EMBEDDED SYSTEM

PROJECT REPORT

Submitted To:

Dr. Asim Rehmet

Submitted By:

2019-CE-07 (Aqsa Ayaz)
2019-CE-09 (Amna Jamshaid)
2019-CE-19 (Esha Sajid)

Acknowledgement

We are glad that we are able to complete this project and understand many things. Through preparation of this embedded system project was an immense learning experience. This is to acknowledge all those without whom this project would not have been reality. Firstly, we would like to express our deepest thanks to our Embedded system teacher Dr. Asim Rehmet who gave his immense support, made us understand how to make this project and dedicated his time towards it. This project would not have been complete without his guidance.

We would also like to thanks Ma'am Sana for her guidance and for providing necessary information regarding the project. I am also very thankful to my group members for their co-operation.

Table of Contents

ABSTRACT:

This project is focused on the design and development of a microcontroller-based calculator with a keypad and a 16x2 Liquid Crystal Display that does elementary arithmetic (LCD). The software only accepts single-digit input and produces double-digit output. This enables for a basic arithmetic programme, yet the same ideas may also be used to multi-digit computations.On single digit integers, this calculator can do the following four functions: addition, subtraction, multiplication, and division (+,-,/,*). A 4x4 keypad is used to enter the numbers. On the 16x2 lcd, the result is shown.

Introduction:

The Pic 16f877A microcontroller calculator is a massive project. Code is comprehensive in the sense that it is long and logical. It covers every part of programming and connecting a 16x2 lcd and a 4x4 keypad with an image. Because of technological advancements, electronic devices with basic circuits may now be built. The introduction of the microcontroller has simplified the design of electrical devices circuits. A microcontroller is a computer on a chip. Mobile phones, video cameras, electrical appliances, and the majority of self-contained electronic equipment all require it to function. Memory, central processing unit, ports, bus, serial connection, and other components make up a microcontroller.

Memory stores both the programme and the data.The Central Processing Unit (CPU) is capable of multiplying, dividing, subtracting, and moving data from registers in memory. Registers are memory locations that are used to do numerous mathematical calculations or any other actions using data that may be located everywhere. A bus is a link between memory and the CPU that allows data to flow from one block to another. It physically depicts a collection of eight, sixteen, or more wires. Address and data buses are the two sorts of buses. The first contains as many lines as the amount of memory to address, whereas the second has the same width as the data.

The first sends the address from the CPU memory, while the second links all of the microcontroller's blocks.

Ports have numerous memory locations, one of which is connected to the data bus and the other to the microcontroller's output lines, which are visible as pins on the electrical component. Input, output, and bidirectional ports are the three types of ports. When working with ports, the first step is to determine which port needs to be used, and then to send or receive data from that port.

When using it, the port functions as a memory location. Something is being written into or read from it, which can be seen on the microcontroller's pins.

A watchdog counter is a free-run counter in which a programme must write a zero every time it runs correctly. The analogue to digital converter (ADC) is in charge of transforming information about an analogue value to a binary number and passing it on to a CPU block for further processing. Finally, the microcontroller is finished, and all that remains is to put it together into an electrical component that it can access.

Problem Statement:

A calculator's goal is to do accurate and efficient computations. It is obvious that, to the greatest extent feasible, a calculator should relieve the user of the need to do mental operations and rely on paper. It permits us to dedicate more time to studying the problem by breaking it down into simpler tasks. Moreover, they are liberated from tedious computations and the same tedious everyday practice.

Specifications of proposed System:

It is specified to perform basic arithmetic operations such as add, subtract, multiply and divide. It is designed to digitalize the basic operators used in everyday life in order to make our work efficient and accurate. It has paced up the daily calculations.

Constraints:

Project Deadlines:

TASK	DEADLINE
Project Proposal	5/10/2022
Project	5/31/2022

Project Budget:

Components	Price in Rupees
Pic16f877 Microcontroller	950
16×2 lcd	349
4×4 keypad	140
Crystal 20 MHz	75
Potentiometer/variable resistor	120
Bread board or PCB for Circuit Designing	60
Battery	180

Number of People:

A group of three people collaborated in doing this task, the members are as follow:

Aqsa Ayaz (2019-CE-07)

Amna Jamshaid (2019-CE-09)

Esha Sajid (2019-CE-19)

Technical Knowledge:

A calculator is a device that performs numerical arithmetic operations. The calculators can perform addition, subtraction, multiplication, and division. Exponential operations, roots, logarithms, trigonometric functions, and hyperbolic functions can all be handled by more advanced calculators. This technology enables us to address complex issues fast and efficiently. Furthermore, it can simplify the situation and allow us to dedicate more time to comprehending the problem. Second, they are liberated from tedious computations and the same tedious everyday practice.

Software Tools:

The following software tools are utilized in programing PIC16F877A:

- PIC C Compiler
- MPLAB
- C++ Language

Working:

When you turn the system on. The 16*2 LCD first row will display the message "Enter First No = ". You are the one to input the number. The number will appear on the 16*2 LCD screen as soon as you input it. Then, on the first row of the 16*2 LCD, a second message will display, requesting you to "Enter Operator = ". The operator has been input. Then a third message will appear, instructing you to "Enter Second No= ". You typed in the second digit. There are 16 buttons on the keypad: ten numeric buttons, four arithmetic operations, equals, and clear. When you've finished typing the second number, on a 16*2 LCD, the acquired answers are shown on the LCD display.

Inshort, the calculator operates as follows:
• Press a number key, then an operation key, then another number, and finally equals to conduct a computation.
• The result of the computation is presented. The outcome of the divide operation is shown as result and remaining.

The calculation procedure selects the appropriate process using the operation input code: add, subtract, multiply, or divide. The calculation's binary result is supplied to a function that converts it to BCD, then ASCII, before sending it to the display. The division result, which is a single digit result with a remainder, is transmitted directly to the display.

Component Requirements:

- Pic16f877 Microcontroller
- 16×2 lcd
- 4×4 keypad
- Crystal 20 MHz
- Potentiometer/variable resistor (For setting Lcd Contrast)
- Bread board or PCB for Circuit Designing
- Power Supply

16×2 LCD:

An LCD (Liquid Crystal Display) screen is a type of electronic display that may be used in a variety of ways. A 16x2 LCD display is a fairly simple module that may be found in a variety of devices and circuits. A 16x2 LCD can display 16 characters per line on each of its two lines. Each character is presented in a 5x7 pixel matrix on this LCD. The 224 distinct letters and symbols may be shown on the 16 x 2 intelligent alphanumeric dot matrix display. Command and Data are the two registers on this LCD. Command register stores various commands given to the display. Data register stores data to be displayed.

Potentiometer:

The potentiometer is a device that compares an unknown voltage to a known voltage to determine its value. It may be used to calculate the emf and internal resistance of a specific cell, as well as to compare the emf of other cells. The potentiometer employs the comparative technique.

A potentiometer is a three-terminal variable resistor that may be adjusted manually. Two of the terminals are attached to the opposing ends of a resistive element, while the third terminal is connected to a wiper that moves across the resistive element. In essence, the potentiometer is a variable resistance divider. A potentiometer is also commonly known as a **potmeter** or **pot**.

Design Overview:

The design overview is divided into two, hardware and software.

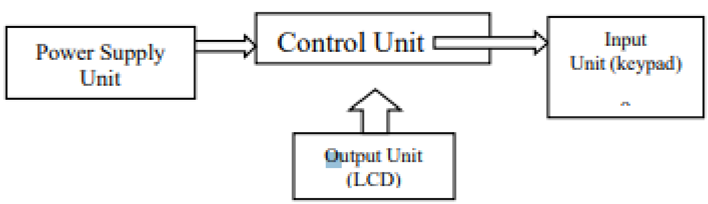
Hardware design:

For easier design, analysis, and integration, the Hardware Design is divided into modules.
- Unit for power supply
- Control Unit
- Input Unit
- Output Unit

Each module was created separately and then assembled to build a calculator. A succession of C-programs linked into the control unit control the various modules.

Power Supply Unit:

The power supply unit supplies the electronic circuit with the required direct voltages and currents, as well as low A.C ripples and high stability. A transformer, a rectifier, a filter, and a regulator make up the power supply.



Input Unit:

The 4 x 4 keypad utilised in this design acts as an input device for the user to connect with the system.

Output Unit (LCD):

The LCD is the output device. A 16 x 2 LCD is used in the calculator. It contains a back-groundlight that makes it easier to read in low-light situations.

Control Unit:

The calculator's CPU is the microcontroller. It offers commands, receives keypad inputs, processes them, and displays the processed result on the LCD.

Software Design:

The programme overview and flow diagram in the software design illustrate how the system operates. It depicts the flow of instructions from one step to the next throughout operation in great detail.

Program outline for implementing the calculator:

The program outline used to implement the calculator is shown below:

1. Single digit calculator produces two digit results.

2. Hardware: x12 keypad, 2x16 LCD, P16F887 MCU

3. MAIN

4. Initialise

5. PortC = keypad

6. RC0 – RC3 = output rows

7. RC4 – RC7 = input columns

8. PortD = LCD

9. RD1, RD2 = control bits

10. RD4– RD7 = data bits

11. CALL Initialise display

12. Scan Keypad

13. REPEAT

14. CALL Keypad input, Delay 50ms for debounce

15. CALL Keypad input, Check key released

16. IF first key, load Num1, Display character and restart loop

17. IF second key, load sign, Display character and restart loop

18. IF third key, load Num2 Display character and restart loop

19. IF fourth key, CALL Calculate result

20. IF fifth key, Clear display
21. ALWAYS
22. SUBROUTINES
23. Included LCD driver routines
24. Initialise display
25. Display character
26. Keypad Input
27. Check row A, IF key pressed, load ASCII code
28. Check row B, IF key pressed, load ASCII code
29. Check row C, IF key pressed, load ASCII code
30. Check row D, IF key pressed, load ASCII code
31. ELSE load zero code
32. Calculate result
33. IF key = ‘+’, Add
34. IF key = ‘-‘, Subtract
35. IF key = ‘x’, Multiply
36. IF key = ‘/’, Divide
37. Add Add Num1 + Num2
38. Load result, CALL Two digits
39. Subtract Subtract Num1 – Num2
40. IF result negative, load minus sign, CALL Display character
41. Load result, CALL Display character
42. Multiply
43. REPEAT

44. Add Num1 to Result

45. Decrement Num2

46. UNTIL Num2= 0

47. Load result, CALL Two digits

48. Divide

49. REPEAT

50. Subtract Num2 from Num1

51. Increment Result

52. UNTIL Num1 negative

53. Add Num2 back onto Num1 for Remainder

54. Load Result, CALL Display character

55. Load Remainder, CALL Display character

56. Two digits

57. Divide result by 10, load MSD, CALL Display character

58. Load LCD, CALL Display character

The keypad and LCD are initially initialised in the main code. The code then waits for the first number from the keypad. The LCD screen is cleared after receiving this number. This number is also shown on the LCD. The code then waits for the user to press the function key.The code waits for the second number and then the equal sign after receiving the function key. After receiving the equal sign, the result is computed and presented on the screen using the desired function.

Functions included in the code for the pic microcontroller calculator are:

- **void main() :**

The heart of the code is the void main() function. The main function is called first. It is followed by all other functions.

- **void lcdcmd (unsigned char):**

This function transmits commands and manages the lcds registers in order to correctly execute them.

- **void lcddata (unsigned char):**

This function delivers data to the lcd display and controls the registers on the lcd so that data may be shown on the 16*2 lcd.

- **void disp-num (float num):**

This function effectively shows the computed value or output result on a 16*2 lcd display.

- **int get number (char ch):**

This function changes the value of a character to an integer. In order to show an integer value on a 16*2 lcd, the value must first be translated to character format. A character 8-bit value may be shown on a 16*2 lcd.

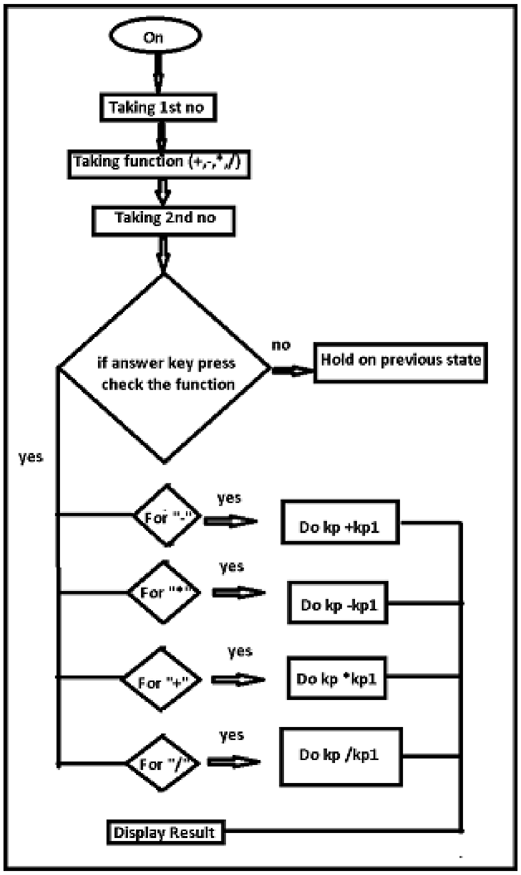
- **void lcdinit ():**

The character lcd display is initialised with this function. 16*2 lcd, 75% font size, flickering cursor, etc.

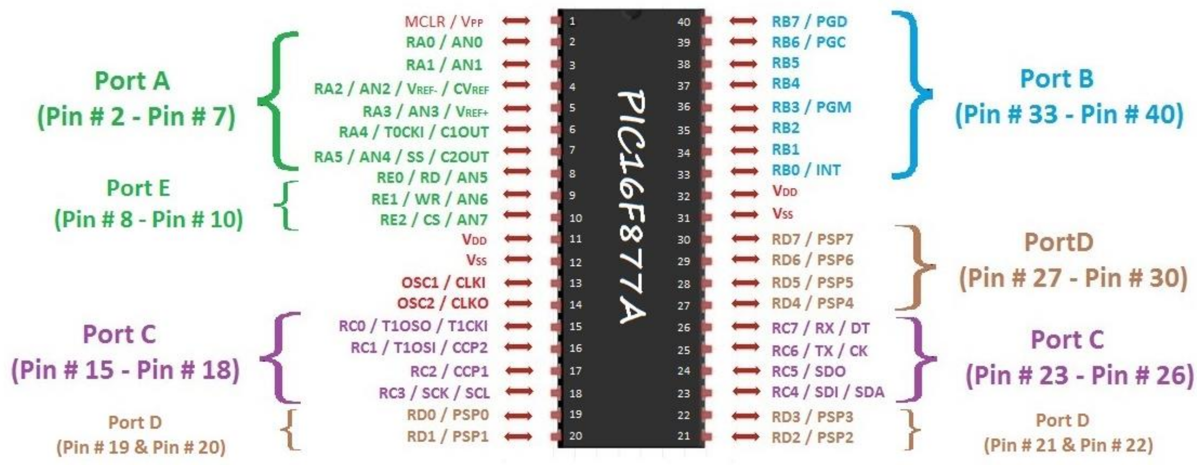
- **char scan_key(void):**

This function determines which keypad key the user has pushed.

Flow Diagram:



Datasheet of PIC16F877A:



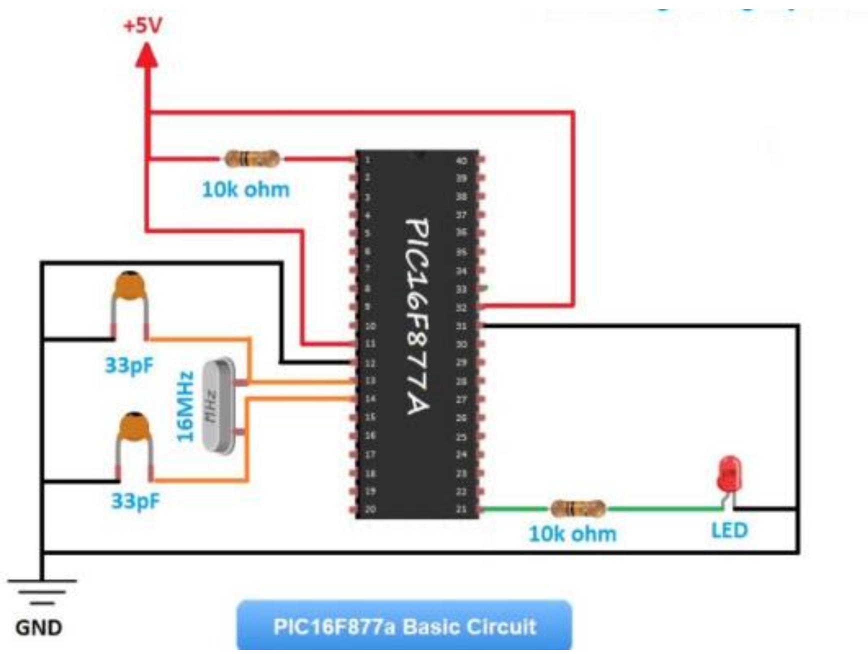
Introduction to PIC16F877A:

- PIC16F877A is the most commonly used PIC Microcontroller because of its operational flexibility, availability and low cost.

- PIC16F877a is a 40-pin PIC Microcontroller by Microchip that uses RISC architecture and is utilised in embedded projects.
- It contains five ports, starting with Port A and ending with Port E.
- It has three timers, two of which are 8-bit and one of which is 16-bit.
- It is compatible with a variety of communication protocols, including the Serial Protocol, Protocol in parallel and Protocol I2C.
- Hardware pin interrupts and timer interrupts are also supported.

Basic circuit of PIC16F877A:

- Each microcontroller has a fundamental circuit that must be designed in order for it to function.
- It's the same as powering your PIC Microcontroller, and it runs on a +5V supply.
- What will you do if you want to switch on the fan? You'll just supply it with power, and that's exactly what we'll do with a PIC.
- However, in this situation, in addition to providing electricity, we must also offer the frequency at which it will function.
- So, now we know that we must construct the fundamental circuit, which includes both power and the frequency at which it will function.
- We utilise a crystal oscillator to supply frequency to the PIC Microcontroller, and for the PIC16F877a, we use a crystal oscillator.



- **Pin # 1:** This Pin is called **MCLR (Master Clear)** and we need to provide 5V to this pin through a 10k-ohm resistance.
- **Pin # 11 & Pin # 32:** These Pins are labeled as **Vdd** so we also need to provide it +5V and you can see these lines are in red color in above figure.

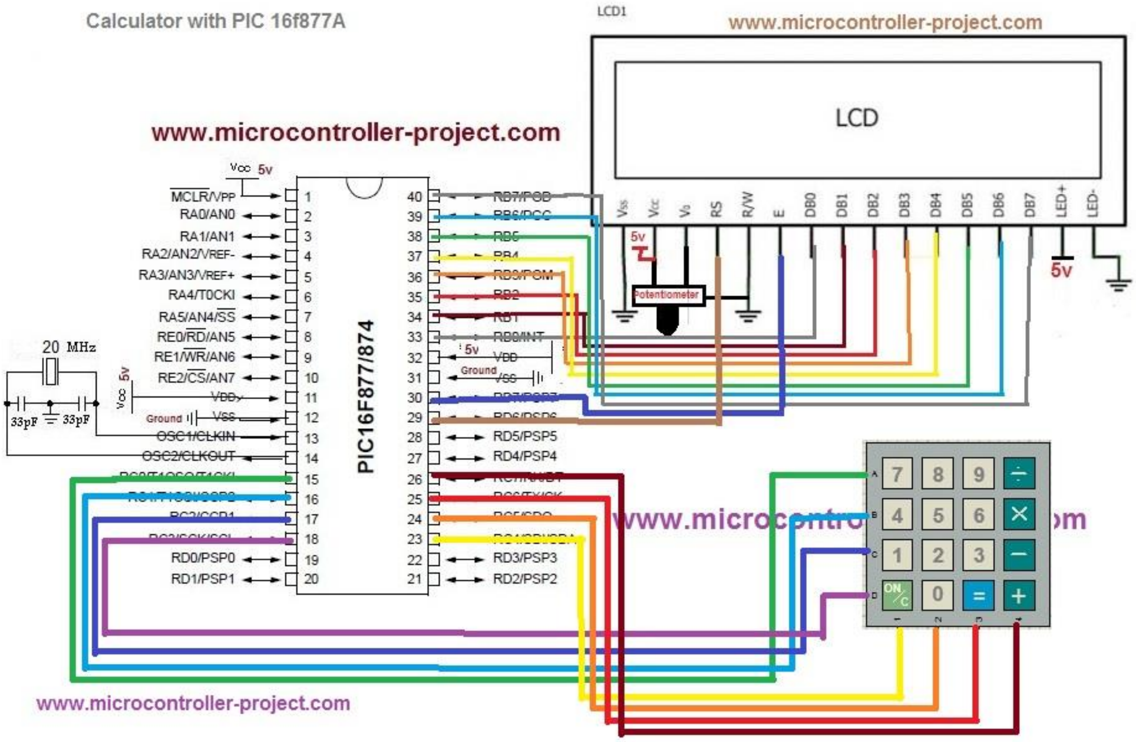
- **Pin # 12 & Pin # 31:** These Pins are **Vss**, so we have provided **GND (Ground)** at this pin and its lines are in black color.
- **Pin # 13 & 14:** These Pins are named **OSC1 (Oscillator 1)** and **OSC2 (Oscillator 2),** now we have to attach our **Crystal Oscillator (16MHz)** at these pins which I have lined in Orange color. After the Crystal Oscillator, we have 33pF capacitors and then they are grounded.

Ports of PIC16F877A:

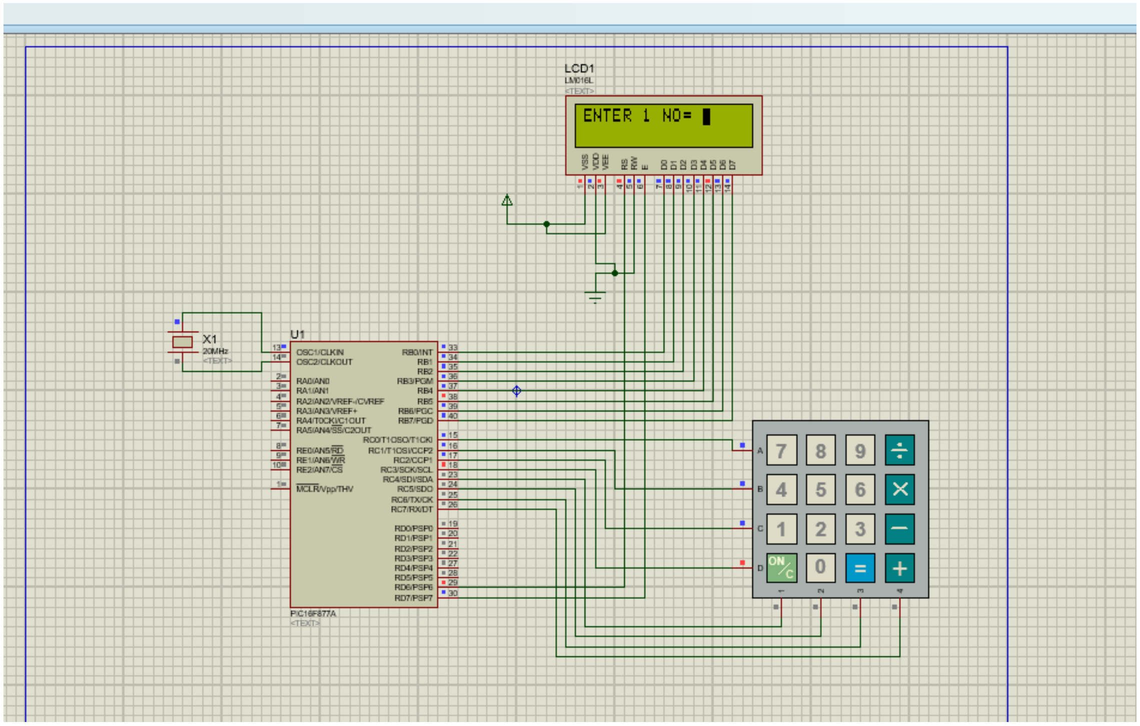
The PIC16F877a has five ports in total:

- Port A: It contains a total of 6 pins, numbered 2 through 7. The Port A Pins are labelled from RA0 to RA5, with RA0 being the first Pin's label.
- Port B contains a total of 8 pins, ranging from Pin # 33 to Pin # 40. The Port B Pins are branded from RB0 to RB7, with RB0 being the first Pin's label.
- Port C has a total of 8 pins. Its pins aren't lined up properly. Pin # 15 – Pin # 18 are the first four pins of Port C, whereas Pin # 23 – Pin # 26 are the last four.
- Port D has a total of 8 pins. Its pins are likewise out of alignment. Pin # 19 – Pin # 22 are the first four pins of Port D, whereas Pin # 27 – Pin # 30 are the last four pins.
- Port E features a total of three pins, ranging from Pin # 8 to Pin # 10. RE0 is the label of the first Pin of Port E, while RE2 is the label of the second Pin of Port E.

Schematic diagram:

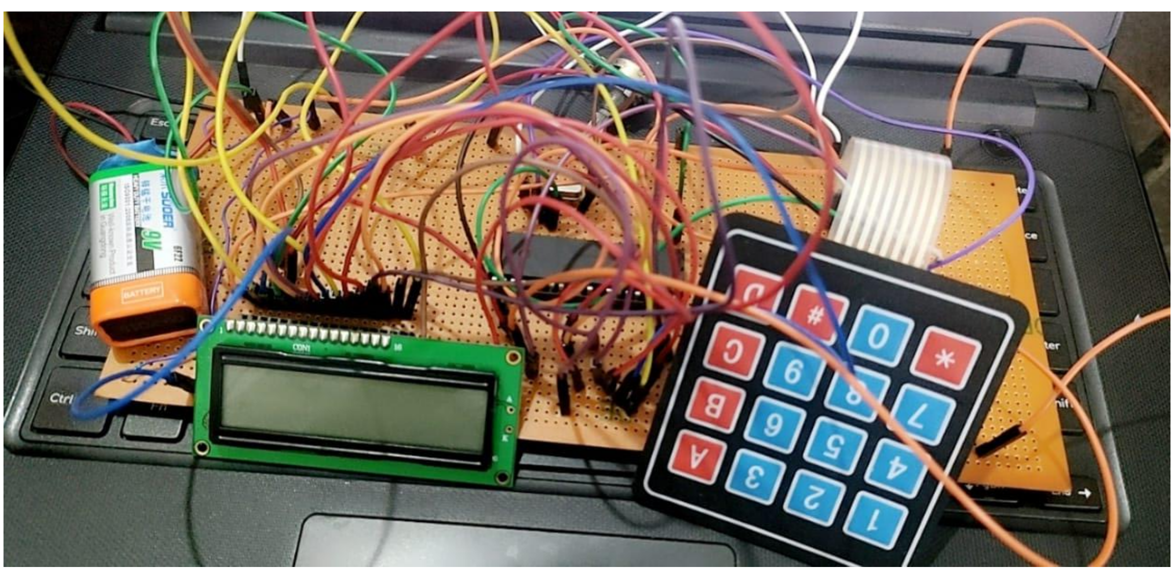


Simulation:



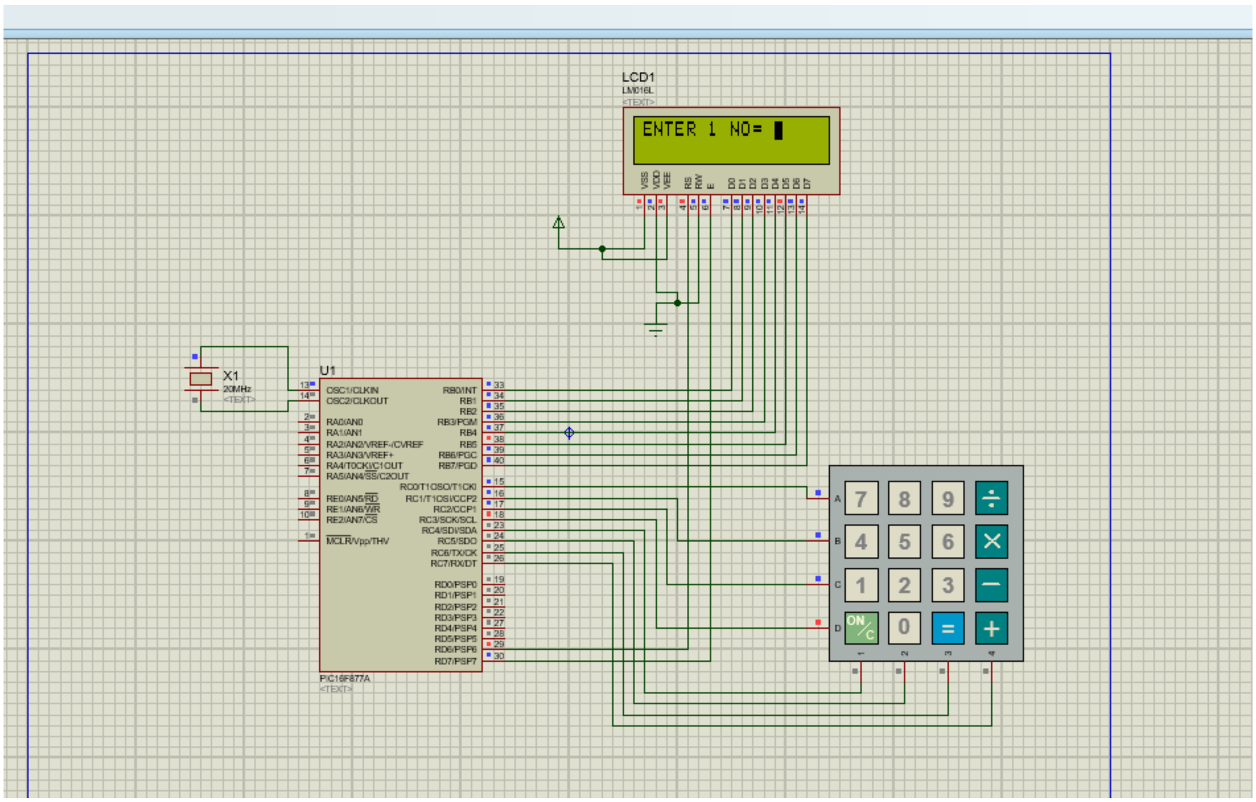
Hardware Implementation:

The system is constructed on the board after the circuit has been analysed and simulated in proteus ISIS Professional 8.0 portable simulation software.



Testing and Implementation:

The system was successfully developed utilising the system design, and the Microcontroller PIC 16F877A was written in C and compiled with the PIC C Compiler. Before the functioning programme was eventually produced and transmitted to the microcontroller chip, a series of programmes were written and simulated using the PIC simulator IDE and MPLAB. Simulations were run during the prototype implementation to ensure that the programmes were operating properly. The findings of a sampling result were compared to those of other calculators, and the results were identical.



Conclusion:

A portable, dependable, low-cost, and speedier means of calculation with a basic design is required. This project created and developed a Microcontroller-based calculator that allows for quick and easy calculations. The PIC 16F887 microcontroller was coded in C and compiled with proteus ISIS Professional 8.0 portable simulation software and PIC C Compiler. The calculator's results were found to be consistent with those of other calculators.

References:

1. Martin Bates (2006). Interfacing PIC Microcontrollers, Embedded Design by Interactive Simulation.

2. Electronics Projects with Microcontroller – www.jap.hu/electronics.html

3. Introduction to PIC Programming – www.quasare/eletronics.com

4. PIC16F887 Datasheet – http://ww1.microchip.com/ downloads/en/devicedoc/35007b.pdf

5. Nebojsa Matic and Dragan Andric (2010) PIC Microcontroller for beginners