EMBEDDED SYSTEM PROJECT[EE3401]

Electrical Engineering Project submitted to the
National Institute of Technology Rourkela
In partial fulfillment of

Bachelor in technology *in*
Electrical Engineering
By

MAYUR KAMAL-122EE0904
AJIT KUMAR-122EE0826
KUSHAL KUMAR-122EE0600

Under the supervision of
Dr. Supratim Gupta

Department of Electrical Engineering
National Institute of Technology Rourkela

LPG Gas Detector Using 8051 Microcontroller

Abstract: In today's world, ensuring safety is of utmost importance, especially in environments where flammable gases like LPG are commonly used. With the increasing reliance on automation, using technology to detect potential hazards can greatly enhance safety and reduce risks. This project focuses on a microcontroller-based **Gas Leakage Detection System** designed to detect the presence of harmful gases automatically. The system uses an **MQ2 sensor** to identify gas leaks and activates an alarm to alert users promptly. This solution not only improves safety but also leverages automation to minimize human intervention in critical situations.

1. Objective

The rapid increase in LPG usage in both residential and industrial sectors has led to a rise in accidents caused by gas leaks. To address this growing concern, we have designed a gas detection system using an **8051 microcontroller (MC).** The system utilizes an **MQ2 gas sensor** to monitor the presence of LPG (Liquefied Petroleum Gas) in the air. When the sensor detects a gas leak, it activates an **alert mechanism**, which includes an **LED indicator**, **buzzer**, and a **relay module**. The relay controls electrical appliances connected to the main power supply.

The primary goal of this system is to provide an **efficient, cost-effective solution** for early detection of gas leaks, improving safety in residential and industrial areas. The design integrates both hardware and software components, ensuring reliable operation. The system also includes **remote monitoring capabilities**, enabling users to keep track of gas levels and receive real-time notifications. This approach significantly reduces the risk of hazardous incidents, ensuring a safer environment by promptly addressing gas leaks.

2. Circuit Diagram

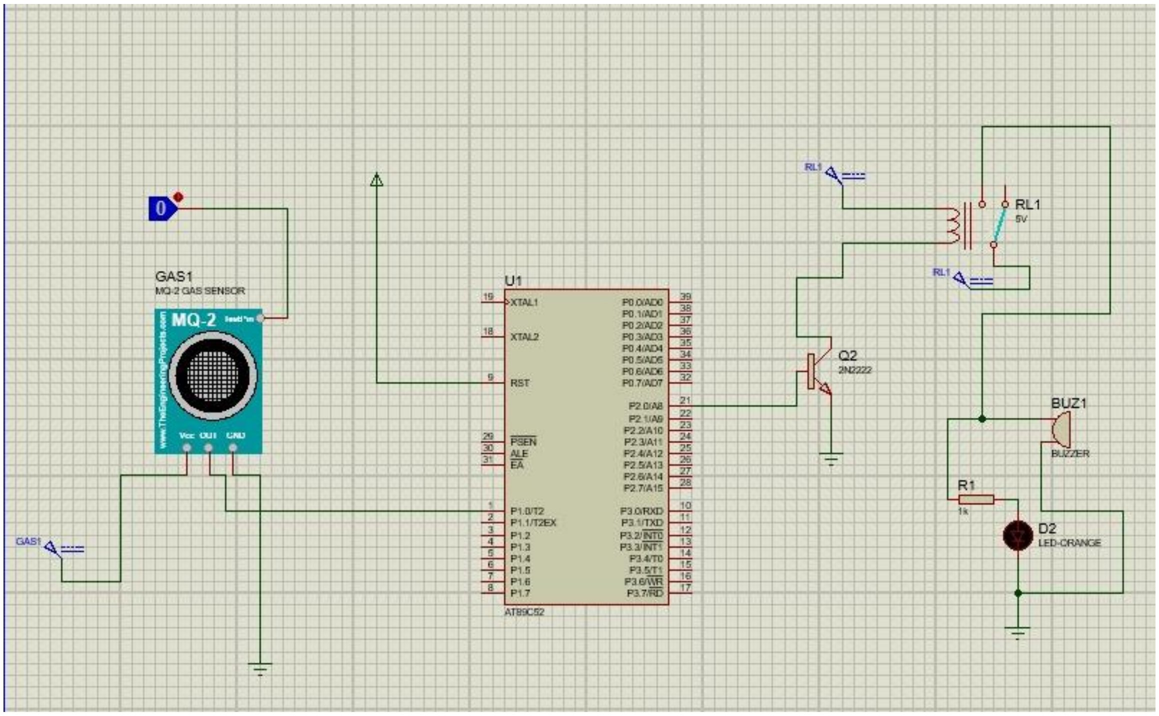

Figure 1: Schematic of the proteus design (inactive)

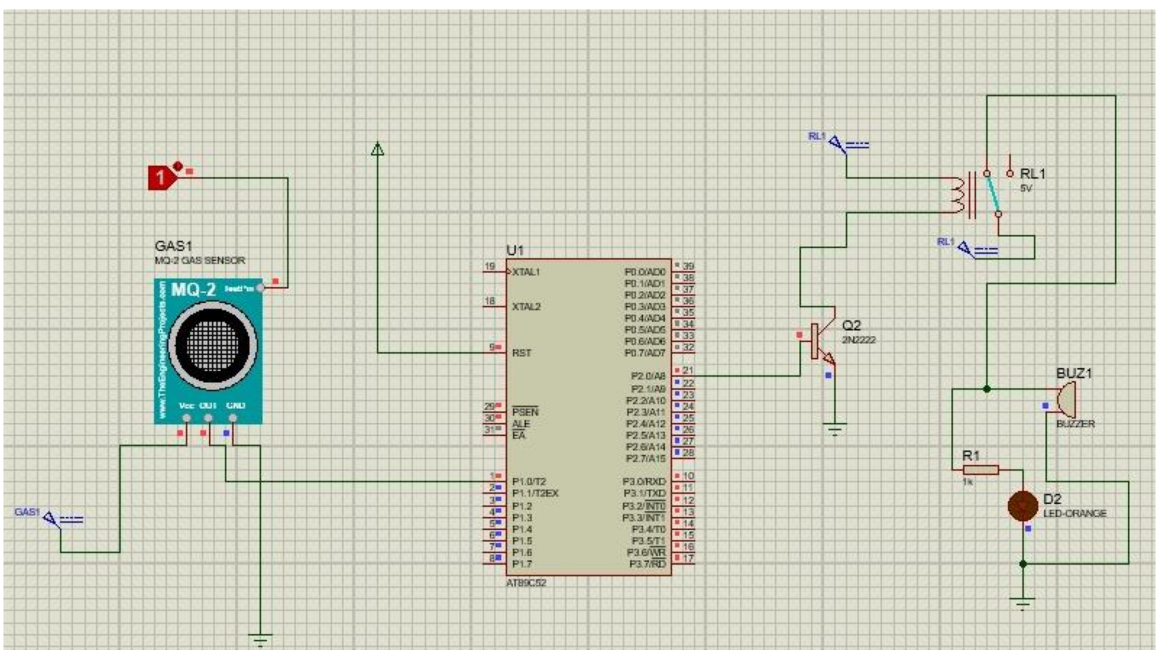

Figure 2: Schematic of the proteus design (Active stage)

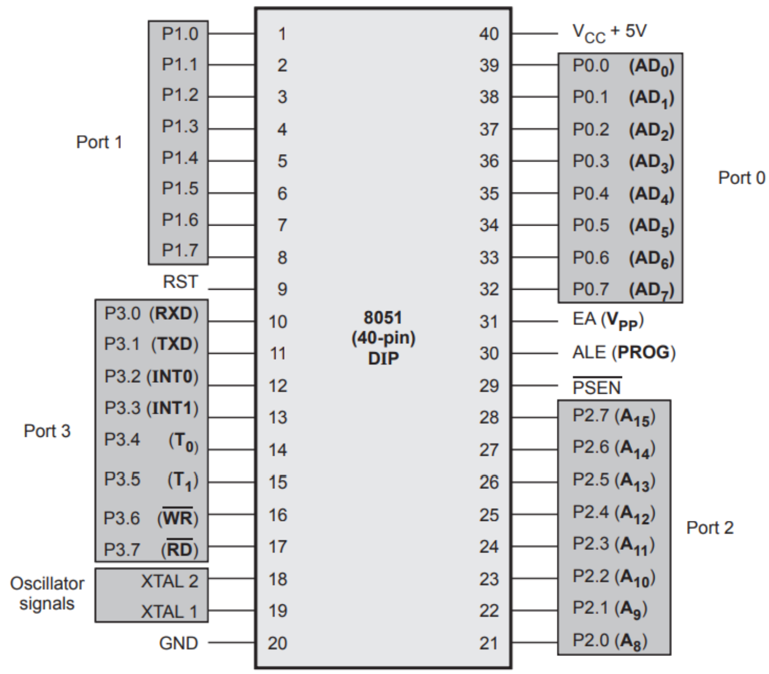

Figure 3: I/O Ports 8051 Microcontroller

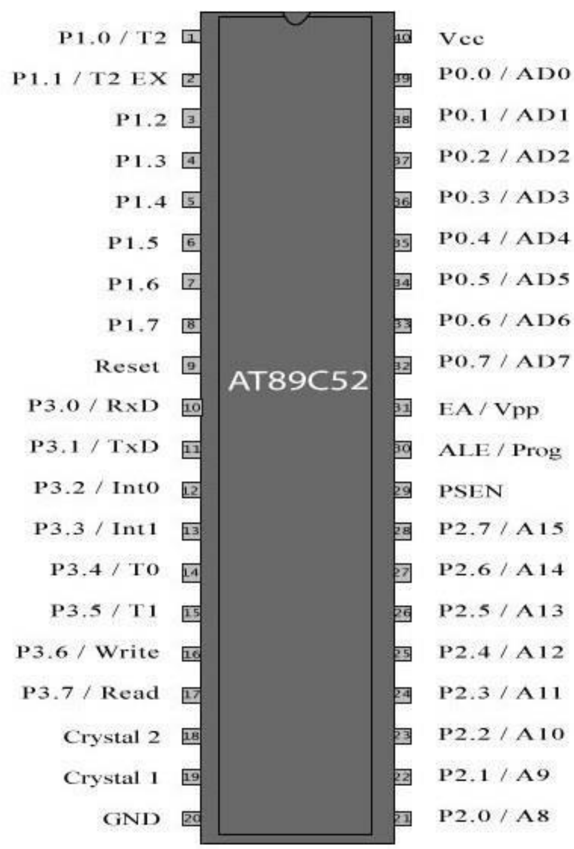

Figure 4: Pin Diagram of AT89C52 Microcontroller

3. Bill of Components

Component	Specification	Quantity	Cost (INR)
Adapter	12V DC	1	80
Development Board	8051 Microcontroller	1	350
Microcontroller	AT89C52	1	110
MQ2 Gas Sensor	Gas Detection	1	80
Wires	Connecting Wires	-	30
Resistor	1 kΩ	1	2
LED	5mm	1	5
Buzzer	5V	1	20
Relay	JQC3F	1	50
Transistor	2N2222	1	18

Total Cost: INR 745

Figure 5: MQ2 Gas Sensor

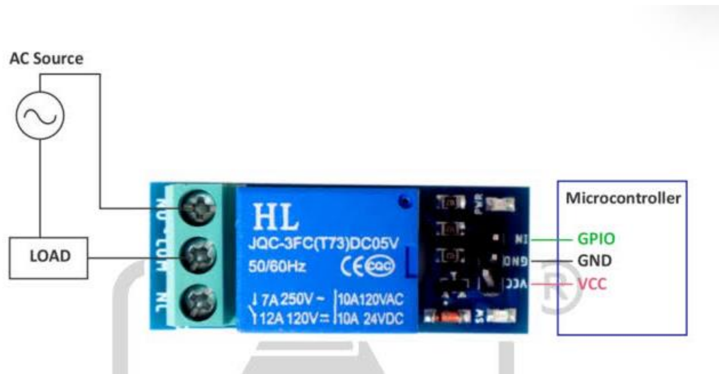

Figure 6: Relay JQC3F

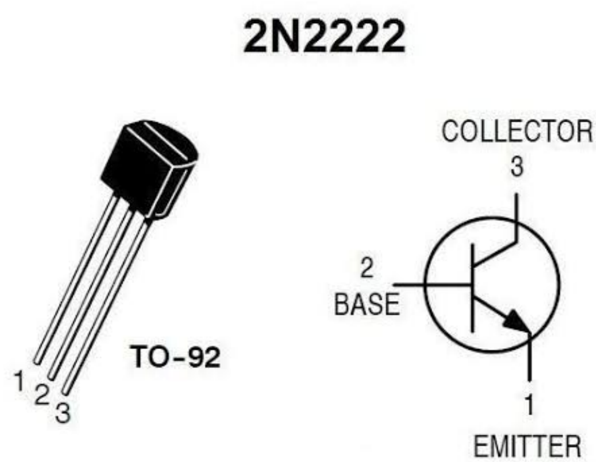

Figure 7: Transistor 2N2222

System Design

The system consists of an 8051-microcontroller interfaced with the following components:

- **MQ2 Gas Sensor**: Provides digital output based on the concentration of LPG in the air.
- **Buzzer and LED**: Serve as alert mechanisms.
- **Relay Module**: Controls electrical appliances connected to the main power supply.

When the gas sensor detects leakage beyond a pre-set threshold, it sends a high signal to the microcontroller. The microcontroller then activates the buzzer and LED while turning off the relay to cut off the power supply, preventing potential hazards.

Process Flow

1. **Initialization**: The system initializes all modules (MQ2 sensor, buzzer, relay).
2. **Detection**: The MQ2 sensor continuously monitors the gas concentration.
3. **Alert Mechanism**: If gas is detected, the microcontroller activates the buzzer and LED.
4. **Reset State**: The system returns to a normal state once the gas levels drop below the threshold.

4. Assembly Code

```
;====================================================================
; Main.asm file generated by New Project wizard

; Created:   Wed Oct 30 2024
; Processor: AT89S52
; Compiler:  ASEM-51 (Proteus)
;====================================================================
; RESET and INTERRUPT VECTORS
;====================================================================
    ; Reset Vector
;====================================================================
; CODE SEGMENT
;====================================================================
;org   0100h
```

```
    ;reti; Set the origin of the program memory to address 0100h
;start:
    MOV P1, #01H    ; Move the value 01H into register P1
    MOV P2, #00H    ; Move the value 00H into register P2

JUMP:
    MOV A, P1       ; Move the value in register P1 to the accumulator A
    CJNE A, #01H, JUMP  ; Compare A with 01H; if not equal, jump to the label JUMP
    CJNE A, #00H, CHECK ; Compare A with 00H; if not equal, jump to the label CHECK

CHECK:
    MOV R3, #0FFH   ; Move the value 0FFH into register R3
HERE1:MOV R2, #0FFH   ; Move the value 0FFH into register R2
HERE2:MOV R1, #0FH    ; Move the value 0FH into register R1
HERE: DJNZ R1, HERE   ; Decrement R1 and jump to HERE if R1 is not zero
    DJNZ R2, HERE2  ; Decrement R2 and jump to HERE2 if R2 is not zero
    DJNZ R3, HERE1  ; Decrement R3 and jump to HERE1 if R3 is not zero
    CJNE A, #01H, JUMP  ; Compare A with 01H; if not equal, jump to the label JUMP

Loop:
    MOV A, P1       ; Move the value in register P1 to the accumulator A
    MOV P2, A       ; Move the value in A to register P2
    MOV A, P1       ; Move the value in register P1 to the accumulator A
    CJNE A, #00H, Loop  ; Compare A with 00H; if not equal, jump to the label Loop
;====================================================================
```

END ; End of the program

Assembly Code Breakdown

1. **Reset Vector Initialization**
   - The program starts executing from the address 0000h, which is the default starting point of the microcontroller when it is powered on or reset.
2. **Port Initialization**
   - P1 is initialized to 01H (sets the least significant bit to 1, rest are 0).
   - P2 is initialized to 00H (all bits set to 0).
   - This prepares the microcontroller's input (P1) and output (P2) ports.
3. **Main Monitoring Loop (JUMP)**
   - Continuously reads the value of P1 into the accumulator (A).
   - Compares the value of A:
     - If A is not equal to 01H, it loops back to JUMP.
     - If A is not equal to 00H, it jumps to the CHECK section.
4. **Delay Routine (CHECK)**
   - A delay routine is implemented using three nested loops with registers R1, R2, and R3.
   - This delay allows for timing adjustments, potentially useful for debouncing or providing a pause before proceeding with the next actions.
5. **Output Control (Loop)**
   - Continuously monitors the status of P1.
   - Mirrors the value of P1 to P2, effectively copying the input to the output.
   - The loop keeps running as long as P1 does not equal 00H.
6. **Program End**
   - The program terminates with the END directive, indicating the end of the source code for the assembler.

Functional Overview

- **Purpose**: The code continuously monitors the input port (P1) and updates the output port (P2) based on the detected value.
- **Delay Mechanism**: Uses a delay loop to create a pause, which can be adjusted by changing the values of the registers.
- **Output Reflection**: The value of P1 is reflected on P2 as long as P1 does not change to 00H.

5. Test Results

Gas Sensor Input	LED Output	Buzzer Output	Relay Output
No Gas (0)	OFF	OFF	OFF
Gas Detected (1)	ON	ON	ON

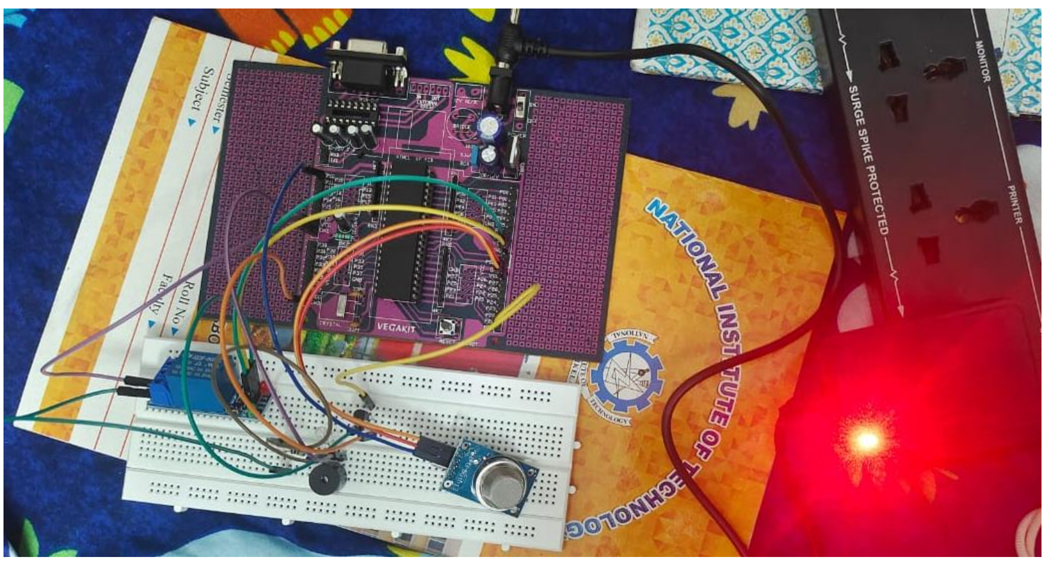

Figure 8: Hardware design of Schematic (inactive)

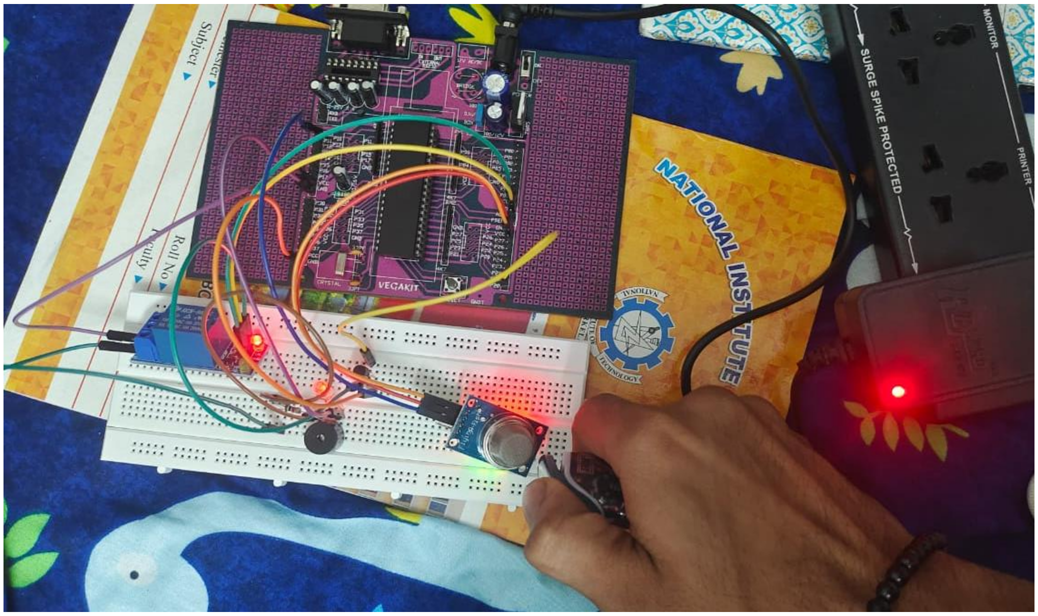

Figure 9: Hardware design of Schematic (Active)

6. Results and Discussion

Our gas detection system was successfully tested using the Proteus simulation environment. The system efficiently detected the presence of LPG gas and activated the alert mechanisms as expected. The relay successfully cut off the power supply to prevent potential fires, demonstrating the effectiveness of the design in reducing risks associated with gas leakage.

Key Observations:

- The system operates in real-time, providing immediate alerts upon detecting gas leaks.
- It is simple, cost-effective, and suitable for both domestic and industrial applications.
- The design can be further enhanced to detect gas concentration levels and send remote alerts via GSM modules for extended functionality.

7. Conclusion

This project demonstrates a practical solution for gas leak detection using an 8051 microcontroller. The prototype efficiently detects LPG leaks and takes prompt safety measures, making it an effective tool for ensuring safety in both households and industries. The system's simplicity, cost-effectiveness, and reliability make it a viable solution for real-time implementation.

8. FUTURE WORK

The current design can be further enhanced with the following features:

1. **Automatic Exhaust Fan Control**: Integrating an exhaust fan into the system controlled by the 8051 microcontroller can help quickly ventilate the area in case of gas leakage. When the gas sensor detects a leakage, the microcontroller can activate the exhaust fan along with the alert mechanisms. This feature ensures that accumulated gas is expelled from the enclosed space, reducing the risk of ignition.

2. **Gas Level Monitoring**: Implementing a weighing scale to monitor the gas cylinder's weight can provide real-time information on the remaining gas levels. This helps in the early detection of potential leaks due to damaged cylinders.

3. **Digital Signal Processing for Gas Concentration**: By incorporating DSP techniques, the system can detect and quantify gas concentration levels, allowing for more precise control of safety measures, such as turning on the exhaust fan at specific thresholds.

4. **IoT Integration**: Extending the system with IoT capabilities can enable remote monitoring and control. This would allow users to receive notifications and control devices like the exhaust fan via a smartphone app, ensuring safety even when away from the premises.