# Project 1:
# Temperature Alarm

ECE 218 Embedded Microcontrollers

By: Lilah Kelly & Samuel Degemu

01/29/2021

**Objectives:**

This project aims to build a temperature-based alarm system that will record and show the current temperature, set off an alarm if the temperature surpassed 80 degrees Fahrenheit, and reset with the press of a button. This requires creating an embedded microcontroller system using the Microchip Microstick II board that contains the ADC on the PIC24, an active buzzer, a serial connection, an analog temperature sensor, and an active low button. The board is meant for housing the active buzzer, temperature sensor and active button, whereas the ADC on the PIC24 and the serial connection are necessary to display the temperature and whether or not the alarm was armed on a monitor. When implemented correctly, the monitor will display “DISARMED XX.X degrees Fahrenheit”, then after pressing and releasing the button, will display “ARMED XX.X degrees Fahrenheit”, then will sound a buzzer and display “ALARM! Press button to stop!” if the temperature is at or surpasses 80 degrees Fahrenheit. This will continue until the button is pressed again, causing the system to once more become disarmed.

**Design:**

**Hardware**

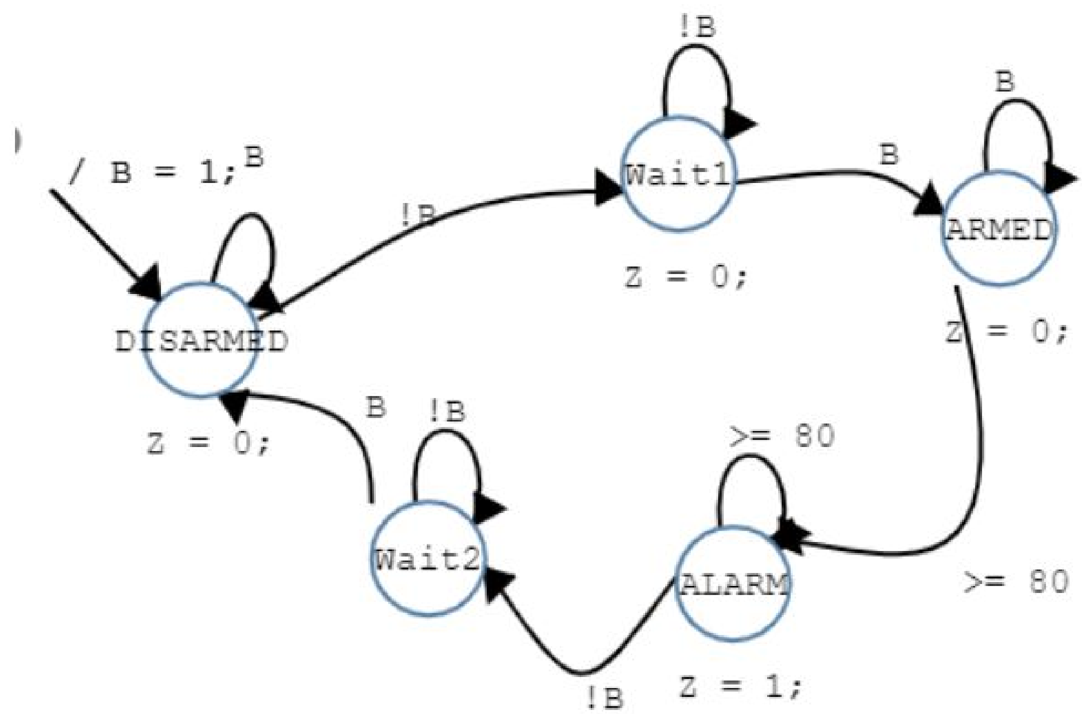


**Figure 1:** HLSM for alarm system

The initial step in creating the temperature-based sensor was to design the state machine on figure 1. After reviewing the drawn out state machine, it was written out as code in the project file. This essential step was key to fleshing out the rest of the details in the project. The next step was to wire the circuit together and create the macros in the code according to the circuit setup. The block diagram on figure 2 shows the inputs on the left side of the pic24 and the singular output on the right. The push button, which serves as the manual input, is pressed to switch to the ARMED and DISARMED states when appropriate. It is also an active low, so when not pressed the input pin is connected to a 10K ohm pullup resistor. The second input, the temperature sensor, outputs a voltage between 0 and 3.24 volts (specific to the microcontroller used), based on its temperature readings. The voltage output and its corresponding temperature reading is listed in the Lmt84 datasheet referenced in the appendix. The output buzzer will sound when the ALARM state has been reached and will serve as an indicator the temperature has reached 80 degrees. Lastly the microcontroller, the PIC24HJ128JP502, will be programmed to run the state machine on figure 1 and will send output voltage to set off the buzzer.

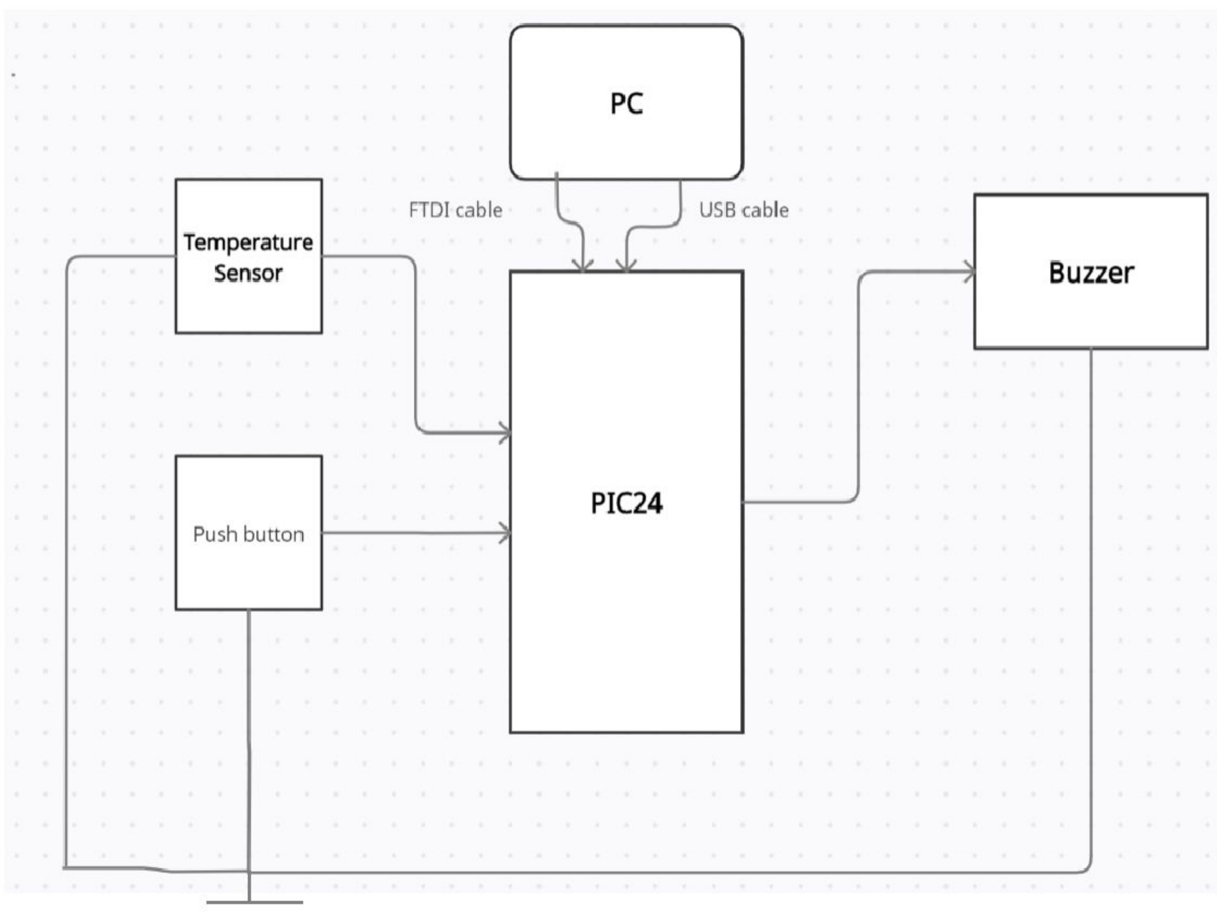


Figure 2: Block diagram of project 1

## Software

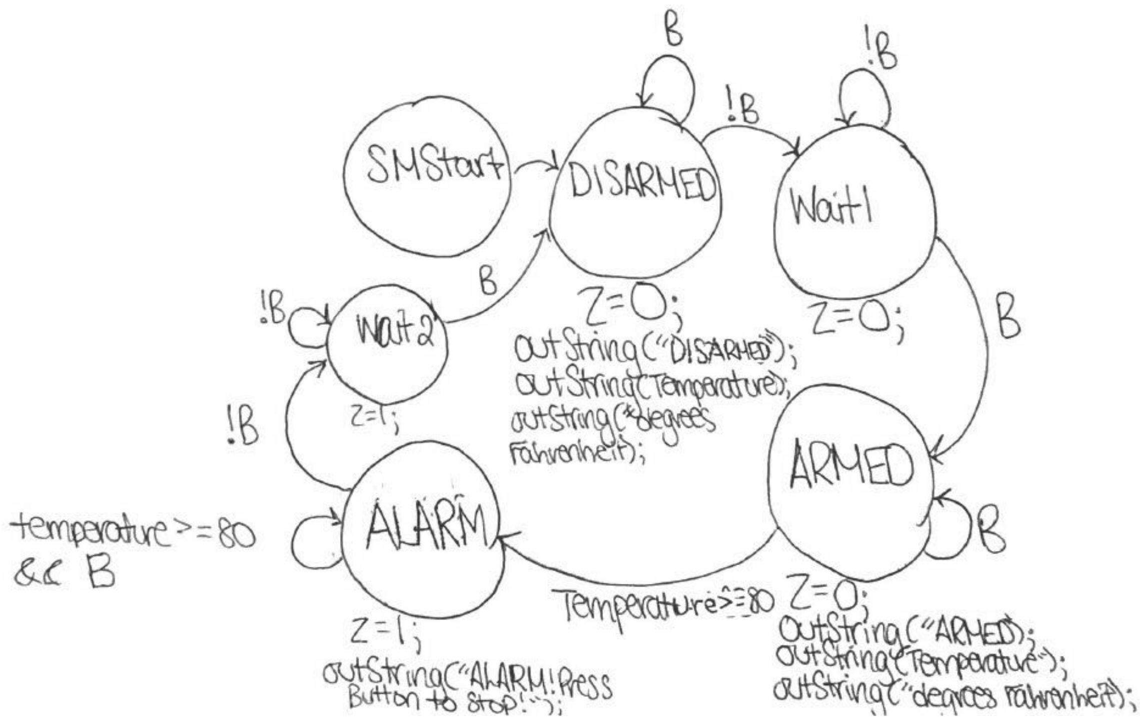


**Figure 3:** Detailed hand drawn HLSM

When the PIC24 reads the voltage input from the temperature sensor, it converts it to celsius values using equation 1 below. This is implemented in c-code through a function definition which is called multiple times before each message output.

$$V - V_1 = \left(\frac{V_2 - V_1}{T_2 - T_1}\right) \times (T - T_1)$$

**Equation 1**

There are two other functions defined in this program. One converting the celsius value returned by the previous function to fahrenheit units and the next arranges said units to a string type using arrays. The state transitions and their corresponding outputs are handled by two switch statements. Within each statement there are cases for each state. In the ARMED case an if statement checks for temperature readings above or equal to 80 degrees fahrenheit. Else it will remain within that state.

**Results**

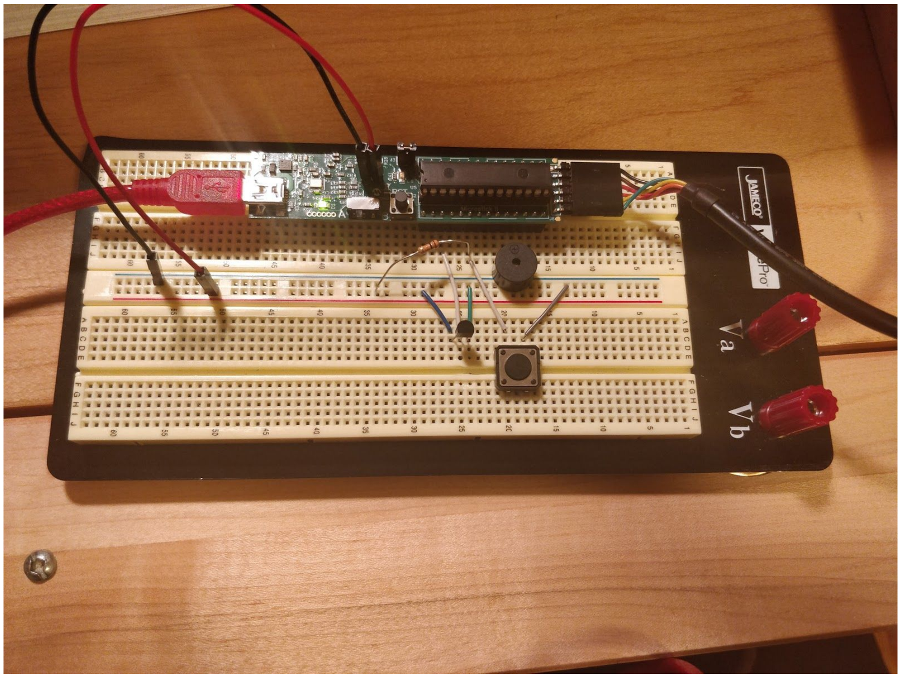

**Figure 4**: Picture of the Temperature Alarm system

Post reset, the PIC24 initializes in the DISARMED state and displays “DISARMED” message along with an initial temperature reading in the windows emulator, putty. Once the button is pressed by the user, the program will switch to the ARMED state which will also be displayed along with a temperature reading. Once armed the PIC24 will check for temperature readings above or equal to 80 degrees Fahrenheit. When the temperature sensor indicates such a reading, the PIC24 program will transition to the ALARM state which will simultaneously set off the buzzer. Here a message displays reading “ALARM! Press button to stop!”. If the button is pressed the program transitions back to the initial DISARMED state and turns off the buzzer.

The temperature readings obtained would occasionally fluctuate between 1 to 7 degrees fahrenheit. This wasn't enough to significantly affect the state transitions since the system was situated in a low temperature area (around 21 degrees celsius).

**Conclusion & Discussion:**

Implementation of this project meant full understanding of labs 1-3, which each addressed at least one of the aforementioned elements of this embedded microcontroller system. For example, Lab 3 was based around A/C conversion, which helped provide information on how to have the temperature display correctly on the monitor after being read from the temperature sensor. This step was one of the most important and yet difficult to implement in the project, so the background knowledge gained from Lab 3 was extremely helpful. Another requirement that was significant in the implementation of the project was the designing and coding of the state machine for the alarm. Without this element working properly, the entire embedded microcontroller system would not function at all.

**Appendix**

List of equipment:

1) Temperature sensor - lmt84
2) Buzzer - AI-1223-TWT-3V-2-R  size A3

3) Microcontroller - PIC24HJ128GP502
4) Simple push button switch
5) Datasheet Lmt84
https://nexus.union.edu/pluginfile.php/577093/mod_folder/content/0/temperature_sensor_lmt84-q1.pdf?forcedownload=1

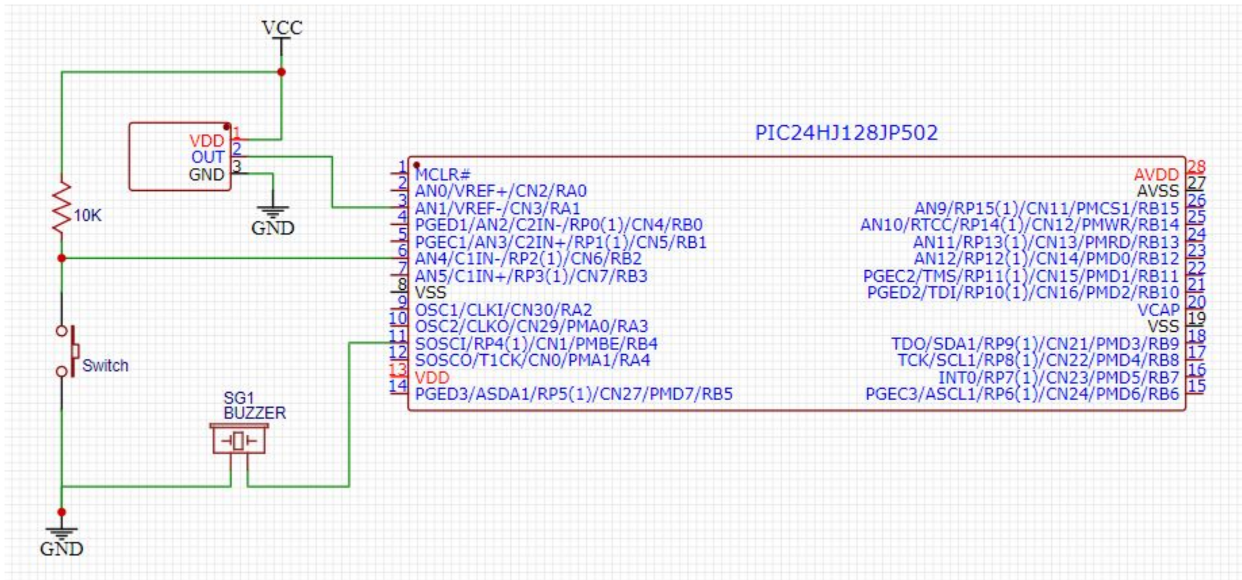


Figure 4: Project 1 schematic