## Objective: AVR Microcontroller GPIO Programming In C.

**Task-1** Assume that 8 LEDs are connected to PORTA of ATMEGA32A. Write a program to generate various blinking patterns(minimum 10)

**Programm :-**

#include <avr/io.h>

#include <avr/delay.h>

#include <util/delay.h>

#define F\_CPU 16000000UL;

void Pattern\_0() {

PORTA = 0xFF;

\_delay\_ms(500);

PORTA = 0x00;

\_delay\_ms(500);

}

void Pattern\_1() {

PORTA = 0xAA;

\_delay\_ms(500);

PORTA = 0x55;

\_delay\_ms(500);

}

void Pattern\_2() {

PORTA = 0xF0;

\_delay\_ms(500);

PORTA = 0x0F;

\_delay\_ms(500);

}

void Pattern\_3() {

PORTA = 0x00;

\_delay\_ms(500);

PORTA = 0x01;

while (PORTA != 0x00) {

\_delay\_ms(500);

PORTA = PORTA << 1;

if (PORTA == 0xFF) {

PORTA == 0x01;

}

}

}

void Pattern\_4() {

PORTA = 0xFF;

\_delay\_ms(500);

int Sequences[8] = { 0xFE,0xFD,0xFB,0xF7,0xEF,0xDF,0xBF,0x7F };

int i = 0;

while (i <= 9) {

PORTA = Sequences[i];

\_delay\_ms(500);

if (i == 9) {

i = -1;

PORTA = 0xFF;

\_delay\_ms(500);

}

i++;

}

}

void Pattern\_5() {

PORTA = 0x00;

\_delay\_ms(500);

PORTA = 0x80;

while (PORTA != 0x00) {

\_delay\_ms(500);

PORTA = PORTA >> 1;

if (PORTA == 0x01) {

PORTA == 0x80;

}

}

}

void Pattern\_6() {

PORTA = 0x00;

\_delay\_ms(500);

int Sequences[7] = { 0x81,0x42,0x24,0x18,0x24,0x42,0x81 };

int i = 0;

while (i <= 7) {

PORTA = Sequences[i];

\_delay\_ms(500);

i++;

if (i == 8) {

i = 0;

}

}

}

void Pattern\_7() {

PORTA = 0xFF;

\_delay\_ms(500);

int Sequences[7] = { 0x7E,0xBD,0xDB,0xE7,0xDB,0xBD,0x7E };

int i = 0;

while (i <= 7) {

PORTA = Sequences[i];

\_delay\_ms(500);

i++;

if (i == 8) {

i = 0;

}

}

}

void Pattern\_8() {

PORTA = 0x00;

\_delay\_ms(500);

PORTA = 0X01 ;

\_delay\_ms(500);

while(1) {

PORTA = PORTA << 1 ;

\_delay\_ms(500);

if (PORTA == 0x80) {

PORTA = 0x01 ;

\_delay\_ms(500);

}

}

}

void Pattern\_9() {

PORTA = 0xFF;

\_delay\_ms(500);

PORTA = 0x80 ;

\_delay\_ms(500);

while (1) {

PORTA = PORTA >> 1 ;

\_delay\_ms(500);

if (PORTA == 0x01) {

PORTA = 0x80 ;

\_delay\_ms(500);

}

}

}

int main(void)

{

/\* Replace with your application code \*/

DDRA = 0xFF;

DDRB = 0x00;

DDRB = DDRB | 0xF0;

while (1) {

if (PINB == 0x00) {

Pattern\_0();

}

if (PINB == 0x01) {

Pattern\_1();

}

if (PINB == 0x02) {

Pattern\_2();

}

if (PINB == 0x03) {

Pattern\_3();

}

if (PINB == 0x04) {

Pattern\_4();

}

if (PINB == 0x05) {

Pattern\_5();

}

if (PINB == 0x06) {

Pattern\_6();

}

if (PINB == 0x07) {

Pattern\_7();

}

if (PINB == 0x08) {

Pattern\_8();

}

if (PINB == 0x09) {

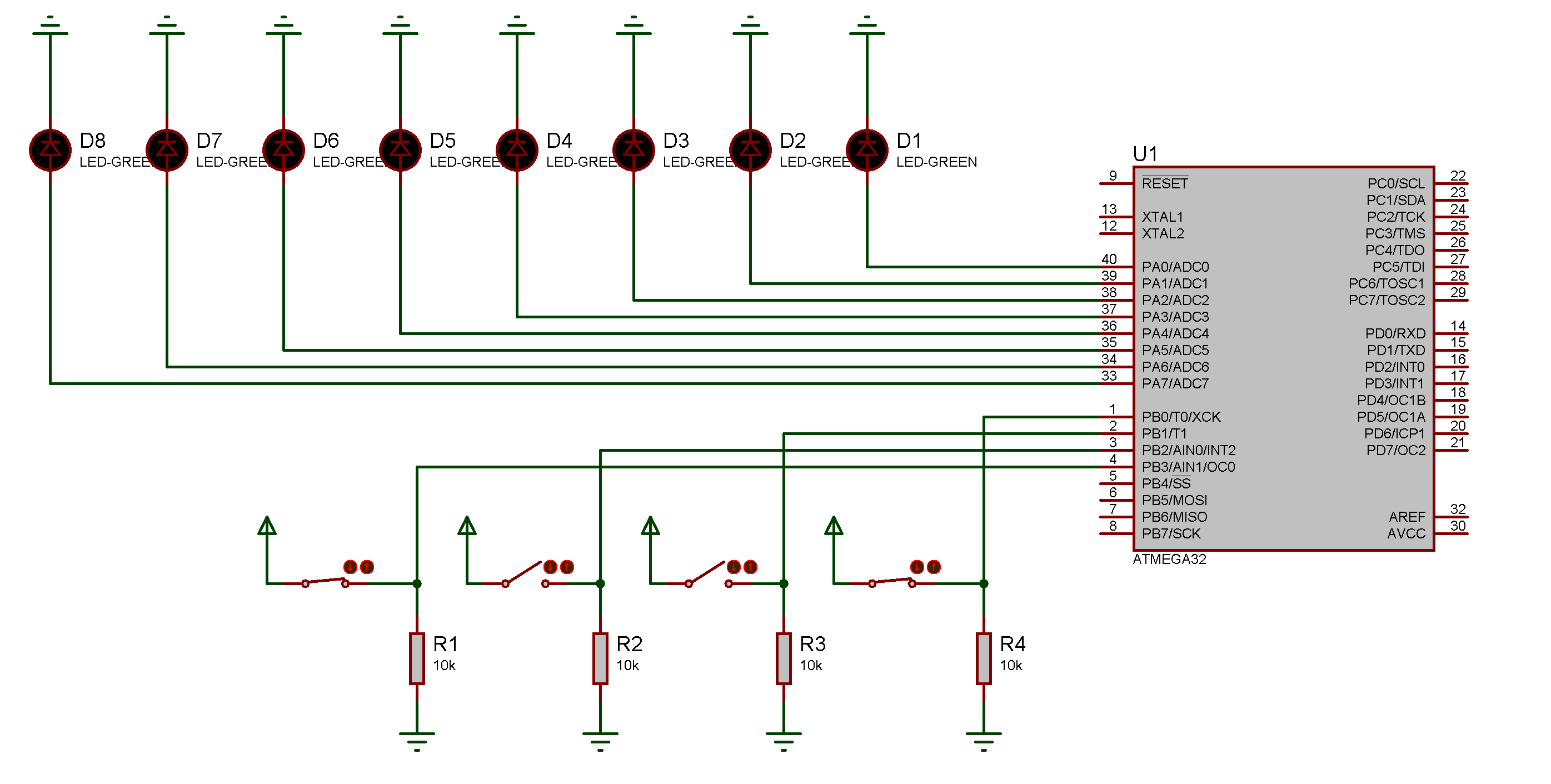
Pattern\_9();

}

}

}

**Circuit :-**



**Objective:** Assume that 8 LEDs are connected to PORTA of ATMEGA32 and two switches are connected to PC0 and PC1. Read the status of both the switches are perform the respective tasks accordingly.

1. If both the switches are off, all LEDs should turn on/off with one second of delay

2. If SW1 is on and SW2 is off, LEDs should turn on/off one by one from left to right(LSB to MSB)

3. If SW1 is off and SW2 is on, LEDs should turn on/off one by one from right to left(MSB to LSB)

4. if Both the switches are pressed, every alternate LEDs should turn on/off.

**Programm:-**

#include <avr/io.h>

#include <util/delay.h>

#define F\_CPU 16000000UL

int main(void)

{

DDRA = 0xFF ;

DDRC = 0x00;

DDRC = DDRC | 0x03 ;

/\* Replace with your application code \*/

while (1) {

if (PINC == 0x00) {

PORTA = 0x00 ;

\_delay\_ms(1000);

PORTA = 0xFF ;

\_delay\_ms(1000);

}

else if (PINC == 0x01) {

PORTA = 0x01 ;

\_delay\_ms(1000);

while (PORTA != 0x80) {

PORTA = PORTA << 1 ;

\_delay\_ms(1000);

if(PORTA == 0x80) {

PORTA = 0x01 ;

\_delay\_ms(1000) ;

}

}

}

else if (PINC == 0x02) {

PORTA = 0x80 ;

\_delay\_ms(1000);

while (PORTA != 0x01) {

PORTA = PORTA >> 1 ;

\_delay\_ms(1000);

if(PORTA == 0x01) {

PORTA = 0x80 ;

\_delay\_ms(1000) ;

}

}

}

else if (PINC == 0x03) {

while (1) {

PORTA = 0xAA ;

\_delay\_ms(1000) ;

PORTA = 0x55 ;

\_delay\_ms(1000) ;

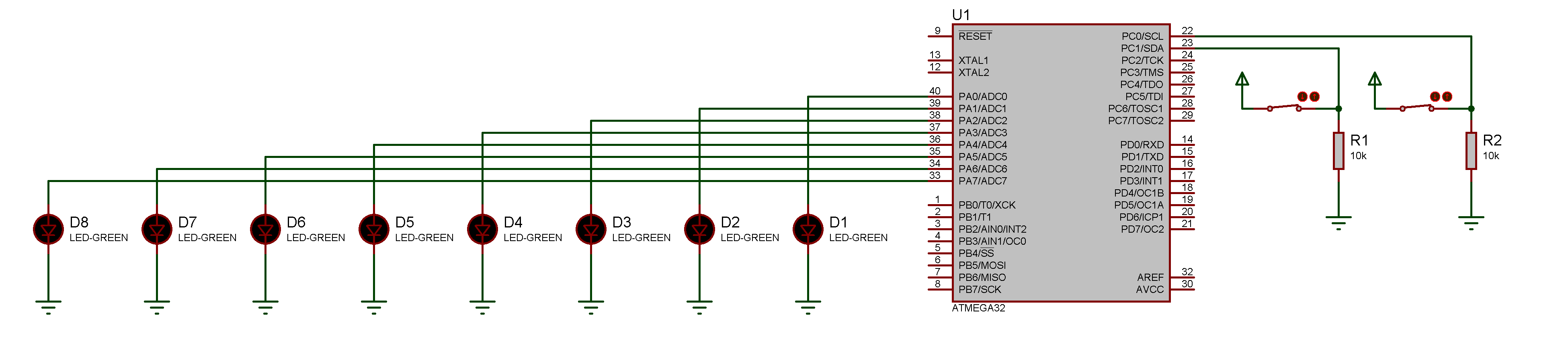
}

}

}

}

**Circuit :-**



**Objective:** Assume that a seven segment display is connected to PORT A. Perform the following tasks

1. **Display decimal up counter**

## Programm:-

## #include <avr/io.h>

## #include <avr/delay.h>

## #include <util/delay.h>

## #define F\_CPU 16000000UL ;

## int main(void)

## {

## int HexCodes[10] = {0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x98};

## DDRA = 0xFF;

## int i = 0 ;

## while (i <= 9) {

## PORTA = HexCodes[i];

## \_delay\_ms(1000);

## i++;

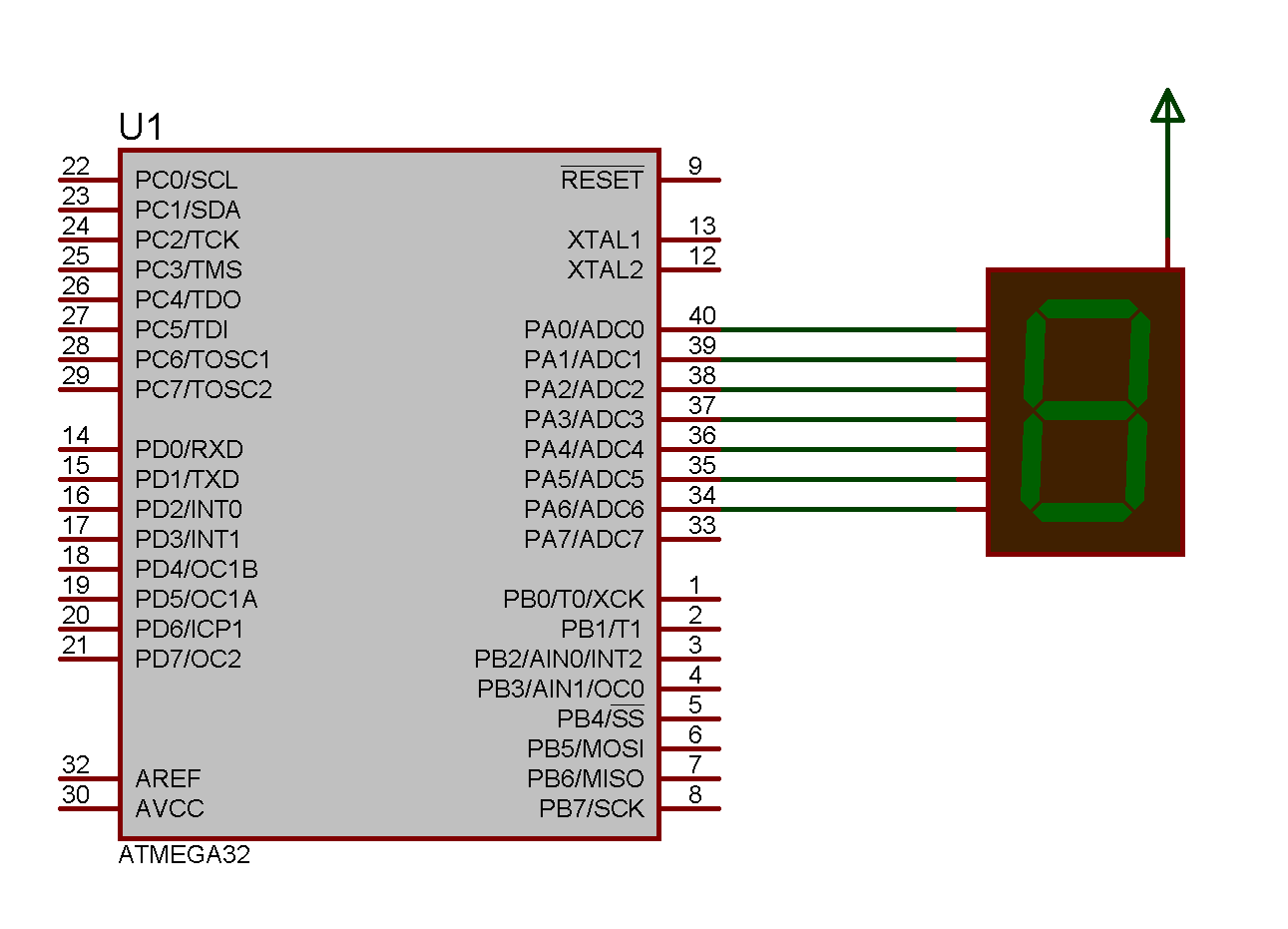
## if(i == 10) {

## i = 0 ;

## }

## }

## }

**Circuit :-**

1. **Dispplay decimal down counter**

## Programm:-

## #include <avr/io.h>

## #include <avr/delay.h>

## #include <util/delay.h>

## #define F\_CPU 16000000UL

## int main(void)

## {

## int HexCodes[10] = {0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x98};

## DDRA = 0xFF;

## int i = 9 ;

## while (i >= 0) {

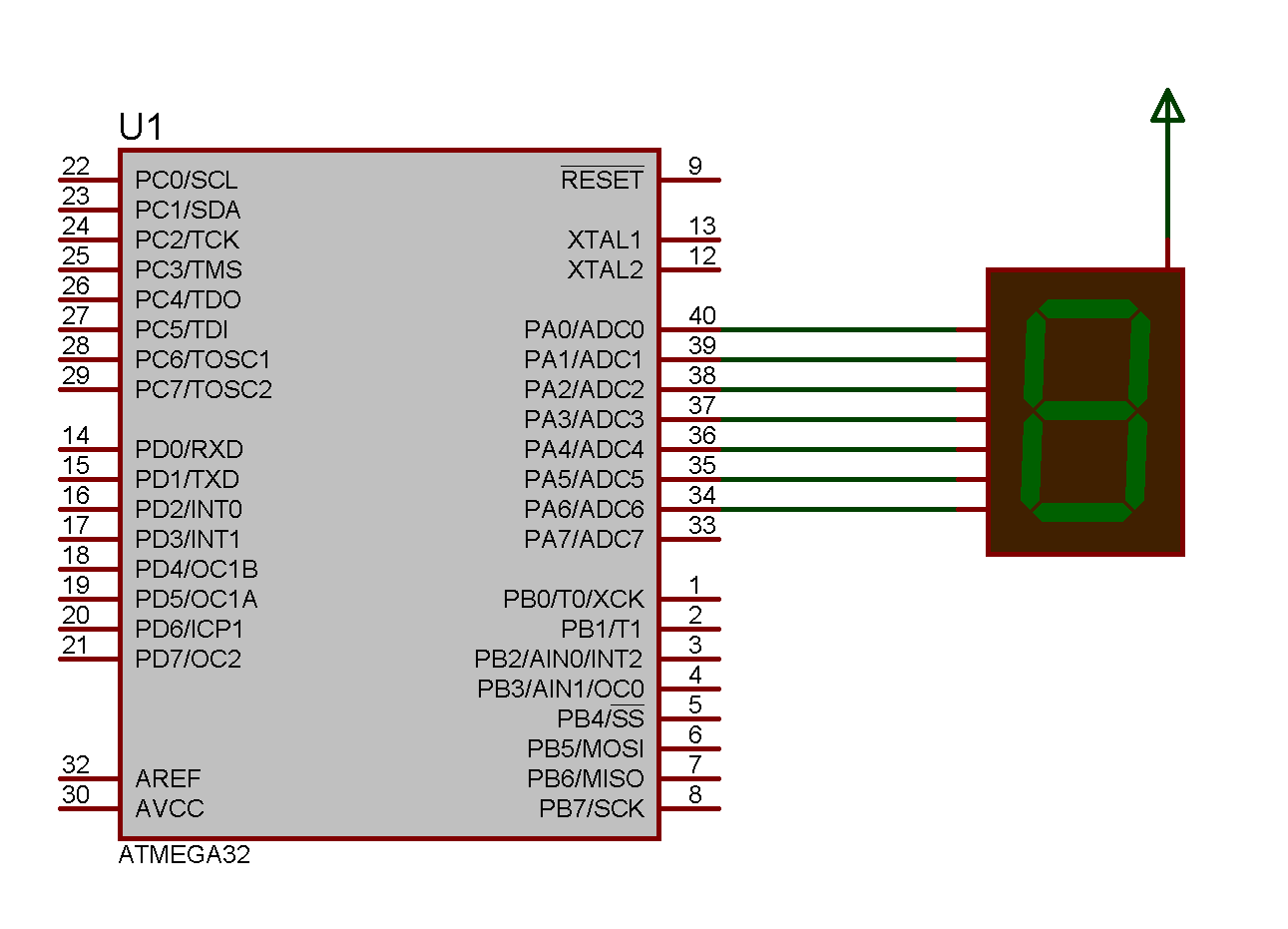
## PORTA = HexCodes[i] ;

## \_delay\_ms(1000) ;

## i--;

## if(i == -1) {

## i = 9 ;}}}

**Circuit :-**

1. **Dispplay hex up counter**

## Programm:-

#include <avr/io.h>

#include <avr/delay.h>

#include <util/delay.h>

#define F\_CPU 16000000UL

int main(void)

{

int HexCodes[16] = {0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x98,0x88,0x83,0xC6,0xA1,0x86,0x8E};

DDRA = 0xFF;

int i = 0 ;

while (i <= 15) {

PORTA = HexCodes[i] ;

\_delay\_ms(1000) ;

i++;

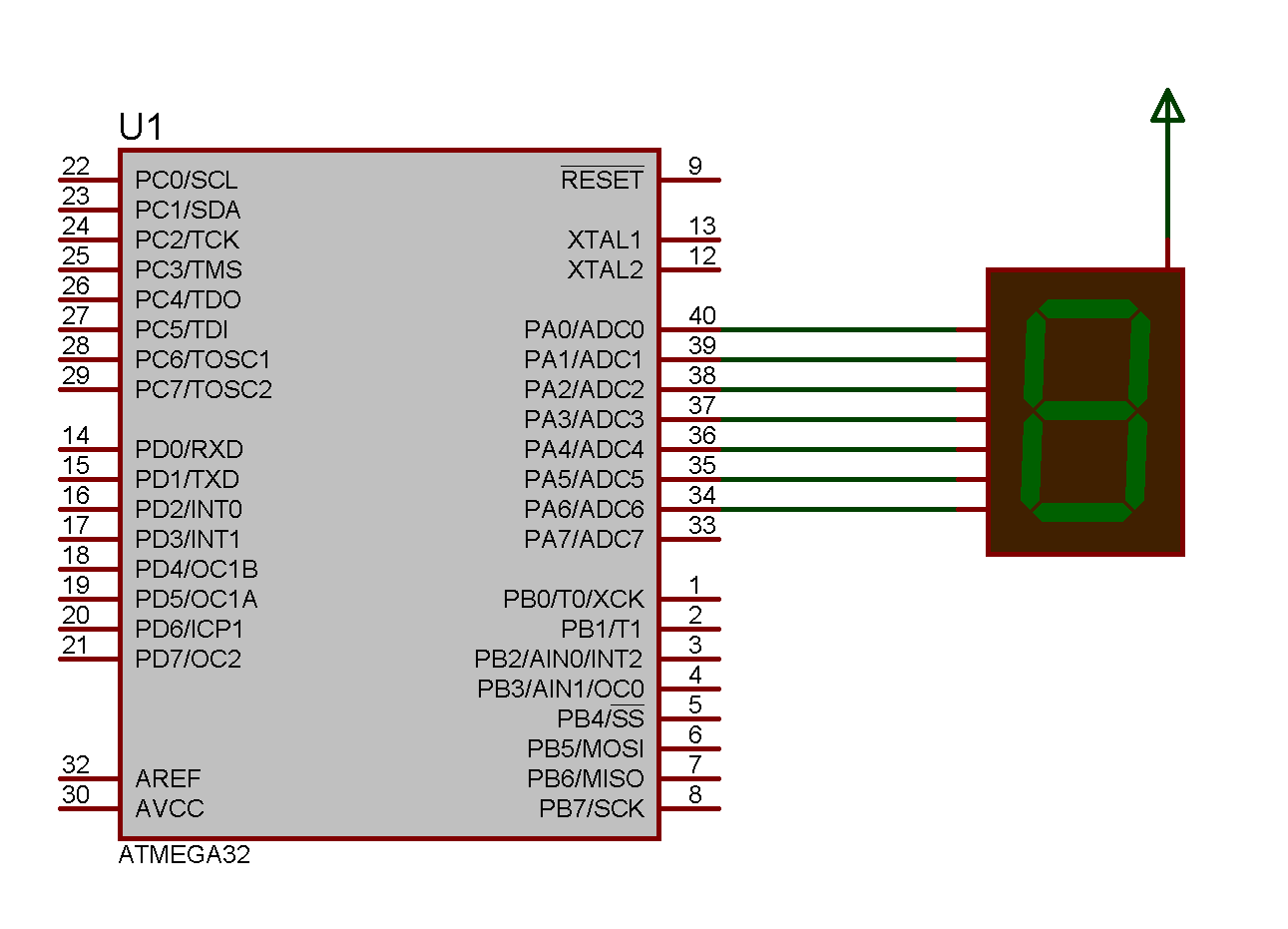
if(i == 16) {

i = 0 ;

}

}

}

**Circuit :-**

**4. Display hex down counter**

## Programm:-

## #include <avr/io.h>

## #include <avr/delay.h>

## #include <util/delay.h>

## #define F\_CPU 16000000UL

## int main(void)

## {

## int HexCodes[16] = {0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x98,0x88,0x83,0xC6,0xA1,0x86,0x8E};

## DDRA = 0xFF ;

## int i = 15 ;

## 

## while (i >= 0) {

## PORTA = HexCodes[i] ;

## \_delay\_ms(1000);

## i-- ;

## 

## if(i == -1) {

## i = 15 ;

## }

## 

## }

## 

## return 0;

## }

**Circuit :-**

**Objective:** Assume that a seven segment display is connected to PORTB and two switches are connected to PC0 and PC1. Perform the following tasks:

1. When both switches are off, it should display decimal up counter

2. When SW1 is pressed, and SW2 is off. display decimal down counter

3. When SW1 is off, and SW2 is pressed, display hex up counter

4. When both switches are pressed, display hex down counter

**Programm :-**

#include <avr/io.h>

#include <avr/delay.h>

#include <util/delay.h>

#define F\_CPU 16000000UL

int main(void)

{

int HexCodes[16] = {0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x98,0x88,0x83,0xC6,0xA1,0x86,0x8E};

DDRA = 0x00;

//PORTA = 0x03;

DDRD = 0xFF;

int i = 0 ;

while (1)

{

// Decimal Up

while (PINA == 0x00) {

if(i == 10) {

i = 0 ;

}

PORTD = HexCodes[i] ;

i++;

\_delay\_ms(500);

}

// Decimal Down

while (PINA == 0x01) {

if(i == -1) {

i = 9 ;

}

PORTD = HexCodes[i] ;

i--;

\_delay\_ms(500);

}

// Hex Up

while (PINA == 0x02) {

if(i == 16) {

i = 0 ;

}

PORTD = HexCodes[i] ;

i++;

\_delay\_ms(500);

}

// Hex Down

while (PINA == 0x03) {

if(i == -1) {

i = 15 ;

}

PORTD = HexCodes[i] ;

i--;

\_delay\_ms(500);

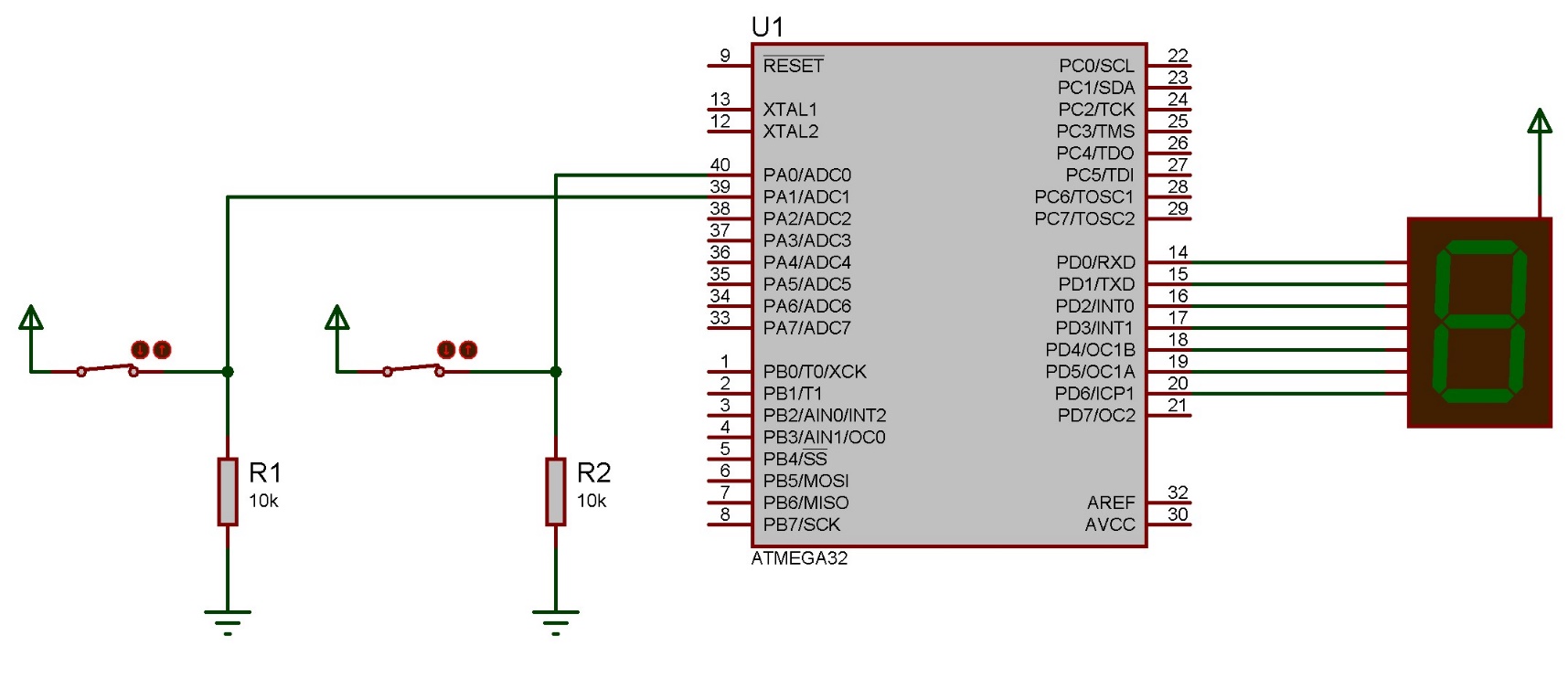
}

}

return 0;

}

**Circuit :-**



## Objective: Assume that two seven segment displays are connected to PORTA. Write a program to display 00 to 99.

**Programm:-**

#include <avr/io.h>

#include <avr/delay.h>

#include <util/delay.h>

#define F\_CPU 16000000UL

int main(void)

{

DDRA = 0xFF ;

DDRC = 0xFF ;

int HexCodes[10] = {0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x98};

while (1) {

PORTA = 0x00 ;

PORTC = 0x00 ;

int i = 0 ;

int j = 0 ;

while(i <= 9) {

j = 0 ;

while(j <= 9) {

PORTA = HexCodes[i];

PORTC = HexCodes[j];

\_delay\_ms(500);

j++;

}

i++ ;

if(i == 10) {

i = 0 ;

}

}

}

}

**Circuit :-**

