

Final Project – Arduino Web Reader

(I can't think of a better word)

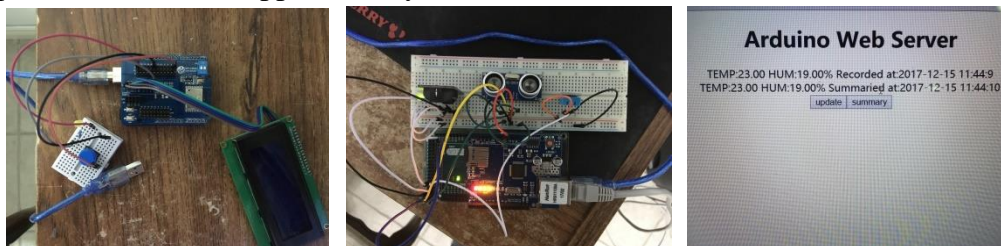
Member: Ye Yuan, Erik Moraru

Introduction:

This is it, the last project for this course. There're few ideas before I start to this project but either too complicate or the knowledge I have is not enough to do it. After checking the documentation of instruction and calculate the table I come up with this idea to get the data collected from Arduino to be accessible from web browsers. After searching the internet for a while, I found an example to build a controller for the led on the Arduino but not get the request from the Arduino itself but from a web request. And so I come up with an idea to use the DHT-11 sensor to collect the data and whenever the server is connected to a client it will show a web page and the data collected from the sensor, and also it will update the data immediately when the client press the button "update", it will show the new data on the page.

Overview:

The project was originally designed to be three parts: one Arduino sever, one Arduino client and a web client. But because of reasons (stated under in the regrets and problems) it was dropped to only the sever side and web site.



(Listed from left to right is the photograph of client side, sever side and website)

(Unfortunately client side was not working and does not used in the project)

The sever collects data from the DHT sensor and use the Ethernet board to send data with a HTML text to the browser. User at the client side can click the update button to update up to 2 seconds delay data from the DHT or press the summary data to get the average data of the day. If a day is past, the server will compute the average temperature and humidity of yesterday and write it to the sd card on it.

The client Arduino was supposed to use the timer and external interruption through the Wi-Fi to the server to get the data. The timer interruption set the Arduino send update to the sever every minute, and the external interruption is happening every time the button was pressed (it is an internal pull-up pin, so by press the button, it will get the signal to low). The Arduino then supposed to print the data received from the server on the 2004lcd display.

Findings:

Besides the Ethernet I also did the research of interruption. It took almost three

days, to finally making some sense to me. The external interruption is a fair game to understand and implement, but the timer one is a disaster. Many places only show you how to do the timer interruption without explaining like the thing is just there and you just know it. After more digging into the subject I finally find out in need to use the timer interruption, you need to specify the prescaler and then calculate how many times it has to tick to generate an interruption signal at a specified interval. Then the predefined function should be used to compare the tick stored in the register and the tick defined by me to see if the time is reached.

When using the timer it is necessary to check which things will be disabled after using the timer. Like `millis()`, `delay()` will be disabled after using the timer0 in a CTC mode. And servo library and pin 9 and 10 will be disabled after using timer1.

Some regrets and problem found:

My first idea is to set a client side and a server side Arduinos, The sever side response to the client, update data when received “update” or “summary” request, and send the data back to client whether it’s on web of Arduino. The web client will receive a HTML text message with the data collected from last update. The Arduino client side should send “update” request to the sever and send ”summary” when user press the button. Both of these sending requests were accomplished by the external and timer interruption. However the process is suspended whenever calling the lcd functions, the guessing is maybe the interruption is causing a conflict?

Another problem found at the server side is when timer interruption was added, the data collected from DHT-11 will always be a NAN, and cause the Arduino entering an infinity loop. Maybe the interruption is doing the same thing as in the client side, corrupting the data line?

I also had few ideas by adding Cds to detecting the sun and microphone sensor to check the wind but they were dropped in the end, only the one detecting whether the sensor was blocked is remained.

Another idea I did not accomplished is using an “history” request instead of “summary” so it will read the last line in the log txt file and print it on the web or Arduino. And a terminate button to terminate the whole process. But time is not enough and so they are also dropped.

Also because my knowledge to the internet is not enough, the sever can only be accessed within the local area network. Maybe later it can be revised to be able to access from anywhere.

The sample log txt file only have one line because the sever side was just finished at 2017-12-14 Thursday.

Adding up:

It’s easy to build a thing by instructions, but it’s hard to build a new thing from nothing. I do had better understanding how the embedded system working after all the research and coding. And the whole time it was very challenging to me, I suffered a lot in the researching. But once that everything is making sense, all I felt is awesome.

And as always, thank you Chris.