Ho Chi Minh City University of Techonology
Faculty of Computer Science and Engineering

– CO2021 - Microprocessors-Microcontrollers –

Proposal
A Smart Low Temperature Dehydration System

Instructor: Dr. Nguyen Tran Huu Nguyen

Name	ID
Ho Le Thuc Quyen	1752454

Contents

List of Figures

Acknowledgement

As with any great endeavor, this project would not have been possible to complete without the help and support of many others. First and foremost, I would like to express my sincere gratitude to my project instructor, Dr. Nguyen Tran Huu Nguyen. His supervision and expert knowledge have added considerably to my work. I would like to thanks him for the course of Microprocessor and Microcontroller. This course have provided the basic knowledge of microprocessor, microcontroller and also the PIC18F8722 board which is used to practise what are learnt on the lectures such as state machine, interrupt and timer. Last but not least, I also would like to give thanks to all my Computer Engineering "brothers" for supporting me.

1 Introduction

This project is aimed to demonstrate a smart low-temperature dehydration (Smart LTD) system, which is reliable and efficient in food preservation. The purpose of this system is drying food to prevent the growth of bacteria, molds, and insects. The Smart LTD system has an additional part, an IoTs part, compared to an original one. The IoTs which stands for the Internet of Things is the connection of devices, such as sensors, mobile phones, vehicles, through the Internet to exchange data with each other. IoTs allows the system to save data for future improvement.

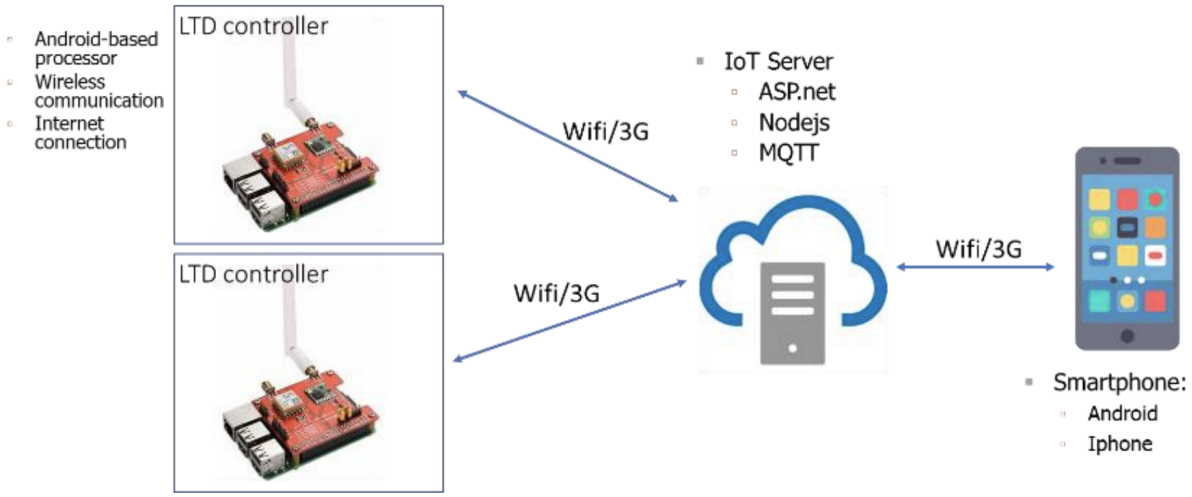

Figure 1: General Design for a Smart LTD system

Figure 1 shows a general design for a system that includes three main parts, an LTD machine, a server, and a mobile phone which runs a user app. First of all, the LTD controller takes control of devices based on temperature and humidity then sends data to IOTs sever in real-time via Wifi or 3G. Finally, data from the server are sent to the user app installed on mobile phones for checking.

This proposal describes the design as well as a simple implementation for a LTD machine consisting of the controller, sensors, and devices.

2 Design

2.1 Overview structure

The LTD machine is divided into three main parts, the sensors which measuring environment's status, the controller and devices which are operated by the signal sent from the controller.

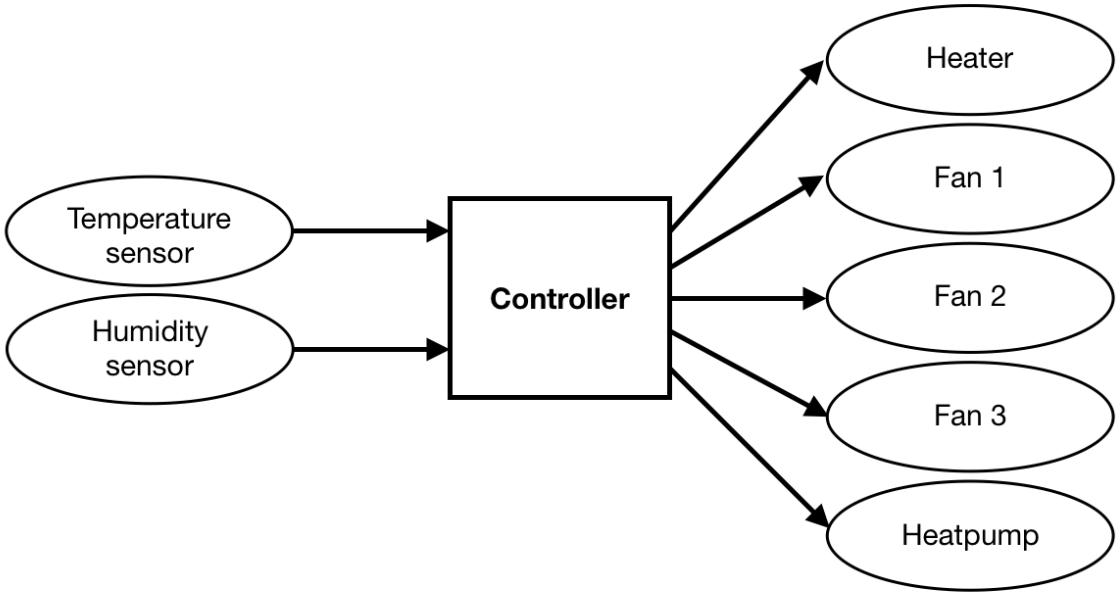

Figure 2: Connection between controller, sensors and devices

The LTD devices including a heater, heat pump and set of fans. First of all, the controller receives the data of temperature and humidity from sensors. After that, it takes control of all devices corresponding to each condition, for example, turn off the heater when the temperature is high enough.
The heater is responsible for warming the room and while it is on, fan 2 is also on to make the air distributed evenly. When the heat pump is on, the heater is off, fan 3 is on and vice versa. Moreover, fan 1 is used to transfer a humid air from the room to outside.

2.2 Operation

In the beginning, the heater is turned on and when the temperature is greater than a defined temperature value, it will be off. Fan 1 also operates in the same way as a heater but it depends on humidity.

In addition, the heater and heat pump can run periodically using a timer. Figure 3 below indicates clearly the description of all the devices's operation.

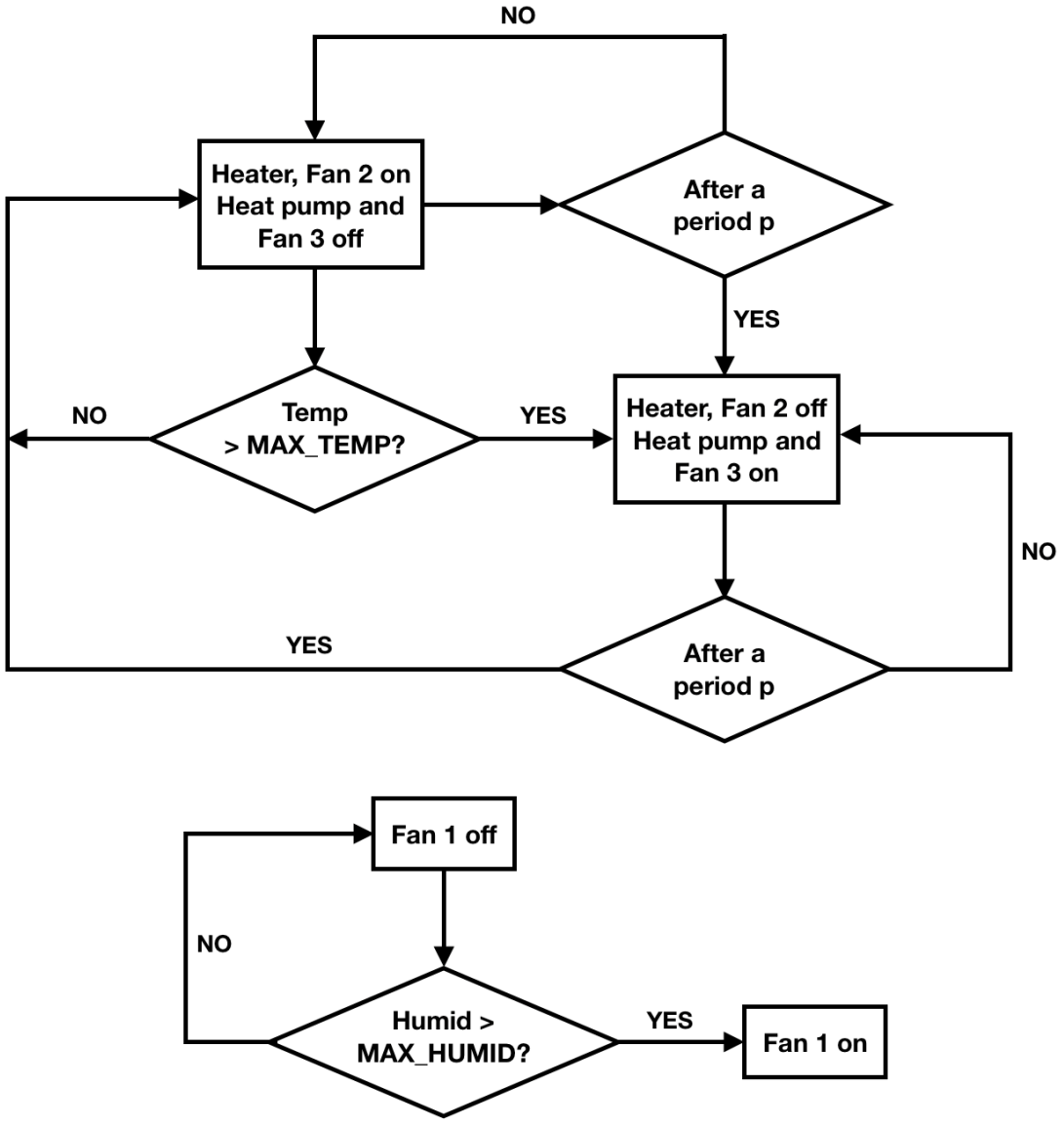

Figure 3: Flow chart of devices operation

The LTD machine is supposed to use at least three temperatures and three humidity sensors measuring the environment to make the system more reliable and easier to check. The controller gathers the information of three sensors of each type and takes the average of the two nearest values as the room

temperature or humidity. In case the other value is much different, much larger or smaller, it should have been broken and need to be replaced.
When making a comparison, if the value is higher than the defined value, the controller will not send a signal to change the state of devices immediately. Instead of that, it will wait for a period of time, 2 seconds, for example, to ensure that there is no noise.

3 Implementation

3.1 Materials

3.1.1 Hardware

In this implementation, one PIC18F8722 and one DHT11 are used.

• **PIC18F8722:** This implementation uses PIC18F8722 which is a microcontroller, as a controller for the LTD machine. It mixes and matches peripherals and I/O types, both digital and analog that is suitable for implementing the system. Furthermore, it have an interrupt and counters helping the arrangement of tasks easier and more efficient. Its datasheet can be found at: bit.ly/PIC18F8722

Figure 4: PIC18F8722

• **DHT11:** DHT11 is a temperature and humidity sensor. It measures the temperature and humidity of environment if it receives a start signal and then sends out the 40-bit data. The benefits of DHT11 are

the price is reasonable and the output is digital which can be implemented easier than analog output. However, the error is larger ($\pm 2^{\circ}$C and 5%) than, for instance, DHT22. The implementation does not use DHT22 because of the lack of budget. Its datasheet can be found at: bit.ly/DHT11DS

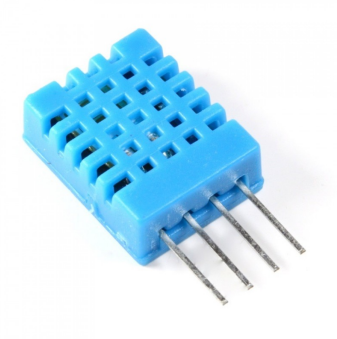

Figure 5: DHT11

3.1.2 Software

• MPLAB X IDE

The MPLAB X integrated development environment (IDE) is an expandable, highly configurable software program that incorporates powerful tools to help you discover, configure, develop, debug and qualify embedded designs for most of Microchip's microcontrollers, including PIC18F8722, and digital signal controllers. [3]

The source code, which can be written in C language, for the IDE is compiled under the MPLAB XC Compiler. This IDE can run on Windows, Linux as well as macOS operating system.

3.2 Pin Connection

PIC18F8722:

• Pin RD0 is used to send signal to and receive data from DHT11. The voltage supply for DHT11 is from pin 5V.

• LATD1, LATD2 and LATD3 represent fan1, fan2 and fan3, respectively. Besides that, LATD6 and LATD7 also use as heater and heatpump. LATD is the pin of set of leds, from led 1 to led 8.

3.3 Methods

3.3.1 Interrupt

Interrupt basically is a signal that suspends a main routine. The MCU will leave a program to execute the specific code (ISR Handler) when an interrupt happens. In a situation when MCU need to check some events happening in short time, for instance, $1\mu s$, interrupt is suppose to use to make the whole system runs more efficiently. [2]
In this project, interrupt is used to send signal and receive signal from DHT11 and to control the operation of devices.

3.3.2 Main states

The system has three main states which are shown on the figure below. For two states named "chooseTemp" and "chooseHumid", users are allowed to change the defined MAX values of temperature and humidity and in other state "idle", the machine operates as the description above. Button RA5 is used to change state.

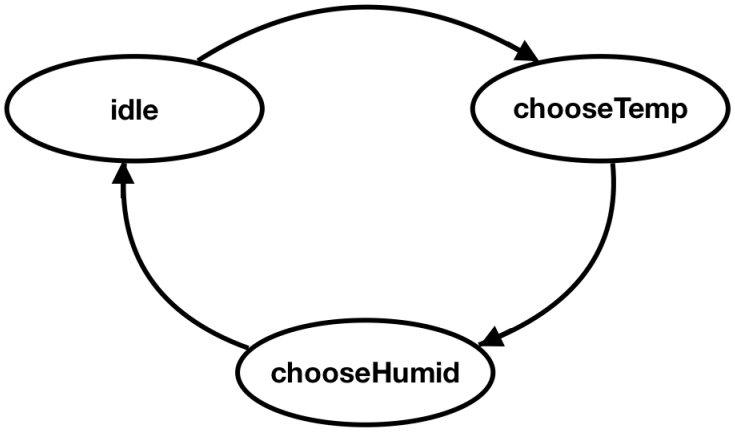

Figure 6: Three main states of the machine

3.3.3 User-defined MAX values

When the users are in state "chooseTemp" or "chooseHumid", they press button RB0 to change the value. In this two states, they have three more sub-states which displayed on figure 7. At first, they stays in "init" and when the button RB0 is pressed, they change to state "iNor" and the MAX value increases by one. If the button is held more than one second, the value will automatically increase

by ten every, in this implementation, $400ms$. Finally, whenever the button is released, the state turns back to state "init".

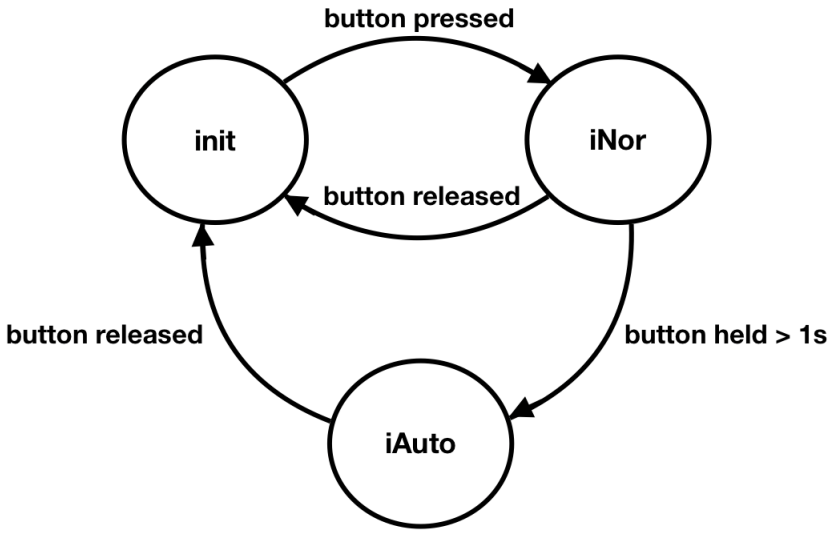

Figure 7: State Machine of defining MAX value

3.3.4 Reading data from DHT11

The figure below shows DHT11 timing diagram, how to get data from DHT11. When the controller want to get information, it has to send out start signal including at least $18ms$-low voltage signal and $20\mu s$ to $40\mu s$-high voltage signal. After that, DHT11 sends response signal to the controller which are $80us$-low and $80us$-high signal and following that is 40-bit data containing information about temperature and humid. To distinguish bit "0" and "1", the controller will detect the time length of sending high voltage from DHT11 which can be seen on the figure.[1]

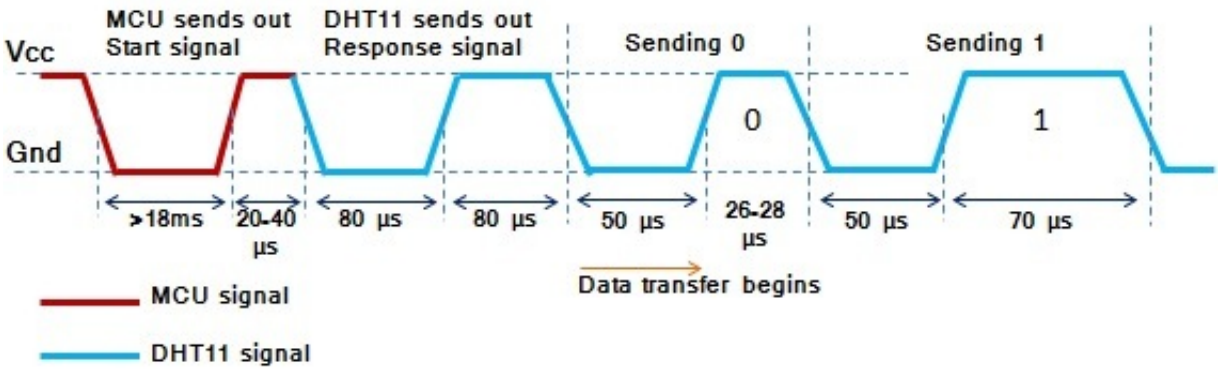

Figure 8: DHT11 timing diagram

The system need to get data from environment regularly, therefore interrupt is expected to use to throw start signal every suitable period of time, about a second.

3.3.5 Controlling output devices

Once the controller have received input data from DHT11, it will send signals to control devices corresponding to each condition.

Figure 9 depicts the state machine for heater, heatpump and their supporting fans, fan 2 and fan 3 respectively. In state "heater", heater and fan 2 are on heatpump and fan 3 are off which are reversed in state "heatpump". There are flags, changeS0 and changeS1, to show when the state changes. At first, the system is in state "heater" and when temperature is larger long enough or when heater run time equals to the defined period, which makes "changeS0" go to "1", the state will change to "heatpump" state. The system changes back to state "heater" only if the temperature is under the limitation and heatpump runs for a defined period of time. The condition that changing the state from heatpump to heater makes "changeS1" go to "1".

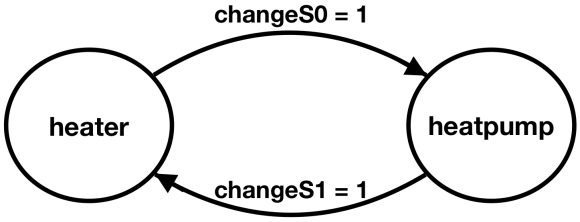

Figure 9: State Machine for heater and heatpump

The operation of fan 1 is quiet similar to heater. The difference is that fan 3 only depends on humidity. When the humidity is higher than the defined one, flag "change" will be "1", otherwise, it holds "0". Therefore, fan 1 is on when "change" is 1 and is off when "change" is 0. Moreover, the speed of fan 1 depends on the value of humidity.

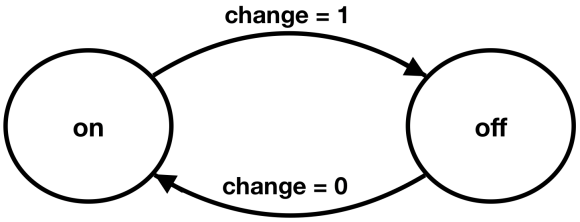

Figure 10: State Machine for fan 3

3.3.6 Preventing noisy data input

All devices will change their state when some conditions are reached, for example, temperature is higher than a defined MAX value. However, the sensor sometimes sends noisy data to the controller which can affect to the operation of devices. To prevent that risk, controller should ensure the period of time that conditions reached is long enough.
As can be seen from the figure, there are four state, namely “normal”, “bgLarger”, “stable” and “bgSmaller” and two defined values named MAX, which representing limited value of temperature or humidity, and STABLE_PERIOD, which is the period of time that make sure the environment’s status changed. The condition for converting into another state is written above each arrow.

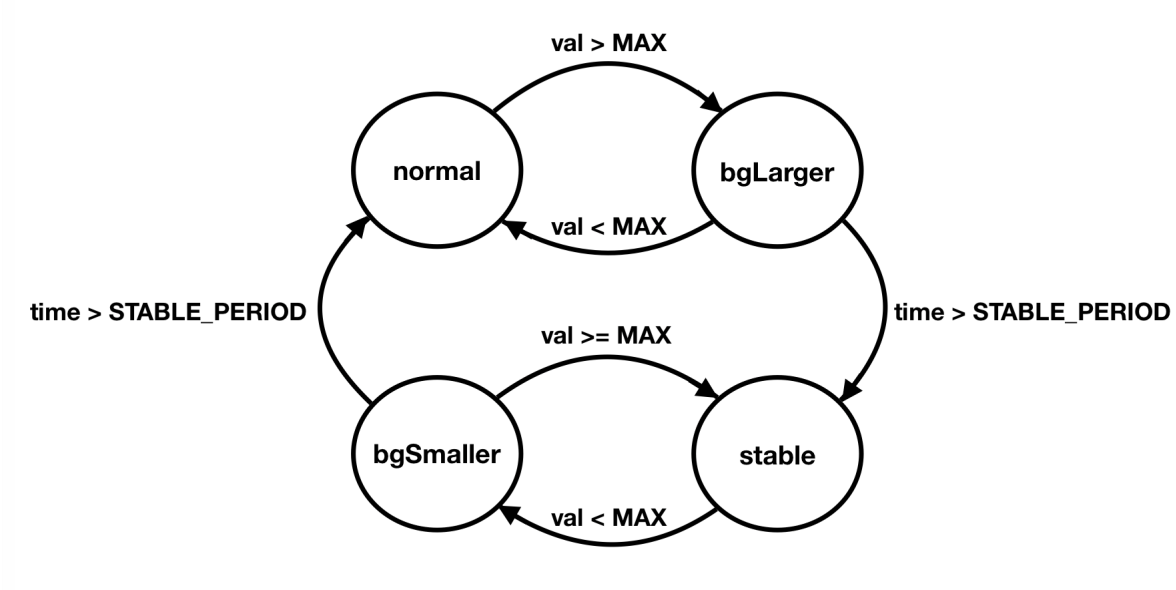

Figure 11: State Machine for receiving data

First of all, the system is at state “normal” and when the received data is grater than MAX, it immediately changes to state “bgLarger” and waits for STABLE_PERIOD time. If the received data is smaller than MAX, the state will turn back to state “normal”. After that, it will go to state “stable”. In this state, devices takes changes based on the conditions which are mentioned above. The conversion between state “stable”, state “bgSmaller” and state “normal” are similar to that of others three states. An only difference is that the condition of received data is reversed.

3.4 Result

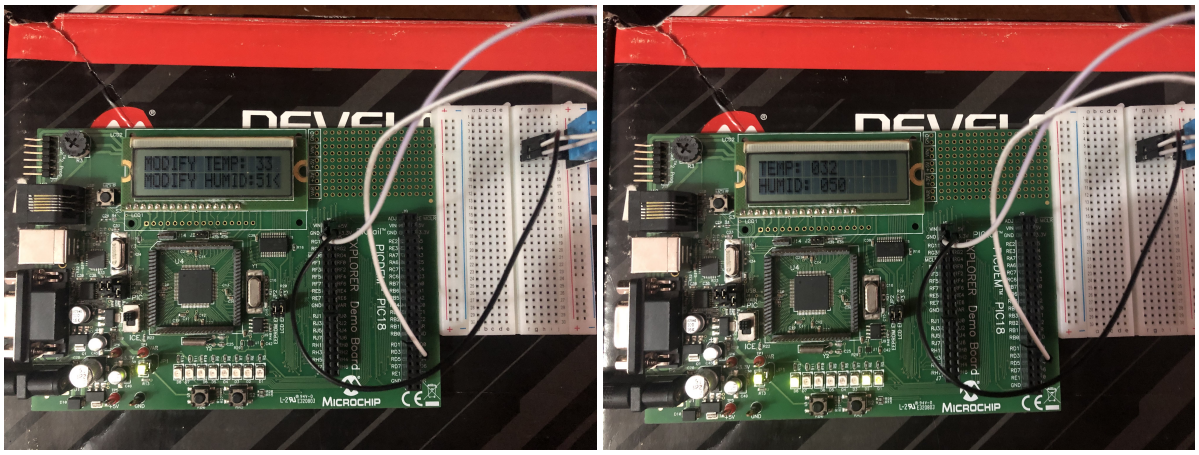

Figure 12: Demonstration on PIC18F8722

The implementation running on the board is what it is expected to show. The pictures, which is on the left, shows the LCD screen when users choosing defined MAX values and an arrow showed users which value they are changing, temperature or humidity. Other image displays environment's temperature as Celsius unit and humidity and all leds's operation which representing heater, heatpump and fans. After testing for a while, the system runs smoothly.

4 Conclusion

The system brings convenience for users who have a demand for drying food and it is easy to use. The results and methods show how the LTD machine implemented. Basing on the knowledge in microcontroller programming, state machine, and components, the illustration of the machine operates smoothly. This project is a great opportunity to apply what has been studied in the lectures and complete all the tasks that had been planned before. In the future, I will using two more DHT11 to make the measurement more reliable. Furthermore, I will make a complete smart low-temperature dehydration system. In other words, I will create my own server and mobile app so that data can be recorded and the system can be checked and run more efficiently.

The source code of the implementation can be found at GitHub: http://bit.ly/SLTDmplab

References

[1] *DHT11 TEMPERATURE AND HUMIDITY SENSOR WITH PINGUINO*. URL: https://www.electronics-lab.com/dht11-temperature-humidity-sensor-pinguino/.

[2] *Interrupts In PIC Microcontrollers*. URL: https://www.deepbluembedded.com/interrupts-in-pic-microcontrollers/.

[3] *MPLAB® X Integrated Development Environment (IDE)*. URL: https://www.microchip.com/mplab/mplab-x-ide.