HELLENIC MEDITERRANEAN UNIVERSITY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

CURRICULUM: INFORMATICS ENGINEERING T.E

Thesis

CALCULATION OF VEGETATION INDICATORS IN CROPS THROUGH OPTICAL METHODS AND IOT TECHNOLOGIES

ΥΠΟΛΟΓΙΣΜΟΣ ΔΕΙΚΤΩΝ ΒΛΑΣΤΗΣΗΣ ΣΕ ΚΑΛΛΙΕΡΓΕΙΕΣ ΜΕΣΩ ΟΠΤΙΚΩΝ ΜΕΘΟΔΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΩΝ ΙΟΤ

Author: Kyprakis Ioannis

Supervisor: Dr. Panagiotakis Spyridon

2022

Acknowledgments

In this spot, I would like to express my deepest appreciation to everyone who helped me with this thesis project; through them, I learned a lot of things and helped me with all the obstacles I encountered in completing the thesis.

Words cannot express my gratitude to my professor Dr. Panagiotakis Spyridon who is also the supervisor of this thesis. With his help, I was able to go through all the difficulties and obstacles I came through. He was very supportive and always there to support me.

I would also like to express my deepest appreciation to Dr. Kamarianakis Zacharias, who helped me with the problems I came across. He also helped me to obtain deep knowledge about microcontrollers, precision agriculture, spectroscopy, and microelectronics.

Finally, I would like to thank all the educational staff of our department for all the knowledge they passed down to us. They do extremely good work although we went through a difficult time with COVID-19. They did their best and gave us everything we needed to move on.

Abstract

Calculation of vegetation indicators is very important to understand a plant's health, and how the plant behaves under certain environmental conditions. Those calculations are extremely helpful to agronomists and farmers and have applications in various fields such as agronomy, biology, botany, and many others. Many devices are made for this purpose, but most come with an excessive cost, low precision, or are too complex to use.

This thesis proposes a precise, low-cost, easy-to-use device that calculates crop vegetation indicators through non-destructive, contactless optical methods and IoT technologies. The optical methods we use to calculate the indices are based on light reflection. The light reflected from the leaf's surface will be measured. Some of the vegetation indices, that we will calculate are Simple Ratio, Normalized Difference Vegetation Index (NDVI), $NDVI_g$, $NDVI_b$), Infrared Percentage Vegetation Index (IPVI), and Enhanced Vegetation Index (EVI).

We will evaluate with experiments the precision of our device. We will use LEDs that emit at certain wavelengths (465nm, 535nm, 630nm, and 840nm) and measure the reflectance from the surface of the leaves. The comparison revealed similar performance, demonstrating a strong correlation with the HR2000+ spectrometer (R^2 = 0.92-0.97), proving the device's high potential for precise plant stress measurements.

The hardware implementation consists of an Arduino Mega, sensors, modules, and other electronic components. We will also use I2C and SPI protocols to achieve communication between our microcontroller and the modules. The device will measure the vegetation indicators and then the values will be stored on an external SD card. The software part of the device was implemented using Arduino IDE v1.8.19.

Table of Contents

1. Introduction

The thesis's main purpose is to study, design, and analyze a portable device that will measure vegetation indexes. Our implementation can be considered the marriage of agronomy and computer science. To achieve that many different approaches in these areas were studied. Those approaches were about the architecture, technology, and theoretical background of the sciences and the devices. Through this study, certain needs were determined about the principles of our device but also the methodology to be followed.

1.1 Incentives

The Fourth Industrial Revolution, 4IR, or Industry 4.0, conceptualizes the 21st century's rapid transformation in technology, industry, and societal patterns and processes as a result of growing interconnectedness and intelligent automation. The word has been widely used in scientific literature and was made famous in 2015 by Klaus Schwab, the founder and executive chairperson of the World Economic Forum. According to Schwab, the changes are more than merely efficiency enhancements; they represent a significant transition in industrial capitalism.

The basic pillars of the Fourth Industrial Revolution are the Internet of Thing (IoT) [1][2][3], big data [4][5][6], artificial intelligence/ machine learning/ computer vision [7][8], cloud/edge computing [9][10], advances in networking (sdn/nfv, 5g) [11][12][13], augmented/virtual reality [14][15], robotics [16][17][18][19], and virtualization [20][21].

Industry 4.0 key themes:

- Interconnection, which is the potential of machines, devices, sensors, and people to link up and communicate with one another via the Internet of Things or the Internet of People (IoP)
- Information transparency, which is the transparency afforded by Industry 4.0 technology provides operators with comprehensive information to make decisions. Interconnectivity enables users to gather enormous volumes of data and information from every stage of the manufacturing process, pinpoint crucial areas that could use improvement, and ultimately increase functionality.
- Technical assistance, which is the ability of systems to use technology to support people in making decisions and solving problems, as well as to assist people with risky or difficult jobs.
- Decentralized decisions are the ability of cyber-physical systems to act independently and as autonomously as possible. Tasks are only transferred to a higher level when exceptions, interference, or conflicting objectives occur.

Nowadays, agriculture technology is advancing faster than ever before. This advancement intends to help farmers, agronomists, and businesses to be more efficient, remunerative, and eco-friendly. Agriculture today is very important and there is an urgent need for intensification and automation of food production but with environmentally friendly methods due to overpopulation, climate change as well as geostrategic changes at a global level.

The fourth industrial revolution in agriculture is called Agriculture 4.0. In 2018, the World Government Summit published its report called Agriculture 4.0 – The Future Of Farming Technology [22], in collaboration with Oliver Wyman. The research discusses the four key trends that will put pressure on agriculture in the foreseeable future: population growth, resource scarcity, climate change, and food waste. There are many modern applications of technology in agriculture that try to achieve the above.

Indoor Vertical Farming [23][24][25]: Indoor vertical farming can boost crop yields, circumvent land-use restrictions, and even reduce the environmental effect of farming by reducing supply-chain travel distance. Growing food in a controlled, enclosed environment while it is stacked one on top of the other is known as indoor vertical farming. Compared to conventional farming techniques, using growing shelves installed vertically [Fig. 1] greatly minimizes the quantity of land needed to cultivate plants. Because it can flourish in a small area, this sort of growth is frequently linked to urban farming. In some configurations, vertical farms are unusual because no soil is needed for plant growth. Most are either hydroponic (vegetables are grown in a bowl of nutrient-rich water) or aeroponic (water and nutrients are routinely sprayed on the plant roots). Artificial grow lights are utilized in place of natural sunshine.

Fig. 1 Vertically installed plants. Photo credit: Oasis Biotech.

Farm Automation [26][27][28]: Automation of the crop or livestock production cycle on farms increases efficiency and is frequently referred to as "smart farming." A growing number of businesses are focusing on robotics innovation to create robots that can automatically water plants [Fig. 2], sow seeds, and operate tractors and harvesters. Even though these technologies are still relatively new, more traditional agriculture businesses are incorporating farm automation into their operations.

Fig. 2 Automated watering using drones. Source: https://stories.pinduoduo-global.com/agritech-hub/how-smart-farming-automation-is-transforming-agriculture

Modern Greenhouses [29]: The industry is currently experiencing a blossoming unlike any other period in its history, in large part because of the enormous recent advances in developing technology. Today, there are more and more large-scale, heavily capitalized, and urban-focused greenhouses sprouting. Modern greenhouses are becoming more and more technologically advanced [Fig. 3], incorporating automated control systems and LED lighting to precisely customize the growing environment. To meet the rising demands for local food year-round, successful greenhouse businesses are expanding dramatically and situating their growing operations close to metropolitan centers. The greenhouse business is also investing more money to achieve these accomplishments, utilizing venture capital and other investment sources to provide the foundation required to succeed in the existing economy.

Fig. 3 Technologically advanced greenhouse. Source https://www.greenhousegrower.com/technology/how-a-new-narrow-sprayer-addresses-unique-greenhouse-conditions/

Precision Agriculture [30][31][32]: With the use of new precision agriculture companies, farmers will be able to maximize yields by managing every aspect of crop production, including moisture levels, pest stress, soil conditions, and microclimates. Precision agriculture helps farmers enhance productivity and control expenses by offering more precise methods for planting and producing crops. Companies that specialize in precision agriculture have a great opportunity to expand. According to a recent study by Grand View Research, Inc., the market for precision agriculture would grow to $43.4 billion by 2025. Farmers of the new generation are drawn to quicker, more adaptable companies that methodically maximize agricultural output.

Agronomists need to be able to see introspectively what is happening in plants without resorting to destructive methods as they did until recently because this will allow automation and acceleration of many processes such as disease prevention in plants, better plant nutrition (fertilization), optimal watering without wasting water, better photosynthesis

Our technology endeavors to achieve all the above at a low cost, and it also helps humans to understand what the plant needs, to be healthier and grow strong. To achieve that we will combine physics, informatics, microelectronics engineering, and agricultural science. Concerning physics, we will use light reflection and spectroscopy to measure the values we need, to understand the plant's condition. We will also use informatics, to program the microcontroller and achieve ease of use for our device. It is particularly important for our device to not have much complexity and that is because often according to statistics farmers are not well trained with modern technologies or they are advanced in years [33]. Thus, we will use microelectronics engineering to combine all the above and create a solid device that will be like every other industrial device. Eventually, we will use agricultural science focused on the process of photosynthesis, which will be explained further on.

Combining all the above, IoT, and precision agriculture we will create a device that uses reflection on plant leaves to calculate vegetation indices. It will be a plant-friendly device since it will not damage or has any effect on the health of the plant. We will measure reflection on certain wavelengths to get the right measurements and compute the vegetation indices we need. It will also use AS7341, which is an eleven-channel spectral color sensor that comes at a low cost.

1.2 Thesis structure

The thesis's first chapter serves as an introduction, giving both a straightforward and analytical description. It also provides the technical background necessary for the implementation and talks about the incentives that helped us decide to deal with this particular construction. Here we provide a short description of the chapters and the sections of our thesis.

In chapter two, the state of the art is analyzed. We will analyze the methods needed to create our device, and obtain useful information and measures. We will cover a lot of concepts like photosynthesis, spectroscopy, plant health monitoring, vegetation indices, light reflectance, absorbance, and transmittance. We will also describe related work.

In chapter three, we will see the device design and how we encapsulated the AS7341 with the LEDs with a 3D printed box in order to be more functional and easier to use. We will also describe how the menu is designed and how its functionality.

In chapter four, we will analyze and describe the hardware part of our device which includes microcontrollers, sensors, protocols, and various electrical components. Next will also illustrate the electronic needs of our device and sensors. Further on will also describe the software implementation required for our device's proper function. A short description of each function’s operability and construction will be given.

In chapter five, statistical analysis will be provided. Diagrams and figures will show the results of the device’s correlation with the industrial device.

Finally, the last chapter comes to conclusions produced since the undertaking of the work, while also suggesting extensions and future work.

2. State of the art

In this chapter, we will analyze the scientific part we need for our device and also the technologies, procedures, and similar work that has been studied.

2.1 Photosynthesis

Photosynthesis is the process by which plants use sunlight, water, and carbon dioxide to create oxygen and energy in the form of sugar. Photosynthesis is, without a doubt, the most important biological process on the entire earth. Consuming carbon dioxide and setting free oxygen, photosynthesis has made the world into the viable environment we live in today. The basic principle of photosynthesis is the fact that plants absorb solar radiation. The spectral range that plant pigments absorb solar radiation is in the range of 400-700nm [34]. In this range, plants begin a series of chemical reactions in which plants from simple inorganic compounds such as minerals, water, and carbon dioxide synthesize organic compounds such as proteins, sugars, and fats.

A wide spectrum of experts, including botanists, biophysicists, and biochemists, have shown great interest in photosynthesis because the process is crucial to the whole food chain. Agronomists particularly put considerable emphasis on the fact that vegetation photosynthetic processes are directly connected to plant stress and health [35].

Chlorophyll refers to several closely related green pigments that have high absorption in the blue and red spectral areas and lower absorption in the green region, which provides plants with their typical green color. The plant molecule with the most nitrogen throughout the growing season is chlorophyll since each molecule includes four nitrogen atoms. Spongy tissue is found underneath the palisade layer of the leaf and plays a significant role in the thermal balance of the plant. When the spongy layer is under drought stress, it quickly collapses and loses its capacity to reflect infrared light. An increase in red reflection and near-infrared absorption accompanies this process. As a result, the amount of chlorophyll and the condition of the sponge tissue in the leaves are reliable measures of the physiological well-being of the plant. There are six distinct types of chlorophyll, but the main types are chlorophyll A and chlorophyll B.

Chlorophyll A is the primary pigment used in photosynthesis. Chlorophyll B is an auxiliary pigment because photosynthesis can take place without it. Chlorophyll A is present in all photosynthesis-capable organisms, but Chlorophyll B is not. Chlorophyll A absorbs light from the orange-red and violet-blue areas of the electromagnetic spectrum. Chlorophyll A transfers energy to the reaction center and donates two excited electrons to the electron transport chain. Chlorophyll A serves as the main electron donor in the electron transport chain, which is its primary function. From there, the sun's energy will eventually transform into chemical energy that the organism can use for cellular functions.

The color of the light that Chlorophyll A and B absorb is one of their primary differences. Blue light is absorbed by chlorophyll B. Chlorophyll B’s central role is to broaden the absorption spectrum of organisms. That way, organisms can absorb more energy from the higher frequency blue light part of the spectrum. Chlorophyll B's role in cells enables organisms to convert a wider range of solar energy into chemical energy. Having more chlorophyll B in the chloroplasts of cells is adaptive. More chlorophyll B is present in the chloroplasts of plants that receive less sunlight. An adaptation to the shade is an increase in chlorophyll B, which enables the plant to absorb a wider spectrum of light wavelengths. The additional energy Chlorophyll B absorbs is transferred to Chlorophyll A.

The structures of chlorophylls A and B are extremely similar. Due to a hydrophobic tail and a hydrophilic head, both have a "tadpole" shape. The head consists of a porphyrin ring, with magnesium in the center. The porphyrin ring of chlorophyll is where light energy is absorbed. Only one atom in a sidechain on the third carbon distinguishes chlorophylls A and B. Unlike in B, where the third carbon is attached to an aldehyde group, the third carbon is attached to a methyl group in A.

2.2 Principles of spectroscopy

The study of how light and other types of radiation are absorbed and emitted by materials is known as spectroscopy. It entails separating light (or more specifically electromagnetic radiation) into its various wavelengths (spectrum), much as how a prism divides light into a rainbow of colors. In reality, photographic plates and a prism were used to do traditional spectroscopy. Spectroscopy as a science began with Isaac Newton splitting light with a prism and was called Optics. Therefore, it was originally the study of visible light which we call color that later under the studies of James Clerk Maxwell came to include the entire electromagnetic spectrum.

Modern spectroscopy projects the light that has been scattered by a diffraction grating onto CCDs (charge-coupled devices), which are comparable to those found in digital cameras. This digital format makes it simple to extract the 2D spectra and alter them to create 1D spectra with a significant quantity of valuable data.

Recently, the definition of spectroscopy has been expanded to also include the study of the interactions between particles such as electrons, protons, and ions, as well as their interaction with other particles as a function of their collision energy.

Spectrometers, spectrophotometers, spectrographs, and spectral analyzers are all terms used to describe spectrum measurement equipment. The majority of laboratory spectroscopic analyses begin with the sample to be examined, followed by the selection of a light source from any desired region of the light spectrum, the light passing through the sample to a dispersion array (diffraction grating instrument), and finally the light being captured by a photodiode. The light dispersion device is necessary for the telescope to be used for astronomical purposes. There are other variations of this fundamental configuration that can be used.

Absorption spectroscopy: Absorption takes place when a substance absorbs energy from a radiating source. Measuring the percentage of energy that passes through a substance to assess absorption is common; absorption reduces the transmitted part.

Emission spectroscopy: Emission indicates that radiative energy is released by a material. The temperature-dependent spontaneous emission spectrum of a substance is called its blackbody spectrum. Instruments like the atmospheric emitted radiance interferometer can measure this characteristic in infrared. Other energy sources, such as flames, sparks, electric arcs, or electromagnetic radiation in the case of fluorescence, can also cause emission.

Elastic scattering and reflection spectroscopy determine how incident radiation is reflected or scattered by a material. Crystallography employs the scattering of high-energy radiation, such as x-rays and electrons, to study the arrangement of atoms in proteins and solid crystals.

Wavelength

The distance over which a periodic wave's shape repeats is known as the wavelength in physics [Fig. 4]. It is a property of both traveling waves and standing waves as well as other spatial wave patterns. It is the distance between two successive corresponding locations of the same phase on the wave, such as two nearby crests, troughs, or zero crossings. The spatial frequency is the reciprocal of the wavelength. The Greek letter lambda (λ) is frequently used to represent wavelength. The term wavelength is also occasionally used to refer to modulated waves, their sinusoidal envelopes, or waves created by the interference of several sinusoids.

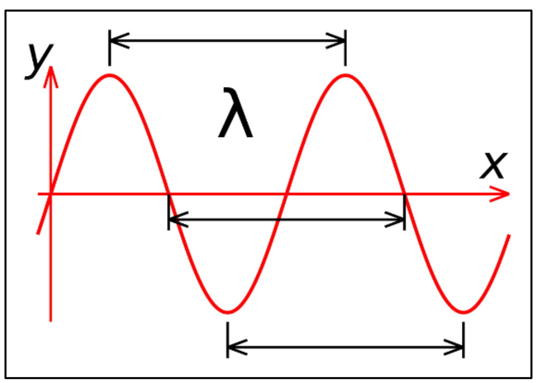

Fig. 4 Wavelength of a sine wave. Source Google.

Electromagnetic Spectrum

The electromagnetic spectrum is the range of electromagnetic radiation frequencies, together with the corresponding wavelengths and photon energies. The electromagnetic spectrum includes electromagnetic waves with frequencies from below one hertz to above 10^{25} hertz, which correspond to wavelengths ranging from thousands of kilometers down to a tiny portion of the size of an atomic nucleus. The electromagnetic waves that are contained within each of these bands are known by a variety of names; starting at the low frequency (long wavelength) end of the spectrum, these include radio waves, microwaves, infrared, visible light, ultraviolet, X-rays, and gamma rays, and ending at the high frequency (short wavelength) end [Fig. 5].

In terms of their generation, interactions with matter, and potential uses, the electromagnetic waves in each of these bands differ from one another. While the limit for short wavelengths is believed to be close to the Planck length, the limit for long wavelengths remains unknown. Ionizing radiation includes the ultraviolet spectrum, soft X-rays, hard X-rays, and gamma rays because their photons are powerful enough to ionize atoms and trigger chemical processes. Ionizing radiation exposure carries the risk of radiation illness, cancer, and DNA damage. Longer wavelengths and visible light radiation are categorized as nonionizing radiation since their energies are insufficient to produce these effects.

Spectroscopy can be used to separate waves of various frequencies over the majority of the electromagnetic spectrum, resulting in a spectrum of individual frequencies. The interactions of electromagnetic waves with matter are studied using spectroscopy.

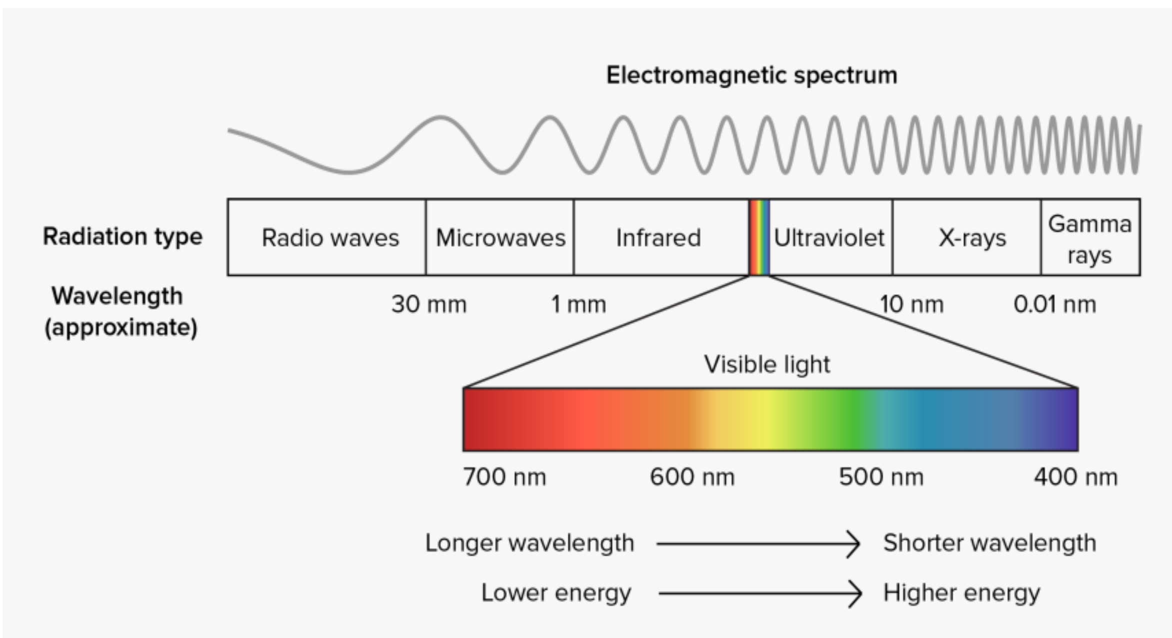

Fig. 5 Electromagnetic spectrum. Source: https://www.pngitem.com/middle/ixmohTo_electromagnetic-spectrum-png-transparent-png/

2.3 Reflectance, absorbance, transmittance

As is widely accepted plant leaves are reacting differently to visible and infrared radiation [36]. Concerning visible radiation, the leaves absorb 80%, reflect 10%, and transmit 10% of the radiation. On the other hand, regarding infrared radiation, the leaves absorb 20%, reflect 50% and, transmit 30% of the radiation [Fig. 6].

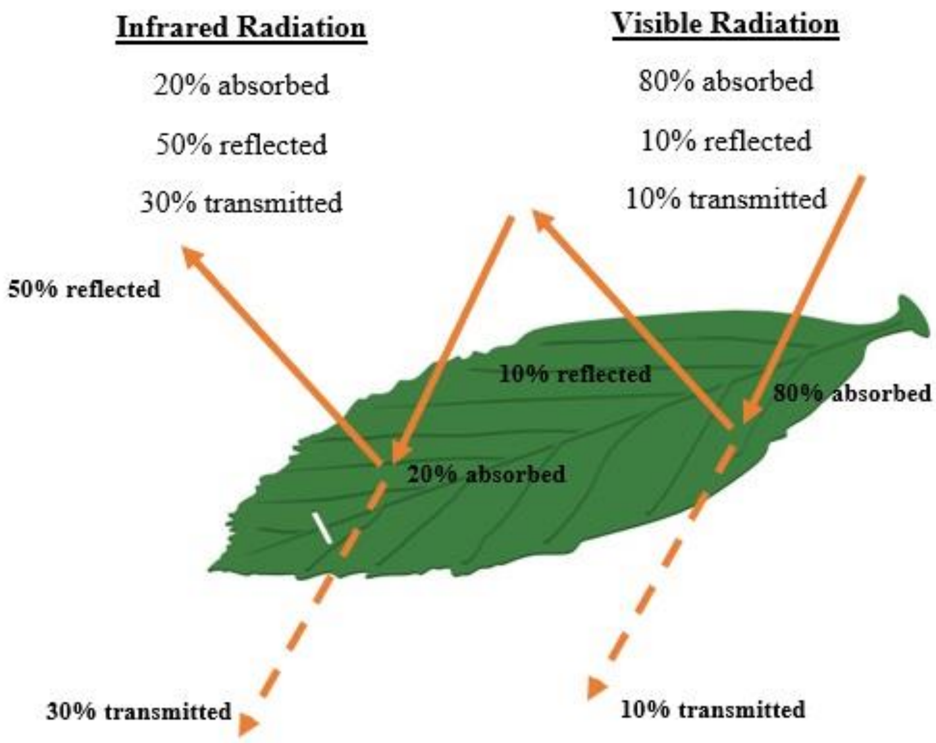

Fig. 6 Leaf's light reflectance, absorption, transmission

Light Reflection

The Greek mathematician Euclid, who carried out several experiments around 300 BC, is credited with some of the earliest descriptions of light reflection. Euclid appears to have had a thorough understanding of how light is reflected. But it wasn't until 1,500 years later that the Arab scientist Alhazen put forward a law outlining precisely what occurs to a light beam when it meets a flat surface and then bounces off into space.

When a ray of light approaches a surface and the light ray bounces back, it is called reflection of light. Light is seen in classical electrodynamics as an electromagnetic wave that can be represented by Maxwell's equations. Each particle in a material emits a tiny secondary wave in all directions like a dipole antenna as a result of the small polarization oscillations that light waves impact the material cause in the individual atoms (or oscillations of the electrons in metals). The Huygens-Fresnel principle states that the sum of all these waves results in specular reflection and refraction.

Depending on the nature of the interaction, light reflection [Fig. 7] is either specular (mirror-like) or diffuse (retaining energy but losing image) [Fig. 8]. According to the law of reflection, the angle at which the wave incident on the surface is equal to the angle at which it is reflected for specular reflection (such as at a mirror), whereas diffuse reflection is produced by rough surfaces that tend to reflect light in all directions. The surface's degree of smoothness or texture strongly impacts how much light is reflected by an object and how it is reflected. Nearly all of the light is evenly reflected when surface flaws are smaller than the wavelength of the incident light (as in the case of a mirror).

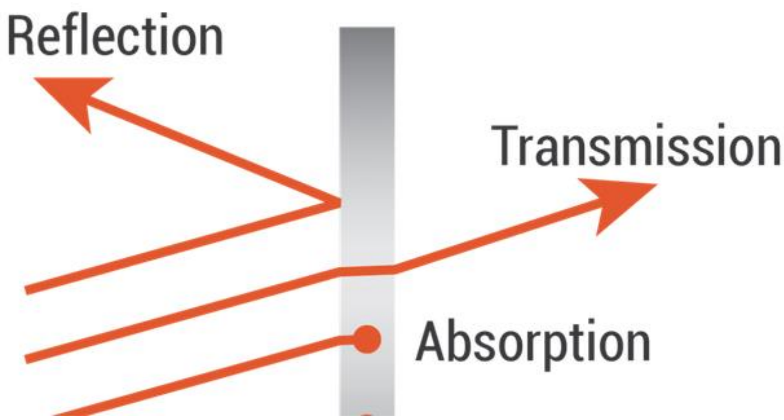

Fig. 7 Reflection, transmission, absorption of light

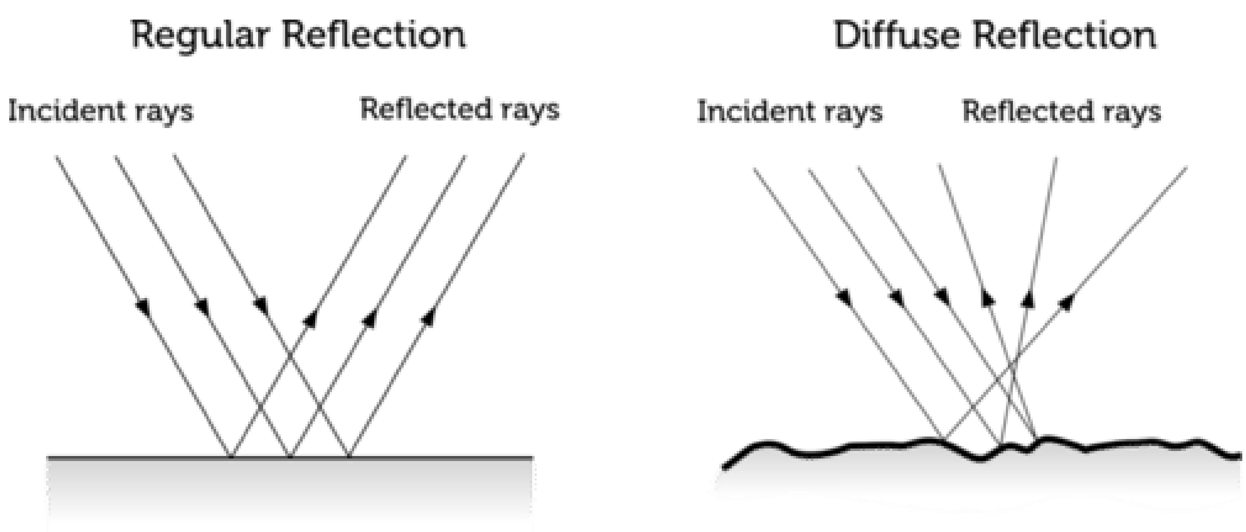

Fig. 8 Regular (specular)-Diffuse Reflection

Light Absorption

Electrons only can exist in distinct energy levels, often known as electron shells; they cannot exist in intermediate states. The ground state of an electron is the lowest possible energy level. Because energy levels are quantized, an electron needs to absorb a specific quantity of energy to transition from one energy level to another. Since there is an energy difference between the two levels, the energy absorbed by the electrons must be entirely equivalent to that difference.

An electron is said to be "excited" when it absorbs energy, which moves it to a greater level of energy away from the atom's nucleus. Excited electrons dislike it when they are in that state. This indicates that they quickly return to their initial energy level upon becoming excited and advancing to a higher one. However, to do so, they must release an energy package known as a photon. The size of the photon that is released is precisely equal to the initial magnitude of the jump that the electron had to undertake.

The term "absorption spectroscopy" refers to spectroscopic methods that assess how much radiation interacts with a sample and is absorbed as a function of frequency or wavelength. The sample takes in photons, or energy, from the emitting field. The absorption spectrum is the variation in absorption intensity as a function of frequency. Throughout the electromagnetic spectrum, absorption spectroscopy is used.

There are two types of absorption spectroscopy: atomic and molecular. Atomic absorption spectroscopy, which is typically applied to gases, is a technique for creating a spectrum when unbound atoms absorb various light wavelengths. Molecular absorption spectroscopy is the technique of creating a spectrum when whole molecules are absorbing different wavelengths of light (usually visible or ultraviolet) [Fig. 7].

Light Transmission

The transmittance of the surface of a material is its efficacy in transmitting radiant energy [Fig. 7]. It is the fraction of incident electromagnetic power that is transmitted through a sample, in contrast to the transmission coefficient, which is the ratio of the transmitted to an incident electric field. Internal transmittance refers to energy loss by absorption, whereas total transmittance is due to scattering, reflection, absorption, etc.

2.4 Plant health monitoring

Plant health monitoring is an attractive and supportable method that could be used for improving crop growth practices. It complements well-known agricultural practices like crop rotation, intercropping, and genetic modification that farmers use to increase yield [37][38]. It also makes it possible to precisely calibrate the best dosage and apply agrichemicals like pesticides, herbicides, or plant growth regulators [39][40]. Current chromatography-based analytical methods, however, are constraining the perspective of plant health monitoring to have an immediate impact on farm management choices [41]. These methods are destructive and laborintensive, requiring laboratory-based extraction and processing of numerous plant samples for each data point, despite being extremely sensitive and quantitative.

Most endoscopy methods in plants are harming their health. In some other cases, those methods require destroying or cutting some parts of the plant to get the measurements needed. An indicative example that can show us how destructive those methods are is the chlorophyll identification of phytoplankton. To achieve this measurement, we need to extract the phytoplankton from the sea by cutting it. Then it is processed with chemicals (acetone) to prepare it for centrifugation. After the centrifugation, it is, measured with the photometry method at 665-645 – 630nm. Subsequently, the plant is processed with hydrochloride, and it is measured again. As we can understand this methodology makes the phytoplankton -or any other plant- useless.

The removal of sugars from fresh plant tissue and measurement of their content using a spectrophotometric technique is another illustration of how harmful endoscopic techniques are. To obtain the specific extract, the selected plant tissues into small pieces are cut, place these pieces in a test tube containing the extractant and extract for a specific period at a specific temperature, periodically shaking the tube. We then take part of the filtrate into another test tube, add some reagents in a specific order and the color develops. Quantitative determination is made by measuring the absorption of radiation of a certain wavelength by the colored filtrate in a spectrophotometer and converting it into a concentration employing a reference curve. As we can see again the plant suffers damage that can affect its health or even destroy it completely.

Thus, we can infer that there is a big need for endoscopy in plants because it allows us to gain knowledge about the health of the plant and its need for nutrients or water. We can conclude that the conventionally harmful approaches did not benefit the plant-based just on these two techniques. Therefore, we aim to develop endoscopic techniques that are gentle on the plant, highly accurate, and contactless.

2.5 Vegetation indices

To highlight specific characteristics of a plant, vegetation indices, which are mathematical combinations of surface reflectance at two or more wavelengths, are a popular way to express the results of a spectral study in a more digestible format [42]. It is employed to emphasize the existence of green, vegetative features, thereby aiding in their differentiation from other objects in the scene. Different aspects of the vegetation cover in the image, such as the percentage of vegetation cover, the amount of chlorophyll content, the leaf area index, etc., could be evaluated depending on the transformation method and the spectral bands used. In general, slope effects and the lighting at the time of measurement do not affect any of the ratio indices. Some of the most common vegetation indices will be explained down below.

Simple Ratio (SR)

The ratio of reflectance measured in the Near Infra-Red (NIR) and red bands is the simplest VI. This is a quick method for separating green leaves from other scene elements and determining the visible relative biomass. Additionally, this value might be very helpful in differentiating between stressed and non-stressed vegetation. The range of values is from 0 to more than 30, where healthy vegetation generally falls between values of 2 to 8.

Equation: $$SR = \frac{\rho_{NIR}}{\rho_{RED}}$$

Green leaves display low reflectance in the blue and red regions, according to their spectral signature (the leaves are reflecting more in the green region and that is the reason they appear green). However, the NIR region has a relatively higher reflectance. So, when an object has a close reflectance in both the red and near-infrared bands, such as soil, the SR value is nearly 1. In contrast, the value for a green object would be much higher than 1.

Normalized Difference Vegetation Index (NDVI)

NDVI is another popular technique for assessing how much of an area is covered by vegetation. This index, which is a ratio, can be used to track crops throughout the growing season because it is unaffected by changes in slope, seasons, illumination, etc. It is computed by dividing the total of the reflectance from the NIR and red band by the difference in reflectance from those bands. NDVI ranges from -1 to 1. Values between -1 and 0 indicate dead plants or inorganic objects. NDVI values for live plants range between 0 to 1, with 1 being the healthiest and 0 being the least healthy.

Equation: $$NDVI = \frac{(\rho_{NIR} - \rho_{RED})}{(\rho_{NIR} + \rho_{RED})}$$

Many researchers report that the NDVI (800-640) = (R800- R640)/(R800+R640) index has a high correlation with biomass, chlorophyll, and leaf area index [43][44]. In research, it was shown that this index at 680 nm and 800 nm remains unchanged with changes in environmental conditions and more specifically light intensity [45]. The index gives a good correlation with the evolution of nutrient stress [46], while according to some other researchers, the index can be used to estimate the nitrogen content [47].

We will also calculate $NDVI_g$, $NDVI_b$, and Infrared Percentage Vegetation Index (IPVI). $NDVI_g$ is an indicator of the photosynthetic activity of the vegetation cover; it is primarily used to determine the moisture content and nitrogen concentration in plant leaves according to multispectral data which do not have an extreme red channel. It has a higher sensitivity to chlorophyll congregation compared to the NDVI measure. When evaluating aging and depressed plants, it is used.

Equation: $$NDVI_g = \frac{(\rho_{NIR} - \rho_{green})}{(\rho_{NIR} + \rho_{green})}\ , NDVI_b = \frac{(\rho_{NIR} - \rho_{blue})}{(\rho_{NIR} + \rho_{blue})}\ , IPVI = \frac{\rho_{NIR}}{(\rho_{NIR} + \rho_{red})}$$

Enhanced Vegetation Index (EVI)

The enhanced vegetation index (EVI) is an "optimized" vegetation index created to improve vegetation monitoring and enhance the vegetation signal by decoupling the canopy background signal and reducing atmospheric influences in high biomass regions. The value range for the EVI is –1 to +1, and for healthy vegetation, it varies between 0.2 and 0.8. EVI is computed by the equation down below [49]:

$$\boldsymbol{EVI = G \times \frac{(\rho_{NIR} - \rho_{RED})}{(\rho_{NIR} + C1 \times \rho_{RED} - C2 \times \rho_{BLUE} + L)}}$$

where:

1. $\rho_{NIR}/\rho_{RED}/\rho_{BLUE}$ are atmospherically-corrected and partially atmosphere-corrected (Rayleigh and ozone absorption) surface reflectance
2. L is the canopy background adjustment that addresses non-linear, differential NIR and red radiant transfer through a canopy, and
3. C1 and C2 are the coefficients of the aerosol resistance term, which uses the blue band to correct for aerosol influences in the red band.

The coefficients adopted in the MODIS-EVI algorithm are L=1, C1 = 6, C2 = 7.5, and G (gain factor) = 2.5.

The EVI is more sensitive to canopy structural variations, such as leaf area index (LAI), canopy type, plant physiognomy, and canopy architecture, compared to the NDVI, which is chlorophyll sensitive. The extraction of canopy biophysical parameters and the detection of vegetation changes are both enhanced by the two vegetation indices in studies of global vegetation.

2.6 Related work

Many non-industrial approaches are implemented in trying to calculate vegetation indices. Most of them use photodiodes to calculate the values required like Plant-O-Meter [50]. Plant-O-Meter is working with reflectance. The instrument can simultaneously illuminate the entire plant thanks to an embedded multispectral source that combines light sources of four wavelengths (465, 535, 630, and 850 nm). Rapid reflectance measurements are produced by sequential lighting and detection, which are wirelessly (Bluetooth) relayed to the android-operated device to process data storage. Plant-O-meter can simultaneously illuminate the entire plant thanks to an embedded multispectral source that combines light sources of four wavelengths. The device was tested in the lab using GreenSeeker and a SPECIM hyperspectral camera. It was also tested on the field using the GreenSeeker. The device also includes a GPS which is used to obtain the position of each measurement.

We also see the same methodology in trying to calculate soil nutrients and properties [51][52][53]. In the last 2 years, we see some approaches with the AS7341 sensor trying to measure LAI (leaf area index) [54] or PAR [55]. The second approach which is trying to obtain PAR values is our point of interest. This device uses AS7341 housed in a special 3D printed construction. To quantify PAR two indices are used PPFD and YPF. The PPFD considers all incident photons throughout a time and space interval within 400 and 700 nm, regardless of their wavelength. The YPF more closely considers the behavior of plant growth. Here, each photon between 360 and 760 nanometers is given a weight based on its wavelength and the relative quantum efficiency curve of plants. For the PPFD experiment, all channels were used, and the PAR data measured are compared with the LI-190 device. For the calibration of the sensor to determine the PAR, the method of referencing the measured data to a calibrated reference sensor was used. The AS7341 was exposed to different light conditions together with a reference PAR sensor (LI-190). By doing this many datasets were obtained, and multiple linear regression (MLR) was used to determine a transfer function of the spectrometer output signals to the PPFD.

Regarding the industrial part of manufacturing devices measuring vegetation indices, technology has thrived in the last few years. Many devices calculate NDVI, PAR, and other vegetation indices. Two of the most famous devices are GreenSeeker and ASD HandHeld 2.

The GreenSeeker device [Fig. 9] is a handheld portable crop sensor device. It is used to evaluate the health of a crop or plant in order to make the right nutrient management decisions. GreenSeeker is working by reflection. It emits light and measures reflectance at 660 nm (Red) and 770 nm (NIR). This way it calculates the NDVI. The recommended distance to measure the crop is 60-120cm. It also includes Bluetooth connectivity that connects the device with a mobile device running a compatible application (e.g., GreenSeeker BT Logger for Android™ mobile devices).

Fig. 9 GreenSeeker Source: user manual.

The ASD HandHeld 2 [Fig. 10] is a portable spectroradiometer that is used to perform rapid, precise, nondestructive, and non-contact or contact measurements. The HandHeld uses a highly sensitive detector array, low stray light grating, built-in shutter, DriftLock™ dark current compensation, and second order filtering, the HandHeld 2 instrument's unique spectrometer generates high signal-to-noise ratio spectra in less than a second. In a variety of circumstances, the HandHeld 2 instrument delivers incredibly accurate, swiftly determined reflectance, transmittance, radiance, and irradiance spectra. A self-contained measurement system is made up of an onboard, reclining color LCD monitor, integrated computer ability, significant internal data storage, laser targeting, and GPS compatibility.

It does not have its own light source so an external one must be used. The data taken can be extracted in .dat form which is compatible with excel. It comes with The Labsphere Spectralon white reference material, manufactured from a sintered polytetrafluoroethylene-based material (PTFE). The reference material is used to optimize the instrument's integration time to the illumination conditions. After the optimization, the device is ready to measure. The values may be stored internally or transferred to the computer via mini-USB (Tethered Mode).

The RS^3 software allows for the HandHeld 2 instrument to process data while in Tethered Mode. It also allows the user to operate the device from the computer without pressing any of the device's buttons.

Fig. 10 HandHeld 2 source: user manual.

2.7 Research questions

There are numerous different approaches for creating devices that measure vegetation indices. High-precision optical techniques usually require expensive equipment used by professionals, which severely limits their use. Furthermore, low-cost optical techniques display a lack of accuracy. Most of the time it is difficult to balance between high-precision, low-cost, and simplicity.

Regarding the industrial devices, their major drawbacks are high-cost, and complexity, they require a lot of training, and sometimes scientific knowledge that a common farmer is missing. Concerning the cost, most of the time is above 1,500 euros which makes them inaccessible to ordinary farmers with low budgets. Complexity and training come with complex and too detailed user manuals that are confusing the user instead of helping him. Some other devices measure only the reflectance at a certain range of wavelength and not a vegetation index, so the user has to make the calculations required.

On the other hand, custom-made low-cost devices provided by researchers also present drawbacks. Some of them are inaccuracy, complexity, and unsustainability. Inaccuracy is the main problem because most of the time the accuracy of these devices depends directly on the ankle and the distance that the measure is taken. A wrong hand movement can destroy an entire dataset if not a delete last measure option is not provided. Also, the complexity has to do again with complex and too detailed user manuals. Researchers give more weight to the scientific part of the device and not to the user-friendly part of it. This results in the user spending hours reading complex user manuals. Unsustainability is motivated by the same factor; researchers do not give much weight to creating a device that is sustainable and easy to use and they are satisfied with the functionality they have succeeded.

All of the aforementioned causes us to wonder the following:

1. How can we make the device simple?
2. How can we make the device easy to use and user-friendly?
3. How can we succeed in the balance between low-cost and high precision and stability?
4. How to measure vegetation indices without harming the plant?

3. Design

In this chapter, we will see all the information regarding our implementation. It is separated into four sections. Firstly, we will analyze the device design. Secondly, we will analyze the principle of menu operation.

3.1 Device Design

In our approach, we will try to create a portable multispectral optical instrument for precisely measuring plant stress. We will try to maintain precision, low cost, and simplicity as the basic principles for our device.

We will use a contactless optical technique based on reflectance in order to be plant-friendly and have zero impact on the plant's health. Such a technique is well accepted and acknowledged as a viable option for monitoring photosynthetic activity and plant health. These optical techniques and others, such as spectroscopy, and hyperspectral imaging, are based on the evaluation of the plant's optical signature in the visible and near-infrared range. In our case, we will use the light reflectance from the surface of the plant to evaluate the plant's condition.

Our approach will use an 11-channel spectrometer sensor, the AS7341 (from Waveshare). This low-cost sensor is typically used in mobile devices for applications such as spectrum identification and color matching. We will use an Arduino Mega 2560 to operate the AS7341. The main idea is that we will use 4 LEDs that are emitting at certain wavelengths. First, the blue LED will emit at about 465nm, then the green LED at about 535nm, the red LED at about 630nm, and the last one which is the IR LED will emit at 840nm.

Furthermore, we will use a set of modules and sensors to achieve a fully independent device that will measure and store data in an SD card. Our device will also be capable to show and delete data. All the hardware parts come at a low cost. This way we can keep the device cheap. AS7341 will help us to keep the simplicity and not have to resort to photodiode ICs, which will add complexity to our device. This sensor is also precise and stable so it will us with high accuracy and stability issues.

Briefly, we will use the AS7341 sensor to get the values of the reflectance. This sensor has eleven channels that measure light intensity at certain wavelengths. Each one of the eleven channels measures a certain spectral range, so we will activate only four channels to measure the reflectance. The channels we will activate the channel F3 (wavelength: 460-480nm) for the blue LED, channel F4 (wavelength: 500-520nm) for the green LED, channel F7 (wavelength: 610-630nm) for the red LED, and channel NIR (wavelength: 850-1050nm) [Fig. 11]. Next, we will use the SD card module to store the measured values. To achieve proper data logging, we also need time and date apart from the measured values. To achieve that we are using a Real Time Clock module the DS3231. We also want to show this data in real-time. The best and most used display tool is the Liquid Crystal Displays (LCD). A DHT-11 is used to get humidity and temperature. To end with we are using 4 LEDs at certain wavelengths. The LEDs were bought according to their intensity and their emittance. They are one of the most important components of our device so particular importance was given. The power supply is connected to Arduino's V_{in} directly. The device's power supply will be two 18650 batteries.

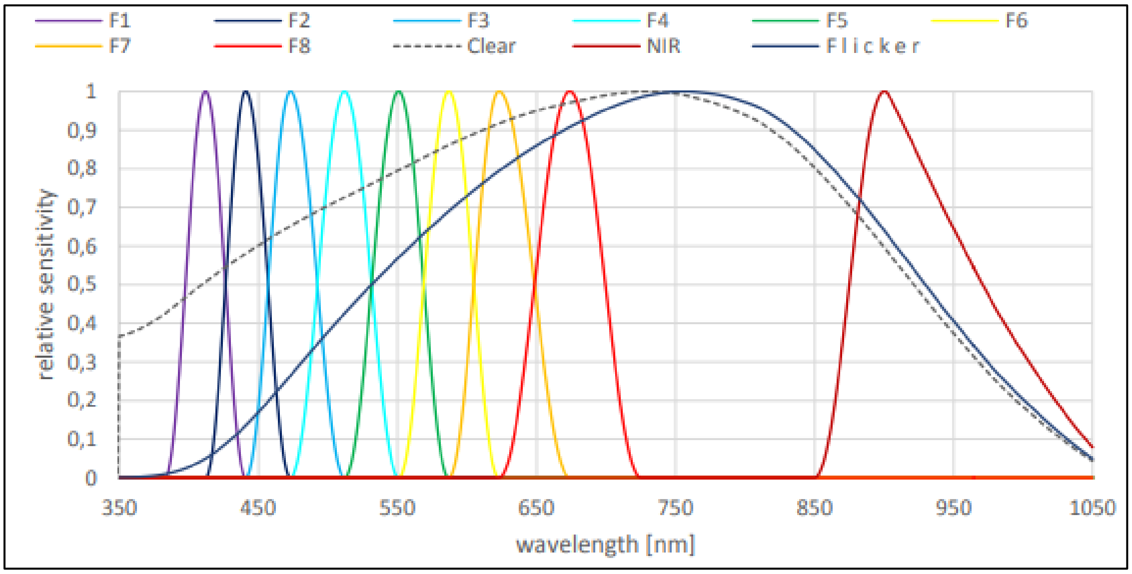

Fig. 11 Normalized Spectral Responsivity source: https://www.waveshare.com/wiki/AS7341_Spectral_Color_Sensor

The entire wiring diagram of the hardware is given below [Fig. 12]. The wiring diagram was designed by using the fritzing software. As it is shown everything is connected either to Arduino or to the breadboard and then to Arduino with wires. The prototype device is shown in [Fig. 13].

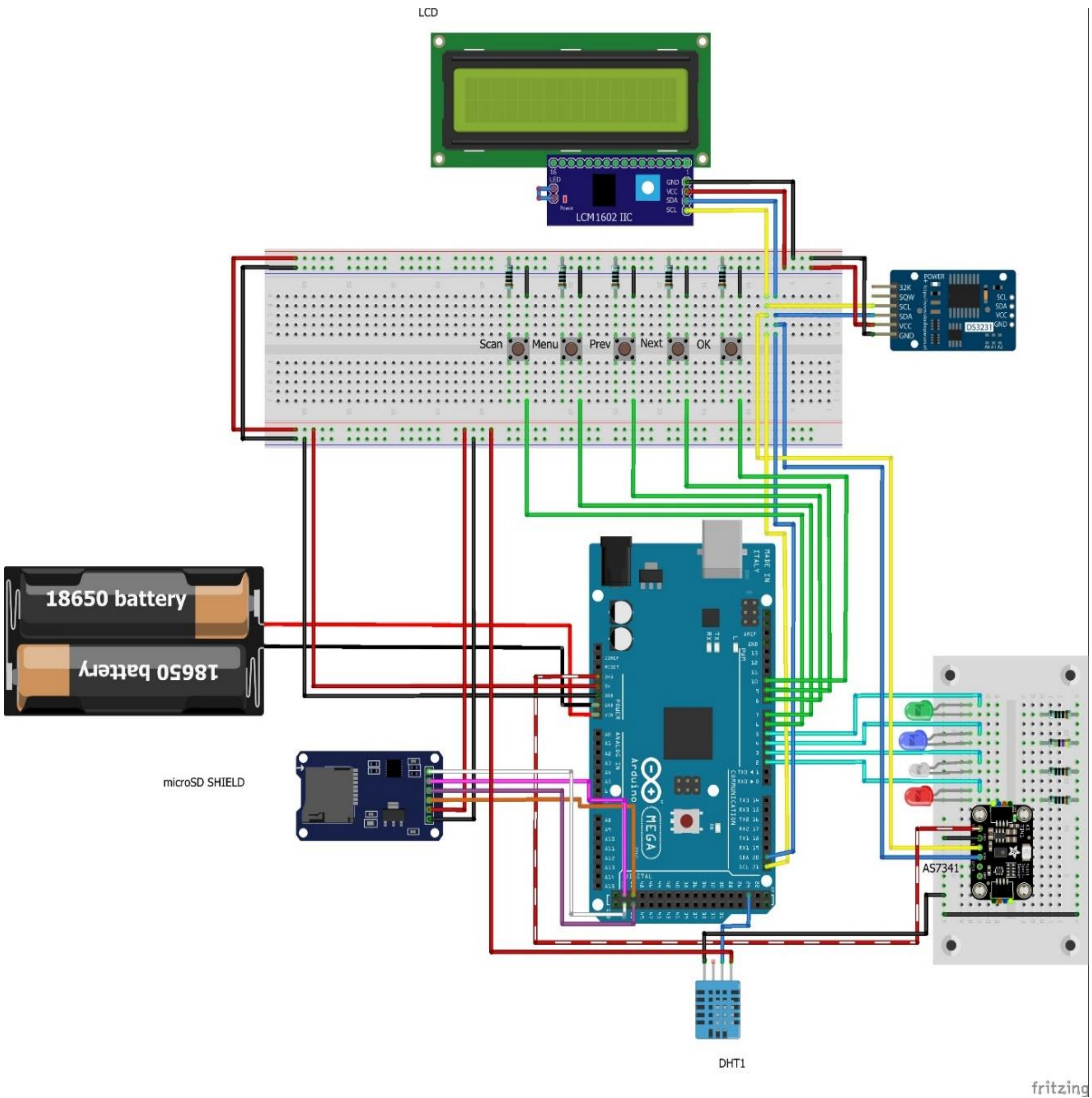

Fig. 12 Wiring diagram of the hardware (Fritzing)

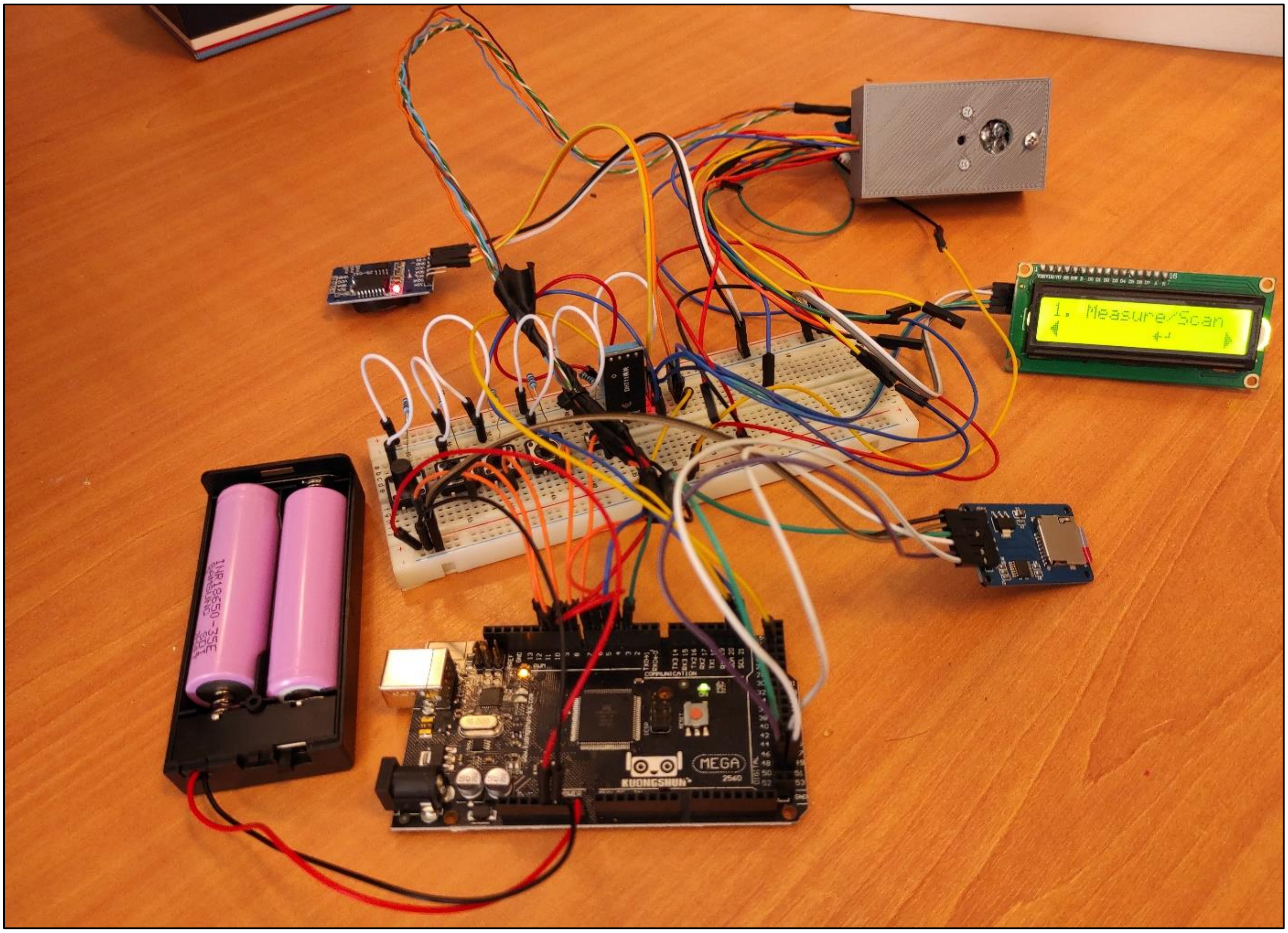

Fig. 13 The prototype model of the device.

For the device to work, the LEDs must emit at the certain wavelengths as we mentioned before, and measure the light reflected from the leaf using the AS7341 sensor [Fig. 14]. To achieve this the LEDs and the sensor must be close enough but not too close at the same time, so we have to find the perfect distance. The angle also must be wide enough, so the reflected light will be able to reach the sensor's photodiodes. Thinking of that we concluded that those two parts must be encapsulated somehow in order to be stable and solid. This will allow us to take more credible measures.

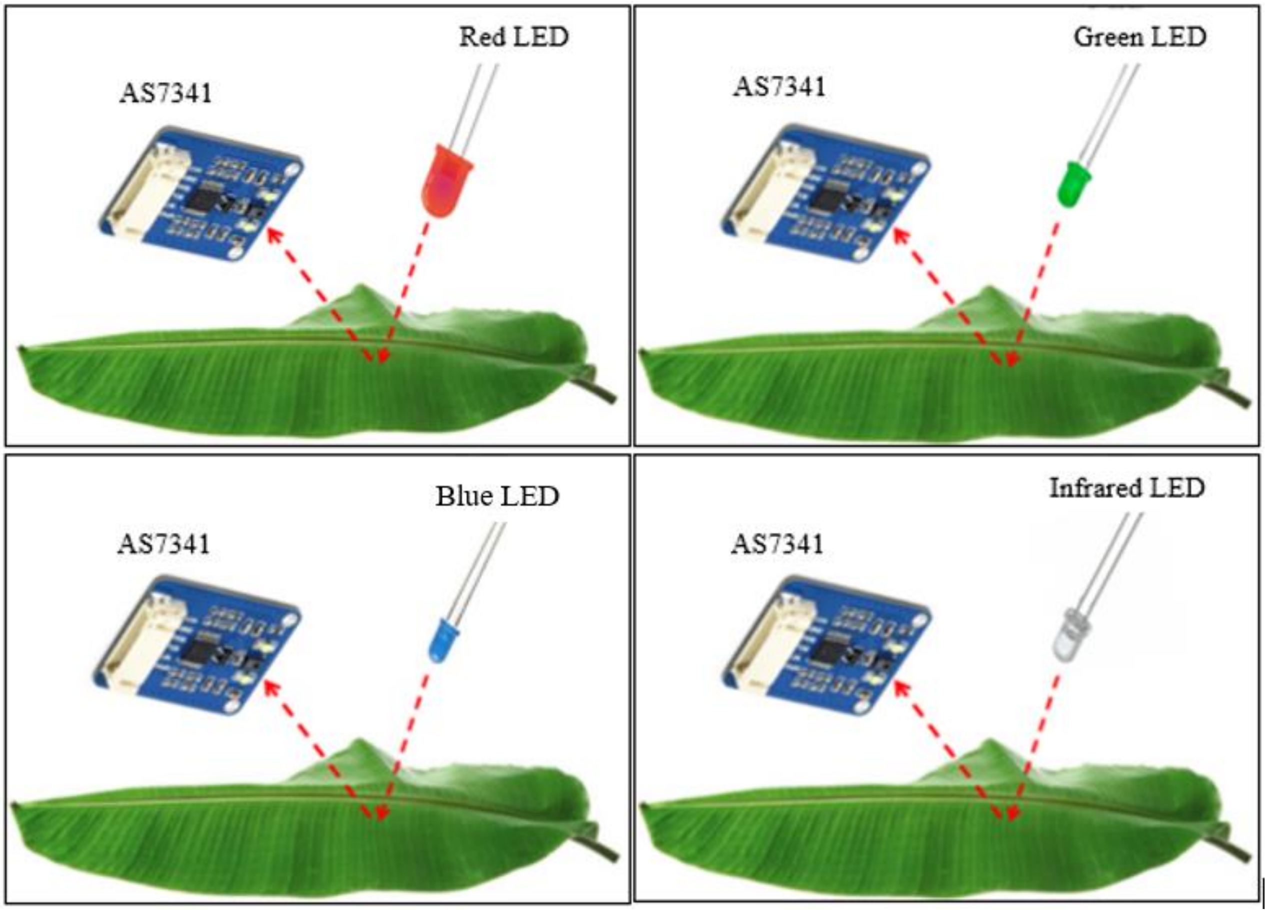

Fig. 14 Plant's leaf reflectance measure for RGB and IR LEDs.

As we aforementioned measuring the leaves and plants with the device, as it is now, is inconvenient. For that reason, we printed a 3D box, which encapsulates the AS7341 sensor and our LEDs [Fig. 15]. To make this construction stable and solid, we placed the LEDs with their resistors on a printed circuit board (PCB) [Fig. 16]. This way, we are guaranteed that we can move around only this part of the device to get values from different leaves and not the whole device. This way we are also reducing the ambient light interface. The 3D box encapsulates the LEDs, and the sensor is $70_{mm}x40_{mm}x23_{mm}$. The surrounding walls are 2_{mm} thick. The distance between the sensor's 4x4-photodiode array and the LEDs is almost 1cm.

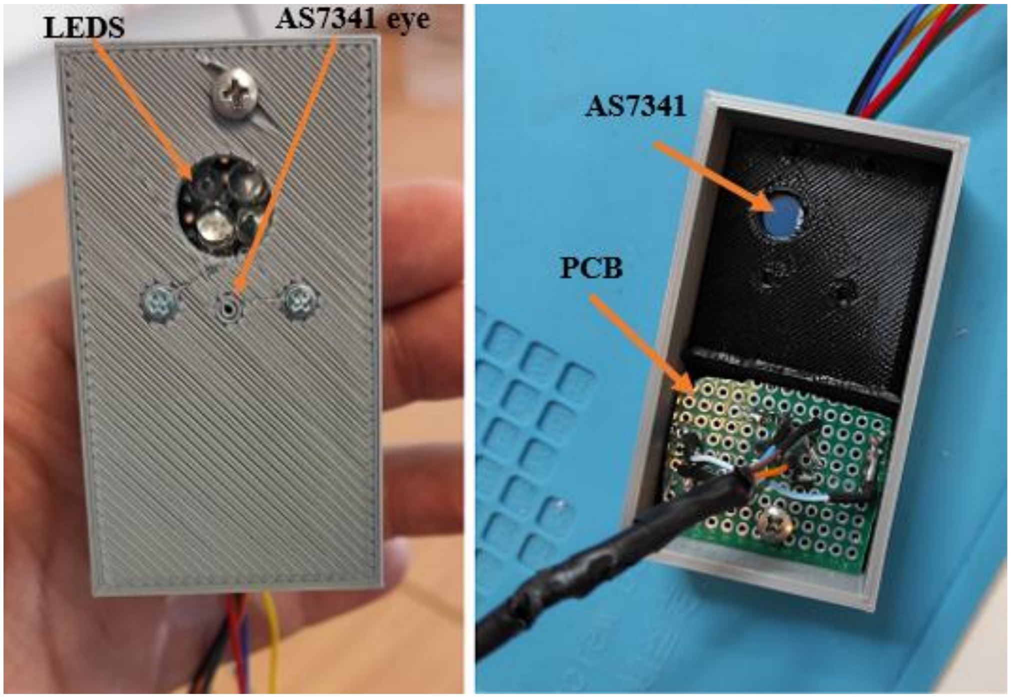

Fig. 15 3D printed box for our LEDs and AS7341

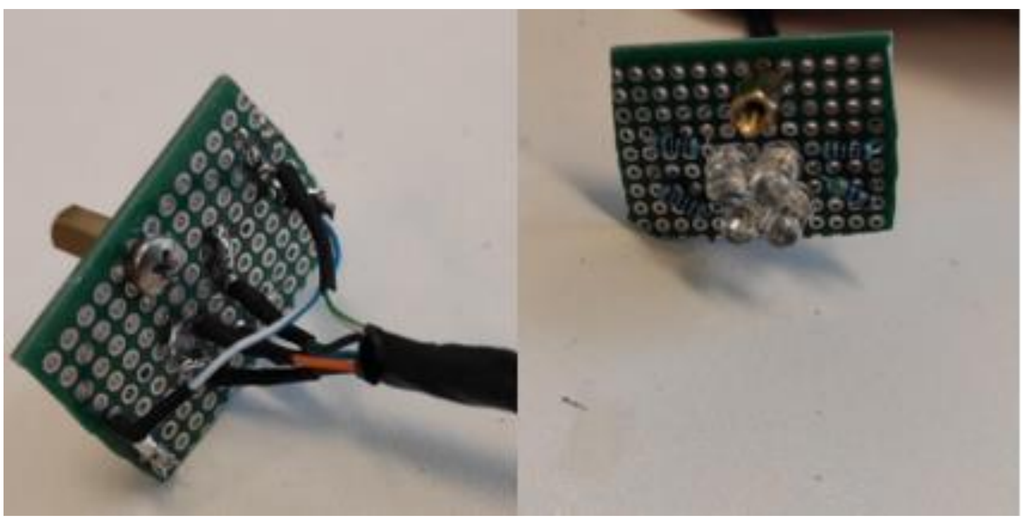

Fig. 16 LED PCB

Down below in [Fig. 17] we can see the final construction of the sensor and LEDs encapsulated in the box, and how we will measure using this construction.

Fig. 17 Blue, Green, Red, and IR measuring.

3.2 Principle of menu operation

When the device is switched on it shows a welcome message on the LCD. While doing that it checks that the modules are connected and work properly. In case something is not connected or has any kind of problem it prints the corresponding error message on the LCD. Next, our device goes to menu mode. To navigate the menu, we can press the menu button. This way we can go through the menu circularly. The menu has six options. [Fig. 18]. Each menu option does a particular task and can be activated by pressing the OK button. Only for the measure option do we have a specific button which is the Scan button. To keep the menu and operations plain and simple we will achieve a friendly-user device.

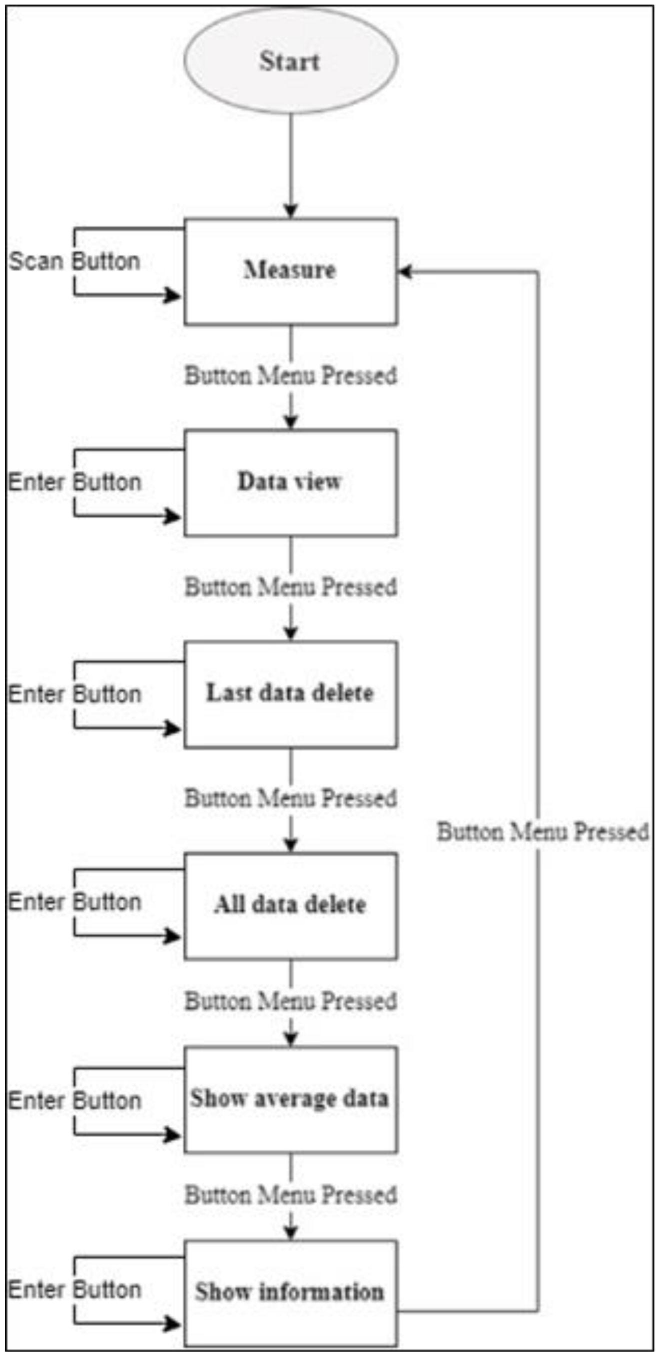

Fig. 18 Menu flow

3.2.1 Scan Mode

While we are in the measure option and by pressing the scan button, we initiate the measurement process. The measurement process includes the blinking of the LEDs for a certain amount of time and at the same time, our spectral sensor measures the reflectance values of the sample, for each LED activated. Afterward, we store those values in a matrix and we use them to calculate all the vegetation indicators that are needed. Then, we open or create a file on an SD card with the name “measures. scv.” This file will be used to write and read data.

First, we write the measurement ID in the file. Afterward, we write the vegetation index value followed by time, date, temperature, and humidity. In the end, we write the special character ‘\n’ to change the line in the file. This way we can separate the values. We also use padding in the values of ID and vegetation index -for vegetation indices we will also set the rule of keeping the two decimal numbers only- to keep the data in a certain form and size.

Additionally, while we are in the measure option in the menu and by pressing the ok button, we change the vegetation index the device measures. The default vegetation index is NDVI and we also added EVI, $NDVI_g$, $NDVI_b$, Simple Ratio, and IPVI.

3.2.2 View Data Mode

In this mode, we can see the data we have measured. Pressing the OK button while we are in this option, we can see the data so far. We can move to the next value or the previous value using the prev and next buttons. If we want to terminate this process, we press the menu button to go back to the menu. After reading the last value and pressing next the viewing mode is terminated and goes back to the menu. The data we show consists of the ID of the measurement, the NDVI value, and the time that the measurement is taken. In chapter 4.2, we will see more details about this menu function.

3.2.3 Last Data Delete

This mode allows us to delete the last measurement that we obtained. The reason for this function is that sometimes the user might move from the default distance or angle. Something might also go wrong in the measurement, and we do not wish to destroy the dataset.

3.2.4 All Data Delete

This mode, as expected, is used to delete the whole file on the SD card. It has a warning question before deleting everything. If we press yes, the whole file is deleted, and the ID goes back to zero. If the delete was successful, we see the proper message on the LCD and go back to the menu.

3.2.5 Show Average

Another useful process is showing average data. In this way, we can gather measurements of a tree's leaves and then see the average NDVI or the average of another VI and conclude conclusions about the plant's health condition. The average data is shown for 2 seconds and then the device goes to menu mode again.

3.2.6 Info

To end with the last option in our menu is the info menu. In this menu, while navigating back and forth with the prev and next buttons we can see the firmware details, manufacturer's details, current time and date, and internal temperature.

4. Implementation

In this chapter, we will see all the information regarding our implementation. It is separated into two sections hardware and software. Firstly, we will analyze the hardware implementation which includes all the hardware parts of our device. We will describe in detail how they work and their electronic features. Secondly, we will analyze the software parts of our device and how they are structured.

4.1 Hardware Implementation

In this section, the hardware parts of our device will be explained. We will also, outline their electronic characteristics and more details about their principles of operation. The basic idea was to develop a device with a low cost, and low power consumption that uses the minimum possible resources of hardware, power, and time. Of course, there is always an underlying trade-off between these factors.

4.1.1 Arduino Mega 2560 Rev3

Arduino is open-source software and hardware, initiative, and user community that develops and produces single-board microcontrollers and microcontroller kits for creating digital devices. The software is released under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL),[1] allowing anybody to distribute the program and produce Arduino boards. Its hardware components are licensed under the CC BY-SA license. Commercial Arduino boards are available from the official website or accredited distributors.

Different kinds of microprocessors and controllers are used in Arduino board models. The boards have a variety of extension boards (called "shields"), breadboards (for prototyping), and other circuits that can be interfaced to the sets of digital and analog input/output (I/O) pins on the boards. The boards have serial communications interfaces, some of which support USB (Universal Serial Bus), which are also used to load programs. The C and C++ programming languages, as well as a standard API known as the Arduino language, which was modeled after the Processing language and used with a modified version of the Processing IDE, can be used to program the microcontrollers. The Arduino project offers an integrated development environment (IDE).

The Arduino project started in 2005 as a teaching tool for students at the Interaction Design Institute in Ivrea, Italy, intending to give amateurs and experts alike a simple, low-cost means to build gadgets that use sensors and actuators to interact with their surroundings. Simple robots, thermostats, and motion detectors are typical examples of the type of equipment designed for beginning enthusiasts.

A sketch is a program written with the Arduino IDE. Sketches are saved on the development computer as text files with the file extension.ino. Arduino Software (IDE) pre-1.0 saved sketches with the extension.pde.

A minimal Arduino C/C++ program consists of only two functions:

setup(): This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch. It is analogous to the function main().

loop(): After setup() function exits (ends), the loop() function is executed repeatedly in the main program. It controls the board until the board is powered off or reset. It is analogous to the function while(1).

There are many Official Arduino boards such as Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Nano, etc. In our device, we will use the Arduino Mega 2560 Rev3. We chose this microcontroller because it is a powerful one with many I/O pins. The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 processor.

Arduino Mega 2560 [Fig. 19] comes with a clock speed of 16 MHz. The operating voltage is defined at 5V. As input voltage, it is recommended to provide within the range of 7-12V (with a tolerance up to 20V). Regarding pinout Arduino Mega has fifty-four digital I/O pins, fifteen of which can provide Pulse Width Modulation (PWM) output [Fig. 20]. On the other hand, it has sixteen analog input pins. As output voltage, Arduino supplies both 5V and 3.3V. Each I/O pin provides a direct current of 20mA (option to get DC of 50mA from 3.3V pin).

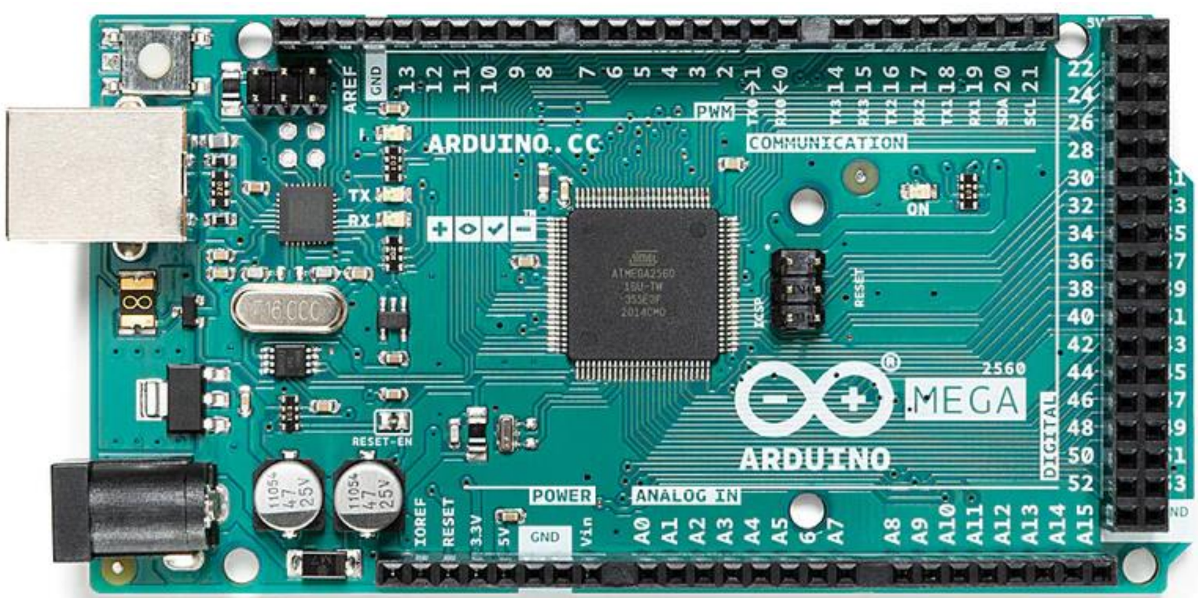

Fig. 19 Arduino Mega 2560

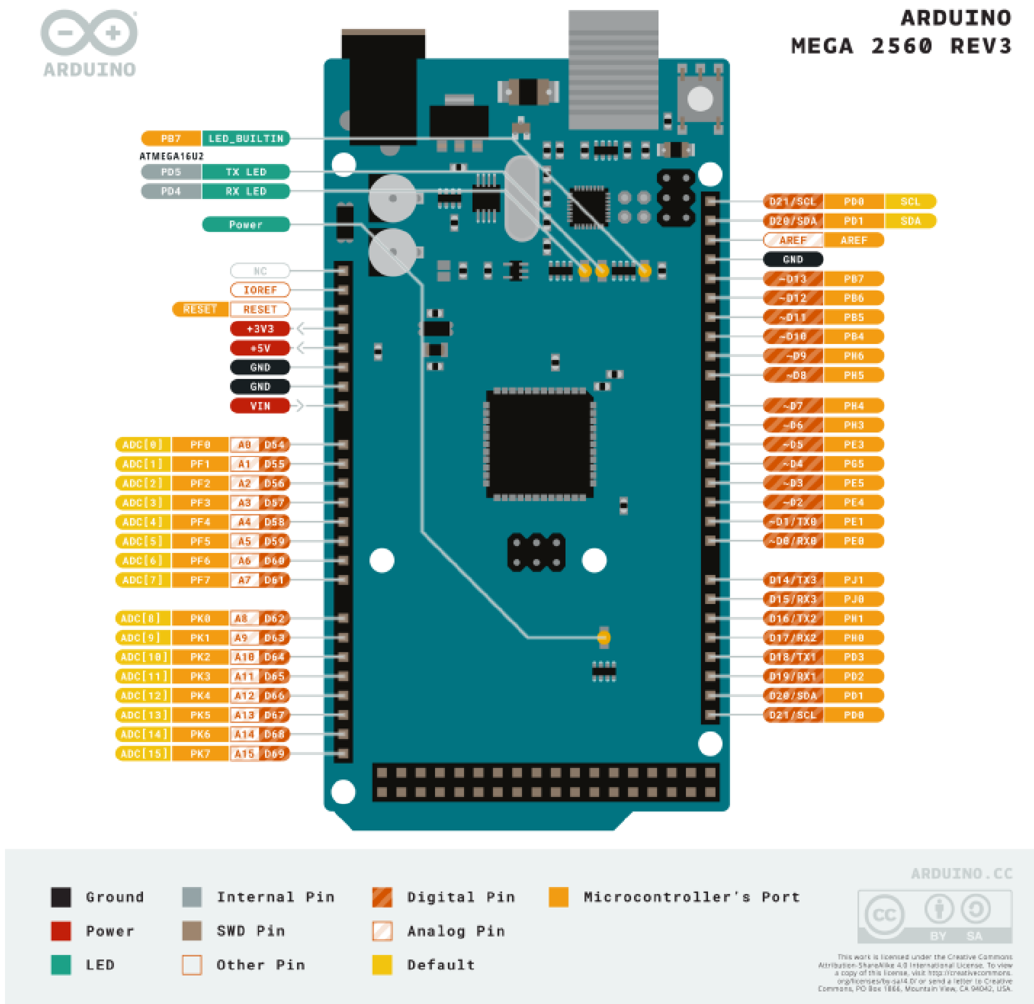

Fig. 20 Arduino pinout source: ***https://docs.arduino.cc/hardware/mega-2560***

Memory is divided to flash memory, SRAM, and EEPROM. Flash memory size is 256KB of which 8KB are used by the bootloader. In flash memory -program space- the Arduino sketch is stored. SRAM (static random-access memory) size is 8KB and it is used by sketch to create and manipulate variables when it runs. Finally, EEPROM size is 4KB and it is used so that programmers can preserve long-term data.

The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. It also supports I2C and SPI. Concerning size, it has 101.52 mm in length and 53.3 mm in width.

In conclusion, Arduino Mega 2560 is a powerful and fast microcontroller. It has a high DC and voltage output that we need for our LEDs. For those reasons, we chose this microcontroller; its only drawback is its size.

4.1.2 DS3231 Real Time Clock

The DS3231 [Fig. 21] is a temperature-compensated crystal oscillator (TCXO) and integrated crystal that functions as a low-cost, incredibly accurate I2C real-time clock (RTC). It has a battery input so it can maintain the precise time even though it may be disconnected from the main power source. The device's long-term precision is increased, and the production line's component count is decreased, thanks to the integrated oscillator. The battery we use is CR2032 which depending on the manufacturer's requirements, has a voltage of 3 volts and a capacity of up to 240 mAh.

Fig. 21 Real-Time Clock DS3231

DS3231 can keep seconds, minutes, hours, days, dates, months, and years. The end date will be automatically changed, including adjustments for leap years, if there are fewer than 31 days in the month. The clock may be set to work in either a 24-hour style or a 12-hour version with bands for AM and PM. It also provides two alarm clocks that may be customized, plus a calendar that can be programmed to produce square waves. I2C bidirectional bus is used to serially transport addresses and data.

Concerning size, it is 38mm long, 22mm wide, and 14mm high, a thing that makes it the most suitable selection for small devices. The module operates within the range of 3.3 to 5.5V. When the V_{cc} is 3.3V the active supply current is 200μA and the standby supply current is 110μA. On the other hand, when the V_{cc} is 5V the active supply current is 300μA and the standby supply current is 170μA. While we are using the battery's voltage, the active battery current is 70μA. Additionally, it comes with the memory chip: AT24C32 (storage capacity 32K). The device also integrates a very precise digital temperature sensor, through the I2C interface to access it (at the same time). This temperature sensor accuracy is $\pm 3^{o}$ C.

Finally, in terms of accuracy, the clock is off by about five minutes every month on average. The DS3231 is far more accurate, though, and can be as precise as a few minutes each year at most since it has an internal Temperature Compensated Crystal Oscillator (TCXO) that is not impacted by temperature. To communicate with this module, we use the RTClib.h library provided by Adafruit Industries.

4.1.3 Liquid Crystal Display (LCD)

The liquid-crystal display (LCD) is an optical device that uses polarizers and the light-modulating capabilities of liquid crystals. Those crystals do not emit light directly; instead, they create images in either color or monochrome utilizing a backlight or reflector. LCDs are capable to display images with low information content.

For each pixel, there is a layer of molecules that are commonly arranged between two transparent electrodes, which are commonly constructed of indium-tin-oxide (ITO), and two polarizing filters, whose axes of transmission are vertical to each other. If there is no liquid crystal between the polarizing filters, light passing through the first filter is blocked by the second polarizer. The orientation of the liquid crystal molecules is determined by the alignment of the electrodes before an electric field is applied. As a result, the polarization of the incident light rotates, giving the object a gray appearance. The central liquid crystal molecules in the layer are entirely untwisted if a high enough voltage is supplied, and the incident light's polarization is maintained as it passes through the liquid crystal layer. This light will then be mostly blocked because it is polarized perpendicular to the second filter, making the pixel look black. The amount of light flowing through each pixel can be altered by varying the voltage applied across the liquid crystal layer, which subsequently yields various shades of gray.

All the LCDs used for microcontrollers include their own controller which manages all their internal processes. Most of them are based on the popular Hitachi HD44780 controller, so it does not matter what size of monitor you use (2 x 16 or 4 x 20). We usually distinguish them by the 16-pin interface. The display has an optional backlight and can display 2 rows of 16 characters. Character Resolution is 5x8 points. The microcontroller receives commands and describes symbols on the LCD. Each pixel can light up individually, so we can generate characters within each grid [Fig 22].

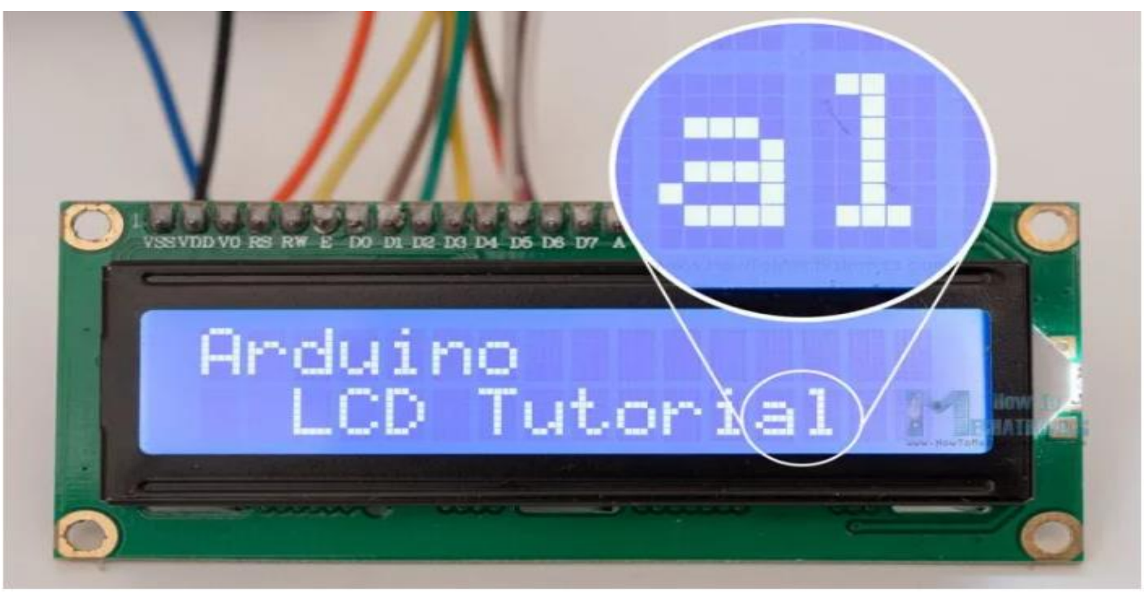

Fig. 22 LCD Character Resolution. Source: https://howtomechatronics.com/tutorials/arduino/lcd-tutorial/

The display can operate in 2-modes:

- 8-bit (when contacts from D0 to D7 are used for information exchange), and data is transferred in one cycle.
- 4-bit (only D4 - D7 contacts are used for communication). In this case, data is sent in two cycles: firstly, high-order 4th bits, then low-order 4th bit.

Pin	Pin Name	Pin Type	Pin Description
1	V_{SS} (ground)	Source Pin	This is a ground pin of LCD
2	V_{CC}	Source Pin	This is the supply voltage pin of the LCD
3	V_0, V_{EE}	Control Pin	Adjusts the contrast of the LCD.
4	RS (register select)	Control Pin	Toggles between Command/Data Register (0 for command mode or 1 for data mode)
5	RW (read/write)	Control Pin	Toggles LCD between Read/Write Operation (0 for writing or 1 for reading)
6	E (enable)	Control Pin	Must be held high to perform Read/Write Operation
7	Data bits (D0-D7)	Data/Command Pin	Pins are used to send commands or data to the LCD.
8	A (Anode)	LED Pin	This is the supply voltage pin for the backlight
9	K (cathode)	LED Pin	This is the ground pin for the backlight

Table 1 LCD Module Pinout

Qapass 1602A-1 (v1.2) will be used for our device. The display format of this module is 16 characters x 2 Lines. Its viewing direction is 6 o'clock. To work without any failures, we must provide it with a 5V power supply and tolerance up to 7V max input voltage. Regarding current consumption, the backlight on the LCD draws about 18-20 mA (in most cases) while the LCD itself draws less than 1 mA.

The Qapass 1602A-1 comes with an integrated I2C Serial Interface Adapter Module for LCD [Fig 23]. Since the microcontroller has limited pin resources, controlling an LCD panel can be space and time-consuming. This module can solve this problem as we now only need to use four cables instead of sixteen. It belongs in the category of serial to parallel adapters, and it uses a PCF8574 chip. It provides two signal output pins, SDA and SCL, used for communication with the microcontroller.

The adapter also comes with a fixed potentiometer that helps us to adjust the contrast of the LCD, so we do not need to use an external one. To use this module LiquidCrystal I2C Display library is used. Library's author is Frank de Brabander, and its maintainer is Marco Schwartz. This library is easy to use, and it is very similar to the LiquidCrystal library.

Fig. 23 Qapass 1602-A with I2C module

4.1.4 Micro-SD card module

One of the most crucial components of every project is data storage. Depending on the type and the size of the data, there are various methods for storing it. One of the most useful storage options is both SD and micro-SD cards, which are found in products like cell devices, microcomputers, and others. It is particularly helpful for tasks that demand data logging. Some microcontrollers have an integrated SD card slot. Arduino Mega 2560 is not of them so we need to use a micro-SD module. The module communicates with Arduino via SPI protocol. The SPI communication protocol is used by several models from various providers, although they all operate similarly. By using the SD.h and SPI.h libraries the Arduino may create/read/write a file on an SD card. The libraries are provided by Arduino.

The micro-SD card module consists of six pins [Fig. 24]. We will use all of them. In Table 2 we will expound on the use of each pin.

Pin number	Pin Type	Pin Name	Pin Description
1	V_{cc}	Source Pin	This is the supply voltage pin.
2	GND	Source Pin	This is the ground pin.
3	MISO	Data Pin	Stands for Master in Slave Out. This is the SPI data out from the SD Card Module.
4	MOSI	Data Pin	Stands for Master Out Slave In. This is the input pin of the SD Card Module.
5	SCK	Control Pin	Stands for Serial Clock as the name implies it is the data synchronization pulse generated by the Arduino.
6	CS	Control Pin	Stands for Chip Select, the Arduino can control this pin to enable or disable the module.

Table 2 Micro-SD module pinout.

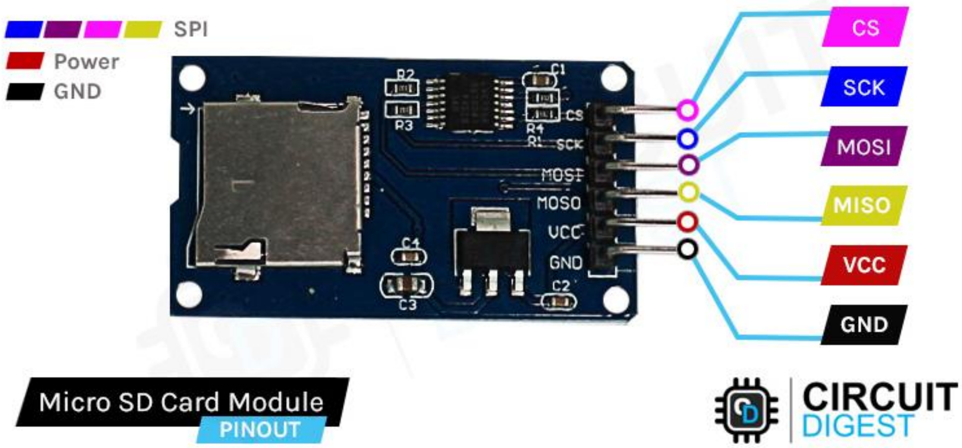

Fig. 24 Micro-SD module pinout. Source: https://circuitdigest.com/microcontroller-projects/interfacing-micro-sd-card-module-with-arduino

To utilize this module, we must correctly format the SD-card and then put it into the SD-card reader of the module. If we do not do this procedure, we will run into issues because the micro-SD reader module can only read FAT16 or FAT32 file systems. If the card is brand new, there is a good probability that it will be factory formatted. In any case, it is preferable to format the card to prevent problems from occurring while it is in use.

The micro-SD card module can work with both 5V and 3.3V input. That is because of the onboard voltage regulator that shifts the voltage from 5V to 3.3V [Fig. 25]. That is also why this module is capable to operate on 3.3V power. The level shifter IC is another crucial component because the SD card module only operates at 3.3V and has a maximum operating voltage of 3.6V. If the SD card is connected straight to 5V, the micro-SD module will be destroyed. Its

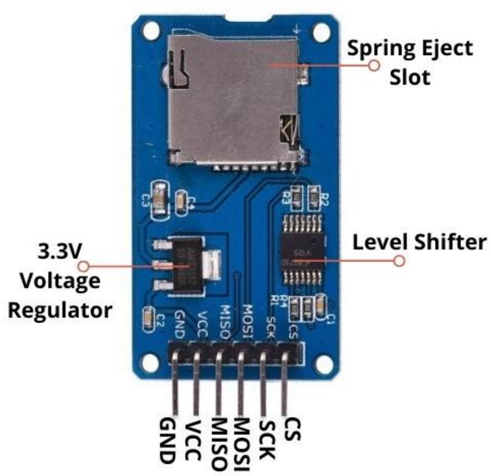

Fig. 25 Micro-SD module's Voltage regulator

The SD card that we will use for this module is SanDisk Ultra MicroSDHC 32GB Class 10 U1 A1. It is commonly used in microcontroller applications and with microcontrollers that have an integrated SD-card slot like the Raspberry PI. Its reading and writing speed are 98 MB/s.

4.1.5 DHT-11

Another basic component vegetation indicator measure like devices is temperature and humidity sensor. Acquiring temperature and humidity while we are measuring NDVI, or other indicators completes our dataset. Speaking about machine learning temperature and humidity will be a heavyweight feature.

The DHT sensors are composed of two parts. The first one is a thermistor. A thermal resistor, or thermistor, is a resistor whose resistance varies with temperature. Practically speaking, all resistors are thermistors, because they change slightly in resistance as a function of temperature, but the change is typically slight and challenging to measure.

The resistance of thermistors is designed to change significantly with temperature, often by more than 100 ohms per degree! The second one is a capacitive humidity sensor. There is also an onboard chip [Fig. 26] that converts the analog signal to digital. The sensor has 4 pinouts. From left to right we have the V_{cc} pin, the serial data pin, a not used pin, and the ground pin which are connected to the module.

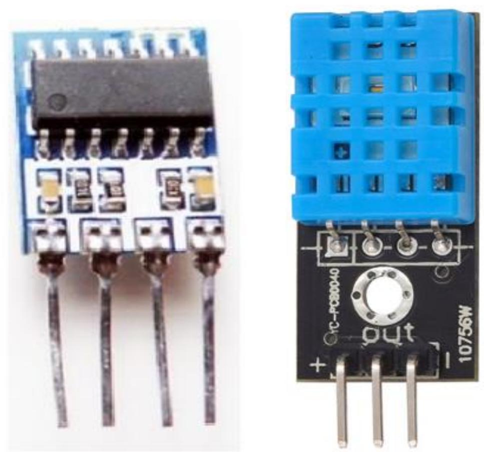

Fig. 26 DHT11 Internal circuit, pinout, and module's pinout

Concerning the module, it has three pinouts. The first one is V_{cc}, the second is the data pin and the last one is the ground pin. The module comes calibrated and ready to use via the typical single-wire interface. Regarding its specifications, the module operates in the range of 3.3V to 5.5V, the operating current is 0.5-2.5mA while measuring and 100-150uA in standby mode depending on the V_{in}. Its sampling period is 1 second so we can take one measure every second. It can measure temperature from 0^{o}C up to 50^{o}C with ±2^{o}C accuracy and humidity from 20% up to 90% with ±5% accuracy. Resolution in both temperature and humidity is 1^{o}C and 1% respectively.

4.1.6 Buttons

Complete electrical circuits are necessary for operation. The numerous wires and parts must allow the electricity to pass through them without interruption. However, circuits that operate only when we want them to are more valuable than those that operate continuously. A switch operates in this manner. Some switches are tucked away inside of equipment, while others are out in the open for us to utilize. There are several common applications for the push button switch, including automobile stereos and elevators. Momentary and non-momentary are the two primary types.

A button or pushbutton as it is mentioned usually is a tiny, enclosed mechanism that, when pressed, completes an electrical circuit. When it is pressed -high state-, a tiny metallic spring interior to the circuit contacts two wires, which allows electricity to flow through it. When it is not pressed -low state-, the spring retracts, and the connection is broken. The main body of the button is made of non-conducting plastic. They can be split into categories that are normally on and off.

In our device, we will use normally off buttons. Using a normally off switch, there is no connection until the button is pressed. Buttons in general have four pins, which are internally connected in pairs. Regarding this, we only need to use two pins -which are not internally connected- to connect it with Arduino.

We will use a pull-up resistor to connect it with Arduino [Fig. 27]. When the button is not pressed, a pull-up resistor causes the input pin to be read as high. To put it in another way, the input pin reads near to V_{CC}, because a small current is passing between V_{CC} and the input pin. The input pin is connected straight to the ground only when the button is pressed. The input pin reads as low because the current goes via the resistor and then to the ground.

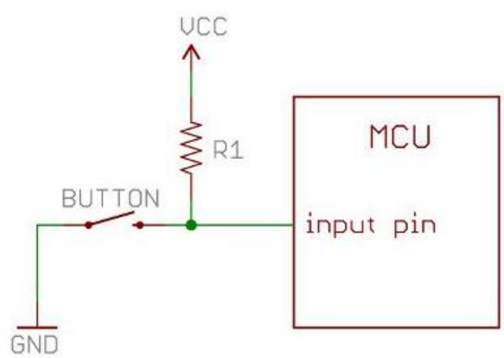

Fig. 27 Pull-up Resistor with button

Due to mechanical and physical problems, pushbuttons frequently produce erroneous open or close transitions when pressed; these transitions may be interpreted as several presses in a very short period, misleading the application. This problem is known as bouncing. To solve this problem, we will use software debounce techniques.

4.1.7 LEDS

A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it. Recombining electrons and electron holes in the semiconductor results in the release of energy in the form of photons. The energy needed for electrons to pass through the semiconductor's band gap determines the hue of the light, which corresponds to the energy of the photons.

The LEDs are one of the most important parts of our device, so they are carefully considered and selected. We will use four LEDs as we said before in certain wavelengths. We will use four LEDs: IR, red, green, and blue. We chose LEDs with high intensity because we are measuring through reflectance. To properly use them, we chose resistors with ±1% accuracy. resistors were chosen with the help of Ohm's law $R = \frac{Vd-Vf}{I}$. We need the value of I_f to be close to 20mA to achieve the nominal values which are stated in their datasheets. To measure their peak emission wavelength, we used the HR2000+ spectrometer.

IR LED

Concerning the IR LED, we used LL-503SIRC2E-1BD made by the LUCKYLIGHT company [56]. It has a viewing angle of 30^o degrees. It has a peak emission wavelength of 837nm [Fig. 28], and forward voltage of 1.47V. We used a 100Ω with a 1% tolerance resistor, so we have approximately 26mA current flowing through the LED.

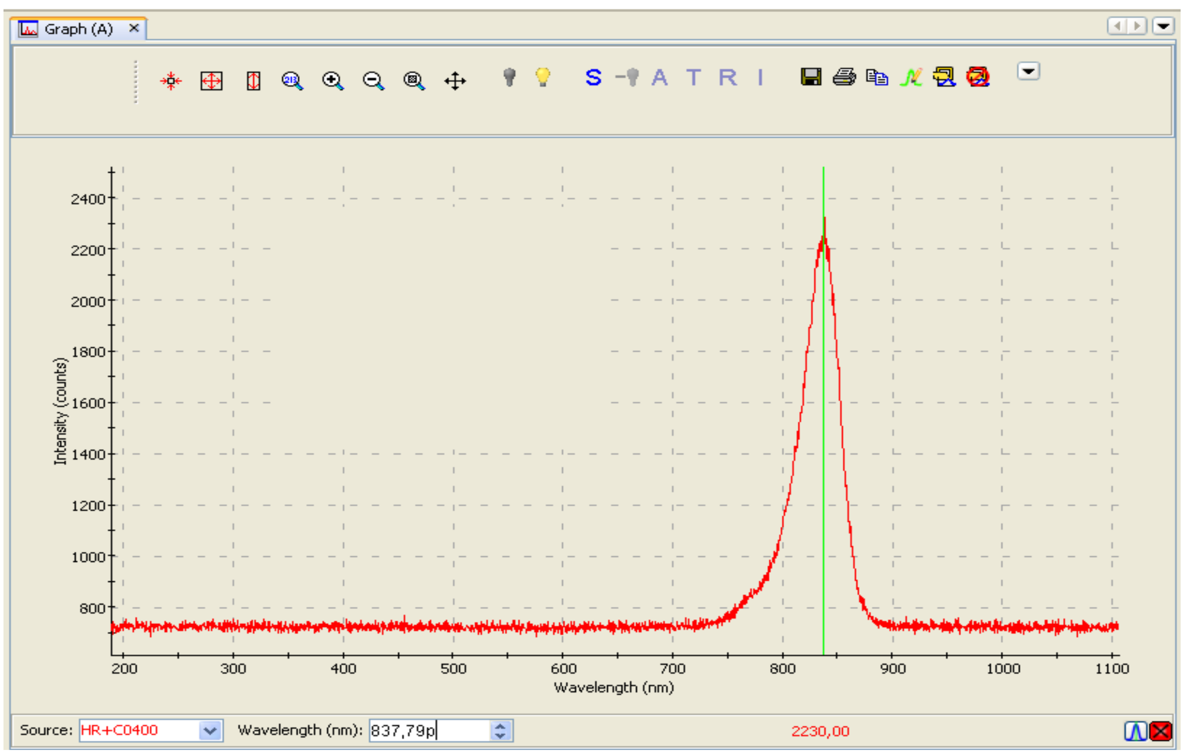

Fig. 28 IR LED's wavelength

RED LED

Regarding the red LED, we used L-53SRC-J4 made by the Kingbright company [57]. It has a viewing angle of 30^{o} degrees. It has a peak emission wavelength of 658nm [Fig. 29], and forward voltage of 1.95V. We used a 100Ω with a 1% tolerance resistor, so we have approximately 23.5mA current flowing through the LED.

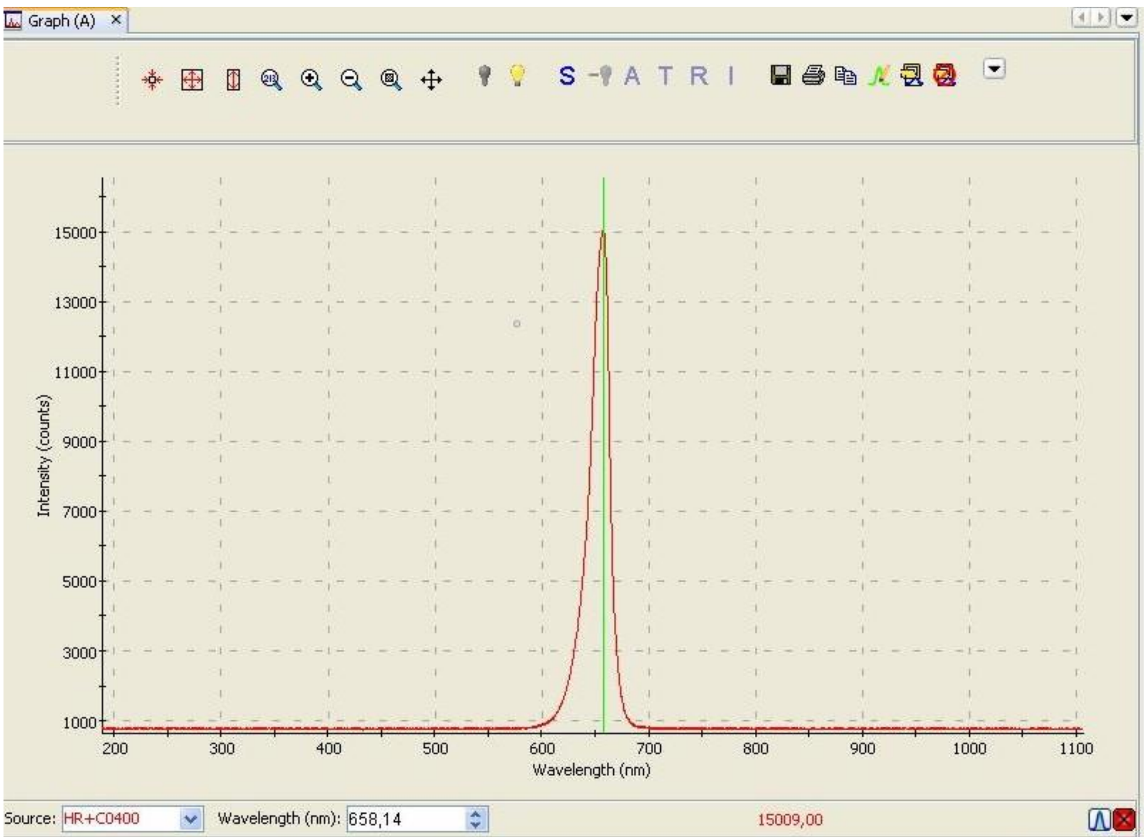

Fig. 29 Red LED's wavelength

GREEN LED

Regarding the green LED, we used LL-504PGC2E-G5-1AC made by the LUCKYLIGHT company [58]. It has a viewing angle of 30^o degrees. It has a peak emission wavelength of 523nm [Fig. 30], and forward voltage of 2.63V. We used a 100Ω with a 1% tolerance resistor, so we have approximately 18mA current flowing through the LED.

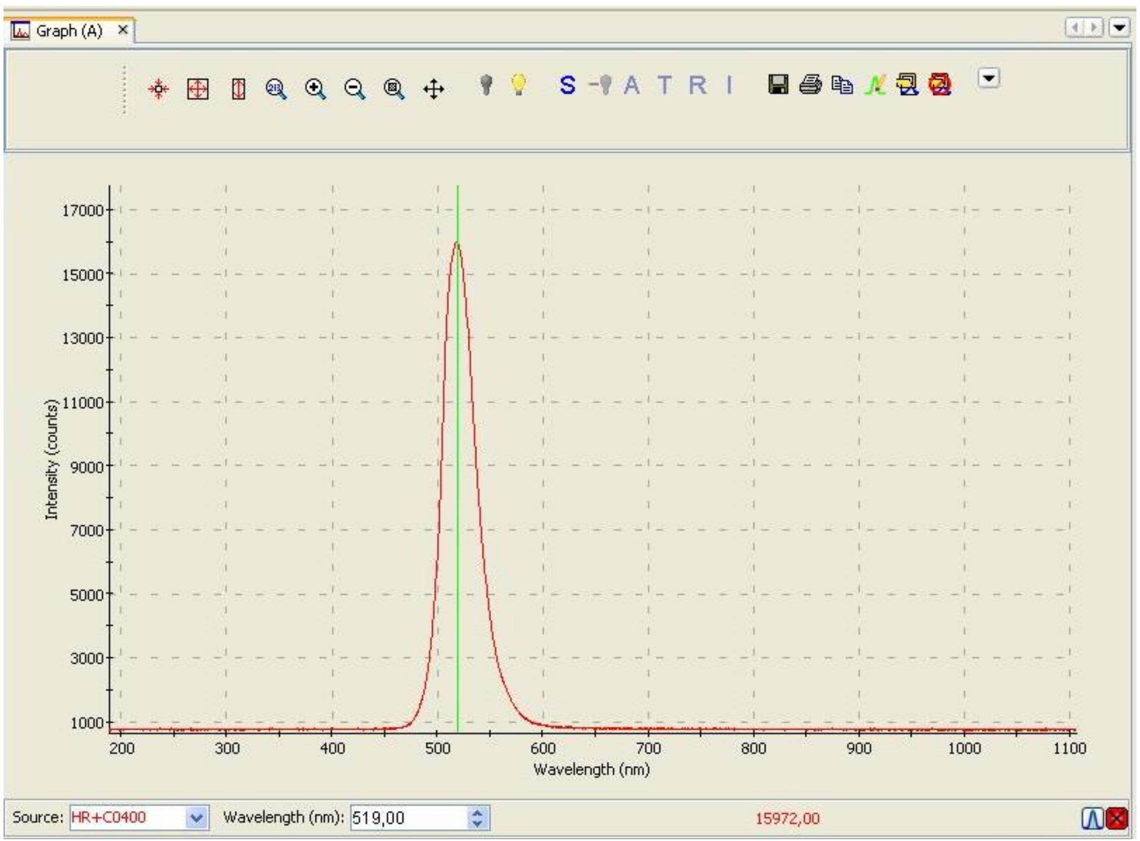

Fig. 30 Green LED's wavelength

BLUE LED

Regarding the blue LED, we used LL-504BC2E-B4-1GC made by the LUCKYLIGHT company [59]. It has a viewing angle of 15^o degrees. It has a peak emission wavelength of 462nm [Fig. 31], and forward voltage of 3.12V. Because of the high V_f in this case we used a 47Ω with a 1% tolerance resistor, so we have approximately 24.3mA current flowing through the LED.

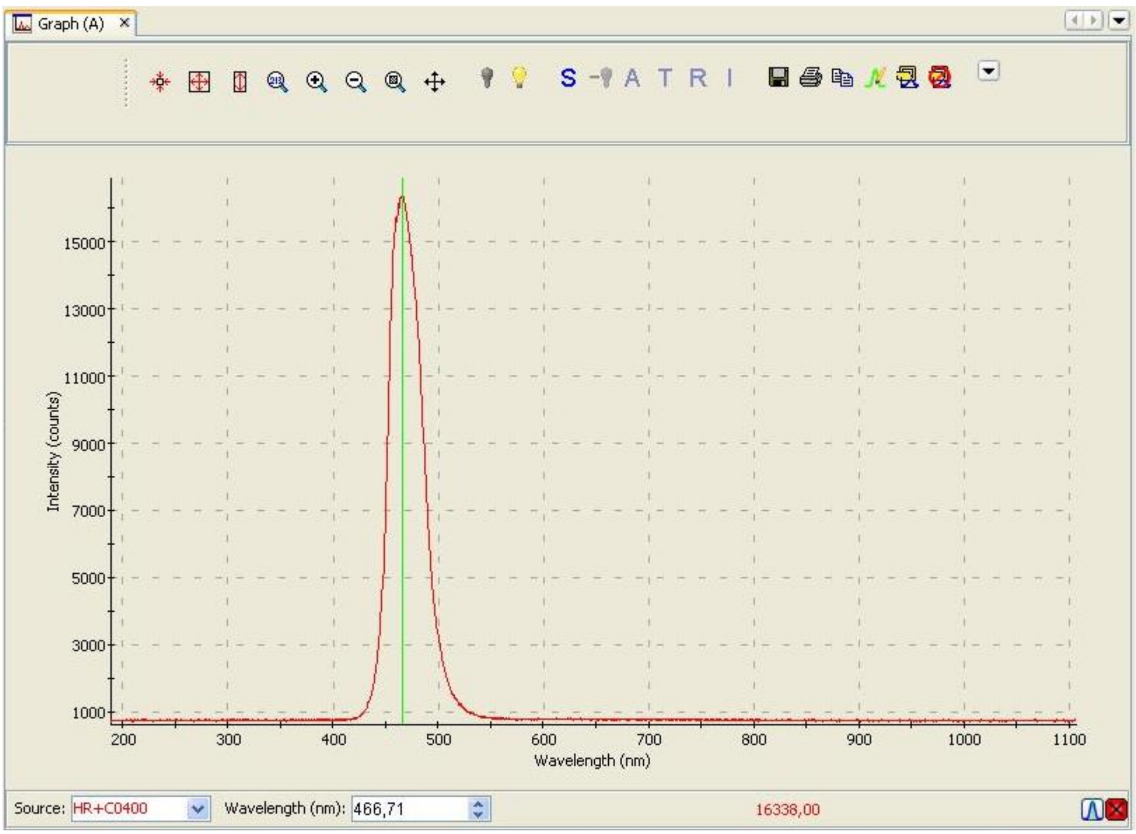

Fig. 31 Blue LED's wavelength

4.1.8 18650 Battery

An 18650 battery is a lithium-ion battery (Li-ion battery). A lithium-ion battery, often known as a Li-ion battery, is a kind of rechargeable battery made up of cells in which lithium ions travel from the negative electrode to the positive electrode through an electrolyte during discharge and back again during charging. The positive electrode in lithium-ion batteries is made of an intercalated lithium compound, whereas the negative electrode is commonly made of graphite. Apart from LFP cells, lithium-ion battery provides a high energy density, little memory effect, and a low rate of self-discharge. However, because they carry combustible electrolytes and can cause explosions and flames if broken or charged improperly, they can be a safety issue.

The name comes from the battery's precise dimensions of 18 mm by 65 mm. That is greater than just an AA battery for comparison. The 18650 batteries have a voltage of 3.6V and have capacitance between 2600mAh and 3500mAh (milliampere per hour). Because of their dependability, lengthy run periods, and capacity to be recharged numerous times, these batteries are utilized in torches, laptops, and many gadgets. They are considered high-drain batteries. This implies that the battery is made to produce high output voltage and current to meet the power requirements of the portable device it is being used with.

For our device, we will use Samsung INR18650-35E [Fig. 32]. This model has a nominal voltage of 3.6V and a charging voltage of 4.2V. Its minimum capacity is 3,350mAh. The charging time is about 4 hours. The manufacturer suggests not discharging deeper than 2.5V. The specific model has a high continuous discharge current of 8A (8000mA).

We will use two batteries connected in series so their output voltage will be summed. This will supply the Arduino with a voltage of 7.2V approximately which is enough to operate.

Fig. 32 Samsung 18650-35E battery

4.1.9 AS7341

AS7341 sensor is an 11-channel spectrometer mostly used for spectral identification and color matching applications used in mobile devices. The spectral response is defined in the wavelengths from approximately 350nm to 1000nm. 6 channels can be processed in parallel by independent ADCs while the other channels are accessible via a multiplexer. 8 optical channels cover the visible spectrum (typically 380-750m), one channel can be used to measure near-infrared light (IR) and one channel is a photodiode without filter ("clear channel"). The device also integrates a dedicated channel to detect 50Hz or 60Hz ambient light flicker. The flicker detection engine can also buffer data for calculating other flicker frequencies externally. The NIR channel in combination with the other VIS channel may provide information on surrounding ambient light conditions (light source detection). The device can also be synchronized to external signals via pin GPIO. Each of the filter pairs can be mapped to one of the six internal ADCs (CH0 – CH5).

AS7341 module -from Waveshare Industries [Fig. 33]- can operate at the voltage of both 3.3V and 5V. Regarding current consumption, it is 20mA. This module uses AS7341-DLGM as the core. AS7341-DLGM needs a supply voltage between 1.7-2V (typical 1.8V). Current consumption is 210-300μA while it is in active mode, 35-60 μA while it is in idle mode, and 0.7-5μA while it is in sleep mode. AS7341 uses the I2C protocol to communicate with other devices. The I2C device address of AS7341 is 0X39.

Fig. 33 AS7341 module from Waveshare

The sensor consists of a 4x4-photodiode array [Fig. 34]. There are two photodiodes above and below the photodiode array that perform specific tasks including flicker detection (FLICKER) and near-infrared response (NIR). At the right and left base, corners are a clear channel (C) which is a photodiode without any filters. Channels pairs F1-F8 measure the spectrum of visible range.

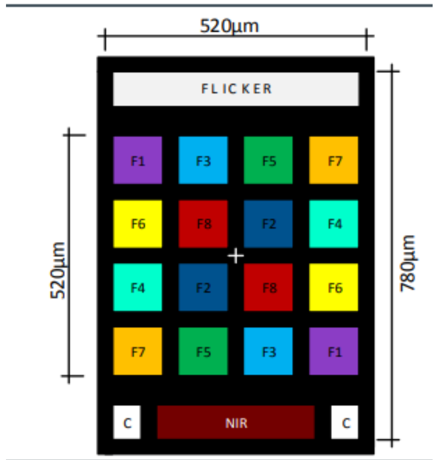

Fig. 34 AS7341 4x4-photodiode array. Source datasheet.

In Table 3 characteristics of channels F1-F8 are analyzed:

Channel	Parameter	Min	Typical	Max	Unit
$R_{e\text{-}F1}$	Center wavelength (λ_ρ)	400	410	420	nm
	Full width half maximum (FWHM)		29		nm
$R_{e\text{-}F2}$	Center wavelength (λ_ρ)	430	440	450	nm
	Full width half maximum (FWHM)		33		nm
$R_{e\text{-}F3}$	Center wavelength (λ_ρ)	460	470	480	nm
	Full width half maximum (FWHM)		36		nm
$R_{e\text{-}F4}$	Center wavelength (λ_ρ)	500	510	520	nm
	Full width half maximum (FWHM)		40		nm
$R_{e\text{-}F5}$	Center wavelength (λ_ρ)	540	550	560	nm
	Full width half maximum (FWHM)		42		nm
$R_{e\text{-}F6}$	Center wavelength (λ_ρ)	573	583	593	nm
	Full width half maximum (FWHM)		44		nm
$R_{e\text{-}F7}$	Center wavelength (λ_ρ)	610	620	630	nm
	Full width half maximum (FWHM)		53		nm
$R_{e\text{-}F8}$	Center wavelength (λ_ρ)	660	670	680	nm
	Full width half maximum (FWHM)		60		nm

Table 3 Channels F1-F8.

In [Fig. 35] the normalized spectral responsivity is shown.

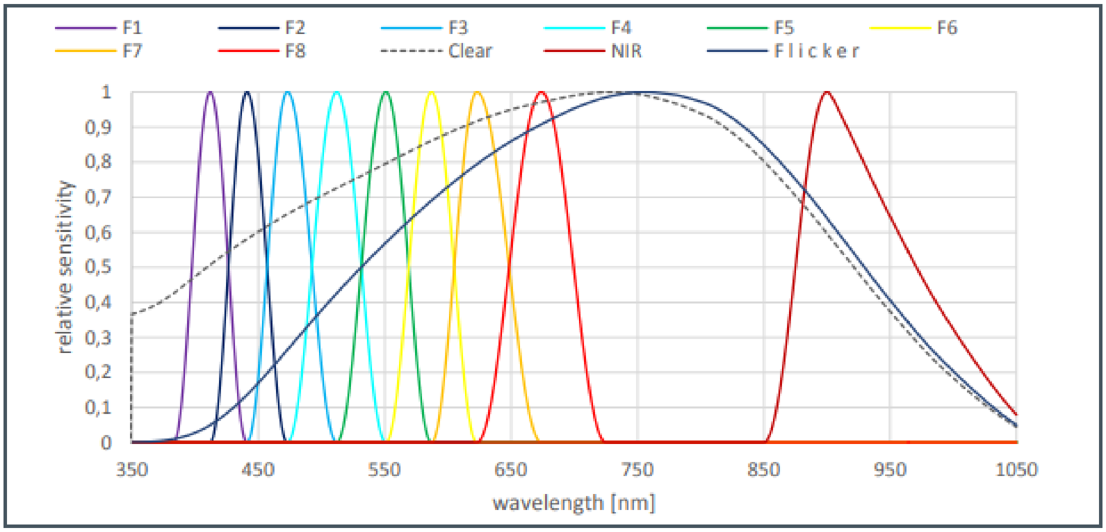

Fig. 35 Normalized Spectral Responsivity. Source datasheet.

The system includes a dedicated 16-bit light-to-frequency converter and 6 independent optical channels. With the serial interface, the 6 channels' gain and integration time can be changed. To automatically set a gap between two subsequent spectral measurements and cut down on overall power consumption, a wait time can be programmed. By attaching them to one of the internal ADCs through a multiplexer (SMUX), you can access the additional channels that are available [Fig. 36].

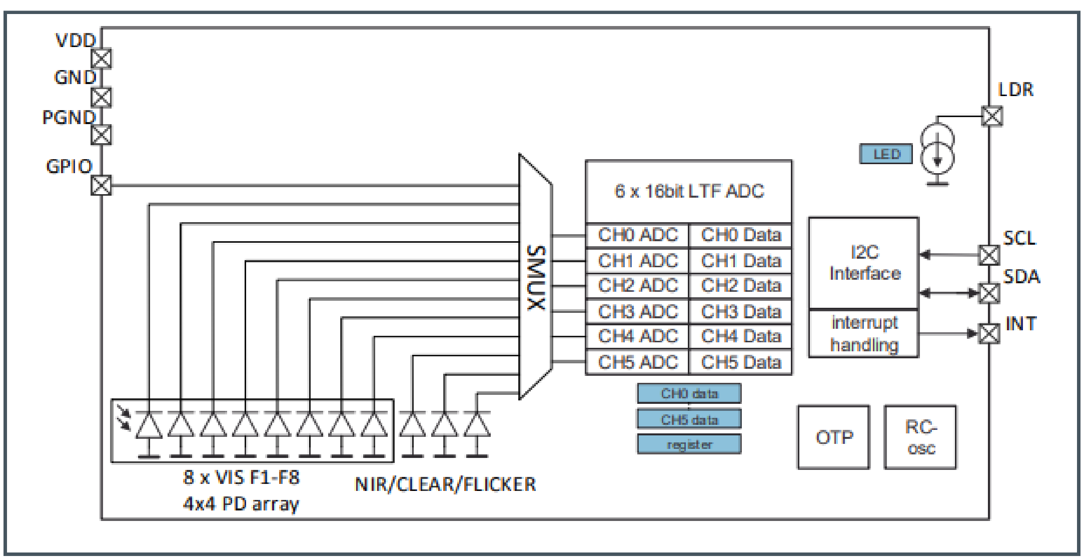

Fig. 36 Block diagram. Source datasheet.

4.1.10 Serial Peripheral Interface (SPI) protocol

The Serial Peripheral Interface (SPI) is a synchronous serial communication interface specification used for short-distance communication, primarily in embedded systems. Motorola created the interface, which has now grown to be accepted as the industry standard. Although some devices support switching between master and slave roles dynamically dependent on an external (SS) pin, most SPI devices typically operate in full duplex mode with a single master. The frame for reading and writing is created by the master (controller) device. Individual chip select (CS), also known as slave select (SS) lines, allow for the selection of multiple slave devices. In contrast to three-, two-, and one-wire serial buses, SPI is occasionally referred to as a four-wire serial bus. The SPI bus specifies four logic signals:

1. SCLK: Serial Clock (output from master)
2. MOSI: Master Out Slave In (data output from master)
3. MISO: Master in Slave Out (data output from slave)
4. CS /SS: Chip/Slave Select (output from master to indicate that data is being sent)

The SPI bus can operate with a single master device and with one or several slave devices. The bus master sets up the clock at a frequency that the slave device can support, usually up to a few MHz, to start communication. The slave device is then chosen by the master using a logic level of "0" on the control signal. The master must wait at least that much before sending clock cycles if a waiting period is necessary, such as during analog-to-digital conversion. A full-duplex transmission of data happens on each SPI clock cycle. The master device sends a bit on the MOSI line and the slave reads it. On the other hand, the slave sends a bit on the MISO line and the master reads it. Even when just a one-directional data flow is planned, this sequence is kept.

Two shift registers [Fig. 37], one in the master and one in the slave, each with an assigned word size, such as eight bits, are typically used in transmissions. They are coupled in a virtual ring topology. Typically, the most important bit is sent out first when shifting data. Both the master and the slave move out slightly on the clock edge and output it to the counterpart on the transmission line. Each receiver samples a bit from the transmission line and sets it as the next least-significant bit of the shift register on the subsequent clock edge. The master and slave have switched register values after the register bits have been moved out and in. The shift registers are reloaded, and the operation is repeated if more data must be transferred. Transmission can go on for many clock cycles until everything is done, then the master deselects the slave and stops twiddling the clock signal. Every unused slave on the bus using its chip select line must ignore the input clock and MOSI line signals and must not drive any signal through the MISO line.

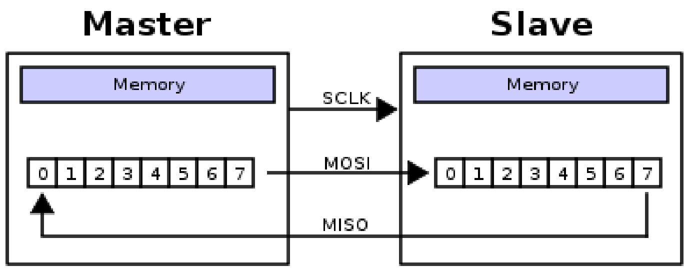

Fig. 37 A typical hardware setup using two shift registers to form an inter-chip circular buffer. Source Google

To connect one or more slaves to the MCU we can use two different slave configurations independent or daisy chain. In the independent slave configuration [Fig. 38], each slave has its own independent chip select line. SPI is typically applied in this manner. Only one chip choice can be asserted by the master at once. On daisy chain slave configuration [Fig. 32b], each slave's input is coupled with each slave's output, etc. Each slave's SPI port is programmed to replicate the data it obtained during the first batch of clock cycles exactly and send it out during the second batch of clock pulses. The entire chain functions as a communication shift register; daisy-chaining shift registers to give a bank of inputs or outputs using SPI is common practice. Until the active low SS line turns high, each slave duplicates input to output during the subsequent clock cycle. Instead of a different SS line being needed from the master for each slave, such a feature simply needs one SS line from the master.

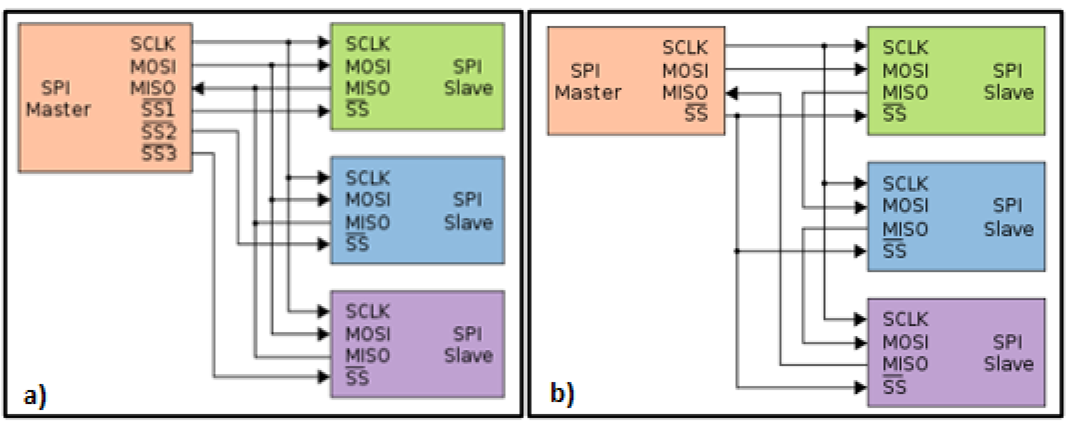

Fig. 38 a) A typical SPI bus: master and independent slaves, b) Daisy-chained SPI bus: master and cooperative slaves

4.1.11 Inter-Integrated Circuit I²C protocol

Inter-Integrated Circuit (I2C or IIC) is a synchronous multi-master, multi-slave protocol. That means that a clock signal shared by the master and slave synchronizes the output bits to the sampling bits. The master is always in charge of the clock signal. Also, a single master can be connected to multiple slaves, or we can have multiple masters controlling single, or multiple slaves. It uses a packet switching method, and a serial communication bus so the data is being transferred bit-bit using a single wire (SDA line). It was invented by Philips Semiconductors in 1982.

To send and receive data I2C uses two bidirectional lines: a serial clock pin (SCL) that the MCU board periodically pulses, and a serial data pin (SDA) that is used to transfer the data between the two components.

I2C uses messages to transfer data. Data frames are used to segment messages. Each message consists of an address frame that holds the slave's binary address and one or many data frames that hold the data being sent. Between each data frame, the message also contains ACK/NACK bits, read/write bits, and start and stop conditions [Fig. 39].

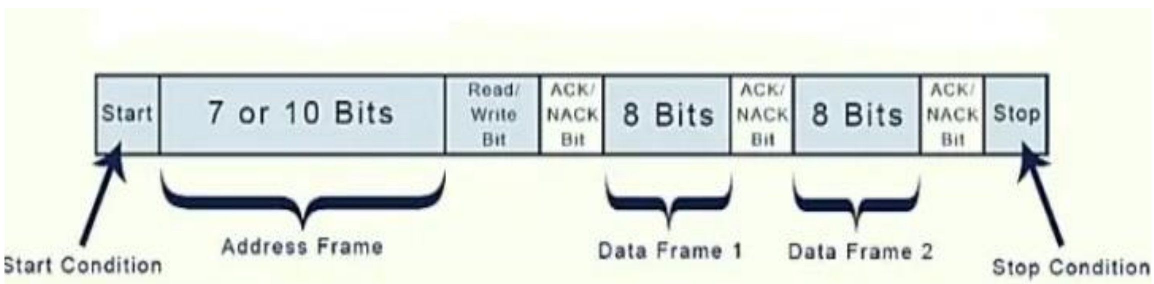

Fig. 39 I2C frame

Start Condition: SDA switches from a high voltage level to a low voltage level before SCL switches from high to low.

Stop Condition: SDA line switches from low to high voltage level after SCL switches from low to high.

Address Frame: A 7- or 10-bit sequence unique for each slave that identifies the slave when the master wants to talk to it.

Read/Write Bit: A single bit specifying whether the master is requesting data from it (high voltage level) or sending data to the slave (low voltage level).

ACK/NACK Bit: Each frame in a message is followed by an acknowledge/no-acknowledge bit. If an address frame or data frame was successfully received, an ACK bit is returned to the sender from the receiving device.

The Arduino theoretically can host up to 128 - the limit of unique I2C addresses. All I2C devices get connected in parallel. Practically every device needs to have a unique address in the range of 8 to 119. That is because address 0 is reserved as a "broadcast" address, addresses 1 to 7 are reserved for other purposes, and addresses 120 to 127 are reserved for future use. So that leaves us with 112 addresses so 112 devices.

4.2 Software implementation

Generally speaking, programming a microcontroller is not an easy task. Due to their low space memory and low processing power, the code written must be laconic. Therefore, the most suitable environment must be chosen to make things easier. Some quality aspects must be followed. To do that we must combine ideally the software and the hardware part of our device. Some of these quality aspects are:

i. Easy to use
ii. Reliability
iii. Testability
iv. Response Time
v. Power consumption
vi. Extensibility
vii. Stability and Portability
viii. Performance concerning CPU and memory usage
ix. Dynamic software deployment and over-the-air upgrade
x. Low-cost

To achieve those, we chose the hardware aforementioned. We tried to write lightweight code without unnecessary variables, or functions. As a programming language, we chose C/C++. This language offers many advantages that help us to achieve the quality aspects that we spoke about previously. Some of the many advantages are:

i. Extremely lightweight
ii. High speed and power
iii. Open source with many libraries
iv. Portability
v. Low-level manipulation
vi. Easy debugging
vii. Fast compilation

Concerning the programming environment, we chose the cross-platform application Arduino IDE. The Arduino Integrated Development Environment (IDE) is the main text editing program used for Arduino programming. It contains a hierarchy of operating menus, a message area, a text terminal, and a toolbar with buttons for standard functions. It supports both C and C++ using special code structure rules. The interface has a minimalist appearance. There are only 5 headings on the menu bar, as well as a series of buttons underneath which allow you to verify and upload your sketches.

Essentially, the IDE translates and compiles your sketches into code that Arduino can understand. Once your Arduino code is compiled it's then uploaded to the board's memory. If there are any errors in the Arduino code a warning message will flag up prompting the user to make changes, thus the code will not compile. If compilation is successful, we see how many bytes and the percentage of the total program storage space that the program occupies. We also see how many bytes global variables use and the percentage of total dynamic memory that they occupy.

Arduino IDE [Fig. 40] also includes a Serial Monitor and Serial plotter. The serial monitor is used by the computer to interface with Arduino. Unlike other platforms for writing software, Arduino lacks a built-in debugger. For monitoring and debugging, users can either use 3rd software or the serial monitor to print the active operations of Arduino. Debugging messages that include variable values and debugging messages can be printed to the serial monitor by using the Serial class. This will use serial pins "0" and "1", which are linked with the USB port, on the majority of Arduino models. The serial plotter as its name suggests is used to produce the real-time graph of our serial data. This way the data is much easier to analyze, and we achieve a visual display. We are capable of making graphs, graphs with negative values, and analyzing waveforms.

With Arduino IDE we can easily include libraries or search the libraries we need through the manage libraries menu. We can also choose between many Arduino boards depending on the one we use and even choose its specific processor. We can also choose between different ports, which gives us the possibility to use many boards from the same computer.

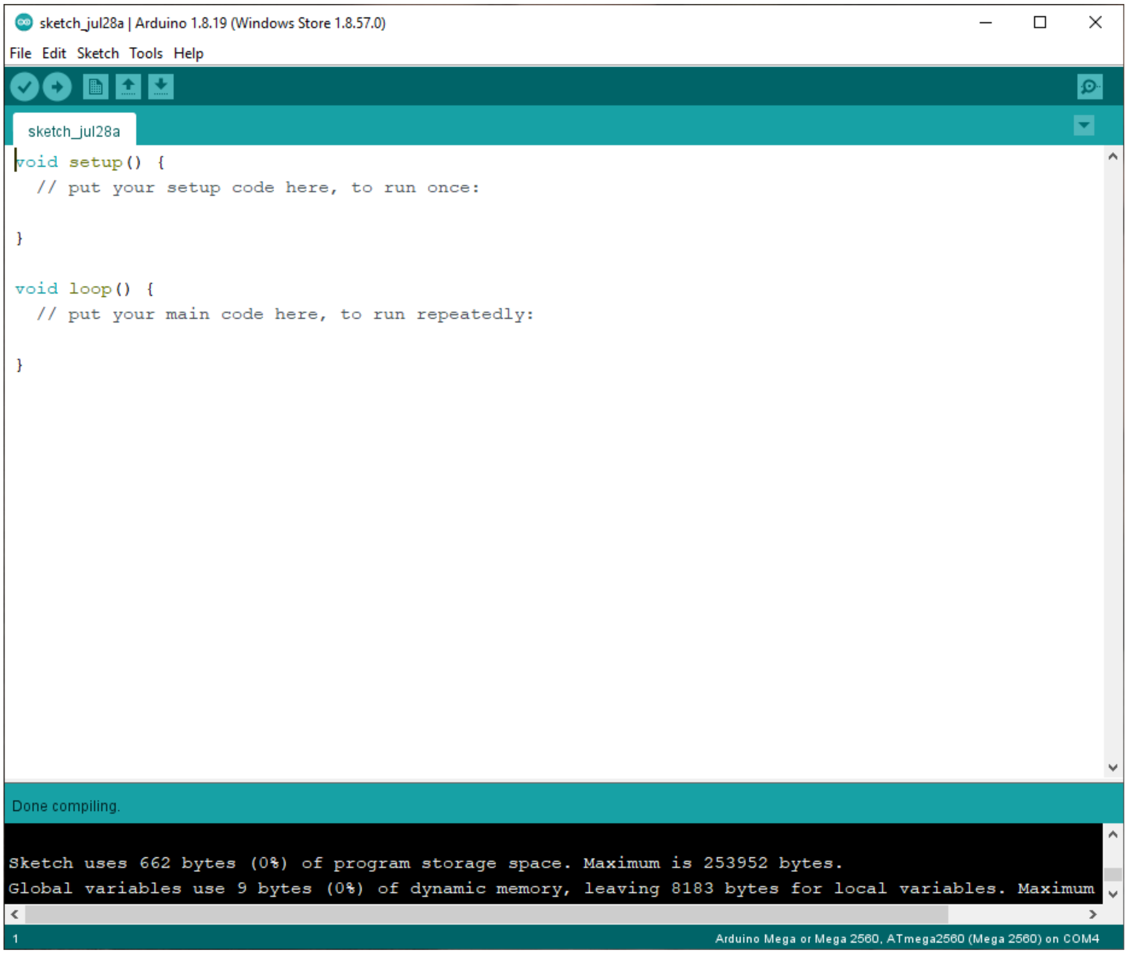

Fig. 40 A blank Arduino sketch

4.2.1 Inclusions and Declarations

The first thing that we can see in our coding is the #include <library> command [Fig. 41]. With this command, we can include every library we need for our sketch. The basic prerequisite is to have the library obtained.

```
#include <LiquidCrystal_I2C.h> // Library for LCD
#include "dht.h"
#include "SD.h"
#include "SPI.h"
#include "RTClib.h"
#include <EEPROM.h>
```

Fig. 41 Including libraries in Arduino IDE.

- LiquidCrystal_I2C: This library is used to communicate with LCD via the I2C protocol.
- dht.h: This library is used to receive values from the dht11 sensor (temperature and humidity).
- SD.h: This library is used combined with SPI.h -which allows us to use SPI communication protocol- to communicate with the SD module and manipulate files as we will explain later.
- RTClib.h: This library is used to communicate with DS3231. It also helps us format the values that we read.
- EEPROM.h: This library is used to store values in EEPROM memory thing which will be very helpful.

In lines 14-16 [Fig. 42] we create objects to use the libraries that were included before. As we can see for the LCD, we also have to give the I2C address. We already know this address from the datasheet. We could also use the sketch i2c_scanner which gives us the addresses in the serial monitor. For DS3231 we do not need to give the address because the library already does this for us. Regarding the next lines before the setup function, we declare some global variables that we will use later on. We also declare a 6x1 string matrix that holds the template of our menu. This way is the simplest. Another way to create a linked menu that goes back and forth is by creating a double-linked list. This way requires more memory sources.

```
14 LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 16, 2); // Change to (0x27,20,4) for 20x4 LCD.
15 dht DHT;
16 RTC_DS3231 rtc; //create an object RTC_DS3231
```

Fig. 42 Creating objects.

4.2.2 Setup function

As we mentioned before setup function [Fig. 43] is one out of two basic functions of our program. In the setup function, we check the connectivity of our modules, and we do all the initializations required. As we can see down below the first thing, we do is start the serial monitor at a 115200 baud rate. A serial monitor will be used for debugging purposes. Next, we initiate the LCD with the command LCD.init(); this is required if we are using the LiquidCrystal_I2C library. Then we also initiate the backlight of the LCD. Moving on we check if the modules of SD and rtc are connected and working without malfunctions. If something is wrong, we print a message on the LCD and then we request the user to restart the device.

If no errors occurred, the program prints a welcome message for one second. After this, we declare the pins that we will use as input (buttons pins), and as output (LED pins).

```
void setup() {

    Serial.begin(115200);
    lcd.init();  // Initiate the LCD.
    lcd.backlight(); // Initiate backlight of LCD noBacklight() to turn off

    if(! SD.begin()){
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Insert SD ");
        lcd.setCursor(0,1);
        lcd.print("and restart!");
        while (1) delay(10);}

    if (! rtc.begin()) {
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Clock Error!");
        while (1) delay(10);}

    lcd.clear();
    lcd.setCursor(4,0);
    lcd.print("Welcome!"); //welcome message
    delay(1000);

    //if (rtc.lostPower()) {
        //rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));/}

    pinMode(btnStep, INPUT);
    pinMode(btnExec, INPUT);
    pinMode(btnStepDown, INPUT);
    pinMode(btnMenu, INPUT);
    pinMode(LED_r, OUTPUT);
    pinMode(LED_b, OUTPUT);
    pinMode(LED_IR, OUTPUT);
    pinMode(LED_g, OUTPUT);
}
```

Fig. 43 Setup function

4.2.3 Loop function

```
void loop() {
```

In the loop function, we constantly check if any of the buttons is pressed. To do so we use the getButton() function that we will describe down below. If the menu button is pressed, 1 is added to a counter. This way and by using the displayMenu() function we print the menu to the LCD. If we are at Measure/Scan on the menu we can press the scan button and call the measure() function. This function starts the measurement procedure. In any different case and by pressing the ok button in we activate the execMenu() function that is responsible for the settlement of the specific menu operations.

4.2.4 getButton function

```
bool getButton(int Btn) {
```

The getButton(int Btn) function is used to do a digital read. We give as an argument an integer that represents the pin of each button. This way we can know which button is pressed every time. This function is also responsible for software debounce. To achieve that we do a double delayed digitalRead and also use a boolean variable. This combined with the checking in loop function gives a perfect software debounce. This function returns a boolean variable.

4.2.5 displayMenu function

```
void displayMenu() {
```

The displayMenu() function is responsible to show the menu on the LCD. Based on the global variable menuIdx -which is manipulated from the loop function- we show the contents of the menuText matrix that we mentioned before.

4.2.6 execMenu function

```
void execMenu() {
  // do what ever you want depending on menuIdx
```

By calling the execMenu function we do certain processes depending on the menuIdx variable. If menuIdx is 0 the function changes what VI we will measure and returns to main menu. If menuIdx is 1 at the same time LCD shows “2. Data View” the function dataShow() is called. If menuIdx is 2 at the same time LCD shows “3. Last Data DEL” and the function dataDelLast() is called. If menuIdx is 3 at the same time LCD shows “4. ALL data DEL” the function dataDelAll() is called. If menuIdx is 4 at the same time LCD shows “5. Average Data” the function avgShow() is called. If menuIdx is 5 at the same time LCD shows “6. Info” the function rotate() is called.

4.2.7 measure function

```
void measure() {
```

The measure function is one of the most important functions in our code. It is used combined with the blinkLed function to take the measures for the plant. We start by getting the date and time from rtc, temperature, and humidity from the DHT11. After this, we obtain the fid value from

EEPROM to be able to see if this is the first measure or not. If fid == “0xFF” then this measure is the first and we store a zero in the EEPROM in address 0. Then we start printing on the LCD the message “measuring. . .” and a progress bar [Fig. 44]. Next, we get the LED values from blinkLed, and we calculate the vegetation indices. When the vegetation indices are calculated they are shown for three seconds on the LCD. We also increase by one the fid value and store it in the EEPROM.

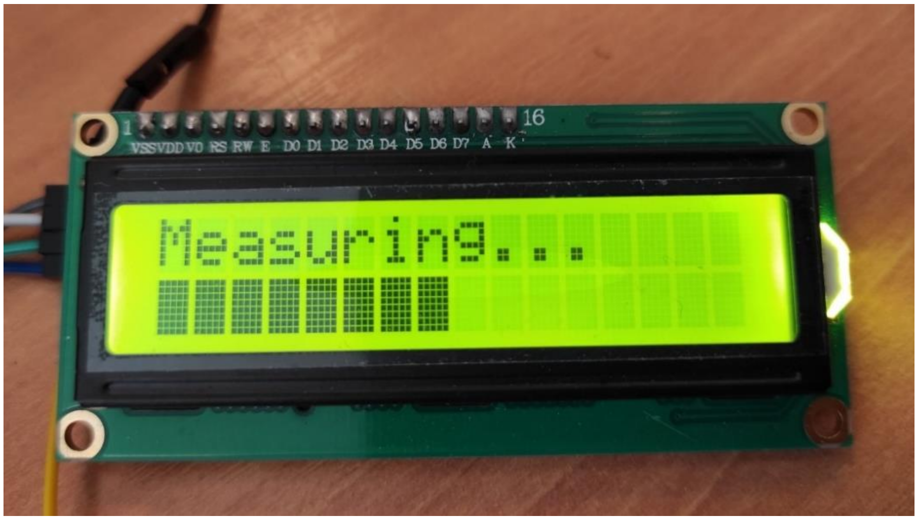

Fig. 44 Device Progress Bar.

Now we are ready to store the values we want on the SD card. We want to store the EVI, time, and date with the format e.g., “Tue,04/08/2022,15:10:20”. We create -if it does not exist- a file named measure.csv and we open it with write rights. If the file could not be created, we print the appropriate message on the LCD. Otherwise with the help of the file.print(data) we start writing in the SD. We separate each attribute by a comma to be able to separate the data later on. We also create some rules to keep each line’s total byte sum the same. We also write the temperature and humidity. Last but not least we write the reflectance values of RGB and IR LEDs separated by a comma and then we write “\n” to change the line inside the file. In the end, we close the file, and we return to the menu. An example of how the data will be imported to the computer through the SD card is shown in [Fig 45]. If we want to measure again, we press the Scan Button and repeat the above process.

	A	B	C	D	E	F	G	H	I	J	K
1	ID	VI	Day	Date	Time	Temp	Hum	Red channel	IR channel	Green Channel	Blue Channel
2	0	0.53	Mon	12-09-22	14:24:42	27	64	765	2464	684	340
3	1	-0.27	Mon	12-09-22	14:25:16	27	64	3327	1902	1052	506
4	2	0.46	Mon	12-09-22	14:25:49	27	63	860	2354	703	351
5	3	0.52	Mon	12-09-22	14:26:23	27	63	766	2394	683	369
6	4	0.5	Mon	12-09-22	14:26:56	27	63	808	2444	662	340
7	5	0.48	Mon	12-09-22	14:27:30	27	63	776	2199	690	350
8	6	0.48	Mon	12-09-22	14:28:03	27	63	777	2186	689	357
9	7	0.48	Mon	12-09-22	14:28:37	27	63	776	2185	687	355
10	8	0.48	Mon	12-09-22	14:29:10	27	62	775	2195	691	343
11	9	0.51	Mon	12-09-22	14:29:44	27	63	827	2568	714	347
12	10	0.5	Mon	12-09-22	14:30:17	27	62	841	2536	710	344
13	11	0.51	Mon	12-09-22	14:30:50	27	61	838	2589	717	348
14	12	0.47	Mon	12-09-22	14:31:24	27	62	841	2350	635	494
15	13	-0.26	Mon	12-09-22	14:31:57	27	62	3391	1978	1043	501
16	14	-0.26	Mon	12-09-22	14:32:31	27	62	3396	1985	1045	501
17	15	-0.26	Mon	12-09-22	14:33:04	27	63	3402	1986	1045	502
18	16	-0.26	Mon	12-09-22	14:33:38	27	63	3400	1981	1045	502
19	17	-0.26	Mon	12-09-22	14:34:11	27	62	3405	1985	1046	502
20	18	-0.26	Mon	12-09-22	14:34:45	27	62	3402	1985	1046	492
21	19	0.5	Mon	12-09-22	14:35:18	27	63	826	2505	776	373

Fig. 45 An example of how the data we pass in SD look in Excel.

4.2.8 blinkLed function

```
int blinkLed(int led){
```

Using this function, we will light our LEDs for a certain time and at the same time, we will get AS7341's corresponding channel value. Each LED as we said before uses a specific channel. The function gets as an argument the LED pin and returns the value measured by AS7341. To obtain accuracy and stability we will use a median filter. We will measure five times with each LED. The LED will blink five times and we will be taking measures. Then a median filter will be applied to the values and finally, we will take the median value. The values will be stored in a matrix to be easier to handle.

4.2.9 dataShow function

```
void dataShow() {
  readFile("/measures.csv"); //call the readFile function
  displayMenu();}
```

The dataShow function is used to call the readFile(const char * path). We used a different function to do the reading for clarity reasons. After the reading, we call the displayMenu() to go back to the menu display.

4.2.10 readFile function

```
void readFile(const char * path){
```

The readFile function is called to read the file from the SD card and show on the LCD the measurements. Combined with the lcdPrint() function we print on the LCD the ID of the measurement the value of measurements and the time that we got the measure. If there is no file, the function prints on the LCD an error message. As input, this function gets the path of the file in the SD catalog.

The basic idea behind this function is to open the file and manipulate the file pointer that exists at the beginning of the file. With the help of this pointer and fseek function. Given that each row in SD is fixed bytes we can change lines and go back and forth by multiplying those fixed bytes by a constant. If we press the arrow next button this number is increased and if we press the arrow previous this number is decreased. The reading is terminated when we reach the last value and press the arrow next again or if we press the menu button anytime. To know how many measures are done, we recall the fid value from EEPROM.

We use the readStringUntil(terminator) function to get the data from the SD card. This function gives us a stream of bytes until the terminator character is detected or it times out. We will not use this whole string, so we have to cut some parts of it. To achieve that we use the lcdPrint function. This function gets as input a buffer that contains the string we read before with the readStringUntil function.

4.2.11 lcdPrint function

```
void lcdPrint(String buffer_){
```

This function is used with readFile to format the message that we will show in the LCD. It uses the substring function to string and take only the parts we need. We want to take the ID, the measured value, and the time. This will give us a formatted result as shown in [Fig. 46].

Fig. 46 Data Show

4.2.12 dataDelLast function

```
void dataDelLast(){
```

This function as its name implies is used to delete the last measure. This is a useful feature because our device is sensitive to distance and angle regarding the measurements. With this feature, we can delete a mismeasurement.

The best and not most complicated way to delete a line from an SD card is to copy all the measures except the one that must be deleted in a temporary file. Then create again the original file and copy the values of the temporary file in it. The temporary file is deleted afterward.

To achieve the above, we must know the number of measurements so we once again recall the fid from EEPROM -now we can conclude the importance of EEPROM-. If the file does not exist an error message is shown on the LCD. If the file exists, we begin the procedure of copying. First, we start reading from measures.csv line by line and we copy those lines adding a "\n" at the end in temp.csv. Then we delete the measures.csv file by using deleteFile(path). Then we subtract 1 from fid and store it again in EEPROM. This way we will not make any mistakes in measurement ID. Then we do the same procedure from temp.csv to the new measure.csv that we create. Afterward, the temp.csv file is deleted, and we go back to the menu.

4.2.13 dataDelAll function

```
void dataDelAll(){
```

This function is used to delete all the measurements. Because deleting everything is risky, so we have a warning message shown on the LCD. The messages displayed are YES and NO. We can change from yes to no with the next or previous arrow and we have to press the ok button to choose. If no is selected, we go back to the menu. If yes is selected, we call the deleteFile function which deletes the measure.csv file. Then we also have to empty the EEPROM to be able to start the measures from the very beginning.

4.2.14 deleteFile function

```
void deleteFile(const char * path){
```

This function is called by two other functions the dataDelAll and dataDelLast. It is a very useful function because it allows us to delete the whole file. This function uses the SD.remove function. Regarding the result of the removal, we display on the LCD an error message or a successful one.

4.2.14 avgShow function

```
292 void avgShow(){
```

The avgShow function is used to show the number of measurements and the average value. The idea to build this function is based on image processing where the picture is manipulated as a single-dimensional array. We are trying to create a formula that will get the whole column of values and will give us the sum of this column. We already have the quantity of the measures stored in EEPROM (fid). To get the sum we do a loop while the file is available. That means that when we go to the end of the file (EOF) the loop stops. While in this loop we get the first value that is in the second 'column'. Then, with every certain repetition, we meet the measured value again, so we add this value to the sum. The stream is separated by a comma. The same procedure could also be achieved by using the substring() function. When the loop ends, we print in LCD the message: "Qua: quantity" and in the second line the message: "Avg: average data". It will be shown for 3 seconds before we go back to the menu.

4.2.15 rotate function

```
327 void rotate()
```

The rotate function is based on the loop function idea. With this function combined with displayInfo, we can create a rotating. Each time the user presses the arrow next or previous a counter is changing. This counter is passed as an argument in the displayInfo function to show the proper information message. The messages shown are software version, manufacturer, a thank for buying message, humidity, system temperature, date, and time. This menu is fully rotational and if we surpass the end, it goes back to the begging. If we press the menu button it goes back to the main menu.

4.2.16 displayInfo function

```
361 void displayInfo(int iid) {
```

This function is used to display the information of our device as aforementioned. It takes an integer as an argument and based on this integer it displays on the LCD the contents of the info matrix. The info matrix contains a thank you note, the version of the software, and the name of the manufacturer. We use DS3231 to get the internal temperature, time, and date, and DHT11 to get the humidity. The other information is fixed strings stored in the matrix.

4.2.17 image function

```
void image() {
  //create our symbols with the help of https://tusindfryd.github.io/screenduino/
```

The image function is used to create simple graphics on our LCD. It has an online UI that allows you to select the pixels of the LCD that you want to make black this way we can create arrows or the enter arrow. The code is generated automatically, and we can copy the whole code to use it. It supports an 8x2, 16x2, and even a 20x4 LCD. It also uses the createChar function that allows us to create custom characters [Fig. 47]. This feature is provided by tusindfryd (GitHub).

Fig. 47 Custom Characters

4.2.18 writeUnsignedIntIntoEEPROM function

```
void writeUnsignedIntIntoEEPROM(int address, unsigned int number)
{
  EEPROM.write(address, number >> 8);
  EEPROM.write(address + 1, number & 0xFF);
}
```

This function is used to write an unsigned integer into the EEPROM. As we previously mentioned EEPROM is the key feature for our device to work. This function takes as arguments the address we want to store the value and the number. We always write the value at the first cell of the EEPROM, so the location we use is zero. Because EEPROM can only be written in bytes we have to break the number into several bytes and store each byte separately.

An unsigned int takes 2 bytes so it must be split in half. We shift 8 times all the bits from the number to the right for the first byte. This way, we only get the first 8 bits. For example, the number 10000 is represented as 10011100010000 in binary [Fig. 48]. If we shift this number eight times to the right, we will obtain 10011100. We make an AND operation on the second byte using 0xFF, which is 255's hexadecimal value (we could also use "& 255" as a substitute for "& 0xFF"). We only receive the 6 bits -up to 8- on the right with this method. So, from 10011100010000 we will eventually get 010000 and we store it in the next address byte (address+1). In this way, we can store the values we want in the EEPROM. To write those bytes, we use EEPROM.write(address, number) function.

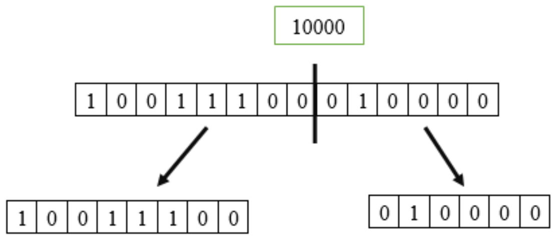

Fig. 48 Unsigned integers in binary and split

4.2.19 readUnsignedIntFromEEPROM function

```
unsigned int readUnsignedIntFromEEPROM(int address)
{
  return (EEPROM.read(address) << 8) + EEPROM.read(address + 1);
}
```

Now to obtain the unsigned integer, we stored in the EEPROM we have to call the readUnsignedIntFromEEPROM. This function takes as an argument the address of the byte we want to read. Since we want to read an unsigned integer, that is split in half we have to reconstruct the number. To achieve that we do the opposite of what we did before. We shift the highest bits by 8, this time to the left. And we add the lower bits to the number. To read from the EEPROM we use EEPROM.read(address) function.

5. Experiments and evaluation

In order to achieve a correct and trustworthy evaluation of our device, we used the HR2000+ spectrometer. HR2000+ is a High-Speed Miniature Fiber Optic Spectrometer that provides optical resolution as good as 0.035 nm (FWHM). The HR2000+ is responsive from 200-1100 nm, but the specific range and resolution depend on the grating and entrance slit selections. Regarding the detector, it has a Sony ILX-511B 2048-element linear silicon CCD array. Regarding sensitivity, we can get 75 photons per count at 400 nm or 41 photons per count at 600 nm. Furthermore, the integration time is 1 ms up to 65 seconds. To get the measures we used the fiber optic extension.

The method we used to evaluate is linear regression. Linear regression analysis is used to predict the value of a variable based on the value of another variable. The variable you want to forecast is called the dependent variable. The variable you are using to forecast the other variable's value is called the independent variable.

To achieve stability and consistency in our measurements we fasten both devices [Fig. 49] and the optical fiber with the LEDs [Fig. 50]. The distance between the device and the leaf was about 4cm long and the same for fiber optic and the LED. To provide more stable measures we have our device parallel to the leaf. If the angle changes so do the value of the sensor. The same methodology was followed for the HR2000+.

To ensure that the results will be trustworthy, invariably, and reliable we will measure 10 lemon leaves and 10 strawberry leaves using our device and the HR2000+. Then we will use the linear regression on the measurements. We will try to be as precise as possible to measure at the same spot of the leaf with each LED. The experiment took place in an almost completely dark room. This way we avoided noise in our measurements or interference from ambient light or bulb light.

We also used paper tapes to fasten the leaf on the paperboard this way we will have a more stable sample to work with. We also watered the leaves while conducting the experiment in order to keep them in the same condition.

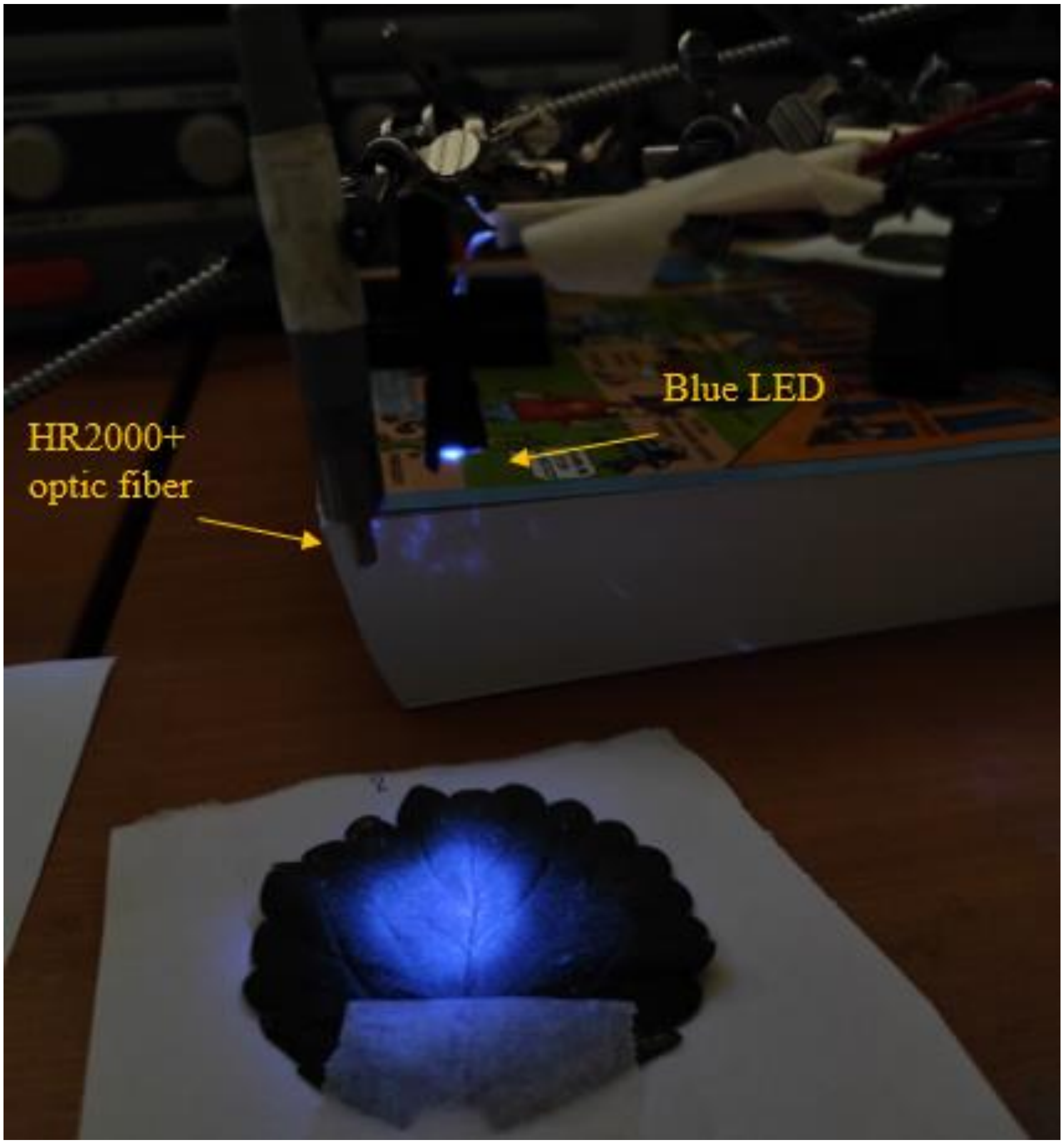

Fig. 49 HR2000+ measuring.

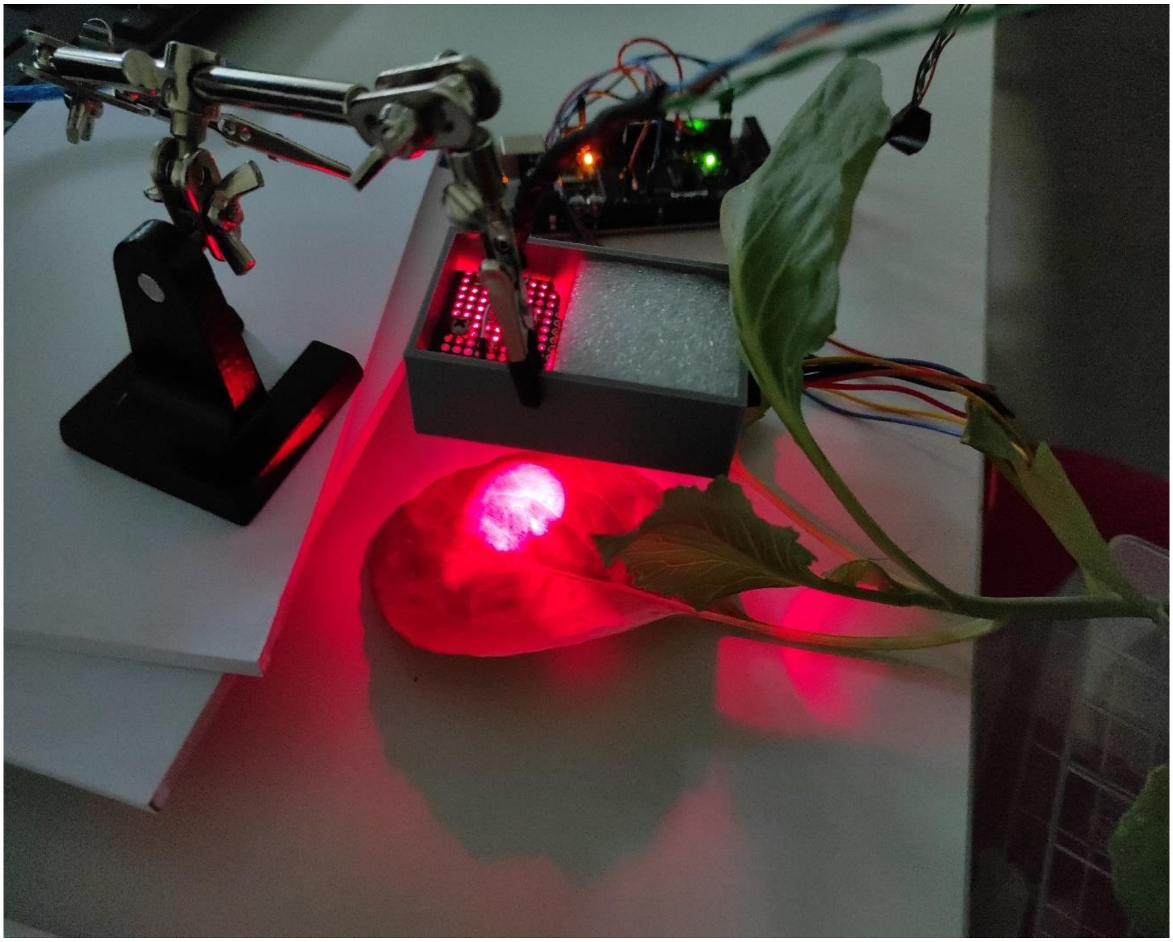

Fig. 50 AS7341 measuring.

5.1 Linear Regression Analysis

To perform the linear regression analysis, we used Microsoft Excel. Excel is an easy-to-use and simple tool with many capabilities. As independent value, we used the HR2000+'s value, and as dependent our device's value. This way we will find the correlation between the two values. We will use the formula of simple linear regression which is $\boldsymbol{y = \alpha + \beta x}$. For each LED we will use the formula to calculate the factors and R-squared.

R-squared (R^2) is a statistical measure that shows how much of a dependent variable's variance is explained by one or more independent variables in a regression model. R-squared measures how well the variation of one variable accounts for the variance of the second, as opposed to correlation, which describes the strength of the relationship between independent and dependent variables. Therefore, if a model's R^2 is 0.50, its inputs can account for around half of the observed variation. Formula for R^2 is $\boldsymbol{R^2 = 1 - \frac{\text{Unexplained Variation}}{\text{Total Variation}}}$.

5.1.1 Blue LED on lemon leaves

For the blue LED, we have the measures shown in [Table 3,4]:

BLUE LED LEMON										
num	LEAF 1	LEAF 2	LEAF 3	LEAF 4	LEAF 5	LEAF 6	LEAF 7	LEAF 8	LEAF 9	LEAF 10
0	163	137	165	150	163	133	203	196	233	150
1	163	137	165	150	163	133	203	196	233	150
2	163	137	165	150	163	133	203	196	233	150
3	163	137	165	150	163	133	203	196	233	150
4	163	137	165	150	163	133	203	196	233	150
5	163	137	165	150	163	133	203	196	233	150
6	163	137	165	150	163	133	203	196	233	150
7	163	137	165	150	163	133	203	196	233	150
8	163	137	165	150	163	133	203	196	233	150
9	163	137	165	150	163	133	203	196	233	150

Table 3 Device's Blue LED measures on lemon leaves.

BLUE HR2000+ LEMON										
num	LEAF 1	LEAF 2	LEAF 3	LEAF 4	LEAF 5	LEAF 6	LEAF 7	LEAF 8	LEAF 9	LEAF 10
0	1252	1066	1296	1176	1250	1040	1369	1320	1500	1175
1	1252	1066	1296	1176	1250	1040	1369	1320	1500	1175
2	1252	1066	1296	1176	1250	1040	1369	1320	1500	1175
3	1252	1066	1296	1176	1250	1040	1369	1320	1500	1175
4	1252	1066	1296	1176	1250	1040	1369	1320	1500	1175
5	1252	1066	1296	1176	1250	1040	1369	1320	1500	1175
6	1252	1066	1296	1176	1250	1040	1369	1320	1500	1175
7	1252	1066	1296	1176	1250	1040	1369	1320	1500	1175
8	1252	1066	1296	1176	1250	1040	1369	1320	1500	1175
9	1252	1066	1296	1176	1250	1040	1369	1320	1500	1175

Table 4 HR2000+ Blue LED measures on lemon leaves.

As we can see for both devices the measures are stable and they also are linear. Now we are ready to do linear regression [Fig. 51] and calculate the R^2 for the blue LED only. Because for the same leaf we get the same value we will use only the first line for each matrix.

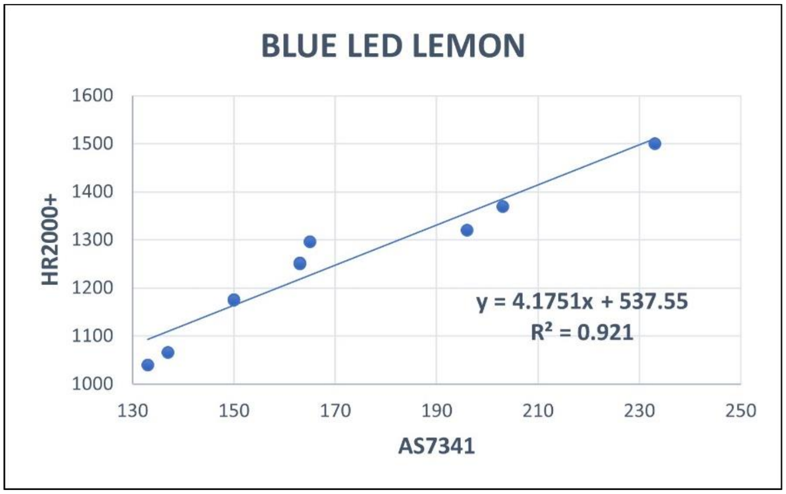

Fig. 51 Blue LED linear regression regarding the lemon leaves.

5.1.2 Red LED on lemon leaves

For the red LED, we have the measures shown in [Table 5,6]:

RED LED										
num	LEAF 1	LEAF 2	LEAF 3	LEAF 4	LEAF 5	LEAF 6	LEAF 7	LEAF 8	LEAF 9	LEAF 10
0	612	500	568	621	587	564	680	625	790	541
1	612	500	568	621	587	564	680	625	790	541
2	612	500	568	621	587	564	680	625	790	541
3	612	500	568	621	587	564	680	625	790	541
4	612	500	568	621	587	564	680	625	790	541
5	612	500	568	621	587	564	680	625	790	541
6	612	500	568	621	587	564	680	625	790	541
7	612	500	568	621	587	564	680	625	790	541
8	612	500	568	621	587	564	680	625	790	541
9	612	500	568	621	587	564	680	625	790	541

Table 5 Device's Red LED measures on lemon leaves.

RED HR2000+										
num	LEAF 1	LEAF 2	LEAF 3	LEAF 4	LEAF 5	LEAF 6	LEAF 7	LEAF 8	LEAF 9	LEAF 10
0	4710	4010	4500	4760	4580	4430	4940	4800	5350	4200
1	4710	4010	4500	4760	4580	4430	4940	4800	5350	4200
2	4710	4010	4500	4760	4580	4430	4940	4800	5350	4200
3	4710	4010	4500	4760	4580	4430	4940	4800	5350	4200
4	4710	4010	4500	4760	4580	4430	4940	4800	5350	4200
5	4710	4010	4500	4760	4580	4430	4940	4800	5350	4200
6	4710	4010	4500	4760	4580	4430	4940	4800	5350	4200
7	4710	4010	4500	4760	4580	4430	4940	4800	5350	4200
8	4710	4010	4500	4760	4580	4430	4940	4800	5350	4200
9	4710	4010	4500	4760	4580	4430	4940	4800	5350	4200

Table 6 HR2000+ Blue LED measures on lemon leaves.

We used the same methodology for the Red LED in order to find the factors and R-squared. The results are shown in [Fig. 52].

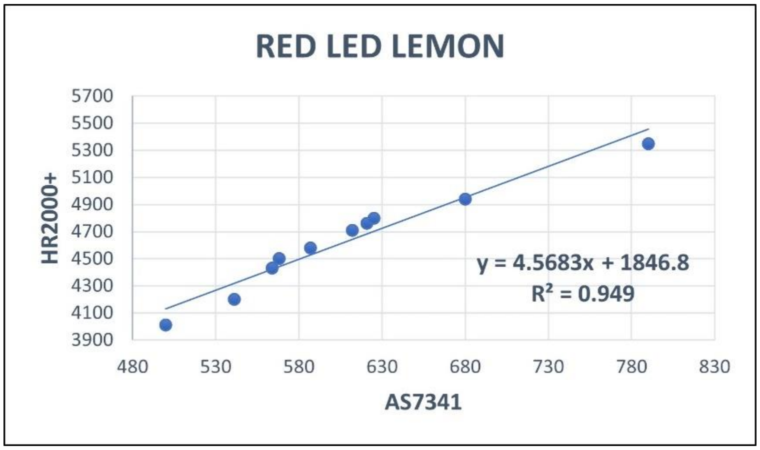

Fig. 52 Red LED linear regression regarding the lemon leaves.

5.1.3 Green LED on lemon leaves

For the green LED, we have the measures shown in [Table 7,8]:

GREEN LED LEMON										
num	LEAF 1	LEAF 2	LEAF 3	LEAF 4	LEAF 5	LEAF 6	LEAF 7	LEAF 8	LEAF 9	LEAF 10
0	603	516	594	654	624	588	677	613	758	552
1	603	516	594	654	624	588	677	613	758	552
2	603	516	594	654	624	588	677	613	758	552
3	603	516	594	654	624	588	677	613	758	552
4	603	516	594	654	624	588	677	613	758	552
5	603	516	594	654	624	588	677	613	758	552
6	603	516	594	654	624	588	677	613	758	552
7	603	516	594	654	624	588	677	613	758	552
8	603	516	594	654	624	588	677	613	758	552
9	603	516	594	654	624	588	677	613	758	552

Table 7 Device's Green LED measures on lemon leaves.

GREEN HR2000+ LEMON										
num	LEAF 1	LEAF 2	LEAF 3	LEAF 4	LEAF 5	LEAF 6	LEAF 7	LEAF 8	LEAF 9	LEAF 10
0	4650	4100	4600	4800	4700	4580	4900	4620	5200	4490
1	4650	4100	4600	4800	4700	4580	4900	4620	5200	4490
2	4650	4100	4600	4800	4700	4580	4900	4620	5200	4490
3	4650	4100	4600	4800	4700	4580	4900	4620	5200	4490
4	4650	4100	4600	4800	4700	4580	4900	4620	5200	4490
5	4650	4100	4600	4800	4700	4580	4900	4620	5200	4490
6	4650	4100	4600	4800	4700	4580	4900	4620	5200	4490
7	4650	4100	4600	4800	4700	4580	4900	4620	5200	4490
8	4650	4100	4600	4800	4700	4580	4900	4620	5200	4490
9	4650	4100	4600	4800	4700	4580	4900	4620	5200	4490

Table 8 HR2000+ Green LED measures on lemon leaves.

We used the same methodology for the green LED in order to find the factors. The results are shown in [Fig. 53]

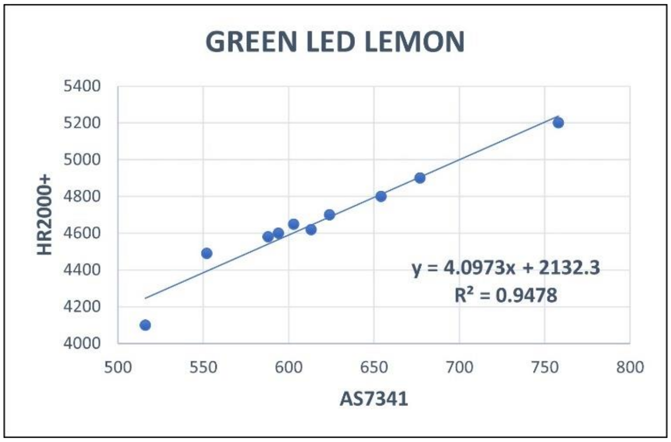

Fig. 53 Green LED linear regression regarding the lemon leaves.

5.1.4 IR LED on lemon leaves

For the IR LED, we have the measures shown in [Table 9,10]:

IR LED LEMON										
num	LEAF 1	LEAF 2	LEAF 3	LEAF 4	LEAF 5	LEAF 6	LEAF 7	LEAF 8	LEAF 9	LEAF 10
0	2059	1900	2000	2084	1962	1944	2172	2032	2350	1865
1	2059	1900	2000	2084	1962	1944	2172	2032	2350	1865
2	2059	1900	2000	2084	1962	1944	2172	2032	2350	1865
3	2059	1900	2000	2084	1962	1944	2172	2032	2350	1865
4	2059	1900	2000	2084	1962	1944	2172	2032	2350	1865
5	2059	1900	2000	2084	1962	1944	2172	2032	2350	1865
6	2059	1900	2000	2084	1962	1944	2172	2032	2350	1865
7	2059	1900	2000	2084	1962	1944	2172	2032	2350	1865
8	2059	1900	2000	2084	1962	1944	2172	2032	2350	1865
9	2059	1900	2000	2084	1962	1944	2172	2032	2350	1865

Table 9 Device's IR LED measures on lemon leaves.

IR HR2000+										
num	LEAF 1	LEAF 2	LEAF 3	LEAF 4	LEAF 5	LEAF 6	LEAF 7	LEAF 8	LEAF 9	LEAF 10
0	15543	14000	14850	15708	15114	14619	16302	15500	17655	13860
1	15543	14000	14850	15708	15114	14619	16302	15500	17655	13860
2	15543	14000	14850	15708	15114	14619	16302	15500	17655	13860
3	15543	14000	14850	15708	15114	14619	16302	15500	17655	13860
4	15543	14000	14850	15708	15114	14619	16302	15500	17655	13860
5	15543	14000	14850	15708	15114	14619	16302	15500	17655	13860
6	15543	14000	14850	15708	15114	14619	16302	15500	17655	13860
7	15543	14000	14850	15708	15114	14619	16302	15500	17655	13860
8	15543	14000	14850	15708	15114	14619	16302	15500	17655	13860
9	15543	14000	14850	15708	15114	14619	16302	15500	17655	13860

Table 10 HR2000+ IR LED measures on lemon leaves.

We used the same methodology for the IR LED in order to find the factors. The results are shown in [Fig. 54].

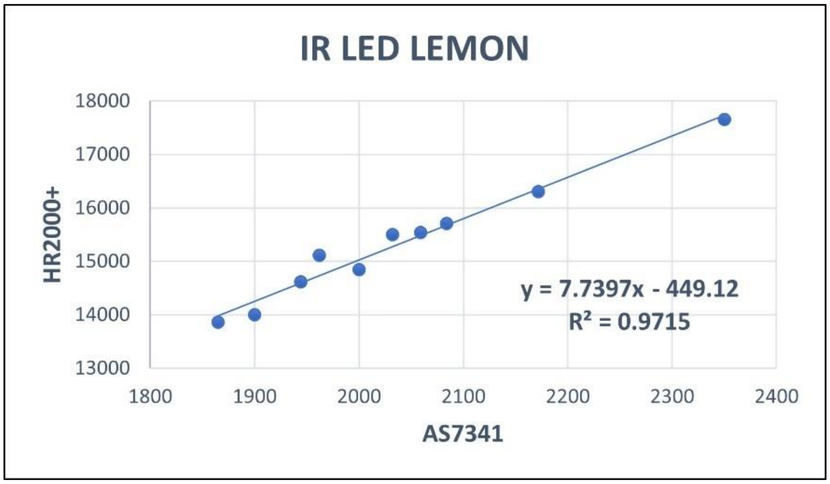

Fig. 54 Infrared LED linear regression regarding the lemon leaves.

5.1.5 Blue LED on strawberry leaves

We will also use the same methodology as before regarding the strawberry leaves. This will help us test if the sensor is stable or does it show any changes depending on the plant we measure. This will show us also if our device can measure different kinds of plants with the same accuracy, and correlation with the HR2000+ as it did before.

For the blue LED, we have the measures shown in [Table 11,12]:

BLUE LED STRAWBERRY										
num	LEAF 1	LEAF 2	LEAF 3	LEAF 4	LEAF 5	LEAF 6	LEAF 7	LEAF 8	LEAF 9	LEAF 10
0	165	158	150	148	175	184	161	154	159	175
1	165	158	150	148	175	184	161	154	159	175
2	165	158	150	148	175	184	161	154	159	175
3	165	158	150	148	175	184	161	154	159	175
4	165	158	150	148	175	184	161	154	159	175
5	165	158	150	148	175	184	161	154	159	175
6	165	158	150	148	175	184	161	154	159	175
7	165	158	150	148	175	184	161	154	159	175
8	165	158	150	148	175	184	161	154	159	175
9	165	158	150	148	175	184	161	154	159	175

Table 11 Device's Blue LED measures on strawberry leaves.

BLUE HR2000+ STRAWBERRY										
num	LEAF 1	LEAF 2	LEAF 3	LEAF 4	LEAF 5	LEAF 6	LEAF 7	LEAF 8	LEAF 9	LEAF 10
0	1522	1450	1390	1330	1590	1650	1480	1460	1483	1560
1	1522	1450	1390	1330	1590	1650	1480	1460	1483	1560
2	1522	1450	1390	1330	1590	1650	1480	1460	1483	1560
3	1522	1450	1390	1330	1590	1650	1480	1460	1483	1560
4	1522	1450	1390	1330	1590	1650	1480	1460	1483	1560
5	1522	1450	1390	1330	1590	1650	1480	1460	1483	1560
6	1522	1450	1390	1330	1590	1650	1480	1460	1483	1560
7	1522	1450	1390	1330	1590	1650	1480	1460	1483	1560
8	1522	1450	1390	1330	1590	1650	1480	1460	1483	1560
9	1522	1450	1390	1330	1590	1650	1480	1460	1483	1560

Table 12 HR2000+ Blue LED measures on strawberry leaves.

As we can see again, for both the devices the measures are stable and they also are linear. Now we are ready to do linear regression analysis [Fig. 55] and calculate the R^2 for the blue LED only. Because for the same leaf we get the same value we will use only the first line for each matrix.

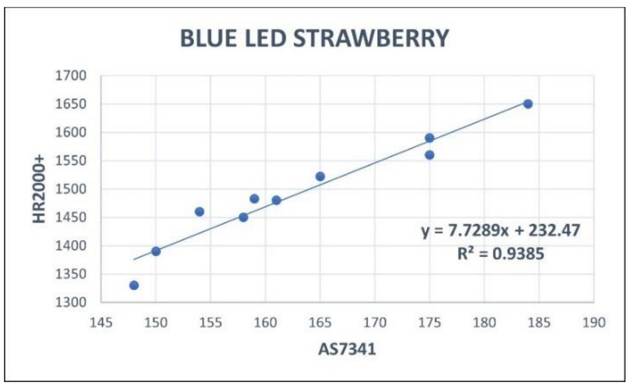

Fig. 55 Blue LED linear regression regarding the strawberry leaves.

5.1.6 Red Led on strawberry leaves

For the red LED, we have the measures shown in [Table 13,14]:

RED LED STRAWBERRY										
num	LEAF 1	LEAF 2	LEAF 3	LEAF 4	LEAF 5	LEAF 6	LEAF 7	LEAF 8	LEAF 9	LEAF 10
0	543	530	512	507	554	566	539	525	537	549
1	543	530	512	507	554	566	539	525	537	549
2	543	530	512	507	554	566	539	525	537	549
3	543	530	512	507	554	566	539	525	537	549
4	543	530	512	507	554	566	539	525	537	549
5	543	530	512	507	554	566	539	525	537	549
6	543	530	512	507	554	566	539	525	537	549
7	543	530	512	507	554	566	539	525	537	549
8	543	530	512	507	554	566	539	525	537	549
9	543	530	512	507	554	566	539	525	537	549

Table 13 Device's Red LED measures on strawberry leaves.

	RED HR2000+ STRAWBERRY									
num	LEAF 1	LEAF 2	LEAF 3	LEAF 4	LEAF 5	LEAF 6	LEAF 7	LEAF 8	LEAF 9	LEAF 10
0	3700	3570	3440	3380	3842	4016	3654	3514	3620	3755
1	3700	3570	3440	3380	3842	4016	3654	3514	3620	3755
2	3700	3570	3440	3380	3842	4016	3654	3514	3620	3755
3	3700	3570	3440	3380	3842	4016	3654	3514	3620	3755
4	3700	3570	3440	3380	3842	4016	3654	3514	3620	3755
5	3700	3570	3440	3380	3842	4016	3654	3514	3620	3755
6	3700	3570	3440	3380	3842	4016	3654	3514	3620	3755
7	3700	3570	3440	3380	3842	4016	3654	3514	3620	3755
8	3700	3570	3440	3380	3842	4016	3654	3514	3620	3755
9	3700	3570	3440	3380	3842	4016	3654	3514	3620	3755

Table 14 HR2000+ Blue LED measures on strawberry leaves.

We used the same methodology for the Red LED to find the factors and R-squared. The results are shown in [Fig. 56].

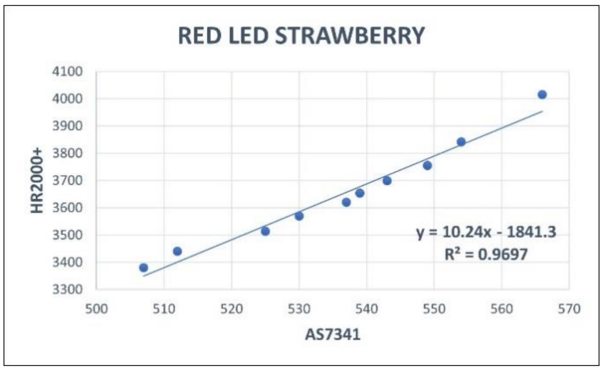

Fig. 56 Red LED linear regression regarding the strawberry leaves.

5.1.7 Green LED on strawberry leaves

For the green LED, we have the measures shown in [Table 15,16]:

GREEN LED STRAWBERRY										
num	LEAF 1	LEAF 2	LEAF 3	LEAF 4	LEAF 5	LEAF 6	LEAF 7	LEAF 8	LEAF 9	LEAF 10
0	610	588	583	578	622	630	605	585	597	614
1	610	588	583	578	622	630	605	585	597	614
2	610	588	583	578	622	630	605	585	597	614
3	610	588	583	578	622	630	605	585	597	614
4	610	588	583	578	622	630	605	585	597	614
5	610	588	583	578	622	630	605	585	597	614
6	610	588	583	578	622	630	605	585	597	614
7	610	588	583	578	622	630	605	585	597	614
8	610	588	583	578	622	630	605	585	597	614
9	610	588	583	578	622	630	605	585	597	614

Table 15 Device's Green LED measures on strawberry leaves.

	GREEN HR2000+ STRAWBERRY									
num	LEAF 1	LEAF 2	LEAF 3	LEAF 4	LEAF 5	LEAF 6	LEAF 7	LEAF 8	LEAF 9	LEAF 10
0	5100	4900	4840	4860	5250	5350	5020	4880	4980	5203
1	5100	4900	4840	4860	5250	5350	5020	4880	4980	5203
2	5100	4900	4840	4860	5250	5350	5020	4880	4980	5203
3	5100	4900	4840	4860	5250	5350	5020	4880	4980	5203
4	5100	4900	4840	4860	5250	5350	5020	4880	4980	5203
5	5100	4900	4840	4860	5250	5350	5020	4880	4980	5203
6	5100	4900	4840	4860	5250	5350	5020	4880	4980	5203
7	5100	4900	4840	4860	5250	5350	5020	4880	4980	5203
8	5100	4900	4840	4860	5250	5350	5020	4880	4980	5203
9	5100	4900	4840	4860	5250	5350	5020	4880	4980	5203

Table 16 HR2000+ Blue LED measures on strawberry leaves.

We used the same methodology for the Green LED to find the factors and R-squared. The results are shown in [Fig. 57].

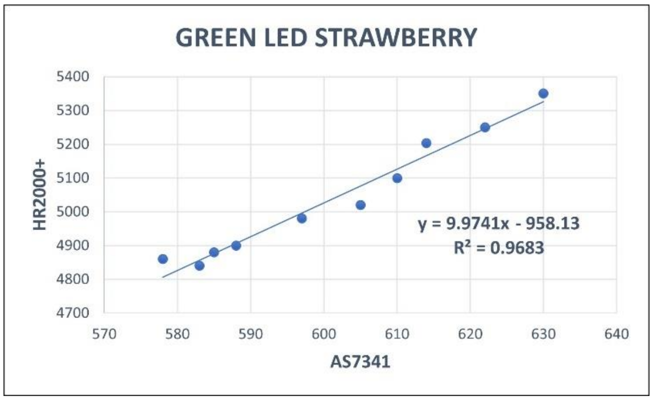

Fig. 57 Green LED linear regression regarding the strawberry leaves.

5.1.8 IR LED on strawberry leaves

For the IR LED, we have the measures shown in [Table 17,18]:

IR LED STRAWBERRY										
num	LEAF 1	LEAF 2	LEAF 3	LEAF 4	LEAF 5	LEAF 6	LEAF 7	LEAF 8	LEAF 9	LEAF 10
0	2040	2000	1980	1950	2100	2120	2040	1980	2035	2090
1	2040	2000	1980	1950	2100	2120	2040	1980	2035	2090
2	2040	2000	1980	1950	2100	2120	2040	1980	2035	2090
3	2040	2000	1980	1950	2100	2120	2040	1980	2035	2090
4	2040	2000	1980	1950	2100	2120	2040	1980	2035	2090
5	2040	2000	1980	1950	2100	2120	2040	1980	2035	2090
6	2040	2000	1980	1950	2100	2120	2040	1980	2035	2090
7	2040	2000	1980	1950	2100	2120	2040	1980	2035	2090
8	2040	2000	1980	1950	2100	2120	2040	1980	2035	2090
9	2040	2000	1980	1950	2100	2120	2040	1980	2035	2090

Table 17 Device’s IR LED measures on strawberry leaves.

IR HR2000+ STRAWBERRY										
num	LEAF 1	LEAF 2	LEAF 3	LEAF 4	LEAF 5	LEAF 6	LEAF 7	LEAF 8	LEAF 9	LEAF 10
0	14392	13892	13307	13123	14965	15300	14212	13652	14086	14628
1	14392	13892	13307	13123	14965	15300	14212	13652	14086	14628
2	14392	13892	13307	13123	14965	15300	14212	13652	14086	14628
3	14392	13892	13307	13123	14965	15300	14212	13652	14086	14628
4	14392	13892	13307	13123	14965	15300	14212	13652	14086	14628
5	14392	13892	13307	13123	14965	15300	14212	13652	14086	14628
6	14392	13892	13307	13123	14965	15300	14212	13652	14086	14628
7	14392	13892	13307	13123	14965	15300	14212	13652	14086	14628
8	14392	13892	13307	13123	14965	15300	14212	13652	14086	14628
9	14392	13892	13307	13123	14965	15300	14212	13652	14086	14628

Table 18 HR2000+ IR LED measures on strawberry leaves.

We used the same methodology for the Green LED to find the factors and R-squared. The results are shown in [Fig. 58].

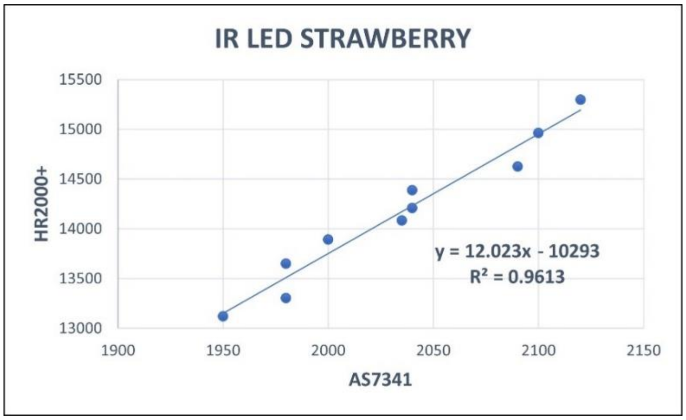

Fig. 58 IR LED linear regression regarding the strawberry leaves.

5.2 Results and conclusions

As we see the R^2 is above 0.92. This shows that the measures between the two devices can correlate with an accuracy of 92% or more. As we can see [Table 19] the blue led has the lowest correlation value. For the strawberry leaves, we have a more stable correlation for red, green, and infrared. On the other hand, for lemon leaves, we see an almost perfect correlation for infrared (0.9715), and a strong correlation for red and green LED.

LED Color	R^2 of lemon leaves	R^2 of strawberry leaves
Red	0.949	0.9697
Green	0.9478	0.9683
Blue	0.921	0.9385
Infrared	0.9715	0.9613

Table 19 R-squared results for each LED

The two devices are highly correlated. This helps us conclude that our device can be used as a reliable and accurate spectrometer that measures reflectance from surfaces and specifically leaves. While we were measuring the leaves, we also obtained the vegetation indices (shown in tables 20 & 21).

VI	SR		NDVI		EVI	
num	L	S	L	S	L	S
1	3.364379	3.756906	0.541745	0.579559	0.802195	0.921458
2	3.8	3.773585	0.583333	0.579559	0.903576	0.91967
3	3.521127	3.867188	0.557632	0.581028	0.858204	0.934318
4	3.355878	3.846154	0.54085	0.589085	0.780516	0.92905
5	3.342419	3.790614	0.539427	0.587302	0.806452	0.939818
6	3.446809	3.745583	0.550239	0.582517	0.796491	0.939086
7	3.194118	3.784787	0.523142	0.578555	0.7885	0.922557
8	3.2512	3.771429	0.529545	0.582009	0.815558	0.914864
9	2.974684	3.789572	0.496815	0.580838	0.729859	0.921166
10	3.44732	3.806922	0.550291	0.582426	0.830198	0.945979

Table 20 Vegetation indices calculated with our device. This table contains Simple Ratio, NDVI, and EVI.

VI	$NDVI_g$		$NDVI_b$		IPVI	
num	L	S	L	S	L	S
1	0.546957	0.5396226	0.853285	0.85034	0.770872	0.789779
2	0.572848	0.5455951	0.865488	0.853568	0.791667	0.790514
3	0.54202	0.5450644	0.847575	0.859155	0.778816	0.794543
4	0.522279	0.5427215	0.865712	0.858913	0.770425	0.793651
5	0.517401	0.5429831	0.846588	0.846154	0.769714	0.791258
6	0.535545	0.5418182	0.871931	0.840278	0.77512	0.789278
7	0.524746	0.5425331	0.829053	0.853703	0.761571	0.791004
8	0.536484	0.5438596	0.824057	0.85567	0.764772	0.790419
9	0.512227	0.5463526	0.81959	0.855059	0.748408	0.791213
10	0.543235	0.545858	0.851117	0.845475	0.775145	0.791967

Table 21 Vegetation indices calculated with our device. This table contains $NDVI_g$, $NDVI_b$, and IPVI.

As we can see the vegetation indices are within the ranges that we expected. This shows us that the device is behaving correctly and as expected. The vegetation indices are long enough in decimal digits so we only keep 6 for now. When the device is going to be used for many measurements, we will only keep 2 decimal digits. The results show us that we dealt with healthy plants. They also proved that we have stability because the range of the vegetation indices is not so big. That is also expected since we measure leaves from the same plant and thus, they have the same health conditions.

Our device is stable enough and the measures with each LED present linearity. The values show a small change regarding the different leaves. This is reasonable because the leaves are different and some of them have higher absorption while others have a higher reflection. This is the reason that the values change, thus for each different leaf with each different LED the results show linearity. When the e.g., the red LED has the highest or the lowest value so do the others with the only exception of some values for the blue LED [Fig. 59] & [Fig. 60]. The same conduct is observed for both plants.

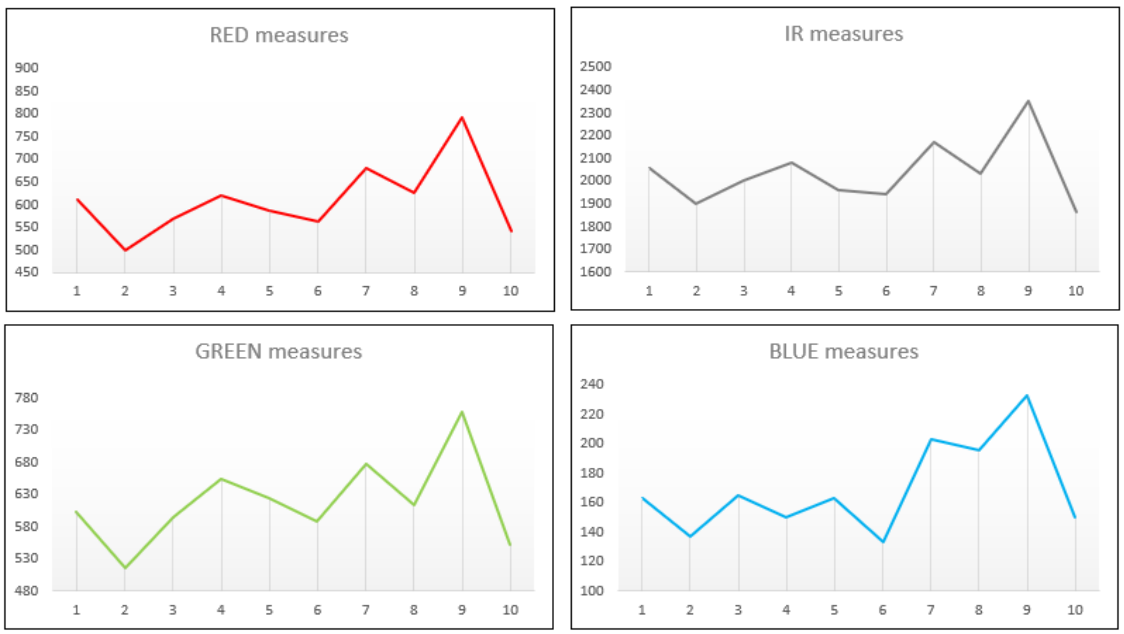

Fig. 59 LEDs measure curves for lemon leaves.

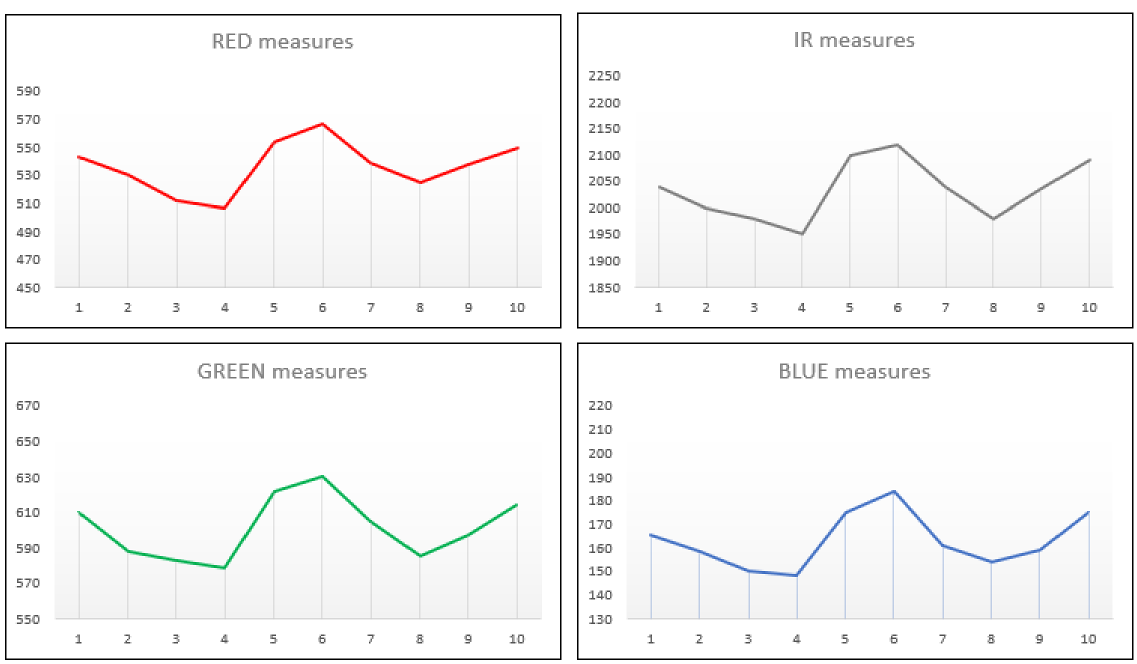

Fig. 60 LEDs measure curves for strawberry leaves.

6. Conclusions and future work

Creating a device that can measure vegetation indices by reflection is not an easy task. It takes many sciences combined to be achieved. It needs to have high precision, stability, and ease of use. It also needs an industrial device to evaluate and correlate the device's measures. It had to undergo a lot of experiments, testing, and statistical evaluation.

The simplicity of the device operation was achieved. We also managed to make the device easy to use and user-friendly. We did not use any complex menu or operation system. Also, we managed to keep the balance between low-cost and high accuracy, precision, and stability and we did it without having to resort to destructive methods of endoscopy.

Furthermore, our device performed well regarding the measures and correlated strongly with the industrial spectrometer HR2000+. AS7341 although that is a low-cost color sensor can perform well-measuring light that is reflected by surfaces and specifically leaves. AS7341 measures were linear and stable. The vegetation indices calculated by our device were within the ranges accepted and they showed diversity according to the leaf used.

In the future, we would like to create datasets by measuring many samples on plants or trees. The user will be able to choose from the menu the plant he wants to measure and then the device will correlate the measure with the existing dataset. This will make our device more accurate and reliable.

We would also like to create a 3D printed box and enclose the device inside. Another useful extension for our device would be a direct charging system, probably with the use of the TP4056 module. This will allow the device to be completely independent and requires a battery monitoring system. A battery monitoring system is a basic tool for devices like ours.

Another idea is to add two lenses, one will be an extension of the LEDs and the other will be an extension of the AS7341's eye. The first one will help us gather the light beam and center it, and the other one will help the sensor to gather more photons, it will also help to collect the light and center it in the photodiode array. This idea is also possible to be achieved by using fiber optics. Once again one optical fiber will be used to gather the light beam and center it, and the other one will be used to gather the light beam and center it in the photodiode array.

Another feature we would like to add in the future is a GPS module that will take latitude and longitude. Those values will also be stored on the SD card. This feature can be combined with a Bluetooth module. This way we can send all the data via Bluetooth to a mobile phone or a computer. There we can process the data or perform cloud computing. To end with another smart idea would be to create a heatmap by using google maps or something similar to it. The heatmap will change colors based on the measurements of the device.

Last but not least when our datasets allow us to do so, we can implement neural networks in our device to make different predictions for better fertilization, watering of the plant, or even predict an upcoming disease even in its early stages.

References

1. Yannis Nikoloudakis, Spyridon Panagiotakis, Thrasivoulos Manios, EvangelosMarkakis and Evangelos Pallis, "Composting as a Service: A Real-World IoT Implementation", Future Internet, Volume 10, Issue 11 (November 2018): https://www.mdpi.com/1999-5903/10/11/107

2. IOT and Industry 4.0: The Industrial New Revolution, Marcelo T Okano: https://www.researchgate.net/profile/Marcelo-Okano-2/publication/319881057_IOT_and_Industry_40_The_Industrial_New_Revolution/links/59c018a5aca272aff2e20639/IOT-and-Industry-40-The-Industrial-New-Revolution.pdf

3. A review of Internet of Things (IoT) embedded sustainable supply chain for industry 4.0 requirements: https://www.sciencedirect.com/science/article/pii/S0360835218305709

4. Koralia Papadokostaki, George Mastorakis, Spyros Panagiotakis, "Handling Big Data in the era of IoT", chapter contribution in the "Advances in Mobile Cloud Computing and Big Data in the 5G Era", Editors: Mavromoustakis Constandinos X., Mastorakis George, Dobre Ciprian, Springer-Verlag (2017), pp. 3-22.

5. Georgios Skourletopoulos, Constandinos X. Mavromoustakis, George Mastorakis, Jordi Mongay Batalla, Ciprian Dobre, Spyros Panagiotakis, Evangelos Pallis, "Big Data and Cloud Computing: A Survey of the State-of-the-Art and Research Challenges", chapter contribution in the "Advances in Mobile Cloud Computing and Big Data in the 5G Era", Editors: Mavromoustakis Constandinos X., Mastorakis George, Dobre Ciprian, Springer-Verlag (2017), pp. 23-41.

6. Prokopis Vavouranakis, Spyros Panagiotakis, George Mastorakis, Constandinos X. Mavromoustakis, "Smartphone-Based Telematics for Usage Based Insurance", chapter contribution in the "Advances in Mobile Cloud Computing and Big Data in the 5G Era", Editors: Mavromoustakis Constandinos X., Mastorakis George, Dobre Ciprian, Springer-Verlag (2017), pp. 309-339.

7. Machine learning in manufacturing and industry 4.0 applications:

https://www.tandfonline.com/doi/full/10.1080/00207543.2021.1956675

8. From Artificial Intelligence to Explainable Artificial Intelligence in Industry 4.0: A Survey on What, How, and Where:

https://www.sciencedirect.com/science/article/pii/S1877050921001393

9. EDGE COMPUTING VS. CLOUD COMPUTING: CHALLENGES AND OPPORTUNITIES IN INDUSTRY 4.0:

https://www.daaam.info/Downloads/Pdfs/proceedings/proceedings_2019/120.pdf

10. Convergence of Blockchain and Edge Computing for Secure and Scalable IIoT Critical Infrastructures in Industry 4.0: https://ieeexplore.ieee.org/abstract/document/9203896

11. Vakintis, Ioannis, Spyros Panagiotakis, "Middleware Platform for Mobile Crowd-Sensing Applications using HTML5 APIs and Web Technologies", chapter contribution in the HandBook "Internet of Things (IoT) in 5G Mobile Technologies", Editors: Mavromoustakis, Constandinos X., Mastorakis, George, Batalla, Jordi Mongay, Springer-Verlag (2016), pp. 231-274.

12. Prokopis Vavouranakis, Spyros Panagiotakis, George Mastorakis, Constandinos X. Mavromoustakis, "Smartphone-Based Telematics for Usage Based Insurance", chapter contribution in the "Advances in Mobile Cloud Computing and Big Data in the 5G Era", Editors: Mavromoustakis Constandinos X., Mastorakis George, Dobre Ciprian, Springer-Verlag (2017), pp. 309-339.

13. Nikoloudakis, Y.; Kefaloukos, I.; Klados, S.; Panagiotakis, S.; Pallis, E.; Skianis, C.; Markakis, E.K., "Towards a Machine Learning Based Situational Awareness Framework for Cybersecurity: An SDN Implementation", MDPI Sensors 2021, 21, 4939: https://doi.org/10.3390/s21144939

14. Virtual reality in the automotive field in industry 4.0: https://www.sciencedirect.com/science/article/pii/S2214785320397224

15. Smart Facility Management Systems Utilizing Open BIM and Augmented/Virtual Reality: https://www.proquest.com/openview/f87405ec152f276bc6058c8d018c60f3/1?pq-origsite=gscholar&cbl=1646340

16. Robotic Process Automation and Artificial Intelligence in Industry 4.0 – A Literature review: https://www.sciencedirect.com/science/article/pii/S1877050921001393

17. Advanced robotics and additive manufacturing of composites: towards a new era in Industry 4.0: https://www.tandfonline.com/doi/full/10.1080/10426914.2020.1866195

18. Disruptive Technologies and Operations Management in the Industry 4.0 Era and Beyond: https://onlinelibrary.wiley.com/doi/abs/10.1111/poms.13622

19. Implementation of Artificial Intelligence and Robotics that Replace Employees in Indonesia: https://link.springer.com/chapter/10.1007/978-981-19-2541-2_61

20. How Virtualization, Decentralization, and Network Building Change the Manufacturing Landscape: An Industry 4.0 Perspective: https://picture.iczhiku.com/resource/paper/WyIStjidhauGDMvc.pdf

21. SDN-enabled network virtualization for industry 4.0 based on IoTs and cloud computing: https://ieeexplore.ieee.org/abstract/document/7890083

22. AGRICULTURE 4.0 – THE FUTURE OF FARMING TECHNOLOGY: https://www.oliverwyman.com/our-expertise/insights/2018/feb/agriculture-4-0--the-future-of-farming-technology.html

23. Indoor Vertical Farming in the Urban Nexus Context: Business Growth and Resource Savings: https://www.researchgate.net/publication/339700059

24. Vertical farming: The future of agriculture: A review:

https://www.researchgate.net/publication/358749034

25. Computational analysis of the environment in an indoor vertical farming system:

https://www.sciencedirect.com/science/article/pii/S0017931021015581

26. Agriculture automation system: https://www.researchgate.net/publication/347857102

27. The Rise of the Drones in Agriculture: https://www.researchgate.net/publication

28. Michael Kalochristianakis, Dimitris Katrinakis, Georgia Atsali, Athanasios Malamos, Thrasyvoulos Manios, Spyros Panagiotakis, "HOLISTIC: an IoT system for residential water recycling based on open-source technologies", in proceedings of the 7th IEEE International Conference on Telecommunications and Multimedia (TEMU) 2016, Heraklion, Crete, Greece, July 25-27, 2016.

29. Research on Innovative Training on Smart Greenhouse Technologies for Economic and Environmental Sustainability: https://www.mdpi.com/2071-1050/13/19/10536

30. Special Issue: Digital Plant Pathology for Precision Agriculture: https://link.springer.com/article/10.1007/s41348-022-00620-9

31. The Role of Remote Sensing Data and Methods in a Modern Approach to Fertilization in Precision Agriculture: https://www.mdpi.com/2072-4292/14/3/778

32. Drivers of Precision Agriculture Technologies Adoption: A Literature Review: https://www.sciencedirect.com/science/article/pii/S2212017313000728

33. Farmers and the agricultural labor force – statistics: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Farmers_and_the_agricultural_labour_force_-_statistics#Agriculture_remains_a_big_employer_within_the_EU.3B_about_9.7_million_people_work_in_agriculture

34. Evaluation of Broadband and Narrowband Vegetation Indices for the Identification of Archaeological Crop Marks: https://www.mdpi.com/2072-4292/4/12/3892

35. Responses of leaf spectral reflectance to plant stress: https://bsapubs.onlinelibrary.wiley.com/doi/epdf/10.1002/j.1537-2197.1993.tb13796.x

36. MICROCLIMATIC LANDSCAPE DESIGN: CREATING THERMAL COMFORT AND ENERGY EFFICIENCY. R. D. Brown and T. J. Gillespie, John Wiley & Sons Inc. (New York), 1995.

37. Intercropping Enhances Productivity and Maintains the Most Soil Fertility Properties Relative to Sole Cropping: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0113984

38. Cropping systems in agriculture and their impact on soil health-A review: https://www.sciencedirect.com/science/article/pii/S2351989420304790?via%3Dihub

39. Emerging Technologies for Monitoring Plant Health in Vivo:

https://pubs.acs.org/doi/10.1021/acsomega.0c05850

40. Nanosensor Detection of Synthetic Auxins In Planta using Corona Phase Molecular Recognition: https://pubs.acs.org/doi/10.1021/acssensors.1c01022

41. Quantitative analysis of major plant hormones in crude plant extracts by high-performance liquid chromatography–mass spectrometry:

https://www.nature.com/articles/nprot.2010.37

42. Change Detection of Vegetation Cover by NDVI Technique on Catchment Area of the Panchet Hill Dam, India:

https://www.researchgate.net/publication/339943342_Change_Detection_of_Vegetation_Cover_by_NDVI_Technique_on_Catchment_Area_of_the_Panchet_Hill_Dam_India

43. Jones C: L ; P:R: Weckler, N:O:Maness, M:L: Stone and R. Jayasekara, 2004. Estimating water stress inn plant using hyperspectral sensing. Annual International Meeting: https://www.researchgate.net/publication/251783293

44. Liu L., J. Wang, W.Huang, C.Zhao, B.Zhang and Q. Tong, 2004. Estimating Winter Wheat plant water content using Red Edge Parameters. International Journal of remote Sensing: https://www.tandfonline.com/doi/abs/10.1080/01431160310001654365

45. Katsoulas N., A.Evlanidi, K.P.Ferentinos, T.Bartzanas, C. Kittas, 2014. A hyperspectral imaging system for plant stress detection: calibration and preliminary results. 25th International Scientific – Experts Congress on Agriculture and Food Industrz, September 25-27, Cesme-Izmir. Turkey: https://www.aua.gr/~ferentin/papers/Agricongress2014-Katsoulas_et_al.pdf

46. LU Yan-li, BAI You-lu, YANG Li-ping, WANG Lei (2008). Diagnosis on Nitrogen Status Using GreenSeeker in Spring Maize: https://agris.fao.org/agris-search/search.do?recordID=CN2009002448

47. HU Hao, BAI You-Lu, YANG Li-Ping, LU Yan-Li, WANG Lei, WANG He, WANG Zhi-Yong, (2010). Diagnosis of nitrogen nutrition in winter wheat (Triticum aestivum) via SPAD-502 and GreenSeeker: https://www.researchgate.net/publication/250262078

48. A remotely sensed pigment index reveals photosynthetic phenology in evergreen conifers: https://www.pnas.org/doi/10.1073/pnas.1606162113

49. Overview of the radiometric and biophysical performance of the MODIS vegetation indices: https://www.sciencedirect.com/science/article/abs/pii/S0034425702000962?via%3Dihub

50. A new low-cost portable multispectral optical device for precise pant status assessment: https://www.sciencedirect.com/science/article/pii/S0168169919301218#b0010

51. Near-Infrared Spectroscopy Technology for Soil Nutrients Detection Based on LS-SVM: https://tel.archives-ouvertes.fr/IFIP-AICT-368/hal-01351828

52. Sensor Technologies for precision Soil Nutrient Management and Monitoring:

https://www.researchgate.net/publication/287626658_Sensor_Technologies_for_Precision_Soil_Nutrient_Management_and_Monitoring

53. Vis–NIR-based optical sensor system for estimation of primary nutrients in soil:

https://link.springer.com/article/10.1007/s12596-019-00517-1

54. Estimation of Leaf Area Index with a Multi-Channel Spectral Micro-Sensor for Wireless Sensing Networks: https://www.mdpi.com/1424-8220/22/13/5048/htm

55. A Novel Approach to Obtain PAR with a Multi-Channel Spectral Microsensor, Suitable for Sensor Node Integration: https://pubmed.ncbi.nlm.nih.gov/34068029/

56. IR LED DATASHEET:https://www.tme.eu/gr/details/ll-503sirc-1bd/diodoi-led-ir/luckylight/ll-503sirc2e-1bd/

57. Red LED Datasheet:https://www.tme.eu/gr/details/l-53src-j4/diodoi-led-tht-5mm/kingbright-electronic/

58. Green LED Datasheet: https://www.tme.eu/gr/details/ll-504pgc2e-g5-1ac/diodoi-led-tht-5mm/luckylight/

59. Blue LED Datasheet: https://www.tme.eu/gr/details/ll-504pgc2e-g5-1ac/diodoi-led-tht-5mm/luckylight/

60. HR2000+ Datasheet: https://www.oceaninsight.com/globalassets/catalog-blocks-and-images/manuals--instruction-old-logo/spectrometer/hr2000-.pdf