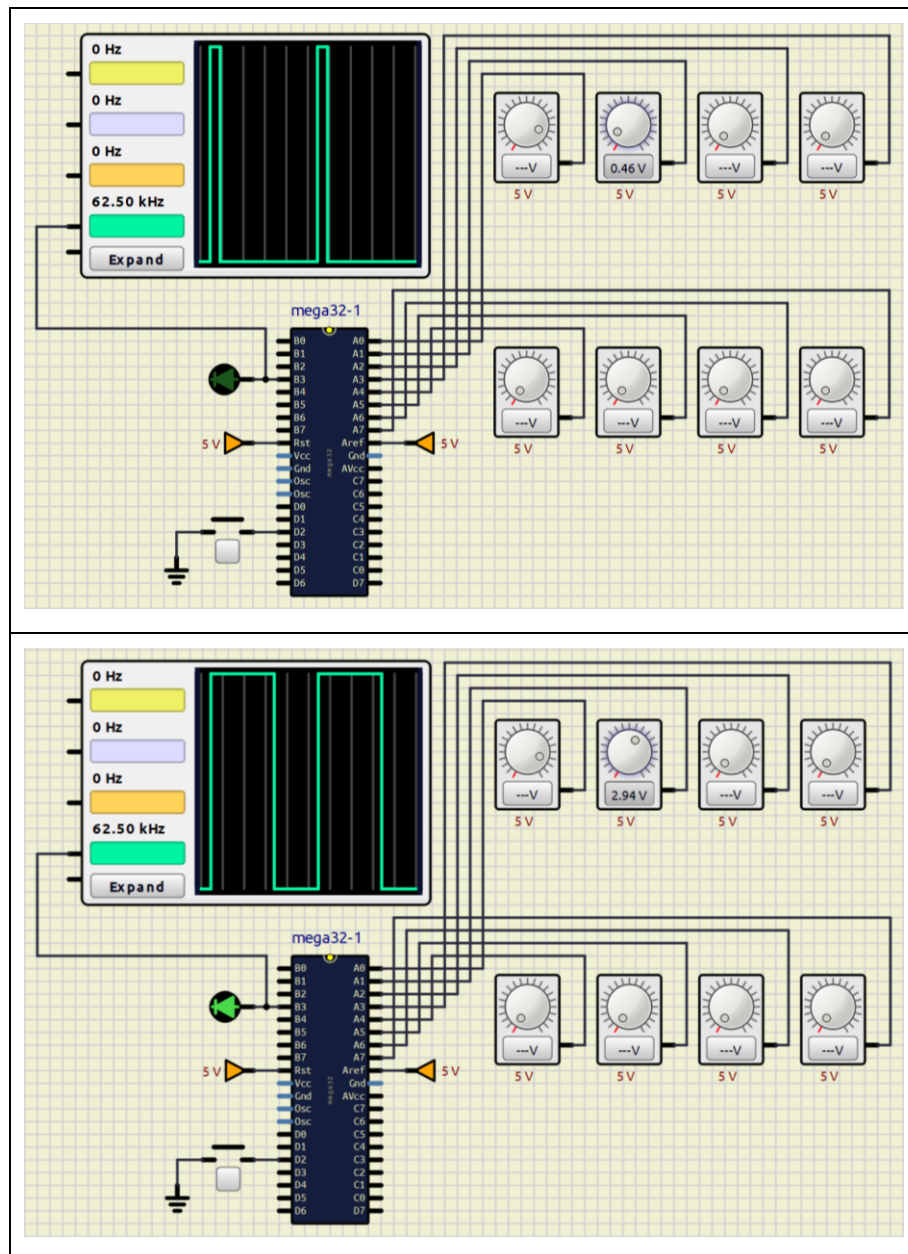
Two different analog voltage levels from variable voltage source 1 are given and the corresponding brightness of the LED as well as the PWN signal is shown control is switched from variable voltage source 1 to 2 by push button