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

Sree Keerthi -122EE0366
Varshitha-122EE0818
B . Chandana-122EE0362

Under the supervision of
Dr. Supratim Gupta

Department of Electrical Engineering
National Institute of Technology Rourkela

Automatic Street Light Controller USING 8051 MICROCONTROLLER

ABSTRACT: Electricity being one of the primitive resources, must be utilised carefully, as the number of vehicles during night time are less frequent, building up a sensor-based street lights helps in saving electricity. The digital World we are living in allows us to use different technologies toautomatically perform certain tasks. Such automation is very useful in certain areas like energy consumption, reducing human efforts, improving the standard of living etc.The project implemented here is one such project where the microcontroller based system automatically controls the street lights.

1. OBJECTIVE

The aim of this project is to automatically turn on or off the street lights by detecting vehicle movement. We implemented this project using an 8051 Microcontroller and two Infrared (IR)sensors.

The job of the circuit is to turn on the first street light when a vehicle's movement is detectedand to turn off that light as soon as the same vehicle's movement is detected by the second street light. The goal is to conserve energy by lighting up only the necessary sections of the highway based on the presence and movement of vehicles.

2. CIRCUIT DIAGRAM

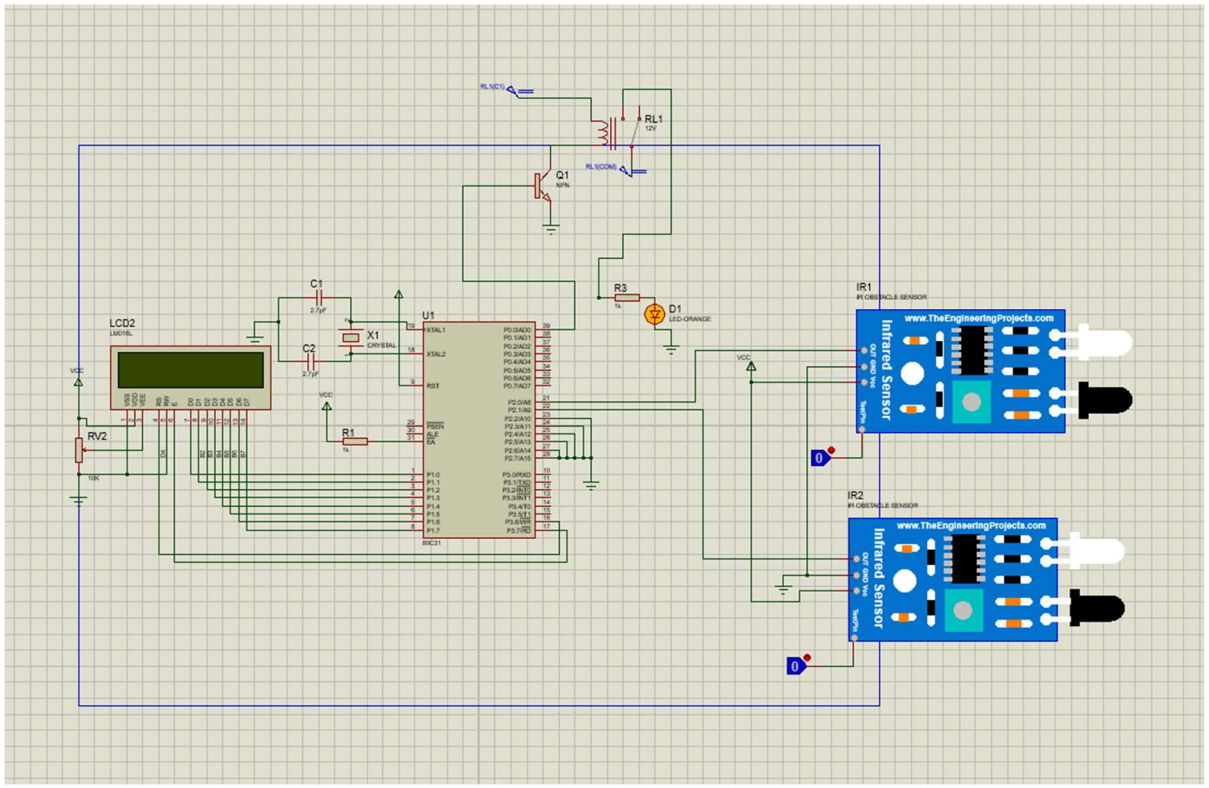

Figure 1: Schematic of the proteus design (inactive)

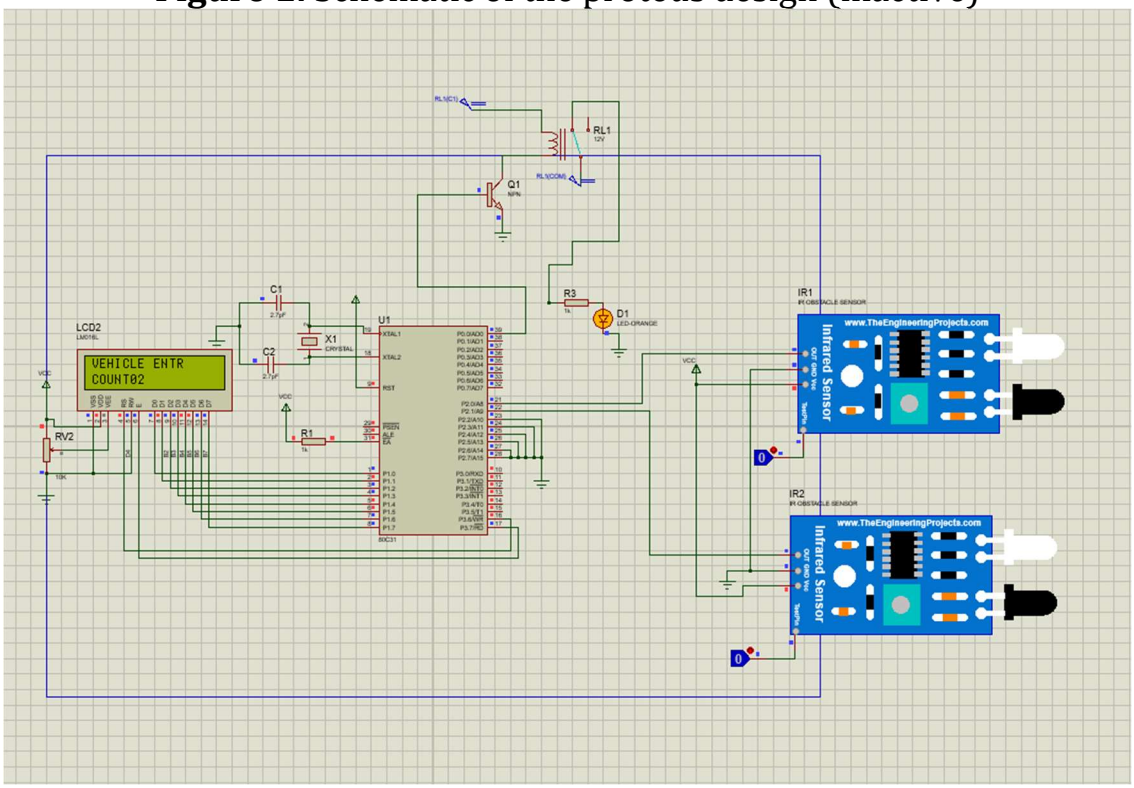

Figure 2: Schematic of the proteus design (Active stage)

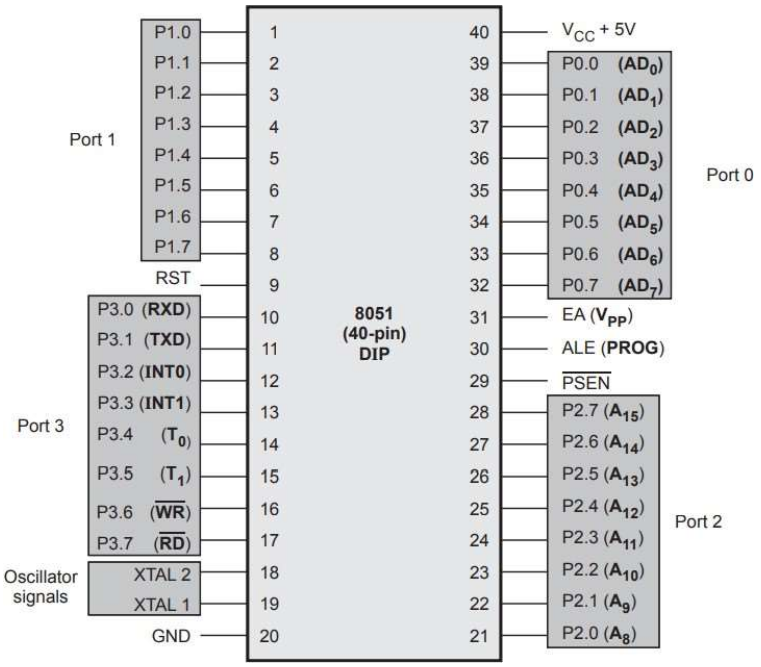

Figure 3: I/O Ports 8051 Microcontroller

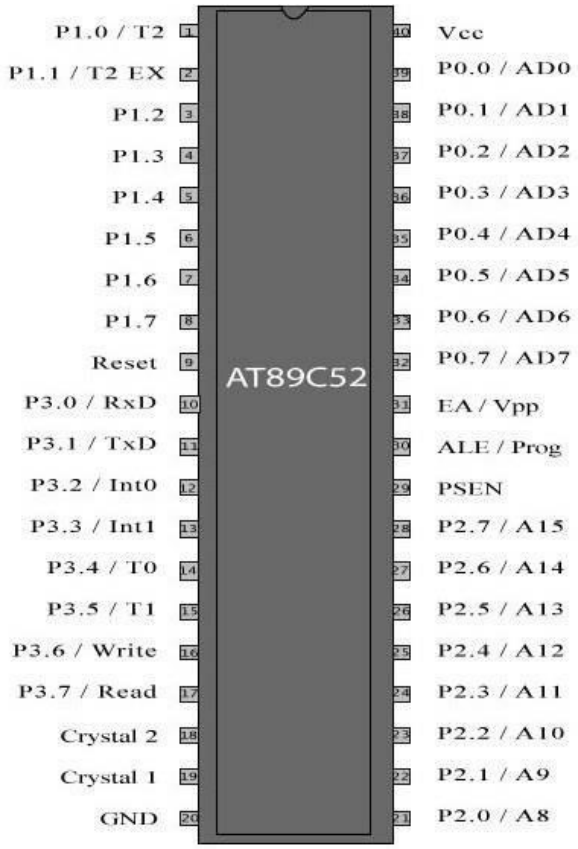

Figure 4: Pin Diagram of AT89C52 Microcontroller

3. Bill of Components

Component	Specification	Quantity	Cost (INR)
Adapter	12V DC	1	80
Development Board	8051 Microcontroller	1	350
Microcontroller	AT89C52	1	110
IR Obstacle Sensor	Object Detection	2	80
Wires	Connecting Wires	-	30
Resistor	1 kΩ	1	2
LED	5mm	1	5
BJT	BC547	1	20
Relay	JQC3F	1	50

Total Cost: INR 730

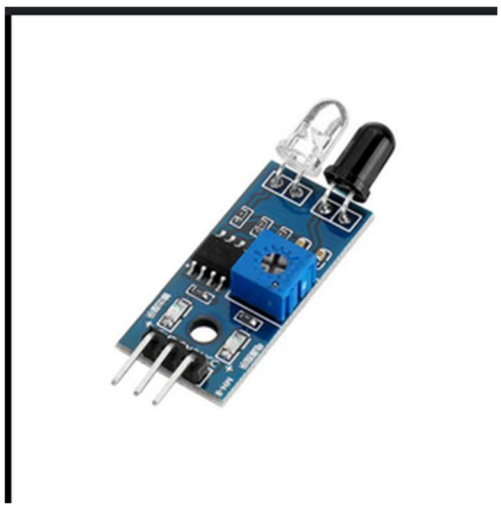

Figure 5:IR Obstacle Sensor

Figure 6: Relay JQC3F

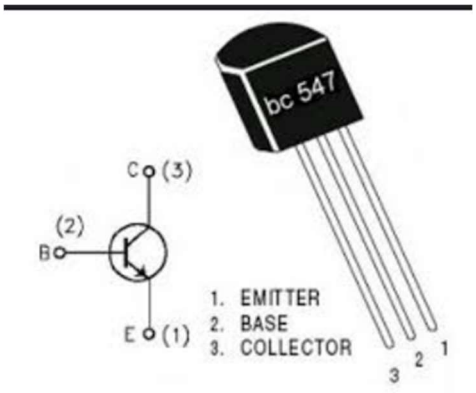

Figure 7: BJT BC547

System Design

This system implements an automatic street light control mechanism using an 8051 microcontroller. The design ensures efficient energy use by turning lights on and off based on vehicle movement detected by infrared (IR) sensors.

Key Components

1. **IR Sensors (2 Units)**
   - Detect vehicle movement at specific locations and send signals to the microcontroller.
   - Sensor placement determines system functionality.
2. **AT89C51 Microcontroller**
   - Processes sensor inputs and controls the streetlights accordingly.
3. **16 x 2 LCD Display**
   - Displays system status, such as active streetlight and sensor data
4. **5V Relay Module**
   - Controls the streetlight circuitry based on signals from the microcontroller.
5. **Lamps**
   - Streetlights that operate based on vehicle detection.
6. **Power Supply**
   - Provides power to the system components.

Process Flow

1. Initial State
- The system remains in standby mode with all streetlights turned off.
- IR sensors continuously monitor for vehicle movement.

2. Vehicle Detection by Sensor 1
- When a vehicle passes **Sensor 1**, it detects the movement and sends a HIGH signal to the microcontroller.
- The microcontroller:
  - Turns on **Streetlight 1**.
  - Displays "Streetlight 1 ON" on the LCD.

3. Transition to Sensor 2
- As the vehicle moves toward **Sensor 2**, the following sequence occurs:
  - **Sensor 2** detects the incoming vehicle, sending a HIGH signal.
  - **Sensor 1** detects the vehicle's departure, signaling that Streetlight 1 is no longer needed.

4. Light Transition
- Based on the sensor signals:
  - The microcontroller turns off **Streetlight 1**.
  - Simultaneously, it activates **Streetlight 2**.
  - The LCD updates to display "Streetlight 2 ON."

5. Reset State
- The system resets itself as the vehicle exits Sensor 2's range.
- All streetlights turn off, awaiting further vehicle detection.

4. Assembly Code

```
; storing essential information on ROM at 0400h (in range of 4K)
;MAKING LOOKUP TABLE
Table:
org 0400H
DB "STREET LIGHT VEHICLE EXIT VEHICLE ENTRY COUNT: NO VEHICLE
INVALID OPERATION"

;START OF PROGRAM FROM 0000h

org 0000H

MOV DPTR,#0400H          ; its used as data pointer that means its locates at
0400H ADDRESS of ROM
```

```
MOV P2,#0FFH              ; assigning PORT 2 as input

MOV P0,#00h

MOV R6,#00h               ; R6 ACTS AS COUNTER

; LCD SETUP,START DISPLAY,CHECKING SENSORS

ACALL Lcd_setup

ACALL Display

ACALL Check

Lcd_setup:
MOV A,#38H          ;setup 2 line 5*7 matrix display
ACALL command
MOV A,#0CH                ;Display ON and cursor OFF
ACALL command
MOV A,#01H          ;Clear the old data
ACALL command
;MOV A,#06H               ;if used then cursor increment mode
;ACALL command
MOV A,#80H                ;cursor home and starts left most point
ACALL command
RET


command:
  MOV P1,A          ;command on port A
  CLR P3.6          ; register select as 0 for command
  ;CLR P3.1         ; display mode as write mode(R/!W)
  SETB P3.7         ; Make latch as 1
  CLR P3.7          ; to falling edge
  ACALL Delay
RET

work:
  MOV P1,A   ;data on port A
  SETB P3.6  ; register select as 1 for data
  ;CLR P3.1  ; display mode as write mode(R/!W)
  SETB P3.7  ; Make latch as 1
  CLR P3.7   ; to falling edge
  ACALL Delay
  RET
```

```
Delay:                                  ;SOME delay for LCD
  MOV TMOD, #01H                ;Program      TMOD -->(0000      0001)2        ...
        Timer0        Mode1
  MOV TL0, #0D4H                        ;Load  lower byte   of      Count
  MOV TH0, #050H                        ;Load  upper byte   of      Count
  MOV TCON, #10H                        ;Program TCON      --> (0001      0000)2
...       start   Timer0
  WAIT: JNB TCON.5, WAIT                ;Wait for overflow
  MOV TCON, #00H                        ;Stop   Timer0
  RET


                        ;welcome message
Display:
  MOV R3,#0Ch                   ;display of welcome message
  MOV R2,#00h
  RET

                        ;checking

Check:
  CLR C
  ACALL Delay
  MOV A,P2              ;read the data
  MOV B,#0CH
  CJNE A,B,find         ;check the data if 00h(initial case)
  SJMP Check

                          ;identifying sensor
find:
  ACALL Delay
  CLR C
  CJNE A,#00h,goon
  ACALL Check
  goon:
  CLR C
  CJNE A,#01h,EXIT   ;checking with entry if not equal than its must be exit
  CLR C
  CJNE A,#02h,ENTRY  ;checking with eXIT if not equal than its must be ENTRY
  ACALL Check

                        ; ENTRY MODE
ENTRY:
```

```
  SETB P0.0
  ACALL Delay
  MOV A,#01H              ;Clear the old data
  ACALL command

  MOV R2,#1Ah                 ; displaying of person entering
  MOV R3,#26h
  ACALL lcd_displayer

  ACALL Entry_count           ; counting of persons
  CLR C


  SJMP Check

                        ;entry count
Entry_count:
  MOV A,#0C0H             ;FORCE CURSOR TO SECOND LINE
  ACALL command

  MOV R2,#28h                 ;displaying of "COUNT:"
  MOV R3,#2Dh
  ACALL lcd_displayer
  ;count increment
  CLR C
  MOV A,R6                ; getting data from R6 register
  ADD A,#01               ; adding "1"
  MOV R6,A                ;new data stored back to R6 counter register
  DA A                        ; converting from hex to decimal value (after
addition only)
  MOV R2,A
  ACALL ConvertDisplay        ; converting data to ASCII code
  RET

                        ; exit mode
EXIT:
  ACALL Delay
  MOV A,#01H              ;Clear the old data
  ACALL command
  CLR C
  CJNE R6,#00h,counter        ; check the counter
  MOV R2,#39h
  MOV R3,#3Fh
  ACALL lcd_displayer         ; if zero then give error message "INVALID"

  MOV A,#0C0H                 ; Force cursor to second line
```

```
  ACALL command

  MOV R2,#41h                    ; showing of error message "Operation"
  MOV R3,#49h
  ACALL lcd_displayer


  SJMP Check

counter:                         ; displaying of person leaving message
  MOV R2,#0Dh
  MOV R3,#18h
  ACALL lcd_displayer

  ACALL Exit_count               ; counting of persons
  CLR C
  CJNE R4,#00H,moveon
  CLR P0.0
  moveon:
  SJMP Check
                          ;exit counting
Exit_count:
  MOV A,#0C0H
  ACALL command
                           ;count decrement
  CLR C
  CJNE R6,#00h,start             ; check for zero if zero then display NO PERSON
  ACALL message
  RET

start:
  MOV A,R6                       ;get data from counter
  CLR C                          ; clear carry other wise it subtraction would be with
carry
  SUBB A,#01                     ; subtract with "01"
  MOV R6,A                       ; store to R6
  DA A                           ; converting from hex to decimal value (after
addition only)
  MOV R4,A                       ; lets store it at R4 (DAA data) upcoming operations
may distrub A


  CLR C
  CJNE R6,#00h,continue          ;BEFORE MOVING ON LETS CHECK IF ANY
PERSON IS THERE
  clr P0.0
```

```
  ACALL message

continue:                ;Displaying "count"
  SETB P0.5
  MOV R2,#28h
  MOV R3,#2Dh
  CLR C
  ACALL lcd_displayer

  MOV A,R4               ;value showing
  MOV R2,A
  ACALL ConvertDisplay

  ACALL Check



ConvertDisplay:
   CLR C   ; Clear carry flag
   MOV B,#10h
   MOV A,R2
   DIV AB      ; Divide A by B, quotient in A, remainder in B
   ADD A, #30h       ; Convert quotient to ASCII
   ACALL work              ; Display the ASCII character

   MOV A, B            ; Move the remainder back to A
   ADD A, #30h       ; Convert remainder to ASCII
   CALL work               ; Display the ASCII character

   RET

message:                 ;Displaying of "NO PERSON"
  MOV R2,#2Fh
  MOV R3,#37h
  ACALL lcd_displayer
  ACALL Check
  RET

lcd_displayer:           ;sending bit by bit to LCD display
  MOV A,R2
  MOV B,R3
do:
  MOVC A,@A+DPTR
  ACALL work
  INC R2
  CLR C
```

```
MOV A,R2
CJNE A,B,do
RET

Here: SJMP Here
END
```

Assembly Code Breakdown

1. Reset Vector Initialization

- The program begins execution at 0000H, which is the microcontroller's default starting address after power-on or reset.

2. Port Initialization

- **P2** is configured as the input port for sensors.
- **P0** is configured as the output port to control streetlights.
- Register R6 is initialized to track vehicle count.

3. Main Monitoring Loop (JUMP)

Purpose:

- Continuously monitors sensor input (Port 2) and identifies vehicle activity.

Reads input from Port 2 and compares it with predefined states:

- Default state (`0CH`): No vehicle detected.
- Redirects to the appropriate handling routine for sensor activity
-

4. Delay Routine (CHECK)

Purpose:

Provides a delay for timing adjustments, useful for sensor stabilization or debouncing.
Uses Timer 0 to generate a delay by counting to a specified value.

5. Output Control (Loop)

Purpose:

Controls streetlights based on vehicle detection by mirroring sensor input to outputs and updating the vehicle count.

Detects a vehicle entering and:

- Turns on the corresponding light.
- Displays the "ENTRY" message on the LCD.
- Updates the vehicle count.

6. Program End

Purpose:

- Marks the end of the program.

The program terminates with the END directive, ensuring proper assembly.

Functional Overview

- **Purpose**: The program implements an automatic streetlight control system using an 8051 microcontroller. It detects vehicle movement using infrared (IR) sensors and efficiently manages streetlight activation to conserve energy. The system lights up only necessary sections of a roadway, ensuring minimal power consumption.
- **Delay Mechanism**: The delay is generated using Timer 0 in mode 1. This stabilizes sensor readings and ensures smooth transitions between the activation and deactivation of streetlights.

  Vehicle Detection and Light Control:

  - **Entry Detection**: When a vehicle is detected by Sensor 1:
    - The microcontroller turns on the first streetlight.
    - The LCD displays "Streetlight 1 ON" along with the vehicle count.
  - **Exit Detection**: As the vehicle passes Sensor 2:
    - The microcontroller turns off the first streetlight and activates the second.
    - The LCD updates to reflect the transition to "Streetlight 2 ON."
    - The vehicle count is updated accordingly.

- **Output Reflection**: The output reflection mechanism ensures that the status of the input (vehicle detection) is accurately mirrored to the output (streetlight activation) in real time.

5. TEST RESULTS

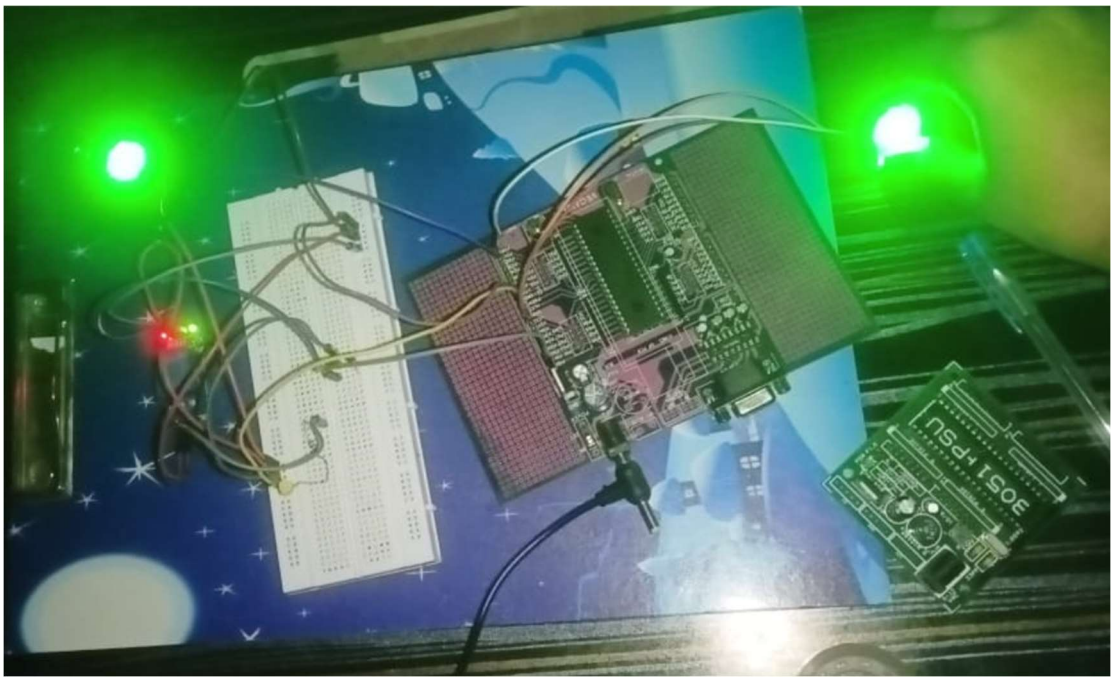

Figure 8: Hardware design of Schematic (inactive)

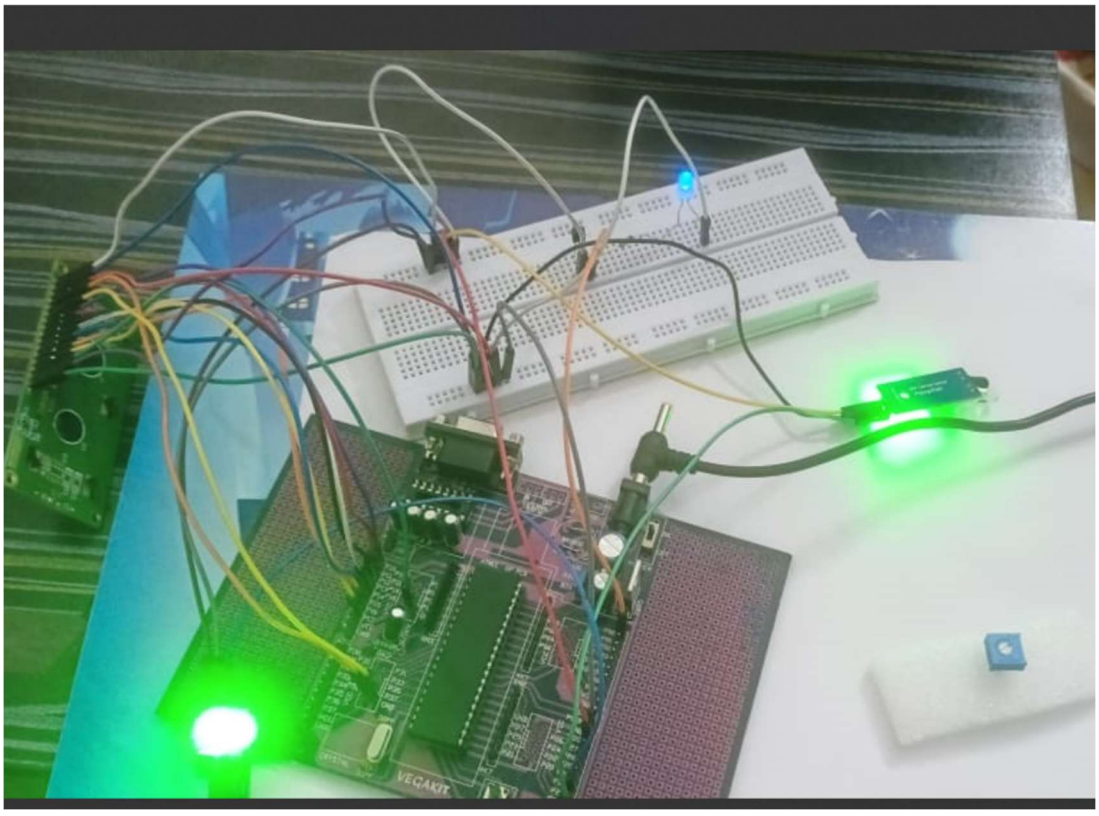

Figure 9: Hardware design of Schematic (Active)

6. Results and Discussion

Key Observations:

The use of an 8051 microcontroller allowed efficient processing and reliable performance.

The delay mechanism ensured smooth operation without sensor interference.

Dynamic output reflection on the streetlights provided a clear and intuitive demonstration of the system's functionality.

7. Conclusion

The system demonstrated practical implementation of an automated streetlight control mechanism using the 8051 microcontroller, achieving the objectives of energy efficiency and intelligent lighting. The experiment validated the system's ability to save energy by adapting to real-time vehicle movement and provided a scalable solution for smart city infrastructure.

8.FUTURE WORK

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

1. **Adaptive Brightness Control**:

Integrate **light-dependent resistors (LDRs)** to adjust the brightness of streetlights based on ambient lighting conditions.

Example: Dim the lights during dawn or dusk to save energy further.

2. **Wireless Communication**:

Add **Zigbee** or **LoRa modules** to enable wireless communication between microcontrollers for seamless coordination of multiple streetlights over a long stretch of road.

3. **Real-Time Data Logging**:

Use **IoT platforms** to log vehicle count and streetlight activity in real time.

This data can be analyzed for traffic patterns and maintenance schedules.

4.Overload Protection:

- Include a **current sensor** to detect overload or short circuits in the streetlight system and automatically turn off the affected lights to prevent damage.