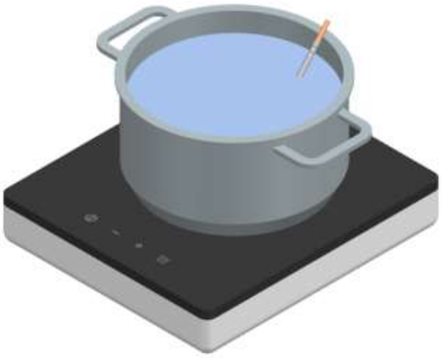
Kompor: smart cooktop

Final report Engineering Design (4WBB0)

[30/10/2023]



Group no: [288]	
Name	Student ID
Jamin van Amelsvoort	1829998
Mees Schilders	1724231
Sander Oomen	1811657
Luka Hendriks	1864831
Jikun Shen	1833847
Georgios Tsormpatzoglou	1812416
Alessandro Brugnera	1889958

This template contains the chapter numbers and titles with a short description of the content and suggested number of pages. It is not allowed to change the chapter titles, font type and size, page layout, and paragraph settings.

The maximum number of pages of the report is 28 (excluding title page, references, and appendices).

Table of contents

1. Group effectiveness (typically 2 pages)

Explain how the backgrounds, majors and interests of individual group members were translated to strengths and weaknesses on group level. Show which of these strengths were effectively used in the design or in the design process, and how weaknesses were resolved. In other words, describe how the group effectiveness developed during this project. This should go beyond stating that co-operation improved over time. *Note: there is no need to mention individuals nor their contributions.*

In this student group, there were many individuals with diverse backgrounds. Each student follows a different course of study and is enrolled in different courses. This diversity in academic backgrounds led to varying perspectives on different aspects of the design concept and, subsequently, the design itself. This diversity greatly strengthened the group as it enabled perceiving a wide range of potential problems and offered diverse solutions to address these issues. Additionally, with each student possessing unique specializations, it became significantly easier to distribute tasks effectively. Each student could focus on their respective areas of expertise, such as coding, constructing electrical circuits, or creating a website, as well as presenting and writing the content for the report. These specializations greatly contributed to the overall improvement of the group, since less time was needed to divide the tasks and finish the individual work.

However, these specializations also facilitated individual learning of the other students, allowing them to acquire new skills beyond their existing knowledge or skills learned from the chosen CBL module. This contributed to the enhanced knowledge of all the students, which improved the quality of the designs. With everyone capable of reviewing each other's work and brainstorming new and innovative ideas together, the collaborative capacity was greatly enhanced.

This last strength initially had a minor weakness at the start of the project, as there was one student with significantly higher technical knowledge of the design. Somehow, this student was the only one who possessed knowledge of the challenges that would be encountered at the beginning of the project, including estimating costs in advance and the complexities of constructing the electric circuit, which can be challenging in various designs. However, due to effective communication within the group and a low barrier to seeking clarification from fellow students, this weakness rapidly transformed into a strength.

Another weakness that was encountered in the first weeks of the project is the unorganized collaboration of the group. Many different ideas were created and were written down in separate SSA's. This caused a mess because, after a few weeks, a lot of information was gathered, but it was all over the place. All the different technical specifications were thus also worked out in different SSA's and the same happened for all the different prototypes that had been worked out. After getting feedback on the working style, this was immediately changed. All the information was gathered from the various reports and put all together to get a clear overview. This proved effective later in the process. Since all the information was gathered in a single file, the collaboration became much smoother and working on the report and further SSAs was significantly easier. In the weeks afterward this working style kept being adopted. This indicated that feedback was taken very seriously, acted upon, and continued to be used later. And thus, this weakness was turned into a strength, because this way of working made it possible to finish writing the report and modifying the design on time.

So overall there was great growth as a group, especially in working together as a team. This came back during meetings, which became much more efficient over time as well as working together on SSAs. Overall, the student group got much better at working together. The group meetings improved, becoming more organized and productive, thanks to different skills and backgrounds. The messy way data was handled in SSAs was also fixed by keeping everything in one place. It was learned to listen to feedback which turned initial weaknesses into strengths. The ability to adapt and get things done made it possible to finish the report and meet design goals on time, showing improvement as a group during the project.

2. Design goal (typically 2 pages)

Motivate the design goal. Why did you choose this problem in your focal area, why this type of user and activity, and why this specific product? Support your motivation by literature, background/experience, group strengths/weaknesses, etc. Why did you feel this would lead to a desirable and innovative product?

To watch television, use a computer, take a shower, get hot water, or use a radiator, energy is essential. Energy is needed for a large amount of daily life activities. Including cooking your food, energy is required in the form of heat. There are multiple ways of heating up food with different energy efficiencies and consequences for the environment. The most common ones are gas with an efficiency of 40%, electricity with 74% and induction with 84% (Leafscore, 2023). So, there is energy loss when converting gas or electricity into heat, but these are not the only losses. On average between 43% and 63% of the heat is lost due to the environment with an open pot and about 20% with a closed lid (Edison EB, 2022). The energy efficiency of the way the food is cooked also matters. Energy-efficient cooking is achieved by, for example, reducing one's cooking time.

For many cooking recipes, boiling water is an essential step, such as when preparing pasta, rice, potatoes, etc. Here water is often brought to a temperature of 100 °C and kept at that constant temperature to keep the food at the same temperature the whole time for the best taste (Revision World, 2023). Extra heat is added to make sure the water stays at a constant temperature of 100 °C, this is the boiling point of water, so the temperature will not go any higher. What the extra heat will do instead is evaporate water, which is unnecessary since the evaporated water does not contribute to the quality of the food. Of course, a pot lid can be put on the pot to significantly reduce the amount of water that will evaporate, but heat is still being added so water will evaporate eventually. This energy used to keep the water at a constant temperature is lost. This is a problem, because so many people in the world boil water to prepare their meals, this means that annually a lot of energy is wasted. This total waste of energy can be significantly reduced.

Giorgio Parisi, a 2021 Nobel Prize winner for physics has released a statement, mainly to the Italians, about this problem of non-energy-efficient cooking and suggests an alternative method of cooking pasta based on reducing one's cooking time (Tapas, 2023). By only boiling the water for 2 minutes and closing the pot, as opposed to boiling the water with the pasta in it for 10 minutes straight, one can save 6 euros annually on cooking pasta alone (Tapas, 2023). This means that 47 million euros can be saved by the Italians with only cooking pasta (Tapas, 2023). This shows that cooking, especially recipes involving boiling water, can be made more energy efficient. There are also other people, such as David Fairhurst, who want to build forth on Parisi's ideas of reducing cooking time. This means that in the field of energy-efficient cooking there is research being done, but there is not yet much innovation for products which incorporate these ideas. But there is a slight problem with Parisi's method, as according to various chefs like Michelin-starred chef Antonello Colonna and chef Luigi Pomata this way of cooking pasta will only lead to unhappiness, as the pasta will have a rubbery texture and in general will not taste that great (Tapas, 2023). They state that a pot needs to be open to cook the best pasta (Tapas, 2023).

The goal of this project is to create a product that combines these ideas of both energy efficient cooking and cooking great food. This will be achieved by building upon Giorgio Parisi's idea that reducing heating time results in significant energy savings. The goal, therefore, is to develop a device capable of controlling the temperature of the water so that it does not reach boiling point, by for example only reaching 95 °C. The main goal is to find a way to heat water for cooking without it evaporating and wasting energy. This will help people prepare pasta, rice, potatoes, and other foods requiring boiling water more efficiently. The concept could potentially result in a product that enhances the energy efficiency of cooking, aligning with the goal of improving cooking appliances to reduce energy consumption. As already mentioned, this would lead to less energy being used, in turn this means money is saved when cooking, which is beneficial for the average person. It also means a more sustainable environment, as less energy is wasted. The goal is to make something that appeals to the public. The belief is that this will help to create a future of energy-efficient cooking. With a blend of practical and technical knowledge, a product will be made and programmed to control the temperature of water to nearly, but not quite, boiling point, resulting in substantial energy savings when heating water for food and this product will save this energy while still being able to cook food in the most delicious way possible.

To reinforce this statement of substantial energy savings, some calculations have been made to roughly calculate the amount of energy that could be saved when cooking 200 grams of pasta for ten minutes. If, for example, the water temperature will be regulated at 100 °C, and the regulating of the temperature does not cost any extra energy, about 950 KJ of energy would be saved when using a power supply with 1000 W of power. This is about 0.26 kWh which is a decent saving of energy. Pasta is not the only type of food that could be cooked with the product, so including potatoes, rice, etc. people would approximately use the product 3-4 times a week, leading to ~20 euros saved a year, but specifics will be mentioned further in this report. This does show that there would be quite some energy saved.

3. Functional design and solutions (typically 2 pages)

Give a list of functions (i.e. the functional specification) for your design. Prioritize the functions according to the MoSCoW method. The *must have's* and *should have's* should be sufficient to make your product work(*). Give the "Solution-Encyclopedia" in which each function is accompanied by an extensive list of solution options.

(*) *Note that the requirements as prescribed by the course do not suffice as specifications of your product.*

Introduction

Firstly, possible functional specifications are listed, ordered using the MoSCoW method. All red specifications have not been integrated into the final product. Then there is a solution encyclopedia in which solutions are listed for each functional specification.

List of functional specifications:

Must:

- The device must be safe to use
- The device must be safe to consume water and foods from
- Usage of the device should not damage the device (electrical parts may not get too hot)

Should:

- The device should be able to contain water or allow for a pot/kettle/etc. to be used
- The device should be able to provide heat to the water
- The device should be more efficient than boiling water using a stove
- The device should be able to measure the temperature of the water
- The device should be able to regulate the input of energy
- The device should be able to transmit the water temperature to the part that regulates the energy input

Could:

- The device could have an option to follow specific temperature plans (the pasta method as discussed during the meeting)
- The device could look aesthetically pleasing
- The device could have a measure that detects water leaking
- The device could have a measure against water leaking (in addition to simply not filling it too much)
- The device could be better if it boils water faster without compromising on other functions
- The device could show how much energy it saves
- The device could be able to be controlled externally

Won't:

- The device will not need the ability to be integrated into a stove

Solution-Encyclopedia:

Must:

<u>The device must be safe to use:</u>

- All components used in the design must be CE certified.
- Limit the temperature it can reach
- Test the prototype thoroughly in the testing phase with multiple different situations the product might encounter.

<u>The device must be safe to consume water and food from:</u>

-Food-safe materials must be used.
- Prevent electricity in the water by proper insulation of the prototype.
- Put the Arduino in an isolated box where no water can enter to ensure no water comes in contact with it.
- Use copper wires for their good conduction and food safety.
- Avoiding plastic with Bisphenol-A (BPA) due to health effects in contact with food.

<u>Usage of the device must not damage the device (electrical parts may not get too hot):</u>

- Use thicker wires to counter overheating of wires.
- Have the system disable itself when temperatures are reached that are too high.
  • For the temperature sensor (In case limited to 125 °C)
  • For the Arduino system (coding the Arduino that it will shut down if the temperature sensor is too hot)

Should:

<u>The device should be able to contain water:</u>

- A compartment in/on the device itself
- A space, like where pans/pots can be put on, which then contains water

<u>The device should be able to heat the contained water:</u>

- Externally applying heat (a pot is heated causing the items inside to be heated):
  Through induction, electricity, or gas.
- Internally applying heat:
  Closed water circuit where external hot liquid is passed through. Electricity via a heating element that is in direct contact with the water. Stir water at a high speed.

<u>The device should be more efficient than boiling water with a stove:</u>

- Internally heating the water with a heating element.
- Not letting the water get any hotter than is necessary.
- Optimally reaching the water temperature.

<u>The device should be able to measure the temperature of water:</u>

- Temperature sensor in the water:
  NTC (negative temperature coefficient), RTD (resistance temperature detector), thermocouple, semiconductor based or infrared sensor.
- Sensor should be connected to an external controller so the measured temperature can be used

<u>The device should be able to regulate the input energy:</u>

- Varying strength of electrical current, through building a circuit or feeding different voltages.
- Adjusting the valves of a cooktop in case of gas.
- Pulse width modulation in the case of induction.

<u>The device should be able to transmit the temperature to the part that regulates the energy input:</u>

- The temperature sensor used should be connected to a computer, such as a laptop, personal computer, or any microcontroller, like an Arduino or an ESP32.

Could:

<u>The device could have an option to follow specific temperature plans (the pasta method as discussed during the meeting):</u>

- Implementation of Wi-Fi in the device, so that a specific temperature plan can be selected by

a user. Either on a website or mobile phone application.
- Let the user create their own temperature plans in the app.
- Having a temperature regulator/input on the device so that the user could set the temperature on the device itself.
- Having physical buttons on the device for temperature plans.

The device could look aesthetically pleasing:
- Hide smaller cables together in bigger cables.
- Put the Arduino in an aesthetically pleasing box.
- Design the path of cables beforehand to counter cables being disorganized.
- 3D print parts.

The device could have a measure that detects water leaking
- Pressure sensor measuring the small difference in pressure of the water on top of the sensor.
- Using an additional sensor at the water level that will sense if the water level dropped based on the refraction or absorption in light.

The device could have a measure against water leaking:
- Use a water sensor to adjust the temperature of the stove to decrease the amount of water being vaporized.
- In case of overboiling an alarm will sound, with a sensor that measures water level.

The device could boil at a fast speed while not compromising on any other functions.
- High power plan to reach desired temperature quickly.
- An option for users to choose between fast heating mode and eco mode to get water to temperature. Could be done by using physical switch or integrated into the app/website.

The device could show how much energy it saves.
- Include a small addition that calculates how much energy is saved based on the energy used by the device.

The device could be controlled externally.
- Implementation of Wi-Fi in the device, so that a specific temperature plan can be selected by a user. Either on a website or mobile phone application.

4. Concept designs (typically 2 pages)

Describe three different concept designs that meet the design goal. The concepts represent different combinations of solutions, but share the same functional specification. Illustrate these concepts with several exploratory prototypes.

The following three prototypes all follow the same functional specification as defined in chapter 3.

Concept 1: Oven

The first concept is based on an oven. To maintain a precise temperature throughout the cooking process, a thermocouple will be employed to measure the inner temperature. This oven must be a convection oven to ensure a uniform temperature distribution. Figures 4.1 and 4.2 show a cardboard prototype of this concept.

The microcontroller used could make use of Wi-Fi to use a phone as control interface to control advanced functions compared to using components that would increase complexity and costs.

Temperature regulation will be achieved by connecting a power relay to the microcontroller. This relay will be placed in series with the oven's resistor, allowing for precise temperature control. Safety concerns are minimal in this concept. The oven manufacturer guarantees its safety, as no additional risks are introduced, and no internal components are tampered with, except for the connection to the resistor, which poses no safety hazards when wired properly.

Due to the thermal inertia of the oven, rapidly changing temperature is challenging. However, its enclosed and insulated design results in substantial energy savings compared to open setups like stovetops.

Pros and Cons of the oven concept:

+ Very well insulated: Since an oven is very well insulated, not much heat will be lost. Allowing the temperature to be controlled accurately.

+ Easy to implement: Integrating the electrical circuit will be easy, as all it needs to do is turn the oven on or off.

- Untargeted heating: the oven heats everything inside, not only the food.

- A lot of heating: the electrical components can be heated too much and get damaged.

- Costs: The costs are high, since you must buy an oven for this design.

Figure 4.1: Inside view of the prototype of Concept 2

Figure 4.2: Side view of the prototype of Concept 2

Concept 2: Gas Stove
This concept is based on a gas stove. The aim is to save energy by avoiding excess heating. To control the temperature, a servo will rotate the gas button of the gas stove. With a thermocouple inside the pot connected to an ESP32, the temperature can be noted. The servo, which is also connected to the ESP32, will rotate the gas button based on the temperature measured by the thermocouple. Figure 4.3 shows a cardboard prototype of this concept.

Safety considerations remain paramount in this concept. As with the other concepts, it must be ensured that the inherent safety features of the gas stove are not compromised. No additional risks may be introduced, and the internal components of the stove must remain untouched, except for the connection to the gas knob, which, when implemented correctly, poses no safety hazards.

While rapid temperature adjustments can be challenging due to the nature of gas stoves, this concept strives to find the optimal balance between energy conservation and culinary excellence. By minimizing excess heat and enabling precise temperature control, this gas stove concept aligns with the commitment to both energy efficiency and culinary satisfaction.

Pros and Cons of the gas stove concept:
+ Simple system: the system by itself would be very simple.
- Hard to regulate temperature: Gas stoves have a more limited temperature control range compared to electric cooktops. It is challenging to precisely adjust the temperature and keep it constant.

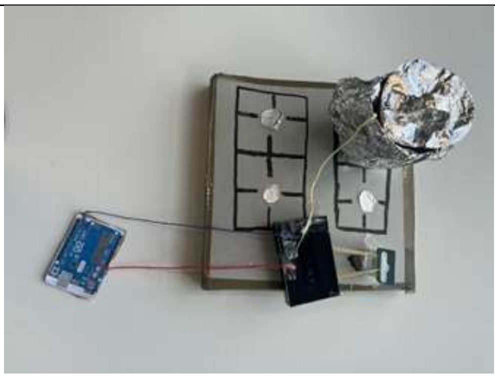


Figure 4.3: Top view of the prototype of Concept 2

Concept 3: Induction stove
This final concept revolves around enhancing the efficiency of induction stoves by leveraging their existing circuitry. To achieve this, an ESP32 microprocessor will be integrated into the system of the cooktop, allowing precise control over power signals and heat output adjustment. This optimization will be driven by real-time temperature monitoring using a DS18B20 sensor, ensuring that the water remains precisely at the desired temperature, thus minimizing energy loss. Figures 4.4 and 4.5 show a cardboard prototype of this concept.

What sets this concept apart is the incorporation of Wi-Fi capabilities enabled by the ESP32. Users can conveniently access and control their cooktop through a dedicated website or mobile application, allowing specific temperature profiles to be selected to suit various cooking needs. This not only enhances convenience but also promotes energy-efficient cooking practices.

Additionally, by designing a user-friendly website that hosts a variety of plans for precise temperature control, catering to a wide range of dishes while minimizing energy wastage, users will have the ability to take control of their cooking processes, empowering them to optimize their culinary endeavors while contributing to energy conservation.

In summary, this concept integrates advanced technology into existing induction stoves, creating a user-friendly and energy-efficient cooking experience through precise temperature control and remote access options.

Pros and Cons of the induction stove concept:
+ Precise temperature control: the use of the thermocouple and microprocessor allows for precise temperature control, ensuring that the dish is cooked at the desired temperature
+ Ease of circuit integration: integrating the microprocessor into the existing circuitry of the induction cooktop is relatively simple.
- Cost: the addition of advanced technological components such as the microprocessor and thermocouple may increase the overall cost.

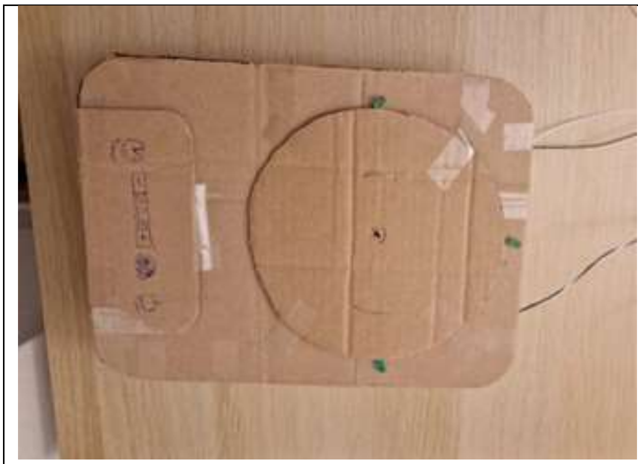


Figure 4.4: Top view of the prototype of Concept 3

Figure 4.5: Inside view of the prototype of Concept 3

5. Final concept design (typically 1 page)

Describe how the final concept design was selected. Show that it meets the functional specification. Highlight its innovative, challenging, and user-friendly character. Include photographs/images of the preliminary design.

The final concept chosen is Concept 3: the induction stove. Figure 5.1 shows the CAD (Computer Aided Design) model made to show the concept design. During a meeting, all prototypes were brought to the table, and the pros and cons of each prototype were discussed. After listing all the advantages and disadvantages, the decision to select the induction stove as the final concept design was made. As this idea was the most innovative, reliable, and the best-working design.

The first prototype that was removed from the list was the oven. The main obstacle with this design was the budget. Since the system would need to go inside the oven, the cost of the oven itself had to be considered in the budget. However, this could not be done without leaving insufficient funds for the other tools required to build the design. Due to this obstacle, it was decided to remove the oven from consideration, despite its positive aspects.

Next, there was a debate between the gas stove and the induction stove. These designs had a few similarities. Both designs would consist of simple circuits, and they would work in a comparable manner. However, the gas stove presented several difficulties, particularly in precisely adjusting and maintaining a constant temperature, which was a crucial factor in the design. The induction stove prototype excelled in this regard, making the decision in favor of the induction stove.

At the end of this discussion another discussion arose. Since the induction stove would be included in the budget that could be used for this design, this could form a problem later on. That is why an electrical stove was introduced in the discussion. This electrical stove is half of the price of the induction stove, and so a safer option to use to stay on budget during the project. In the end the choice was made for the induction stove once again. This was done because the induction stove was a more innovative tool than the electrical stove and the induction stove is likely used more in future households than the electrical stove. Next to that, calculations were done, where out was concluded that it should be possible to stay exactly in the budget by using the induction stove, so this should not be a problem.

Figure 5.1: CAD model of the preliminary design

6. Technical specification (typically 2 pages)

Give a complete technical specification for the final concept design. Use the MoSCoW prioritization method. Show that technical specs are measurable.

For the induction stove design, the following technical specifications were made, ordered by MoSCoW prioritization:

Must:

- The wires must be safe to use at temperatures up to 125 °C.

Should:

- The device should have a lifetime of a regular cooktop or longer.
- The device should be able to boil water in 5 minutes or less.
- The cooktop needs to be supplied with a voltage of 220-240 volts.
- The cooktop needs to be able to provide 1000-2000 watts of energy to the water container.
- The ESP board may not get colder than -40 °C, nor hotter than 125 °C.
- The thermometer may not get hotter than 125 °C.
- The thermometer should be able to accurately measure temperature around 100 °C with an uncertainty of ±1 °C.
- The power supply needs to provide a voltage of 5 volts.

Could:

- The device could be controlled by a website or app.
- The website/app could show a report displaying the energy that is used by the device.
- The website/app could show a report displaying the energy that is saved by using the device compared to other cooking methods.

Won't:

- The device does not need the ability to operate in temperatures above 130 °C.

The must specification can be measured by bringing the wires to a temperature close to 125 °C and checking if they still work without any problems, it should also be checked whether the material surrounding the wires does not show any defects at this temperature.

The should specification requiring the device to boil water within 5 minutes can be measured by using the device to heat water and measuring the temperature, if the water reaches 100 °C within 5 minutes this requirement is met. It can be checked whether the specification requiring a supplied voltage of 220-240 volts is met by measuring the voltage of the source the cooktop is connected to. The energy the cooktop can supply to the water can be found by measuring the temperature of the water when it is heated by the cooktop, and calculating the required energy required to heat the water that much. Checking whether parts of the design stay within certain temperatures can be done by using the device and measuring the temperature of the location where the part would be. Measuring the accuracy of the thermometer at 100 °C can be done by using a thermometer which is known to be accurate at that temperature and checking if both thermometers give the same reading. The voltage output of the power supply can easily be measured using a voltmeter.

7. Detailing (typically 2 pages)

Describe the steps that were taken in the detailing process. Elaborate on the design of 3 key components and show underlying calculations, modelling, or programming. Show how the results led to the optimization of the design.

With the technical specifications determined, it was time to start detailing the design. This included thinking about how the individual components would be connected and used. This process started with listing various options for parts or ideas that could be used in the final design. These options were evaluated to determine which ones were the best for each task. In the end, it was decided that the following three options were the best fit to be used in the final design.

The design makes use of a thermometer to measure the temperature of the water that is contained in a pot. This thermometer is placed in the pot and continuously tells the ESP board (more on that below) what the temperature inside the pot is. The thermometer used is the DS19B20 Temperature Sensor. This temperature sensor can measure the temperature with an accuracy of ±1 °C for temperatures up to 100 °C and thus meets the technical specifications as described in chapter 6.

One of the most important components in the design is the microcontroller. This is the part that makes sure the temperature of the water stays constant. It does this by comparing the measured temperature with the desired temperature and turns the power of the cooktop off or on, depending on whether the measured temperature is higher or lower than the desired temperature. The part that is used in this design is the ESP32-D0WDQ6 (TinyTronics, n.d.). This part is used, because it has a maximum operating temperature of 125 °C, which means it will be able to work within the cooktop. See Appendix A for the code used by the ESP32.

Finally, there is the website that allows the user to input the desired temperature and to run specific temperature plans. The website can be accessed by connecting to the microcontroller using Wi-Fi. Once this is done, the website can be used to turn the cooktop on or off and to activate certain temperature plans for cooking different kinds of food. If it is desired, the website can also be used to manually input a desired temperature as opposed to a temperature plan. The amount of energy saved by using the device compared to regular cooking can also be found. It was a goal to make the website as simple as possible without a loss of functionality, this is the case so that it is easy to use, even for people who are less skillful in using technology.

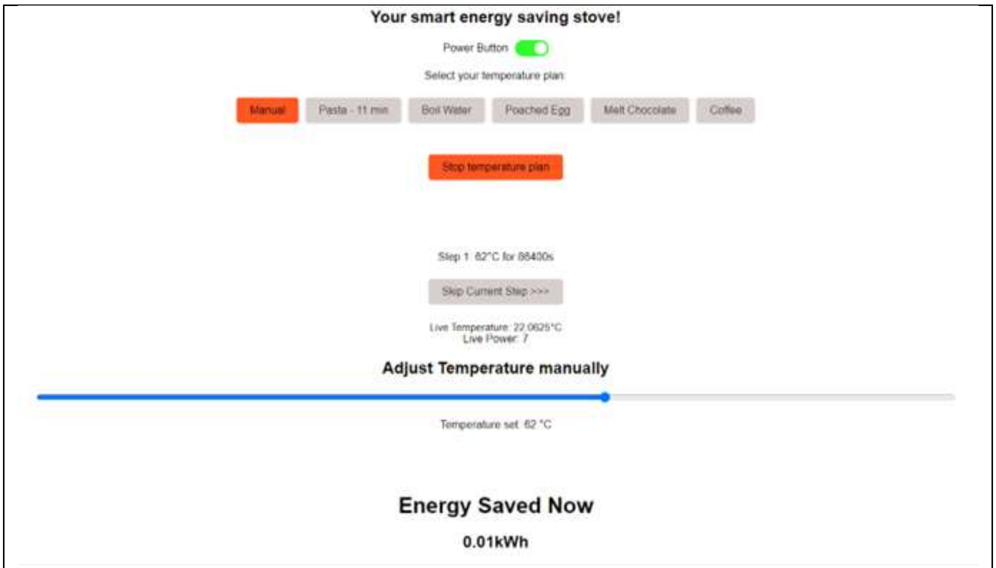


Figure 7.1: A screenshot of the final website

8. Realization (typically 5 pages)

Describe the steps to assemble all the parts into the complete system. Give an assembly drawing (blueprint). Describe the preliminary tests to check the mechanics and electronics (breadboard or Thinkercad modelling). Describe the fit-tests or provide exploded views to check if parts fit or connect. Provide photographs or images for illustration.

List the parts that are manufactured and provide a concise PfP(*). Draw up a BoM(**) for all parts (purchased and manufactured) and compute the total cost required to build the prototype.

Include clear photographs/images of the prototype and some of its details.

(*) PfP (plan for production): a list of manufacturing steps for the parts that are manufactured, including the manufacturing techniques and materials to be used.

(**) BoM (bill of materials): a list of parts (both purchased and manufactured) with prices. Purchased parts should specify name, type, brand and supplier. The prices add up to the total cost.

The realization of the Kompor - Smart Cooktop started by analyzing all the requirements to search for appropriate components, do a cost analysis and finally create a final prototype that includes all the required functions.

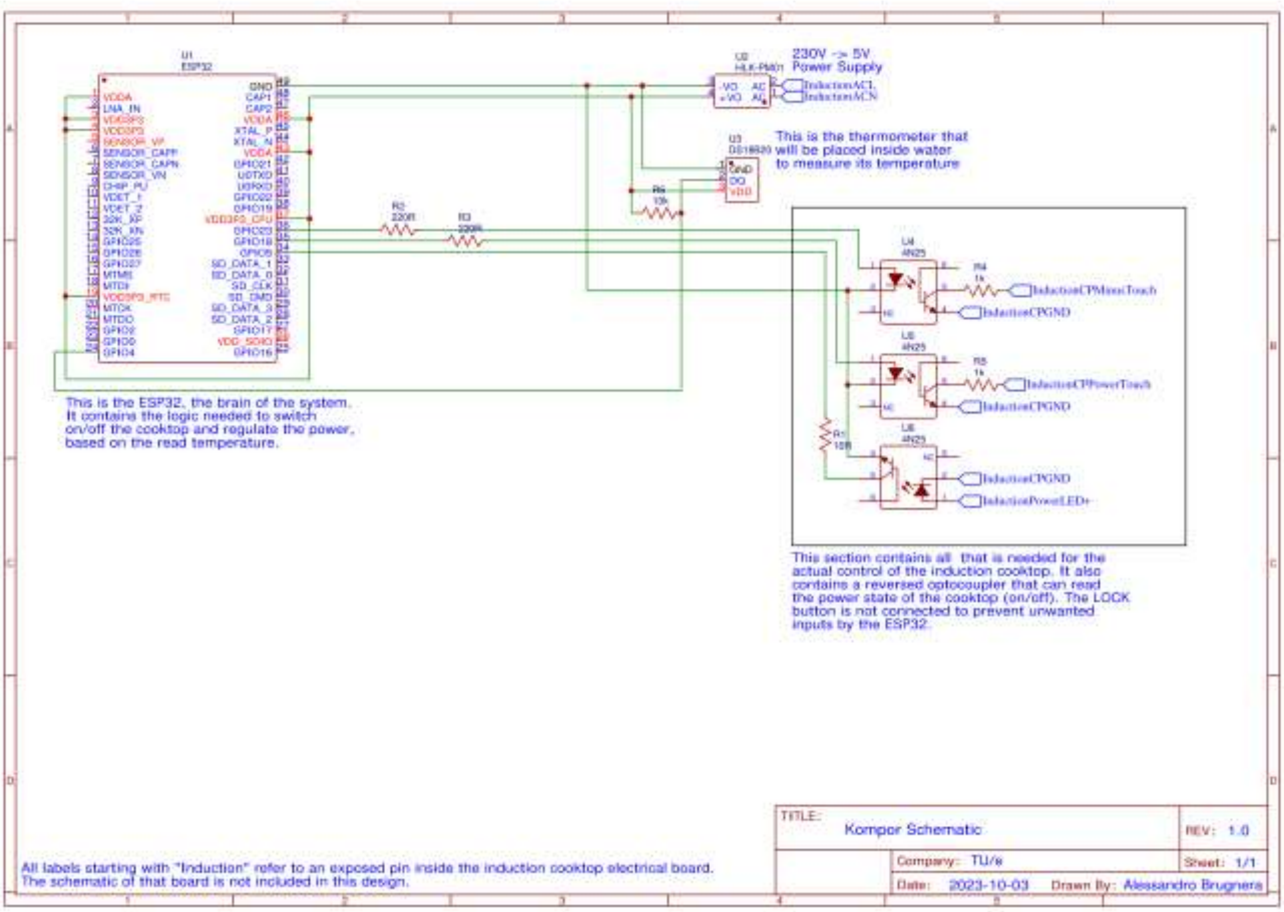
The main component of the assembly is an induction cooktop by Ikea which needed to be modified with the rest of the components in order to add features and enhance the existing ones. By making this choice, security also had to be ensured in regard to the high voltage and high current electronics. After checking that all the requirements were met with the chosen components, they were ordered. The following is a concise list of all the materials, with (when possible) links to the website they were bought from. In Figure 8.1 they have been divided in major and minor parts.

- Ikea Cooktop Single Plate 2kW €40 @ Ikea.com
- ESP32 Dev Board €5.5 @ Tinytronics
- DS18B20 thermometer €6 @ Tinytronics
- HLK-PM01 5V Power Supply €5 @ Tinytronics
- EL817 Optocouplers €0.2 x 3 @ Tinytronics
- Wires (various colors) €0.25 x 6 m @ AliExpress *This insertion is not the one used as that one became invalid. Prices are similar.*
- Resistors (various values) €0.005 x 4 @ AliExpress *This insertion is not the one used as that one became invalid. Prices are similar.*
- PCB Terminal Block €0.1 @ Brought from home
- Zip Tie €0.01 @ Brought from home
- Adhesive Tape €0.02 x 2m @ Brought from home

Total: €59.25

Figure 8.1: Prices of used components divided in major and minor parts.

The first step is designing the circuit that will be added to the pre-existing circuit of the cooktop. This circuit includes all the necessary components and connections to interface with a smartphone via Wi-Fi and with the cooktop via galvanically insulated contacts. The schematic is provided below.



The second step involved manufacturing the circuit. For this, a lot of testing was done to be sure that everything worked as expected and that all the materials that were needed were ordered. After the testing phase, the cooktop was opened, and the test fit was done by including all the components inside the cooktop case. This accounted also for the electromagnetic interference that the big induction coil will generate while operating. This was needed as it was likely that some components would be affected more than others and could work unreliably in certain locations of the case (e.g., directly under the coil). Once all the components had their respective placement it was time for soldering. All the components were soldered while ensuring safe distances with the wires between the HV line and the LV line. The wires have been twisted to prevent the magnetic field from generating unexpected signals in the wires. The thermometer wire was shielded, and the shield was grounded.

The third step (parallel to the second) involved the creation of the webpage graphics and mechanics. Those were needed to have a clear view of the smartphone interface and to define an API interface needed to connect backend and frontend together. This passage involved the creation of all the buttons for the interface, an automatic update system that synchronizes all the connected devices and the look and feel of the website.

The fourth step was to create the backend software. This included all the mechanics needed to control the temperature efficiently (PID controller), the safety, the communication with the devices and the temperature reading. It is divided into various sections, and it is built with energy efficient architecture in mind (RISC V). Each section is responsible for one of the mentioned parts. There is also a boot code which is ran only once in the beginning and is needed to set up the 2.4 Ghz Wireless Access Point and the initial self-tests for safety. The rest of the sections run in a loop configuration, one after another at high speeds since a Dual Core processor is used, which uses one core only for Wireless communication, making it effectively a Single Core processor. A flowchart displaying how the code works can be seen in Figure 8.3.

The fifth step included a lot of testing by making sure that everything worked, the safety measures triggered at the right time, the interface was easy to use, the circuit was stable, and the software worked without crashing.

The last step was to close the case again and attach a label with simplified instructions on how to use the cooktop by providing easy-to-use QR codes for both connecting to the Kompor and later to go to the Kompor web interface.

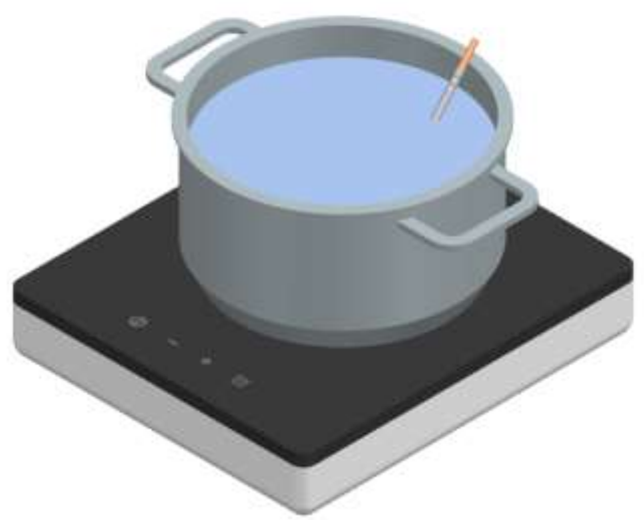


Figure 8.2: CAD model of the realized design

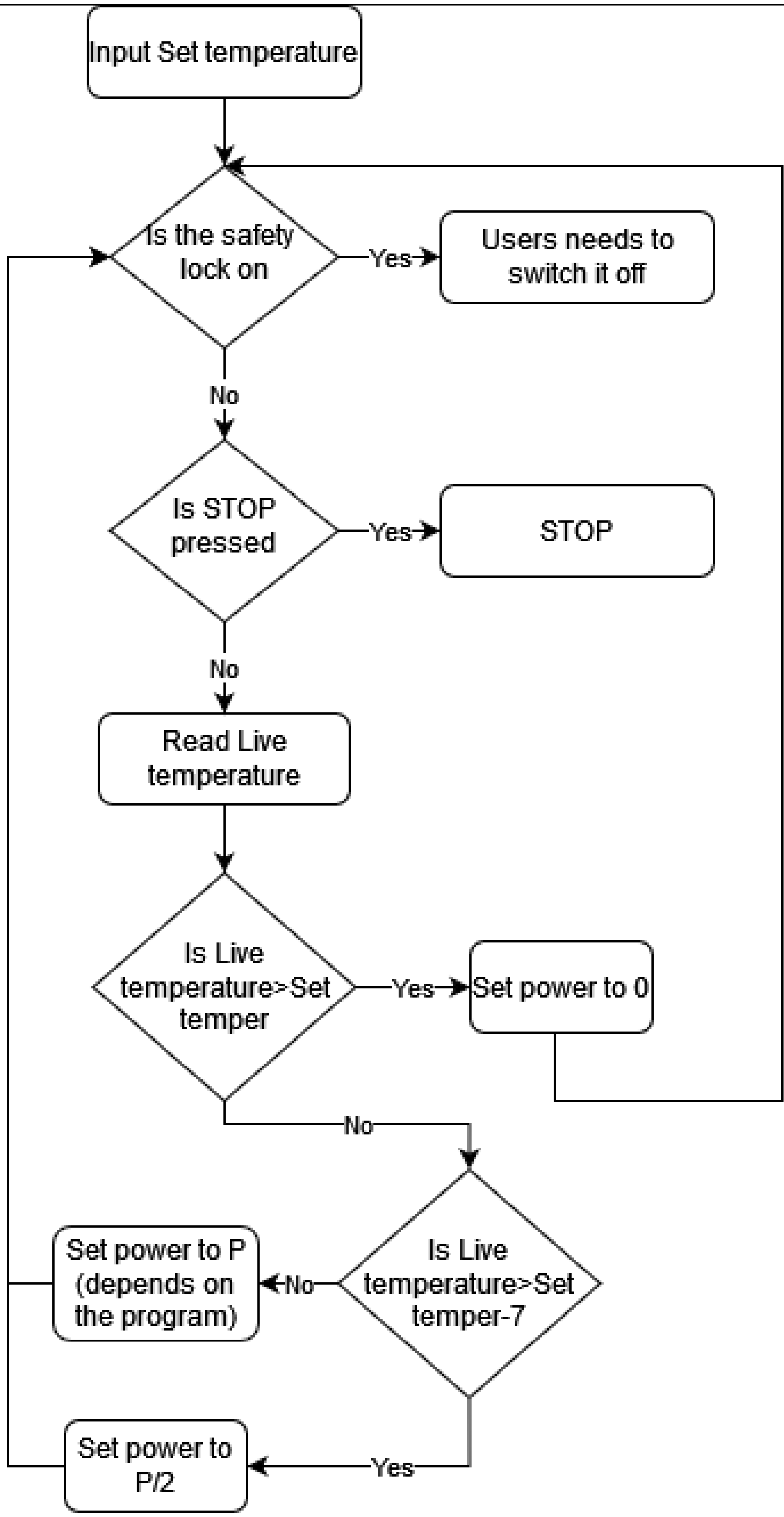


Figure 8.3: Temperature regulation algorithm

9. Test plan (typically 2 pages)

Describe the system test plan for the final prototype. Describe the measuring instruments and the test procedures. Provide photographs, images, or drawings if required. Explain how the tests cover all functional and technical specifications of the prototype.

First the components purchased were tested individually to make sure they were not faulty. The results of these tests can be found in Table 9.1 below.

Table 9.1: Unit Test

Component	Tests	Results	Instruments used	PASS /FAIL
Ikea Cooktop Single plate 2kW	Power up the cooktop and check that the lock, power, plus and minus buttons all work as intended.	If the lock button is not pressed the device does not respond. The power button does turn the heating on and off. The plus increases the power level by one and the minus decreases the power level by one. A bug was found that sometimes the cooling fan continues operating even if the cooktop is shut down. Turning the cooktop on and off resolves the issue	-	PASS
ESP32	The ESP32 was tested with the product.	The pin I/O worked correctly. No problem was discovered.	-	PASS
DS18B20 thermometer	Compare the temperature reading with a conventional thermometer.	The DS18B20 has the same reading as the conventional thermometer in a sufficiently quick time, less than a second.	Analog thermometer ESP32 Laptop	PASS
	It was placed into water, both with and without salt.	It is water and salt resistant. It does not react chemically or leave taste in the water.		
	Measure increasingly hot temperatures.	It can accurately measure the temperature up to at least 120 °C, which suffices.		
HLK-PM01 5V Power Supply	Connect the input pins to 230 V AC from the plug and measure the voltage on the output pins.	The voltage was close to 5 V DC.	Digital Multimeter	PASS

Optocouplers	The optocouplers were tested directly with the prototype. Once the positive output pin was soldered to the power button of the cooktop and then to the ground emulating human touch, the correct amount of voltage was inputted (little over than 2V) to switch on the optocoupler.	Once the optocouplers were properly wired, the power switched on when voltage was applied to the input and off when no voltage was applied.	Digital Multimeter Power Supply	PASS
Resistors and capacitors	Measure the values using a digital multimeter.	The values were as close as possible to the labeled ones.	Digital Multimeter	PASS

Then tests were done to make sure the product works as a whole.

Table 9.2: Prototype Testing

Function	**Tests**	**Results**	**PASS/FAIL**
Ordinary Use	Check that the induction cooktop's normal functions work.	The modifications and additions made to the device have not altered any of its original and normal usage (power usage, increasing power, decreasing power, and locking the device)	PASS
Basic features	Connect to the website. Power it up using the website. Try to operate it without disabling the safety lock. Check if compatible with ordinary use is impacted	It takes a few seconds to connect to Kompor's Wi-Fi, but the website loads correctly. When the safety lock is on, pressing the website's buttons does nothing. The website's control can always be bypassed when pressing the physical buttons on the device	PASS
Displays	Check if all the displays are correct: Temperature, power level, Energy saved, meal plan status and step.	The live temperature is displayed and updates accurately. The power level displayed on the website is the same as the one on the device when setting manually or through the temperature plans. The energy saved is displayed and works as intended	PASS
Manual use	Set temperature manually. Measure how long it takes to boil water. Does it reach the set temperature? Does it reduce the power level when approaching? Does it stop when the set temperature is reached, and does it turn on again to maintain	The water consistently starts boiling under 4:30 minutes. When the water reached 7 °C below the set temperature the power was halved. As soon as the desired temperature was reached the power was cut off and when it fell below the desired temperature the power was turned on again. Hence the temperature regulation works. The stop	PASS

	it? Does it terminate when the stop button is pressed?	button terminated the manual setting and switched off the power	
Temperature plans	Test temperature plans with water. All temperature plans do what they are supposed to. Do they terminate when the stop button is pressed?	All temperature plans set the power and temperature correctly. They mention what is the current step and how long it is. The time given does not consider the time it takes to boil water. They immediately terminate with the stop button.	PASS
Cook pasta	Final test, can it cook pasta? Is the pasta well cooked? Is it tasty? How long did it take? How much energy was saved?	The pasta was slightly undercooked, but the taste was good. It took approximately 14 minutes, but in other cases this depends on the volume of water and the shape of the pot. The energy saved was 0.42 kWh	PASS

The above tests cover all the testable functional and technical specifications. For the should section, in Table 9.2, one of the results of the manual testing was that the water boiled in under 4:30 minutes, which is under 5 minutes. The only power supplied to the cooktop was 230 V AC from the powerline. The accuracy and upper limits of the thermometer and the power supply were tested, as described in Table 9.1, and met the required specifications. During all tests in Table 9.2 except the first one, the device is tested remotely using the website. During the second to last test, the temperature plans were confirmed to work properly.

The functional specifications were also met. Starting with the must, the lock button was tested and so was the ability to manually, from the device or the website, turn off the power. This, along with the built-in safety features, makes the device safe to use. As described in Table 9.1, the only device in the water, the thermometer, does not chemically react and all bacteria are exterminated in nearly boiling water, so the food and water are safe to consume. Throughout the tests no problem with the heating was discovered. Due to the cooktop being an induction stove, energy is transferred directly to the pot through electromagnetic induction.

Most tests described in Table 9.2 used a pot with water and heated it up. In Table 9.1 the tests involving the thermometer can be found, where it was tested whether the thermometer can accurately measure the temperature. In the manual test, the ability of the device to read the temperature and adjust the power accordingly was successful.

10. Design evaluation (typically 2 pages)

Evaluate your design in view of the intended design goal. Did you and your group succeed in making the best product you possibly could, has it become the creative, innovative, user-friendly product you envisioned?

Suggest at least 3 improvements in the design. How will these improvements influence the next iteration of the design cycle?

Important: don't be defensive – the capacity for self-criticism is a valuable skill!

The product design is a promising step toward achieving the goal of energy-efficient cooking by addressing the issue of unnecessary energy loss during the cooking process, particularly related to water temperature control. Here is an overall evaluation of the product:

Strengths:

Innovation: The product incorporates innovative features such as real-time temperature thermometer, precise temperature control, and remote access through a website. These features are aligned with the goal of energy-efficient cooking and demonstrate a creative approach to cooking appliances.

Energy Efficiency: The core focus of the product is energy efficiency. By maintaining a precise water temperature and avoiding unnecessary fluctuations, energy loss is minimized during cooking. This aligns with the aim of reducing energy consumption and saving money for users.

User-Friendly: The inclusion of Wi-Fi capabilities and a user-friendly website makes the product convenient and accessible for a wide range of users. The product's simplicity in operation is a significant advantage, as it ensures that even those less familiar with technology can benefit from it.

Safety: The product pays attention to safety concerns by specifying that wires must be safe to use at temperatures up to 125 °C. Also, safety controls were integrated into the programs of the ESP2 which can shut down the power when anything goes wrong. This safety aspect is important for a cooking appliance.

Improvements:

- An improvement could be the way the thermometer is attached to the device. In case the thermometer breaks down or stops functioning the cooktop must be opened and the thermometer must be uncoupled. In a further design this can be improved. For example, by having a slot with pre connected wires to the device. The user only has to connect the (new) thermometer to this slot. This way the thermometer could be easily replaced by the user itself. This would reduce the carbon footprint of the device and have a better impact on the lifespan of the device as well. Since the device can still operate after a small easy fix by the user itself.
- A second improvement could be to add more meals and temperature plans. The main focus was on pasta but there are so many other recipes that require boiling water and can be optimized using the device.
- A third improvement would be to include a thermometer that can handle higher temperatures, to vastly broaden the possible use cases. For example, it could be used when cooking with oil that reaches temperatures far above 100 °C. The reason this was not implemented was that the thermometers that can reach higher temperatures are less accurate, during this project accuracy was found to be more important. With a higher budget two thermometers can be used so that accuracy at lower temperatures is not compromised.

- Finally, a feature could be added on the website to add programs. That would make implementing new code much easier, because in the current version the device must be opened, which is cumbersome. It would make it much more convenient to add new functions and meals. Moreover, a user with some programming knowledge would appreciate such a feature. However, this cannot be overused, as the ESP32 has limited times it can be written on, although it is in the tens of thousands.

In summary, the product shows great promise in terms of innovation, energy efficiency, and user-friendliness, aligning with the intended design goal of creating an energy-efficient cooking solution. However, there are still some improvements, which could be included in further iterations.

11. Individual contributions (typically 3 pages)

Describe each group member's individual contribution to the project. Describe what material was studied or which skills were developed, what were the learning achievements, and how did it help to improve the design process or the development of the product itself. Indicate when during the design process the topic was studied and when the results were implemented. Refer to previous chapters where appropriate. (No need to add a name to each contribution)

One of the students worked on the CAD design CBL module. This helped improve the design process, because it made it possible to create clear visualizations of the design. It was specifically useful when preparing the midterm and final presentations, as during these presentations it was very important that the idea what clear to those who did not participate in the creating of the design. It was also useful while writing the report, because here it was particularly important to make ideas clear as well. Chapters five and eight show some of the CAD models created for this project. The total workload was not great enough to be problematic regarding other aspects of the project work, as the number of models needed was not large.

The second student chose the 'Project Risk' module to gain insight and supported the group with this information. In addition to participating in group tasks such as soldering and manufacturing, the student assumed sole responsibility for this 'Project Risk' CBL module. Consequently, he also contributed to other aspects of the problem-based learning project. The workload associated with the module did not prove overly burdensome for the student. By reading and studying the relevant literature and watching instructional videos on risk analysis, he facilitated his understanding of the subject matter. Subsequently, he compiled all the acquired knowledge into a project risk report, wherein every detail was outlined and elucidated. During the meetings, this report was discussed several times, and feedback was provided by other students. Additionally, further research was conducted to address the remaining questions in the report. The CBL module held significant importance and relevance for their project, particularly in the context of potential product marketing. It also offered valuable assistance in identifying possible project risks, such as issues related to time management, budget constraints and infringement around patents. For example, the student ensured that the project budget remained comfortably below 70 euros, allowing for coverage of price fluctuations or unforeseen expenditures.

The third student worked on the CBL module of Sustainable product development. This is to gain more knowledge and find possible improvements regarding the device's sustainability. This furthermore gave insight into the overall carbon footprint of the device and looking into the different logistic paths to distribute the device to the public. This gave insights into the various parts of the production and logistic steps and their associated emissions. The conclusion of this report has been used for one of the improvements to the device. The student also worked on the front-end of the website by making use of HTML, JS and CSS code. The website was one of the crucial factors to make sure that the device could function. The student also edited the videos for the midterm and final presentation. This to show the audience a better understanding of the device's concept, production and working.

The fourth student's main responsibility was to solder most of the circuit parts, which was most of the product manufacturing. There the Basics of electronics CBL was useful because all the electronic components needed to be soldered correctly. The workload was not too much, although he had to learn how to solder. The actual work was done in one day of focused and long work. A secondary responsibility was to perform calculations about the energy the product is saving. Using principals from the heat transfer section of the Physical Modeling

CBL he was able to come up with some graphs and numbers. The workload should not have been that much but due to unmaterialized ambitions it took up a good part of two days.

The fifth person has done the CBL: "Human Factors Design to Enable, Facilitate, or Encourage Sustainable Behavior", this CBL broadened the perspective of looking at the design of the product, by focusing on the user's perspective. They did not have much experience in this field but read a lot of chapters of the book mentioned in the CBL to fully understand the theme. This way they were able to fully reflect on the design. This was done in the last two weeks, so not much could be changed on the design, but there was a lot of reflecting, mainly regarding the clearness of the website, as there is not much explanation on it, so more information was included. They also worked on the safety of the product, researching the safety of induction cooktops, electrical circuits in general, etc., everything that was thought of. The CBL also helped to look at the safety of the product from the user's perspective, which helped make the product safer. Also worked on the presentation and report.

The sixth student has completed the CBLs Basics Of Electronics and Sensing And Computing. Basics Of Electronics helped him to design the schematic for the entire system. Sensing And Computing on the other hand helped him to choose the components to be used in the circuit, in particular the part where all the sensors are described. The computing part instead was helpful for the second responsibility which was to design and create the software. The software part of this student was to design and create the bootstrap code, the interfaces between backend and frontend and between software and hardware. As a side responsibility he had also used the Basics Of Electronics when soldering. He had to do only a small part of it when difficult components (optocouplers) needed to be soldered.

The last student worked on the CBL programming design. The knowledge of programming language helped him to build basic software for tiny electronics. To be specific, he spent 7 hours learning micro python, 3 hours having meetings to work on solutions for technical specifications and 5 hours on programming the main file for esp32. The workload is not too much, although he started by learning, the whole work was done in two days.

12. Statement about use of AI

State if and how AI tools, like OpenAI/ChatGPT, were used in this project.

Throughout this project the usage of AI tools was limited. AI was used to point out any spelling or grammatical errors in text not caught by a human reading it. Furthermore, AI was used once to gain a brief explanation of a topic related to heat transfer.

13. References

You may use any referencing style and add references in this chapter or as footnotes to the text.

DS18B20 Temperature Sensor with Cable - Waterproof - High temperature - 2m. (n.d.). TinyTronics. Retrieved October 23, 2023, from https://www.tinytronics.nl/shop/en/sensors/temperature/ds18b20-to-92-thermometer-temperature-sensor-with-cable-waterproof-high-temperature-2m

Easy ways to save energy while cooking. (2020, June 17). Energized by Edison. Retrieved October 28, 2023, from https://energized.edison.com/stories/easy-ways-to-save-energy-while-cooking

ESP32-D0WDQ6 Development Board - with Li-ion Li-Po Charge Circuit. (n.d.). TinyTronics. Retrieved October 23, 2023, from https://www.tinytronics.nl/shop/en/development-boards/microcontroller-boards/with-wi-fi/esp32-d0wdq6-development-board-with-li-ion-li-po-charge-circuit

Here's the trick to saving money cooking pasta, according to Nobel-Winning Scientist. (2023, January 12). Tapas. Retrieved October 28, 2023, from https://www.tapasmagazine.es/en/trick-saving-money-cooking-pasta-scientist/

Nt, L. M. B. H. H. (2023, October 20). *Which is the More Energy Efficient Stovetop - Gas, Electric, or Induction?* LeafScore. Retrieved October 28, 2023, from https://www.leafscore.com/eco-friendly-kitchen-products/which-is-more-energy-efficient-gas-electric-or-induction/

Water based cooking methods. (n.d.). RevisionWorld. Retrieved October 28, 2023, from https://revisionworld.com/gcse-revision/food-preparation-and-nutrition-gcse-revision/food-science/water-based-cooking-methods

14. Appendices

Appendix A.

The code used by the ESP to operate. The programming language used is Python.

File: boot.py

```
# This file is executed on every boot (including wake-boot from deepsleep)
import esp esp.osdebug(None)
import webrepl
webrepl.start()
```

File: main.py

```
import network
import usocket as socket
import esp
import gc
import time
import ds18x20
import onewire
import machine
import json

# Disable debug output
esp.osdebug(None)

# Enable garbage collector
gc.enable()

machine.freq(240000000)

# Configuring the Access Point Interface
#network.hostname("kompor")
ap = network.WLAN(network.AP_IF)
ap.config(essid='Kompor - Smart Cooktop')
ap.config(dhcp_hostname="kompor")
ap.config(max_clients=5)
ap.active(True)

# Configuring the ESP32 to listen on port 80 (standard HTTP port)
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.bind(('', 80))
server_socket.listen(5)
server_socket.settimeout(0)

# Setting up OneWire interface + DS18B20 interface
ow = onewire.OneWire(machine.Pin(4))
ds = ds18x20.DS18X20(ow)
thermometers = ds.scan()
for i in range(10):
    thermometers = ds.scan()
    if len(thermometers) == 0:
        time.sleep_ms(2000)
    else:
```

```
        break
#print(thermometers)

powerPin = machine.Pin(23, machine.Pin.OUT, value=0)
minusPin = machine.Pin(18, machine.Pin.OUT, value=0)
detectPowerPin = machine.Pin(5, machine.Pin.IN, machine.Pin.PULL_UP)
HEATER_ON = 0
HEATER_OFF = abs(1 - HEATER_ON)

current_power = 0

# How to activate heater: turn it off, then on, the set power (Going back to a known state
before changing power)
def activateHeater(power=9):
    if power > 9 or power < 1:
        return
    global powerPin, minusPin, detectPowerPin, current_power

    if current_power == power:
        return

    if detectPowerPin.value() == HEATER_ON:
        powerPin.on()
        time.sleep_ms(200)
        powerPin.off()
        time.sleep_ms(200)
    if detectPowerPin.value() == HEATER_OFF:
        powerPin.on()
        time.sleep_ms(200)
        powerPin.off()
        time.sleep_ms(200)
    for i in range(9 - power + 1):
        minusPin.on()
        time.sleep_ms(200)
        minusPin.off()
        time.sleep_ms(200)
    current_power = power

def deactivateHeater():
    global powerPin, detectPowerPin, current_power
    if detectPowerPin.value() == HEATER_ON:
        powerPin.on()
        time.sleep_ms(500)
        powerPin.off()
    current_power = 0

# This function keeps the ap active with an error message
# and halts the rest of the code until a power cycle occurs
def fail_with_error(err="ERR"):
    global ap
    deactivateHeater()
    ap.config(essid=err)
    while True:
        time.sleep_ms(500)

if len(thermometers) == 0:
    deactivateHeater()
    fail_with_error('ERR: No Therm')
```

```
# Program steps: [temperature, duration, max_power, start_time_immediately]
programs = [
    {"name":"Manual", "steps": [[0, 86400, 7, False]]},
    {"name":"Pasta - 11 min", "steps": [[90, 3600, 9, True], [85, 600, 6, False], [82, 300,
4, False]]},
    {"name":"Boil Water", "steps": [[95, 3600, 9, True]]},
    {"name":"Poached Egg", "steps": [[90, 1, 9, False], [90, 3600, 4, False]]},
    {"name":"Melt Chocolate", "steps": [[40, 3600, 6, False]]},
    {"name":"Coffee", "steps": [[50, 200, 4, True]]}
    ]

running_program = -1
current_step = 0
current_temperature = 0.0
step_start_time = -1

energy_used_theory = 0
energy_used_for_program = 0
energy_used_dt = 0

last_temperature_read = 0
failed_temperature_readings = 0

# Similar to the loop() function in arduino...
while True:
    # Accepting and processing connections
    try:
        if gc.mem_free() < 102000:
            gc.collect()
        conn, addr = server_socket.accept() # Accept a connection if any
        print('Got a connection from %s' % str(addr))
        print(gc.mem_free())
        request = conn.recv(1024)
        request = str(request)
        print('Content = %s' % request)
        conn.send('HTTP/1.1 200 OK\n')
        conn.send('Connection: close\n')
        response = '404'
        if request.find('/stop') > -1:
            deactivateHeater()
            running_program = -1
            current_step = 0
            conn.send('Content-Type: application/json\n')
            response = json.dumps({'ok': True, 'result': True}, separators=(',', ':'))
        elif request.find('/status') > -1:
            conn.send('Content-Type: application/json\n')
            response = json.dumps({'ok': True, 'result': {'status': (1 if running_program
!= -1 else 0),
                                                          'program': running_program,
                                                          'step_index': current_step,
                                                          'step':
programs[running_program]["steps"][current_step]         if         current_step         <
len(programs[running_program]["steps"]) else [],
                                                          'temperature':
current_temperature,
                                                          'current_power': current_power,
                                                          'energy_used':
energy_used_for_program,
```

```
                                                              'energy_saved':
(energy_used_theory - energy_used_for_program),
                                                              'manual_temperature':
programs[0]["steps"][0][0]}}, separators=(',', ':'));
        elif request.find('/run') > -1:
            program_number  =  int(request[request.find('/run')  +  4:request.find('  ',
request.find('/run'))])
            response = json.dumps({'ok': True, 'result': program_number}, separators=(',',
':'))
            running_program = program_number
            current_step = 0
            step_start_time = -1
        elif request.find('/manual') > -1:
            temperature  =  int(request[request.find('/manual')  +  7:request.find('  ',
request.find('/manual'))])
            response  =  json.dumps({'ok':  True,  'result':  temperature},  separators=(',',
':'))
            programs[0]["steps"][0][0] = temperature
            running_program = 0
            current_step = 0
            step_start_time = -1
        elif request.find('/programs') > -1:
            conn.send('Content-Type: application/json\n')
            response = json.dumps({'ok': True, 'result': programs}, separators=(',', ':'))
        elif request.find('/next') > -1:
            conn.send('Content-Type: application/json\n')
            current_step += 1
            response  =  json.dumps({'ok':  True,  'result':  current_step},  separators=(',',
':'))
        elif request.find('/chart.js') > -1:
            conn.send('Content-Type: application/javascript\n')
            conn.send('Cache-Control: max-age=31536000\n')
            conn.send('Content-Encoding: gzip\n')
            page = open('chart.js.gz')
            response = page.read()
            page.close()
        else:
            conn.send('Content-Type: text/html\n')
            conn.send('Content-Encoding: gzip\n')
            page = open('home.html.gz')
            response = page.read()
            page.close()
        conn.send('\n')
        conn.sendall(response)
        conn.close()
    except:
        time.sleep_ms(5) # Don't overload the processor when there is nothing to do

    # Reading temperature from thermometer
    try:
        if time.ticks_diff(time.ticks_ms(), last_temperature_read) > 1500:
            last_temperature_read = time.ticks_ms()
            if len(thermometers) > 0:
                ds.convert_temp()
                time.sleep_ms(200)
                current_temperature = ds.read_temp(thermometers[0])
                failed_temperature_readings = 0
                #print(current_temperature)
    except:
```

```
        #print('Reading failed')
        failed_temperature_readings += 1
        if failed_temperature_readings >= 5:
            fail_with_error("ERR: Therm Read Failed")

    # Safety check
    # This works even without a program running.
    # The user will need to remove the thermometer from the food before going higher
    if current_temperature >= 110:
        fail_with_error('ERR: Temp Too High')

    # Safety check
    # A human turned the heater off manually while program was running
    if current_power > 0 and detectPowerPin.value() == HEATER_OFF:
        deactivateHeater()#fail_with_error('ERR: Human Power Off')
        running_program = -1
        current_step = 0

    # Running programs...
    if running_program > -1:
        if current_step > len(programs[running_program]["steps"]) - 1: # We reached the end
of the program
            deactivateHeater()
            running_program = -1
            current_step = 0
            step_start_time = -1
        else:
            dt = time.ticks_ms() - energy_used_dt
            energy_used_theory += (2 * dt) / 3600000
            energy_used_for_program += ((current_power / 9) * 2000 * (dt / 1000)) / 3600000
            energy_used_dt = time.ticks_ms()
            step = programs[running_program]["steps"][current_step]
            if step_start_time == -1: # We are waiting for the food to heat up to the minimum
temperature
                if current_temperature < step[0] and not step[3]:
                    if abs(current_temperature - step[0]) < 5:
                        activateHeater(max(1, step[2] // 2))
                    else:
                        activateHeater(step[2])
                else:
                    step_start_time = time.ticks_ms()
            elif time.ticks_diff(time.ticks_ms(), step_start_time) >= step[1] * 1000: # We
finished the current step
                current_step += 1
                step_start_time = -1
            else: # We are waiting for the current step to finish
                if current_temperature < step[0]:
                    if abs(current_temperature - step[0]) < 5:
                        activateHeater(max(1, step[2] // 2))
                    else:
                        activateHeater(step[2])
                else:
                    deactivateHeater()

    else:
        energy_used_theory = 0
        energy_used_for_program = 0
        energy_used_dt = time.ticks_ms()
```

File: home.html

```
<!DOCTYPE html>
<!-- This file has JS and CSS all within it, this can later be splitted if wanted. -->
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0,
minimum-scale=1.0, maximum-scale=1.0">
    <title>Kompor</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: white;
            margin: 0;
            padding: 0;
        }
        header {
            background-color: white;
            color: #fff;
            text-align: center;

        }
        #logo {
            width: 60%;
            margin-top: 2rem;
            height: auto;
        }
        nav {
            text-align: center;
            margin-top: 20px;
            display: none;
        }
        a {
            text-decoration: none;
            color: #333;
            background-color: #D7CFCD;
            padding: 10px 20px;
            margin: 5px;
            border-radius: 5px;
            display: inline-block;
        }
        .buttonlight.active, #stopButton {
            background-color: #ff5722;
            color: #000;
        }
        .toggle-container {
            display: flex;
            align-items: center;
            justify-content: center;
            margin: 20px auto;
        }
        .toggle-label {
            margin-right: 10px;
        }
```

```
    .switch {
        position: relative;
        display: inline-block;
        width: 50px;
        height: 24px;
    }
    .switch input {
        display: none;
    }
    .slider {
        position: absolute;
        cursor: pointer;
        top: 0;
        left: 0;
        right: 0;
        bottom: 0;
        background-color: #ff2819;
        transition: 0.4s;
        border-radius: 34px;
    }
    .slider:before {
        position: absolute;
        content: "";
        height: 18px;
        width: 18px;
        left: 3px;
        bottom: 3px;
        background-color: white;
        transition: 0.4s;
        border-radius: 50%;
    }
    input:checked + .slider {
        background-color: #2EFE2E;
    }
    input:checked + .slider:before {
        transform: translateX(26px);
    }

    .slider-container {
        text-align: center;
        margin: 100px;
        display: none;
    }
    #tempslide {
        width: 20%;
    }
    #temperature-value {
        margin-top: 20px;
    }
    #live-temperature {
        text-align: center;
        margin-top: 20px;
        font-size: 24px;
    }
    #live-temperature{
        font-size: 15px
    }

</style>
```

```
    <!-- <script src="/chart.js"></script> -->
    <!-- Graph script -->
</head>
<body>

    <header>
        <img                                                                           id="logo"
```
src="data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAQsAAABVCAYAAABepsFhAAAAAXNSR0IArs4c6QA
AAARnQU1BAACxjwv8YQUAAAAJcEhZcwAADsMAAA7DAcdvqGQAABiqSURBVHhe7Z15rFXFHcfPQbYHj+2xP3iIoA8RE
QVUlE19FtQiiOKCyiJFUBBR3ACJtYogroiiIC7gLq64xGqa6h+kpkkTY2oabWOTtrbWbmmrrrdqov85v7u/cN2fmN2e
/l8d780m++f3uPTNzzz3LnNnOjKcC99wDMHs2wOGHA3TqBCC+SqTBgwHmzgXYtk18jAZ+8INS+j178mlx6tMHYMIEg
MsvFx8dDsc+Q97A+g2aVdOnC2MCy5bx4TOIkn0Q4HNUEFi5kb8hcWr9emZkiUUPk0cNDcI4HI6Q4HNUEFi5kb8hcWr9emZkiUUPk0cNDcI4HI6qYtyIRWjqVGGagaY
mPlwOUdIOh6NKtCNbLAMGkEN07EhOQYwYQY7D4agW7bzjjye3OPxnn/XJLTFrFjkFMWcOOQ6Ho1rImxpOOQW8H/9Yf
pGL2lrP/+KLcEZBZKo6dO/ueb17e15NTelzQ4Pnv/kmm77D4agSMG0a2zaQSvfdJwyPETZKgwcL43A4WixYwjBu3BS
iZAzgjDPY8KxmzxbG4XC0aGDLFv4GTqJDDhGGxwhr06JFwjgcjv0C4wZOKlGNoSRCwBFH8OF1XXKW5YdwOx6HfM74Avm/u66BBBALNmGSNUKwVs2AB
wzjkAxx5b+m0mQ+aA7dsB5s0DmDIFYOhQgJEjAU44AeCaazLtN5x3HsDxxwMcfDDAoYcCzJgBcMUVkQ3aeZBp4/7iu
cD3ghoaAI47zvpbcoQw/t+xYwE6d24+X/j+0amnAtx9d0X2MwAefxzgooosAGhsB6urC1wyqXTuA4cNBHsc77qjoviC
wejXIBzWe/4MOAujfH2D8+Mjflft/zDEA/fqBbLtcuzZ+P+HII80/m0RXXilMCdixcgw+jK+ICqBbw/e/z+5ZEeEGLC
4WSyo3MYDFj0H9HEQVlgZkz2TiGLrxQmGgw8zficcJM7JFHhJsduP9P+Hs/3CNCNvPj2MTaI6TNEKAyZO5H8
ricRDh5LJDVbhYfRo/ndUbd0qTJj5SHHq3l0Y4Vx9Nb+dk3HkSqRCU9Q6rTTJZnKQz3mrxmkWf4MGPm+EmNPMNnEUvwgqQL0
4qiW5FDuLU4iVTEm82ctPExxvY4NTYKw5Nrsmpt06NTEKjW5NrsmptEJWKETahKLqHvVb6NquoilXl1swEfPGFByNGyJ3LCjesNTcdOpC
TkjffJEejXz9yUjJmDDka06aRk5KGBnKqyPDh5KREn9tVZdQocjLwqn5Wz8hpyAmT
CBH7Nfevb536qn0KYIFCzx/0SL5H1peZoF89JEsKn9LzxBjkFMmUKOcXgz5uX7SIaNIgcjWJr17kMPTP
B9OxJTkqi4k2aRE4GPvuMaRE4GPvuMHMHrr5OTgj/9iRyNf/5wHcgriz38mp4T/xhu/u9119EmOjTc0/l9s5M7Hb7
H3MT3/qwfz52TIM9eSnAByMyDWZPjKmOD7CGKDjHsCoHxpqaWeveQzCSAv9j4VdDbt/9MsrDlpntOGc8KaQEs
OM/6mTbKrF+Vt3Vru+vU/+6ycSQOPnaaYtY3kuvxh7BCqdz8Fj5EQDWk8e0F0sdHFxxTngIC3FcAIFSDXbfMrT78tdWvrF59hO6LcqyOmS9m
kXXruSkJCKz8HfsyH5+1GPw+efkpAMeffkpAMefNB+nY4eTU7U5JlNEaQkWXTPAf+Skx1+1qvnJ2J3a4LIs
nfLdu5BRMly7kpCQis8j6cpikb19ychcBx7fhr1rSMB01OzMwiY84qEReBOCq+HBk3fjx9X7vdEYAB+bZQvTA
RM5BZC1iGnbh6++Ic1lUf8J28IrQefO5KQkqhoS1fgZR9bGZpW4a6cVYGYWav0tLWvXSuMj7KXXHwqH76Ps4sBK
pkvzKHTiQnIKpryenALJmFrbV3bJmFlFD4bOmGUfwzCKqWvTpp+Skx1+1qvnJ37490OSn56CNyCiRr206FKDSz8NetC
xW35IJARa14Jm6uqH72EDluapyMhlwTXPAoLY2N5GP65FISo4HTeQI5CxVuql
TyWkZmJnFH/9ITjH4P/uZb6FJ0ciaWdhu4nwMW8m2gxb7eFr+1OR9+B/ts0+ll13heah9hub+S+7GmT/rsp2cMg4rZkshZi/f
fLSS+SkRG98zFN9zfmjrp2Mjxo/NdfL7atI2dJ0cwsKtB16v/6135hjWX4lqpl2YEA7A4iNz1RT6gMTy+9tFUma2Zhi/f
11+SkJCqzyFMljSLjvkY2Yn7yCTkpaWoih0gyUMnGO++Qw5D22snTBmPj738nJxuzAIefzjfzd9fCzNtcDLv
vWWfHeBPvFkqTIQ+F4JuWF++UtyEnLVVeQw2EZ2xmF72metzvMvzzvwb+Qw/Q95BRM1nR/+1tywtiGwCfB37w5dF36S5b
kuk5tb7Gm/s/r1pFjknke0ffeI6cAcr86bZVdGbCHHHuLjZdWwYcLwmydcd9fBBppMxWjuAAMSNMlMaNE8YOtr8YcZJIm
4MzIPX+KaIkDPRwiRXTjZnrZbKBA+WkNXIuT5wxbMgQPlwSWRa2yj1P68yZwjQD3/seHJhJl0KUTJnc8zMyNzcW7Y1
P1hooZl6KXDMz22RZUkCu1aCHTSOcPRnX3OC2xYh2wYoePrHEzUFJhMg1vygzRgNfrjLCJVXMEg8nTQ5qWpqhLFjh
E8rXD8ky5yjES/2IXFLRSTSWWcJkxFc+MRIMI+YeRhV5BNBj1OAKPkQepiKq0MHYaLBm8mIl0KUTJnc8zMyNzcW7Y1
wKUTJsMDcuWycqmrbNmHswMaNfLxKiptgRgFfsDTi5FHS3sWAwjOKQKLoRT/BUsjM1LpmzBAmDM62bYSrlPr0ESYau
ZqXHi+tlEWCcDYuY3sW3XuvMCVg2TI+TBr16tTM6pV9SshZu0VjqrORztmjDB25OpiepwiFFNdDlHozM6a6CesyKX
StDi5dNBBwphUZY6OBF26mRYD4qSUXoxtWTVxojAljG1Zbkp887ElUvMAyUKbD8w0ihaZ58tTDRGnAJFPxGPnJ5Ni
1yIbLPraBjx8iiiNCOndO/YkY+XR7W1oadyFLkaplQpEwQZ27JKuYlyNRoqouQMUk28UqREZk27kIqKVUksDzcOuWa
tHr8IpSpZVGpNh4TrXyZeUzOBKMlI8kwXFxIunrx6tXCTk2lxF0533ilMCRwb2xh2HS6pRThOGBo
47i41RKuDDxpk3CzUdhVQFs+BQZECWbCJz60EinAFHy6Shk0RpcH0LURynJxOQ3Ch74DOtFyhb5bt3
437MJV+jGWZ8TZoQcsoSTpecC99XSip25/Qe7IbUlDFXkzOZpJ9rFUollVvOA1Mdd1dq1cvLYxMdQnGf62cKQbS4HH

```
MD/nk3YYKysJpcVuewgt1J7WmVYm5YdgCK70vDFHHzVWx20gy/Z1NQ0C9/uFPJXrChuwNU+ALtzvV/9qjSaMRBO/IJ
vI/bv73lDh3r+6tWF/0c53TuOPMSBVih8TwMHV+HblfjSlJB1BKgFOcYB00HhaM/gRTEcpIbq0yd23gIO+WQOjg3uK
w7ZxlG5gerrPf/iixOlixcruemoq/P8f/wj9Bty1unf/c7zfv/70nBmfN0cNW2a5y9YUPg505G/j4PF1GOD1wxeO6j
GxnxT68UgS5nq+Uapo7DxFQXcDzx2S5dW/Hg4HIWRei0XVTnnZnU4HPsJeZcBQFFSDoejNYLVWtlrpN34mVTwQlQOh
6MFAU1N/I2fUZSso8qYr6g7HEWjTUGfF5lprF/vMg2Ho7WBb/AGpYIiRck7HI7WRN71SFlhl7fD4Wh9wJYtcoizcdN
nFCXrcDhaK7B9O8CoUWwGkFh51glxOBz7H7JbFQdq4ZB1nGxIzxRUjR0LmVaoczgcDofD4XA4HA6Hw+FwOBwOh8Phc
DgcDofD4XA4HNVjzpw50NjYiAN9yho1ahQsXrwYbr31VjjjDPglVdecQOBCuSKK66AYaWlJ8uqqamBCy64AJ5//vn
94li/9NJLsHr1anl90FeO1srVV18duljjBeHsI4cLF++3DiunPqUFmuCaTGr5VebV199VWYOnTt3Du3vwoULW9R+O
gpk2bJloZOdRBMmTEDb5tixYwf0LK0BAzfffHPmY6CXJJLqvoiV+asJlkCFYVVEZnFlaSFuqQceeCB3eo4C2Lx5c/m
k6MKnxssvvwx4gxxTWhFeV5vihBNOCP3/H/7wh5mOAVVbrhDE0ffp0WLNmDZxSWmOE1ciRI9HucwYMGGDsW6C8mcXw4
cND6d1999250nMUxHGlRYANYZ1Z2BDXXHNNKMyePXBT0ojYK2xqETQXdSIYeZZb+P+2004xwpBYBtlEIYyh
rZoHHs3379kZ6GzZsyJSeo2A6duxonJxAS5YsQRviwQcfhN69e8PJlkVZNm3aGeeeSYcccQRslGUvpbF94suugjGj
BkDU6ZMgcsvv9yIf9lll8HRRx8NmIGtxLmSTT/gueeeg7lz58rwvvXr1gvr6imE9WysPmAx+vDDDw+FxXx4v8yCO
PlA119JV36KGHoh+rWmUJZjiRxLiMuLZFFGeCGWVatWydPLWATydwLJ4MGD5D5P9CuCf6xcNw5VRXkNqqArN21NaUCt3
sWCHjbNu2TV6reCyx2pjgWLRudj74FHKxSAAWzXNc2lT5Zwn4FI7WHfxQMQGRpSl2fQL9PkOB0M8UWuWxbRSY4rR
YNUPtjr0zZj2/Q3+mM3X5xEYNQWSiVKy6abRl3f0BhBQnwsRgFEkBd8rKgW1VNebCX6q6XFd3S9A0q9QUVHNNcrNsv
eqT79prr6Vv5Mg5//3338el6P21a9eWvrTw5ZdfkscjirXked7XX39NXnawBf3888/3PvjgA/pGjhPB4iR9Ss836nL
8Kfn222/JS4aoOpBn8uGHH3qzZs3CC5Fl48aN8iINRjN+9913aCKpra80lzccO3fuDBQcc2Hbf/U1h
Lq6OvLSoV637du3Jy/MkCFD2tbyhSKTwExBwrW0b9iwwV0+/fj19Mhk0aBB5PJ07dybP87766ivy0tGpUydvh3vuuUdWibCPf82aNTLM2W2+9Ja0oc
kurQ+/4SLjzI0o30triq8eYi3/ssceSZ0eU/Dyx3/TJZMWKFeQ147333iOvGdv/+ivc7ei/iyLhTy7KiZ7Mcff0x
emPHjx5PXtsCLSwqLwvogFhzoIoxNKsb2008/vRxm4SSJxnaSirE9aFCjp25IopQht2H3qfp9IGZAjhEmSfGfisGGs
OsUq2IoqholxUgrqXCcibAedd8Zoh6SAGN78MYw10CKUhs5uR4RrAYKK7E1jAfvDenD41XRuJYQVK01hL1ImIHOmze
v3BNn6/KnMTgBxnamF8MIE7S14QOK2mg4tT0s73zEdh3dcccdaCW2wUbqABZbVxy9MGTtDSFZhxZfcskl0L17d3YbK
ugT5xoEAyVptxAY8XTde++9aGMRRXEjbhKNGDECbRkcbShShMYlEPSJm0DdrByMYAyhQM0dgKSdQ7JBhOfUksbjyMIkl
UgzEnevDpGOHiZOvKbfU88cQTxsGIEuby6tuWUX36KOytoG5WVvQuAGJsC6QMOze2xem1116TGQq3TVVcySDh4LJUU
It6InGt7lhKoBGdkcKn4/XXX4++ATdWg5NavUGo+9qqYKwFwr3voalM3HgY6vkq07dvXyMMJ3UIfwC+JClMKjU1NaF
t2+AYBnXQk64ePXQwQ36xi45KEIaou9SzvUWJLdrBOyBYLOZKHzgGk6q9/DDDxtDvfHpiE8kzJTUCx+7CNXGuqinE
IalInUkXGs9xqXGx0xs2bIlGG1qCIetRwyzLoMjK7kbEp/cNGdGJFi64waM4Xc0FNyASlGs8Bypx/7xxx9nR2fid9x
NPHPmTCMs9j48+eSTRlgESzzceAi8nuhBwRI1vkh/JwfTop4RB4cyoKrFodad9wWVmvwHX5AjNxPqMOq2CJYkyY2FS
gmGpk6dilaSJ2Hw9EKOeecc0IZhCq1kdfhcLRxaEoBq9Qh3w6Ho41jq4YoiqQ8SMnhcLQNsOHy008/9VA4onfAgE
4b0pMXuB5/wdcoujmqVnL3gAAAABJRU5ErkJggg==" alt="Logo"> <!-- Kompor logo -->

        <h2 style="color:black;">Your smart energy saving stove!</h2>
    </header>

    <div class="toggle-container">
        <label class="toggle-label" for="switch">Power Button</label>
        <label class="switch">
            <input type="checkbox" id="switch" autocomplete="off">
            <span class="slider"></span>
        </label>
    </div>

<nav id="buttonlist">
<p1>Select your temperature plan:</p1>
<br><br>
```

```
<div id='programs'></div>
<br><br>
        <a href="#" onclick="fetch('/stop')" id="stopButton">Stop temperature plan</a>
    </nav>
          <div id="countdown"></div>

    <div class="slider-container" id="tempslidediv">
        <div style="display: none;">Step <span id="live-step">0</span></div><br>
        <a       style="display:       none;"       href="#"       onclick="fetch('/next')"
id="nextStepButton">Skip Current Step >>></a>
        <div id="live-temperature">Live Temperature: <span id="counter">21</span>°C </div>
        <div>Live Power: <span id="live-power"></span></div>
        <h2>Adjust Temperature manually </h2>
        <input style="width: 80%" type="range" id="tempslide" min="0" max="100" value="25"
step="1">
        <p id="temperature-value">Temperature set: <span id="tempdisplay">25</span> °C</p>
        <br><br><br>
    <div>
        <h1>Energy Saved Now</h1>
        <h2><span id="energySaved"></span>kWh</h2>
    </div>
        <!-- <p1>Energy saved last week:</p1> -->
<!-- <center><canvas id="myChart" style="width:85%;"></canvas></center> -->
</div>


<!-- <script>
const xValues = ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun'];
const yValues = [2,1,4,6,7,5,3];

new Chart("myChart", {
  type: "line",
  data: {
    labels: xValues,
    datasets: [{
      fill: false,
      lineTension: 0,
      backgroundColor: "rgba(255,0,0,1.0)",
      borderColor: "rgba(255,0,0,0.1)",
      data: yValues
    }]
  },
  options: {
    legend: {display: false},
    scales: {
      yAxes: [{ticks: {min: 0, max:10}}],
    }
  }
});
// Script of the Graph ^
</script>-->

    <script>
        const toggleSwitch = document.getElementById('switch');
        const buttonlist = document.getElementById('buttonlist');
        const temperatureSliderContainer = document.getElementById('tempslidediv');
        const temperatureSlider = document.getElementById('tempslide');
        const temperatureDisplay = document.getElementById('tempdisplay');
```

```
        toggleSwitch.addEventListener('change', () => {
            if (toggleSwitch.checked) {
                buttonlist.style.display = 'block';
                temperatureSliderContainer.style.display = 'block';
            } else {
                buttonlist.style.display = 'none';
                temperatureSliderContainer.style.display = 'none';
            }
        });


        temperatureSlider.addEventListener('input', () => {
            const temperature = temperatureSlider.value;
            temperatureDisplay.textContent = temperature;
            fetch('/manual' + temperatureSlider.value)
        });
// Script to make sure things show up only after switching the button

const buttons = document.querySelectorAll('#buttonlist .buttonlight');


buttons.forEach(button => {
    button.addEventListener('click', () => {
        // Don't do it here, wait for server to tell which one is active.
        /*buttons.forEach(btn => {
            btn.classList.remove('active');
        });


        button.classList.add('active');*/
    });
});


const stopButton = document.getElementById('stopButton');


stopButton.addEventListener('click', () => {

    // Same as other buttons
    /*buttons.forEach(button => {
        button.classList.remove('active');
    });*/
});
// Code to let buttons light up when pressed
// Random number increase to simulate temp increase code

// Start polling requests
function refreshStatus() {
    fetch('/status')
    .then((response) => response.json())
    .then((status) => {
        if (!status.ok) return;
        document.querySelectorAll('.buttonlight').forEach(btn => {
            btn.classList.remove('active');
        });
        if (status.result.status) {

document.querySelectorAll('.buttonlight')[status.result.program].classList.add('active')
```

```
            document.querySelector('#nextStepButton').style.removeProperty('display')
            document.querySelector('#live-
step').parentElement.style.removeProperty('display')
            document.querySelector('#live-step').innerText = (status.result.step_index + 1)
+ ": " + status.result.step[0] + "°C for " + status.result.step[1] + "s"
            document.querySelector('#energySaved').innerText = (status.result.energy_saved
?? 0).toFixed(2)
        } else {
            document.querySelector('#live-step').parentElement.style.display = 'none'
            document.querySelector('#nextStepButton').style.display = 'none'
        }
        document.querySelector('#live-power').innerText = status.result.current_power
        document.querySelector('#counter').innerText = status.result.temperature
        document.querySelector('#tempslide').value = status.result.manual_temperature ?? 21
        document.querySelector('#tempdisplay').innerText                                  =
status.result.manual_temperature ?? 21
    })
}
setInterval(refreshStatus, 2000)
setTimeout(() => {
    fetch('/programs')
    .then((response) => response.json())
    .then((programs) => {
        if (!programs.ok) return;
        programs.result.forEach((program, index) => {
            let newBtn = document.createElement('a')
            newBtn.onclick = () => {
                fetch('/run' + index)
                .then(refreshStatus)

                return false // Prevents default behavior
            }
            newBtn.classList.add('buttonlight')
            newBtn.href = '#'
            newBtn.innerText = program.name
            document.querySelector('#programs').appendChild(newBtn)
        })
    })
}, 300)
</script>
</body>
</html>
```