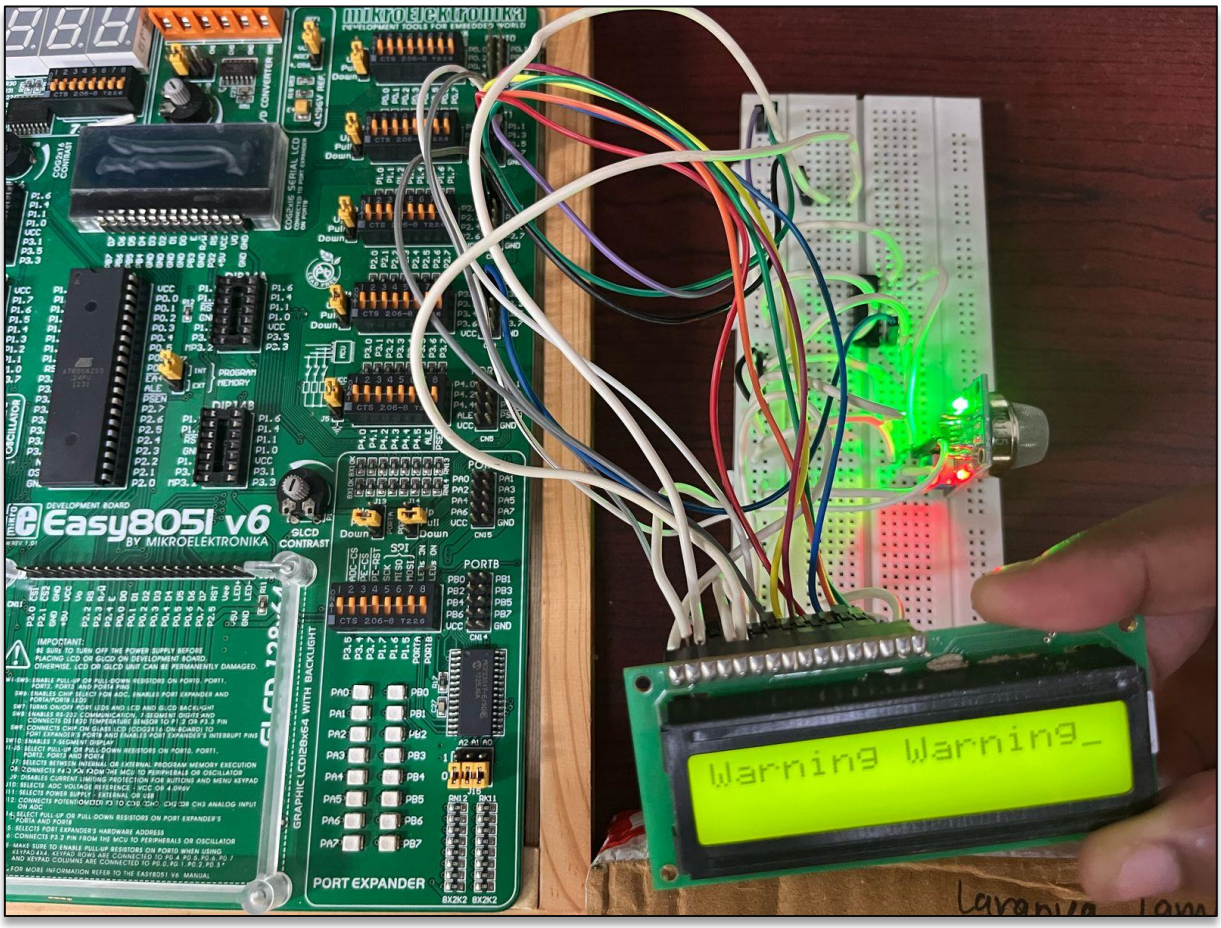
EMBEDDED SYSTEMS LAB PROJECT



GAS DETECTION SYSTEM

USING 8051 MICROCONTROLLER

BY –

LAVANYA TAMGADE-122EC0354

KIRTIKA PRIYADARSHINI-122EC0350

YELURINAGA SAI SUSMITHA-122EC0815

Abstract

Gas leakage presents a serious safety hazard in both industrial and residential environments. This research focuses on addressing this critical issue by designing a system that promptly detects gas leaks and notifies gas plant supervisors or homeowners. The system also sends SMS alerts to designated individuals, ensuring timely awareness of potential dangers. The relevance of this work is particularly significant in the current pandemic, where industries face workforce shortages due to social distancing measures. By providing real-time notifications, the proposed system serves as a vital tool for mitigating risks and maintaining safety, even with reduced on-site personnel.

Introduction:-

Problem Statement

GAS DETECTION SYSTEM USING 8051 MICROCONTROLLER

Objectives

The primary aim of this experiment is to develop a **Gas Leakage Detection System** designed to promptly alert supervisors in industrial plants or homeowners about potential gas leaks. The key objectives are:

1. **Early Detection of Gas Leaks**: Create a system that quickly identifies gas leaks in both industrial and household settings to ensure timely response and safety.

2. **Notification Mechanism**: Integrate a notification system to alert relevant personnel through visual indicators, audible alarms, or other communication methods.

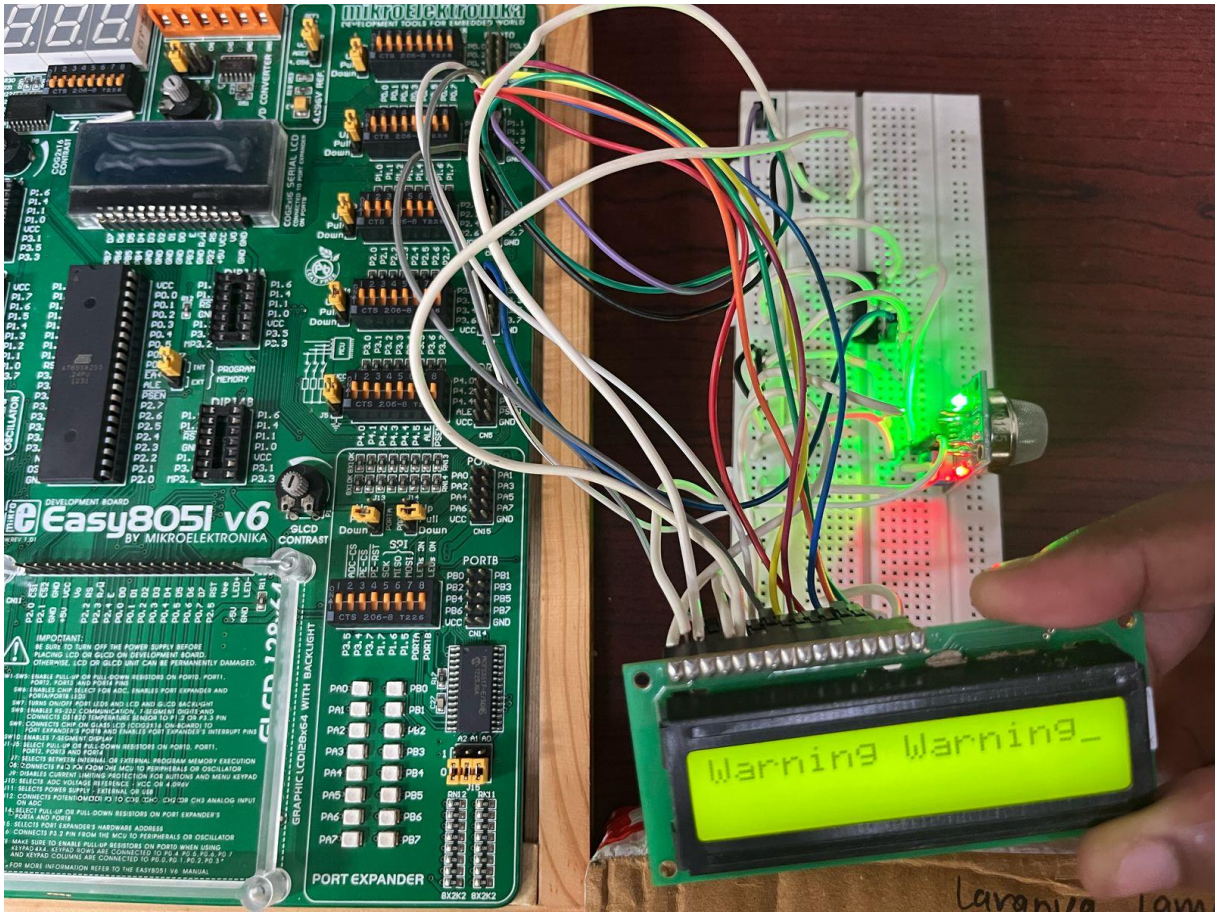
3. **Simulation Using Proteus Software**: Simulate the system's functionality using Proteus software to provide a virtual demonstration, validate its performance, and identify potential improvements.

Scope

The system can be used in homes, industries, and laboratories for real-time detection of gas leaks using embedded systems.

Hardware Design

Includes 8051 microcontroller, MQ-135 gas sensor, LCD display, connectors, and power supply.



Components list

1. Gas Sensor (MQ-6)
2. 8051 Microcontroller
3. LCD Display
4. **Proteus Software** for simulation
5. **Connectors and Wiring** (Male-to-Male / Female-to-Female)

Circuit diagrams

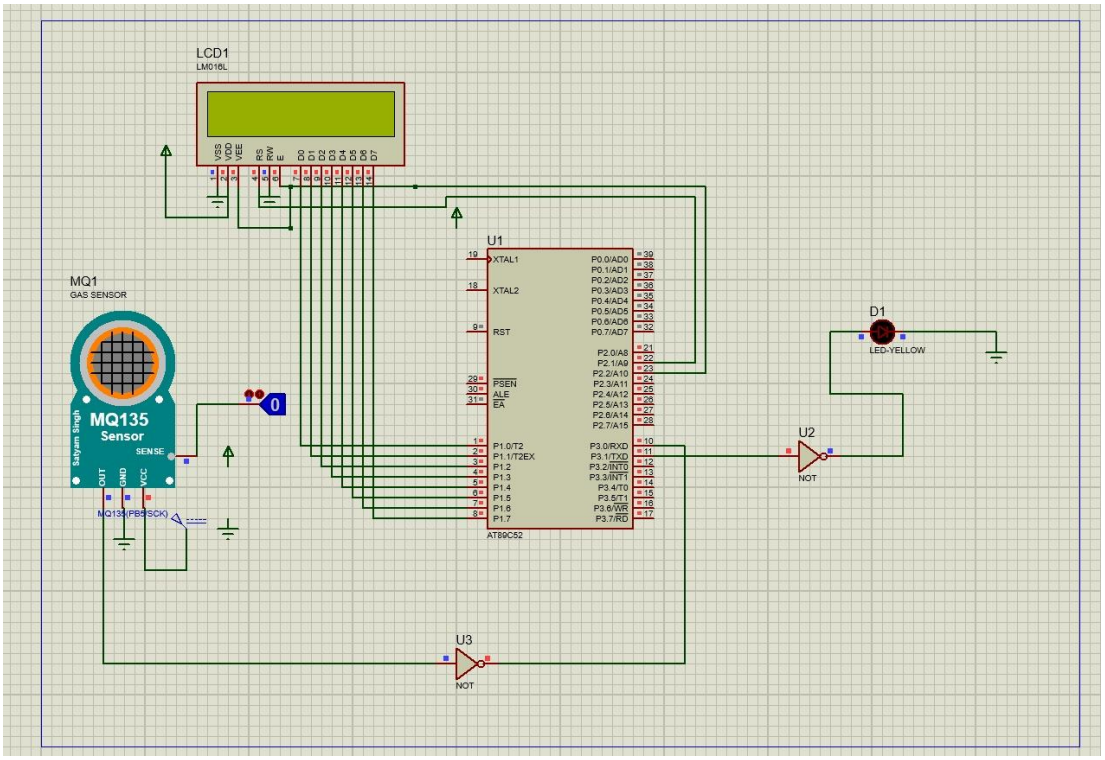


Figure 1: Schematic of the proteus design (inactive)

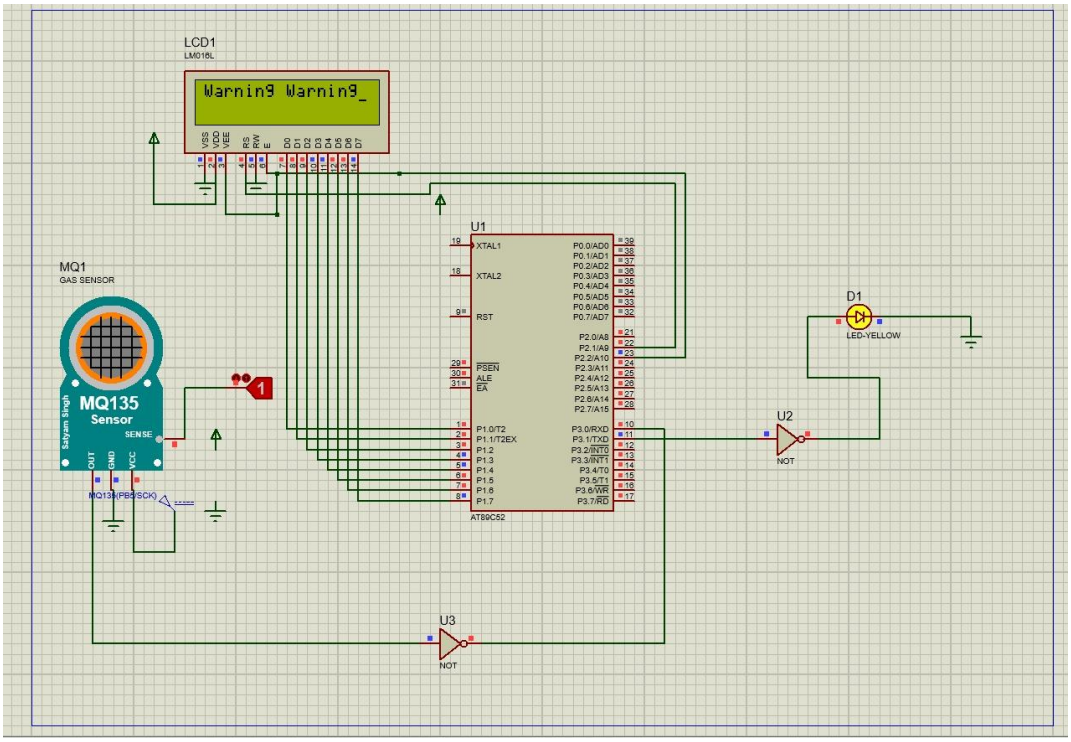


Figure 1: Schematic of the proteus design (active)

Interface details

1) **Microcontroller:** AT89S52-based development board
2) **Integrated Components:**
    - MAX232 for serial communication
    - On-board 16x2 alphanumeric LCD with adjustable contrast
    - DS1307 Real-Time Clock (RTC) module
3) **Power Supply:**
    - On-board 5V voltage regulator
    - Supports battery or DC input for uninterrupted operation
4) **Additional Features:**
    - Port expander for added I/O functionality
    - Reset button for system reinitialization
    - RS-232 connector for external interfacing
5) **Core System Functionality:**
    - 8051 microcontroller manages all components
    - MQ-135 gas sensor detects methane/propane
    - Sensor output processed via ADC
    - LCD provides real-time leakage alerts
    - Microcontroller triggers alarms when gas exceeds threshold

Software Design

```
ORG 0000H

JMP START

ORG 0100H

START:

 MOV P3, #0FFH ; Initialize Port 3 with 0xFF

 MOV R3, #23 ; Initialize R3 with 23

Here1:

 MOV R2, #255 ; Initialize R2 with 255
```

```
Here2:
 MOV R1, #255 ; Initialize R1 with 255
Here:
 DJNZ R1, Here ; Decrement R1 and jump if not zero
 DJNZ R2, Here2 ; Decrement R2 and jump if not zero
 DJNZ R3, Here1 ; Decrement R3 and jump if not zero
LOOP:
 MOV A, P3 ; Move the value of Port 3 to accumulator A
 CJNE A, #0FFH, LOOP ; Compare A with 0xFE and jump to LOOP if not equal
 CJNE A, #0FEH, DETECT ; Compare A with 0xFF and jump to DETECT if not equal
DETECT:
 CLR P3.1 ; Clear bit P3.1
 SETB P0.0 ; Set bit P0.0
 Mov A, #38H ; Load A with 38H
 ACALL com ; Call subroutine com
 MOV A, #0EH ; Load A with 0EH
 ACALL com ; Call subroutine com
 MOV A, #80H ; Load A with 80H
 ACALL com ; Call subroutine com
 MOV A, #01H ; Load A with 01H
 ACALL com ; Call subroutine com
 MOV DPTR, #STR ; Load DPTR with the address of the string STR
REV:
 MOV A, #00H ; Load A with 00H
 MOVC A, @A+DPTR ; Move code from the code memory to A using DPTR as a pointer
 JZ FINISH ; Jump to FINISH if A is zero
 ACALL L_D ; Call subroutine L_D
 INC DPTR ; Increment DPTR
 SJMP REV ; Jump to REV
```

```
FINISH:
 SJMP FINISH ; Jump to FINISH
 com:
 ACALL DEL_ROUTINE ; Call subroutine DEL_ROUTINE
 MOV P1, A ; Move the value of A to Port 1
 CLR P2.1 ; Clear bit P2.1
 SETB P2.2 ; Set bit P2.2
 CLR P2.2 ; Clear bit P2.2
 RET ; Return from subroutine
L_D:
 ACALL DEL_ROUTINE ; Call subroutine DEL_ROUTINE
 MOV P1, A ; Move the value of A to Port 1
 SETB P2.1 ; Set bit P2.1
 SETB P2.2 ; Set bit P2.2
 CLR P2.2 ; Clear bit P2.2
 RET ; Return from subroutine
DEL_ROUTINE:
 MOV R0, #0FFH ; Initialize R0 with 0xFF
L1:
 MOV R1, #0FFH ; Initialize R1 with 0xFF
L2:
 DJNZ R1, L2 ; Decrement R1 and jump if not zero
 DJNZ R0, L1 ; Decrement R0 and jump if not zero
 RET ; Return from subroutine
STR:
 DB 'Warning Warning', 0 ; Define a string "Warning Warning"
END
```

Algorithm overview

Initialization:

- The program begins at memory address 0000H (using ORG) and jumps to the START label.
- Port 3 (P3) is initialized with 0xFF, and registers R3, R2, and R1 are set with specific values.
- Three nested loops (Here1, Here2, and Here) are used for countdown operations with the DJNZ (Decrement and Jump if Not Zero) instruction.

Loop and Detection:

- In the LOOP section, the value of P3 is read, and specific conditions (0xFE or 0xFF) are checked. If met, the program jumps to the DETECT section.
- In DETECT, specific actions occur, such as manipulating bits in P3.1 and P0.0, and calling subroutines (com, L_D).

Subroutines:

- com: Handles port manipulations and calls a delay routine (DEL_ROUTINE).
- L_D: Involves additional port manipulations and control flow.

Delay Routine (DEL_ROUTINE):

- Registers R0 and R1 are initialized and used in nested loops to create time delays, a common practice in assembly-level microcontroller programming.

String Manipulation:

- The program retrieves a string, "Warning Warning," from code memory and processes it character by character.

Subroutine Calls and Control Flow:

- Subroutines are invoked using ACALL (Absolute Call), and flow control is managed using conditional (CJNE) and unconditional (SJMP, JMP) jump instructions.

End of Code:

- The END directive signifies the end of the assembly program.

Flowcharts

Figure 2 illustrates the process flow of the Gas Detection System. The system begins by initializing its key modules, including the LCD display, LED indicators, and the MQ-6 gas sensor. Once the initialization is complete, the output from the gas sensor is provided as a digital input to the 8051 microcontroller. The microcontroller then processes this input and evaluates it against the conditions specified in Table 1.

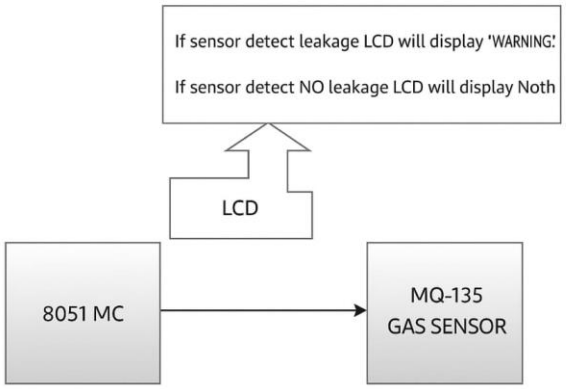


Figure 2: Block diagram of Gas Detection System (Circuit Connection)

Digital Output of gas sensor	Alert Status
1	Alert not generated
0	Alert generated

System Response

Summary When No

Alert is Generated:

1. The **LCD** displays: "NO Alert detected."
2. The **LED** remains OFF.

When an Alert is Generated:

1. The **LCD** displays: "Alert detected."
2. The **LED** turns ON.

Key code segments

- **Program Initialization:**
  - The program starts at memory address 0000H and jumps to the main logic at 0100H.
  - Port 3 is initialized to receive sensor input.
- **Delay Loop Setup:**
  - Nested delay loops using registers R1, R2, and R3 are implemented to create timing delays essential for stable sensor reading and LCD display control.
- **Main Detection Loop:**
  - Continuously checks Port 3 for sensor data.
  - Compares incoming values to detect a change in gas levels (active-low signal logic).
- **Alert Handling:**

  On detecting gas leakage:

  - Triggers output signals like turning ON LEDs or buzzer.
  - Initializes LCD display and clears previous content.
- **LCD Display Routine:**
  - Loads and displays a warning message ("Warning Warning") on the LCD.
  - Uses pointer logic to read and display characters sequentially from code memory.
- **Subroutines Used:**
  - com: Sends command instructions to the LCD.
  - L_D: Sends display data to the LCD.
  - DEL_ROUTINE: Creates a software delay for timing management.
- **Loop Control:**
  - The program loops back to monitor gas levels continuously after displaying the alert.

Implementation

Development process

- Followed a code-first approach with iterative testing and debugging.
- Synchronized LCD display outputs with the alert system (buzzer).
- Focused on timing precision to ensure reliable alert messages.
- Addressed hardware-software integration issues during development.

Testing approach

- **Simulation Tool:** Used Proteus software to simulate and validate the circuit design.

- **Verification Steps:**
- **LCD Interfacing Challenges:**
    - Required accurate timing and sequence control to ensure smooth LCD operation.
- **Buzzer Voltage Compatibility:**
    - A relay was added to support the 12V buzzer, as it wasn't compatible with the 5V microcontroller output.
- **Sensor Logic Handling:**
    - MQ-135 sensor outputs an active-low signal, which required careful handling in software logic.
    - Emphasized the need for detailed datasheet reference during development.

Results

Output on LCD shows **“Alert detected”** or **“No Alert detected”**

Performance data

- **System response time was fast, with detection occurring within 1–2 seconds of gas exposure.**
- **Detection accuracy was high under controlled simulation conditions in Proteus.**
- **LCD responded instantly to status changes, ensuring real-time feedback.**
- **System consistently triggered the buzzer when gas concentration crossed the threshold.**

Key findings

Gas Sensor Output	System Response
1 (No gas detected)	"No Alert Detected" on LCD
0 (Gas detected)	"Alert Detected" on LCD + Buzzer ON

Conclusion

The integration of embedded technology into gas leakage detection systems can significantly enhance safety in industries and households. The widespread use of LPG, from cylinders to industrial pipelines, presents a pressing security challenge. This prototype addresses this issue by offering a reliable, accurate, and delay-efficient solution using the 8051 microcontroller.

While many real-time gas leakage detection methods exist, they often come with drawbacks such as extended detection times or reliance on external visual systems. The proposed design overcomes these limitations, making it a suitable choice for real-time implementation across a variety of applications.

Achievements

- Successfully simulated a real-time gas detection system using Proteus.
- Accurately displayed gas status messages on an LCD.
- Integrated visual and audible alerts through microcontroller programming.
- Demonstrated reliable response to varying gas conditions using the MQ-135 sensor.

Limitations

- **Buzzer Voltage Issue:** Required a relay due to mismatch with microcontroller output voltage.
- **Sensor Signal Type:** MQ-135 outputs active-low signals, which added logic complexity.
- **LCD Synchronization:** Precise timing was needed to avoid display glitches during alerts.

Future improvements

- **GSM Module Integration:** Send SMS alerts to users during gas leakage incidents.
- **Enhanced Detection Range:** Use additional or more sensitive gas sensors.
- **IoT Connectivity:** Integrate with cloud platforms for remote monitoring and data logging.
- **Mobile App Interface:** Provide real-time alerts and system status via a smartphone application.

References

Here are the authoritative datasheets and documentation sources for the components used in this gas leakage detection system:

1. **MQ-135 Gas Sensor Datasheet**

   https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-135.pdf

2. **8051 Microcontroller (AT89S52) Datasheet** –

   https://ww1.microchip.com/downloads/en/devicedoc/doc1919.pdf

3. **MAX232 (Serial Communication IC) Datasheet** –

   https://www.ti.com/lit/ds/symlink/max232.pdf

4. **DS1307 Real-Time Clock (RTC) Datasheet** –

   https://datasheets.maximintegrated.com/en/ds/DS1307.pdf

5. **16x2 LCD Display Datasheet** –

   https://www.sparkfun.com/datasheets/LCD/HD44780.pdf