

**INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR**  
**EMBEDDED SYSTEM LAB REPORT**  
**Course Code – EE39004**

---

**Experiment - LED blinking and Programmable Waveform Generation**

---

**Name:** Ayush Pandit  
**Roll No.:** 19EE10067  
**Department:** Electrical Engineering

---

## **Experiment 1a:**

### **1. Aim of the Experiment:**

To Program ATMEGA32 to produce voltage signal pulse used for the blinking of a LED

### **2. Requirements:**

1. ATMEGA32
2. LED

### **3. Procedure of the Experiment:**

To perform the experiment we need to build the required program in assembly language and use the hex file in simulation in Proteus.

The Algorithm and pseudo code are mentioned here:-

Set output at port A

Infinite loop:

Set port A output to high

Delay function

Set port A output to low

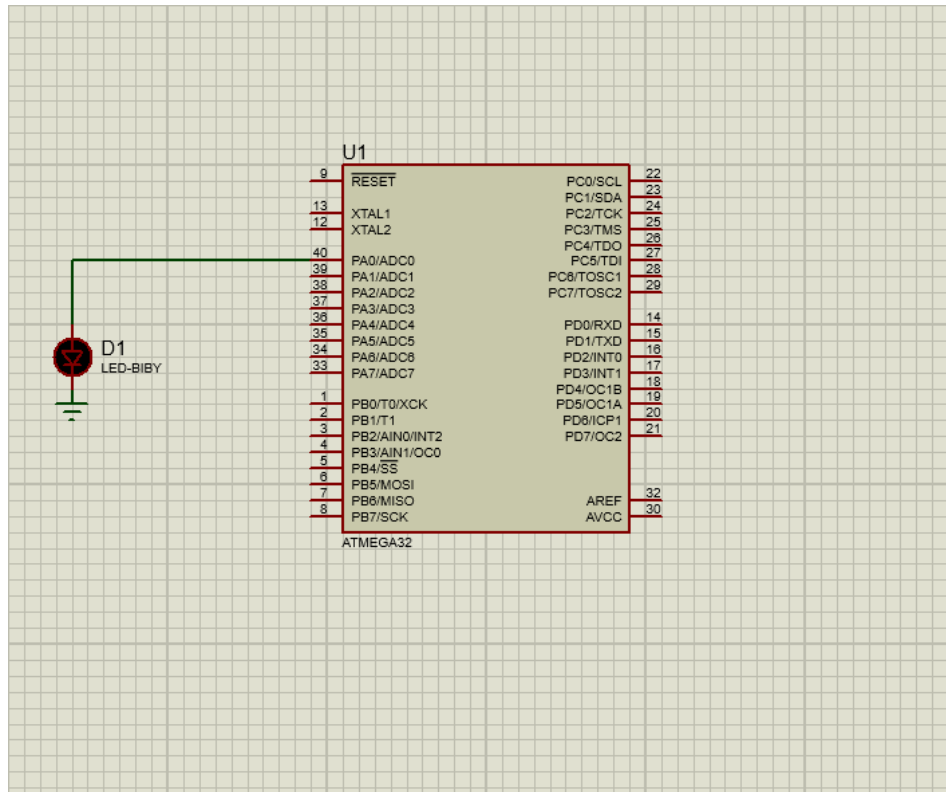
Delay function

Delay function:

Nested loop

Return subroutine

#### 4. Schematic diagram in Proteus:



#### 5. Assembly Program:

```
;
;
; LED_code.asm
```

```
;
; Created: 14-01-2022 14:55:20
; Author : ayush
;
```

```
; Replace with your application code
.INCLUDE "m32DEF.INC"
.ORG 0X00
```

```
LDI R16, HIGH(RAMEND)
OUT SPH, R16
LDI R16, LOW(RAMEND)
OUT SPL, R16
```

```
LDI R16, 0XFF
OUT DDRA, R16
```

LOOP:

```
LDI R16, 0xFF
OUT PORTA, R16
CALL DELAY
LDI R16, 0x00
OUT PORTA, R16
CALL DELAY
```

DELAY:

```
LDI R16, 0
LDI R17, 0
LOOP1:
DEC R16
BRNE LOOP1
DEC R17
BRNE LOOP1
RET
RJMP LOOP
```

## **6. Results:**

The simulated output as observed in the lab is produced. The LED blinks as long as the simulation runs, due to the infinite loop programmed in the assembly code.

## **7. Discussions:**

Our aim of the experiment is to program ATMEGA32 for the blinking of the LED, so we must produce a square wave voltage output from ATMEGA32 to produce the desired output. The clock frequency of ATMEGA32 is 1MHz. So, if we use normal instructions for switching, the program will use only a few clock cycles for the blinking of the LED, so we use the extra DELAY function to increase the computing time by nested loops. This makes the blinking of the LED noticeable.

---

## **Experiment 1b:**

### **1. Aim of the Experiment:**

To Program ATMEGA32 to generate voltage signal which produces sawtooth waveform when passed through a digital to analog converter.

### **2. Requirements:**

1. ATMEGA32
2. DAC0808(Digital to analog converter)

3. Resistors - 3 (5k $\Omega$ )
4. Capacitor - 1 (0.1 $\mu$ F)
5. Oscilloscope

### 3. Procedure of the Experiment:

To perform the experiment we need to build the required program in assembly language and use the hex file in simulation in Proteus.

The Algorithm and pseudo code are mentioned here:-

Set output at port B

Infinite loop:

Set port B output to high(11111111)

Fall:

Delay function

Decrease the output of port B by 1

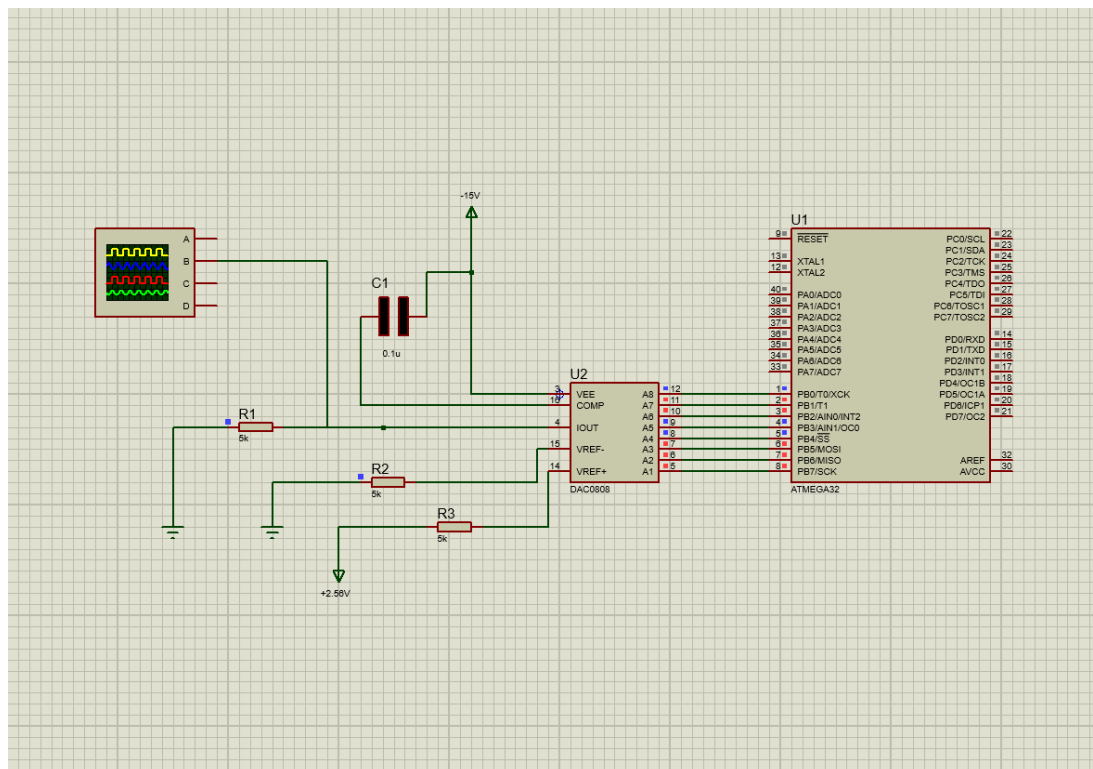
Call Fall function till output of port B is low(00000000)

Delay function:

Nested loop

Return subroutine

### 4. Schematic diagram in Proteus:



## 5. Assembly Program:

```
;
;
; sawTooth.asm
;
; Created: 14-01-2022 15:36:57
; Author : ayush
;

; Replace with your application code
.INCLUDE "m32DEF.INC"
.ORG 0X00

LDI R16, HIGH(RAMEND)
OUT SPH, R16
LDI R16, LOW(RAMEND)
OUT SPL, R16

LDI R16, 0XFF
OUT DDRB, R16

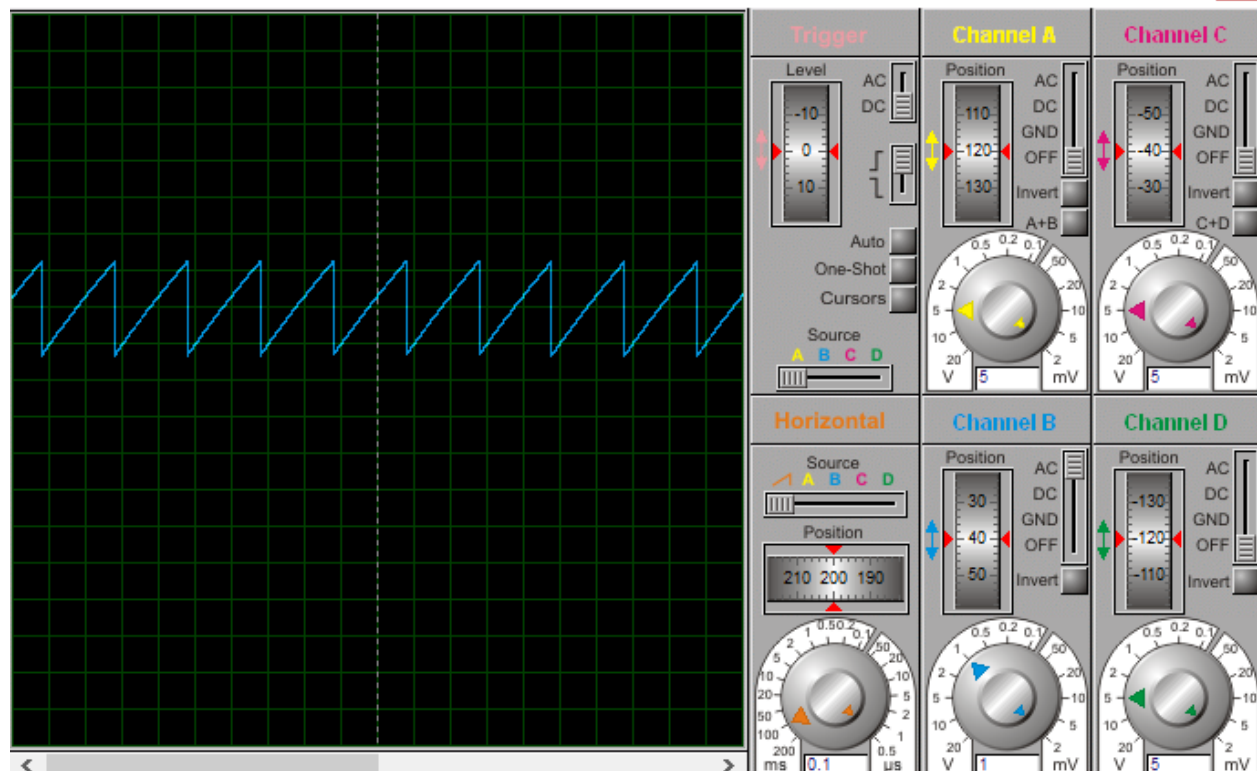
LOOP:
    LDI R16, 0XFF
    FALL:
        OUT PORTB, R16
        CALL DELAY
        DEC R16
        BRNE FALL

DELAY:
    LDI R17, 0
    LOOP1:
        DEC R17
        BRNE LOOP1
    RET
    RJMP LOOP
```

## 6. Results:

The simulated output as observed in the lab is produced. The DAC output can be observed in an oscilloscope, which shows a sawtooth wave output.

Digital Oscilloscope



## 7. Discussions:

We give input 0XFF to 0X00 to DAC with a delay. DAC produces minimum -ve voltage output when the input is high(0XFF) which gradually increases and becomes 0V as the input becomes 0X00. Hence the output voltage increases from minimum to maximum output voltage gradually because of the delay introduced in the program. But it jumps back to 0XFF in very few clock cycles as there is no delay involved. Thus a sawtooth wave is generated.

---