2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS) | 978-1-6654-0521-8/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICACCS51430.2021.9441845

# Design of Standalone Asynchronous ESP32 Web-Server for Temperature and Humidity Monitoring

Paul Macheso[ad*],Sylvester Chisale [b],Chisomo Daka[b],Nelson Dzupire[c],Justice Mlatho[d] and Didacienne Mukanyirigira[a]
[a]African Center of Excellence in Internet of Things (ACEIoT), University of Rwanda,
College of Science and Technology, Kigali, Rwanda.
[b]Department of Applied Studies, Malawi University of Science and Technology (MUST), Limbe, Malawi.
[c]Department of Mathematical Sciences, Chancellor College, University of Malawi, Zomba, Malawi.
[d]Department of Physics, Chancellor College, University of Malawi, Zomba, Malawi.
[*]email: paulmacheso@gmail.com

***Abstract*—This article describes the design process for a low cost asynchronous web-server based on ESP32 interfaced with DHT 22 sensor for monitoring temperature and humidity in real-time. Environmental management is a key research area where many strategies are being put by researchers in order to control and manage temperature and humidity in order to keep processes running hence there is need to develop a low-cost and power consuming web server to monitor real time conditions of temperature and humidity in various settings.**

***Index Terms*—ESP32, Web Server, DHT 22, Temperature, Humidity ,Access Point.**

## I. Introduction

Asynchronous web servers store, process, and distribute web pages to web clients in real time [1]. Web clients are web browsers on computers and smartphones. A special protocol called Hypertext Transfer Protocol (HTTP) is used to communicate between the web client and the web server. In order to analyse, examine or make decisions regarding environmental parameters of temperature and humidity, a real-time monitoring system is required. [1-2]. For, example to have an overview scope of temperature and humidity in a certain setting the prototype can be employed to undertake the desired tasks, such as monitoring of environmental parameters is very important in various industrial processes and applications.

The requested services for web clients and the hosting of websites can be provided by the web server. Through a router and the internet, web clients can link to the web server.. This is a low-cost web server solution which can store data locally and is accessible through a section of web clients. The client server architecture makes a significant contribution significantly to the growth of the Internet of Things devices which have taken the talk in computing world [3]. Figure 1. Shows a typical client -server architecture where if a web client wants to access the server, a request is sent to the server.

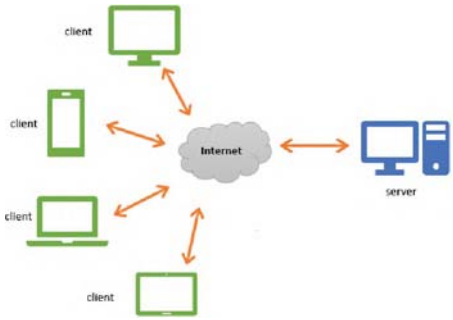


Fig. 1. Client Server Architecture [3]

## II. Existing Literature

There is growing interest in the research of embedded web-servers which are to be used in the development of IoT devices. In [3] there was a proposition of refining a design of a previously developed system for temperature and relative humidity. The prototype was developed using Arduino Uno, SHT 11 and Ethernet Shield boards. The system merits included working in stable and accurate conditions but it had a drawback of the need every time to refresh the webpage on the client side every time to have current sensor data. The microcontroller boards which were used were expensive and for connection to a network an ethernet cable was required which would prove a big challenge to connect in remote areas. Furthermore, the sensor node consumed more power.

In [4] the author presented some logging of temperature and humidity sensor data which was displayed on thing speak cloud server platform in near real time. The system was developed with NodeMCU ESP8266 interfaced with DHT 11. The merits of the developed system is that there was no prediction of the sensor data of any environment and low

978-1-6654-0521-8/21/$31.00 ©2021 IEEE

power consumption. The drawbacks of the proposed system involved the usage of third-party cloud servers which are prone to attacks and also the sensor which used was less accurate.

The proposed system in this paper is to embed a reliable standalone asynchronous web server for temperature and humidity measurement in ESP32 microcontroller which is reliable and accurate to be used for various uses.

## III. Objectives and Contribution

The main objective of this research study is to design a standalone asynchronous ESP32 web server for temperature and humidity monitoring and its uplifted by the following specific objectives.

- Reading temperature and relative humidity from DHT22 sensor to any web client
- Designing a responsive web dashboard interface for displaying the sensor data
- Development of an asynchronous ESP32 web server
- Calculating the significant power usage of the ESP32 web server integrated with the DHT22 sensor

## IV. System Design

### A. NodeMCU-ESP32

ESP32 is a low-cost and low-power open-source microcontroller with integrated WI-FI and dual-mode Bluetooth [3]. The ESP32 employs a Tensilica Xtensa LX6 microprocessor in both dual-core and single core variations.The specifications of NodeMCU-ESP32 are shown in table I.

| Specifications | NodeMCU-ESP32 |
|---|---|
| Operating Voltage | 2.2 to 3.6 V |
| GPIO | 36 ports |
| ADC | 14 ports |
| DAC | 2 ports |
| Flash Memory | 16 Mbyte |
| SRAM | 250 Kbyte |
| Clock Speed | Up to 240 MHz |
| WI-FI | 2.4 GHz |
| Sleep. Current | 2.5 µA |

TABLE I: Specifications for NodeMCU-ESP32

### B. DHT 22

The DHT22 is a digital temperature and relative humidity sensor [7]. This sensor has an A-D converter chip that converts analog to digital and outputs a digital signal with temperature and humidity sensor data. This makes this sensor to be easy to use with any microcontroller, including the ESP32. The specifications of the DHT22 are given in the table II.

| Specifications | DHT 22 |
|---|---|
| Communication Protocol | One-wire |
| Power supply Range | 3 to 6 v |
| Temperature Range | -4 to 80 °C (+/- 0.5 °C) |
| Humidity Range | 0 to 100 % (+/- 2%) |
| Sampling Method | 2 Seconds |
| Arduino Libraries | Adafruit DHT library |

TABLE II: Specifications for DHT22

## V. System Implementation

### A. Arduino IDE

It is an open source Arduino software which is a place to write instructions which form a program or a sketch for the Arduino boards and test them out [1]. It makes code writing easy and allows sketch uploading to the boards. It runs on Windows, Linux and Mac OS X.

In this study some factors were taken in consideration before sketching the programming ESP32. Firstly, addition of ESP32 package to the IDE. Secondly, addition of WI-FI library.

Finally, the DHT and Adafruit Unified Sensor libraries are being used to read the data from the DHT 22 sensor, and the ESPAsyncWebServer and Async TCP libraries are being used to build an asynchronous web server.

### B. HTML Script, CSS and JavaScript

The program sketch has an inclusion of HTML (Hypertext Markup Language). A format which commands the computer how to display the webpage [3]. The web page was styled using CSS (Cascading Style Sheets). Finally, in the program sketch there is some JavaScript which make the webpage responsive and updates the temperature and humidity approximately every 10 seconds.

## VI. Technology Description

### A. Circuit Schematic Diagram

The circuit schematic diagram outlines step by step how to interface the DHT22 with the ESP32 microcontroller. In this scenario the data pin of the DHT22 is connected to GPIO 27 of the ESP32. The VCC and GND of the DHT 22 are connected to the ESP32 supply voltage and GND respectively. Finally, the program sketch is flashed into the ESP32S microcontroller using the Arduino IDE.

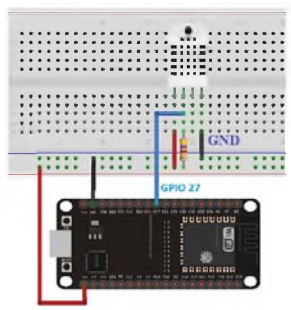


Fig. 2. Circuit Schematic DHT22 interfaced with ESP32 [1]

### B. Setting the ESP32 Web Server Access Point (AP)

The ESP32 acts in 3 modes as an Access Point [AP] as a WI-FI station or both Access Point and WI-FI station [7]. In this paper the ESP32 is set as a WI-FI station and the router as an access point as shown in Fig. 3. To control the ESP32 there is need to connect to the local network.

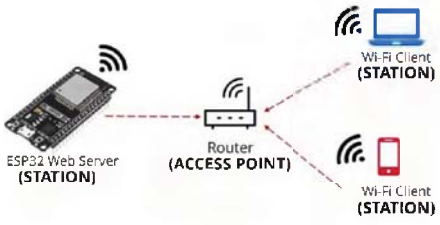


Fig. 3. ESP32 as a WI-FI station [1]

## VII. Development

### A. Simple System Architecture

In this paper on the design of standalone asynchronous ESP32 web server for temperature and humidity monitoring is presented. The web client can access the temperature and humidity data via web browser application through an http request to the server built on top of the ESP32 [1-2]. The ESP32 program sketch serves the request and reads the temperature and humidity data from the DHT 22 sensor module. The standalone asynchronous ESP32 server sends a response by sending HTML with temperature and humidity data. The simple architectural development of the system is depicted in the Fig. 4.

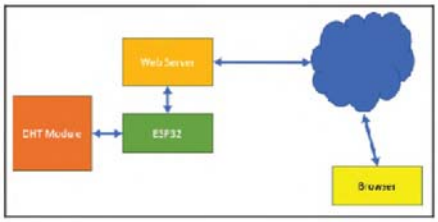


Fig. 4. Simple System Architecture

### B. Prototype Development

The desired prototype is designed with a perfect hardware interface as shown in Fig.5.The DHT 22 was interfaced with the ESP-32 were the web server was running.

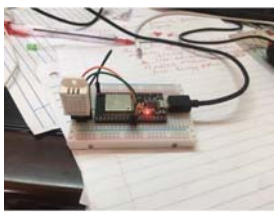

Fig. 5. Prototype Design

## VIII. Results and Discussions

### A. Implementation Results

To get the results on a web page the IP addresse must be identified by opening the serial monitor on the Arduino IDE. After that the IP address (198.168.8.101) is punched in in any web browser of any web client connected in the same network. Fig.6. shows two different web clients accessing web page using the same IP address displaying sensor data of temperature and humidity automatically eliminating the need to refresh the webpage.

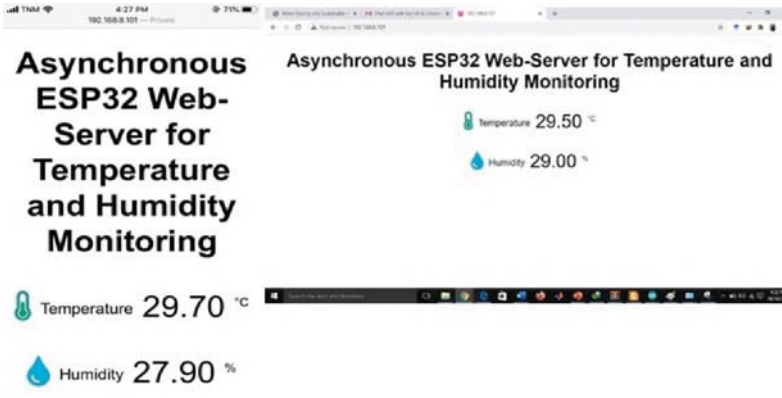


Fig. 6. Web clients connected to the ESP32 Server

### B. Power Consumption for the Design

The power consumption of the devices in this work is analysed individually and found by measuring using multimeter to calculate current and voltage. This is to calculate the full overall power consumption of the standalone web server monitoring temperature and humidity in real-time. Table III shows the measured power consumption of both the ESP32 and DHT22.

| Device | Voltage Drawn | Current Drawn | Power |
|---|---|---|---|
| ESP32 | 3.3V | 150 mA | 495mW |
| DHT22 | 3.3V | 2.5 mA | 8.25mW |
| Overall Total Power Consumption | | | 503.25mW |

TABLE III: Current and Voltage drawn by the devices

## IX. Business Benefits

Many developing countries are of developing tailor custom made temperature and humidity monitoring solutions in Internet of Things and embedded systems. The motivation of conducting this research is to meet the expansion in the IoT technology across the digital divide that is often constrained by costs of infrastructure. The induction of these standalone asynchronous web servers for temperature and humidity measurement will increase efficiency and reduce latency, reduce interference and packet loss and help digitize Africa as more billion devices will be connected to the internet.

## X. Conclusion and Future Scope

The main aim of this paper was to design a standalone asynchronous web-server for temperature and humidity monitoring using ESP32 microcontroller and DHT22 I. The ESP32 acts in dual mode as a sensor node with its GPIO interfaced with the DHT22 as well as a standalone web server where web clients can easily access sensor data.

This article also presents ways of using ESP32 to migrate from using 3rd party web servers to standalone web servers which reduce latency and packet loss. The total power consumption of the system was about 503.25 mW which was needed to design the whole system and be able to monitor sensor data.

The further scope of work could involve eliminating the WI-FI router access point and creating a soft access point on the ESP32 microcontroller whereby web clients (stations) can connect and retrieve sensor data. Other future plans for this project include integrating measuring and data collection capabilities by adding other environmental monitoring sensors and extending online data analytics with internet graphs on the web page of the web client.

## Acknowledgment

The authors would like to acknowledge the University of Malawi, Chancellor College, Physics Department for allowing the project to run through their premises and providing internet support to the sensor node as it was deployed.

## References

[1] Available Online https://randomnerdtutorials.com/esp32-dht11-dht22-temperature humidity-web-server- arduino-ide/ retrieved on 20th October 2020

[2] Simić, Mitar. "Design and development of air temperature and relative humidity monitoring system with AVR processor based web server." 2014 International Conference and Exposition on Electrical and Power Engineering (EPE). IEEE, 2014.

[3] Allafi, Ibrahim, and Tariq Iqbal. "Design and implementation of a low-cost web server ESP32 for real-time photovoltaic system monitoring." 2017 IEEE electrical power and energy conference (EPEC). IEEE, 2017.

[4] Deekshath, R., Dharanya, P., Kabadia, M. K. D., Dinakaran, M. G. D., and Shanthini, S. (2018). IoT based environmental monitoring system using arduino UNO and thingspeak. International Journal of Science Technology and Engineering, 4(9).

[5] Available online https://randomnerdtutorials.com/esp32-access-point-ap-web-server/ retrieved on 20th October 2020

[6] Roy, V. B. R., Dessai, S., and Yadav, S. S. P. (2009). Design and development of ARM processor-based web server. International Journal of Recent Trends in Engineering, 1(4).

[7] Available Online https://randomnerdtutorials.com/esp32-dht11-dht22-temperature-humidity-sensor-arduino-ide/

[8] Temperature Sensor https://randomnerdtutorials.com/9-arduino-compatible-temperature-sensors-for-your-electronics-projects/

[9] https://lastminuteengineers.com/esp32-dht11-dht22-web-server-tutorial/

[10] Khaing, Khin Kyawt Kyawt. "Temperature and Humidity Monitoring and Control System with Thing Speak." International Journal of Scientific Research and Engineering Development (2019).