Microprocessors project

Title: Smart Street Light Controller

Members: Aimen Laidi/Moustafa asmar

Advisor: pr Mustafa TÜRKBOYLARI

1. Introduction

This project presents the design and implementation of a Smart Street Light Controller using the PIC16F877A microcontroller. The system automatically controls a street light based on environmental light conditions and human motion detection. An LDR sensor is used to distinguish between day and night, while a PIR sensor detects motion at night. The system also displays real-time status information on a 16x2 LCD. The entire system was designed and simulated using Proteus, and the firmware was developed using the XC8 compiler.

The main objective of this project is to demonstrate the practical use of microcontroller peripherals such as digital I/O ports, timers, interrupts, and LCD interfacing in a real-world embedded system application.

2. Circuit Schematic Diagram (Proteus Design)

The complete circuit schematic was designed and simulated using Proteus ISIS. The major components used in the circuit include:

- PIC16F877A microcontroller
- Light Dependent Resistor (LDR)
- Passive Infrared (PIR) motion sensor
- LEDs representing street lights
- 16x2 Character LCD (LM016L)
- 20 MHz crystal oscillator with capacitors
- Current-limiting resistors and pull-up resistors

2.1 Pin Connections

- RB0: Connected to the LDR output to detect day/night conditions.
- RD2: Connected to the PIR sensor output for motion detection.
- RC1: Connected to an LED representing the street light.
- PORTD (RD0–RD7): Connected to the LCD (RS, EN, D4–D7) in 4-bit mode.
- OSC1/OSC2: Connected to a 20 MHz crystal oscillator.
- MCLR: Pulled up using a resistor for proper reset operation.

The Proteus schematic verifies correct pin mapping, power supply connections, and signal flow between sensors, the microcontroller, and output devices.

3. Well-Commented Source Code with Header Documentation

The firmware for this project was written in Embedded C using the XC8 compiler. The code follows a modular structure with clear separation between initialization routines, interrupt handling, LCD functions, and application logic.

3.1 Code Header

The code begins with a header section that specifies the file name, project description, and target microcontroller. Configuration bits are defined to set the oscillator mode, disable watchdog timer, enable power-up timer, and disable code protection.

3.1.1 Complete Circuit Schematic Diagram

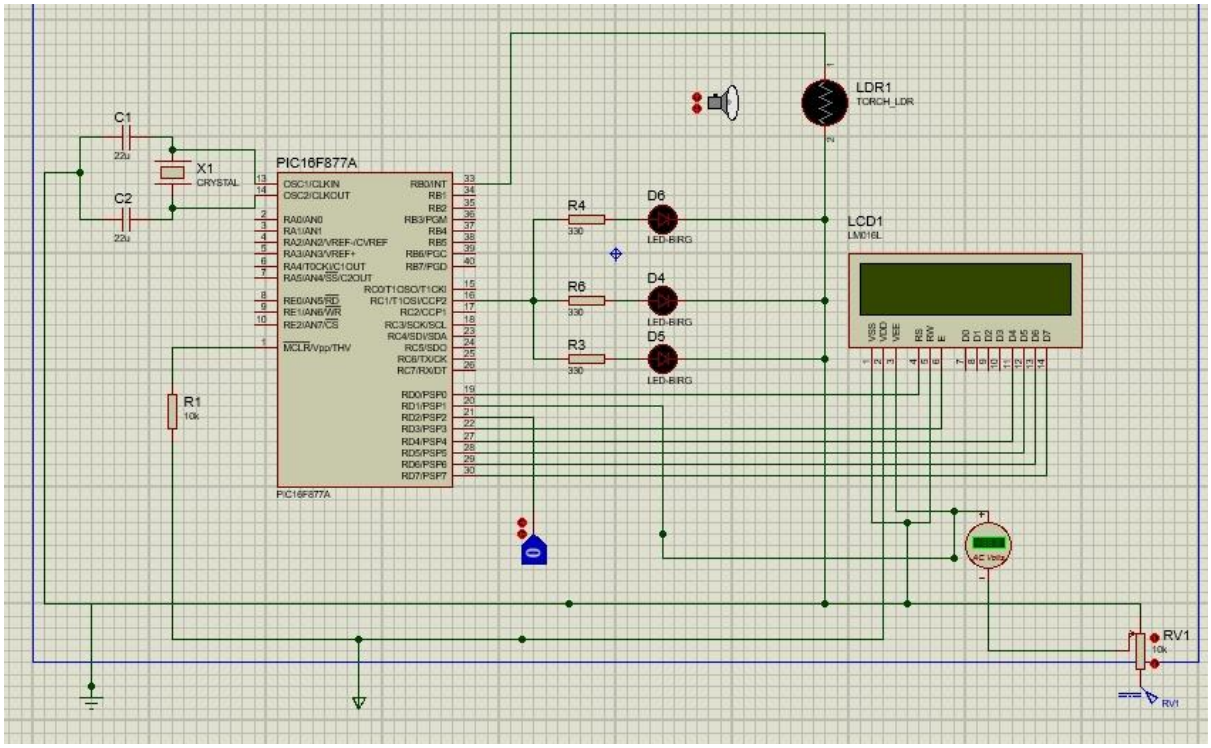

Figure 1: Proteus simulation circuit of the PIC16F877A system

3.2 Main Program Logic

The main() function initializes the system and then enters an infinite loop. Inside the loop:

- The LDR input is read to determine day or night.
- The PIR sensor is checked to detect motion.
- The LED (street light) is controlled based on sensor inputs.
- The LCD is updated to display the current system status.

4. Peripheral Initialization Documentation

4.1 I/O Port Initialization

- TRISB0 is configured as input for the LDR sensor.
- TRISD2 is configured as input for the PIR sensor.
- TRISC1 is configured as output for the LED.
- PORTD is configured for LCD data and control signals.

4.2 Timer2 Initialization

Timer2 is used to generate periodic interrupts that simulate real-time tracking (seconds, minutes, and hours).

- Prescaler is set to 1:16.
- PR2 register is set to 249.
- Timer2 overflow triggers an interrupt approximately every 0.8 ms.

4.3 Interrupt Configuration

- Timer2 interrupt is enabled using TMR2IE.
- Peripheral interrupts are enabled using PEIE.
- Global interrupts are enabled using GIE.

The interrupt service routine increments time variables without blocking the main program execution.

4.4 LCD Initialization

The LCD is initialized in 4-bit mode to reduce pin usage. Commands are sent in two nibbles (high nibble followed by low nibble). Proper delays are used to ensure reliable LCD operation.

5. System Block Diagram and Flowchart

5.1 System Block Diagram Description

The system can be divided into the following blocks:

1. Input Sensors: LDR and PIR sensors collect environmental data.
2. Microcontroller Unit: PIC16F877A processes sensor inputs and executes control logic.
3. Output Devices: LED (street light) and LCD display.

Data flows from sensors to the microcontroller, which then controls the outputs accordingly.

5.2 Flowchart Description

1. Start system
2. Initialize ports, timer, interrupts, and LCD
3. Read LDR value
4. If day → turn off light
5. If night → check PIR sensor
6. If motion detected → turn on light
7. Update LCD display
8. Repeat loop continuously

6. Testing Methodology and Results

6.1 Testing Methodology

The system was tested under multiple scenarios using Proteus simulation:

Test Case	LDR State	PIR State	Expected Output	Result
1	Day	No Motion	Light OFF	Pass
2	Night	No Motion	Light OFF	Pass
3	Night	Motion Detected	Light ON	Pass

6.2 Results and Observations

- The system correctly distinguishes between day and night conditions.
- Motion detection only affects the system during night mode.
- LCD accurately displays current system status.
- Timer and interrupt-based time tracking works reliably.

The simulation results confirm that the system operates as intended and meets all project requirements.

7. Conclusion

This project successfully demonstrates the design and implementation of an intelligent street light control system using the PIC16F877A microcontroller. By integrating sensors, timers, interrupts, and an LCD interface, the project showcases practical embedded system concepts taught in the Microprocessors course. The system is efficient, modular, and suitable for real-world applications.

8. References

- PIC16F877A Datasheet, Microchip Technology
- XC8 Compiler User Guide
- Embedded Systems course lecture notes