Princess Sumaya University for Technology

King Abdullah II Faculty of Engineering

Computer Engineering Department

Microprocessors and Embedded Systems

Project Report:

DeepSleep AlertClock

Student Name	***Student ID***	***Major***	***Email***
Jude Marrar	20210695	NIS Engineering	jud20210695@std.psut.ed.jo
Selen Qarajah	20210622	NIS Engineering	sel20210622@std.psut.ed.jo
Layan Al-Himsi	20200295	Computer Engineering	lay20200295@std.psut.edu.jo

Table of Contents

List Of Figures

Abstract

This report presents the design and development of the DeepSleep AlertClock, a smart alarm clock specifically designed for heavy sleepers, using the PIC16F877A microcontroller. Users can enter their desired wake-up time, and the clock employs a dual-buzzer system to ensure effective waking of the users, this involves turning on the lights and engaging in physical activity as part of the process to deactivate the alarm.This report details the mechanical, electrical, and software components that automate the DeepSleep AlertClock and discusses the design process, challenges encountered, and recommendations for future improvements.

Introduction And Background

The DeepSleep AlertClock project was born out of a simple yet profound insight which is that traditional alarm clocks often fail to wake up heavy sleepers effectively. Recognizing this challenge, we set out to design a sophisticated solution that ensures users wake up fully and on time.

Drawing upon our expertise in embedded system design, we conceived the DeepSleep AlertClock—a user-friendly alarm system with innovative features like a dual-buzzer mechanism and physical interaction requirements which includes getting out of bed by first turning the lights ON and then running to catch the robot and deactivating the alarm. Our aim is to revolutionize the wake-up experience for heavy sleepers, offering a seamless morning routine and improved productivity.

Design

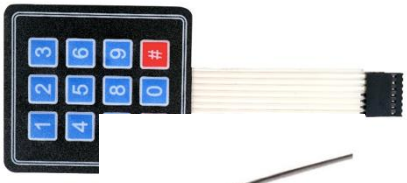
Mechanical Design:

As shown in Fig our project included the following components:

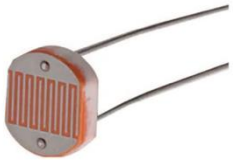
1. **Buzzer:**
   Buzzers are electromechanical devices that produce sound or audible alerts. For our project, we will be using two active buzzers for auditory alarm. The first buzzer, will turn off when the light intensity is increased (indicating room lights are ON) and the second will stay active until the alarm is deactivated.

2. **LCD Screen:** are used for displaying information in various electronic devices. For our project, we will be using it to display the timer for setting the alarm.

3. **Keypad:** A keypad is an input device that consists of a set of buttons arranged in a matrix pattern. For our project, we used the 4x3 keypad for simple input of the required time to set the alarm.



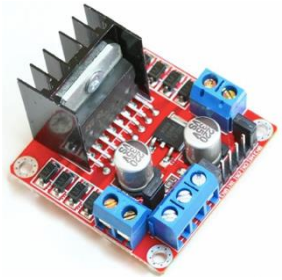
4. **LDR Sensor:** are tiny light-sensing devices also known as photoresistors. An LDR is a resistor whose resistance changes as the amount of light



falling on it changes, which was be implemented in our project for detecting the light intensity in the room and turning off the (light) buzzer.

5. **DC motors:** motor that turns energy from a direct current and turns this into mechanical energy. We will be using 2 motors for all four directions (front-right, front-left , backwards-right and backwards-left) this achieved by an H-bridge.

6. **H-bridge:** electronic circuit that switches the polarity of a voltage applied to a load. In our project, we connected the H-bridge to enable both our motors and allow them to steer in both directions.



7. **Push button: are** switches, electrical actuators , that when pressed, either close or open an electrical circuit. It will be implemented in such a way that when pressed it will deactivate the alarm.

8. **IR sensor:** is a radiation-sensitive optoelectronic component with a spectral sensitivity in the infrared wavelength range 780 nm. We used IR sensor to detect the distance using light waves and make our robot move in a different direction when facing a near obstacle.

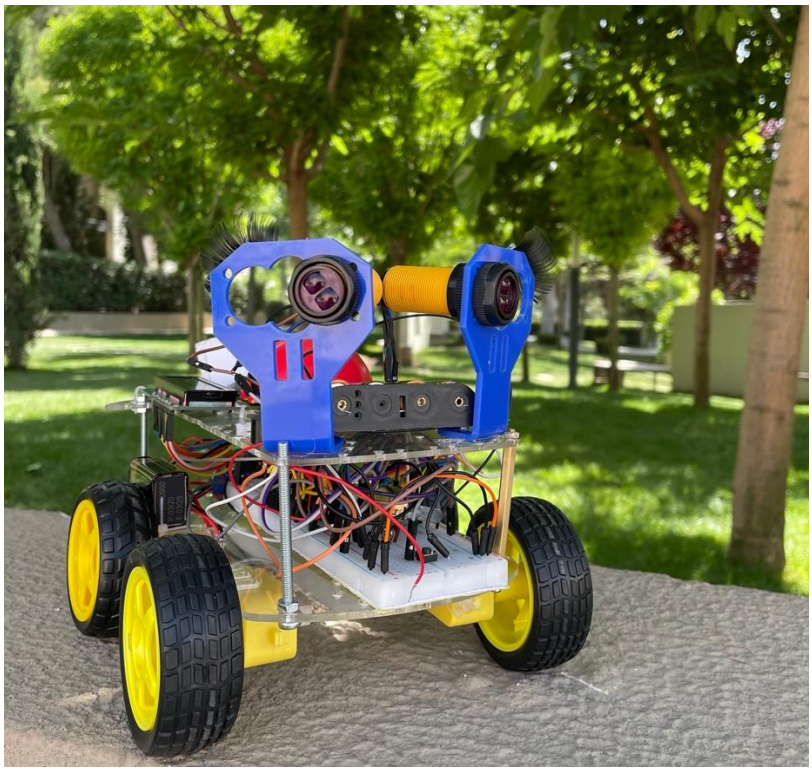


Figure 1: Final Design Front Face

Figure 2: Mechanical Design from the Side

Electrical Design:

Here is a comprehensive depiction of the electrical connections in our circuit utilizing the PIC 16F877A on Proteus.

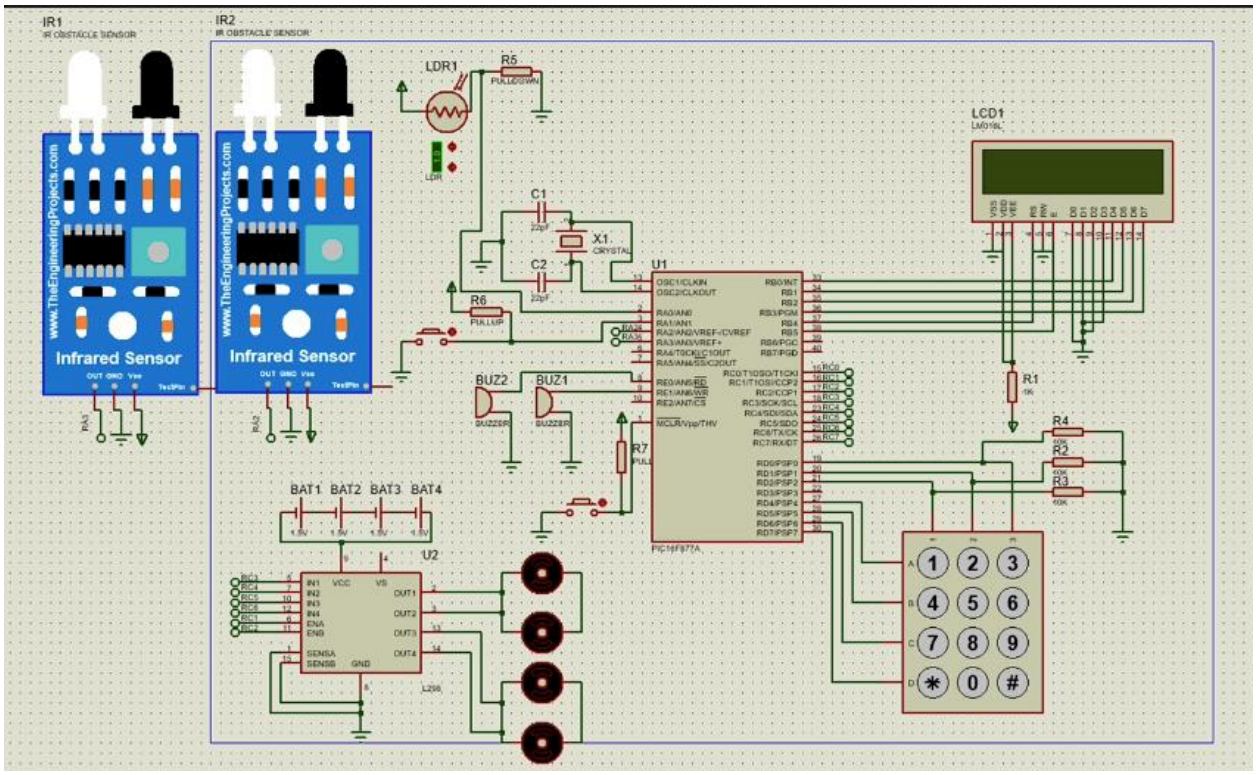


Figure 3: Circuit Design

Pin Connection

Pin	Port	Use	Connection
1	MCLR’	Mater Clear	Push Button
11,32	V_{DD}	High Voltage Source	5V
12,31	V_{ss}	Low Voltage Source	0V
13,14	OSC1,OSC2	Oscillator	8MHZ Crystal Oscillator
2	RA0	LDR Sensor	Connected with LDR
3	RA1	Push Button	Push Button Output Pin
4	RA2	IR Obstacle	IR Obstacle Output
5	RA3	IR Obstacle	IR Obstacle Output
8	RE0	Buzzer	Buzzer Output
9	RE1	Buzzer	Buzzer Output
16	RC1	ENA	H-Bridge EnableA
17	RC2	ENB	H-Bridge EnableB
18	RC3	IN1	H-Bridge IN1
23	RC4	IN2	H-Bridge IN2
24	RC5	IN3	H-Bridge IN3
25	RC6	IN4	H-Bridge IN4
19	RD0	Keypad	Keypad Col1
20	RD1	Keypad	Keypad Col2
21	RD2	Keypad	Keypad Col3
27	RD4	Keypad	Keypad Row1
28	RD5	Keypad	Keypad Row2
29	RD6	Keypad	Keypad Row3
30	RD7	Keypad	Keypad Row4
33	RB0	LCD	LCD D4 Pin
34	RB1	LCD	LCD D5 Pin
35	RB2	LCD	LCD D6 Pin
36	RB3	LCD	LCD D7 Pin
37	RB4	LCD	LCD D4 Pin
38	RB5	LCD	LCD D5 Pin

Software Design:

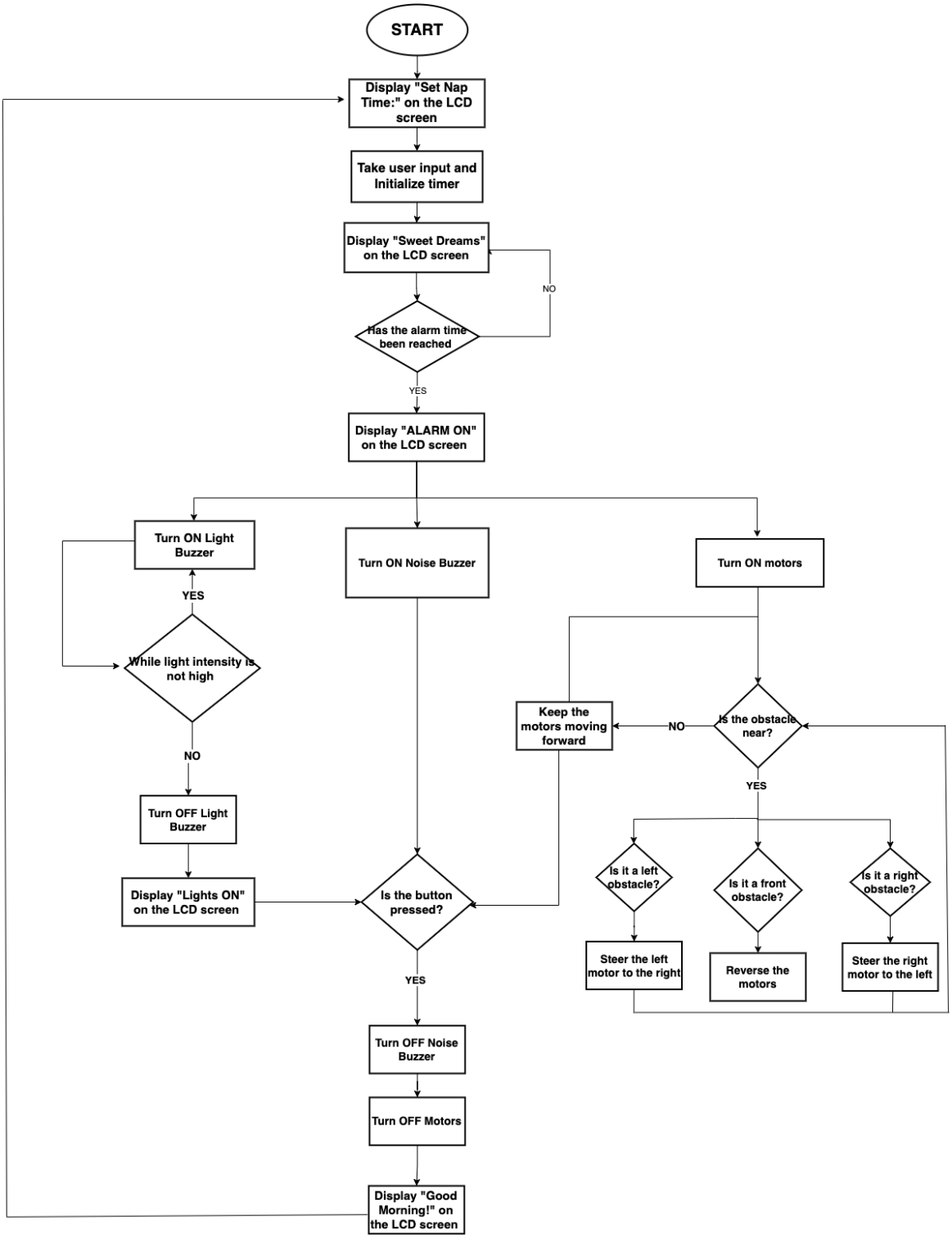


Figure 4: Software Design

Problems and Recommendations

While working on this project, our team faced many complications, some that we managed to solve, and some that we unfortunately were unable to overcome. Some of these problems are:

- Software Problem:

Initially, we intended to utilize an ultrasonic sensor to detect nearby obstacles, and thus, we introduced an "init_sonar" and "read_sonar" function. However, upon integrating an additional interrupt into our existing code, we encountered a setback: the added code alone consumed nearly 2K EEPROM program words, nearing the limit imposed by the free version of MikroC Pro (2K program words). Consequently, we were compelled to optimize our code by reducing redundant commands and function calls. One notable modification involved revising the "Scan_keypad()" function, sourced from the library, which originally employed switch cases. Instead, we simplified it into a direct equation, namely "kp = (kp % 14) + 48 - ((kp % 14) / 4)," facilitating the direct calculation of the ASCII value of the entered key.
Before:

```
unsigned char Scan_Keypad(void){
    kp=0;  // Reset key code variable
    // Wait for key to be pressed and released
    do
      // kp = Keypad_Key_Press();          // Store key code in kp variable
      kp = Keypad_Key_Click();             // Store key code in kp variable
    while (!kp);
   // Prepare value for output, transform key to it's ASCII value
    switch (kp) {
      case  1: kp = 49; break; // 1
      case  2: kp = 50; break; // 2
      case  3: kp = 51; break; // 3

      case  5: kp = 52; break; // 4
      case  6: kp = 53; break; // 5
      case  7: kp = 54; break; // 6

      case  9: kp = 55; break; // 7
      case 10: kp = 56; break; // 8
      case 11: kp = 57; break; // 9

      case 13: kp = 42; break; // *
      case 14: kp = 48; break; // 0
      case 15: kp = 35; break; // #

    }
    return kp;
}
```

After:

```
unsigned char Scan_Keypad(void){
    kp=0;  // Reset key code variable
    // Wait for key to be pressed and released
    do
      // kp = Keypad_Key_Press();          // Store key code in kp variable
      kp = Keypad_Key_Click();             // Store key code in kp variable
    while (!kp);
   // Prepare value for output, transform key to it's ASCII value
    /*switch (kp) {
      case  1: kp = 49; break; // 1
      case  2: kp = 50; break; // 2
      case  3: kp = 51; break; // 3

      case  5: kp = 52; break; // 4
      case  6: kp = 53; break; // 5
      case  7: kp = 54; break; // 6

      case  9: kp = 55; break; // 7
      case 10: kp = 56; break; // 8
      case 11: kp = 57; break; // 9

      case 13: kp = 42; break; // *       --> invlaid input when setting the time
      case 14: kp = 48; break; // 0
      case 15: kp = 35; break; // #       --> will be used as enter

    }*/
    if (kp==15){     //case '#' can't be handeled using the equation
     return 35;
    }

    kp=(kp%14)+48-((kp%14)/4);
    return kp;
}
```

```
Used RAM (bytes): 303 (86%)  Free RAM (bytes): 49 (14%)
Used ROM (program words): 2039 (25%)  Free ROM (program words): 6153 (75%)
```

- Mechanical Problem:

Another setback was also due to usage of the ultrasonic sensor which when initiated used to set the code into an infinite loop due to its inability to detect nearby objects therefore providing a constant zero reading, after a lot of code tracing, trial and errors and research we have come to the conclusion that the ultrasonic measurements is not really accurate and has the big flaw, that it depends on the reflectivity of the object. Soft objects can dampen the reflection significantly until you don't get a reliable measurement anymore so we have decided to move on with an IR obstacle sensor instead.

- Connecting on Breadboard:

During the transition from the MikroE Kit to the breadboard, we faced several challenges. A significant issue arose with the LCD screen, which failed to display the intended text despite proper wiring. After troubleshooting, we discovered that the resistor used for LCD contrast was too low. We replaced it with a potentiometer, which resolved the problem and provided flexibility to adjust the contrast. Additionally, the keypad didn't show any numbers on the LCD and after extensive research and step-by-step disconnection of the keypad pins, we identified the problem with pins RD0-RD2, which represent the keypad columns. To resolve the issue, we connected each of these pins to a pull-down resistor (1KΩ), effectively solving the problem.

Conclusion

In today's fast-paced world, every minute counts. The value of time is paramount, and keeping appointments requires unwavering discipline. To combat the challenges of a busy schedule and ensure we wake up refreshed and ready to tackle the day. This inspired the creation of the DeepSleep AlertClock, a device designed to help individuals wake up promptly and stay on schedule to enhances productivity, time management, and reinforcing the importance of discipline in achieving one's goals.

Through this project, we aim to encourage the sleeper to take physical action upon waking and accomplish their planned tasks.