Automated Plant Holder for Compact Area

Automatiserad växtbehållare för kompakta ytor

MATILDA THORBURN

ISAK FORSBERG
Automated Plant Holder for Compact Area

ISAK FORSBERG
MATILDA THORBURN

Bachelor’s Thesis at ITM
Supervisor: Nihad Subasic
Examiner: Nihad Subasic

TRITA-ITM-EX 2018:55
Abstract

To facilitate home gardening this project enables automated hydroponic gardening on a small scale. Hydroponic means that the growth medium is water instead of soil, which simplifies the measuring and control of the plant holder. Hydroponics is water efficient and therefore more environmentally friendly.

The objective of this thesis is to develop a minimized construction capable of fitting in a compact home, requiring minimal effort from the user to sustain the plant. The project focuses on three determining factors for hydroponics; light, pH level, and nutrition. By measuring the light intensity, the lamp can be switched on only when necessary which minimizes energy consumption.

The system is controlled by a micro controller and with multiple sensors the pH level, nutrient content, water temperature, and light intensity of the system can be measured. These are then used to regulate the actuators of the system: lamp, dosage pump for pH buffer, and nutrient solution.

Results show that the system is stable and can be self-regulated in the desired range for nutrition concentration and pH level during a substantial time period.
Referat

Automatiserad växtbehållare för kompakta ytor

För att förenkla växtodling i hemmet har detta projekt gjort det möjligt att odla hydroponiskt automatiskt på liten skala. Hydrokultur innebär att odling sker i vatten istället för jord, vilket förenklar mätningar och styrning av växtbehållaren. Hydroponisk odling är vatteneffektivt och därmed mer miljövänligt.


Resultatet visar att systemet är stabilt och kan självregleras inom de önskade intervallen för pH och näringskoncentration under en längre tidsperiod.
Acknowledgements

We would first like to thank our thesis advisor Nihad Subasic for supporting us during the project. We would also like to thank Staffan Qvarnström for his guidance and help with components. Lastly we would like to thank Anna Zetterlund for lending us a photometer and making the LDR readings possible.

Matilda Thorburn and Isak Forsberg
Stockholm, May 2018
List of Abbreviations

AC - Alternating current.
DC - Direct current.
DLI - Daily light integral.
EC - Electrical conductivity.
LDR - Light dependant resistor.
LED - Light-emitting diode.
Lm - Lumen, the total quantity of visible light emitted by a source.
Lux - The total quantity of visible light emitted by a source per m².
PAR - Photosynthetically active radiation.
ppm - parts per million.
S - Siemens, unit of electric conductance.
## Contents

### List of Figures

### List of Tables

<table>
<thead>
<tr>
<th>1 Introduction</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Background</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Purpose</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Scope</td>
<td>2</td>
</tr>
<tr>
<td>1.4 Method</td>
<td>2</td>
</tr>
<tr>
<td>1.4.1 Light Control</td>
<td>3</td>
</tr>
<tr>
<td>1.4.2 Nutrition and pH Control</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2 Theory</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Hydroponic System</td>
<td>5</td>
</tr>
<tr>
<td>2.2 pH level</td>
<td>6</td>
</tr>
<tr>
<td>2.3 Nutrition and Conductivity</td>
<td>6</td>
</tr>
<tr>
<td>2.4 Light</td>
<td>7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3 Demonstrator</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Hardware</td>
<td>9</td>
</tr>
<tr>
<td>3.1.1 Light Sensor</td>
<td>10</td>
</tr>
<tr>
<td>3.1.2 Water Tank, Nutrition and pH</td>
<td>11</td>
</tr>
<tr>
<td>3.1.3 Measuring EC Level</td>
<td>11</td>
</tr>
<tr>
<td>3.1.4 Measuring pH level</td>
<td>12</td>
</tr>
<tr>
<td>3.1.5 SD Card Reader</td>
<td>13</td>
</tr>
<tr>
<td>3.1.6 Microcontroller</td>
<td>13</td>
</tr>
<tr>
<td>3.2 Software</td>
<td>13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4 Results</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Stability of EC and pH Levels</td>
<td>15</td>
</tr>
<tr>
<td>4.2 Light Intensity and DLI</td>
<td>15</td>
</tr>
</tbody>
</table>
List of Figures

2.1 Diagram of an Ebb and Flow hydroponic system made in Google Drawings. 6
3.1 Diagram of the hardware structure drawn in draw.io Diagrams. . . . . . 9
3.2 Correlations of resistance and lux, drawn in Matlab R2017b. . . . . . 10
3.3 Diagram shows test of pH meter, drawn in Matlab R2017b. . . . . . 13
3.4 Diagram of the software structure drawn in draw.io Diagrams. . . . . 14
4.1 EC and pH readings during longterm test, drawn in Matlab R2017b. . 16
4.2 Light intensity during longterm test, drawn in Matlab R2017b. . . . 16
A.1 Photo of the final hydroponic system with the electronics in the container to the right. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25
A.2 Photo of the resistance of the LDR measured with a light meter in comparison with a photometer. . . . . . . . . . . . . . . . . . . . . . 26
A.3 Photo of pH-buffer and nutrition with peristaltic pumps adding the solutions to the water tank. . . . . . . . . . . . . . . . . . . . . . 26
A.4 Photo of growth tray with pH-probe and containers of pH-buffer and nutrition. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27
A.5 Photo of the water tank with waterpump, temperature sensor and EC-meter. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27
A.6 Circuit of EC meter and temperature sensor drawn in Fritzing. . . . . 28
A.7 pH meter used in water tank. . . . . . . . . . . . . . . . . . . . . . . . 28
A.8 Temperature of water tank, drawn in Matlab R2017b. . . . . . . . . . 29
A.9 Addition of nutrient and pH buffer during the longterm test, drawn in Matlab R2017b. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
List of Tables

3.1 The K value in different concentrations of salinity. 12
Chapter 1

Introduction

This introduction chapter gives a thorough briefing about background, purpose, scope and method of this project.

1.1 Background

Domestic growth of vegetables and herbs is getting more popular among adolescents in Sweden along with a growing interest for organic and local produce. Many people live in cities and have a limited amount of space. This makes it hard to grow plants, especially during the wintertime because of the short days and minimal exposure to sunlight in the north [1]. When growing at home, the plants require a lot of supervision and effort. This is a limiting factor to why domestic growth is not possible for everyone. It also means that the plants cannot be left unsupervised for a longer period of time.

Hydroponic gardening uses water as growth medium instead of soil. This makes it easier to control the moisture for the root system since it is easier to regulate the water level than the moisture of the soil. Commercial farming uses this method to generate fast-growing crops all year around to supply cities with fresh vegetables, but it is also a popular method for interested home-growers [2]. When using water as a growth medium it is also easier to control and regulate the amount of nutrition given to the plant. In addition to nutrition, the pH level is of importance, both for the plants and for microbe control of the hydroponic system. By measuring and correcting for each variable individually, a system can be created that minimizes the labour required to sustain the system [3]. The hydroponic closed system is water efficient and minimal water loss only occurs due to evaporation [4].

Bachelor’s theses in this subject area have been done before. In 2017 there was a project called "Automated hydroponics greenhouse" which has been used as inspiration for this project [5].
1.2 Purpose

The purpose of this project is to make it easier to grow vegetables and herbs at home. This will be achieved by minimizing the effort required by the user in order to sustain the plant. This project will hopefully make more people interested in having their own garden at home. By having a micro controller that recognizes and regulates the needs of the plant, the user can leave the plants for longer periods of time. The thesis must also investigate and answer the following research questions:

1. In what range can the system be automated to minimize the effort from the user?

2. What are the limiting factors for this system and is the system reliable?

1.3 Scope

The time is limited to 15 credits during one semester and to the resources given by the department of Mechatronics at the Royal Institute of Technology, Stockholm.

The project is limited to making a system intended for household use, it is supposed to fit in a small apartment and therefore a plant stand, BITTERGURKA from IKEA, has been chosen as frame for the product [6]. It has a built-in LED-lamp and the container can fit four pots of size 10.5 cm in diameter. The system is limited to this stand and in addition a water tank, a nutrition solution, and a pH buffer will be added. The volume of the water tank is a limiting factor for the time the system can be self-sufficient.

This project will focus on growing the herb basil. Because it is a forgiving plant, grows fast, and requires a lot of sunlight. This will make it easy to see the difference if the plants are given enough light or not. When growing in a hydroponic system the environmental factors are the temperature of the water tank, pH level, salinity of the solution, and the light intensity of the surroundings. For an automated system, all these factors are measured and recorded but within the size of this project there is no capacity to develop a system that can control all of them. The factors that this project will focus on are the light intensity, conductivity and pH level of the solution.

1.4 Method

In order to answer the research questions, an automated plant holding system is built with hydroponic as the growth method. The absorbed amount of water is proportional to the concentration of nutrients in the solution. How much the plants absorb depends on environmental factors such as temperature, pH level of the solution, salinity, level of oxygen and electrical conductivity of the solution, humidity, and light intensity of the surrounding environment [7].
1.4. METHOD

1.4.1 Light Control

A light sensor is used to continuously measure the amount of light the plants are given from natural lighting. The total quantity of visible light emitted by a source is measured in the SI-unit lumen (lm) and the light sensor measures the amount of lux, i.e. the amount of one lm/m². To compensate for insufficient light the LED is turned on to ensure that the correct amount of light is given to the plants. By regulating the light, the system can keep the electrical light source off when not needed and thus conserve energy.

1.4.2 Nutrition and pH Control

The water itself does not contain any nutrition, therefore a nutrient solution must be added to the water to maintain growth of the plants. The nutrition is controlled by measuring the electrical conductivity of the water. This shows the concentration of salt ions present in the solution and is a simple way of determining the concentration of nutrients present in the solution [8]. The desired electrical conductivity, EC, range differs for different plants and for basil this range is 1.6-2.2 S [9]. When using hydroponics the system is more sensitive to changes in pH level. The pH level ensures that the right amount of nutrients are absorbed by the plants [3]. A close to neutral pH level and 5.5-6.5, is desired for most plants.
Chapter 2

Theory

This chapter accounts for the theory of hydroponic system and the components required to make the system successful.

2.1 Hydroponic System

The primary goal of hydroponics is to remove as many barriers as possible between the roots and the nutrients the plants need. Hydroponic gardening is possible in a few different ways, but the main technique is the same. All plants require water, air, light, mineral salts, and support for the roots. The differences from growing with soil is that the nutrients are supplied directly to the plants through water. By removing the soil, many time-consuming and unreliable aspects of gardening are eliminated. This facilitates clear and assertive results in the project. There are a handful of different methods of implementing a hydroponic system that all share the same principles; e.g. aeroponics, deep water culture, drip systems, Ebb and Flow and nutrient film technique [3]. Most of them have a water solution with nutrients that is supplied to the root system. How the water is supplied differs between the systems. In some, the roots are constantly immersed in water, in others the water is pumped to the roots. Some of the systems drain the excess water and others let the root support medium continuously absorb enough water for the roots. All of the systems have both advantages and disadvantages and are designed for different types of plants. The system used in this project is called an Ebb and Flow system and is based on the natural occurrence of ebb and flow and is displayed in Figure 2.1. It is a closed system where the growing area is flooded recurrently with a water/nutrient-solution by an electrical pump. Before the solution is pumped to the growing area it is tested and regulated accordingly for pH level and EC. A drainage pipe controls the water level in the growth tray and excess water is returned to the water tank. This is repeated hourly to ensure that the plants are supplied with enough nutrients and to circulate and oxygenate the water.
2.2 pH level

The pH scale is used to specify if a solution is acidic or an alkaline. pH is defined as the negative logarithm of the correlative hydrogen ion activity \((a_{H^+})\) in a solution [10].

\[
pH = -\log_{10}(a_{H^+}) = \log_{10}\left(\frac{1}{a_{H^+}}\right)
\] (2.1)

The pH scale ranges from 0 to 14 where 0 is strongly acid and 14 is strongly alkaline. The neutral pH value is 7 which is obtained when the concentration of hydrogen ions and of sodium hydroxide are equal. The pH scale is logarithmic which means that one step increase in pH corresponds to a factor ten in hydrogen ion activity. When regulating the pH level, it has to be done cautiously due to the nonlinear scale making it difficult to regulate without continuous feedback. Most plants thrive at a pH level between 5.5 and 6.5 [11].

2.3 Nutrition and Conductivity

All plants require a wide range of nutrients, both macronutrients and micronutrients, to ensure growth. All nutrients need to be maintained at a sustainable level to ensure the stable growth and well-being of the plants. If the nutrient level is too high the risk of algae and bacteria growth increases and when nutrients levels are low, the risk of deseases increases. Nutrients needed by the plants differ, but most plants thrive in a similar amount of nutrients. This thesis does not focus on optimal nutrient composition and therefore a general nutrient solution was used to ensure sufficient growth and pH control.

The easiest way to measure nutrition level in a hydroponic system is to track the EC level. EC is a way of determining salt concentration, i.e. salinity, of a
solution and is measured in Siemens. It is defined as a material’s ability to conduct electrical charges. A higher EC of a solution, means a higher salinity. The EC level has a linear correlation with temperature, making it easy to take into consideration when measuring. The relation is around 2% increase in EC per 1 degree increase [12]. The EC level is a good way to measure the salinity of the solution but it does not indicate the distribution of nutrients within the solution. Nutrient solutions in hydroponic systems should range from 1.6-2.2 S depending on the stage of growth of the plants[9].

2.4 Light

In order to power the process of photosynthesis the plants need energy from light, preferably sunlight. When plants are not given enough sunlight, artificial light is needed in order to maintain optimal growth. Different plants desire different intensity of light and different types of light, but they all use light in the visible range, foremost in the red and blue wavelength [13]. The intensity and amount of light required to fuel photosynthesis is defined as Photosynthetically Active Radiation, PAR. The amount of PAR occurring naturally depends on weather, time of day and season and is measured in micromoles per square meter per second, \( \mu \text{mol m}^{-2} \text{s}^{-1} \) [13]. In commercial growth the PAR is measured during the day and with the addition of artificial light, they certify that enough light reaches the plants [14]. The amount of light the plants need during a day can be calculated as the Daily light integral and should be above 10 for basil to grow properly [15].

\[
DLI = \sum_{i=1}^{n} PAR_i \cdot \Delta t \cdot \frac{3600}{1000000} \mu \text{mol m}^{-2} \text{s}^{-1}
\] (2.2)

In addition to PAR, light can also be measured in lumen, lux and watt. Lux is one lumen per square meter and is the intensity of light reaching a surface. During a sunny day the leaves are hit with about 100 000 lux, but if the plants are placed behind a window this value is a quarter of that, about 25 000 lux. To aid the growth of the plants they need a minimum of 1 600 lux [16]. Watt is the heat energy that a light source consumes. A light with high energy can burn the plants and is also high in energy consumption. Therefore it is preferable to use LED as a light source because of its low value of watt and high value of lumen. The LED used in IKEA’s BITTERGURKA pot of 310 lux contains more red and blue light than a regular lighting which helps the photosynthesis. The height placement of the LED makes sure that the plants are at a 30 cm distance from the lamp so they do not get burned. The LED consumes less energy and emits less heat than a regular lightbulb. There are sensors that measures the PAR value directly but those are expensive and intended for commercial use. In this project a sensor which measures light intensity is used. With the help of a conversion table, the measured light intensity is converted to PAR.
Chapter 3

Demonstrator

This chapter will focus on the construction of the automated plant holding system. The system consists of sensors that send signals to a micro controller and actuators which react to those signals, the structure of the system is showed in figure 3.1. The system is based on the plant container BITTERGURKA as the growth tray with the lamp included. The water tank is constructed of a hanging planter, from the same IKEA series as the plant container. Images of the final system are displayed in Appendix A.

![Diagram of the hardware structure](image)

**Figure 3.1**: Diagram of the hardware structure drawn in draw.io Diagrams.

### 3.1 Hardware

The automatic plant holder is controlled by a micro controller, specifically the Arduino Uno. In order to regulate the system, the Arduino relies on multiple sensors,
i.e. LDR, EC and pH meter, temperature sensor. The readings of those are used to regulate the actuators of the system accordingly, i.e lamp, dosage pump for pH buffer, and nutrient solution.

3.1.1 Light Sensor

The LDR that is used, measures the light intensity per square meter, in lux. To convert from lux to PAR, a calibration factor of 0.0185 is used when the lamp is turned off and 0.0135 when the light is turned on [17]. To establish a relation between the resistance of the LDR and the light intensity, measurements with a photometer were made. With a logarithmic scale, a linear equation with coefficients was calculated as Eq. 3.1 [18], displayed in Figure 3.2a and Figure 3.2b. Equation 3.1 can be rewritten as Eq. 3.2 where the coefficients A and B were calculated with MATLAB [19].

\[
\log(Lux) = A \cdot \log(R) + B \tag{3.1}
\]

\[
Lux = R^A \cdot 10^B = R^{-1.4256} \cdot 1.8357 \cdot 10^6 \tag{3.2}
\]

With the conversion factor from Eq. 3.2, a lux value is recorded and stored to a memory card every hour. At the end of the day these values are evaluated and used for the DLI. If the DLI does not reach the required level of 10, the missing time is calculated and a relay will turn the LED on to ensure that the correct DLI is achieved.

![Diagram of the logarithmic correlation of lux and resistance of the LDR.](image)

![Diagram of the linear correlation of lux and resistance of the LDR.](image)

(a) Diagram of the logarithmic correlation of lux and resistance of the LDR. (b) Diagram of the linear correlation of lux and resistance of the LDR.

Figure 3.2: Correlations of resistance and lux, drawn in Matlab R2017b.
3.1. HARDWARE

3.1.2 Water Tank, Nutrition and pH

A separated water tank is needed to make sure that the roots of the plants are always immersed in water. It is also needed to ensure that the right concentration of nutrients are fed to the plants. The risk of applying nutrients directly into the growth tray is that the plants can be exposed to unhealthily low pH levels or high salinity levels.

The diluted concentrations of nutrient solution and pH lowering solution are added to the external water tank via two small 12V peristaltic dosage pumps. To ensure that the water tank has the right concentration of nutrients that suits the plant, a pH meter and an EC meter are used.

A submersible water pump, that runs on 12V is used to carry the water from the external water tank to the growth tray. The water pump is activated for 5 seconds during which it fills the growth tray. The excess water then flows back to the water tank through a drainage pipe.

3.1.3 Measuring EC Level

The EC in a solution is calculated by measuring the resistance between two probes. A standard Europlug is used as the probe and is fully submerged in the fluid which is measured. An AC is applied to get a closed circuit in the fluid and the resistance is measured. When the resistance is known, the EC can be calculated as the inverse of the resistance. EC is dependant on the surrounding temperature and this needs to be taken into consideration when measuring the EC. When calculating EC, it is common to use a linear approximation for small changes in temperature [20]. This is suitable for this system which is placed indoors where minimal temperature changes occur.

To validate the EC meter’s accuracy, it needs to be calibrated. When calibrating the EC meter, a solution of a known concentration of salinity is used. The calibration determines the cell constant K, i.e. the ratio of the distance between the probe’s two electrodes in cm and the surface area of the two electrodes in cm\(^2\) [21]. Solutions of purified water and various amounts of NaCl were used to establish K. The values are shown in Table 3.1 and vary from 1.17 to 9.375 S. The optimal salinity of the nutrient solution is between 1.6-2.2 S [9]. The EC meter’s reliability is also tested by measuring the recommended concentration of the nutrient solution given by the instructions. A solution with 1.8 ml of the nutrient solution Flora Nova Grow in 1l of fresh water is supposed to correlate to a EC between 1.3 - 1.8 S. Calibration of K gives it a value of 2.8. Since the EC should be around 1.6-2.2 S, K is further calibrated in the nutrient solution and it is determined to 3, following previous experiments [20], [5].
Table 3.1: The K value in different concentrations of salinity.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>750</td>
<td>1.17</td>
<td>4.12</td>
</tr>
<tr>
<td>1600</td>
<td>2.5</td>
<td>3.10</td>
</tr>
<tr>
<td>2000</td>
<td>3.125</td>
<td>2.91</td>
</tr>
<tr>
<td>4000</td>
<td>6.25</td>
<td>2.48</td>
</tr>
<tr>
<td>6000</td>
<td>9.375</td>
<td>2.2</td>
</tr>
</tbody>
</table>

3.1.4 Measuring pH level

The pH level is measured with a liquid pH-probe and is powered with 5V [22]. The probe is connected to a Value Detect Module to convert the signal of the probe to an analog value. Calibration of the pH offset is made by regulating a potentiometer, available on the module, to match the voltage to the corresponding value of the solutions of pH 4 and 7. The values gives a conversion factor of 0.176 from voltage to pH level displayed in Eq. 3.3. The module gives a signal in the range 0-1023, equivalent to 0-5V. pH level 7 corresponds to 512 or 2.5V and pH level 4 to 619 or 3V. The value of the probe is then calculated by comparing how much the pH of the probe differs from f pH 7 as shown in Eq. 3.4 [23].

\[
step_{pH} = \frac{V_{pH_7} - V_{pH_4}}{7 - 4}
\]  

(3.3)

\[
pH_{probe} = \frac{7 - (V_{pH_7} - V_{probe})}{step_{pH}}
\]

(3.4)

To get precise readings of the sensor, ten measurements are taken at one minute intervals. The lowest and highest readings are discarded and an average of the remaining eight values is used as the pH level. As Figure 3.3 shows, it takes longer for the meter to stabilize the closer the pH is to 7. In the test, the pH probe was submerged in water and after 88 minutes, pH lowering buffer was added to the system. After an additional 27 minutes more water was added. The spikes in the graph shows the small fluctuations of the pH probe.
3.2. SOFTWARE

Figure 3.3: Diagram shows test of pH meter, drawn in Matlab R2017b.

The pH buffer used in the water tank is a diluted pH lowering solution. Because the probe is very sensitive to electric current and any electrical leakage will lead to disturbance, the pH probe is submersed in the growth tray.

3.1.5 SD Card Reader

All data collected continuously from the sensors, i.e. EC and pH meter and light sensor, as well as the addition of nutrients and pH buffer to the water tank need to be saved to a memory card and be accessible for the Arduino and the user. This is accomplished with a SD card module for the Arduino. The module runs on 5V directly from the Arduino. The values are stored in textfiles.

3.1.6 Microcontroller

The sensors and actuators are controlled by an Arduino Uno board which is based on the ATmega328P microchip [24]. The board is powered with 9V through a micro-USB and an AC to DC adapter.

3.2 Software

The software is written in C++ and follows the structure as shown in Figure 3.4. The script runs once every hour and begins by measuring the light intensity of the LDR. The value is stored both as lux and converted to PAR depending on the lightsource. After 24 hours, the DLI is evaluated from the SD-card and if it doesn’t reach the desired level, the script calculates the missing time in milliseconds and turns the lamp on for the remaining amount of time. The water pump is then activated during
5 seconds to fill the growth tray. Then the pH level is measured and if the level is above 6.5, pH lowering solution is added. After that, temperature of the water is measured and is used together with the EC probe to determine the EC level. Finally, if the EC level is below 1.5, nutrient solution is added. All the measurements are then stored to the SD-card.

Figure 3.4: Diagram of the software structure drawn in draw.io Diagrams.
Chapter 4

Results

In this chapter, results gathered from a conducted longterm test is presented. To test the stability of the system, a longterm test was initiated for 4 days. The system was placed in a secluded room during the test with the major light source coming from natural light. The test was initiated with 6L of water in the system. The test was conducted with 7 basil seedlings in the plant holder during the test.

4.1 Stability of EC and pH Levels

The initial system contained 6L of water with 8ml of diluted nutrients and 4ml pH lowering solution. The initial EC was 1.20 and pH was 5.87. During the test period both EC and pH level showed stability and were kept in the accepted range.

The EC level started at 1.20, under the minimum value 1.50, and therefore nutrient solution was added frequently for the first 19 hours to reach the right level. When the correct level was reached, EC level was kept stable at 1.50-1.53 S for the remaining time. The addition of nutrient solution to the water tank is shown in Figure A.9a. The EC level is shown in Figure 4.1a. The EC is correlated to the water temperature which remained between 22-24°C, the water temperature is shown in Figure A.8.

The pH level was initially 5.87 and Figure 4.1b shows the pH level changing per hour. The readings fluctuated within the allowed range and the pH exceeded the limit of 6.5 only three times during the test. On the three occasions, pH lowering solution was added, which is shown in Figure A.9b.

The test was only conducted for 118 hours but showed stability and reliability throughout the whole period. Therefore this is a good indication of the system’s longterm reliability.

4.2 Light Intensity and DLI

Measurements of the light intensity were recorded every hour during the test. As Figure 4.2a shows, a clear correlation of the light intensity and time of day can be
CHAPTER 4. RESULTS

(a) Diagram of EC readings per hour.  (b) Diagram of the pH readings per hour.

Figure 4.1: EC and pH readings during longterm test, drawn in Matlab R2017b.

seen. The light intensity of the surrounding light reads at peaks around 1 200 lux and the light intensity with the lamp turned on is approximately 2 700 lux.

During the test period the lamp was turned on for four hours every day. The accumulated PAR of the system is displayed in Figure 4.2b. With the major light source being the natural light, this value was 5.3-6.2 mol m$^{-2}$ s$^{-1}$. The four hours with the lamp on gave an accumulated PAR of 4.7-4.8 mol m$^{-2}$ s$^{-1}$, which represents 44-47% of the total DLI. The distribution of the different light sources contribution to the DLI are shown in Figure 4.2b.

(a) Diagram of the readings of LDR, drawn in Matlab R2017b. (b) Diagram of the accumulated DLI per day and distribution of light source.

Figure 4.2: Light intensity during longterm test, drawn in Matlab R2017b.
Chapter 5

Discussion and Conclusion

This chapter discusses the results and their reliability. It also aims at concluding the project.

5.1 Discussion

In this chapter, the results presented in the previous chapter as well as the light intensity, and the level of automation are discussed.

5.1.1 Light Intensity and LDR

The purpose of an LDR is to detect the presence of light. To configure it, a photometer was used to establish a relationship between light intensity and resistance. The photometer was available whilst readings for calibration were made. The relationship could not be verified in the final system as the photometer was no longer available. No other photocells were tested or calibrated. As Figure 3.2a shows, the values of light intensity above 100 lux is within a small range of the resistor's capacity of 0-20 kΩ. This means that the reading of the LDR might not correspond to the exact actual illumination. Further, when the light is not directly hitting the surface of the LDR, the readings are not exact and there is a risk that the accumulated DLI without the lamp turned on is lower than the actual value.

To ensure that the requirement of light intensity is reached, the accumulated DLI is compared with a desired value. Because the given intensity of the lamp is known, the remaining time can be evaluated. Results show that the lamp is turned on for 4 hours per day on average. Though the script only runs every hour so the missing time might be slightly below the actual time. The accumulated natural light during 20 hours only gives about 5 mol m$^{-2}$s$^{-1}$ according to the readings of the LDR. This means that the lamp needs to be turned on a couple of hours in order to give the plants the right amount of light. During the longterm test, the lamp was turned on for four hours and the DLI requirement of 10 was reached. To aid the growth of the
plants, they need a minimum of 1600 lux and to reach that the system either needs to be placed near a window or the lamp needs to be on.

### 5.1.2 Nutrition and pH

A standard Europlug is used as the EC meter probe and was calibrated by testing the probe in solutions with various concentrations of salinity. To get the right amount of NaCl, a kitchen scale was used to measure 1g of NaCl in 500cl of purified water. The kitchen scale had a precision of minimum 1g, which made it difficult to weigh exactly 1g. With higher precision, the EC calibrations could be made more accurate. After the initial calibration, an additional test was made to control the reliability of the probe in the nutrient solution and is therefore considered to be accurate. The plug has been shown to be reliable and stable as an EC meter in this system, which is supported by previous research [5], [20]. The temperature sensor submerged in the water tank showed stability as well. Because of the correlation of temperature and EC it is crucial that the temperature is measured correctly for the EC meter to give a correct reading.

When looking at how the EC level changes during the longterm test, shown in the diagram 4.1a. The EC level is very stable and that only small amounts of nutrient solution is needed to maintain a good habitat for the plants. When the test was conducted, the plants were only seedling. This could maybe play a roll with how much nutrient is needed, bigger plants need more nutrients. With bigger plants in the system, a bigger variation in the EC level is expected.

To measure pH level, a liquid pH probe is used and calibrated with solutions of pH 4 and 7. The pH probe gives accurate readings when measuring a solution that is isolated from other sensors. When measuring pH level in the system, the readings fluctuate and the probe is sensitive to leakage of electrical current. The pH probe is not made for longterm use and needs to be re-calibrated from time to time, which needs to be taken into consideration by the user. A longterm pH probe is expensive and not affordable within the scope of the project.

In diagram 4.1b the changes in pH level during the longterm test is shown. The pH readings is fluctuating a bit more than the EC, but is kept within the allowed range. Just like the EC level, it could be bigger differences in the pH level if the plants are more fully grown.

### 5.1.3 Level of Automation

The level of automation originates from three areas, the need of light, the need of nutrients and the need of fresh water. These three areas have different levels of automation. While the DLI takes all the readings of the LDR into consideration, the EC and pH level is measured and controlled at the same instance. The control of water level in the growth tray is mechanical. During 5 seconds the water pump is activated and fills the growth tray. A drainage pipe of optimal height ensures that the right amount of water remains in tray. The system is, after the longterm
5.2. CONCLUSION

test, considered to be self-regulating and can grow basil from seedling to fully grown plant. The only handling which is needed by the user is during the setup phase. The system needs fully filled nutrient and pH-lowering containers and needs to be filled up with the right amount of water from the beginning. The liquid containers have no alerting indicators if they run empty and neither does the water tank. A regular checkup is required manually by the user. Because of a desire to make the system suit compact areas the water tank as well as the nutrient and pH buffer containers were made small. However this limits the range of the automation of the system and thus requires maintenance observation checkup.

The system is intentionally made slow to eliminate fluctuations and therefore guarantee that the plants are never exposed to large concentrations of pH lowering solution or nutrient solution that could harm the plants.

5.2 Conclusion

The longterm test that has been conducted shows that the system fulfills its purpose by minimizing the efforts of growing plants and automatically giving the plants a good habitat without requiring maintenance from the user. The system needs to be set up by the user in order to work correctly. Thereafter, the system is automated so that it is self-sustainable over an extended period of time. The conducted test show that the system is stable and reliable for a prolonged time and the user can leave the plants unsupervised.

The limiting factors of this system are the lack of alerts of the liquid level in the nutrient and pH solution as well as the water tank. This means that the system requires a regular observation by the user. Because of a desire to make the system suit compact areas the water tank is made small. The volume of the water tank is also a limiting factor for the time the system can remain self-sufficient.
Chapter 6

Recommendations and Future Work

The objective of this project was to construct an autonomous plant holder and to minimize the effort required by the user. Although the system is stable and tested for longterm use, there are improvements and additions that can be made in order to improve the system as a whole.

The light used in this project is part of the chosen plant holder and therefore included in the construction. It is optimized to give plants light in the right range of wavelengths and intensity. It is fitted for the growth tray but for future work a larger lamp could be considered for higher efficiency. The LDR used to evaluate the light intensity is the only component explored for this purpose in this project. There are several methods to identify the light intensity, i.e. photo transistors, solar cells, photo diodes, however this area was not investigated further in this thesis. Therefore there might be other methods with higher accuracy than the chosen LDR.

The aim of this project is to make the system compact and suited for compact areas. Therefore the size of the water tank is minimized. A smaller storage unit for the electronic components could be adapted to the system to minimize the system further.

The air humidity of the system was not considered in this project since the plants are not encapsulated. The humidity would therefore be the same as that of the room where the system is placed in. To prolong the sustainability of the system and further reduce the effort of the user the system could be enclosed. This would minimize water evaporation occurring and thus prolong the period of autonomy.

Plant species differ in required amount of light, pH level and EC. The system variables are optimized for basil however pre-made programs for other species could be developed to facilitate for the user. A mobile application or wifi connection to a camera could make observations of the system possible.
Bibliography


Appendix A

Images of the Process

Figure A.1: Photo of the final hydroponic system with the electronics in the container to the right.
APPENDIX A. IMAGES OF THE PROCESS

Figure A.2: Photo of the resistance of the LDR measured with a light meter in comparison with a photometer.

Figure A.3: Photo of pH-buffer and nutrition with peristaltic pumps adding the solutions to the water tank.
Figure A.4: Photo of growth tray with pH-probe and containers of pH-buffer and nutrition.

Figure A.5: Photo of the water tank with waterpump, temperature sensor and EC-meter.
APPENDIX A. IMAGES OF THE PROCESS

Figure A.6: Circuit of EC meter and temperature sensor drawn in Fritzing.

Figure A.7: pH meter used in water tank.
Figure A.8: Temperature of water tank, drawn in Matlab R2017b.

Figure A.9: Addition of nutrient and pH buffer during the longterm test, drawn in Matlab R2017b.
Appendix B

Arduino Code

/* Isak och Matilda – Grupp 4 – KEX

* SD card:
* MOSI – pin 11 mörkgrå
* MISO – pin 12 ljusgrå
* SCK – pin 13 lila
* CS – pin 3

* NutrDosing = 9 nutrition
* pHDosing = 10 pH

* pHreadings:
  pHreading A3;
  pHPower – 6

* Waterlevelpump:
  Waterpump – pin 8

* Temperaturesensor:
  Signal – pin 2

* EC-meter:
  Signal – pin A1
  ECpower – pin A2

* LDR S – pin A0
* Relay to light:
  lamp = 4

// include libraries

#include <Arduino.h>
#include <SPI.h> // SDkort
#include <SD.h> // sdkort
#include <stdio.h>
#include <stdlib.h>  // atof
#include <OneWire.h>
#include <Wire.h>
#include <DallasTemperature.h>

// Lightmeasuring variables:
int LDR_PIN = A0;
bool lightOn = false;
float lightfactor = 0.0135;
float ldrLux;

// float PARvalue;
float PPFValue;

// Logging to SD-card variables, declare filename
File LightData;
File PARData;
File AddedData;

// Light Control
unsigned long lightinterval = 8.64 * pow(10, 7); // 24 hours in milliseconds

unsigned long previousMillis = 0;
float PARsam;
float totalPAR;
float missingPAR;
int newValues = 24;

// Lamp Control
unsigned long lampinterval; //= 1.08 * pow(10, 7); // 3 hours
int lamp = 4;

// Temperature
// Data wire is plugged into pin 2 on the Arduino
#define ONE_WIRE_BUS 2
// Setup a oneWire instance to communicate with any
OneWire oneWire(O ONE_WIRE_BUS);
// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);
int temp;

// EC meter
int ECPin = A1;
int ECPower = A2;
float EC25 = 0;
float Temperature = 10;
int R1 = 1000; // Resistance used for the plug
```c
int Ra = 25; //Resistance of powering Pins
int ppm = 0;
float Rc = 0;

//pH readings:
unsigned long int avgValue;
float phValue;
int pHreading = A3;
int pHPower = 6;

int NutrDosing = A4; //dosage pump for Nutrition
int pHDosing = A5; // dosage pump for pH buffer

// Reading Water Level
int WaterPump = 8; // Output to relay to control waterpump

// MAIN
int count = 1;

/******************** SETUP *****************************/
void setup() {
    // LDR
    pinMode(LDR_PIN, INPUT); // lightsensor
    pinMode(WaterPump, OUTPUT); // Waterpump powering switch
    pinMode(lamp, OUTPUT); // Relay to lamp
    pinMode(NutrDosing, OUTPUT);
    pinMode(ECPin, INPUT); // EC readings
    pinMode(ECPower, OUTPUT); // Setting pin for sourcing current, EC to GND as well
    pinMode(pHPower, OUTPUT);

    Serial.begin(9600);

    // oled.begin(&Adafruit128x64, I2C_ADDRESS);
    // SDcard
    while (!Serial) {
        ; // wait for serial port to connect. Needed for native USB port only
        Serial.print("Initializing_SD_card...");
    }
    if (!Serial.begin(3)) {
        Serial.println("initialization_failed!");
        return;
    }

    // Initialize files
    Serial.println("initialization_done.");
    LightData = SD.open("light.txt", FILE_WRITE);
    if (LightData) {
        LightData.println("LDR, Temperature, EC, pH, millis, MissingPAR, TotalPAR");
    }
```
APPENDIX B. ARDUINO CODE

```cpp
Serial.println("LDR, Temperature, EC, pH, millis, 
--- Missing PAR, Total PAR");
else {
  Serial.println("Error writing to file _light.txt");
}
LightData.close();
PARData = SD.open("PAR.txt", FILE_WRITE);
PARData.close();
AddedData = SD.open("Added.txt", FILE_WRITE);
AddedData.close();

// EC meter
int R1 = (R1 + Ra); // Taking into account Powering Pin Resistance
 delay(1000);
sensors.begin();
delay(1000);

// Make one reading first to rule out bad readings
sensors.requestTemperatures(); // Send the command to get temperatures
Temperature = sensors.getTempCByIndex(0); // Stores Value in Variable

/******************** NUTRITION DOSAGE ********************/  
void TurnNutrDosPumpOn(){
  /* Depending on EC readings the nutrition dosage pump is tuned on */
  if (EC25 < 1) {
    digitalWrite(NutrDosing, HIGH); // pumps 1.7 ml of nutrient solution
    delay(1000);
    digitalWrite(NutrDosing, LOW);
    Serial.println("Nutrients added 1.7ml");
    AddedData = SD.open("Added.txt", FILE_WRITE);
    if (AddedData) {
      AddedData.print(millis());
      AddedData.println("Nutrients added 1.7ml");
    } else {
      Serial.println("Error writing to file _Added.txt");
    }
    AddedData.close();
  }
  else if (EC25 < 1.5) {
    digitalWrite(NutrDosing, HIGH); // pumps 0.83 ml of nutrient solution
    delay(500);
    digitalWrite(NutrDosing, LOW);
    Serial.println("Nutrients added 0.8 ml");
    AddedData = SD.open("Added.txt", FILE_WRITE);
    if (AddedData) {
      AddedData.print(millis());
      AddedData.println("Nutrients added 0.8 ml");
    } else {
      Serial.println("Error writing to file _Added.txt");
    }
    AddedData.close();
  }
else if (EC25 < 1.5) {
  digitalWrite(NutrDosing, HIGH); // pumps 0.83 ml of nutrient solution
  delay(500);
  digitalWrite(NutrDosing, LOW);
  Serial.println("Nutrients added 0.8 ml");
  AddedData = SD.open("Added.txt", FILE_WRITE);
  if (AddedData) {
    AddedData.print(millis());
    AddedData.println("Nutrients added 0.8 ml");
  } else {
```
else {
    Serial.println("Error
writing


to


file


Added. txt");
}  
AddedData.close();
}
else {
    digitalWrite(NutrDosing, LOW);
}

/***************** EC CHECK ******************/
/* Measures EC in the tank with compensation  */
void GetEC(){

    float PPMconversion = 0.64;
    float TemperatureCoef = 0.019;  // this changes depending on what chemical
    we are measuring
    float K = 3;  // From calibration (Data in google drive)
    float EC = 0;
    float raw = 0;
    float Vin = 5;
    float Vdrop = 0;
    
    // float buffer = 0;
    sensors.requestTemperatures(); // Send the command to get temperatures
    Temperature = sensors.getTempCByIndex(0); // Stores Value in Variable
    delay(100);
    // Estimates Resistance of Liquid
    digitalWrite(ECPower,HIGH);
    raw = analogRead(ECPin);
    raw = analogRead(ECPin);  // This is not a mistake, first reading will be low because if charged a capacitor
    digitalWrite(ECPower,LOW);
    
    // Converts to EC
    Vdrop = (Vin*raw)/1024.0;
    Rc = (Vdrop*R1)/(Vin-Vdrop);
    Rc = Rc-Ra;  // accounting for Digital Pin Resistance
    EC = 1000/(Rc*K);
    
    // Compensating For Temperature
    EC25 =  EC/ (1+ TemperatureCoef*(Temperature-25.0));
    ppm = (EC25)*(PPMconversion*1000);
}

void PrintECReadings(){
    Serial.print("Rc: ");
    Serial.print(Rc);
APPENDIX B. ARDUINO CODE

Serial.print("EC: ");
Serial.print(EC25);
Serial.print("Simens ");
Serial.print(ppm);
Serial.print("ppm ");
Serial.print("Temperature: ");
Serial.println("C");

/**************************************************************************
* Measures the light intensity of the sensor *
* calculates a PAR-value depending on if the light *
* is on or off. *
**************************************************************************/

void GetLDR (bool lightOn) {
  int rawData;
  float resistVoltage, ldrVoltage;
  float ldrResistance;
  float factor;
  int maxValue = 1023;
  int maxVoltage = 5;
  float sunfactor = 0.0185;

  Serial.print("light on/off:");
  Serial.println(lightOn);
  if (lightOn) { //compensates for lamp
    factor = lightfactor;
  } else {
    factor = sunfactor;
  }

  rawData = analogRead(LDR_PIN);
  ldrVoltage = (float) rawData * maxVoltage / maxValue;

  //ldrVoltage = maxVoltage - resistVoltage;
  resistVoltage = 5 - ldrVoltage;
  ldrResistance = ldrVoltage / resistVoltage * 10;
  ldrLux = (pow(ldrResistance, -1.426))*(1.8357 * pow(10, 3));
  PPFDvalue = ldrLux * factor;
}

/**************************************************************************
* Saves Lux and PAR-value to SDcard *
**************************************************************************/

void SaveToSDCard() {
  /* open the file. note that only one file can be open at a time, 
  so you have to close this one 
  before opening another. 
  Add measurements to the file LightData and PARData. */

  LightData = SD.open("light.txt", FILE_WRITE);
  if (LightData) {
      /* write the measurements to the file */
      /* close the file */
  }
}
LightData.print(ldrLux);
LightData.print(",");
LightData.print(Temperature);
LightData.print(",");
LightData.print(EC25);
LightData.print(",");
LightData.print(phValue);
LightData.print(",");
LightData.println(millis());
LightData.print(",");
LightData.println(missingPAR);
Serial.print("Added to file:");
Serial.print(ldrLux);
Serial.print(",");
Serial.print(Temperature);
Serial.print(",");
Serial.print(EC25);
Serial.print(",");
Serial.print(phValue);
Serial.print(",");
Serial.print(millis());
Serial.print(",");
Serial.println(missingPAR);
Serial.print(",");
Serial.println(totalPAR);
}
else {
  Serial.println("Error writing to file: light.txt");
}
LightData.close();
PARData = SD.open("PAR.txt", FILE_WRITE);
if (PARData) {
  PARData.println(PPFDvalue);
  Serial.print("Added to file PAR:");
  Serial.println(PPFDvalue);
} else {
  Serial.println("Error writing to file PAR.txt");
}
PARData.close();

/******************** LIGHTCONTROL ON/OFF ********************/
/* Evaluates if the light should be turned on or off
 * if the sum of the past 24 values have been below a given limit,
 * the sum of the readings of
 * the PAR-file accumulated during the day is calculated.
 */
int lightControl() {
  //Zero-set variables before running program

APPENDIX B. ARDUINO CODE

```cpp
int newValues = 48;
int countSum = 1;
int lineCount = 0;
float tempSum; // temporary sum
unsigned long currentMillis = millis();
float PARsum = 0;

// read from file and sum up the new PAR-values
if (currentMillis - previousMillis >= lightinterval){
    PARData = SD.open("PAR.txt");
    if (PARData){
        while (PARData.available()){ // reads next line as string
            String value = PARData.readStringUntil(\n');
            lineCount ++ ;
            tempSum = atof(value.c_str()); // converts string to float
            while (lineCount >= newValues){ // sum up new values
                PARsum += tempSum;
                countSum ++;
            }
            // calculates DLI
            totalPAR = (PARsum/countSum) * 24 * 0.036;
            Serial.print("The DLI is: ");
            Serial.println(totalPAR, 7);
            previousMillis = millis();
            newValues += 29; // 144 // make sure it only adds the latest values
            if (totalPAR < 10) {
                Serial.println("The total PAR-value is not high enough, turn the light on");
                lightOn = true;
                digitalWrite(lamp, HIGH); // turns the light on
                PARData.close();
                PARsum = 0;
                missingPAR = 15 - totalPAR;
                lampinterval = missingPAR*3600*1000 / (310*lightfactor); // calculates time lamp should be on
                return lightOn;
            }
        }
    }
}
PARData.close();
lightOn = false;
return lightOn;
}
PARData.close();
lightOn = false;
return lightOn;
```

/*************** TURN LIGHT ON/OFF *******************/
/* turns the light on if it is set as true, keeps the light on
until the calculated time for missing light is reached. */
void TurnLightOn()
{
  if (lightOn)
  
    unsigned long currentMillis = millis();
    // turns the light off if its been the missing time
    if (currentMillis - previousMillis >= (lampinterval - 1000))
     
      digitalWrite(lamp, LOW);
    
}
else
   digitalWrite(lamp, HIGH);

}

/*************** WATERLEVEL *******************/
void WaterLevelCirc()

  /* turns waterpump on and runs for 5 seconds,
circulates the water in the system */
  Serial.print("START_WATERPUMP");
  digitalWrite(WaterPump, HIGH);
  delay(5000);  // run waterpump in 5 seconds to fill growth tray
  digitalWrite(WaterPump, LOW);
  Serial.println("STOP_WATERPUMP");
}

/*************** pH level *******************/
void GetpH()

  takes ten readings, once every 60s, discards the highest and lowest
  value, take average of the rest to get a stable reading */

  int buf[10];
  digitalWrite(pHPower, HIGH);
  delay(3000);
  for(int i=0;i<10;i++) {
    buf[i]=analogRead(pHreading);  // take 10 readings every 5s
    delay(60000);
  }
  for(int i=0;i<9;i++) { // sort readings from size and discard
    for(int j=i+1;j<10;j++){
      //
### APPENDIX B. ARDUINO CODE

```cpp
if(buf[i] > buf[j]){
    temp = buf[i];
    buf[i] = buf[j];
    buf[j] = temp;
}
}
float avgValue = 0;
digitalWrite(pHPower, LOW);
for(int i = 2; i < 8; i++){
    avgValue += buf[i];
}
float pHVol = (float)avgValue * 5.0 / 1023.6;
phValue = 7 + ((2.5 - pHVol) / 0.176);
Serial.println("pH value:");
Serial.println(phValue);

void TurnPHDosPumpOn(){
    /* Adds pH-lowering buffer */
    if (phValue > 7.5) {
        digitalWrite(pHDosing, HIGH); // pumps ml of nutrient solution
        delay(1000);
        digitalWrite(pHDosing, LOW);
        WaterLevelCirc();
        delay(60000);
        Serial.println("pH-buffer added: 1.67 ml");
        AddedData = SD.open("Added.txt", FILE_WRITE);
        if (AddedData) {
            AddedData.print(millis());
            AddedData.println("pH-buffer added: 1.67 ml");
        } else {
            Serial.println("Error writing to file _Added.txt");
        }
        AddedData.close();
    }
    else if (phValue > 6.5) {
        digitalWrite(pHDosing, HIGH); // pumps 1.7 ml of nutrient solution
        delay(500);
        digitalWrite(pHDosing, LOW);
        Serial.println("pH-buffer added: 0.83 ml");
        AddedData = SD.open("Added.txt", FILE_WRITE);
        if (AddedData) {
            AddedData.print(millis());
            AddedData.println("pH-buffer added: 0.83 ml");
        }
    }
    else {
        Serial.println("Error writing to file _Added.txt");
    }
}
```

40
else {
    Serial.println("Error_writing_to_file_Added.txt");
} 
AddedData.close();

} else {
    digitalWrite(pHDosing, LOW);
}

/MMAIN

*/

void loop(){
    //digitalWrite(lamp,HIGH);
    Serial.print("Count:");
    Serial.println(count);
    GetLDR(lightOn);
    WaterLevelCirc();

delay(120000);
    GetpH();
    TurnPHDosPumpOn();
    GetEC();
    PrintECReadings();
    TurnNutrDosPumpOn();
    SaveToSDCard();

    lightOn = lightControl();
    TurnLightOn();
    count ++;
    Serial.println("" );
    //delay(120000);
    //delay(60000);
    delay(2810900);
    // measure every 5 min 300000, 30 min 1800000
}