Home automation using 8051 microcontroller

Abstract

This project involves the design and implementation of a basic home automation system using the 8051 microcontroller. The system demonstrates the use of sensors and actuators to control home appliances automatically. The core functionality includes toggling a relay and controlling an LED based on input from a push button and a motion sensor.

Introduction

Home automation is an emerging field aimed at providing automated control of various household appliances, improving convenience, safety, and energy efficiency. The 8051 microcontroller, a widely used microcontroller in embedded systems, offers a suitable platform for developing such systems. This project leverages the 8051 microcontroller to create a simple yet functional home automation system.

Objectives

To design a home automation system using the 8051 microcontroller.

To control a relay and an LED based on input from a push button and a motion sensor.

To implement the system in assembly language for the 8051 microcontroller.

Components

8051 Microcontroller: The central unit for processing and controlling the system.

Relay Module: Controls high voltage appliances, connected to P1.0.

Push Button: User input for manual control, connected to P1.1.

Motion Sensor: Detects motion to control the LED, connected to P1.2.

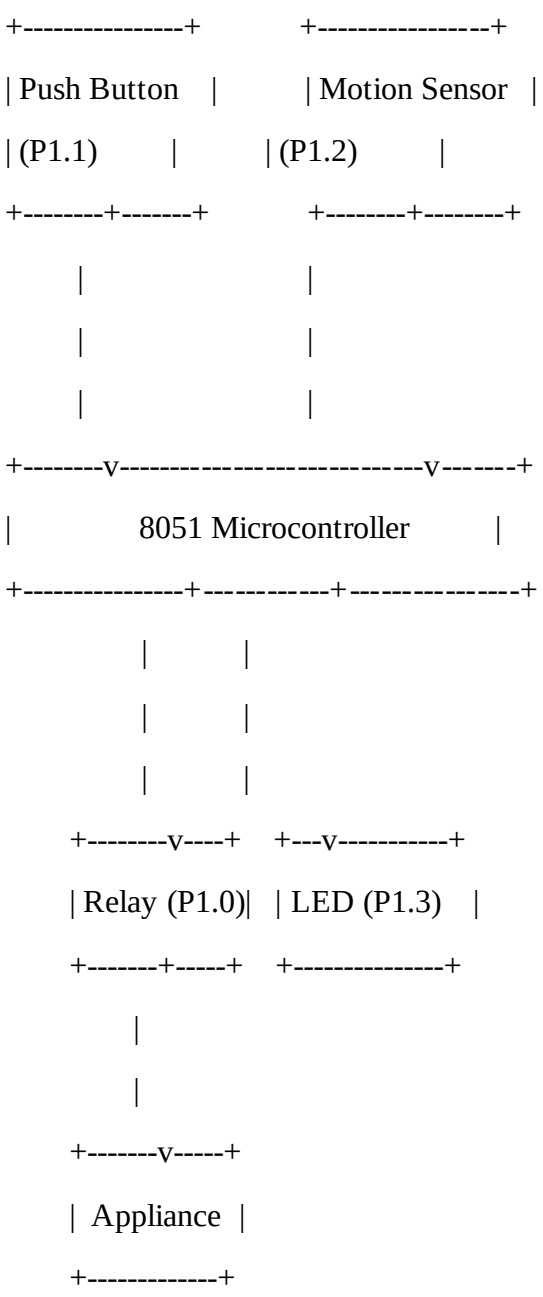
LED: Indicates motion detection, connected to P1.3.

Power Supply: Provides power to the system.

Connecting Wires: Used to connect the components.

Breadboard or PCB: For assembling the components.

Block diagram



C program

```
#include <reg51.h>

// Define ports
sbit relay = P1^0;       // Relay connected to P1.0
```

```
sbit button = P1^1;       // Push button connected to P1.1
sbit motionSensor = P1^2; // Motion sensor connected to P1.2
sbit tempSensor = P1^3;   // Temperature sensor connected to P1.3
sbit led = P1^4;          // LED connected to P1.4

void delay(unsigned int time) {
unsigned int i, j;
for(i=0; i<time; i++)
for(j=0; j<1275; j++);
}
void main() {
unsigned int tempValue;
while(1) {
if(button == 0) {    // If button is pressed
relay = ~relay; // Toggle relay state
while(button == 0); // Debounce delay
}
if(motionSensor == 1) { // If motion detected
led = 1;           // Turn on LED
}
 else
{
led = 0;           // Turn off LED
}
// Simulate temperature sensor value
tempValue = P0; // Assuming temp sensor is connected to Port 0
// Simple logic to control relay based on temperature
if(tempValue > 30) { // If temperature is higher than 30 degrees
```

relay = 1; // Turn on relay

}

else

{

relay = 0; // Turn off relay

}

delay(500);

}

}

```
#include <reg51.h>
sbit relay = P1^0;        // Relay connected to P1.0
sbit button = P1^1;       // Push button connected to P1.1
sbit motionSensor = P1^2; // Motion sensor connected to P1.2
sbit tempSensor = P1^3;   // Temperature sensor connected to P1.3
sbit led = P1^4;          // LED connected to P1.4

void delay(unsigned int time) {
    unsigned int i, j;
    for(i=0; i<time; i++)
        for(j=0; j<1275; j++);
}
void main() {
    unsigned int tempValue;
    while(1) {
        if(button == 0) {   // If button is pressed
            relay = ~relay; // Toggle relay state
            while(button == 0); // Debounce delay
        }

        if(motionSensor == 1) { // If motion detected
            led = 1;            // Turn on LED
        } else {
            led = 0;            // Turn off LED
        }

        // Simulate temperature sensor value
        tempValue = P0; // Assuming temp sensor is connected to Port 0

        // Simple logic to control relay based on temperature
        if(tempValue > 30) { // If temperature is higher than 30 degrees
            relay = 1;      // Turn on relay
        } else {
            relay = 0;      // Turn off relay
        }

        delay(500);
    }
}
```

Assembly language program/code

```
ORG 0000H          ; Origin, start address for code
MOV P1, #0FFH      ; Set P1 as input (except for P1.0 and P1.3)
MAIN:
; Check Push Button (P1.1)
JB P1.1, CHECK_MOTION    ; If button not pressed, jump to CHECK_MOTION
CPL P1.0                 ; Toggle Relay (connected to P1.0)
SJMP DEBOUNCE            ; Jump to debounce delay
CHECK_MOTION:            ; Check Motion Sensor (P1.2)
JB P1.2, MOTION_DETECTED ; If motion detected, jump to MOTION_DETECTED
CLR P1.3                 ; Turn off LED (connected to P1.3)
SJMP MAIN                ; Jump back to MAIN loop
MOTION_DETECTED:
SETB P1.3                ; Turn on LED (connected to P1.3)
SJMP MAIN                ; Jump back to MAIN loop
DEBOUNCE:
MOV R2, #50H             ; Load R2 with delay count
DELAY1:
MOV R1, #0FFH            ; Inner delay loop
DELAY2:
DJNZ R1, DELAY2          ; Decrement R1 until 0
DJNZ R2, DELAY1          ; Decrement R2 until 0
SJMP MAIN                ; Jump back to MAIN loop
 END
```

```
ORG 0000H           ; Origin, start address for code
MOV P1, #0FFH       ; Set P1 as input (except for P1.0 and P1.3)

MAIN:
    ; Check Push Button (P1.1)
    JB P1.1, CHECK_MOTION   ; If button not pressed, jump to CHECK_MOTION
    CPL P1.0                ; Toggle Relay (connected to P1.0)
    SJMP DEBOUNCE           ; Jump to debounce delay

CHECK_MOTION:
    ; Check Motion Sensor (P1.2)
    JB P1.2, MOTION_DETECTED ; If motion detected, jump to MOTION_DETECTED
    CLR P1.3                ; Turn off LED (connected to P1.3)
    SJMP MAIN               ; Jump back to MAIN loop

MOTION_DETECTED:
    SETB P1.3               ; Turn on LED (connected to P1.3)
    SJMP MAIN               ; Jump back to MAIN loop

DEBOUNCE:
    MOV R2, #50H            ; Load R2 with delay count
DELAY1:
    MOV R1, #0FFH           ; Inner delay loop
DELAY2:
    DJNZ R1, DELAY2         ; Decrement R1 until 0
    DJNZ R2, DELAY1         ; Decrement R2 until 0
    SJMP MAIN               ; Jump back to MAIN loop

END
```

Implementation

Hardware Setup: Connect the components as per the circuit diagram. Ensure the relay, push button, motion sensor, and LED are connected to the specified pins of the 8051 microcontroller.

Programming the Microcontroller: Use an assembler like Keil uVision to compile the assembly code and generate the hex file. Use a programmer to upload the hex file to the 8051 microcontroller.

Testing: Verify the functionality by pressing the push button and triggering the motion sensor. The relay should toggle with the button press, and the LED should turn on with motion detection.

Results

The system successfully demonstrated basic home automation functionalities:

The relay toggled state with each press of the push button.

The LED turned on when motion was detected by the motion sensor and turned off otherwise.

Conclusion

This project successfully implemented a simple home automation system using the 8051 microcontroller. The system showcased the capability of the 8051 microcontroller in controlling home appliances based on user input and sensor data. Further enhancements could include integrating more sensors, adding remote control capabilities, and expanding the system for comprehensive home automation.

Future Work

Remote Control: Incorporate Bluetooth or Wi-Fi modules to control appliances remotely via a smartphone app.

Additional Sensors: Integrate sensors such as humidity, light, and gas sensors for more advanced automation.

User Interface: Develop a graphical user interface using a touchscreen display or a mobile application for easier control.

References

Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin McKinlay. "The 8051 Microcontroller and Embedded Systems: Using Assembly and C". Pearson Education, 2007.

Keil uVision IDE documentation and user guides.