Hochschule Bremen | Fakultät 4 | SoSe22
Microcontrollers and Applications | Christoph Flores | Plant Care Helper
Laurin Oppermann: 5084078
Janica Bartelt: 5095180

# Documentation

# Plant Care Helper

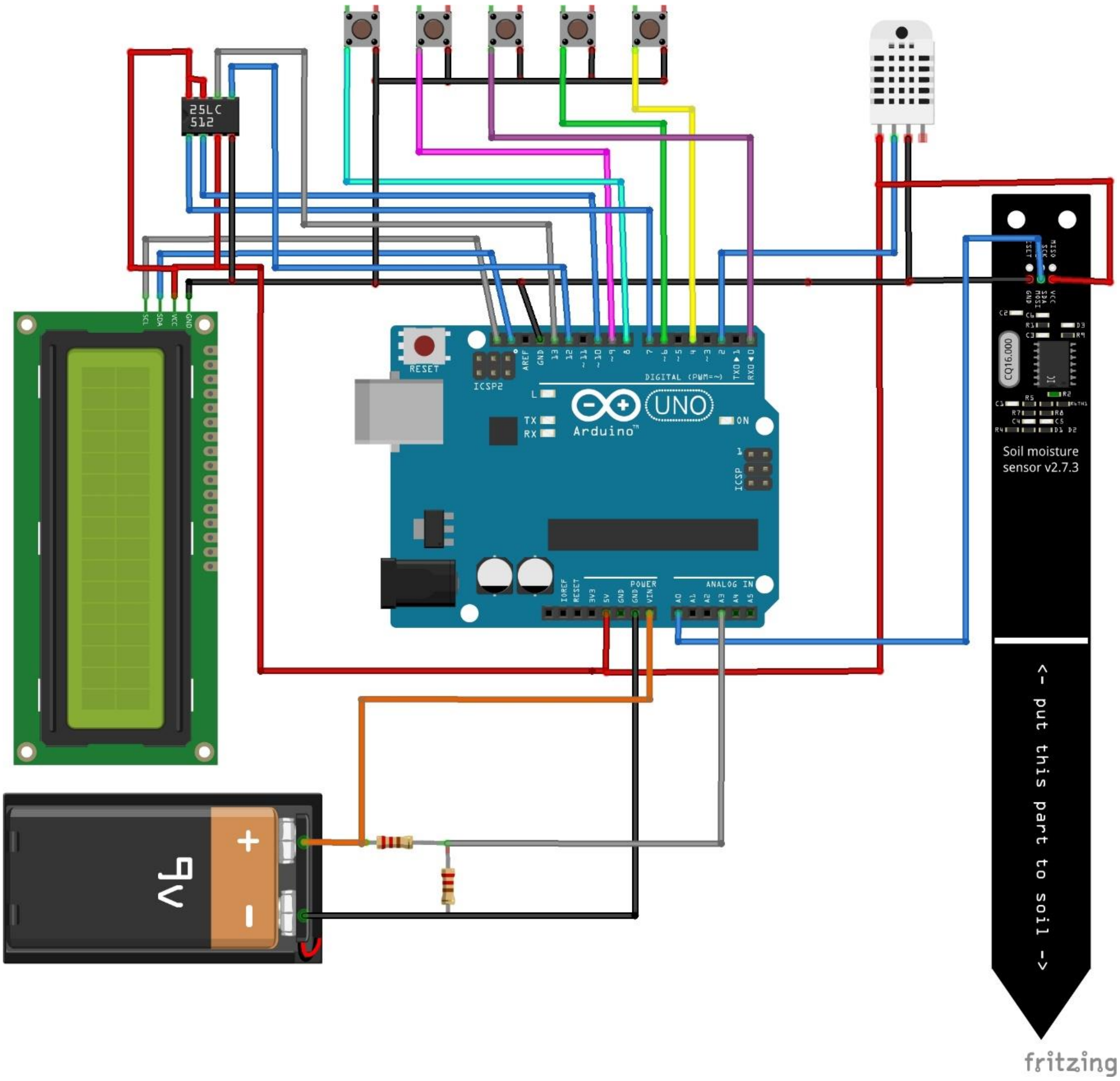
## Inhaltsverzeichnis

## Introduction

Different plant species have different needs, with a large collection of plants it is difficult to find the right watering time for each plant. With a Plant Care Helper you can simplify your watering routine, so fewer plants die from underwatering or overwatering. In addition, controlling the humidity can prevent the leaves from drying out.

## Hardware

The overall wiring diagram was made with the Fritzing software.




To connect the soil moisture sensor, a standard headphone jack was used, because it has three cable cores that can be connected to the pins of the soil moisture sensor.

[1]

Pin connections of the soil moisture sensor:

| | | | |
|---|---|---|---|
| Sensor | GND (black) | VCC (red) | AOUT (yellow) |
| Socket | 1 | 3 | 2 |
| Headphone jack | GND (black) | R (white) | L (yellow) |

# Plant parameters

| | | Soil moisture | | Humidity | |
|---|---|---|---|---|---|
| **Plant typ** | **moisture** | **min** | **max** | **min** | **max** |
| Alocasia | MOI/WET | 4 | 7 | 50% | 60% |
| Anthurien | MOI/WET | 4 | 7 | 50% | 60% |
| Begonia | MOI/WET | 4 | 7 | 50% | 60% |
| Cactus | DRY | 2 | 4 | 35% | 60% |
| Calathea | MOI/WET | 4 | 7 | 50% | 60% |
| Colocasia | MOI/WET | 4 | 7 | 50% | 60% |
| Fern | WET | 5 | 8 | 60% | 80% |
| Hoya | MOI/DRY | 3 | 5 | 50% | 60% |
| Jewel Orchid | MOI/WET | 4 | 7 | 70% | 80% |
| Micans | MOI/WET | 4 | 7 | 50% | 60% |
| Monstera | MOI/WET | 4 | 7 | 50% | 60% |
| Peperomia | DRY | 2 | 5 | 35% | 60% |
| Philodendron | MOI/WET | 4 | 7 | 50% | 60% |
| Pothos | MOI/WET | 4 | 8 | 50% | 60% |
| Sansevieria | DRY | 3 | 4 | 35% | 60% |
| Spider Plant | MOI/WET | 4 | 7 | 50% | 60% |
| String of … | MOI/DRY | 3 | 5 | 50% | 60% |
| Syngonium | MOI/WET | 4 | 7 | 50% | 60% |
| ZZ | DRY | 2 | 4 | 35% | 60% |

Most of the plant parameters are from sources [2] and [3].

I classify the missing parameters from my experiences with my own plants. Some plants have the same needs, so I was able to estimate the values of these plant. The scale was also helpful to classify missing plants. The soil moisture scale is taken from the usual scales for measuring moisture for plants.

**Humidity in %**

| 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|
| DRY | | | | | MOI | | | | |

**Soil moisture**

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|
| DRY | | | MOI | | | | WET | | |

# Programming

The data from the sensors gets sampled using an Interrupt Service Routine. This interrupt is configured to happen every 100ms (10Hz). Because the air sensor has a sampling rate of 1Hz, a counter is used to only sample the air sensor every 10 interrupts (1Hz).

## Air sensor

The air sensor is a DHT11. It has a GND, 5V and data pin. The data pin is used as a one-wire-bus to notify the sensor and then receive its data.

The SimpleDHT-Library already implements this communication and breaks it down into a method to read the humidity (%) and temperature (°C) as bytes.

### Soil sensor

The soil sensor has a GND, 5V and AOUT pin. It outputs its value as an analog voltage between 0V and 5V, which can be read using the Arduinos ADC-pins.

This analog reading is then converted to a percentage assuming a linear function between the configured minimum and maximum voltage readings.

To detect whether a soil sensor is currently connected, the sample-and-hold mechanism of the Arduinos ADC is taken advantage of:

First, an analog pin assigned to 0V is read and a first sample from the soil sensor is taken. Then, an analog pin assigned to 5V is read and another sample rom the soil sensor is taken. If a soil sensor is connected, it’s able to discharge/charge the ADCs capacitor before reading the soil sensors sample, making both samples (ideally) the same. If no soil sensor is connected, the soil sensor samples will (ideally) match the readings from 0V and 5V.

Analog reads return a value between 0 and 1023. In practice the difference between the two samples seems to be at most 2 if a soil sensor is connected. In contrast the difference is around 700 if no soil sensor is connected, making it easy to detect a soil sensor.

### LCD

The LCD is a standard HD44780-compatible display with an I2C-Controller. The LiquidCrystal_I2C-Library is used to control it.

### EEPROM

To store the plant presets an external serial EEPROM is used. Reading and writing was manually implemented using its datasheet.

### Battery

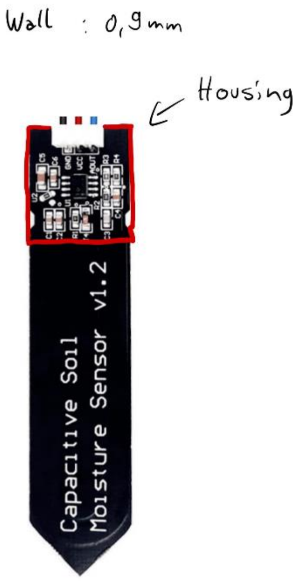
To read the battery level, two big resistors were used to create a voltage divider that doesn’t produce high currents. This voltage is then read through an analog pin and converted to a percentage assuming a linear function between the configured minimum and maximum voltage. This percentage gets displayed (as a kind of bar graph) when starting the PlantCareHelper.

## Design

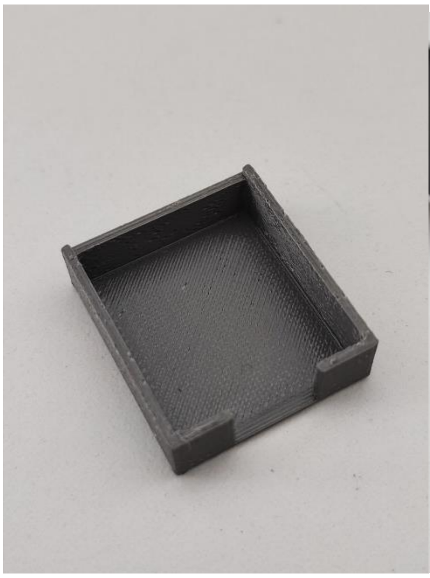
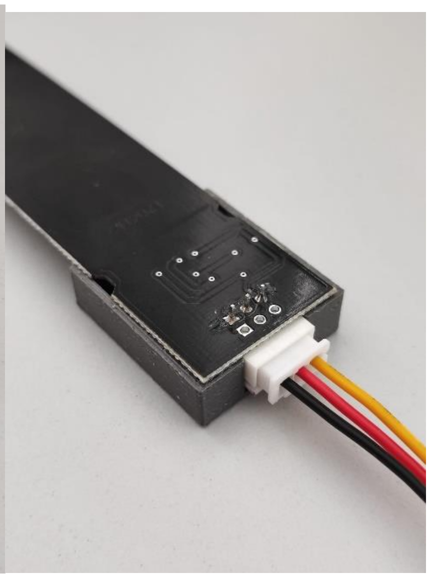
### Sensor Housing

We wanted to produce the housing for the sensor wand with a 3D printer. With 3D printed parts it is easy to produce many of the sensor wand housings.

Our first idea was, to use the white plug of the sensor to connect the sensor wand to the Arduino box. The Picture below shows what we first imagined for the wand housing.




After our first sample prints, we realized that we should use a multiple of the 3D printer nozzle, so there will be no complications with that. These are pictures of our first test prints. One is just the housing and the other is with the sensor in it. The sensor already fitted in our design.

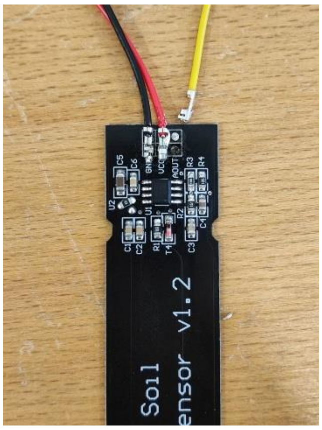



But shortly we noticed that the white plug is difficult to use. We also considered to make the sensor wand waterproof, but we decided that it is not necessary that the wand have to be 100% waterproof. Maybe later we want to fill the wand with glue, like in modelling.

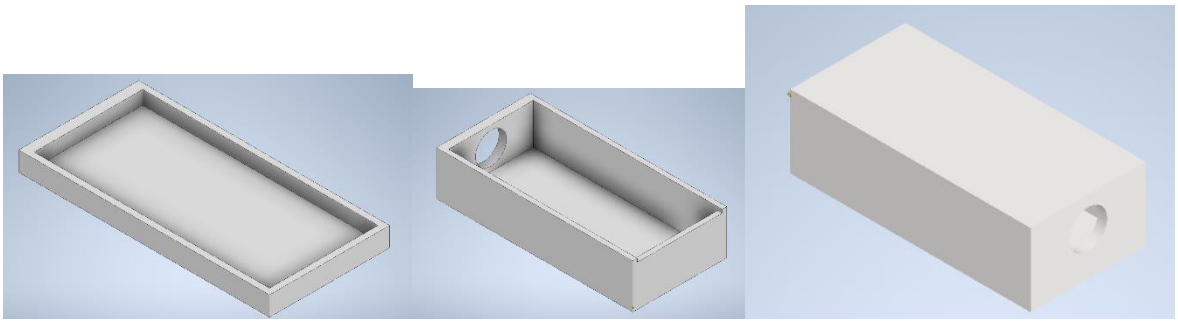
For these reasons we  chose a jack plug instead.

The design had to be adjusted due to the added socket of the jack plug. The housing has become longer and a hole for the socket was added.

First, we wanted to desolder the white plug completely. Due to difficulties with soldering the plug off and the cable on, we wanted to leave the plug on the sensor.




The final design of the housing looks like the fowling picture.



The back of the housing is extended, so that the pins on the back of the senor can fit in. The hole at the front of the housing is for the jack plug.

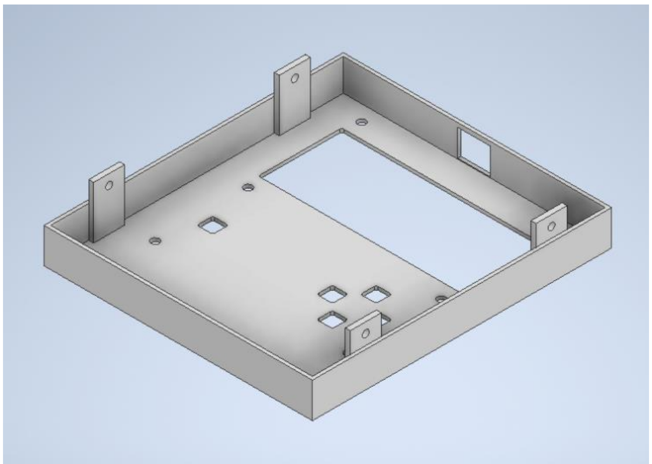
## Arduino housing

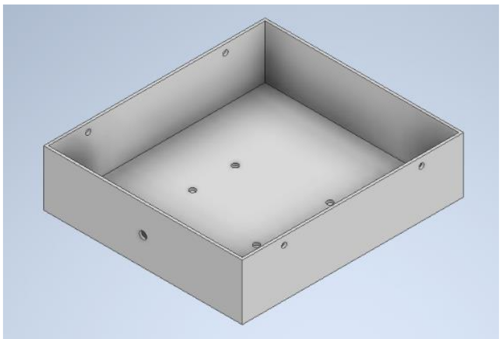
There were no complications with the Arduino housing design. We laid out all the components and measured the sizes for the housing.

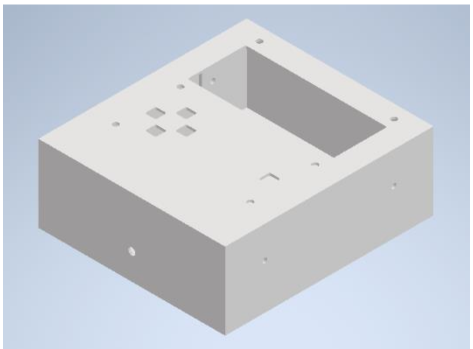
The lid has cut-outs for the humidity sensor, the buttons and the display. The board for the buttons and the display are screwed to the housing, for this reason the six holes are there.

The tabs on the side match to the holes on the side of the bottom housing. This was added to allow that the housing can be opened and closed any time.

At the bottom of the base are holes to fix the Arduino. There is another hole on the side, through which the data cable runs.



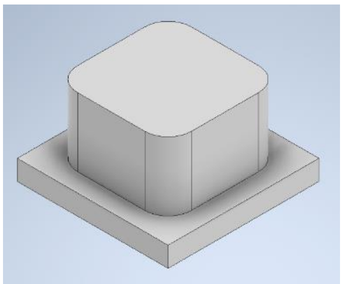




The Battery fixture is getting glued to the bottom part of the housing.

## Buttons

We found the existing buttons to be too small to press, that´s why we decided to design attachments for the buttons.



There is enough tolerance between the housing and the buttons, so that they can be pressed easily.

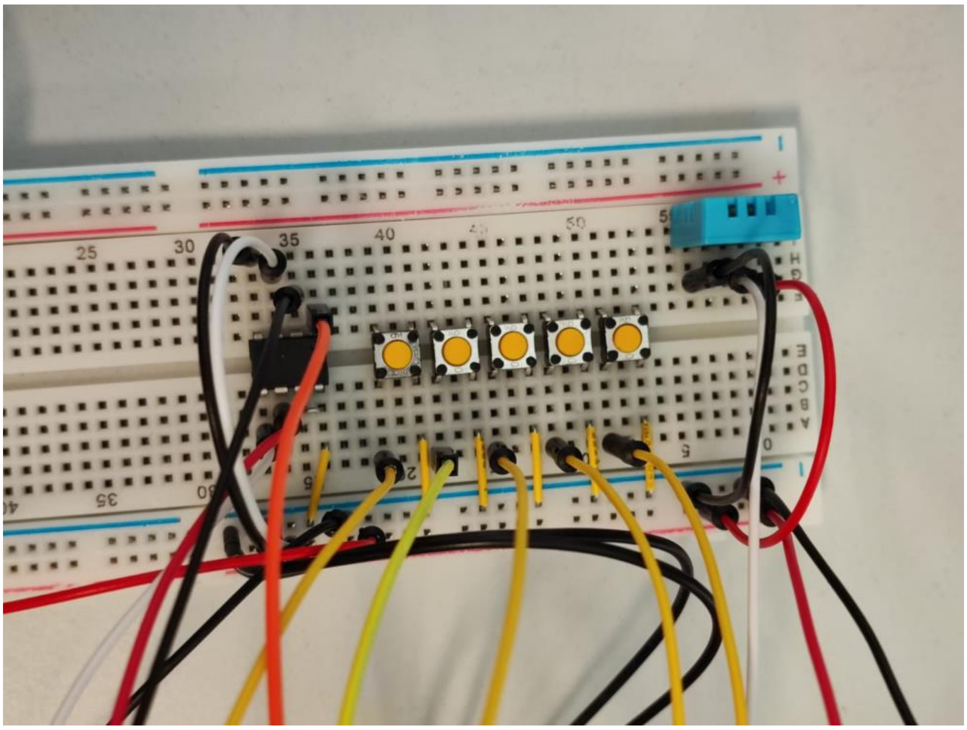
## Testing

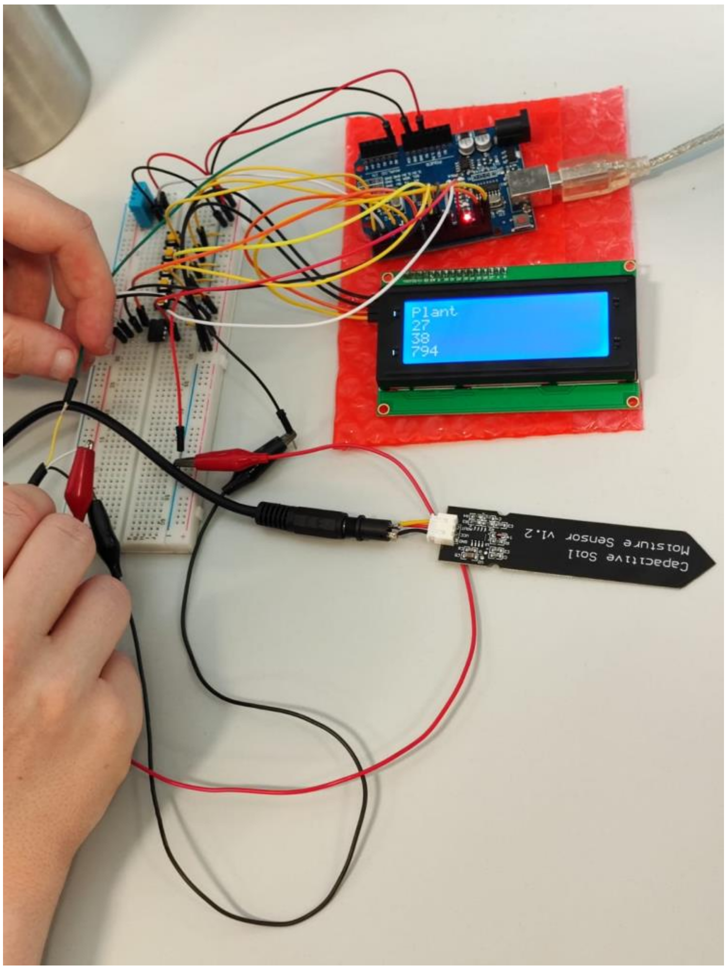
These are the test cases with which we tested our setup.

- Selection of plant preset
- Test without sensor wand → doesn‘t display soil moisture
- Test without sensor wand in soil → soil moisture should be very low
- Test in dry soil → soil moisture should be low and
- Test in wet soil → soil moisture should be high
- Test in water → soil moisture should be very high and should display warning
- Test humidity sensor by blowing air onto it → humidity should increase
- Test temperature sensor by touching it with a finger → temperature should increase

Test Setup:

This is our test setup for testing the hardware components. First, we started testing with a breadboard and ...






## Usage Scenario

The user works his way up from plant to plant with the Plant Care Helper to get an overview of the conditions of the individual plant. For each plant, the user has to plug the cable into the sensor wand in order to be able to check the required values. The user has to adjust the plant type by cycling through the different plants with a button, before checking each plant. The Plant Care Helper has a LCD display, on this the user can see the soil moisture and the humidity. The soil moisture value can be used to determine whether this plant should be watered, has enough water or too much. The humidity sensor is mounted on the display housing.

## Conditions

Red = not fulfilled

Orange = partially fulfilled

Green = fulfilled

| **Function** | **Must / Can** |
|---|---|
| Display shows soil moisture as absolute value (g/cm³) or percentage | Must |
| Display shows humidity as absolute value (g/cm³) or percentage | Must |
| Buttons to navigate the menu (up, down, ok) | Must |

| | |
|---|---|
| Display shows when plant should be watered / not watered depending on selected values | Must |
| Display shows warning when plant is too wet | Must |
| Plant selection with pre-defined values | Must |
| Manual parameter selection | Can |
| Easy handling with the settings of the Plant Care Helper | Must |
| Easy expandability of sensor wand | Must |
| Easy duplicability of the Plant Care Helper | Can |
| Construction, production and testing of housing for microcontroller and display | Can |
| Construction, production and testing of housing for the sensor wand | Must |
| Plug connection via breadboard | Must |
| Contacts soldered | Can |
| Powered by battery | Must |
| Battery level shown on display in percent | Must |
| Making the sensor housing water tight, if even necessary | Can |
| Automatic idle off the microcontroller (no on/off switch) | Can |
| Display temperature | Must |

# Reflection

## Successes

The integration of the sensors worked especially well

Designing the housing also worked better than expected. We had to make some adjustments but most of that was due to the plug issue of the sensor wand.

## Non-successes

We had some issues with the plug of the sensor wand. Through the exchange with a jack plug the problem could be solved. But this took a lot of time away from testing and programing.

We thought long and hard about getting the sensor wand waterproof.  In the end we decided that it would take too much effort to get the wand 100% waterproof and it would be unnecessary. The sensor is only used indoors and if the user is careful when watering it shouldn´t do anything bad. But replacing the plug and filling the contacts with glue helped to make it a bit waterproof.

# Tasks

We completed most of the tasks for the Plant Care Helper together as a team, but we have divided some subtasks according to our strengths.

## Subtasks

### Janica Bartelt

The main task was to design the housing for the soil moisture sensors as well as the housing for the microcontroller. As well as creating the housing prototypes to test the design.

Another important task was to determine the plat parameters.

### Laurin Oppermann

Implemented the microcontroller software, including display output, sensor input and button interactions.

# Components and Libraries

| **Component** | **Description** | **Driver Library** |
|---|---|---|
| Microcontroller | Board ATmega328 with USB cable | Arduino Library (https://github.com/arduino/ArduinoCore-API) |
| Soil moisture sensor | Bodenfeuchtesensor Hygrometer Modul V1.2 kapazitiv kompatibel mit Arduino und Raspberry Pi | |
| Humidity sensor | DHT11 Temperatursensor und Luftfeuchtigkeitssensor kompatibel mit Arduino und Raspberry Pi | SimpleDHT DHT11 Sensor Library (https://github.com/winlinvip/SimpleDHT) |
| LCD Display | HD44780 2004 LCD Display | LiquidCrystal I2C (https://www.arduino.cc/reference/en/libraries/liquidcrystal-i2c/) |
| Button | Preexisting short-stroke buttons | |

The Arduino_AVRSTL-Library (https://www.arduino.cc/reference/en/libraries/arduino_avrstl/) was also used to be able to work with C++-strings on the Arduino.

# Sources

[1]https://de.wikipedia.org/wiki/Klinkenstecker

[2]https://www.globus-baumarkt.de/info/rat-tat/ratgeber/urban-jungle/

[3]https://feey.ch/blogs/pflanzen-blog/temperatur-luftfeuchtigkeit-erhoehen-fuer-pflanzen