**THE PROPOSED CLOUD ENABLER COFFEEMAKER (CEC)**

The CEC is a novel application that combines embedded and a cloud system, by controlling the coffee maker machine over the internet through the use of an android application. The mode of operation involves of locating users GPS coordinate to start up the coffee machine, or through the android application. Thez\ starts from the android application, which sends the different states the cloud server, based on GPS location or the android button state, which updates the database state of the cloud system. The arduino device, through and external shield (1Sheeld), performs a concurrent pull check from the server, and instantiates a startup operation if state on the online database match up with the start state of the application.

**Hardware Components:**

The hardware components listed below are used in developing the CEC application.

* 1sheeld
* Mobile phone (Android O/S 4.0 and above)
* Coffee machine
* Arduino UNO R3
* Power tail II (Input Switch 3-15 v dc)

**Software Components:**

* Android source code
* PHP source code
* Arduino source code
* MYSQL code

**Description of Hardware Components:**

1. 1Sheeld: This device serves the purpose of eliminating the need to use different sheeld using moreto reduce the reliability on adding external shields to the arduino, by simply extending the phone features to the device [1].

The 1sheeld is an Arduino shield for Android that allows mobile devices to be utilized as shields when connected through Bluetooth. This takes away the need to purchase various shields such as GPS and Ethernet shields.

This is a hardware device that extends the functionality of the Arduino microcontroller by connecting to a mobile application to use the functionalities of the mobile phone such as the WI-FI, GPS, GSM, LCD screen and all other features of the phone. Mobile phone (Android O/S 4.0 and above): This extends the features on the Arduino through the 1sheeld device by using an application that opens the functionalities of the mobile phone to the Arduino through a Bluetooth connection.

1. Coffee machine: This is the end product that triggers itself on through the Powertail device to start making coffee. This machine have a on and off button and makes coffee when plugged in to a 120 volt power.
2. Powertail II: An isolated dc actuated power cord (NO or NC) for controlling power to 120vac appliances with a Microcontroller[2]. The relay produces 120v AC from low voltage as low as an input of 3 – 12 vdc. This device gets triggered on through the input received from arduino.
3. Arduino UNO R3: The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button [3].

**Description of Software Components:**

1. PHP source code: This contain http request which is referenced by both the android application and the Arduino device. This online server provides various state and a database that maintain the state of communication between the android application and the coffee maker.
2. Arduino source code: The source code on this platform through the extension of the 1sheeld device enables request calls to be made over the internet to a remote server to get the state on the server. A thread exists that runs infinitely by constantly requesting for state change from the server. If the state on the server matches up with a start state, this triggers on the power 1sheeld by sending in low voltage, and in turn, starts up the coffee machine device.
3. Android source code: An application is developed using android programing language (java), to communicate with an online repository through a HTTP protocol to update the database content. It performs a PUSH call to the server, by updating various states to be consumed by the arduino device.
4. MYSQL code: The various states of the android application are stored in MYSQL database system, which is updated by the mobile application when state is changed to start and by the arduino when coffee maker machine is started.

**IMPLEMENTATION**

**Mode of Operation:**

A mobile application is developed to send status code to the cloud and send GPS coordinate as well. The coffee application can be started using two approaches, the startup option on the developed android application and the use of GPS coordinate. The cloud system is the intermediary between the android application and arduino device to which request and response called is initiated on.

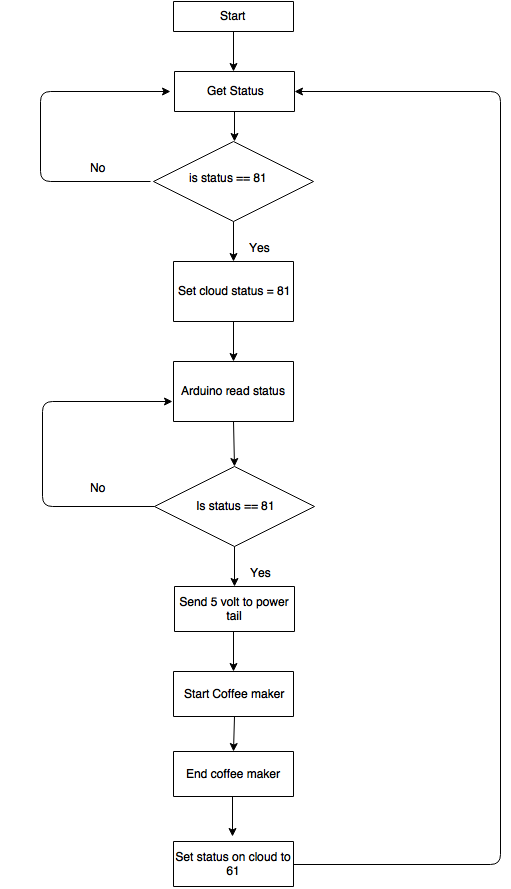
This option contains two button named “Make Coffee” and “Reset Coffee”. The make coffee option helps initialize the coffee maker option to send a signal to the arduino in order to power up the coffee machine. And the reset coffee option is used to ensure the coffee is refilled with both water and extra coffee after a coffee have been made previously. The modes of communication are in three states, named “61”, “71” and “81”. State “71” represents the ready state of the coffee machine which implies that the machine is in a ready state to make coffee. State “81” is used to send signal to the server which signifies to the arduino to start making coffee. Finally state “61” is the unready state of this system, which is initiated immediately after state “81”. This mode ensures that the coffee machine is refilled before initiating another system to make another coffee. To clear off this state, a reset button can be initialized to change the state from “61” to the ready state of “71”. The GPS module is the second option of initiating the coffee maker machine by constantly reading the location of the mobile device, and initiating the system to start making coffee when close to a particular region or location. The cloud system is the intermediary between the android application and arduino device to which request and response called is initiated on. This module consists of the request and response web pages and the database system which handles the storage of various states. The mode of communication is through HTTP (Hypertext transfer protocol). Arduino code consists of an infinite loop which constantly listens to the state of the system and the GPS coordinate from the cloud system. Operation only takes place when a change in state is initialized to start or when the GPS coordinate is within range. If the status is changed to make coffee or GPS is in range, the arduino device sends a HIGH signal to port 13 which in turn sends a 5 volt signal to the powertail. The code runs for 300,000 milliseconds (5 minutes), before it stops operation and starts listening for status from the cloud. This device receives low power voltage and converts to 120 volt, which is high to power up a coffee maker. In the developed system, the powertail serves the purpose of a relay, and is triggered on by the arduino device.

**Steps to start up system with android application:**

1. Launch the android application.
2. Log in to the system.
3. Click on Reset button, and refill the coffee machine.
4. Click on button to start the coffee machine.

**Steps to start up coffee machine using GPS navigation:**

1. Set location in range to map out distance to initiate starting coffee maker.
2. Walk into location assigned by device.
3. Arduino device repeatedly listens and map location of user to determine if in range.
4. If GPS coordinate maps range coordinate, initiate coffee startup.

Flowchart:

Issues Encountered:

* Arduino connection to the server: As explained above, the arduino device receives different states from the server.
* GPS Issue:
* Coffee Maker

Constraint:

Arduino: The coffee maker is designed to with limited water and coffee capacity intake. The goal of CEC is to make coffee over the internet, without the physical presence of a user. This was achieved in this project, but this is based on the assumption that the coffee machine is filled with water and coffee. The constraints of this work include the system’s inability to detect when the coffee machine is low on either water or coffee. The second limitation is in relation to the GPS coordinate, as the longitude and latitude value differ slightly due.

Server

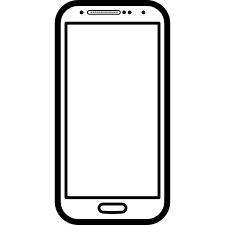
The use of power

Architectural diagram:



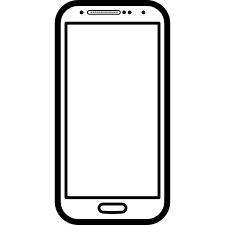
HTTP Response

HTTP Database Request



1 Sheeld

Arduino UNO R3



Power on {120 V}

Power on {3,5 V}

Bluetooth connection

Power Tail II

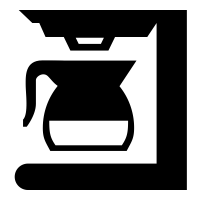


Figure 1, [4,5,6]

References:

1. "1Sheeld - Replace Your Arduino Shields With Smartphone". *Seeed Studio Bazaar*. N.p., 2016. Web. 10 Apr. 2016.

<http://www.seeedstudio.com/depot/1Sheeld-Replace-your-Arduino-shields-with-smartphone-p-1870.html>

1. <http://www.powerswitchtail.com/Documents/80136%20PST%20II%20Instructions%202015-11.pdf>
2. <http://www.farnell.com/datasheets/1682209.pdf>
3. "The Noun Project". *The Noun Project*. N.p., 2016. Web. 10 Apr. 2016.

<https://thenounproject.com/term/coffee-maker/21129/>

1. "Unlimited Web Hosting, Reseller Hosting, Domain Names, Cloud VPS SSD Hosting - Icestorm.Com". *Icestorm.com*. N.p., 2016. Web. 10 Apr. 2016.

<http://www.icestorm.com/>

1. "Mobile Phone Popular Model Samsung Galaxy S4". *Freepik*. N.p., 2016. Web. 10 Apr. 2016.

<http://www.freepik.com/free-icon/mobile-phone-popular-model-samsung-galaxy-s4_734341.htm>