Report

- Goal

  The main goal of this project is to make a small weather station. A formal weather station measures temperature, humidity, and wind direction and speed typically. I think the temperature and humidity of the weather are the two basic components and are easier to acquire compared to the other items. Consequently, my small weather station will contain the measure of the temperature and humidity.

  What's more, I also record the brightness of the environment and the distance of user. The brightness is to decide whether it needs to turn on the light or not. And the distance is in the purpose of if there's no one nearby, the whole station won't work.

- Component
  - ✓ led

    I use blue and white led in this project. White led acts as “cloud”, and blue led indicates “rain”. The whole white led is in series and operates refer to a single signal. The blue led is divided into five groups and each group is connected in series.

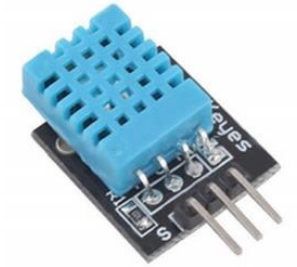
  - ✓ RGB led module

    This is used for displaying the temperature and act as the “sun”. If the temperature is higher, the value of red is increased and blue is decreased. On the contrary, the value of red is decreased and blue is increased when it is cold.

  - ✓ Temperature and Humidity Sensor Module(DHT11)

    DHT11 will return the temperature and humidity it receives. And then it sends the temperature to RGB led module to visualize the temperature. The humidity is an indication for the “rain” to decide whether to turn on or not.

  - ✓ photoresistor

    The photoresistor can get the brightness of the nearby environment. If the nearby environment is bright enough, the “cloud” won’t turn on. Otherwise, the darker the environment is, the brightness the “cloud” shines.

  - ✓ Ultrasonic Sensor

    The ultrasonic sensor will return the target object’s distance. I use it to detect whether there’s people nearby. If there is, the weather station will work normally. However, if there isn’t, the station will be in the sleep mode.

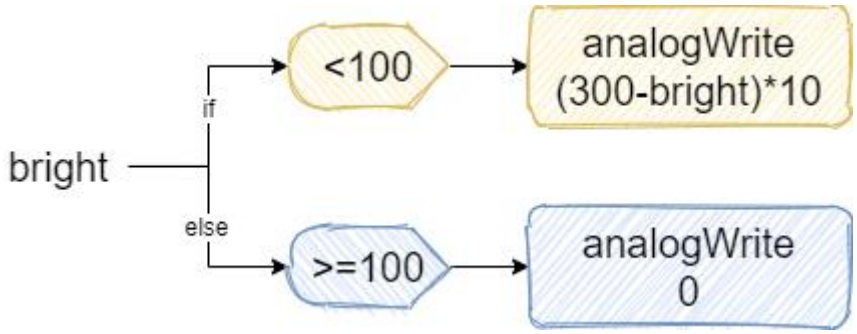
  - ✓ LCD

    LCD is convenient for us to observe each item. It will show the value of the temperature, humidity, and brightness.

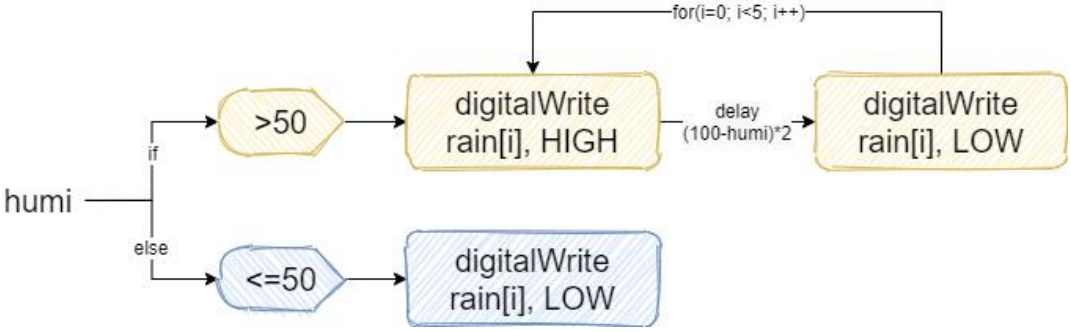
    

To enlarge the difference in color of the “sun”, I simply shrink the range of the temperature from 0-50 into 0-35 since it’s around 0 to 35 degree Celsius here in Hsinchu. As a result, the input temperature will be divided by 35 then multiply 255 to translate the temperature(0-35) into rgb(0-255).

The “cloud” turns on the light only when the value measured by photoresistor named “bright” is higher than 100. If it’s true, Arduino gives the “cloud” pin the value calculated by the formula (300-bright)/10. It makes that the darker the environment is, the brighter the “cloud” is.



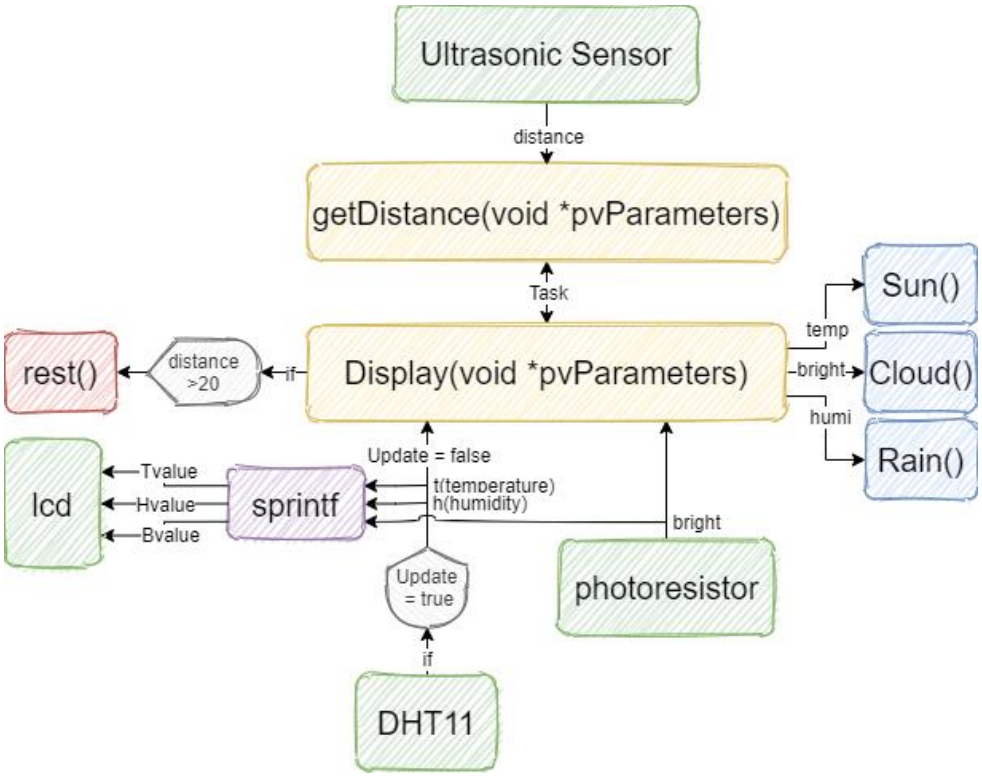
The “rain” which is controlled by the humidity works only when the value of humidity reaches 50. If it rains, 5 groups of the blue led turn on and off in turn. There’s a short delay between every on and off. The duration of lighting is based on the value of humidity. The delay time formula is (100-humi)*2. So, if we have the weather that contains higher humidity, the rainfall will drop rapidly to simulate the scene of heavy rain.



As for the LCD display, we first need to transform the int variable into char array using sprint in Arduino. Then we can display the value of each item on lcd correctly.

- Program flow chart

The detail diagram of this project is shown as the following picture. The functions “Sun(int temp)”, “Cloud(int bright)”, and “Rain(int humi)” are main for the clearness of the code. They receive signals from Arduino and perform the result accordingly. The function “getDistance()” and “Display()” are written in FreeRTOS. “getDistance()” will write the measure result to the global variable “distance”, And “Display()” will determine the station whether to work according to the variable ”distance”. If not, then call the function “rest()” to let everyone get into rest.



As for the ISR part, it is used for counting the time. Since the weather won’t have an enormous change in a short time, I let DHT11 measure only once in an hour. So, I declare two variables, “second” and “minute”. Every one second, the program will call ISR, and the variable ”second” pluses 1. If the “second” is equal to 60, the variable “minute” pluses 1 and the ”second” revert to 0. Same as “second”, “minute” will return to 0 if equal to 60. If it meets the condition that “minute” is equal to 0, the Boolean variable “Update” turns to be true.