

Smart coffee maker

Houbo Ni

Department of Electrical Engineering
Blekinge Institute of Technology
Karlskrona
Sweden
2017



in cooperation with



Abstract

Nowadays, more and more people drink coffee not only from Starbucks or other shops, but also brew coffee from coffee maker which with proper sizes which they can be put in the resting room of companies or homes.

The products of coffee maker polarize in the market. Some of them have powerful functions and nice tastes with high prices. The others are simple, cheap but losing enjoyment of drinking coffee and people need to keep an eye on status of coffee maker. Most of people prefer the cheaper type. But more functions will take more convenience.

Actually users can have them both. In this thesis, I design some functions based on the cheap type of coffee maker which will make drinking coffee more convenient and enjoyable.

I use Arduino Uno as my control system, cooperating with sensors and special circuits. My system monitor working time of coffee maker, temperature of coffee and water level of both coffee pot and tank in the coffee maker. Then users can receive these values from their cell phone and doing remote control. Meanwhile considering about prices, I choose the most suitable sensors for measurement. I also do some jobs for saving energy and safe. The system locks or turns off the switch automatically for them. At last, all the sensors, wires and control chip are put in good positions in the maker.

Keywords:

Arduino, Cheap, Convenience, Monitor, Remote control, Smart coffee maker

Acknowledgements

Firstly, I express my great thanks to my supervisor Kristian Nilsson. He gave me great supports with my thesis work and professional guidance.

Secondly, I am very appreciating for my 3+1 project manager professor Sharon Kao-Walter. Without her effort, I would not come to BTH and enjoy the study here.

Third, I also want to say thank you to my friends and classmates who gave me nice ideas, especially Eleni Koutrouli, Jiayu Shen, Yuchi Kang, Nan Pan and Emirhan Kutlu

Houbo

Contents

Abstract	2
Acknowledgements	3
Contents	4
List of figures	5
List of tables	6
List of symbols	7
List of acronyms	8
1 Chapter: Introduction	9
2 Chapter: Survey of related work	11
3 Chapter: Problem statement, objectives and main contribution	15
4 Chapter: Solution	16
4.1 Circuit	16
4.2 Working status	16
4.3 Timer.....	19
4.4 Monitor temperature	19
4.5 Liquid level measurement.....	21
4.5.1Coffee level in the coffee pot.....	21
4.5.2Water level in the coffee maker tank	22
4.6 Power consumption	24
4.7 Communication with cell phone	26
4.8 Placement for sensors and circuits.....	29
4.9 Total price for coffee maker	31
5 Chapter: Conclusion and future work	32
6 Chapter: Template description	33
Reference	33
Appendix 1: Codes of Arduino	35

List of figures

Figure 1-1 World coffee consumption (data from ICO)	9
Figure 1-2 Tower T13001	9
Figure 1-3 E6 from Jura	9
Figure 2-1 Coffee maker	11
Figure 2-2 Heating system	11
Figure 2-3 Arduino UNO	12
Figure 2-4 Webpage control Arduino	14
Figure 2-5 Ethernet shield	14
Figure 4-1 Circuit	16
Figure 4-2 Remote controller	17
Figure 4-3 Judging whether coffee pot in the maker	18
Figure 4-4 Mechanical switch	18
Figure 4-5 Mechanical switch for balance	19
Figure 4-6 DS1822	20
Figure 4-7 Circuit for DS1822	20
Figure 4-8 Sensor position	21
Figure 4-9 Brightness sensor	22
Figure 4-10 Positions of brightness sensors	22
Figure 4-11 Ultrasonic sensor with box	23
Figure 4-12 Position of Ultrasonic sensor	24
Figure 4-13 Circuit for power measurement	25
Figure 4-14 Working	27
Figure 4-15 Coffee pot is not in the maker	27
Figure 4-16 Coffee pot is the maker but no coffee inside	28
Figure 4-17 Maker lays down and maker don't work	28
Figure 4-18 Circuits board	29
Figure 4-19 Placement for sensors and circuit board	30

List of tables

Table 1-1 Different coffee maker	10
Table 2-1 Technical data about coffee maker.....	12
Table 2-2 Technical data about Arduino Uno	12
Table 4-1 Technical data for remote control switch.....	17
Table 4-2 Technical data for DS1822.....	20
Table 4-3 Technical data about Ultrasonic sensor.....	23
Table 4-4 Technical data about LA55P	24

List of symbols

Symbol	Quantity	Unit
<i>m</i>	mass	g(/kg)
<i>V</i>	volume	l(/ml)
<i>U</i>	voltage	V
<i>p</i>	power	W
ρ	density	kg/m ³
<i>g</i>	gravitational acceleration	N/kg
<i>h</i>	height	m
<i>p</i>	pressure	Pa
<i>f</i>	frequency	Hz
<i>T</i>	temperature	°C
<i>I</i>	current	A
<i>U</i>	voltage	V
<i>l</i>	length	cm

List of acronyms

Acronym	Unfolding
ICO	International Coffee Organization
HTML	Hyper Text Markup Language
XML	Extensible Markup Language
IP	Internet Protocol
CSS	Cascading Style Sheets
SRAM	Static Random-Access Memory
AC	Alternative Current
DC	Direct Current
WIFI	Wireless Fidelity

1 Chapter: Introduction

Coffee becomes an irreplaceable drinking in our daily life, especially for people who need to refresh themselves and improve working efficiency. According the data from ICO[1], the consumption of coffee in the world is increasing. When people want to drink a cup of coffee and they are staying at home or busying with their jobs, it is unreasonable to let them go to shops or Starbucks to buy. So the better choice will be brewing coffee by home using coffee maker.

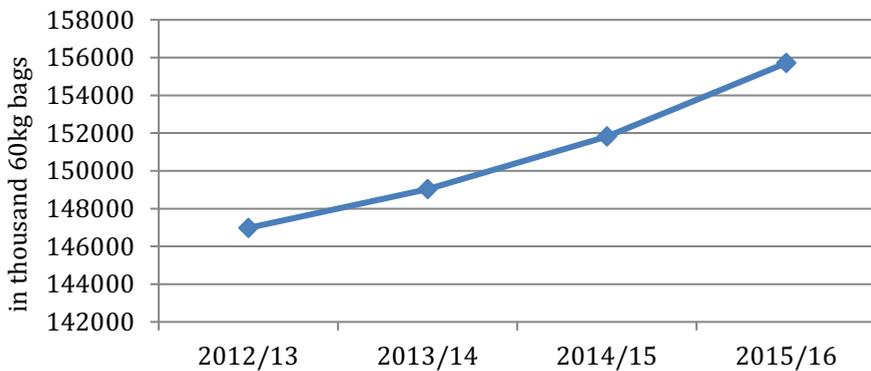


Figure 1-1 World coffee consumption (data from ICO)

In the market, people can find different types of coffee maker different ways to brew coffee. The most normal types are drip coffee maker, single serve coffee maker and espresso machine. Also in this market, products have different prices with diverse functions. For example: T13001 from Tower[2] (Figure 1-2) and E6 from Jura [3](Figure 1-3).



Figure 1-2 Tower T13001



Figure 1-3 E6 from Jura

Table 1-1 Different coffee maker

Coffee maker	Jura E6	Tower T13001
Price	7927 SEK	285 SEK
Coffee	Bean	Ground coffee
Functions and features	Intelligent water system, thermo-block heating system, preheating, smart connect, the choice of different types of coffee, making milk foam and so on	Heating, anti-drip feature, Permanent washable nylon filter and so on

From table 1-1, we can see that E6 is more intelligent and automatic, but price is more than 27 times comparing with T13001. And price still take up a big percentage of consideration for buying coffee maker. Although there are more useful functions which will make drinking coffee more convenient and enjoyable, many people will buy the cheaper at last.

For low price coffee maker, it is inconvenient that user has to check whether there is water in the coffee tank by themselves and turn on or off directly by standing near the coffee maker. For the safe problem, if water level or temperature over the limitation, people cannot close coffee maker immediately and maker will still work, when it is laid down by kids or pets at home or coffee pot is not in the coffee maker. And if coffee maker works too long time, it will waste lots of power.

What I will do in this paper is: monitoring the temperature and water level of coffee pot and coffee tank and then send value to cell phone; turning on or off the coffee maker is judged by system or human.

2 Chapter: Survey of related work

In this paper, the coffee maker which I use is drip coffee maker (Figure 2-1). It only has one switch for turn on or off. Ground coffee is put in the filter holder with a coffee filter. And it is recommended that approximately 7-8 g coffee for each cup. Of course, it can be changed by different preference of different people. Water is poured into tank of coffee maker. The capacity of tank is 1.5 litres. If we drink 125 ml for one cup, then we will have 12 cups for one brewing with the full tank. Some marks for water level in the tank are showed on the surface of tank. It gives the max water level line with number 12.



Figure 2-1 Coffee maker



Figure 2-2 Heating system

After turning on switch, water will be heated by internal electrical resistance and then hot water is push up and into filter holder. Coffee will flow into coffee pot along detachable lid with aroma pipe. The design for filter holder is special for anti-drip which means after moving out coffee pot, coffee in the filter will not flow or drip down. By the way, the heating system is close to the hot plate (Figure 2-2). One reason of this design is heating water and another one is keeping coffee warm which is in the coffee pot.

Table 2-1 Technical data about coffee maker

Name	Unit	Value
Voltage	V	230
Power	W	1000
Capacity	l	1.5

Arduino Uno[4] is the control system (Figure 2-3). Arduino is a kind of convenient, flexible, easy electronics platform. It includes many kinds of Arduino board and Arduino IDE. People program in the Arduino IDE with codes which are similar with Java and C. After verifying codes, program can be uploaded to the Arduino board. One of an important thing is that the pin of Arduino can only get or offer 0V to 5V.



Figure 2-3 Arduino UNO

Table 2-2 Technical data about Arduino Uno

Name	Unit	Value
Operating voltage	V	5
Input voltage	V	7-12
Digital I/O pins	/	14

Analog input pins	/	6
Flash memory	KB	32
SRAM	KB	2
Clock speed	MHz	16

About the measurement of water or coffee level, there are many ways can be used with the physical, electronics and mechanical theory[5] [6].

- Floats: Floats float on the surface of liquid and then use mechanical or magnetic equipment to get the position of floats
- Liquid pressure: Using pressure sensor get the bottom pressure of water tank and then according to formula the height of liquid can be calculated.

Formula for liquid pressure:

$$p = \rho gh \text{ (if it is water, } \rho = 10^3 \text{ kg/m}^3 \text{ } g = 9.8 \text{ N/kg)}$$

- Capacitance transmitter: Different liquid has different dielectric and the change of liquid level will course the change of capacitance.
- Load cell: Fixing sensors into water tank's support structure and using the increase weight of stuff and the liquid density calculate the height of liquid.
- Ultrasonic sensor: Using the time which ultrasound spend on going and reflecting back from the liquid surface calculated the distance between sensor and the liquid level.
- Laser sensor: Working theory is very similar with ultrasonic sensor, but in laser sensor, ultrasound is replaced by light.
- Radar sensor: Similar theory with microwaves and how good reflection can be get depends on dielectric constant of liquid.

Before we choose the measure method, we should know what kind of liquid, the environment of liquid and what result we want to get. Then we can get the most suitable one.

There are some works about using cell phone to control the switch of coffee maker by Arduino and its model Ethernet shield. This system also need router to send webpage codes to cell phone.

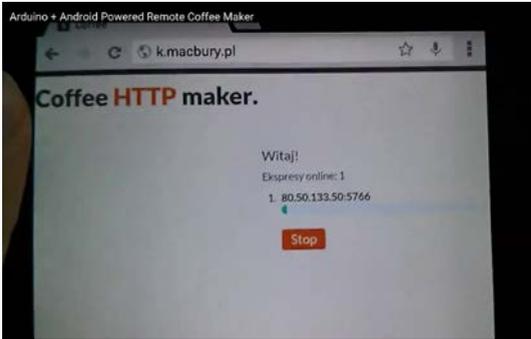


Figure 2-4 Webpage control Arduino



Figure 2-5 Ethernet shield

3 Chapter: Problem statement, objectives and main contribution

Some coffee makers are cheap but with less functions, it is inconvenient for people to drink coffee which they have to stand near the coffee maker to check whether there is water in the maker tank or liquid level in the coffee pot. And sometimes it will cause some safe problems and waste energy if no one keeps an eye on the status of coffee maker.

The objectives of this thesis aimed to add functions to coffee maker which let people drink coffee more easily and save their time.

Main contribution:

1. Get the values of coffee temperature, the liquid level of both coffee pot and maker tank and power cost by the coffee maker from sensors.
2. Monitor working status then send all information include values to Arduino.
3. Arduino connect with cell phone and users do remote control.

4 Chapter: Solution

I totally use five sensors and one router to finish all these functions.

4.1 Circuit

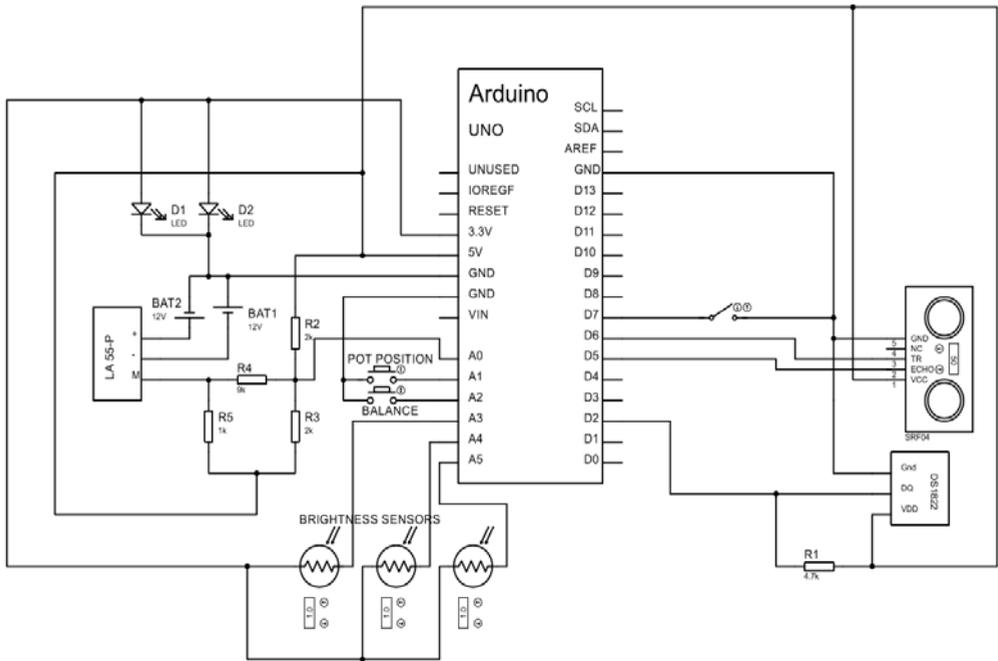


Figure 4-1 Circuit

4.2 Working status

At the beginning of all the work, the switch should be turned on. For the coffee maker which I use, it has single switch. Once the switch is set to the open position, maker will work immediately after giving power, otherwise it will never work.

User and system can turn on or off the switch from their cell phone, which means the switch should be controlled by Arduino. For this part, I use a remote controller switch (Figure 4-2) and put it between power cable and socket. But there are small changes which connects two lines to Arduino on that. Whether it allow current go through or not is controlled by Arduino not remote control. Arduino gives a high signal then it will work and vice versa.



Figure 4-2 Remote controller

Table 4-1 Technical data for remote control switch

Name	Unit	Value
Voltage	V	230
Power max	W	1000
Current max	A	5

There are two situations which coffee maker will not work or system stops maker to keep safe and avoid coffee overflow.

The first one is that if coffee pot is not in the coffee maker when user turns on it using cell phone and he or she doesn't know about this, hot water will flow into coffee filter continually and this will not stop until there is no water in the coffee maker tank. The whole content of coffee filter is less than 1.5 l. It will be dangerous and unpleasant that hot water flow at the surface of coffee maker and on the desk.

I use a simple mechanical switch (Figure 4-4) to solve this problem. One contactor of switch is connected to the ground of Arduino and the other one is connected to pin. If the switch is pushed which means coffee pot is in the coffee maker, the pin will get low signal. On user's cell phone, it will show the status of the position of pot. "Coffee pot is not in the maker" will be showed on the website if the switch is not pushed. Otherwise it will show the coffee level in the coffee pot. The position of switch is at surface of coffee maker at one side and it is vertical (Figure 4-3).



Figure 4-3 Judging whether coffee pot in the maker

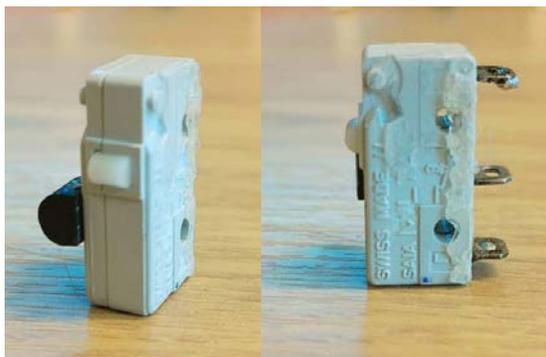


Figure 4-4 Mechanical switch

The second situation is that if the coffee maker is laid down by others or pets at home accidentally and user can not open the coffee maker to work from other place as well or system will stop maker by itself. And Arduino will remind user.

For this function, I also use a mechanical switch and connection is the same as the first one. I fix it at the bottom of coffee maker (Figure 4-5). When the maker is at normal position, desk will push the switch and pin will get 0

V. “Maker lays down” with red colour will be showed on the website or it will not show any words.

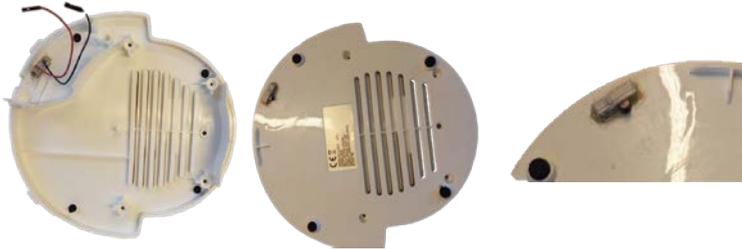


Figure 4-5 Mechanical switch for balance

4.3 Timer

Timer is used to monitor the working time of coffee maker and another important effect is to stop coffee maker automatically. Some people may forget their coffee while they are busy with other works, during these time, maker is still working. It will waste energy and coffee will not taste good if it is heated at a long time.

I program a timer and it works on the Arduino. When it arrives stopping time, system will turn off the switch automatically. On the cell phone, user can see “Out of time, maker turn off automatically”. In my coffee maker, I set ten minutes.

This time can also be showed one the cell phone, but it is not fluent for the time counting. I set 6-second-time interval for function asking Arduino to update time. But it is still real time, just need time to wait to refresh.

4.4 Monitor temperature

The best temperature for coffee drinking is around 85 °C. So monitor the temperature of coffee can let people know whether it is a good time to drink. At first, I want to get the temperature directly, but coffee will be drunk by people, so it will have some high requirements about sensor which tough coffee in the coffee pot. Meanwhile after it become coffee, it is very hot and sensor cannot be influenced by high temperature easily. Considering about

these two reasons, I choose DS1822[7] [8] as my sensor. It will touch the surface of coffee pot which made by glass (Figure 4-8). Although it will lose some degree of temperature between inside and outside glass, it doesn't matter and very accurate value is not necessary.

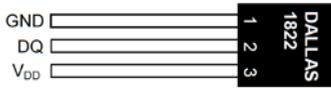


Figure 4-6 DS1822

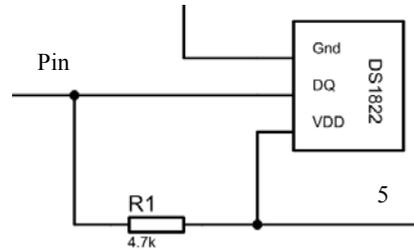


Figure 4-7 Circuit for DS1822

Table 4-2 Technical data for DS1822

Name	Unit	Value
Price	SEK	31
Power-supply min	V	3.0
Power-supply max	V	5.5
Temperature min	°C	-55
Temperature max	°C	125
Accuracy	°C	2

From technical data (Table 4-2), the range of temperature for DS1822 is -55 to 125 °C including working temperature of coffee maker. Accuracy of this sensor is 2 °C between -10 to 85°C . And price is proper.

On the cell phone, user will receive the temperature at real time. And the temperature is for coffee, so only pot is in the coffee maker that it will be showed.

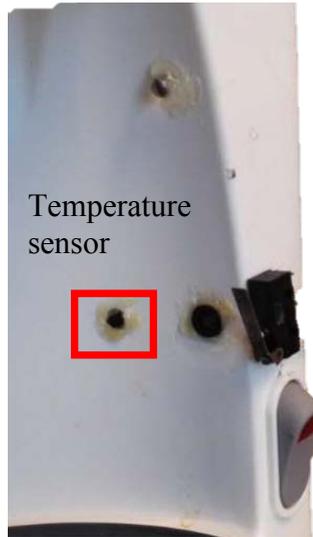


Figure 4-8 Sensor position

4.5 Liquid level measurement

In this part, I use some sensors to measure liquid level. All the values will send to Arduino and cell phone which will let user know if it is empty in the coffee maker tank or how much coffee in the coffee pot without going to the coffee maker and checking.

As said before, coffee will be drunk by people and all the measurement methods should be harmless to human body. So non-contact measurement will be at the top of list, such as light sensor and ultrasonic sensor. And high temperature in the coffee pot will make measurement become more demanding.

4.5.1 Coffee level in the coffee pot

This measurement will tell user how much coffee left in the coffee pot and let user decide whether continue to letting maker work or drink coffee first.



Figure 4-9 Brightness sensor

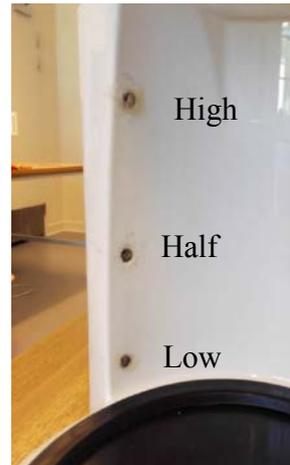


Figure 4-10 Positions of brightness sensors

There are three coffee levels: low, half and high (Figure 4-10). At the website, it will be showed “No water in the pot”, “There are little water in the pot”, “At least half coffee in the pot” and “The coffee pot is full”. When the coffee level at high position, the maker will stop working.

I choose brightness sensor[9] [10] to measure coffee level and fix them at one side where coffee pot will be put in. And at the other side I fix two LED in case of less light in the room. When the coffee level is higher than the position of brightness sensor, the sensor can't receive most of light because of the color of coffee. Arduino will receive two different voltages depending on whether get light or not. I set a threshold voltage value to judge whether coffee level over this position. Pins for getting brightness sensors' value are A5, A4 and A3.

One thing should be noted that brightness sensor will be influenced by temperature easily. I divide into 6 levels of threshold value with different temperature.

4.5.2 Water level in the coffee maker tank

If there is no water in the coffee tank, no coffee will be made out. System will check whether there is water in the tank, if not, it will remind user to add water through his or her cell phone. “Please add water” will be showed if there is no water and if it is, user will see “Has water”

The sensor I choose to measure water level is ultrasonic sensor HC-SR04[11] [12].

Table 4-3 Technical data about Ultrasonic sensor

Name	Unit	Value
Supply voltage	V	5
Ultrasonic frequency	kHz	40
Maximal range	cm	400
Minimal range	cm	3
Resolution	cm	1

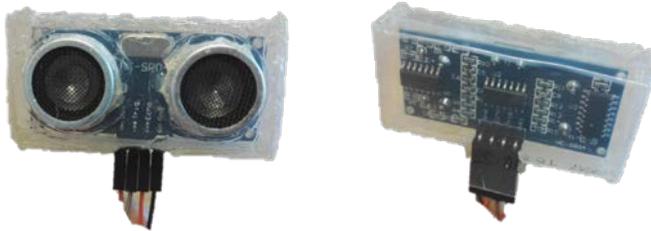


Figure 4-11 Ultrasonic sensor with box

There are two probes on the module (see Figure 4-11). One ultrasonic probe sent high frequency wave (ultrasonic wave), and the other one receive the wave which is reflected by edge of different medium. Sensor get the time from wave being sent out to being received. Then distance can be calculated through the speed of voice spreading in one medium.



Figure 4-12 Position of Ultrasonic sensor

I fix ultrasonic sensor at the top of the coffee maker tank (Figure 4-12). And I also note that when the maker work, especially at the end of working which most of water becomes coffee, there will have steam in the tank. To avoid water influencing sensor's work, I make a small box to put sensor in. The box will cover whole the sensor except two probes.

I set that after the water level is 12 cm away from the sensor, then system will note user to add water.

4.6 Power consumption

Monitoring the power consumption is for saving energy. The formula to calculate the power is :

$$P = UI$$

Voltage as we know is 220V, and I use sensor to get the current. The sensor is LA55P[13] [14], which is a kind of current transducer.

Table 4-4 Technical data about LA55P

Name	Unit	Value
Supply voltage	V	±12..15
Conversion ratio	/	1:1000

The current in the wire is converted into small current from the data pin of LA55P. But this small current cannot be used by Arduino, so I design a circuit

to change the current to voltage and let this voltage is suitable for Arduino (Figure 4-13).

The first step is change current to voltage and I use a resistor with 1000 ohm. After this, the voltage is from -5V to 5V because the current is a AC signal. Arduino pin can only get 0V to 5V, over this range Arduino will be damaged. These formulas show how to convert.

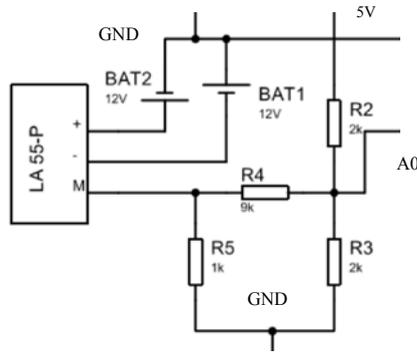


Figure 4-13 Circuit for power measurement

$$\frac{U_I - U_{A0}}{R_4} + \frac{4.74 - U_A}{R_2} = \frac{U_A}{R_3}$$

$$\frac{U_I - U_{A0}}{9.176k} + \frac{4.74 - U_{A0}}{2.001k} = \frac{U_{A0}}{1.996k}$$

$$3.9940U_I - 3.9940U_{A0} + 18.3153 * 4.74 - 18.3153U_{A0} = 18.3612U_{A0}$$

$$3.9930U_I + 86.985162 = 40.6705U_{A0}$$

$$U_{A0} = 0.0981579U_I + 2.13878$$

The resistors I use is not the same as theory and when Arduino works, the voltage from its supply voltage pin is not 5V, it is 4.74V. U_{A0} is the voltage at Arduino pin.

Then using formula:

$$U_{I_{max}} = (U_{A0_{max}} - 2.1388)/0.0982$$

$$U_{I_{min}} = (U_{A0_{min}} - 2.1388)/0.0982$$

$$I = \frac{U_{I_{max}} - U_{I_{min}}}{2 \times \sqrt{2}}$$

To ensure getting the maximum and minimum value, system will take 20 values at a time.

For the supply voltage of sensor, it requires +12V and -12V DC. When I do tests, I use output DC power supply. After making sure that the circuits and codes can work, I use two 12V alkaline high voltage batteries[15] instead (Figure 4-13).

4.7 Communication with cell phone

One of the interesting features is user can use cell phone to get all the values and can remote control switch. For wireless communication, there are two choices: Bluetooth or WIFI. Both devices are equipped on the most of mobile terminals today which means not only cell phone can connect with this coffee maker, but also computers, laptops, tablet computers and so on. Here I choose WIFI, because WIFI can cover more areas. Arduino Ethernet shield (Figure 2.5) and router are used to communicate with cell phone.

After confirming connection with the client, Arduino will send markup language to cell phone. HTML5 is a kind of markup language I use here. It is based on the HTML standard and can show the content of webpage. Most of HTML5 is finished by IntelliJ IDEA and then move into Arduino IDE.

The working processing is that sending HTML5 to cell phone after cell phone connect coffee maker's WIFI. The IP address is 192.168.1.177. There are some functions with XML in the head and then these functions will request Arduino send values.

For the interface, I use CSS to finish design. CSS is used to describe the style of HTML[16].

Coffee Maker

Temperature: 26.44 degree || Power: 866.13

working time: 3 : 24

There is little coffee in the pot

Has water

Turn off

Figure 4-14 Working

Coffee Maker

Temperature: 0.00 degree || Power: 0.00

working time: NOT WORKING

Coffee pot is not in the maker

Please add water

Turn on

Figure 4-15 Coffee pot is not in the maker

Coffee Maker

Temperature: 23.37 degree || Power: 0.00

working time: NOT WORKING

No coffee in the pot

Please add water

Turn on

Figure 4-16 Coffee pot is the maker but no coffee inside

Coffee Maker

Temperature: 23.44 degree || Power: 0.00

working time: NOT WORKING

No coffee in the pot

Please add water

Maker lays down

Turn on

Figure 4-17 Maker lays down and maker don't work

When moving HTML5 language into Arduino IDE, we should pay attention that the dynamic memory of Arduino UNO is limited, only 2 KB. And the HTML5 language will take up a lot of memory. If dynamic memory is too less, it will cause some problems, such as program is unstable with messy codes or the system will not work.

Generally, sending language to client use this language:

```
client.println("<!DOCTYPE HTML>");
```

But this will cause the problem I said before, if there too much languages are needed to be sent, so I add "F" before strings.

```
client.println(F("<!DOCTYPE HTML>"));
```

This "F" will tell compiler to keep strings in the program, which means they will in the Flash, not copy to SRAM (SRAM is dynamic memory).

The second problem which will also cause the messy code and let Arduino not to work is that the frequency of webpage requesting value is too fast. In my program, I set 6-second-interval to send requirement. And in one function, I can't put too much value sending, because it will also cause messy codes.

The third problem is that, when we use Ethernet shield, pin 7 to pin 13 will be taken up for WIFI communication and can't use them to input or output.

4.8 Placement for sensors and circuits

I weld all the resistors on one small board and put board, LED and sensors in the coffee maker which it has empty place inside except one mechanical switch (Figure 4-18 and 4-19).

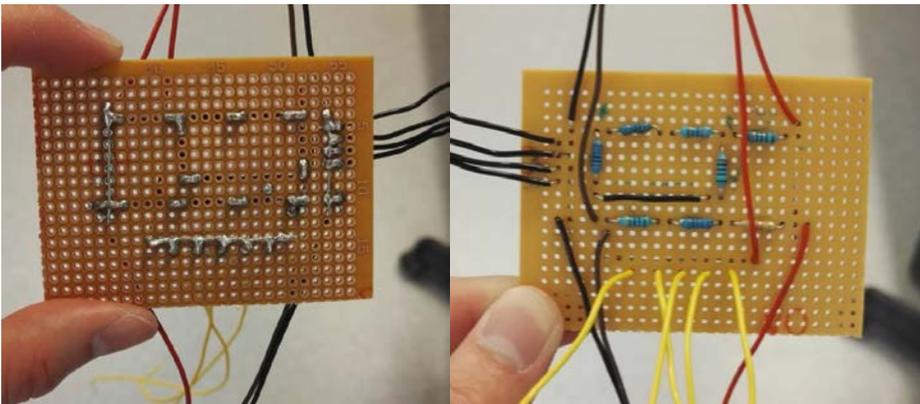


Figure 4-18 Circuits board

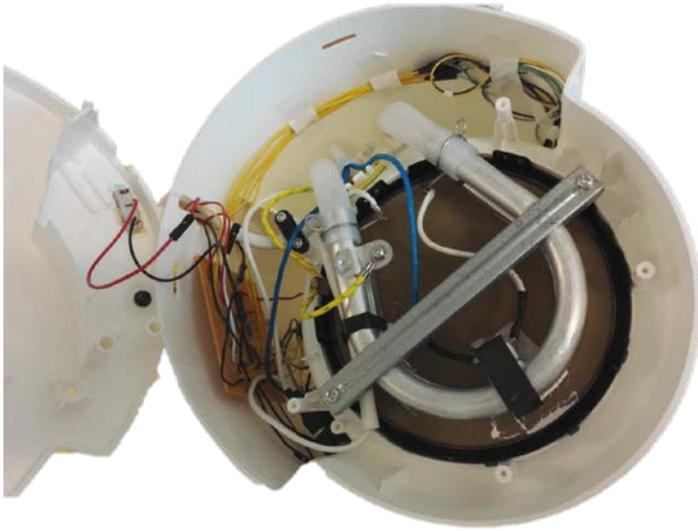


Figure 4-19 Placement for sensors and circuit board

To fix the sensor, I use hot melt glue which is liquid status at high temperature and solid status at normal temperature.

4.9 Total price for coffee maker

Name	Price (SEK)	Number
Coffee maker	200.00	1
HC-SR04	69.50	1
LA 55-P	118.00	1
Brightness sensor	34.36	3
LED	4.20	2
DS1822	45.73	1
Battery	19.57	2

Total price: 583.85 SEK

Note: the more user buy, the lower price will be. The price in the table is for only buying one.

5 Chapter: Conclusion and future work

After giving power to Arduino and router, cell phone can search WIFI from coffee maker. I input IP address into browser, it will upload and show the content of webpage after waiting for few seconds. Coffee pot is put into the maker, otherwise it cannot work. Clicking turn button on the webpage then whole the system will work.

When making coffee, webpage will show temperature and power. Water and coffee level are showed all the time after connect WIFI. If pot is taken away or maker lays down, the system will stop at once. Time also be showed on the webpage, and if coffee maker doesn't work, it shows "NOT WORKING".

For the future work, there are some places can be improved or add more powerful functions.

First for the coffee temperature measurement, I put only one sensor at one position. This may not get the temperature so exactly because of the temperature loss by the glass or coffee level is lower than the position. Maybe next time the number of sensor can increase.

Second, I use brightness sensor to judge coffee level. It will be a little unstable and influenced by environmental brightness easily even I fix two LED. Capacitive sensor will be another choice, if still require non-contact.

Third, improving the codes. These codes which Arduino runs now are still will cause messy codes on the webpage and then Arduino cannot work sometimes. I think it should be communication problems between Arduino and WIFI.

About functions, concentration measurement of coffee can be a good improvement.

6 Chapter:

Template description

Reference

- [1] International Coffee Organization, “Coffee consumption table.pdf.” [Online]. Available: <http://www.ico.org/prices/new-consumption-table.pdf>, Apr-2017.
- [2] Tower, “Tower T13001.” [Online]. Available: <https://www.towerhousewares.co.uk/tea-coffee-makers-c8/10-cup-coffee-maker-p45>, Apr-2017.
- [3] Jura, “E6 - JURA United Kingdom.” [Online]. Available: <https://uk.jura.com/en/homeproducts/automatic-coffee-machines/E6-EU-15079#tabs>, Apr-2017.
- [4] Arduino, “Arduino - ArduinoBoardUno.” [Online]. Available: <https://www.arduino.cc/en/main/arduinoBoardUno>, Apr-2017.
- [5] Kevin Hambrice and Henry Hopper, “A Dozen Ways to Measure Fluid Level and How They Work | Sensors,” Dec. 2004, Apr.2017.
- [6] Tarun Agarwal, “Liquid Level Sensor and Types of Level Sensors,” *Buy Electronics & Electrical Projects in Europe*, 04-Jul-2014, Apr-2017.
- [7] Farnell, “DS1822+.” [Online]. Available: <http://se.farnell.com/maxim-integrated-products/ds1822/temperaturgivare-2-gr-c-to-92/dp/2515551>, Apr-2017.
- [8] Maxim integrated, “DS1822.pdf.” [Online]. Available: http://www.farnell.com/datasheets/2001086.pdf?_ga=2.174292962.1577571560.1495719214-266873468.1495719165, Apr-2017.
- [9] Elfa Distrelec, “Brightness sensor,” *Brightness sensor, Panasonic Automotive & Industrial Systems, AMS 302 | Elfa Distrelec Sweden*. [Online]. Available: <https://www.elfa.se/en/brightness-sensor-panasonic-automotive-industrial-systems-ams-302/p/17650001?q=Brightness+sensor%2C+AMS+302%2C+Panasonic+Automotive+%26+Industrial+Systems&page=1&origPos=1&origPageSize=25&simi=89.51>, May-2017.
- [10] “Brightness sensor data.pdf.” [Online]. Available: https://www.elfa.se/Web/Downloads/ta/_e/xoAMS3_Data_E.pdf?mime=application%2Fpdf, May-2017.

- [11] Elfa Distrelec, “Ultrasonic distance sensor.” [Online]. Available: <https://www.elfa.se/en/ultrasonic-distance-sensor-raspberry-pi-pi-2b-raspberry-pi-debo-sen-ultra/p/30036820?queryFromSuggest=true>, Apr-2017.
- [12] “Ultrasonic sensor data.pdf.” [Online]. Available: https://www.elfa.se/Web/Downloads/_t/ds/SEN-US01_eng_tds.pdf?mime=application%2Fpdf, Apr-2017.
- [13] Elfa Distrelec, “Current transformer.” [Online]. Available: <https://www.elfa.se/en/current-transformer-70-lem-la-55/p/17696241?q=la55p&page=1&origPos=1&origPageSize=25&simi=86.26>, Apr-2017.
- [14] LEM, “LA55P data.pdf.” [Online]. Available: https://www.elfa.se/Web/Downloads/_t/ds/LA55P_eng_tds.pdf?mime=application%2Fpdf, Apr-2017.
- [15] gpbatteries, “GP battey.pdf.” [Online]. Available: http://www.farnell.com/datasheets/1865177.pdf?_ga=2.118753217.912138683.1495724379-375474587.1480710525, May-2017.
- [16] Wikipedia, “HTML5.” [Online]. Available: <https://en.wikipedia.org/wiki/HTML5>, May-2017.

Appendix 1:

Codes of Arduino

In other files

Arduino codes

```
#include <SPI.h>
```

```
#include <Ethernet.h>
```

```
#include <OneWire.h>
```

```
#include <DallasTemperature.h>
```

```
#include <SR04.h>
```

```
SR04 sr04 = SR04(5,6);
```

```
#define ONE_WIRE_BUS 2
```

```
OneWire oneWire(ONE_WIRE_BUS);
```

```
DallasTemperature sensors(&oneWire);
```

```
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
```

```
IPAddress ip(192,168,1,177);
```

```
EthernetServer server(80);
```

```
EthernetClient client;
```

```
String readString="";
```

```
unsigned long nowtime=0;
```

```
unsigned long starttime=0;
```

```
unsigned long time=0;
```

```
int minute = 0;
```

```
int second = 0;
```

```
float balancepin;
```

```
float levelhigh;
```

```
float levelhalf;
```

```
float levellow;
```

```
float ItoVPin ;
```

```
float cm;
```

```
float ltoV = 0;

float t = 0;

float p = 0;

String b;

String level;

String balance;

String turnoff;

void setup() {

  Serial.begin(9600);

  Ethernet.begin(mac, ip);

  server.begin();

  sensors.begin();

  pinMode(6, OUTPUT); // TrigPin

  pinMode(5, INPUT); // EchoPin

  pinMode(7, OUTPUT); //pin 7 is switch

  pinMode(A0, INPUT); //POWER A1 has problem

  pinMode(A1, INPUT); //mechanical switch for the position of pot

  pinMode(A2, INPUT); //balance

  pinMode(A5, INPUT); //high position

  pinMode(A4, INPUT); //half position

  pinMode(A3, INPUT); //lowposition

}

void loop() {

  float a[20]={0};
```

```

float maxv = 2.150;

float minv = 2.150;

float Vin1 = 0;

float Vin2 = 0;

//Temperature

sensors.requestTemperatures();

t = sensors.getTempCByIndex(0);

//Power

for(int i=0;i<20;i++){

    ltoV = analogRead(ltoVPin)*4.740/1023.000;

    a[i]=ltoV;

    delay(1);

}

for(int i=0;i<20;i++){

    if (a[i]>maxv){

        maxv = a[i];

    }

    if (a[i]<minv){

        minv = a[i];

    }

}

Vin1 = (maxv-2.15)/0.0982;

Vin2 = (minv-2.15)/0.0982;

p = 220*(-Vin2+Vin1)/(2*sqrt(2));

```

```
//Water level

cm=sr04.Distance();

if (cm >= 12.00){

    b="Please add water";

}

if (2.00 < cm & cm < 12.00){

    b="Has water";

}

//Coffee position

levelhigh = analogRead(A5);

levelhalf = analogRead(A4);

levellow = analogRead(A3);

if (analogRead(A1)==0){

    if (levellow > 654 && t <=27){

        if (levelhalf > 655){

            if (levelhigh >= 650){

                level = "The coffee pot is full";

            }

            else{

                level = "At least half coffee in the pot";

            }

        }

    }

    else{

        level = "There is little coffee in the pot";

    }

}

}
```

```
else{
    level = "No coffee in the pot";
}

if ((levellow >= 656) && t > 28.00 && t <= 38.00){
    if (levelhalf > 658){
        if (levelhigh > 650){
            level = "The coffee pot is full";
        }
        else{
            level = "At least half coffee in the pot";
        }
    }
    else{
        level = "There is little coffee in the pot";
    }
}

if (levellow > 659 && t > 38 && t <= 45){
    if (levelhalf > 660){
        if (levelhigh > 655){
            level = "The coffee pot is full";
        }
        else{
            level = "At least half coffee in the pot";
        }
    }
}
```

```
else{  
    level = "There is little coffee in the pot";  
}  
}
```

```
if (levellow >= 662 && t > 45 && t <= 55){  
    if (levelhalf > 662){  
        if (levelhigh > 659){  
            level = "The coffee pot is full";  
        }  
        else{  
            level = "At least half coffee in the pot";  
        }  
    }  
    else{  
        level = "There is little coffee in the pot";  
    }  
}
```

```
if (levellow >= 665 && t > 55 && t <= 65){  
    if (levelhalf > 665){  
        if (levelhigh > 662){  
            level = "The coffee pot is full";  
        }  
        else{  
            level = "At least half coffee in the pot";  
        }  
    }  
}
```

```
}  
else{  
    level = "There is little coffee in the pot";  
}  
}
```

```
if (levellow >= 670 && t > 65){  
    if (levelhalf > 670){  
        if (levelhigh > 660){  
            level = "The coffee pot is full";  
        }  
        else{  
            level = "At least half coffee in the pot";  
        }  
    }  
    else{  
        level = "There is little coffee in the pot";  
    }  
}
```

```
}  
else  
{  
    level="Coffee pot is not in the maker";  
    digitalWrite(7, LOW);  
    time = 0;  
    t = 0.00;
```

```
}
```

```
// Maker position(balance)
```

```
balancepin = analogRead(A2);
```

```
if (time >= 600){
```

```
    turnoff = "<div>out of time, turn off automatically</div>";
```

```
}
```

```
else{
```

```
    turnoff = "";
```

```
}
```

```
if(balancepin!=0){
```

```
    balance= "<h2 style=\"color:red\">Maker lays down</h2>";
```

```
    time = 0;
```

```
}
```

```
else{
```

```
    balance = "";
```

```
}
```

```
if (time >= 600 || balancepin!=0){
```

```
    digitalWrite(7, LOW);
```

```
}
```

```
if (digitalRead(7)==HIGH){
```

```
    nowtime = millis();
```

```
    time = nowtime - starttime;
```

```
    time = time/1000;
```

```
    minute = time/60;
```

```
second = time%60;
}
else{
    p = 00.00;
}
delay(5);

// WIFI
client = server.available();
if (client) {

    boolean currentLineIsBlank = false;
    while (client.connected()) {
        if (client.available()) {
            char c = client.read();
            readString += c;

            if (c == '\n') {

                if(readString.indexOf("?on") >0) {
                    digitalWrite(7, HIGH);
                    starttime = millis();
                    break;
                }

                if(readString.indexOf("?off") >0) {
                    digitalWrite(7, LOW);
```

```
time = 0;
break;
}
```

```
if(readString.indexOf("?getTemperature") >0) {
    client.println(F("<i>Tempreture: </i><b> "));
    client.println(t);
    client.println(F(" degree</b><i>&nbsp;| |&nbsp;Power: </i><b>"));
    client.println(p);
    client.println(F(" </b>"));
    break;
}
```

```
if(readString.indexOf("?getCoffeelevel") >0) {
    client.println(F("<h3>"));
    client.println(level);
    client.println(F("</h3><h3>"));
    client.println(b);
    client.println(F("</h3>"));
    client.println(balance);
    break;
}
```

```
if(readString.indexOf("?getTime") >0) {
```

```
client.println(F("working time: "));  
  
if (time ==0){  
    client.println(F("NOT WORKING"));  
}  
  
else{  
    client.println(minute);  
    client.println(F(" : "));  
    client.println(second);  
}  
  
client.println(turnoff);  
  
break;  
}  
  
delay(2);  
SendHTML();  
  
break;  
}  
  
}  
  
delay(10);  
client.stop();  
readString="";  
}  
}
```

```

void SendHTML()
{
    client.println(F("HTTP/1.1 200 OK"));

    client.println(F("Content-Type: text/html"));

    client.println(F("Connection: close"));

    client.println();

    client.println(F("<!DOCTYPE HTML>"));

    client.println(F("<html><head><meta charset=\"UTF-8\"><style>div.container {width:
100%;}header{padding: 1em;color: white;background-color: black;clear: left;text-align:
center;}#wrapper { text-align: center;}"));

    client.println(F(".column {width: 350px; margin: 10px;display: inline-block;text-align: left;}container
{margin:auto;text-align: center;}"));

    client.println(F(".button {position: relative;background-color: #000;border: none;font-size:
20px;color: #FFFFFF;padding: 20px;width: 120px;text-align: center;transition-duration: 0.4s;text-
decoration: none;cursor: pointer;margin: auto; }</style>"));

    client.println(F("<script type=\"text/javascript\">function send2arduino(){var xmlhttp;xmlhttp=new
XMLHttpRequest();element=document.getElementById(\"Maker\");if
(element.innerHTML.match(\"Turn on\")){element.innerHTML=\"Turn off\";
xmlhttp.open(\"GET\", \"?on\", true);}else{ element.innerHTML=\"Turn
on\";xmlhttp.open(\"GET\", \"?off\", true); }xmlhttp.send();}"));

    client.println(F("function getTime(){var xmlhttp;xmlhttp=new XMLHttpRequest();});

    client.println(F("xmlhttp.onreadystatechange=function(){if (xmlhttp.readyState==4 &&
xmlhttp.status==200)"));

    client.println(F("document.getElementById(\"Time\").innerHTML=xmlhttp.responseText;});xmlhttp.op
en(\"GET\", \"?getTime\", true); xmlhttp.send();}window.setInterval(getTime,6000);"));

    client.println(F("function getCoffeelevel(){var xmlhttp;xmlhttp=new XMLHttpRequest();});

    client.println(F("xmlhttp.onreadystatechange=function(){if (xmlhttp.readyState==4 &&
xmlhttp.status==200)"));

    client.println(F("document.getElementById(\"Coffeelevel\").innerHTML=xmlhttp.responseText;});xml

```

```

http.open(\"GET\", \"?getCoffeelevel\", true);
xmlhttp.send();window.setInterval(getCoffeelevel,6000);));

client.println(F(\"function getTemperature(){var xmlhttp;xmlhttp=new XMLHttpRequest();});

client.println(F(\"xmlhttp.onreadystatechange=function(){if (xmlhttp.readyState==4 &&
xmlhttp.status==200)\"));

client.println(F(\"document.getElementById(\"Temperature\").innerHTML=xmlhttp.responseText;};x
mlhttp.open(\"GET\", \"?getTemperature\", true);
xmlhttp.send();window.setInterval(getTemperature,6000);</script>\"));

client.println(F(\"</head><body><div align=\\\"center\\\"><header><h1>Coffee
Maker</h1></header><br/><br/><div class=\\\"container\\\"><div class=\\\"column\\\">\"));

client.println(F(\"<div id=\\\"Temperature\\\"></div>\"));

client.println(F(\"<br/><h2>\"));

client.println(F(\"<div id=\\\"Time\\\"></div></h2>\"));

client.println(F(\"<div id=\\\"Coffeelevel\\\"></div><br/><br/><div class = \\\"wrapper\\\">\"));

client.println(F(\"<button align=\\\"right\\\" class=\\\"button\\\" id=\\\"Maker\\\" type=\\\"button\\\"
onclick=\\\"send2arduino()\\\">Turn on</button>\"));

client.println(F(\"</div></div></body></html>\"));

}

```