Interfacing 16*2 LCD Display with ATTINY85

Components Required:

- ATtiny85
- 16X2 LCD
- Breadboard
- Battery

ATtiny85:
The ATtiny85 is a compact 8-bit microcontroller from the AVR family, known for its low power consumption and small size. Despite having only 8 pins, it supports multiple functionalities, including digital I/O, PWM, ADC, and I2C/SPI communication. It is widely used in embedded applications where space and power efficiency are crucial.

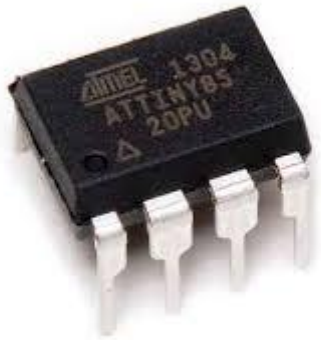

Fig23.1: ATtiny85

LCD Display:
A Liquid Crystal Display (LCD) is a visual output device commonly used in embedded systems for displaying text and simple graphics. The 16x2 LCD used in this project has two rows and 16 columns, allowing it to display up to 32 characters. It operates using a parallel interface, requiring multiple data pins for communication. The LiquidCrystal library simplifies the control of the display by managing initialization, text positioning, and real-time updates.

Fig23.2: 16X2 LCD Display

Battery:

A battery serves as the power source for the ATtiny85 and the LCD display, providing a portable and reliable energy supply. The choice of battery depends on the voltage and current requirements of the system. Common options include lithium-ion, lithium-polymer, and AA/AAA batteries. Efficient power management is essential when using batteries to extend the operational life of the circuit.

Fig23.3: Battery

Project:

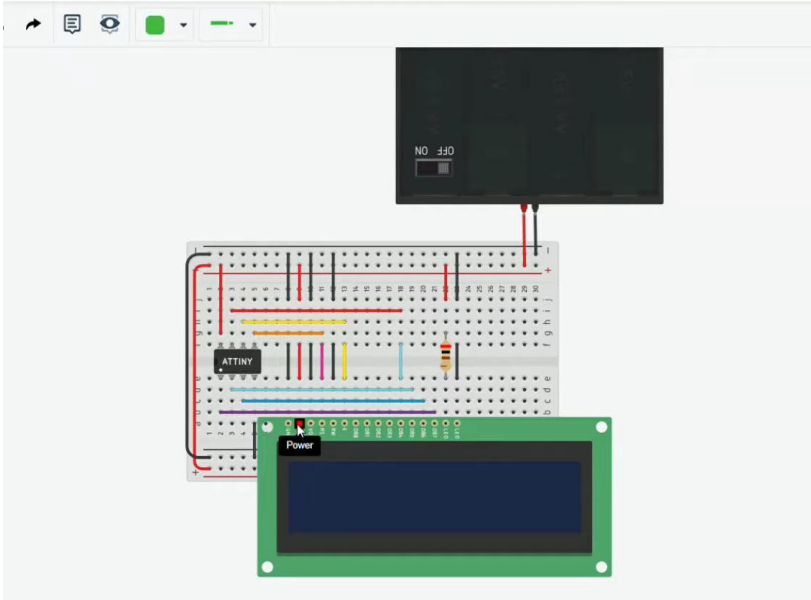

Fig23.4: Interfacing ATtiny85 with LCD

Explanation:

An LCD display has been interfaced with an ATtiny85 to show text and dynamic values. The system uses a 16x2 Liquid Crystal Display (LCD), which allows for two lines of text with up to 16 characters per line. The ATtiny85, a small yet powerful microcontroller, handles the display operation using the LiquidCrystal library, which simplifies communication with the LCD.

The initialization occurs in the setup() function, where the lcd.begin(16, 2) command sets the display size to 16 columns and 2 rows. The lcd.print() function is then used to display static text, with "This is me" appearing on the first row and "hello" on the second row. The setCursor() function positions the cursor at the beginning of the second line before printing.

In the loop() function, the program continuously updates the display by printing the number of seconds elapsed since the microcontroller started running. The millis() function provides the elapsed time in milliseconds, which is divided by 1000 to convert it into seconds. This value is printed at the 15th column of the second row, ensuring a real-time updating effect. The delay(100) function slows down updates to improve readability.

This project demonstrates fundamental concepts of interfacing an LCD with a microcontroller, including initialization, cursor positioning, and real-time display updates. It can be expanded to show sensor data, user inputs, or interactive menus, making it useful for various embedded system applications.