Interfacing Resistance readout circuit through Atmega32 and 20x4 LCD

Internship Report

Submitted by

Parishmita Das

Guided by

Dr. Shaibal Mukherjee

DISCIPLINE OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

JANUARY 2020

CANDIDATE'S DECLARATION

We hereby declare that the internship report entitled **"Interfacing Resistance readout circuit through Atmega32 and 20x4 LCD "** submitted in partial fulfillment under the supervision of **Dr. Shaibal Mukherjee, Associate Professor in Electrical Engineering,** IIT Indore is an authentic work.

Further, I/we declare that I/we have not submitted this work for the award of any other degree elsewhere.

Signature and name of the student(s) with date

It is certified that the above statement made by the students is correct to the best of my/our knowledge.

Signature of Intern supervisor with date **Signature of HOD with date**

Introduction

Just as human beings acquire information about their environment through their senses and process such information using their brain, electronic systems perform such functions by means of sensors and processing digital devices such as microcontrollers (μC) or microprocessors (μP). Nowadays, such small but smart devices have become essential in many fields (industrial, automobiles, aircraft,medical devices, consumer electronics, home appliances, *etc*.) so that it is hard to imagine a society without them.Figure 1(a) shows the classical block diagram of a sensor electronic interface [1]. First of all, the sensor transforms a signal from a given energy domain (such as thermal, magnetic, mechanical,chemical or radiant) to the electrical domain by changing-for example-its electrical resistance orcapacitance. Afterwards, the signal conditioning circuit, which generally relies on operational amplifiers (OpAmp), performs some or all of the following tasks in the analogue domain: sensor output-to-voltage conversion, amplification, filtering, linearization and/or demodulation. The resulting analogue signal is then digitized via an analogue-to-digital converter (ADC). Finally, a digital system(e.g., a μC) acquires, stores, processes, controls, communicates (to other devices or systems) and/ or displays the digital value with information about the measured.

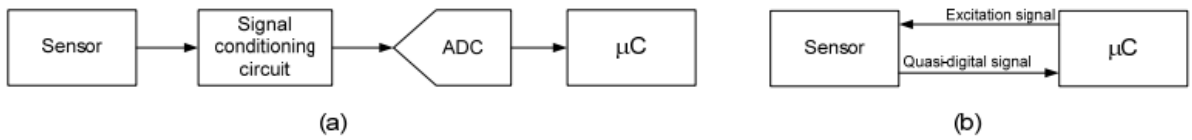


Figure 1: (a) classical sensor electronic interface, (b) direct interface circuit.

Hybrid Nanodevice Research Group (HNRG) Lab intended to produce portable,affordable sensor which is Sensitive,Selective,Stable(SSS) and is also able to fulfill the properties of repeatability and reproductability.The set-up in figure 2 is used to test the efficiency of any sensing device produced

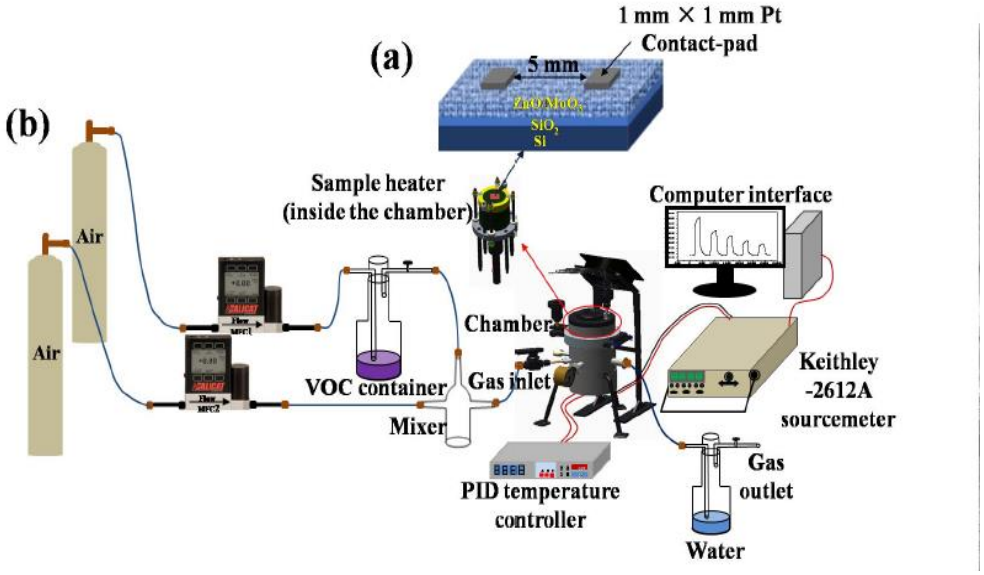
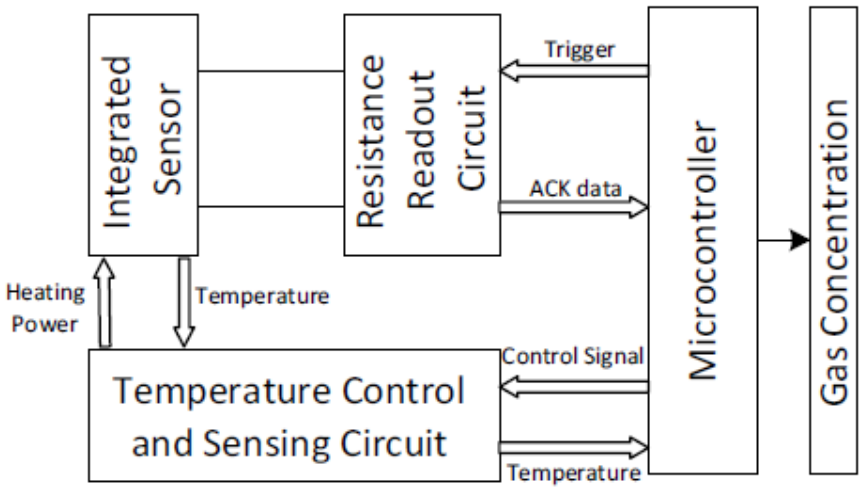


Figure 2:Schematic of (a) sensor device and (b)VOC sensing measurement set-up

This set-up is appropriate for uses in laboratories but not very appropriate for regular purposes as it is not affordable,portable,for regular purpose.To overcome this, HNRG is working on portable gas sensors . They are generally working with Metal Oxide Sensors owing to their capability to detect multiple gases with high sensitivity. The MOX based gas sensors behave electronically like sensing resistor of dynamic nature whose baseline value of resistance (sensitivity) could vary from a few kΩ to GΩ, depending on sensor layer properties, heater temperature and fabrication process. In presence of the gas, resistance value of sensing layer may increase or decrease more than a decade depending on gas present (oxidizing or reducing gases). Since MOX sensor shows better sensitivity at elevated temperature for a gas, various attempts have been made to optimize the micro-heater and its power requirements.



The sensor circuitry could be explained in brief from the figure 3:

Figure 3:Gas sensor system

The circuitry of the sensor consists of two parts, temperature controller and sensor readout circuit.This report mainly focuses on the latter and its interfacing with 20x4 LCD using atmega32.

Resistive Readout Circuit

Electronic system overview

The following circuit from figure 4 assures charging acknowledgement at one time constant corresponding to RC pair (RXC).

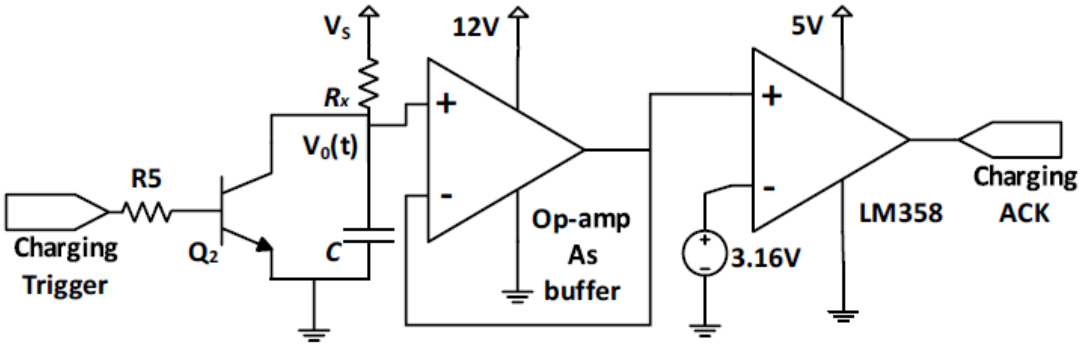


Figure 4

Operating Principle

For RC pair in figure 5 Charging equation is given as-

$$V_0(t) = V_S\left(1 - e^{-\frac{t}{RC}}\right)$$

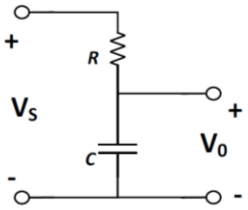


Figure 5:Operating Principle:Capacitor Charging

For t = 1 time constant (t = τ = RC),

$$V_0(\tau) = V_S\left(1 - e^{-1}\right) = 0.0632V_S$$

VS (=5 V) has been used as charging voltage. So,

$$V_0(\tau = 1) = 5 * 0.0632V = 3.162V$$

As in readout response, figure 6, 'Trigger charging' stays high, to discharge fixed capacitor, C (Using Q2, providing short circuit path to capacitor). At 'Trigger charging' (active low), depending on the value of RS, fixed capacitor C will start charging and voltage V(t) will start risingtowards VS. The acknowledgement 'charging ACK' occurs as V0 (t) reaches to aforementioned 3.162 V. So, time measured between 'trigger charging' and 'charging ACK' events will be time constant for sensor resistance (RS) and fixed capacitor (C = 0.01uF).

Time measurement is performed by MCU which starts timer module as 'trigger charging' asserts HIGH to LOW and measures till 'charging ACK' is HIGH. It can be seen from circuit and circuit response (Fig. 11) that 'trigger charging' is active low and acknowledgement flag is active high.

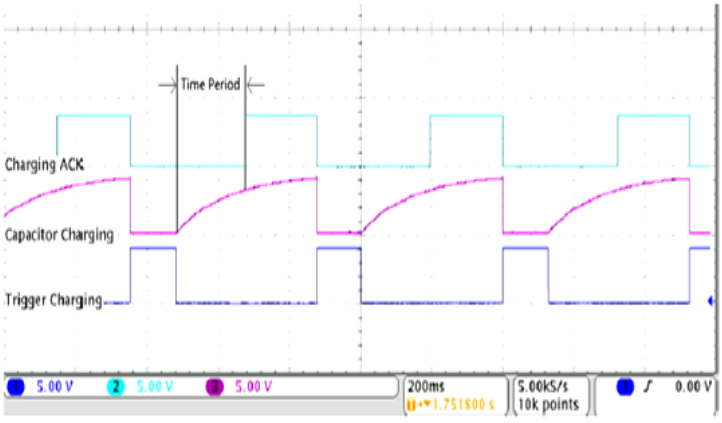


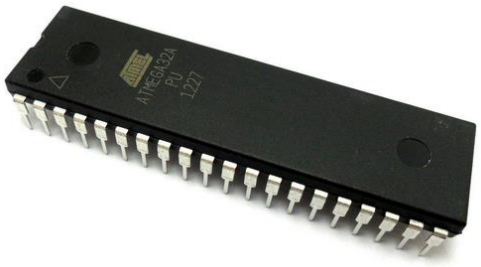
Figure 6: Readout circuit:Captured response on oscilloscope

LCD and Keyboard Interfacing

Components:

Atmega 32

Schematic:



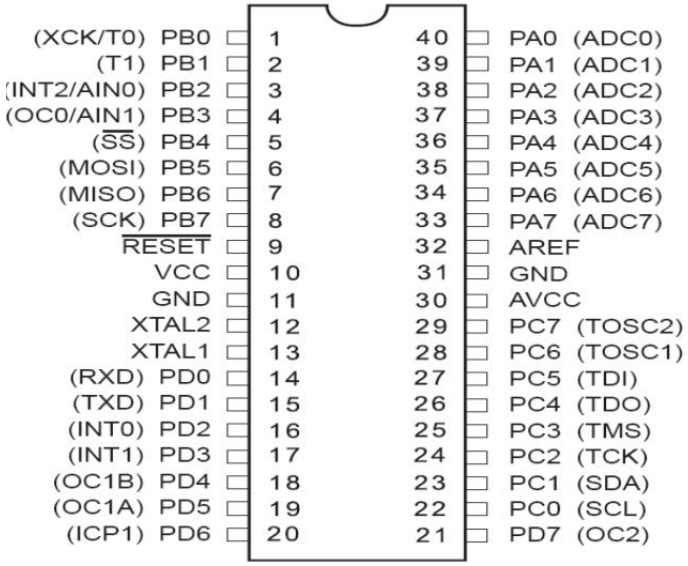
Specification:

The atmega32 is a high perfomance 8bit microcontroller, with ATMEL AVR MEGA architecture it has 32 progrmmable i/o pins and 131 powerfull instruction set.

It has:

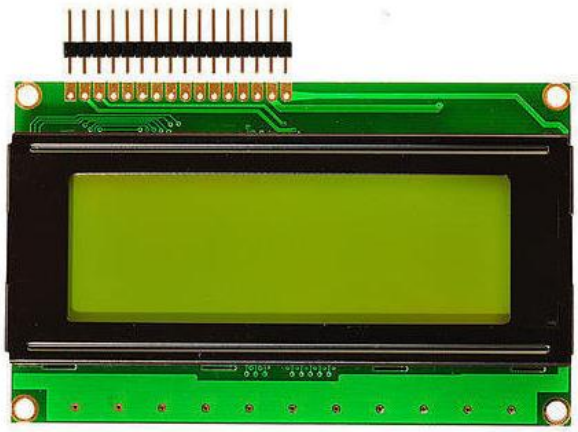
- 32 byte of In-System Self Programmable Flash
- 1024 Bytes of EEPROM
- 2K Byte of Internal SRAM
- Programming Lock for Software Security
- 32 Programmable I/O Lines
- 40-pin PDIP
- Operating Voltages:4.5 – 5.5V for ATmega32
- 0 – 16 MHz for ATmega32
- 8-channel, 10-bit ADC

Pin diagram:



20x4 LCD

Schematic:



Specification:

• Type: Character

• Display format: 20 x 4 characters

• Built-in controller: ST 7066 (or equivalent)

• Duty cycle: 1/16

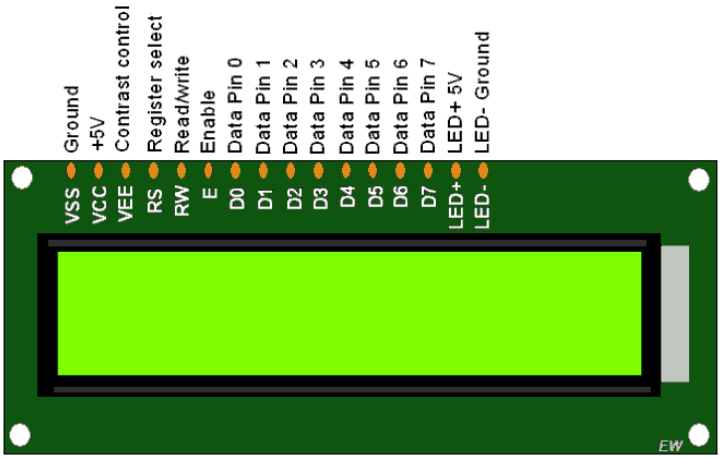
• 5 x 8 dots includes cursor

• + 5 V power supply (also available for + 3 V)

• LED can be driven by pin 1, pin 2, pin 15, pin 16 or A and K

• N.V. optional for + 3 V power supply

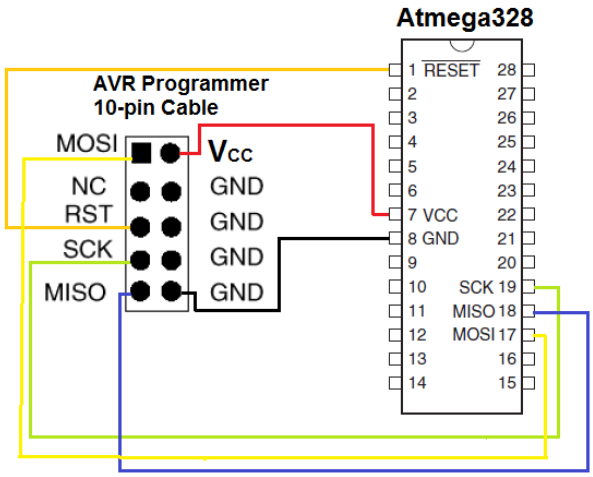
Pin diagram:



AVR Programmer

Schematic:

Pin diagram:

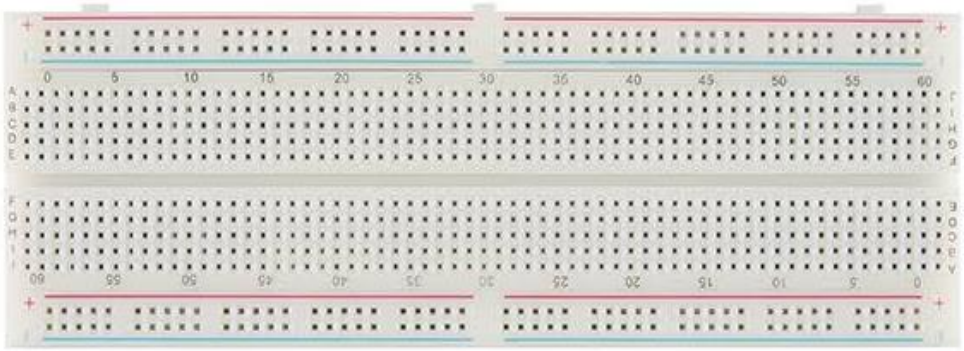


Specifications:

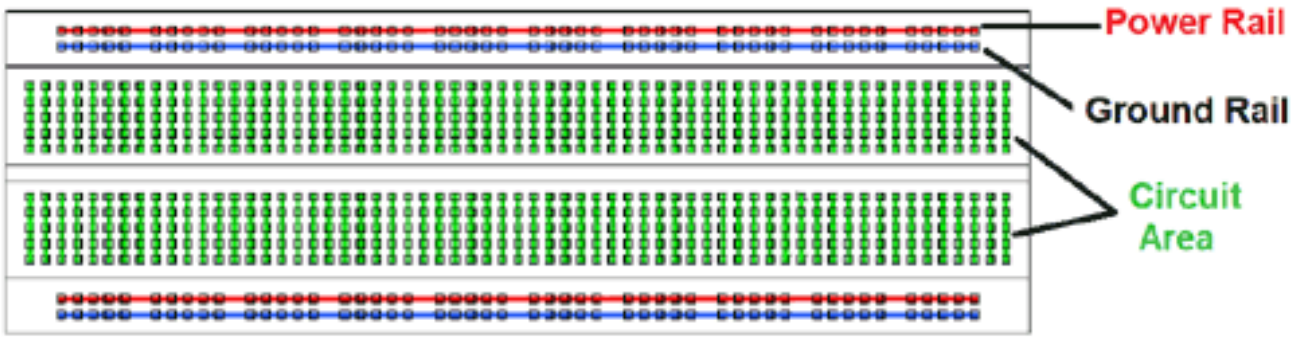
- USBasp is a USB in-circuit programmer for Atmel AVR controllers.
- It simply consists of an ATMega8 and a couple of passive components.
- The programmer uses a firmware-only USB driver, no special USB controller is needed, and thats make it low cost USB programmer.
- The Programmer works on USB port and can be used with Laptops.
- Software is compatible with Windows Vista and Windows 7.
- High speed programming and does't require external power.

Breadboard

Schematic:



The rows and columns of the breadboard are sorted as shown in th following figure

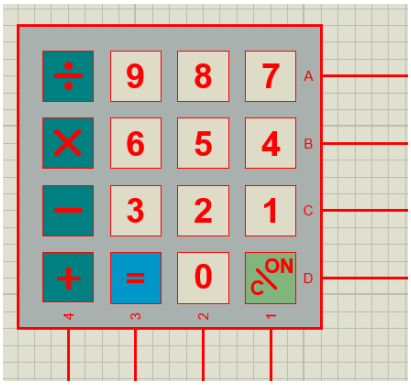


Specification:

- Most breadboards provide a grid of contacts where the spacing between contacts points is 1/10-inch square.
- Breadboards provide a varying number of contacts. Depending on the manufacturer, a breadboard could contain as few as 75 or as many as 900 separate connection points.
- Many breadboards are rated for five volts at one amp. A second common option provides a 15-volt, one-third amp rating.
- Most breadboards have a current limit of one amp or less, due to the nature of their contacts. Often breadboards can withstand only 1/3 amp.
- Most breadboards cannot withstand frequencies above 10 MHz.

Keyapad

Schematic:



LCD interfacing:

Code in AVR C Programming

LCD Library:

```
#define LCD
#ifndef
#include <avr/io.h>
#include <util/delay.h>
#include <stdlib.h>
#include <math.h>
#define Lightenable PIND2
#define Readwrite PIND1
#define Bipolarmood PIND0
char FirstColumnPositionofLCD[4]={0 , 64 , 20 , 84};
void InitializeLCD (void);
void Check_If_LCD_Is_Busy (void);
void Peek_A_Delay (void);
void Send_A_Command (unsigned char command);
void Send_A_Character (unsigned char character);
void Send_A_String (char *stringofcharacters);
void GotoLCDLocation (uint8_t x, uint8_t y);
void Send_A_StringtoLCDLocation(uint8_t x, uint8_t y, char *stringofcharacters);
 void InitializeLCD()
{
	DDRD |= (1<<Lightenable)|(1<<Readwrite)|(1<<Bipolarmood);
	_delay_ms(15);
	PORTD &=~(1<<Lightenable);
	Send_A_Command(0x01);
	_delay_ms(2);
	Send_A_Command(0x38);
	_delay_us(50);
	Send_A_Command(0b00001100);
	_delay_us(100);

 }
```

```
 void Check_If_LCD_Is_Busy()
 {

	DDRB = 0x00;
	PORTD |=(1<<Readwrite);
	PORTD &=~(1<<Bipolarmood);
	while (PORTB >= 0x80)
	{
		Peek_A_Delay();
	}
	DDRB=0xFF;
}
void Peek_A_Delay()
{
	PORTD |= (1<<Lightenable);
	asm volatile ("nop");
	asm volatile ("nop");
	PORTD &=~ (1<<Lightenable);
}
void Send_A_Command(unsigned char command)
{
	Check_If_LCD_Is_Busy();
	PORTB = command;
	PORTD &=~(1<<Bipolarmood);
	PORTD &=~(1<<Readwrite);
	Peek_A_Delay();
	PORTB=0;
}
void Send_A_Character(unsigned char character)
{

	Check_If_LCD_Is_Busy();
	PORTB = character;
	PORTD |= (1<<Bipolarmood);
	PORTD &=~ (1<<Readwrite);
	Peek_A_Delay();
	PORTB=0;
}
void Send_A_String(char *stringofcharacters)
{

	while(*stringofcharacters>0)
	{
	  Send_A_Character(*stringofcharacters++);
	}

}
void GotoLCDLocation(uint8_t x, uint8_t y)
{

	Send_A_Command(0x80 + FirstColumnPositionofLCD[y-1] + (x-1));
}
void Send_A_StringtoLCDLocation(uint8_t x, uint8_t y, char *stringofcharacters)
{
	GotoLCDLocation(x, y);
	Send_A_String(stringofcharacters);
}
#endif
```

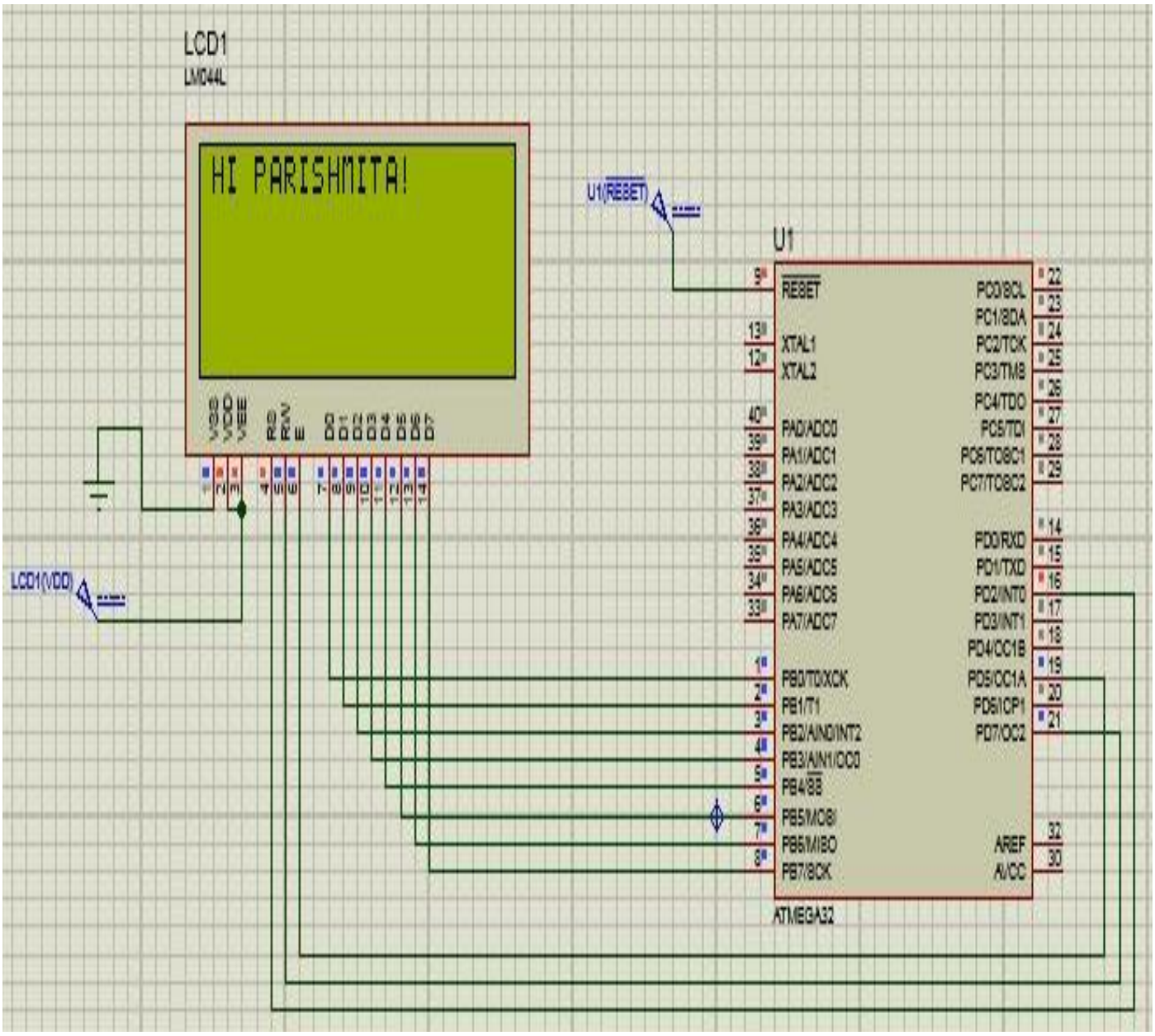
Main Code

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

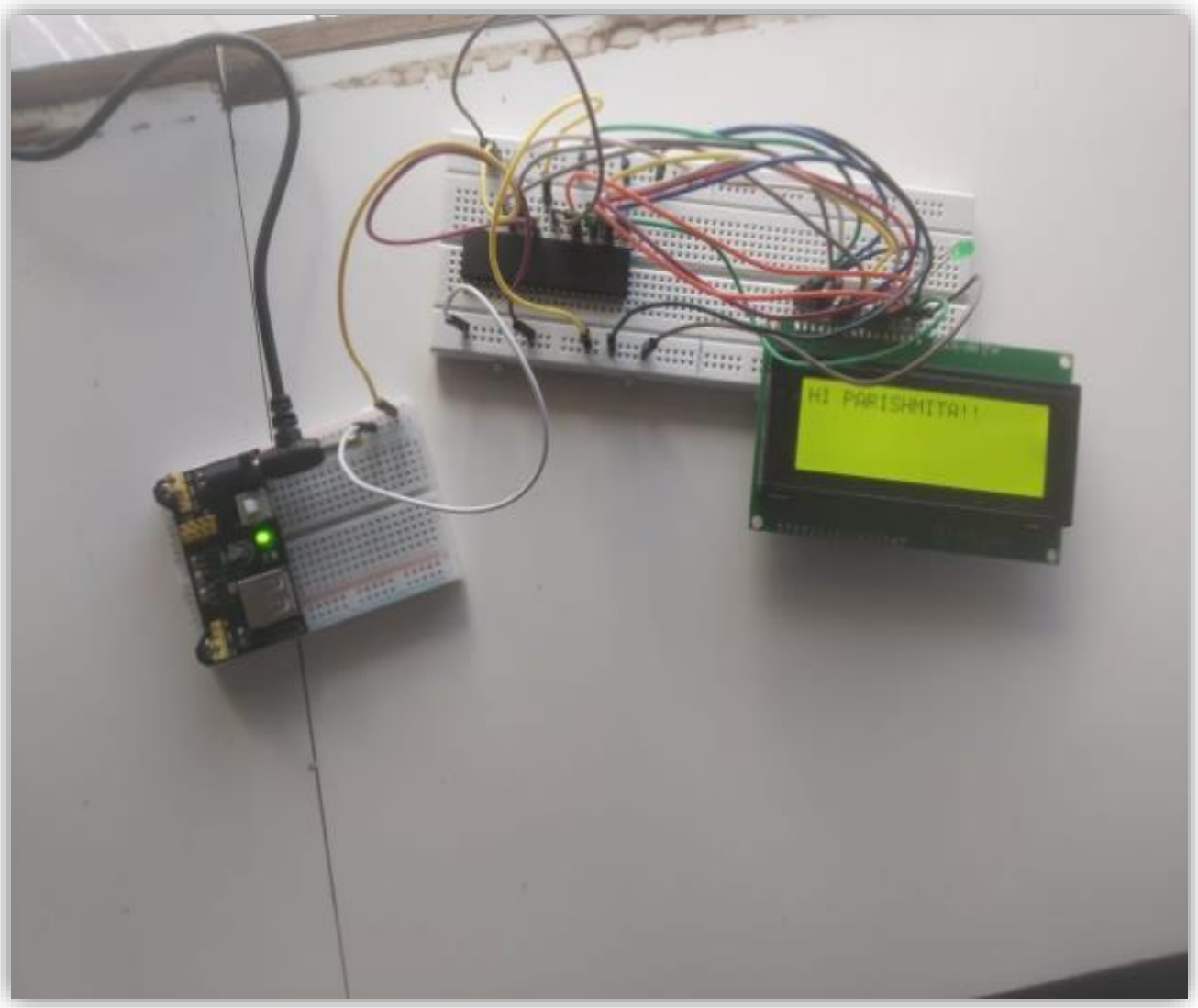
```
int main()
{
	InitializeLCD();
	Send_A_String("HI PARISHMITA!");
	while(1)
	{
	}

}
```

Simulation in Proteus

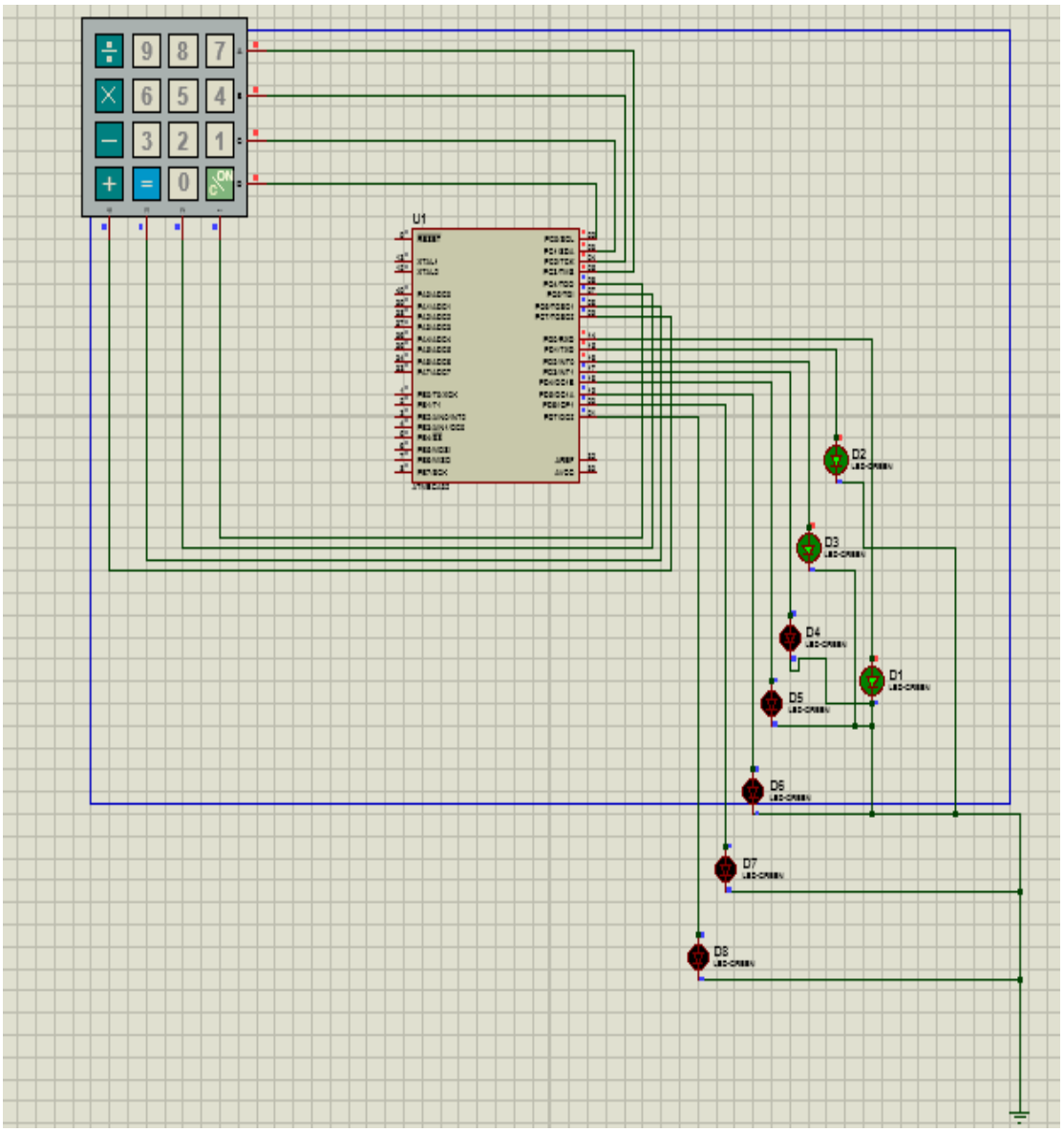


Final Circuitry



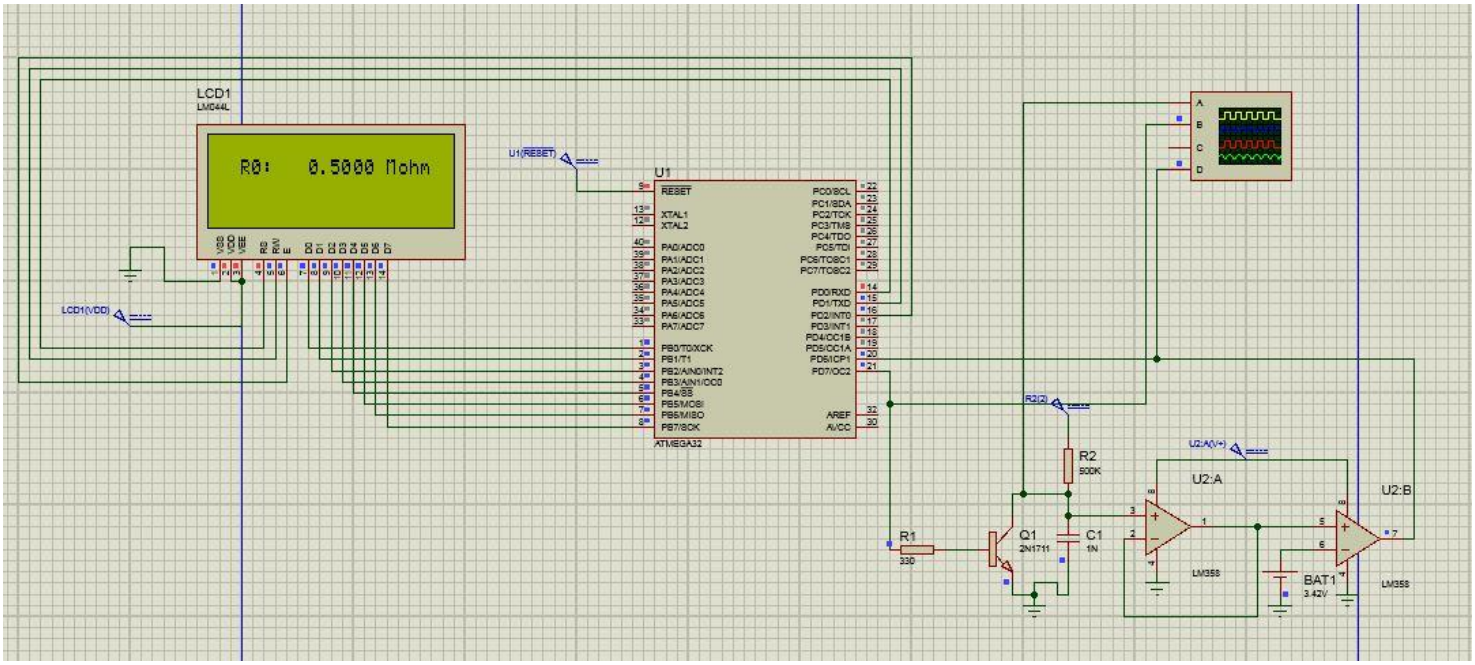
Keyboard Interfacing:

Proteus Circuit Simulation

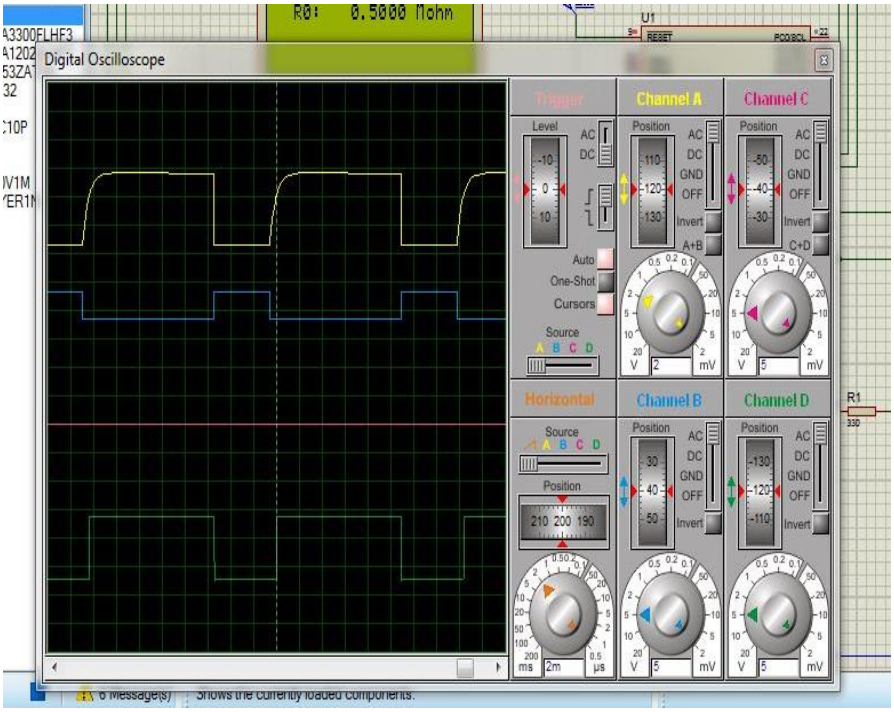


Resistance Readout Circuit Interfacing

Readout Circuit



Readout Circuit: Captured Response on Oscilloscope



References:

1. Pallàs-Areny, R.; Webster, J.G. *Sensors and Signal Conditioning*, 2nd ed.; John Wiley & Sons:New York, NY, USA, 2001.

2. Huising, J.H. Smart sensor systems: Why? Where? How? In *Smart Sensor Systems*; Meijer, G.C.M., Ed.; Wiley: Chichester, UK, 2008; pp. 1–21.

3. Reverter, F.; Pallàs-Areny, R. *Direct Sensor-to-Microcontroller Interface Circuits. Design and Characterization*; Marcombo: Barcelona, Spain, 2005.

4. Cox, D. *Implementing Ohmmeter/Temperature Sensor*; Microchip Technology AN512: Chandler,AZ, USA, 1994.

5. Bierl, L. *Precise Measurements with the MSP430*; Texas Instruments: Dallas, TX, USA, 1996.

6. Richey, R. *Resistance and Capacitance Meter Using a PIC16C622*; Microchip Technology AN611: Chandler, AZ, USA, 1997.

7. Grassi LM, Malcovati P, Capone S, Francioso L, Siciliano P, Baschirotto A (2009) A CMOS integrated interface circuit for metaloxide gas sensors. Sens Actuators B Chem 142(1):82–89, ISSN0925-4005