CSE321: Real-Time and Embedded Systems

Project 3

Katherine Stock
University at Buffalo
Fall 2021

Table of Contents

Introduction **3**

Specifications **4**

Features **4**

Applications **5**

Block Diagram **6**

Functionality **7**

Diagram (FSM State Diagram) **8**

Bill of Materials **9**

User Instructions **10**

Schematic 10

How to Build the System 10

How to Use the System 11

Test Plan **12**

Development Timeline **12**

Results **13**

Recommendations for Improvement **13**

Introduction

The system described in this report is a threaded weather predictor and carry along recommender. Its purpose is to read the current temperature and humidity outside – with the DHT11 sensor placed on the sill of an open window or outside of a door – and suggest items that the user might need for those conditions.

The user will visually read output from the system LEDs and 7 segment display via a key provided in the ‘How to Use the System’ section of this report. The key is also provided as key.pdf.

The purpose of the system is to keep the user prepared for the weather which falls into the category of safety. Carry along items include rain boots, an umbrella, snow boots, a hat and mittens, light coat, water, sunscreen, and sunglasses to protect the user from weather-related safety concerns such as hypothermia, frostbite, sunburn, dehydration, or injuries from slipping and falling in snow or rain.

Specifications

Specifications of each input and output are as follows:

- DHT11 - a flag bit set every 5 seconds indicates that the DHT11 should be read again. The sensor read updates the temperature and humidity values for the sensor that are then used to update the critical section values and the state of the machine every 5 seconds. The sensor reads a maximum of 90% humidity and a temperature range of 32F-122RF. The condition boundaries to set each weather state are:
  - Humidity 85% or higher
    - Temperature 32 degrees Fahrenheit – Snowy
    - Temperature higher than 32 degrees Fahrenheit – Rainy
  - Humidity less than 85%
    - Temperature between 32 and 60 degrees Fahrenheit – Cold
    - Temperature between 60 and 80 degrees Fahrenheit – Moderate
    - Temperature between 80 and 122 degrees Fahrenheit – Hot
- LEDs - 5 LEDs all with a different color (one green, one red, one yellow, one blue, one white) that correspond to the weather conditions described above. Depending on the state of the machine, one and only one LED will be lit up at all times.
  - Snowy - White
  - Cold - Blue
  - Moderate - Yellow
  - Hot - Red
  - Rainy - Green
- 7 Segment Display - The display will show 1 number in the range of 1-5 corresponding to the weather conditions. These numbers can be used with the key to show the user what to carry with them that day. Only one number will be displayed at a time, decided by the state of the machine.
  - Snowy - 1
  - Cold - 2
  - Moderate - 3
  - Hot - 4
  - Rainy - 5

Features

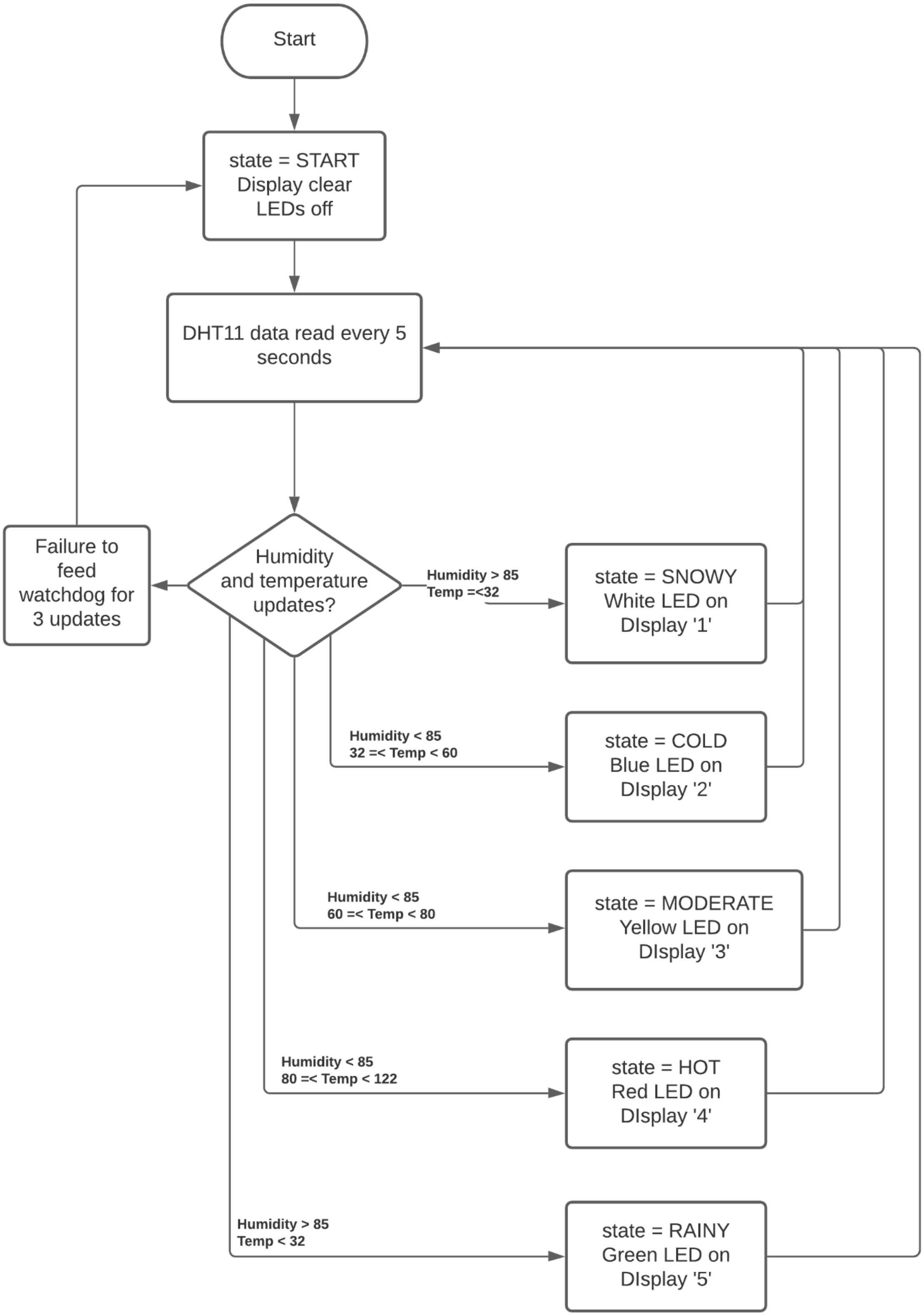
Features of the weather prediction system include the following:

- A DHT11 sensor to read current air temperature and humidity percentage.
- 5 LEDs of different colors which update every 5 seconds to indicate the current weather state.
- 7 segment display which updates every 5 seconds displaying a number correlated to carry along item suggestions.

Applications

The application of this system is to predict the weather conditions for that day and use its recommendations to decide what to carry and how to dress that day. This system is best used in climates with great variation in temperature between 32 and 90 degrees Fahrenheit.

Block Diagram



Functionality

Upon start, the tickers immediately begin running and the first sensor read will occur 5 seconds after start. A flag-based ISR is triggered on one ticker which will signal the sensor to be read. The choice for a flag-based interrupt to trigger the read was made because the DHT11 API read function cannot be run in an ISR itself without errors. The other ticker adds a DHT11 ISR to a threaded EventQueue that is dispatching forever. This queue allows the global state, humidity, and temperature variables to be a protected critical section and only update when told by the threaded queue (on the condition that the sensor read returned okay). The combination of tickers and the EventQueue were chosen for synchronization due to the time-based constraints of the DHT11 and the protection of the critical section offered by the queue thread.

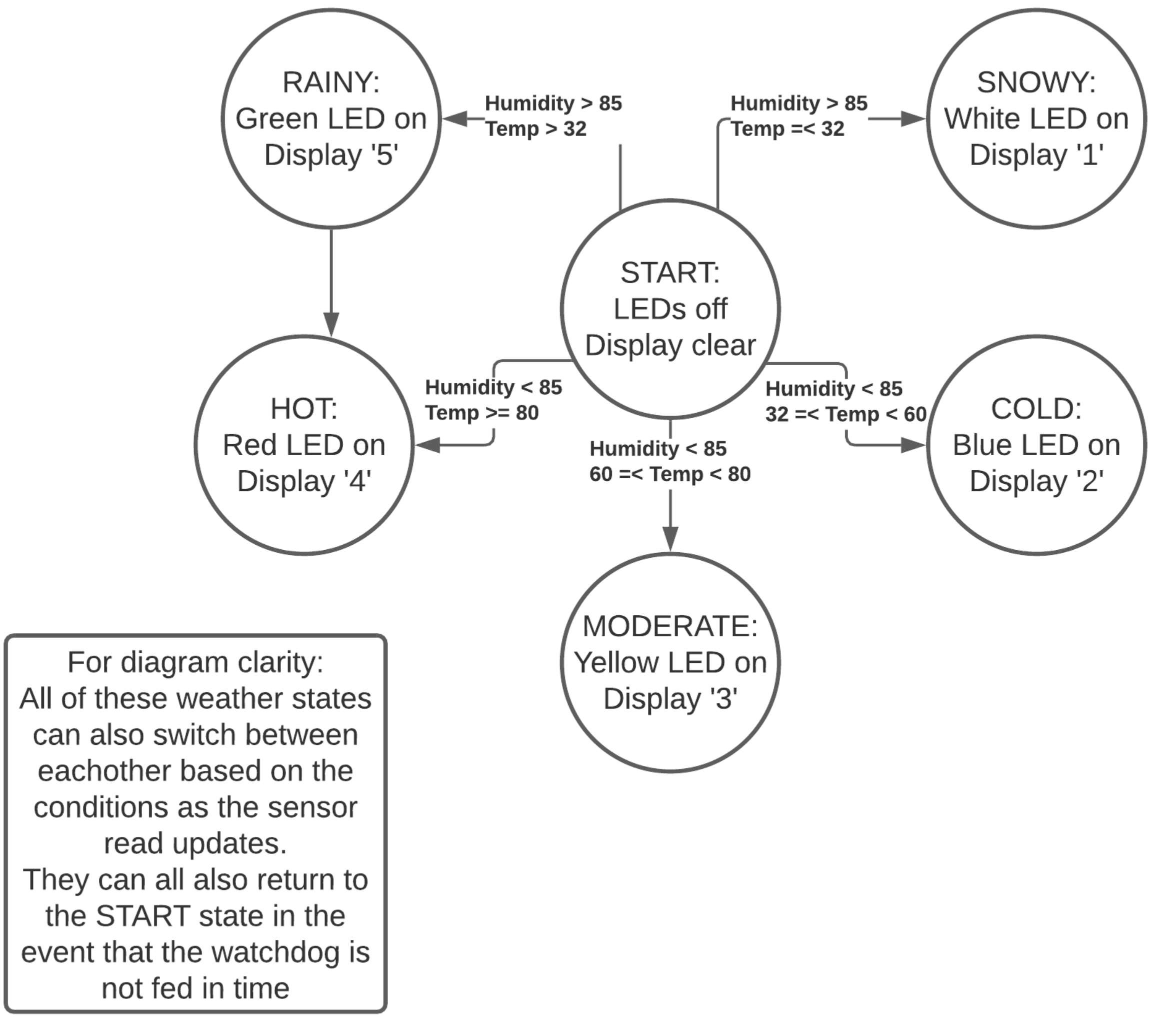
The first read is almost always cold as the sensor is still warming up to correctly reading the room temperature and humidity, but this adjusts by the next read.

Once started, the system will continue to run and read the current conditions with no intervention necessary. If the user quickly changes the environment (ex. Suddenly moving indoors from outdoors or vice versa), the sensor will need about 15 seconds to adjust as the temperature of the sensor slowly changes. After this, the sensor will be adjusted to the new area and the LED/7 segment output will be accurate to the new space. These 5 LEDs are turned on and off via bitwise driver control within a state machine model.

After every 5 seconds, the LED and 7 segment output has been updated and can be considered reliable after the 15-20 second adjustment period when changing areas. The user can now read the output visually and interpret its meaning using the provided key in key.pdf and in the "How to Use the System" section of this report. The user will then know the expected weather conditions and plan to bring the recommended items accordingly.

The system watchdog is fed every 5 seconds when there is an okay read of the DHT11. Should the watchdog not be fed for 15 seconds – reading fails for 3 consecutive cycles – the system will return to the safe start state and the system will refresh to continue 5-second updates.

Diagram (FSM State Diagram)



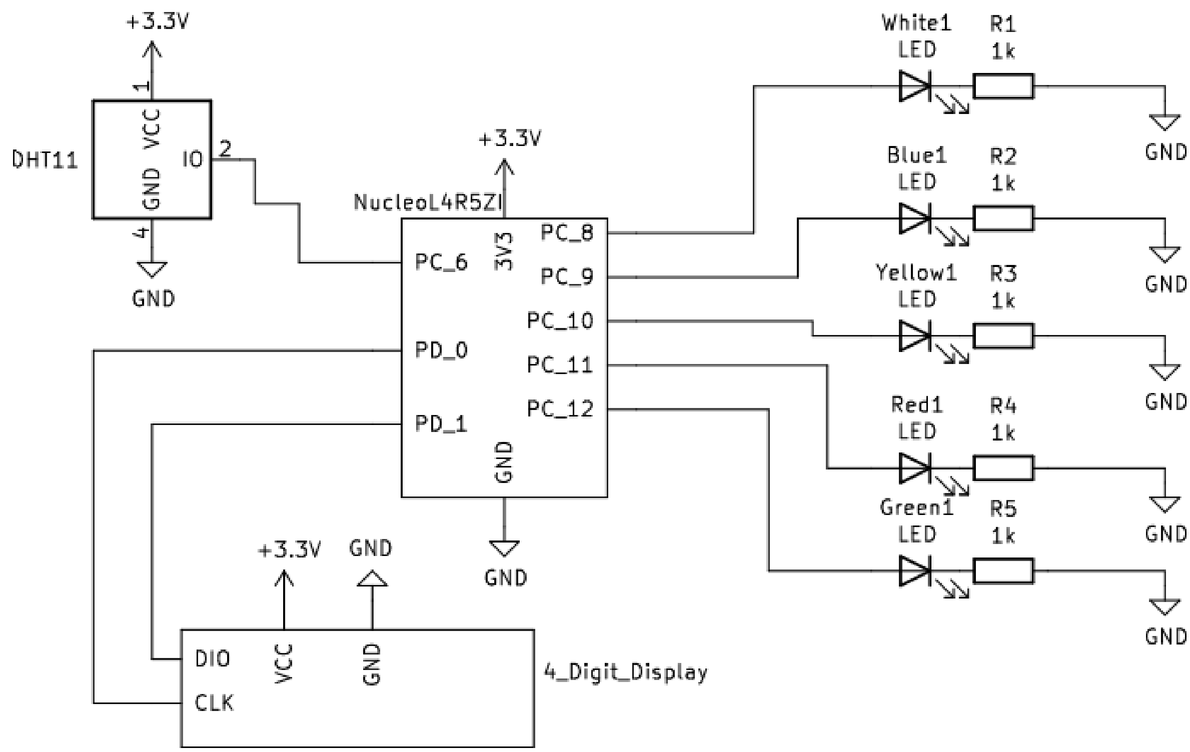
Bill of Materials

	Item Name	Quantity	Price
1	Nucleo-L4R5ZI	1	$28.14
2	Breadboard	1	$15.40
3	USB a to Micro USB B cable	1	$2.68
4	4-Digit 7 Segment Display (and F/F 4 jumper adapter)	1	$5.49/2 pack
5	DHT11 Temperature Humidity Sensor Module (and F/F 3 jumper adapter)	1	$6.59/2 pack
6	Male/Male Jumpers	14	$3.90/140 pack
7	LEDs	5	$3.95/25 pack
8	Resistors (1kOhm)	5	$0.95/20 pack
	Total		**$67.10**

Technology requirements only include access to a computer with internet and MBED Studio to program the Nucleo.

User Instructions

Schematic



How to Build the System

The hardware of the system can be built following the schematic above. The 3.3 Volt power supply of the Nucleo is connected to one of the power rails on the breadboard so it can be used by both the DHT11 and the 7 segment display. One ground supply of the Nucleo is also connected to one of the ground rails on the breadboard so it can be used by the LEDs, DHT11, and 7 segment display.

Pins PC_8, PC_9, PC_10, PC_11, and PC_12 should all be connected to the breadboard with an LED (of appropriate color as indicated in the diagram) in sequence with a 1kOhm resistor which is then connected to the ground rail.

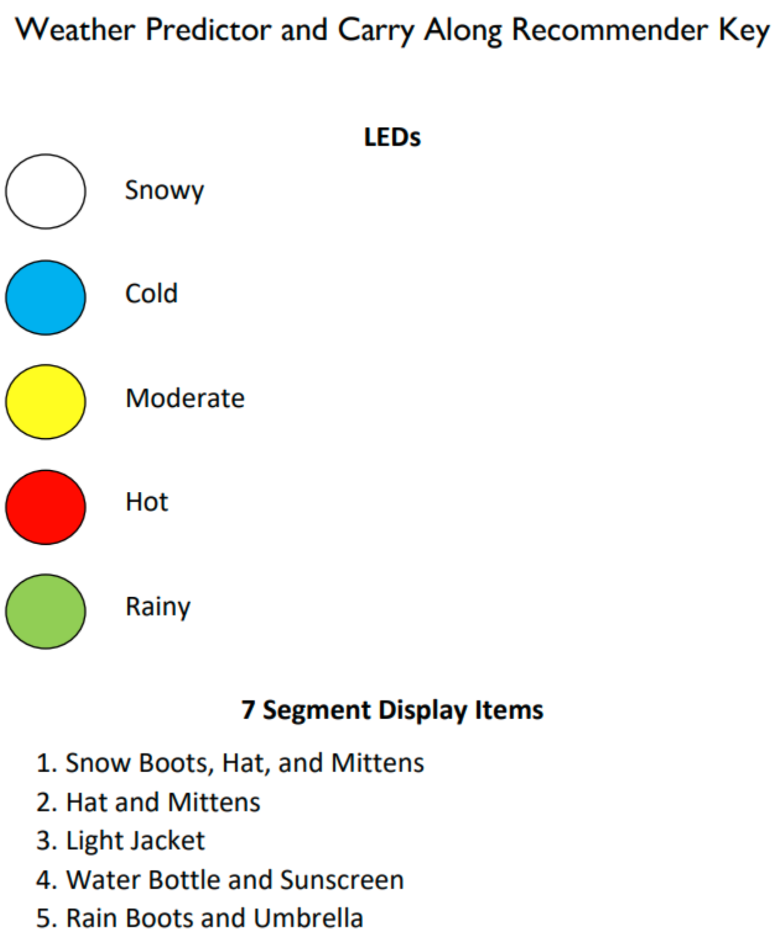
In addition to the 3.3 Volt power rail and the ground rail, the DHT11 should connect the output pin to PC_6 and the 4-Digit-Display should connect the CLK pin to PD_0 and the DIO pin to PD_1.

How to Use the System

The system is used by placing the sensor in the area the user wishes to know the weather conditions for. The sensor must be in the open air and unobstructed to get an accurate read of the weather conditions.

At start, the first weather read will occur after 5 seconds and reads as Cold conditions regardless of the actual weather conditions. After 10-15 seconds, the system will be accurately reading the conditions and communicating the weather and what items to bring along to the use. This information is shared in the form of LEDs (one will be on) and a number displayed on the 7 segment display. The system will continue to run and update until it loses power.

The user can now use the following key (also provided in key.pdf) to know what the conditions outside are like and decide what to bring with them on that day.



Weather Predictor and Carry Along Recommender Key

LEDs

Snowy

Cold

Moderate

Hot

Rainy

7 Segment Display Items

1. Snow Boots, Hat, and Mittens
2. Hat and Mittens
3. Light Jacket
4. Water Bottle and Sunscreen
5. Rain Boots and Umbrella

Test Plan

Testing this project posed to be a challenge as testing must be done in December where weather fits into only 3 categories possible for the weather predictor – snowy, cold, and possibly rainy and moderate. This made testing the entire system difficult and required simulated weather conditions using a room with a fireplace to mimic warm, dry weather conditions that would trigger the 'HOT' state.

Testing pieces along the way made the process go smoothly. The DHT11 code was tested independently of the rest of the system by having it first read-only and print the updated variables, and then read synchronously with the EventQueue to read and update every 3 seconds,

The 7 Segment Display was also tested independently by writing and clearing values independently of the synchronization and the rest of the system.

LEDs were tested independently by turning each on and the rest off, switching every 3 seconds to ensure that bit masking was done correctly. LEDs and 7 segment display output for each state were also tested independently of input and scheduling to ensure that both worked together to communicate the proper weather conditions and suggested items.

Development Timeline

Development started with the implementation of LEDs and the corresponding device bit masking. Similar use of LEDs occurred in both project 2 which made this a logical place to start implementation and testing. A key was developed at this time using each of the basic 5 LED colors as well as the numbers that would eventually display on the 7 segment. A copy of this key can be found in the GitHub repository with title key.pdf and is also located in the 'User Instructions' section of this document.

Once the bit masking was tested and adjusted as necessary, the 7 Segment Display was added to the project. States were created for each weather condition and tested individually with the 7 segment display, then in tandem with the corresponding LED. LED matters especially for the project because of the color-based meaning.

The final peripheral to implement was the use of the DHT11 temperature and humidity sensor. This naturally had to be implemented with the scheduling and interrupt portion of the project because of the constraints of the sensor. Getting these to work together proved most challenging due to the limitations of functions able to be run in an ISR. The sensor.read() of the given API had to be run in main so the ISR was moved to instead set a flag bit that would trigger the sensor read in main. Directly adding an event to the EventQueue after this sensor read to protect the critical variables holding state, temperature, and humidity also proved challenging. Instead, using a second Ticker to add the value update ISR to the EventQueue allowed for the safe update of the critical section and scheduling to work with the Ticker setting the read flag.

With this, testing in different environments could be done and final adjustments to temperature and humidity boundaries could be made.

Results

The project implementation was successful. Slight changes to output peripherals were made since the proposal to better accommodate the use of the bitwise configuration on LEDs and to diversify the outputs from being two display screens. The system was tested in multiple simulated weather conditions and each visual aspect updated correctly according to the boundaries set by the system.

The developed system meets all of the requirements specified in the project proposal, with slight peripheral choice modifications, as well as meeting software technique requirements.

Recommendations for Improvement

One major limitation of this project is the DHT11 API. The sensor read most often returns a CHECKSUM error, 90% of the time this error returns (which is why either the CHECKSUM or OK status are ‘okay’ in terms of the system) with accurate readings but this issue occasionally causes a bad temperature read which can jump between condition boundaries. This was observed most often with occasional extraneous reads of high temperature when in the moderate state which cause the state to be set as hot until the next sensor read.

Another possible addition is the use of a keypad peripheral so the user can set boundaries for each condition based on their climate. Hot and cold in Florida have a different meaning than they do in Buffalo so this addition would allow more people across the world to use the system with accurate results.