Project 3: Climate Monitor System

Prepared For: Dr. Jennifer Winikus
CSE 321 Realtime Embedded Systems
Department of Computer Science & Engineering
School of Engineering and Applied Sciences
University at Buffalo

Prepared By: Brett Sitzman, Junior
Fall 2021 Semester
Department of Computer Science and Engineering
School of Engineering and Applied Sciences
University at Buffalo

Introduction

For project 3 the class was asked to create a real-time embedded system that can help solve a real-world problem. This project seeks to tackle food waste minimization by increasing the efficiency of indoor growing. To increase the efficiency of growing, this system monitors climate variables that impact crop growth.

The device can be set to monitor a user-inputted temperature and humidity range. When it detects a value outside of those ranges, it will sound an alarm. This alarm consists of a notification on the LCD, ringing the buzzer, and flashing the LED. The alarm will not stop until the user ends it. The device also has the capability to report the current climate information without monitoring any ranges.

This report covers the different elements that were a part of the design process for the climate monitoring system. The report begins with an overview of the features and specifications. Then, it explains how the required internal features are integrated. Next, it goes into the steps of the design process. After that, there is a block diagram to provide a general overview of the system, and a flow chart to show the control flow. Then, there is a user instructions section which contains a schematic, instructions on how to build the system, and instructions on how to use the system. The report ends with test plan instructions and a timeline of the project's development. At the back of the report, the references used to make this document are listed.

Features & Specifications

There are 4 modes the system can be in:

- IDLE: display current climate information on LCD
- INPUT: allow user to enter temperature and humidity range
- MONITOR: display current climate information on LCD; but trigger alert mode if the current climate information is outside of the user-inputted range
- ALERT: sound an alarm, flash an LED, and display a notification on the LCD

Specifications:

1. Can measure humidity in range 20%-95%
2. Humidity readings have an error range of ±5%
3. Can measure temperature in range 0-50℃
4. Temperature readings have an error range of ±2℃

Inputs

Keypad:

- A – Enter input value, switches device to MONITOR mode once all values are gathered
- B – Switch to IDLE mode
- C – clear current input entry (only when inputting value)
- D – Switch to INPUT Mode
- 0-9 – Enter value for temperature/humidity range
- There is no programmed functionality for the '#' and '*' keys

DHT11:

- Sensor detects temperature & humidity data

Outputs

LCD:

- Prompts user for input (minimum temperature, maximum temperature, minimum humidity, maximum humidity)
- Alert user to invalid inputs
- Displays current climate information read from the DHT11 sensor
- Displays notification when the device detects the climate is out of range

Buzzer:

- Ring in 1 second intervals when the device detects a climate out of range

LED:

- Flash in 1 second intervals when the device detects a climate out of range

Required Internal Features

Watchdog	The watchdog is started inside the main function. It is fed when we poll the keypad for input.
Synchronization Technique	In the ISRs a flag named input_modifed is set; its purpose is to tell the program the user has inputted a value. The ISR also updates an internal variable called input_str, which is the critical resource. When the program detects that input_modified is true, it knows to print the updated input_str to the LCD.
Bitwise Driver Control	The program uses bitwise driver control for the keypad interactions. Port F is enabled, and it polls through pins PF_12, PF_13, PF_14, and PF_15, which are connected to the rows.
Critical Section Protection	As described in number 2 (synchronization technique), I used a flag to protect the critical resource. The critical resource was the user input, named input_str in the project code. The flag prevented the program from printing the user value multiple times when it hasn't been changed.
Threads	There are 2 threads intentionally incorporated (not the main function or ISRs). The first thread (named t_lcd) is used to update the LCD with the latest information, once a second. This thread runs forever. The other thread (named t_monitor) is started when the device enters monitor mode and runs until it detects the climate is out of range, in which case it enters alert mode.
Interrupts	Interrupts are triggered when the user presses one of the buttons on the keypad. Each one represents a column in the matrix. When used with the bitwise driver control, we can determine the exact row and column corresponding to the button pressed.

Design Process

Ask	The first stage of the design process is where the project purpose was determined. It began by imagining the possible scenarios where the allowed peripherals would be useful. For this project it was decided to include the DHT11 temperature & humidity sensor and the buzzer module to develop some sort of climate monitoring system that can be used for indoor growing centers.
Research	The research stage began by examining the data sheets for both the DHT11 and buzzer. Once that information was clear, the DHT11 provided code was briefly scanned to get an idea on how it can be incorporated into the program.
Imagine	For the imagine step, the previously used peripheral requirement was brought into consideration. For this project, multiple of the previously used peripherals would serve a useful purpose in this project. The LCD, keypad, and an LED were incorporated into this system.
Plan	In the planning stage, the keypad device drivers from the project 2 submission were reused to create a template for this project. The LCD configuration was also left the same.
Create	The create stage started by looking at the schematic for project 2. Since the keypad code and LCD initialization were reused, it was imperative to ensure these peripherals were plugged into the same pins as they were in the previous project. Then the pins to connect the LED, buzzer and DHT11 to were selected. Once the physical prototype was done, each of the peripherals were programmed to perform their desired function.
Test	The test stage happened simultaneously with the create stage. The testing was all done by hand: the program was ran on the Nucleo and manually checked for errors. It was checked that the program can get all forms of user input. It was also checked that the climate monitoring system responds properly in all the edge cases.
Improve	The improve stage started once the desired functionality of the climate monitoring system was implemented. In this stage, quality of life changes that make the device easier or more appealing to use were added. For example, now when the device is in INPUT mode, the user can press C to clear their current entry (in other modes pressing C changes the unit between Celsius and Fahrenheit). Another example is the addition of the LED flashing when the user presses a button on the keypad.

Block Diagram

Below is a block diagram that gives a general overview of the climate monitoring system. It shows the how the user can change the states of the system using the buttons on the keypad. It is important to note that there are 4 inputs (Min Temperature, Max Temperature, Min Humidity, Max Humidity) that need to be entered and validated when switching from INPUT to MONITOR mode.

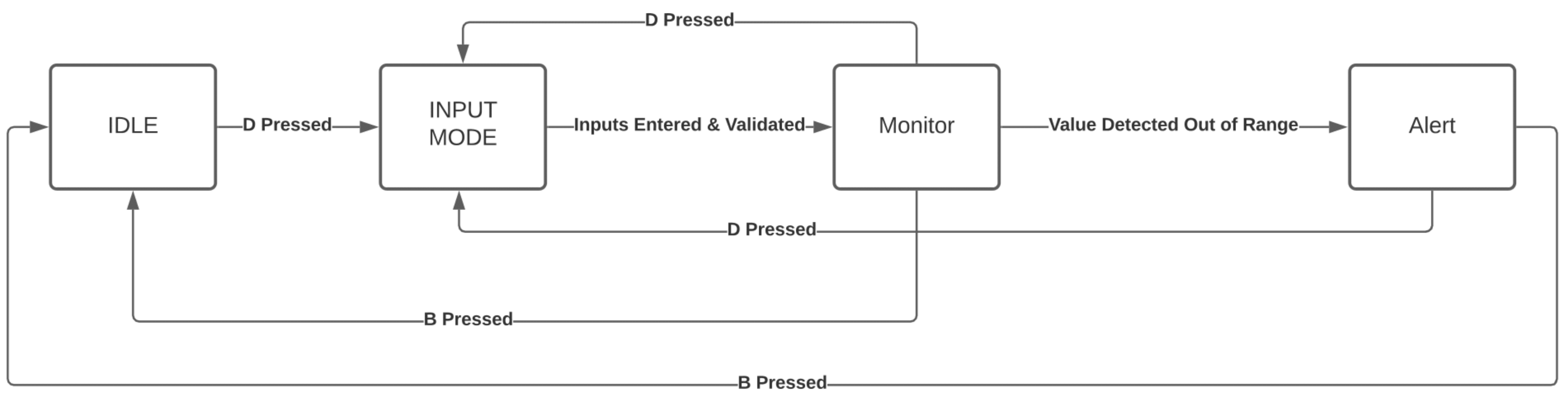

Figure 1: Block Diagram (source: LucidChart - https://www.lucidchart.com/)

Flow Chart

Below is a flow chart that portrays the control flow of the code for the climate monitoring system. There are 3 threads shown in the diagram below: main, t_lcd, and t_monitor.

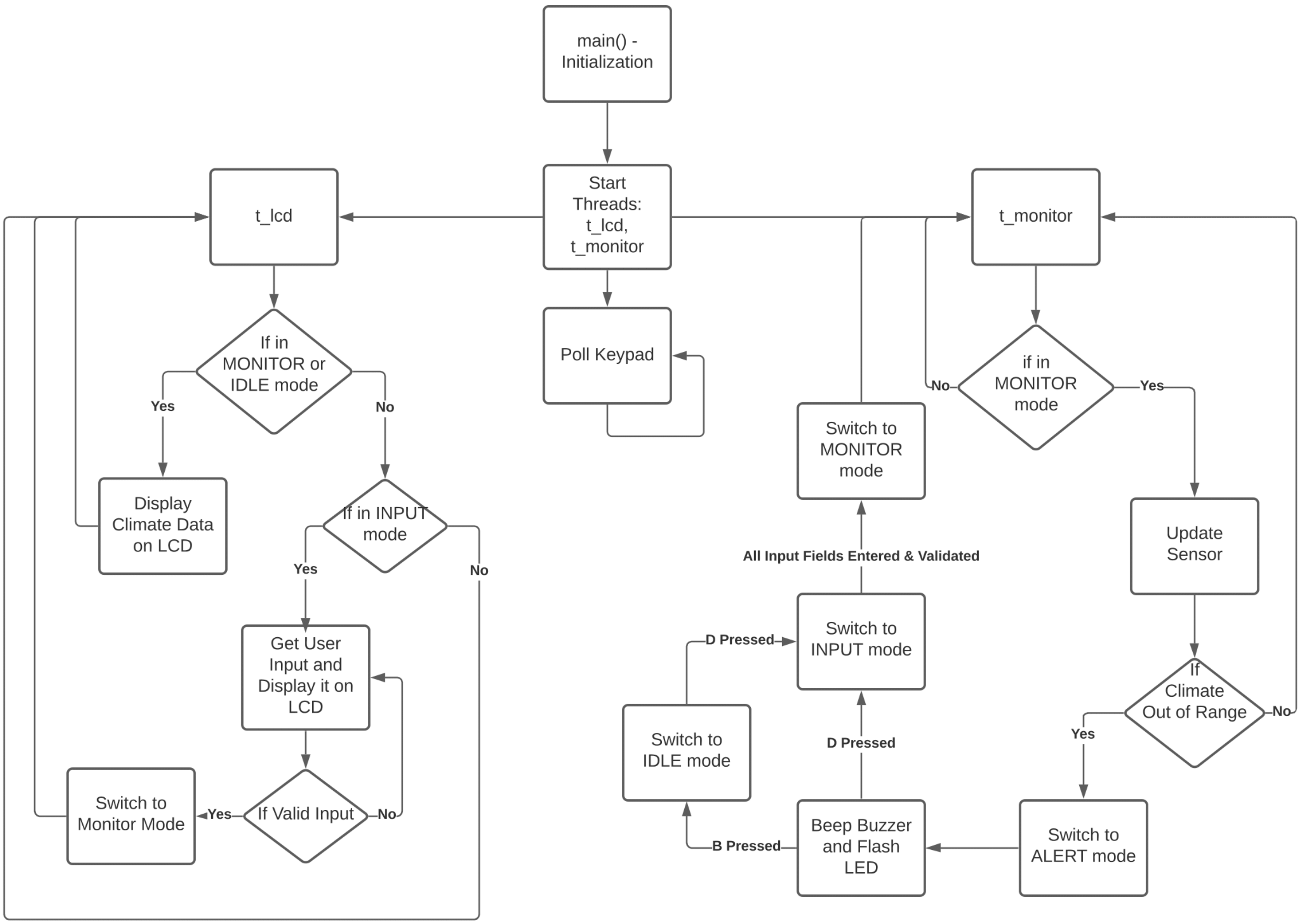

Figure 2: Flow Chart (source: LucidChart - https://www.lucidchart.com/)

Bill of Materials

Materials required to build this project:

- Nucleo-L4R5ZI
- Micro-USB cable
- Keypad
- DHT11 Temperature & Humidity Sensor
- Buzzer
- LCD
- Solderless Breadboard
- LED
- Jumper Wires

User Instructions

Schematic

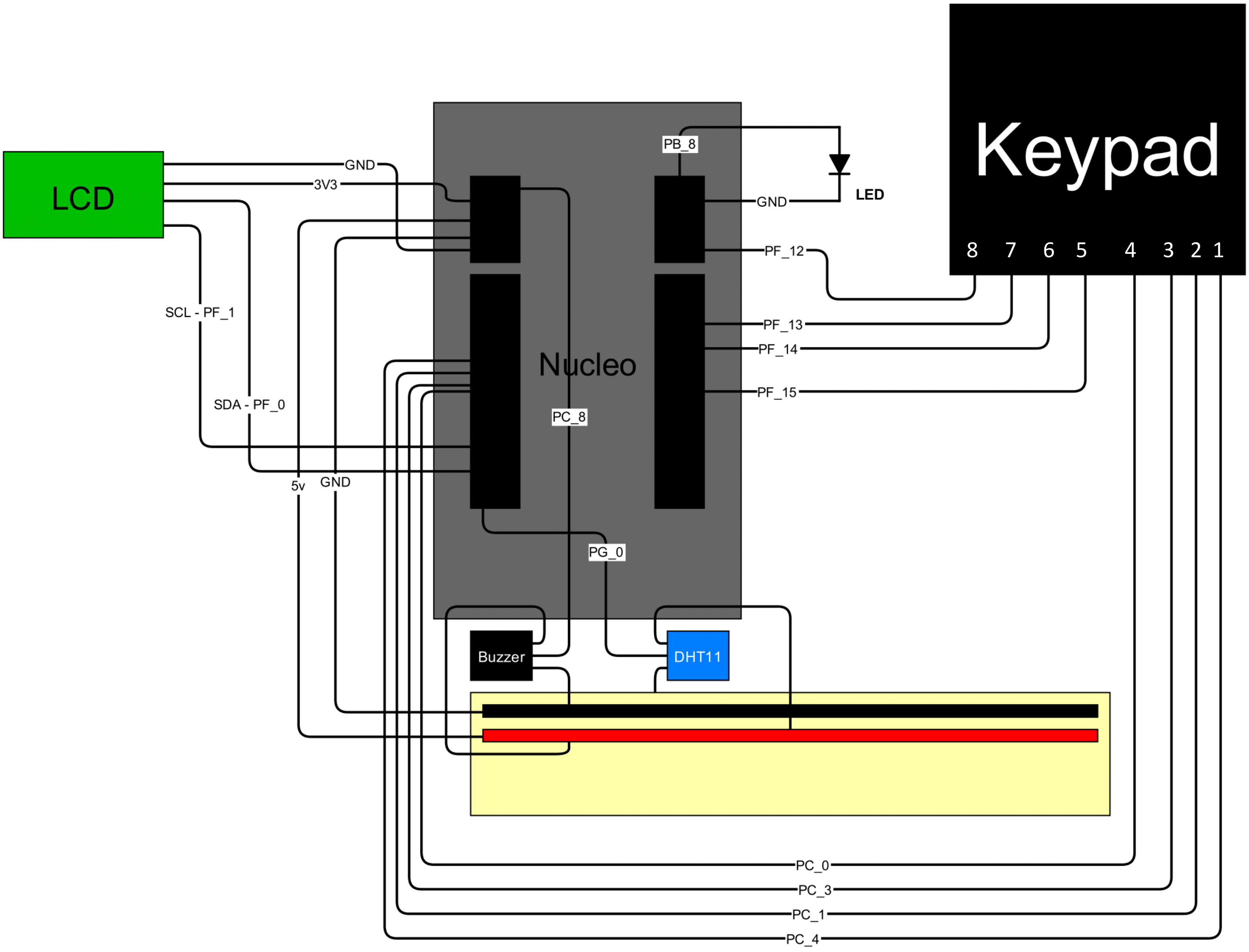

Figure 3: Schematic (source: SchemeIt - https://www.digikey.com/schemeit/project/)

How to Build the System

1) Start by connecting the keypad as follows:
    a. Line 1: PC_4
    b. Line 2: PC_1
    c. Line 3: PC_3
    d. Line 4: PC_0
    e. Line 5: PF_15
    f. Line 6: PF_14
    g. Line 7: PF_13
    h. Line 8: PF_12
2) Next connect 5V to the power and GND to the ground lines of the breadboard
3) Connect the DHT11 to the power and ground lines on the breadboard
    a. Connect the signal pin to PG_0
4) Connect the Buzzer to the power and ground lines of the breadboard
    a. Connect the signal pin to PC_8
5) Connect the positive side of the LED to PB_8 and the negative side to GND
6) Configure the LCD as follows
    a. GND: GND
    b. VCC: 3V3
    c. SDA: PF_0
    d. SCL: PF_1
7) Connect the Nucleo to a computer and run the program

How to Use the System

1) When the device is first started, it will be in IDLE mode
    a. From here the current temperature and humidity can be seen
    b. Pressing C on the keypad will change the units between Celsius and Fahrenheit
2) Press D to enter INPUT mode
    a. In INPUT mode the user is prompted to enter 4 different values
    b. Values are entered with the A button
    c. If the user input is invalid, they will be prompted to enter the input again
        i. Valid temperatures are between 0 and 50 degrees Celsius
        ii. Valid humidity values are between 30% and 95% RH
    d. Once all inputs have been gathered, the device switches to MONITOR mode
3) The device will monitor the climate and trigger an alarm if it detects the climate is out of range
    a. Pressing B will put the device in IDLE mode, and stop the monitoring process
    b. Pressing D will allow the user to modify the monitoring range
4) If the alarm is triggered the LED will flash and the Buzzer will ring on an interval. Also, the LCD will display a notification.
    a. Pressing B will put the device in IDLE mode, and stop the alarm
    b. Pressing D will switch the device to INPUT mode, allowing users to set a new range to monitor

Test Plan

This section will provide an overview of how the climate monitoring system was tested. Here are the steps taken to ensure the device has the proper functionality:

1) Power on the device
    a. Does it display the current temperature and humidity data?
2) Press C on the keypad
    a. Does the unit of measurement change to Fahrenheit?
3) Press C again
    a. Does the unit of measurement change back to Celsius?
4) Press D to enter INPUT mode
    a. Is the user prompted for "Min Temperature?"
5) Enter a numeric value and press A to confirm it
    a. Is the user prompted for the next input, "Max Temperature?"
6) Enter a numeric value and press A to confirm it
    a. Is the user prompted for the next input, "Min Humidity?"
7) Enter a numeric value and press A to confirm it
    a. Is the user prompted for the next input, "Max Humidity?"
8) Enter a numeric value and press A to confirm it
    a. If the input values are valid, does the device enter MONITOR mode?
    b. If the input values are invalid, does it tell the user and then re-prompt them for input?
9) Testing different ranges
    a. If the user sets a range that the current climate falls within, does the alarm remain untriggered?
    b. If the user sets a range that the current climate doesn't fall within, does the alarm get triggered?
10) The alarm is triggered
    a. Is the buzzer ringing and LED flashing?
    b. Does pressing B stop the alarm and display the climate information?
    c. Does pressing D stop the alarm and prompt the user for input?

Development Timeline

Date	Description
11/19/2021	Midpoint Submission, project planning.
11/20/2021	Copied project 2 main file and cut out the parts that weren't needed for this project, so that all was left was the code for the LCD initialization and the keypad.
12/2/2021	Added DHT library (from UBLearns) to the project and configured the device to print climate information on the LCD. A separate thread was established to update the LCD with output. There was an issue with printing floats, so the ability to switch the units to Fahrenheit was temporarily disabled
12/3/2021	Created a thread that monitors the climate data and triggers an alarm if it detects a value out of range. At this point in development the range was hardcoded rather than obtained from user input and the alarm was simply a print statement.
12/5/2021	Developed sequence for gathering input. Defined the user control scheme (via keypad).
12/6/2021	Configured buzzer to ring and LED to flash when the alarm is triggered.
12/7/2021	Fixed the issue with printing floats, configured device to work with Fahrenheit measurements as well.
12/8/2021	Code cleanup (with extra comments), quality of life improvements such as the LED lighting up on keypress and being able to clear entries by pressing C (only in INPUT mode).
12/9/2021	Header comment completion.

Outcome of Implementation: The implementation was successful and the device functions properly. No errors were detected while following the test plan.

Future Considerations: Bounce can sometimes interfere with the input collection process. In an improved version, bounce should be addressed with a more efficient solution. Along with that, modifying the input collection process to validate input after each entry rather than validating all 4 at once would create a more user friendly experience.