Princess Sumaya University for Technology
King Abdullah II Faculty of Engineering
Electrical Engineering Department

Embedded systems
Final Design Project
Smart Home System

Authors (Group 4):

Omar Al-Refai	20200304	Computer Engineering	oma20200304@std.psut.edu.jo
Aya Al Abed	20200817	Computer Engineering	aya20200817@std.psut.edu.jo
Rula Sufian	20200495	Computer Engineering	rul20200495@std.psut.edu.jo

Supervisor:
Dr. Belal Sababha

January 19, 2024

Abstract

This project presents a smart home embedded system designed around the PIC16F877A Microcontroller Unit (MCU) to serve as the computational and control hub. The system integrates various connected devices to automate and elevate both home security and environmental experiences. The core functionalities include a stepper motor responsible for managing movable components of a window shutter. A light sensor is employed to regulate the shutter position based on detected light levels, accompanied by the activation of an LED when the shutter is lowered. Additionally, a temperature sensor interfaces with a fan connected to a DC motor, allowing for real-time monitoring and adjustment of home temperature to optimize comfort.

The security aspect of the system is demonstrated through the implementation of an Infrared (IR) sensor placed on the window. This sensor is designed to identify potential intrusions, triggering a buzzer to promptly alert homeowners of security breaches. By seamlessly combining these elements, the smart home embedded system showcases a comprehensive approach to home automation, integrating security features and environmental controls for an enhanced and intelligent living environment.

TABLE OF CONTENTS

1 INTRODUCTION

In an era marked by technological innovation, the evolution of smart homes has emerged as a transformative paradigm for modern living. This project explores the realm of home automation, emphasizing the integration of a PIC16F877A Microcontroller Unit (MCU) as the central processing unit orchestrating the interaction of diverse devices. The resulting smart home embedded system goes beyond conventional boundaries, aiming to enhance both security and environmental control.

This project demonstrates how smart home systems can meet the changing needs of users. By combining security features and environmental controls smoothly, it shows how technology can transform homes to be adaptable, safe, and in tune with the preferences of occupants.

1.1 BACKGROUND

The PIC16F877A microcontroller, which is the center of this project, is made by Microchip Technology and is a versatile 8-bit component widely used in embedded systems. It operates at speeds up to 20 MHz and comes with built-in features like pins, converters, timers, and communication tools. An 8 MHz clock will be used in this project. This microcontroller is known for its simplicity, reliability, and ease of reprogramming, making it suitable for various projects, from educational to industrial. Its straightforward design makes it a popular choice for both beginners and experienced developers in the world of embedded systems.

Several sensors and actuators are used in the project. The first actuator used is a servo motor, which is a compact and versatile electric motor designed for precise control of angular or linear position, velocity, and acceleration. Servo motors operate in a closed-loop system, utilizing feedback from an encoder or potentiometer to continually adjust and maintain the desired position. Servo motors can be considered as analog devices in the sense that they operate based on continuous control signals. The control input to a servo motor is typically a pulse-width modulation (PWM) signal. The width of the pulse determines the desired position of the servo motor.

Another actuator used in this project is a DC motor. A DC (Direct Current) motor is a widely used electrical device that converts electrical energy into mechanical motion through the interaction of magnetic fields. It operates on the principle of electromagnetism. The rotating DC motor is a vital component in small fans, providing a reliable and efficient means of generating airflow for cooling purposes, as will be used in the project. The DC motor needs a voltage that our MCU cannot sustain, also this voltage needs to be varied with our temperature values. Therefore, we will need to connect it to a H-Bridge to solve this problem. An H-bridge is commonly connected to a DC motor to facilitate bidirectional control of its rotation. However, we will be only using one direction since it is basically a fan. The H-bridge consists of four switches arranged in the shape of the letter 'H,' with each switch controlling a specific connection to the motor. By selectively opening and closing these switches, the polarity of the voltage applied to the motor terminals can be manipulated, determining the direction of rotation. This versatility is crucial in applications such as robotics, where precise and reversible motor movements are essential. The H-bridge configuration provides an efficient means of motor control, enhancing flexibility with a 12V power supply. We settled on the L298 H-Bridge for its popularity.

Moving on to the sensors, the first sensor used is an LDR (Light Dependent Resistor) sensor, which are passive electronic devices that exhibit a change in resistance in response to variations in light levels. This property makes them valuable components in various applications where the detection or measurement of light is essential. LDR sensors are analog sensor. The resistance of an LDR changes continuously in response to varying light levels. As the incident light increases, the resistance of the LDR decreases, and as the light decreases, the resistance increases. This analog relationship allows the LDR to provide a continuous range of resistance values corresponding to the intensity of light it is exposed to.

The second sensor used is the IR (Infrared) sensor, which detects and responds to infrared radiation. There are different types of IR sensors, including passive infrared (PIR) sensors, which detect changes in heat, and active infrared sensors, which emit and receive infrared radiation. PIR sensors are often used in motion detection or object detection. While IR sensors are highly useful, they have limitations, such as sensitivity to environmental factors like temperature changes and ambient light. Additionally, their range and effectiveness may vary based on the specific application. IR sensors can be digital or analog, where the sensor used in the project is digital. IR sensors that operate in digital mode, as used in this project, produce binary or discrete output signals. Common examples include digital proximity sensors or motion detectors that output a digital signal (high or low) depending on the presence or absence of an object or motion. Digital IR sensors are often used in applications where a simple on/off response is sufficient.

In addition, there are different types of temperature sensors used in circuits. A common temperature sensor used in embedded systems is LM35. The LM35 is a precision analog temperature sensor widely used for measuring temperature in electronic circuits and various applications. Developed by National Semiconductor (now part of Texas Instruments), the LM35 has become a popular choice due to its accuracy, simplicity, and ease of integration.

2 DESIGN

The design of the smart home was revolved around three different parts. The first part was the IR sensor which had a digital interfaced with the microcontroller and was used with the buzzer. The second part was the LDR sensor that was used with the servo to control opening and closing the shutter with three different angles depending on the light intensity. The third part was the temperature sensor which was used to control the speed of the fan that was attached to the DC motor. An edge bridge was also used to control the speed of the motor depending on the range of temperature inside the smart home. Lastly, an LED was also connected which turns on when the shutter is fully closed, and turns off when it is fully opened. The following table shows the list of the components used with their costs:

Table 1 Design components with cost

Component	Cost / JD
PIC16F877A microcontroller	7.50
IR sensor	2.00
5V magnetic buzzer	0.45
LDR sensor	1.40
MG90s Servo motor	4.00
LM35 temperature sensor	4.00
L298 H-bridge	2.75
DC motor with fan	1.70
LED	0.05
Resistors	1.00
Crystal Oscillator 8MHz	0.45
Breadboard (2)	3.80
40 wires Female to Male	1.50
40 wires Male to Male	1.50
Total Cost	32.10

In the following sections, the software, electrical and mechanical aspects of the design will be thoroughly explained and analyzed.

2.1 SOFTWARE

The first step of the design process was writing the code which was downloaded on the microcontroller and tested on the components. The code was written on MicroC Pro for PIC program. The code was written in C programming language.

2.1.1 Initialization and IR sensor interfacing with buzzer

In the beginning of the code, all variables for every part of the code were initialized as well as the functions used.

Afterwards, the code checks for an external interrupt triggered by an IR sensor, specifically monitoring the second bit (0x02) of the INTCON register. Upon detection of the interrupt, it enters a loop that iterates five times, activating a buzzer connected to RB7 on each iteration.

The activation is achieved by setting and clearing the most significant bit of the PORTB register, creating a buzzing effect with short delays in between. After the loop, the external interrupt flag (INTF) is cleared, acknowledging the interrupt and preparing the system for potential subsequent interrupts.

2.1.2 LDR sensor with Servo Motor

Then, the code includes the interrupt service routine (ISR) for the Timer0 overflow interrupt on a PIC16F877A microcontroller, indicating it will be executed approximately every 1ms. Inside the ISR, the Timer0 register (TMR0) is set to 248, and counters (Mcntr and Dcntr) are incremented. If Dcntr reaches 500 (after 500ms), the program initializes and reads an LDR sensor using the ADT_init_LDR and LDR_Servo functions, respectively. Based on the obtained voltage value, it triggers specific actions related to servo motor control and updates the current_state variable accordingly. The code also modifies the state of PORTD0, where the LED is connected, based on the servo motor's angle. Each angle for the servo motor has its own function. Finally, the Timer0 interrupt flag (T0IF) is cleared, preparing the system for the next Timer0 overflow interrupt. This code segment appears to manage the periodic execution of tasks related to an LDR sensor and servo motor control at a 1ms interval on a PIC16F877A microcontroller.

The ATD_init_LDR function initializes the analog to digital conversion for LDR readings, configuring it to operate on Channel 1 (RA1) with a clock source of Fosc/16. It sets up the ADCON1 register for all channels to be analog, sets the conversion clock to 500 KHz, and configures the result to be right-justified. The ATD_read_LDR function initiates an ADC conversion by setting the appropriate bit in ADCON0, waits for the conversion to complete using a while loop, and then reads the converted analog value from the ADC result registers (ADRESH and ADRESL). The result is combined into a 16-bit variable and returned. These functions facilitate the acquisition of analog readings from the LDR sensor for further processing in the main program.

The functions, Rotation0, Rotation45, and Rotation90, control the rotation of a servo motor connected to PORTB on a PIC microcontroller. Each function iterates 50 times, generating pulses by toggling PORTB's fifth bit (RB5) to produce pulses of specified durations using

Delay_us functions. The angles correspond to 0, 45, and 90 degrees for Rotation0, Rotation45, and Rotation90, respectively. These functions facilitate precise servo motor control for applications requiring specific rotational positions

2.1.3 Main function and DC motor with Temperature sensor

n the main function, it initializes the port directions and values, as well as various counters and timers. It sets up interrupts and enters a continuous loop where it reads temperature using a temperature sensor, adjusts the speed of a DC motor based on the temperature reading, and controls a PWM signal to drive the motor. The temperature readings are obtained through analog-to-digital conversion, and the motor speed is adjusted according to different temperature ranges. The program uses external interrupts, analog-to-digital conversion, and pulse-width modulation to manage the DC motor's behavior in response to temperature variations.

The code configures the microcontroller to enable global interrupts (GIE) Timer0 overflow interrupts (T0IF) and external interrupts (INTE).

The program enters a continuous loop (while(1)) where it reads temperature using the read_temperature function and adjusts the speed of a DC motor accordingly using the PWM function.

The read_temperature function initiates and reads the analog temperature sensor (TMP) on Channel 3 (RA3) using analog-to-digital conversion. The temperature is then used to determine the speed of the DC motor and set specific values for the PORTC register for visual indication.

The program maps different temperature ranges to corresponding motor speeds. If the temperature is above 40.0°C, the motor runs at full speed (99), with a visual indicator on PORTC. In other temperature ranges, different speed values are set, and PORTC is updated accordingly.

The functions ATD_init_TMP and ATD_read_TMP handle the initialization and reading of the analog temperature sensor using the Analog-to-Digital Converter (ADC) module.

The PWM function generates a PWM signal for controlling the speed of the DC motor. It takes parameters for the period (p) and duty cycle (d). The signal is output on PORTB, specifically RB4.

2.1.4 Flowchart

The code can be summarized in a flowchart as follows:

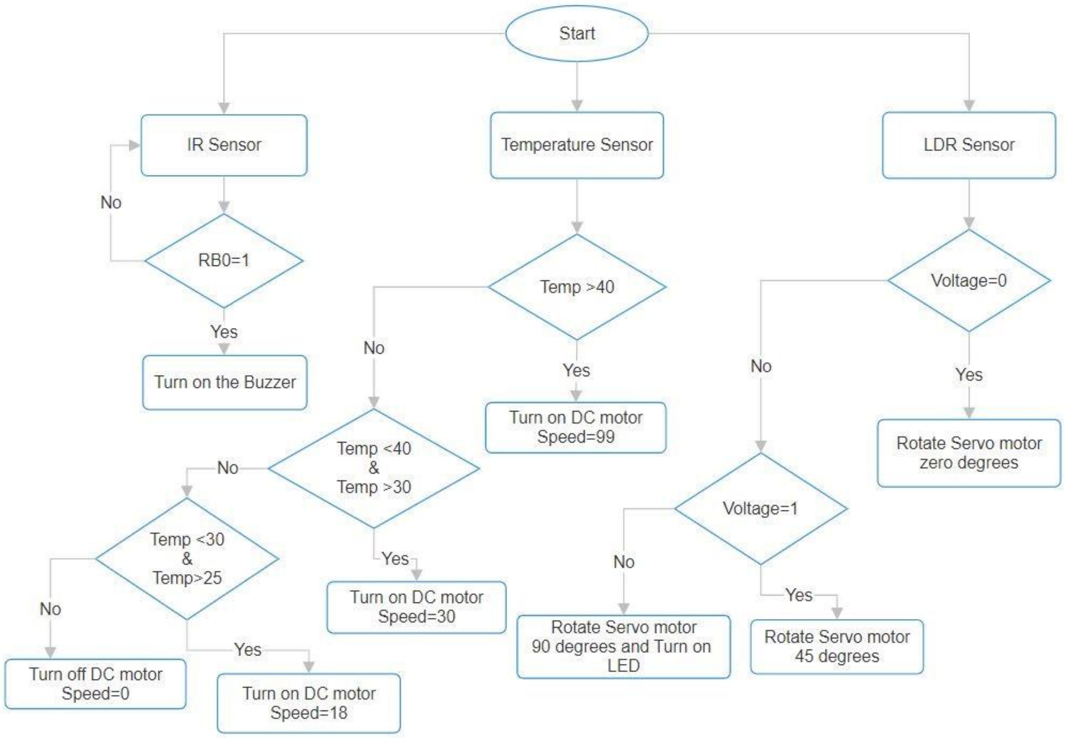

Figure 1 Flowchart for the Smart home design.

2.2 ELECTRICAL

The electrical configuration for the smart home system is outlined as follows:

- 5 volts connected to the 16F877A PIC, sourced from the power supply.
- Ground connected to the 16F877A PIC.
- 8MHz clock connected to the 16F877A PIC.
- MCLR connected to 5V.
- IR sensor wiring:
  - VCC connected to 5V.
  - Ground connected to ground.
  - Output connected to RB0.

- Buzzer wiring:
    - Anode connected to RB7.
    - Cathode connected to ground.
- LDR sensor wiring:
    - Analog output connected to RA1.
    - VCC connected to 5V.
    - Ground connected to ground.
- Servo wiring:
    - Input connected to RB5.
    - VCC connected to 5V.
    - Ground connected to ground.
- DC motor connected to outputs (OUT1 and OUT2) of the H-bridge.
- Inputs of the H-bridge:
    - One connected to ground.
    - One connected to 5V.
- Enable connected to RB4.
- 12V input from the power supply for the H-bridge.
- Ground connected to ground for the H-bridge.
- Temperature sensor wiring:
    - Output connected to RA3.
    - VCC connected to 5V.
    - Ground connected to ground.
- Common ground ensured for all components.
- LED:
    - 330-ohm resistance connected.
    - Connected to port D0.

A block diagram was constructed to show the connections as follows:

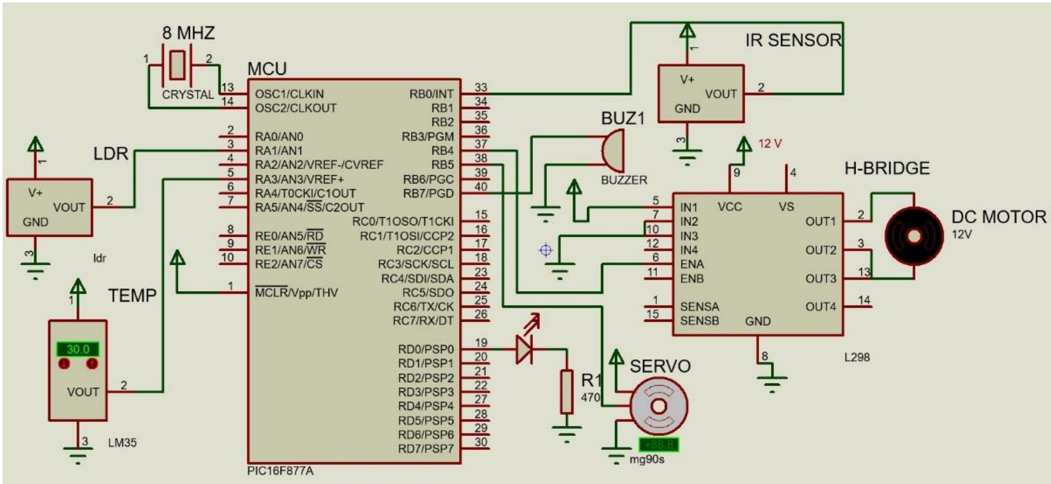

Figure 2 Block diagram for the design

2.3 MECHANICAL

The mechanical aspect of the design was relatively straightforward, utilizing cardboards for both the floor and walls of the smart home model. Two windows were incorporated, one designated for the shutter mechanism, and the other for the IR sensor. The shutter itself was crafted from lightweight Balsa wood, securely attached to the servo arm to ensure proper functionality. The buzzer and LED were affixed to the top of the structure, and a designated opening was created to accommodate the DC motor with the fan. The breadboard containing all the electrical connections was strategically positioned at the back of the model for accessibility and organization. Figures 3 and 4 show the smart home model with all the connections.

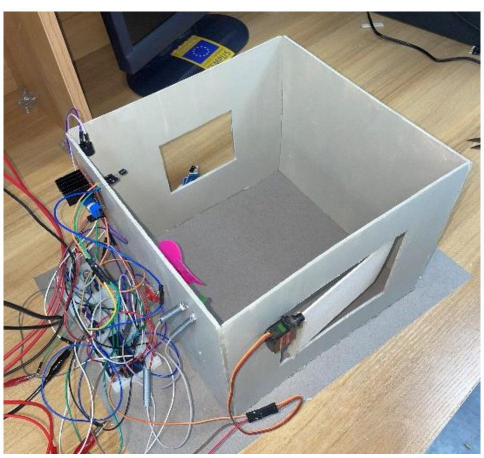

Figure 4 Smart home model from one perspective

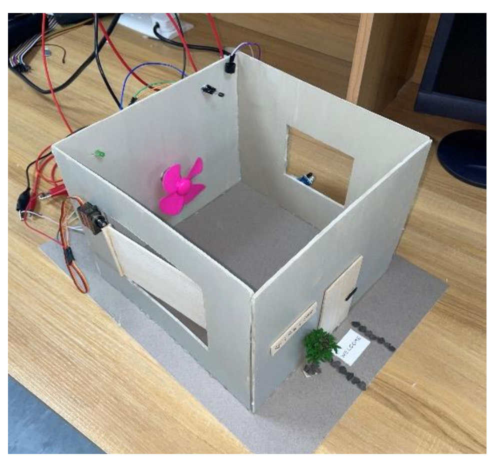

Figure 3 Smart home model from another perspective

3 PROBLEMS AND RECOMMENDATIONS

3.1 PROBLEMS

Several challenges were encountered during the design process. Firstly, there were issues with the durability of the PIC pins, leading to breakage and necessitating welding for restoration. Additionally, inconsistencies were observed in the accuracy of the temperature sensors, with some purchased units displaying oscillating and unstable readings during testing. Furthermore, there were some difficulties in the malfunctioning of the kit utilized for code downloading and microcontroller testing, initially causing confusion about the source of coding issues. After thorough investigation, it was determined that the testing kit itself was problematic, not the code. Lastly, certain power supplies exhibited functionality issues, displaying instability and impacting the overall reliability of the system. These challenges required troubleshooting and problem-solving to ensure the successful implementation of the smart home design.

3.2 RECOMMENDATIONS

To address the encountered issues, several recommendations are provided. First and foremost, it is advised to take extreme care when dealing with the PIC to prevent any potential breakage or damage to its pins. Additionally, opting for high-quality sensors is recommended to ensure accurate and reliable readings, ultimately saving time and avoiding inconsistencies. When conducting tests, it is suggested to directly use the sensors on the breadboard instead of relying solely on the kit, enhancing the precision of troubleshooting efforts. Lastly, before initiating the project, it is crucial to verify the stability and functionality of the power supply to guarantee uninterrupted operation throughout the development process. These recommendations aim to enhance the overall reliability and efficiency of the smart home design

4 CONCLUSION

In conclusion, the design of the smart home system successfully integrated both electrical and mechanical components to create a responsive and efficient model, despite encountering challenges. The collaborative effort in addressing these issues resulted in a functioning smart home model that effectively demonstrated automation features, combining security measures with environmental controls. Overall, the project's success reflects the team's resilience and problem-solving skills in achieving a seamless integration of technology for a smart and responsive home environment.