IoT Coffee Maker

CS480

Haluk Gavas
Binghamton University
hgavas1@binghamton.edu

ABSTRACT

Automation and IoT integration of home appliances and other devices can bring many benefits to our lives. From remote control to automated behavior the applications for these kinds of devices are endless. However, there is one roadblock in this advancement, money. Even though IoT-enabled appliances are superior to their simple counterparts they are also considerably more expensive. Not everyone can own an IoT-enabled device. However, unlike buying expensive IoT-enabled appliances building a DIY solution is considerably cheaper and sometimes more effective. With just a few IoT devices people can convert their simple and boring appliances to IoT devices with great functionality.

KEYWORDS

Insert keyword text, Insert keyword text, Insert keyword text, Insert keyword text

1 Introduction

IoT integration of home appliances provides great benefit to our lives. Due to financial limitations, not everyone can benefit from this technology. However, IoT integration can be achieved by using DIY solutions. One example of this is how I transformed my simple Coffee maker into an IoT coffee maker with water sensors and app integration. In this report, I will describe my experience and findings during my development process.

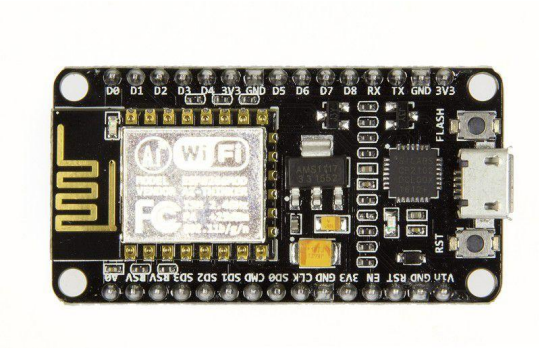
2 Motivation

My main motivation with this project is to demonstrate how even the most basic devices could be improved with the help of IoT. In my case enjoying a cup of fresh coffee is a must of my mornings. However, waking up from the bed and making coffee while being half asleep is a hassle. So I wanted to build something that could automatically make my coffee before I wake up. So I ended up creating an IoT-based coffee maker setup that integrates with my phone Alarm and makes me coffee when my phone alarm goes off.

3 Hardware

My hardware consists of three components:

1. Nodemcu ESP8266: An SoC with wifi capabilities that will be connected to the web server which will connect to the app via the Nodemcu firmware.

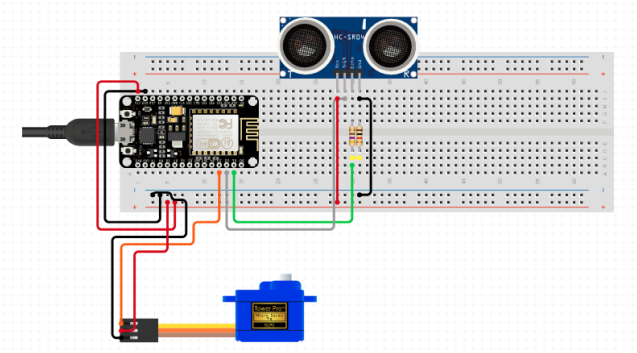


2. HCSR04 ultrasonic sensor: An ultrasonic sensor that measures the distance between the sensor and the water. It will be used to determine the water level.

3. Servo Motor 5g:A motor that activates coffee maker.

The nodemcu acts as my main logic board and it's responsible for sending and receiving data from the web server to the application as well as controlling the servo motor behaviour. The HCSR04 ultrasonic sensor is used to calculate the remaining water amount inside the water tank of the coffee maker. The servo motor is used to trigger the coffee maker controls and start the device.

4 Design



My hardware connection is like the diagram above. The nodemcu is connected to a power supply via a micro-USB cable. The hcsr04 is connected to the Nodemcu with VCC, ECHO, TRIG, and Gnd pins. The servo is connected to the nodemcu via D1 and D2 pins. After that, this setup is integrated into the coffee maker. The ultrasonic sensor is placed inside the water tank to observe the water level, the servo motor is fixed on top of the coffee maker to press the buttons and the nodemcu is connected to the side of the coffee maker to control the device.

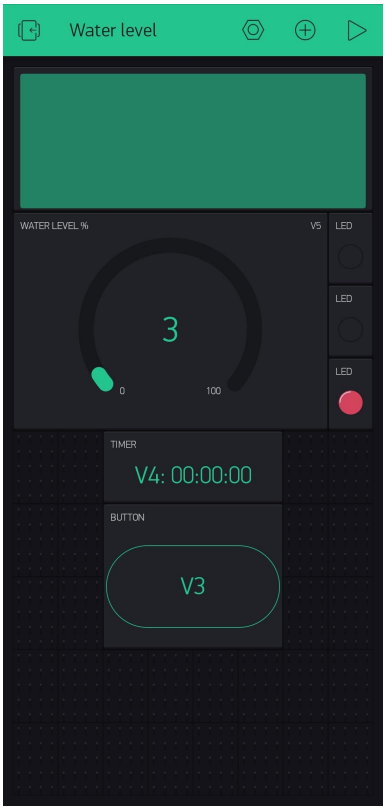
5 Implementation

The code implementation of the project is done in Arduino ide. During the development of the libraries. BlynkSimpleEsp, esp8266, HCSR04, and Servo were used. These libraries are used to provide functions for my respective hardware devices. I also used the Blynk app while doing app integration with my Nodemcu.

The first thing I did when I was doing my implementation is to set up Nodemcu. I created virtual pins and attached those to the servo motor and ultrasonic sensor. After that, I created a function that will make the servo motor trigger the coffee maker. After that, I created a function that will calculate the remaining water percentage of the water container and then send it to the application via the web server. I accomplished this with the following formula:

```
duration = pulseIn(ECHOPIN, HIGH);
distance = (duration/2) / 29.1;
distance = 12 - distance;
distance = (100-(distance*100)/12);
```

After the setup for the hardware and the Nodemcu software was complete I moved on to create the Blynk application. The Blynk application has a few components that are used to visualise data I am getting from the Nodemcu and send commands to the Servo motor via the web server.



Here is my Blynk application. As I explained it includes a few components. These are:

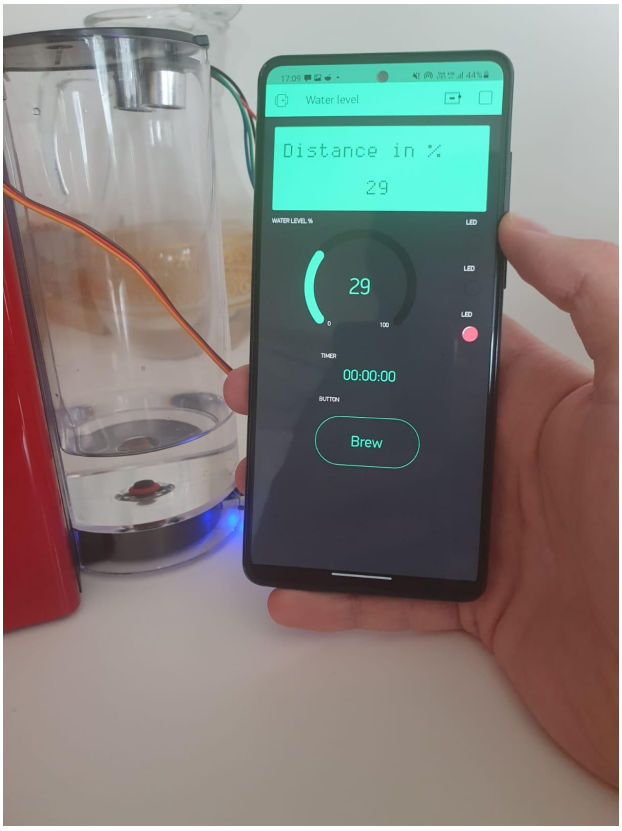
1. An virtual lcd and gauge component to showcase the water percentage inside the coffee maker.
2. 3 virtual led's to show the water level percentage. Blue for over %70, green for %70-%30, red for below %30.
3. A timer to set a timer of when the coffee maker should start.
4. A button for starting the coffee maker manually without a timer.

6 Usage

The usage scenario for the coffee maker is pretty simple:

1. Open the Blynk application.
2. Check to see if the water level is enough by looking at the water level indicators
3. Set a timer or press the button to start the coffee maker

When started the Blynk applications starts receiving signals from the HCSR04 connected to the Nodemcu via Webserver.



As can be seen in the image above the HCSR04 correctly measures the water distance and then sends the data to Nodemcu. The data then is processed in the Nodemcu and used to calculate the remaining water percentage which is then sent to the application display via the web server.

After hitting the Brew button or setting up a timer like in the image above, the coffee maker either starts immediately or starts at the specified time depending on your choice.

7 Limitations

The biggest limitation of this application is refilling the coffee maker. Even though the coffee maker starts and stops automatically and tells you the remaining water level, there is no way of refilling the coffee or water automatically. This means the application needs some form of human interaction and is not autonomous. The application also does not have any error handling features. For example if the coffee capsule is stuck in the machine the application has no way of knowing this or notifying the user about it.

8 Future Work

Even though the core aspects of this project is done I am planning to continue working on it incrementally. I have a few component upgrades that requires me to get additional hardware. The potential upgrades I am thinking of adding are:

1. Connect a water supply via water pump to automatically refill water.
2. Create a setup for refilling coffee.
3. Connect Nodemcu webserver with my Amazon account to order coffee when the machine is running low on coffee

Conclusion

This paper demonstrates how to connect servo motor nodemcu and an ultrasonic sensor to create a IoT enabled coffee maker. The main objective of this project is to demonstrate the usability and practicality of DIY IoT solutions in an individual's life. Not everyone needs a lot of money to buy and use appliances with IoT technology. With enough time, effort and research everyone can experiment with IoT technologies at home.

References

https://nodemcu.readthedocs.io/en/release/
https://docs.blynk.cc/
http://web.eece.maine.edu/~zhu/book/lab/HC-SR04%20User%20Manual.pdf