Circuit Diagram:

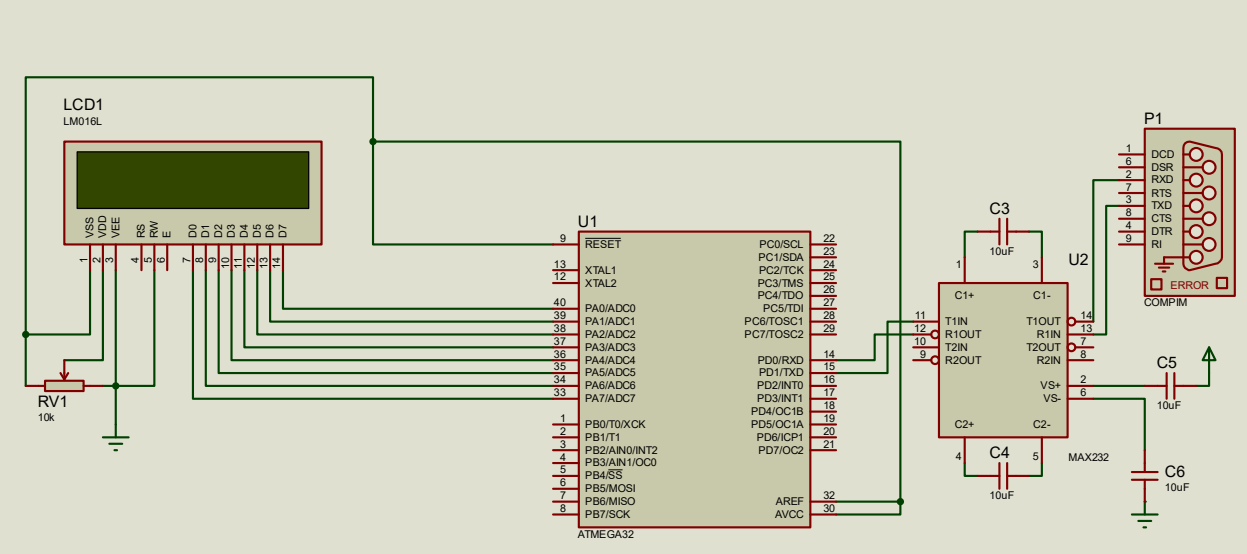

Figure 01: Circuit Diagram for serial data communication of ATmega32 with USART

Problem Statement:

Serial Data Communication using USART of ATmega32A with PC where:

Data transmission: Asynchronous; Baud Rate: 9600; 8 bit data; XTAL: 8 MHz

Code:

```
#include <avr/io.h>
#include <util/delay.h>
#include <stdlib.h>
#include "lcd.h"                                          // include LCD library

#define BAUD 9600                                         // define baud
#define BAUDRATE ((F_CPU)/(BAUD*16UL)-1) // set baudrate value for UBRR

#ifndef F_CPU
#define F_CPU 16000000UL                                  // set the CPU clock
#endif

// function to initialize UART
void uart_init (void)
{
      UBRRH=(BAUDRATE>>8);
      UBRRL=BAUDRATE;                                              //set baud rate
      UCSRB|=(1<<TXEN)|(1<<RXEN);                       //enable receiver and transmitter
      UCSRC|=(1<<URSEL)|(1<<UCSZ0)|(1<<UCSZ1);// 8bit data format
}

// function to send data - NOT REQUIRED FOR THIS PROGRAM IMPLEMENTATION
void uart_transmit (unsigned char data)
{
      while (!( UCSRA & (1<<UDRE)));                  // wait while register is free
```

```
	UDR = data;                                             // load data in the
register
}

// function to receive data
unsigned char uart_recieve (void)
{
	while(!(UCSRA) & (1<<RXC));                     // wait while data is being
received
	return UDR;                                             // return 8-bit data
}

// main function: entry point of program
int main (void)
{
	unsigned char a;
	char buffer[10];

	uart_init();                                            // initialize UART
	lcd_init(LCD_DISP_ON_CURSOR);                   // initialize LCD
	lcd_home();                                             // goto LCD Home

	while(1)
	{
		a=uart_recieve();                               // save the received data
in a variable
		itoa(a,buffer,10);                              // convert numerals into
string
		lcd_clrscr();                                           // LCD clear screen
		lcd_home();                                             // goto LCD home
		lcd_puts(buffer);                               // display the received
value on LCD
		_delay_ms(100);                                         // wait before next
attempt
	}

	return 0;
}
```

Circuit Diagram:

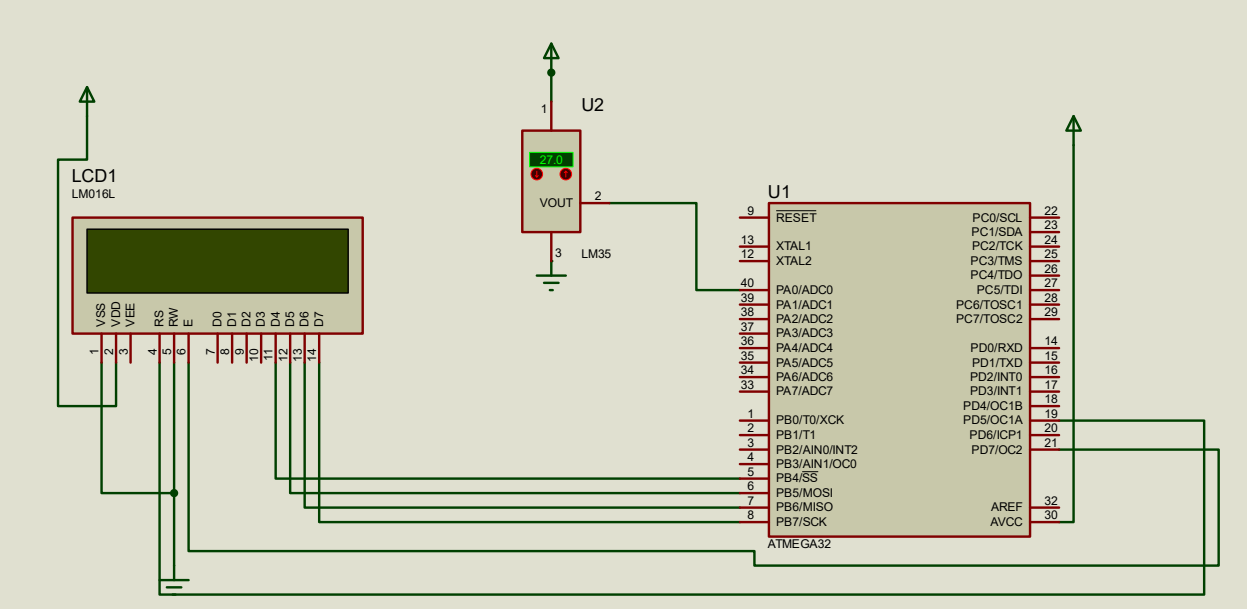

Figure 01: Circuit Diagram for Analog to Digital conversion in ATmega32

Problem Statement:

Convert LM35 sensor data from analog to digital and show the result in LCD display.

Code:

```
#include <avr/io.h>
#include <util/delay.h>

#include "lcd.h"

#define LTHRES 500
#define RTHRES 500

// initialize adc
void adc_init()
{
	// AREF = AVcc
	ADMUX = (1<<REFS0);

	// ADC Enable and prescaler of 128
	// 16000000/128 = 125000
	ADCSRA = (1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);
}

// read adc value
uint16_t adc_read(uint8_t ch)
{
	// select the corresponding channel 0~7
	// ANDing with '7' will always keep the value
	// of 'ch' between 0 and 7
	ch &= 0b00000111;  // AND operation with 7
```

```
        ADMUX = (ADMUX & 0xF8)|ch;     // clears the bottom 3 bits before ORing

        // start single conversion
        // write '1' to ADSC
        ADCSRA |= (1<<ADSC);

        // wait for conversion to complete
        // ADSC becomes '0' again
        // till then, run loop continuously
        while(ADCSRA & (1<<ADSC));

        return (ADC);
}

int main()
{
        uint16_t adc_result0, adc_result1;
        char int_buffer[10];
        DDRC = 0x01;           // to connect led to PC0
        DDRA = 0xoo; // input

        // initialize adc and lcd
        adc_init();
        lcd_init(LCD_DISP_ON_CURSOR);

        // display the labels on LCD
        lcd_puts("left  ADC = ");
        lcd_gotoxy(0,1);
        lcd_puts("right ADC = ");

        _delay_ms(50);

        while(1)
        {
                adc_result0 = adc_read(0);      // read adc value at PA0
                adc_result1 = adc_read(1);      // read adc value at PA1

                // condition for led to glow
                if (adc_result0 < LTHRES && adc_result1 < RTHRES)
                PORTC = 0x01;
                else
                PORTC = 0x00;

                // now display on lcd
                itoa(adc_result0, int_buffer, 10);
                lcd_gotoxy(12,0);
                lcd_puts(int_buffer);

                itoa(adc_result0, int_buffer, 10);
                lcd_gotoxy(12,1);
                lcd_puts(int_buffer);
                _delay_ms(50);
        }
}
```