Albin Westerlund

Design and Implementation of an Autonomous Groundwater Monitoring System for Remote Deployment Using Low-Power IoT Technology

School of Technology and Innovations
Master's thesis in Automation and Computer Science
Energy and Information Technology

Vaasa 2025

Acknowledgements

As this thesis comes to an end, I would like to thank my supervisor Professor Mohammed Elmusrati, and my instructor Mikael Palosaari for their guidance and feedback.

I'm also grateful to Omicron Ceti Ab and Läkarmissionen International for giving me this opportunity.

Lastly, I would like to give a big thank you to my family for all their valuable support throughout my academic journey.

UNIVERSITY OF VAASA
School of Technology and Innovations
Author: Albin Westerlund
Title of the Thesis: Design and Implementation of an Autonomous Groundwater Monitoring System for Remote Deployment Using Low-Power IoT Technology
Degree: Master of Science in Technology
Discipline: Master's Programme in Computing Sciences
Supervisor: Mohammed Elmusrati
Evaluator: Timo Mantere
Instructor: Mikael Palosaari
Year: 2025 **Pages:** 123

ABSTRACT:
Several regions in Africa are dependent on groundwater as a source of water. Having access to a groundwater supply can additionally mitigate water shortage during droughts. Läkarmissionen International, a Swedish NGO, provides humanitarian aid to refugee camps and rural communities including access to the groundwater table through wells and boreholes. To use groundwater sustainably, its level must be monitored continuously over long periods to identify its behavior from extraction and to predict level trends. Due to the high cost of commercial systems, Läkarmissionen International is interested in a more affordable approach for a groundwater monitoring solution that would meet their specific needs for use in challenging conditions inside active boreholes. This thesis explores the possibility of using low-cost and widely available components for an energy-efficient groundwater monitoring system while still meeting the set specifications. After evaluating existing methods and solutions available on the market, hydrostatic pressure sensing was determined as the most promising method. A groundwater logger prototype was then developed around an ESP32-S2 microcontroller chosen for its low cost, low energy consumption, high availability, and supported onboard wireless communication capabilities. A 4-20 mA submersible gage-type pressure sensor with an attached vent tube and communication wires was chosen for its simplicity and ability to automatically compensate for atmospheric pressure. After evaluating the performance of the prototype under real and simulated conditions, the prototype demonstrated its potential, however its performance did not meet the application specifications. To address the identified shortcomings, modifications are suggested for a new system layout, hardware components, and firmware functionality to improve the performance of the prototype.

KEYWORDS: Groundwater Monitoring, Low-Power, IoT, ESP32-S2, Wireless Communication

VASA UNIVERSITET
Fakultet inom Teknologi och Innovation
Författare: Albin Westerlund
Titel: Design and Implementation of an Autonomous Groundwater Monitoring System for Remote Deployment Using Low-Power IoT Technology
Examen: Diplomingengörsexamen
Läroämne: Master's Programme in Computing Sciences
Övervakare: Mohammed Elmusrati
Utvärderare: Timo Mantere
Instruktör: Mikael Palosaari
År: 2025 **Sidor:** 123

ABSTRAKT:
Flera regioner i Afrika är beroende av grundvatten som vattenkälla. Att ha tillgång till grundvatten kan dessutom mildra vattenbristen under torka. Läkarmissionen International, en svensk icke-statlig organisation, ger humanitärt bistånd till flyktingläger och landsbygdssamhällen bland annat genom att ge tillgång till grundvatten genom brunnar och borrhål. För att använda grundvatten på ett hållbart sätt måste vattennivån övervakas kontinuerligt under långa perioder för att identifiera nivåförändringar vid användning och för att förutspå trender. På grund av de höga kostnaderna för kommersiella system är Läkarmissionen International intresserad av ett mera prisvärt tillvägagångssätt för en grundvattenövervakningslösning som skulle uppfylla deras specifika behov för användning under krävande förhållanden inuti aktiva borrhål. I detta examensarbete undersöks möjligheten att använda billiga och lätt tillgängliga komponenter för ett energieffektivt övervakningssystem för grundvatten samtidigt som funktionskraven uppfylls. Efter att ha utvärderat befintliga metoder och lösningar som finns tillgängliga på marknaden fastställdes hydrostatiskt tryck som den mest lovande metoden. En grundvattenlogger prototyp utvecklades med hjälp av en ESP32-S2 mikrokontroller som valdes för sin låga kostnad och energiförbrukning, enkel tillgänglighet och stöd för trådlös kommunikation. En vattentät 4-20 mA manometertrycksensor med ventilationsrör och kommunikationskablar valdes på grund av sin enkelhet och förmåga att automatiskt kompensera för atmosfärstryck. Efter att ha testat prototypen under verkliga och simulerade förhållanden visade den potential men uppfyllde inte prestandakraven. För att förbättra de identifierade bristerna föreslås en ny systemlayout, hårdvarukomponenter och inbyggd programvara för att förbättra prototypens prestanda.

NYCKELORD: Groundwater Monitoring, Low-Power, IoT, ESP32-S2, Wireless Communication

Contents

Figures

Tables

Algorithms

Abbreviations

LMI	Läkarmissionen International
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
IoT	Internet of Things
MCU	Microcontroller Unit
ToF	Time of Flight
FZP	Fresnel Zone-Plate
PLA	Polylactic Acid
GNSS	Global Navigation Satellite System
UAV	Unmanned Aerial Vehicle
GPR	Ground Penetrating Radar
UART	Universal Asynchronous Receiver Transmitter
I2C	Inter-Integrated Circuit
SPI	Serial Peripheral Interface
SoC	System on a Chip
CPU	Central Processing Unit
PMU	Power Management Unit
GPIO	General-Purpose Input Output
ULP	Ultra Low Power
ADC	Analog-to-Digital Converter
BMS	Battery Management System
FFD	Fractional Frequency Deviation
PCB	Printed Circuit Board
RTC	Real Time Clock
DFS	Dynamic Frequency Scaling
ML	Machine Learning

1 Introduction

This thesis is done in cooperation with Omicron Ceti Ab and Läkarmissionen International (LMI), based in Stockholm, Sweden. LMI is a Swedish NGO active in 25 countries globally working in humanitarian and developmental contexts. The goal of this thesis is to research groundwater measuring solutions and techniques suitable for LMI's applications in Africa. One of LMI's focus areas is water and sanitation work in sub-Saharan countries. The work includes water supply for large refugee camps as well as for rural communities. It includes siting and drilling of boreholes, installations of pumps, reservoirs and pipe networks while also training stakeholders in long-term operation and maintenance of water points (LM International, 2024). The requirements and limitations of possible solutions will be presented in the following chapter.

According to The African Development Bank, by 2025, 25 African countries will suffer from ongoing water shortages with half of the rural population lacking access to clean drinking water (Boucher, 2005). Lack of water can cause unrest among the population and between countries, especially in locations where different countries utilize the same transboundary aquifer (Robins & Fergusson, 2014). Using groundwater as a water source can mitigate the difficulties caused by droughts where surface water is inadequate. Many countries in Africa, especially rural population centers, sometimes depend to a high degree on groundwater for agriculture, industries, and domestic use, making access to safe drinking water essential (Farr et al., 2005).

Monitoring groundwater levels offers numerous benefits and is important for several reasons. According to Fulton et al. (n.d.), groundwater levels can change over time due to pumping, climate change, or natural environmental events. Although small changes in the groundwater level are difficult to spot, analyzing long-term time series of groundwater levels for different aquifers can reveal positive or negative trends. This data can identify areas with underutilized groundwater resources or serve as an early warning system for future groundwater depletion allowing governments and organizations time to act. Additionally, monitoring groundwater levels can provide insights into the flow

direction of groundwater, the rate at which water flows inside the aquifer, and subsequently the volume of groundwater that can be utilized. Furthermore, a model of the groundwater level and flow direction can be created with the help of data from groundwater monitoring. This resource is valuable when determining suitable locations for future boreholes (Fulton et al., n.d.). For LMI, continuous monitoring of groundwater levels is important to provide a baseline on natural and seasonal variations to help understand possible impacts of climate change effects.

Only a small volume of all existing groundwater can be used without detrimental consequences according to National Academies (2022). Overuse of groundwater resources may result in a reduction of baseflow to rivers and water reservoirs causing them to shrink. Unmonitored and unregulated groundwater pumping activity near saltwater shorelines is one factor that can lead to saltwater intrusion into the freshwater aquifer system according to Water Resources Mission Area (2019). Salt in the water supply makes it unsuitable for farming and drinking. A declining groundwater level caused by overuse can alter the flow of freshwater which has a natural tendency to flow towards the sea. This causes the zone of transition between freshwater and saltwater to move inland closer to pumping stations (Water Resources Mission Area, 2019). In Stumm & Comos' report from 2017, they state that the limited groundwater recharge on Manhattan Island, New York, has made it incapable of moving the saltwater front back towards the sea to any significant extent, despite the fact industrial pumping of groundwater stopped over 70 years ago (Stumm & Como, 2017).

There exist several different methods and techniques for measuring groundwater levels. Most of the research supports in large the utilization of borehole-based measuring methods and techniques for measuring groundwater levels over short and long periods. These are, although not limited to, measuring water pressure with a transducer, measuring depth to water level using a measuring device lowered down from the surface (Fulton et al., n.d.), measuring air pressure caused by the rising water level inside an air line (Cunningham & Schalk, 2011), and measuring depth to water by reflecting a sound signal

off of the water surface (Solinst Canada Ltd., 2024). Geoelectrical (Wilopo et al., 2020) and Ground penetrating radar (Salih et al., 2022) are examples of nonintrusive surface-based approaches. Measuring changes in gravitational pull caused by moving groundwater utilizing orbiting satellites is a novel solution for measuring groundwater but is beyond the scope of this thesis (National Academies, 2022). No matter which commercially available measuring solution of the previously mentioned methods is considered, they all face the challenge of expensive equipment and installation which constitutes a significant barrier for groundwater level measuring.

2 Background Information

This thesis is made for LM International (LMI), which is a global nonprofit organization providing humanitarian aid and basic human needs including water, sanitation, and hygiene for communities in need in several different countries. The focus of this thesis and its applications is primarily directed at LMI's operations in Sahel and Horn of Africa regions. In these regions, LMI collaborates with and supports different countries to achieve the sustainable water and sanitation goals set out by the United Nations. With the help of donors, LMI can among other things help communities and refugee camps gain access to clean drinking water by drilling and constructing wells that provide access to the groundwater table if no other water source is suitable.

2.1 Limitations to consider

After discussions with LMI, it became evident that they face several challenges in their operations in the region. These challenges impose requirements and limitations on infrastructure and potential groundwater measuring solutions. Rivalries between communities and villages are one problem leading to intentional malicious damage to wells and associated infrastructure causing operation problems, repair costs, and extensive waiting times depending on the location. To minimize deliberate and unintentional destruction of surface equipment, LMI has experimented with installing solar panels and other equipment elevated from the ground on roofs or socially important buildings to make them less accessible. This problem sets requirements for the design of the sensor system such as size, durability, longevity, as well as where it can be installed. Placing the sensor inside the well can make it less accessible, however, in cases where it is not possible, an inconspicuous design of the system is advantageous to avoid drawing attention to it.

The remote nature of some boreholes as well as the long distance between them puts limitations on data communication due to lacking or unreliable phone service, making wireless communication of data and information challenging. The electric grid poses

another challenge in these remote locations far from bigger population centers with more developed infrastructure. It is sporadic, unreliable, or lacking completely making it unsuitable to entirely rely on. This calls for an independently powered solution. LMI circumvents this problem by using electricity produced directly from photovoltaic solar panels to operate the pumping equipment where manually powered pumps are not installed. The use of batteries as energy storage is avoided to cut costs of installation and maintenance. Another limitation to consider when designing a sensor system is the local culture and superstition, consequently making the use of cameras for any type of data acquisition unfavorable. The general technical knowledge in the remote areas where LMI operates is limited, meaning the sensor system should be able to operate reliably for long periods requiring minimal maintenance once they are installed.

LMI is currently testing in limited numbers a commercially available submersible groundwater level sensor called the TD-Diver. This sensor's relatively high initial investment has made LMI interested in more cost-effective solutions to lower the economic barrier to groundwater monitoring. Depending on the specific model, the measurement accuracy of the TD-Diver is from ±0.5 cm to ±5 cm, the maximum range is 100 m, and with a battery life of up to 10 years according to its manufacturer (Van Essen Instruments, 2024). Since the accuracy this sensor provides exceeds the requirements of LMI's applications, they have decided to prioritize affordability over accuracy.

When creating a groundwater monitoring program, it is important to utilize several observation wells unaffected by human activities to be sure that the collected data is of high quality and represents groundwater conditions in diverse environments and different aquifers (Taylor & Alley, 2001). Although this is the optimal approach to measuring groundwater levels, LMI has opted to deploy the TD-Diver sensor in some of their active boreholes in Chad to avoid the additional cost resulting from drilling boreholes dedicated to monitoring. This is due to restrictions in available funding.

2.2 Typical well application in Sahel and Horn of Africa regions

There are some variations in the well and pump design used in Chad. A typical drilled borehole has a diameter of 10 to 20 cm and a depth of down to 200 meters. The borehole is drilled down to a depth where the local groundwater aquifer or aquifers produce enough water to satisfy the needs of the community located near the well. The depth of where the groundwater table begins can vary greatly between different geographical locations. In the case of Chad, the groundwater table can be expected to begin immediately below ground level in the southern region, and in the north, groundwater can be expected at depths of more than 250 meters according to the British Geological Survey (2011).

Both manually operated and electrically powered pumps are used. Manually operated pumps only require one pipe inside the well to extract water resulting in more space inside the borehole compared to using electrical submersible pressure pumps. This type of pump is lowered down the borehole until it reaches a suitable depth under the groundwater surface and is then used to pump the groundwater. The electricity used to power the pumps is generally provided directly by photovoltaic panels mounted on the surface. This design, without an energy storage solution, saves on installation and maintenance costs but only allows pumping to take place during the day when the sun provides enough energy for reliable operation. This leads to fluctuations in the groundwater level during the day in the case of electric pumps. Water levels in boreholes installed with manual pumps experience water level fluctuations whenever the pump is used and are not affected by the time of day. At night and during other times when the sunlight is insufficient for electricity production and the electric pumps are offline, the water level inside the boreholes starts to rise and, after enough time, recovers to the stable level of the surrounding aquifer. If regular and heavy pumping takes place at locations with low flow rate aquifers, a stable water level might never be reached. Fluctuations in the groundwater level caused by seasonal events are expected to be low compared to fluctuations caused by pumping activities. The water level inside an active borehole is estimated to change by up to 20 meters in height daily. An electric pumping

installation consists of the pump itself, an electrical cord, and a water raising pipe made of PE, PVC or steel with a diameter of around 50-100 mm which is used to transport water from the pump to a surface storage tank. This tank serves as a buffer storage from which water is then used.

LMI operates in different locations within Chad to support communities in need of water. They range from more permanent small villages, which have some resources, electricity, and pre-existing wireless communication capabilities, to refugee camps located in arid desert environments with minimal natural resources and no permanent infrastructure. According to the Climatic Research Unit (2024) of the University of East Anglia, the average surface air temperature in Chad between 1991 and 2020 varies from below 15 °C to over 40 °C, with precipitation ranging from negligible to over 120 mm in a single month depending on location and time of year. In these conditions, the expected lifetime of a submersible pressure pump is between 5 and 10 years, with some pumps lasting up to 15 years in service.

2.3 Specific application requirements

The main priority for LMI is to get data on the groundwater level inside their boreholes to get a better picture of the water usage, long-term groundwater level trends, and an early warning signal of decreasing water levels to prevent boreholes from running dry and subsequent water shortage. A measurement accuracy of 10 cm is deemed sufficient for these purposes. Currently, their knowledge of the condition of the boreholes after construction is limited. Implementing a cost-effective sensor solution that makes it possible to install in both newly constructed and older boreholes can provide insight into this problem but requires the sensor package to be versatile and adaptable across various settings.

The limited space inside a borehole, typically 10 to 20 cm in diameter and partly occupied by the pumping equipment, creates limitations on the sensor design in terms of size.

Although no specific target size is set, the TD-Diver submersible sensor currently being tested by LMI measuring 22 mm in diameter and 110 mm in length (Van Essen Instruments, 2024) serves as a benchmark for size considerations. Generally, a smaller design offers increased flexibility and can be adapted for a wider range of boreholes. To prevent entanglement between the sensor and its safety wire and pumping equipment, which can be especially problematic in deeper spiraling boreholes, LMI uses a PE or PVC pipe to guide the sensor down to a suitable water depth. Since the sensor system should ideally be installed inside the borehole casing or guided down through the hose to prevent easy access, it must be capable of operating in harsh conditions for extended periods. Components that are submerged must be waterproof, while other parts not in direct contact with water can still experience water splatter or moisture. Therefore, the entire system must be water-resistant to ensure maximum service life. Additionally, any component that has a possibility to contaminate the water supply must be constructed from non-toxic materials.

The minimum service life of the sensor is not specified, however, minimizing downtime and service expenses is important when considering the price and quality of individual components. More expensive components constructed to a higher standard can achieve a longer lifespan but would also lead to an increase in price. Contamination of sensors that are submerged or otherwise in contact with groundwater is a possible problem that can affect their measurement accuracy, and service life, and create maintenance needs. In this regard, a noncontact sensor solution is more favorable. When assessing the service life for a sensor in terms of battery life and storage capacity, the TD-Diver submersible sensor serves as a useful comparison. It can store 72000 data points consisting of date and time, pressure, and temperature, and can achieve a battery life of up to 10 years depending on temperature and usage, for example measuring frequency (Van Essen Instruments, 2024). The TD-Diver’s data storage of 72000 data points is approximately 8 years of continuous operation if a measurement is taken once every hour. Measuring frequency is a determining factor when designing a sensor’s data storage capabilities, including backup storage. A higher measurement frequency results in more

data and requires increased storage capabilities. As many sensors will be deployed in remote locations, frequent service visits are not possible. Therefore, the storage needs to be sufficiently sized to store measurements taken over extended periods without the need for frequent manual data unloading if the sensor has no wireless communication capability.

Equally important as the storage capacity is how to power the sensor independently for extended periods. Although the pumping equipment is powered by photovoltaic solar panels on the surface, visible or otherwise easily accessible components should be avoided to minimize the risk of vandalization. Hiding the sensor system inside the borehole would make it less accessible but also require an integrated energy storage solution. Energy storage in remote IoT devices is commonly made of primary or secondary chemical batteries but recent advances in supercapacitor research have shown its potential as IoT energy storage according to Hasan et al. (2023). Despite this potential, supercapacitors will not be considered for this application due to their low volumetric and specific energy densities. A novel solution for powering the sensor would be to draw a small amount of electricity from the electricity supply of the pumping equipment if present. This energy source could power the sensor directly or charge a small battery pack that can power the sensor at all times.

The expected battery life of a sensor system can vary depending on its chosen battery configuration. When deciding on a battery configuration, several factors need to be considered. They include battery chemistry, size, specific and volumetric energy density, power delivery, longevity, safety, rechargeability, and specific energy consumption of the IoT platform to be powered (Hasan et al., 2023). Rechargeable batteries can give a system a longer service life since the batteries can be recharged several times but require more frequent maintenance compared to a non-rechargeable battery with higher energy density. Energy consumption of an IoT device when active can be several times greater compared to when it is idle and sleeping in a low-power state (Klements, 2020). Maximizing the time a system is spent in a low-power state is important to achieve the longest

possible battery life. Configuring a sensor system to enter a low-power state between measurements, along with less frequent measurements, results in decreased energy consumption and a longer battery life. IoT devices enabled with wireless communications generally have higher energy needs due to wireless communication many times being the most power-intensive component in a system (Hasan et al., 2023).

Selecting an appropriate method for retrieving data from the deployed sensors is an important decision in a groundwater monitoring program. Wireless communication between sensors and a central location used for data collection, analysis, and storage is the optimal solution in terms of efficiency and minimizing manual labor. Sensors having the ability to wirelessly communicate a live data feed to a central location and receive software updates enables faster response times to sensor issues and easier software maintenance if an error occurs after deployment. A General Packet Radio Service (GPRS) or Global System for Mobile Communications (GSM) modem enables a sensor to be connected to cellular networks for long-distance wireless communication (Teguh & Usup, 2021), however, in locations where network infrastructure is lacking or unreliable, other approaches need to be explored.

Satellite-based wireless communication technologies that provide wide coverage are a viable option for remote locations lacking other ground-based means of wireless communication. According to Chen et al. (2023), a new and innovative satellite network technology specifically designed to cater to Internet of Things (IoT) devices is emerging. Satellite IoT is still in its infancy and more research is needed to solve problems this technology suffers from. Widespread wireless IoT devices in remote places operating from limited power sources benefit from the strict design requirements set upon the satellite IoT technology in terms of low cost and low-power consumption (Purivigraipong et al., 2020).

An alternative approach to retrieving data from a sensor is to use a local smartphone or other smart device capable of working as a relay. The sensor transmits data to a

smartphone using low-power, short-range wireless communication, and the next time the smartphone connects to a cellular network, the data is automatically transferred to a database. This method is particularly beneficial for sensors in remote locations where wireless network service is not available, as it eliminates the need for a qualified technician to travel to the sensor to offload data. The process could be automated with software and would require minimal technical knowledge for the person performing the task.

Wireless sensor network technology is worth considering in locations with several boreholes but only some of them are in reach of cellular networks. Sensors within range of other sensors in a network are interconnected with each other and can relay data between themselves. This allows sensors without the possibility of a direct cellular network connection to relay their data through other sensors to reach a sensor node connected to a cellular network. This node can then forward all the data to a database using its connection.

3 Review of Existing Methods

This chapter discusses existing and potential methods, techniques, and ideas for measuring a water level, and the potential for use inside an active borehole. Several methods have been proven to work for water level measuring applications, but using existing technologies in active boreholes creates new challenges. Size limitations, obstructions, electromagnetic interference, measuring depth from surface level, and drawdown distance are some challenges that must be considered.

3.1 Hydrostatic pressure

One approach to measuring water levels is by using hydrostatic pressure. Hydrostatic pressure sensors are a well-established technology and are used by several industries. This type of sensor works by measuring the pressure of a column of liquid directly above the sensor, which is at a known depth measured from the surface according to Behera et al. (2022). The height of the liquid column above the sensor can then be derived from the hydrostatic pressure using the following formula

$$h = \frac{p}{\rho * g}, \tag{1}$$

where h is the distance from the surface of the liquid to the point of measured pressure p in Pascal, the density of the liquid is represented by ρ, and gravitational acceleration is represented by g (Behera et al., 2022). Depending on the sensor configuration, the measured hydrostatic pressure can be the absolute pressure which includes both the pressure of the liquid and the atmospheric pressure which must be subtracted to get an accurate distance (Van Essen Instruments, 2024).

Several different variations in the operating principle exist. Some common principles include the piezoresistive effect, piezoelectric effect, capacitance change, and strain gauges (RayMing, 2023). These types of sensors function by measuring the force applied

from a liquid through a liquid-proof diaphragm onto a pressure-sensitive material. The electrical characteristics of the material change when it interacts with the pressure through bending or deforming. This physical change of the material causes it to change in capacitance, or to produce a voltage differential which can then be measured (RayMing, 2023). Another operating principle relies on giant magnetoresistance. This type of sensor uses a Hall effect sensor to measure changes in the magnetic field caused by a moving magnet attached to a deformable diaphragm (Behera et al., 2022).

Pressure sensors come in three different categories called gauge, differential, and absolute based on what the sensor pressure reading is referenced to. Absolute pressure is a pressure measured relative to an absolute vacuum. Gauge pressure is a measurement of the difference in pressure between the applied pressure of a medium and the ambient pressure surrounding the sensor, usually atmospheric pressure. A differential pressure sensor measures the difference in pressure between two locations in a system and is unaffected by changes in the surrounding pressure or the atmospheric pressure. This type of sensor can be used to determine if the flow of a medium in a pipe is obstructed, for example by a closed valve, by measuring the pressure difference before and after. The greater the obstruction, the greater the pressure difference (Avnet Abacus, n.d.).

Hydrostatic pressure sensors offer some advantages over other water level measuring techniques such as ultrasonic and laser according to Behera et al. (2022). In active borehole applications, rapid changes in water level and pumping can introduce bubbles and foam into the water which can interfere with ultrasonic and laser-based sensors since they rely on a clean water surface for signal reflection. According to the author, hydrostatic pressure sensors remain unaffected by this. Zheng et al. (2024) mention the problem with electromagnetic interference on strain gauges and piezoresistive-based pressure sensors. This can cause significant problems in active boreholes that are equipped with electrical pumping equipment. Zheng et al. (2024) suggest shielding and filtering to mitigate these effects. Temperature variations are another cause of accuracy deviation and need to be taken into account according to Keeland et al. (1997), however, they state

that groundwater temperature is expected to remain relatively stable. The measurement can be compensated with the help of a temperature sensor if the temperature variations are high enough to cause a significant error. Additionally, the sensor should avoid coming into contact with the bottom to prevent sediment-related issues.

Several important factors need to be considered when selecting a pressure sensor to make sure it is suitable for the application. Different sensors at different price points differ in quality and accuracy. This includes specifications such as linearity, repeatability, hysteresis, and error span. Additionally, pressure sensors typically have a specified pressure range which the sensor can safely operate within. If the pressure goes outside of this range, the sensor can get irreversible damage. The optimal installation of a water pressure sensor is below the lowest expected level. This means that the sensor should be capable of safely measuring the pressure from the highest expected water level (Understanding Pressure Sensor Specifications, n.d.).

LMI has tested some submersible water pressure sensors with positive results proving this method works for the application, however, commercially available sensor systems with acceptable accuracy are not affordable. This type of sensor has a clear advantage over ultrasonic and laser-based sensors since they are less affected by obstructions found in an active borehole with less-than-perfect environmental conditions. A drawback of this type of sensor is its susceptibility to electromagnetic interference. This could prove to be a problem in boreholes equipped with electrical water pumps. The extent of this problem is unknown and would require testing.

3.2 Depth gauge and water well sounder

Depth gauges and water well sounders come in different designs and methods of operation with more advanced measuring devices being electrical. The wetted tape method is perhaps one of the simplest implementations and is accurate to a depth of around 27 meters (Trimmer, 2000). This method works by lowering a measuring tape down the well

until it comes into contact with the water table. At this point, the user should check the length of the measuring tape against a point with permanent or otherwise known elevation. After removing the measuring tape from the well, the user can calculate the actual distance to the water table by removing the wet part of the tape from the length of tape used. Although this is a cheap and simple method, the short maximum measurable distance that still provides accurate measurements is a disadvantage.

An electric sounder is a more advanced method that takes advantage of the electrically conductive property of water. In its most basic form, it consists of an insulated wire with a weight at the end, low voltage electricity provided by a battery, and a light bulb or electrical multimeter if a higher sensitivity is needed. A wire consisting of two conductors can be used if the well casing is made from a non-conductive material and therefore cannot act as the second conductor. The method works by lowering the wire down the well until it reaches the water table. The electrical conductivity property of the water creates continuity when the wire comes in contact with the water surface allowing electricity to flow through the completed circuit. This flow of electricity can be observed using the light bulb or multimeter. At this time, the user takes a reading of the length of the wire lowered down the well compared with a static reference point at a known elevation (Trimmer, 2000).

The main advantages of the wetted tape and electric sounder techniques are the low initial investment and simplicity yet outnumbered by their disadvantages. An electric well sounder device has further reach than the wetted tape method but is still limited to shallow wells and boreholes. According to Trimmer (2000), a drilled borehole spirals as it goes down. This can cause interference between pumping equipment inside the borehole and the measuring tape or wire and possibly cause it to become entangled. There is a risk in this case that the measuring equipment brakes if removed by force. These methods require manual measurement taking and logging of data as well as repeated access to the water table inside the wells. Regular, frequent, and long-term measurements are essential to get high-quality data, which is challenging due to the manual

nature, in addition to the increased risk of water contamination from outside sources. Sercu (2023) states that these methods for measuring water levels are outdated and newer automated methods should be preferred.

Manual measurements are needed to calibrate hydrostatic pressure measurements from pressure sensors. However, to get information on not only seasonal variations but also weekly and daily groundwater fluctuations caused by factors such as variable borehole extraction requires extensive manual measurements, making it very labor intensive and ineffective.

3.3 Air line pressure measurement

Using air pressure as a proxy measurement is another technique that can be used to indirectly measure groundwater levels. This technique, commonly found in drilling and water operations, provides a fast and easy method for measuring the water table according to Garber and Koopman (1968). This method works by measuring air pressure inside a pipe that is affected by the groundwater level. This method requires a pipe capable of withstanding internal air pressure, a pressure gauge, and a source of compressed air. The pipe is lowered down a borehole until the end of the pipe is lower than the lowest possible water level resulting from pumping or other natural events. Next, compressed air is introduced into the system until the pressure reading becomes stable. At this point, all water has been pushed out of the air line. After detaching the compressed air connection, the pressure reading will decrease and settle if the system is working correctly without leaks. This pressure value can then be used in the formula below to calculate the distance between the water table and the surface

$$d = k - (p * 2.307 \frac{\text{ft}}{psi}), \quad (2)$$

where d is the water level measured from the surface in feet, k is the length of the air line in feet, p is the pressure reading measured in pounds per square inch, and last the conversion value of 2.307 ft/psi (U.S. Geological Survey, n.d.).

Trimmer (2000) suggests that “An air line provides the most convenient method for repeated testing of deep wells over 300 ft deep” and U.S. Geological Survey (n.d.) states that one advantage the air line method provides over the wetted tape method is its ability to being unaffected by a disturbance in the water caused by pumping. They also state that this method’s accuracy is not affected if used in spiraling boreholes. There are, however, other factors that influence the accuracy of this method. A calibrated system using an accurate analog or digital pressure gauge can provide an accuracy of 0.1 foot, but, according to the U.S. Geological Survey (n.d.), “Water-level measurements using a pressure gauge are approximate and should not be considered accurate to more than the nearest foot”.

The two biggest causes of deviation in measurement accuracy in an air line system are imprecision in the length of the air line and the accuracy of the pressure gauge used to take measurements according to Garber and Koopman (1968). Other additional factors that have an impact on the accuracy to a lesser extent are changes in air density relative to altitude and temperature changes, which have an effect on air and water density as well as the thermal expansion of the system (Garber & Koopman, 1968).

The pressurized air line method can be used to measure a water level, however, its requirements and limitations make it unsuitable for the use case of LMI. It requires an additional pressure-capable pipe which adds cost and makes the borehole more obstructed. The existing water pipe could potentially be used, but this would disrupt normal pumping operations while measuring, and information related to the pumping process, including the rate at which the water changes during pumping, cannot be measured. Having a reliable source of compressed air in remote locations without electricity is challenging in addition to the space required by the hardware, which ideally should be

installed inside the borehole to not be visible. Lastly, the low precision this method provides is not within an acceptable range and it is not capable of measuring with a high enough frequency needed for the use case of LMI, making this method less than ideal.

3.4 Time-of-flight based techniques

Time-of-flight (ToF) is a method used by some sensors to measure the distance to an object. Light Detection and Ranging (LiDAR), Radio Detection And Ranging (Radar), and ultrasonic are types of sensors utilizing this ranging method. In general, these types of sensors work by sending out signal pulses and measuring the time it takes to receive an echo, and from this calculate the distance between the sensor and the object. Based on the time and known signal speed, a distance of as low as a few millimeters can be calculated, although several different factors contribute to how accurate the measurement is (Paul et al., 2020).

These types of sensors are in the family of noncontact sensors able to measure a distance through air in contrast to previously mentioned methods which are in direct contact with the water. An advantage noncontact sensors have is less hardware degradation over time due to better operating conditions compared to contact sensors. Ultrasonic and radar sensors are currently used to a small extent to measure water levels in rivers, while LiDAR technology is less used in water-measuring applications and more commonly found in other applications (Paul et al., 2020).

3.4.1 LiDAR

One of the main challenges with using LiDAR technology in water level measuring applications is the water reflectivity according to Paul et al. (2020). A small amount in the range of a couple of percent of the incoming signal is reflected from a water surface with low rugosity and turbidity according to the research paper. The amount of the signal

reflected towards the sensor also depends on the signal's angle of impact. The researchers tested LiDAR technology for its suitability to water level measurements using a commercially available LiDAR sensor produced by Garmin, called the Garmin Lidar Lite rangefinder sensor costing 130 USD in 2020. During testing, the LiDAR sensor provided an accuracy of 1 cm in a distance measurement of up to 10 meters and 3 cm for a distance of up to 30 meters, both of which are within the specified limits of the sensor. This results in an approximate accuracy error of 0,1 % of the measured distance.

Paul et al. (2020) also tested the impact of water rugosity on the measurement accuracy by measuring the water level in different rivers at an angle of 90° to the water surface. They concluded that the measurement accuracy of the sensor improved from 8,2 cm down to 1,7 cm with higher surface rugosity in the range of 5 cm to less than 1 cm. The measuring distance varied between 3,6 meters and 10,2 meters. Higher water turbidity results in better signal reflectance but was concluded to not have an impact on the accuracy of the measurement.

The angle of impact is another factor that has an effect on the accuracy of the sensor and needs to be taken into account. Decreasing the angle of impact increases the amount of the signal being reflected away from the sensor resulting in worse accuracy. Lastly, the impact temperature has on the accuracy was investigated. Based on the test results, the highest accuracy was achieved at around 20 °C while the deviation increased at both lower and higher temperatures with a maximum deviation of 9 cm at 80 °C. The negative effect temperature has on the sensor's accuracy can be decreased with improved temperature compensation of the sensor and its sensitive electronics (Paul et al., 2020).

As Paul et al. (2020) concluded, there has been little research done on LiDAR technology as a water level sensor. Although LiDAR technology proved to be capable of measuring a water level, the low reflectivity of water makes it challenging to use especially as the distance increases. The need for a clear line of sight to the water surface limits this

technology to wider wells with fewer obstructions and shorter distances between the sensor and water since deeper boreholes are not necessarily straight. Another factor to consider is the cost and energy needs of this technology compared to other technologies. Paul et al. (2020) state that an increase in light energy gives better precision and less noise, but this would increase battery requirements. The suitability of LiDAR technology for the use case of LMI remains unknown as it is not a proven method for water level measuring applications in deep boreholes and would require more research.

3.4.2 Ultrasonic

According to Toa and Whitehead (2021), ultrasonic sensors use sound waves with high frequency outside the audible frequency range of humans as their wave signals. They operate by emitting ultrasonic sound pulses and timing how long it takes for the pulses to be reflected by a surface and echo back to the receiver. Distance can then be calculated using the formula below where the time difference is t, and the speed of sound is v.

$$Distance = \frac{t * V}{2} \tag{3}$$

What medium the soundwave propagates through must be known to calculate a distance. This is because the speed of sound is dependent on the medium it travels through. Another factor that affects the speed of sound is the temperature of the medium. Air humidity affects the speed but to a lesser degree than previously mentioned factors (Toa & Whitehead, 2021).

Ultrasonic sensors can be used to detect different kinds of liquids and solid materials if they can reflect enough sound energy for the sensor to detect. The acoustic impedance mismatch between two mediums determines the acoustic reflectivity and how much energy is reflected. Higher impedance mismatch reflects more sound energy. This means

materials with a low impedance mismatch are difficult to measure. The energy reflection caused by impedance mismatch between two mediums can be calculated using the following formula

$$R = \left(\frac{Z2-Z1}{Z2+Z1}\right)^2, \quad (4)$$

where R is the reflection coefficient, Z2 impedance of the reflecting medium, and Z1 impedance of the travel medium (Toa & Whitehead, 2021).

The reflection coefficient between air and water can be calculated to be approximately 0,988 using the impedance mismatch formula and the acoustic impedance values from Table 1. This means close to 99 % of the incoming sound energy is reflected, while the remaining continuous through the water.

Table 1. Density, acoustic velocity, and acoustic impedance for air and water (Toa & Whitehead, 2021).

Material	Density kgm^{-3}	Acoustic Velocity ms^{-1}	Acoustic Impedance $kgm^{-2}s^{-1} * 10^6$
Air	1,3	330	0,00429
Water	1000	1450	1,45

Another factor to consider when using ultrasonic sensors is energy attenuation. Qiu et al. (2022) state in their report that a sound wave loses its energy to absorption of the surrounding medium and leads to the received echo having less energy than the initial emitted energy. The further an ultrasonic wave propagates through a medium, the more energy is lost on the way and less is received by the sensor. A higher sound frequency can offer greater accuracy but at a shorter range since sound energy attenuation increases as the sound frequency increases.

Sound waves have a relatively slow propagation speed according to Qiu et al. (2022). This allows them to use less precise computing hardware giving ultrasonic sensors a price advantage in the field of ToF sensors. Another advantage ultrasonic sensors have is their low sensitivity to interference from outside sources. These sensors are less affected by surrounding lighting conditions and electromagnetic interference since they use sound waves instead of radar waves or light. The advantages the technology provides have made ultrasonic sensors reach widespread adoption by different industries. They are used in a range of applications including distance measuring, vehicle positioning, and object detection and avoidance. Although ultrasonic sensors can achieve millimeter accuracy, Qiu et al. (2022) claim their maximum measurable range is usually less than 10 meters, however, sensors capable of measuring longer ranges do exist. Two examples of sound-based ranging devices available on the market are the Well Watch 670 from Eno Scientific (Eno Scientific, 2024) and the Sonic Water Level Meter from Solinst (Solinst Canada Ltd., 2024). Solinst claims their sensor can measure a water level down to 600 meters with an accuracy of 3 cm inside wells and boreholes while accounting for obstructions and angles that can cause false readings. Eno Scientific on the other hand claims up to 1200 meters and 3 cm accuracy. Solinst claims 500 hours of use with one charge while no specific battery operation time was found for Eno Scientific.

As mentioned earlier, ultrasonic sensors emit a sound pulse and measure the time it takes until the sound echo is received, and based on this calculate a distance. This simple measuring method relies on only receiving one easily recognizable echo and does not work in environments where more than one echo is received. False echoes occur when something else than the intended surface to be measured reflects all or a portion of the emitted sound energy back to the sensor receiver according to Massa (n.d. -a). This can cause the sensor to calculate the wrong distance if not configured correctly. Massa says this problem can be mitigated by adjusting the sensor threshold for the time interval of the wrong echo. Figure 1 and Figure 2 show a graph of a sensor's threshold magnitude in red and signal echo magnitude in black as a function of distance. Both figures show a false echo resulting from an obstruction and the desired echo from the water surface.

Massa says that a false echo can result from obstructions like a pipe located in the sensor's measuring path. In Figure 1, the false echo causes the sensor to show the wrong distance but by increasing the sensor detection threshold, as is shown in Figure 2, the sensor ignores all echoes of lower magnitude than the threshold. In Massa's example, this leads to the sensor calculating the distance to the water instead. This works for all water levels, even if the water level would rise to the same level as the obstruction since the echo from the water is greater in magnitude than the threshold magnitude set at the distance of the obstruction. Although this technique looks promising for avoiding false echoes, it requires more advanced sensors since all ultrasonic sensors are not capable of being configured this way (Massa, n.d. -a).

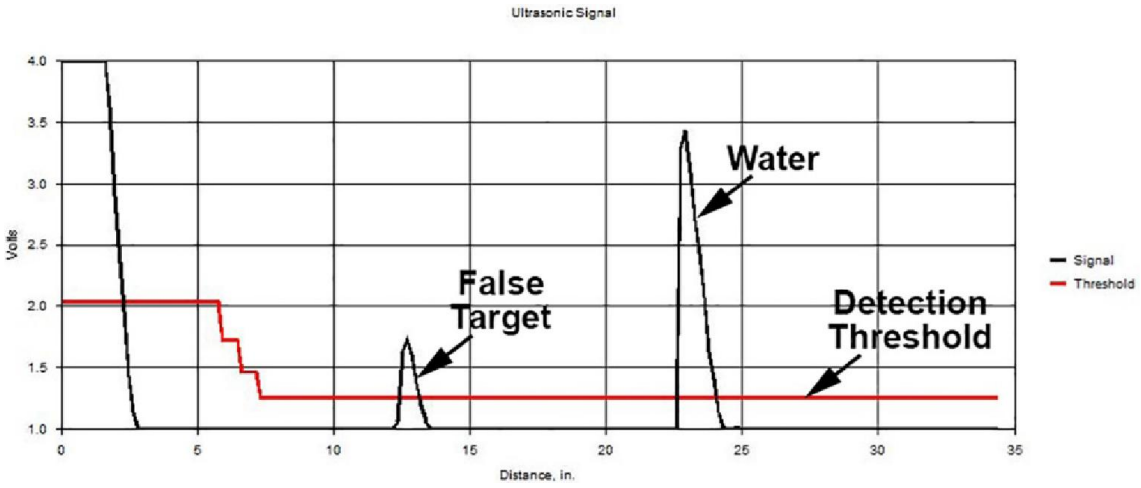

Figure 1. "Ultrasonic Waveform from a MassaSonic™ PulStar™ Plus Sensor Showing a False Target Being Detected Instead of the Echo from the Surface of the Water" (Massa, n.d. -a).

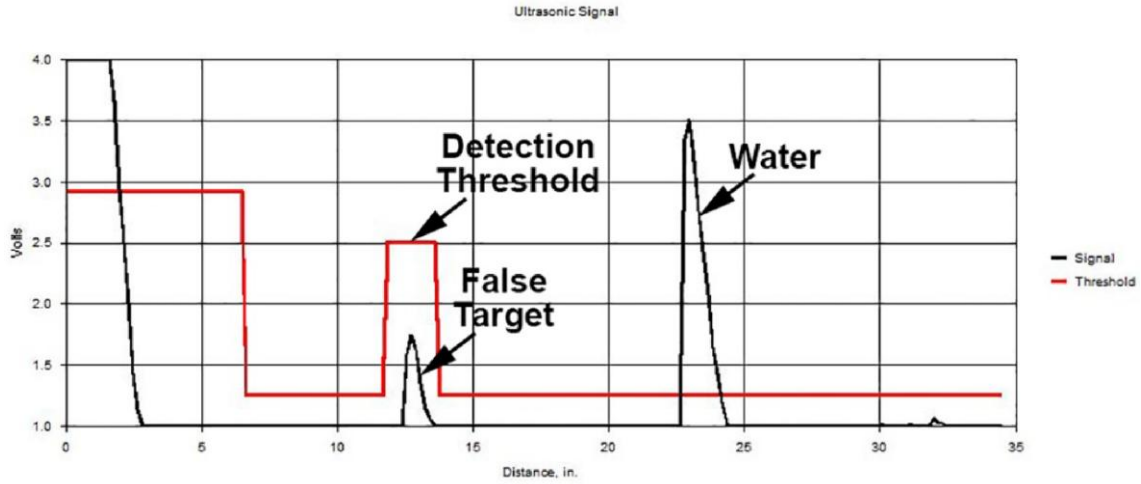

Figure 2. "MassaSonic™ PulStar™ Plus Ultrasonic Waveform With the Same Targets As In Figure 1, But With the Detection Threshold Modified to Ignore the False Target" (Massa, n.d. -a).

False echo detection in the previous example made by Massa (n.d. -a) is relatively trivial compared to false echo detection inside a borehole. The false echo shown in Figure 1 and Figure 2 is visible and there is only one obstruction causing a false echo, but using this technique inside an active borehole is most likely a greater challenge since obstructions exist at all distances causing several false echoes.

One factor that affects an ultrasonic sensor's performance is its beam angle. The beam angle is how wide of an angle the sound waves have when emitted from the sensor. How wide or narrow the beam angle is depends on the design of the sensor. Increasing the diameter of a sensor's vibrating surface while producing sound with the same wavelength results in a narrower beam angle. Figure 3 shows a 2D plot of beam patterns for different diameter to wavelength ratios with the highest ratio producing the narrowest beam and the lowest producing the widest beam. Although an ultrasonic sensor has a specific beam angle at which the soundwaves are emitted, it may still detect sound reflections arriving from an angle outside the original beam angle due to secondary lobes (Massa, n.d. -b).

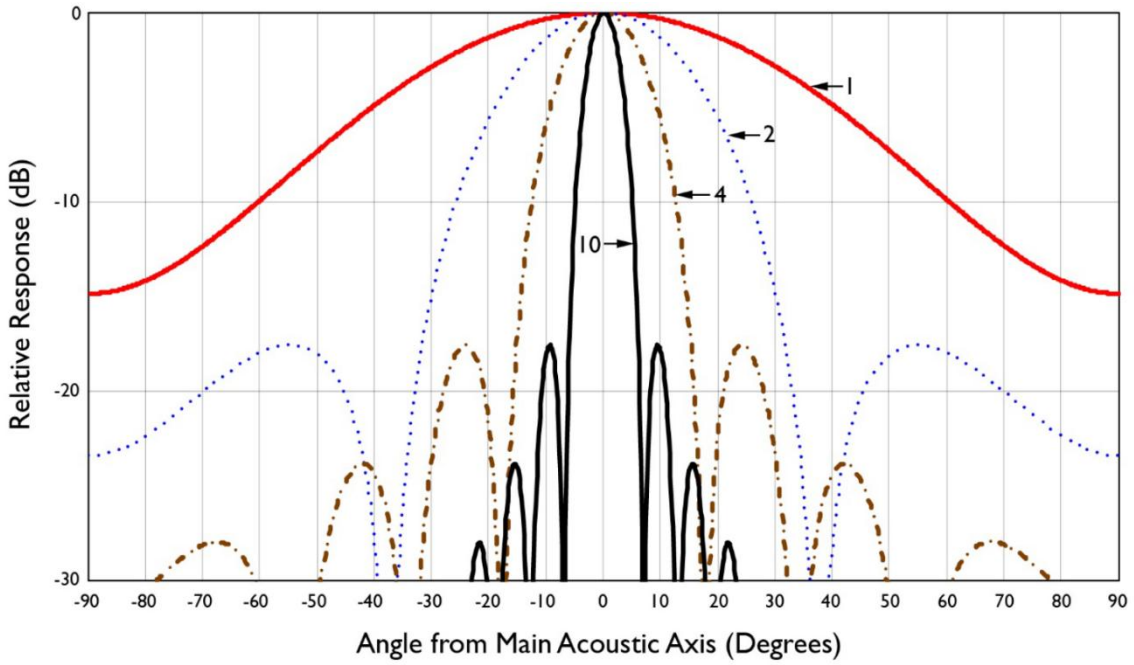

Figure 3. "2D Plots on Rectilinear Coordinates Showing the Beam Patterns of Four Different Circular Piston Radiators in an Infinite Baffle Having Diameter to Wavelength Ratios, D/λ, of 1, 2, 4 and 10" (Massa, n.d. -b).

According to Smoot (2021), a sensor with a narrower beam angle has a longer signal reach compared to one with a wider beam angle because the radiating sound energy is concentrated into a smaller area. Reducing the beam angle also leads to a reduced area of detection. Long-range and small areas of detection are beneficial in a borehole

application since the desired water surface to be measured has a small surface area and fluctuations cause large measuring distances. A small area of detection additionally leads to fewer false echoes from the borehole casing and pumping equipment.

One approach to modifying an ultrasonic sensor's beam angle and wave propagation pattern is by funneling the waves through a sound tube attached to the sensor. Mahler et al. (2022) evaluated sound tubes and their impact on cheap ultrasonic sensors. They managed to achieve improvements in accuracy in their specific test application and environment by using a sound tube compared to not using one, but they also saw some interference caused by the tube. They concluded that the design of the sound tube influences how the soundwaves propagate after leaving the tube. They further suggest the beam angle of the sound waves after leaving the sound tube could be reduced by using tubes of different lengths, diameters, and geometry.

Another approach to modifying an ultrasonic beam is by using a Fresnel Zone-Plate (FZP). Schindel et al. (1997) assess the effect of using a micromachined FZP as a lens for focusing high-frequency ultrasonic waves. According to the authors, FZP works by having rings of different diameters machined on a thin plate. The rings are designed to either be in-phase with the sound wave to let it pass through or out-of-phase with the sound wave to decrease the sound energy. This leads to a more focused energy beam compared to not using an FZP. The authors tested an FZP designed to focus the ultrasonic beam into a point and managed to achieve a focus point of less than 3 mm. Other designs of Fresnel Zone-Plates can, according to the authors, be used for different beam patterns including a line-focus which can potentially be beneficial for borehole applications. Recent advances in 3D additive manufacturing printing technology have made it possible to print FZP ultrasonic lenses with promising results according to Wang et al. (2021).

Sound-based distancing is a proven technology for measuring water levels and is suitable for boreholes with obstructions according to the previously mentioned solutions available on the market. Additionally, they have an acceptable accuracy and a low sensitivity

to electromagnetic interference which is beneficial for the use case of LMI. The drawback of the commercially available solutions is their high cost and installation. They are designed to be installed on top of the borehole casing making them challenging to hide. It is unknown if this type of sensor works reliably if installed inside the casing. The stated battery life is also not suitable for long-term autonomous operation without an external power source.

As previously mentioned, the speed of sound is dependent on temperature. The Solinst sensor incorporates temperature compensation by automatically measuring the surface temperature using an internal sensor, and a user measured and manually added temperature from the water surface. The manufacturer does not explicitly state how the measurement adjustment is calculated but appears to rely on these two points to calculate a linear temperature gradient. Although this is a simple approach that may work adequately in many cases, the temperature gradient is not always linear, especially in locations affected by human activity according to Kłonowski et al. (2024). This design is not capable of automatically adjusting for a change of temperature at the water level as well as a nonlinear gradient. This would require installing additional temperature sensors or manual temperature readings at regular intervals inside the borehole from multiple depths.

Continuous battery operation time is another problem with the discussed products. The time is shorter than ideal in good operating conditions, and it is unknown how operating in high surface temperatures in Africa will affect the battery. The high acoustic reflectivity of the air-to-water interface is helpful since it reflects most of the sound energy, however, the signal loses energy as the wave propagates through the air. This means measuring longer distances in deeper boreholes requires more emitted sound energy decreasing battery time.

The commercially available systems do not fully meet the requirements of LMI, making it necessary to develop a custom system addressing the limitations, but this would

involve solving technical challenges. One of them is to develop an algorithm to identify the correct echo from a multitude of false echoes resulting from the borehole casing and obstructions. In some boreholes, it may not even be possible to reliably identify the correct water surface reflection. While this technology may work with acceptable results in some use cases of LMI, it would need thorough field testing to know its feasibility.

3.4.3 RADAR

Compared to ultrasonic and LiDAR, RADAR distance sensors emit electromagnetic radio waves with a wavelength in the millimeter range to measure a distance. The short wavelength of millimeter radio waves gives radar sensors the ability to sense a change in distance of less than a millimeter. Traditional RADAR sensors use pulsed signals, but modern RADARs can use signal modulation techniques. One type is the frequency-modulated continuous wave which can be used to measure the distance to an object, its velocity, and its angle (Iovescu & Rao, 2020).

Catsamas et al. (2023) explored the possibility of using a relatively cheap off-the-shelf RADAR module for measuring the height and speed of river water. They used the radar model XM132 by Acconeer, which can be bought for less than 20 € at the time of writing. The XM132 radar is a pulsed coherent radar, meaning it transmits pulses of radio waves at a known starting phase according to its manufacturer (Acconeer, 2025). This radar module operates at 60 GHz, and it can achieve millimeter precision (Catsamas et al., 2023).

Catsamas et al. (2023) created a sensor housing and radar lens using polylactic acid (PLA) and 3D printer technology to keep the cost low together with a potting compound for waterproofing the electronics. A lens to decrease the radar wave beam angle was needed since the radar beam angle is between 40° and 80° and a smaller angle would give better accuracy with fewer false readings according to the researchers. The radar lens they designed focuses the wide-angle spread of radar waves into a narrower beam

as illustrated in Figure 4, as well as focusing incoming radar waves into the radar receiver creating a gain of 21 dB compared to not using the lens. This gain increases the sensitivity of the receiver.

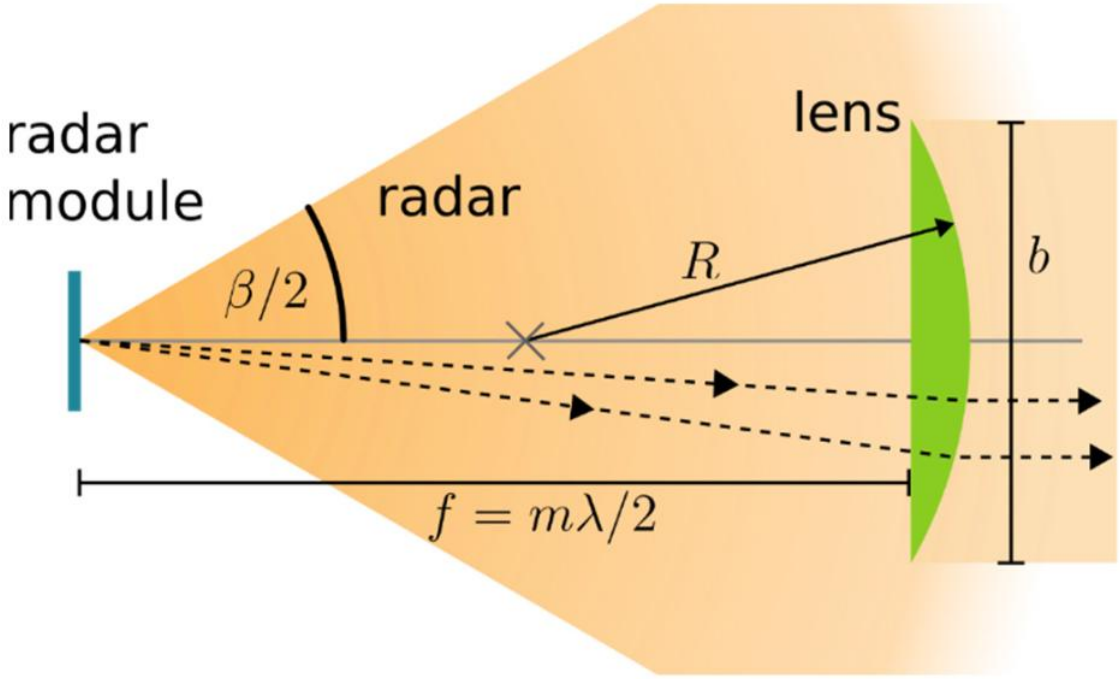

Figure 4. "An optical diagram displaying the function of the radar lens. The falloff of radar intensity is illustrated in the orange background. Particular radar rays are displayed as dashed lines. Note how the radar lens focuses the rays from the radar module into a parallel beam reducing the intensity falloff with distance. b is the diameter of the lens, R the radius of curvature, and f the focal length. m is an arbitrary integer. ß is the beam width angle of the radar emission. Paths of the rays through the lens are reversed for incoming radar signals." (Catsamas et al., 2023).

Catsamas et al. (2023) conducted field tests comparing their radar sensor against a commercially available submerged water velocity and depth sensor in the field. The measured distance ranged from 100 mm to 550 mm. The measurements obtained by the radar sensor closely follow the measurements of the submerged sensor proving the radar sensor to be successful in river measuring application. The measurements can additionally be improved by using a simple algorithm to remove outliers according to the researchers.

The results from the experiment conducted by Catsamas et al. (2023) demonstrate that radar technology can be used to measure the distance to a water surface with acceptable accuracy, however, it is not proven to work inside active boreholes, and no off-the-shelf solutions are available on the market. As with sound-based distancing methods, using radar technology inside a borehole would require developing a new system and addressing several issues. While the negative effect of noise in the signal can potentially be reduced through frequency modulating techniques (MATLAB, 2023), the ability to reliably detect the correct echo from false echoes remains unclear. An alternative radar-based sensor, for example, guided wave radar, may offer advantages by avoiding obstructions. This method works by transmitting the radar wave from the antenna through a guiding cable down to the medium to measure. When the wave hits the medium transition interface, a part of the wave is reflected which the sensor then times to calculate a distance (Instrumentation Tools, 2019).

3.5 Additional methods from literature

García-López et al. (2022) present a novel cost-effective solution for automated large-area groundwater monitoring by using a UAV-borne LiDAR system. The system utilizes LiDAR and a Global Navigation Satellite System (GNSS) module attached to a quadcopter, also known as an Unmanned Aerial Vehicle (UAV), to fly autonomously on a preconfigured flight path between groundwater access points. The LiDAR system scans the ground below and subsequently measures the distance to the water surface inside wells or other points where access is available. In situ level measuring systems that are currently used are simple and reliable according to García-López et al. (2022), but frequent maintenance combined with large distances make them less cost-effective than the tested UAV-borne LiDAR system. Reducing overall cost by phasing out traditional in situ level sensors and related installation and maintenance is what makes this system a better choice for some applications, however, it has big drawbacks influencing its effectiveness. The

maximum flight time of the tested quadcopter is 55 minutes, which limits its maximum flight range. This makes a system like this better suited for smaller areas with a high density of measuring points and less suited for large-area, low-density applications where flight distances are too long. Additionally, the need for access to the groundwater table either through wells with a wide accessible uncovered opening, bodies of water that are linked to the groundwater table, or other points where access to the groundwater is available makes a system like this feasible only in some locations. Another limiting factor of this method is the low measuring frequency. For an application where the measuring frequency needs to be several times a day or hour, this UAV LiDAR approach is not feasible.

Mahmoudzadeh et al. (2012) suggest using a non-intrusive Ground Penetrating Radar (GPR) to collect accurate and highly detailed depth data. This method works by transmitting radio waves into the ground and recording the signal reflections coming from different ground layer interfaces. Based on this data, the depth to where the saturated soil interface starts can be calculated. The authors claim that by choosing the most optimal antenna frequency for the GPR, it is possible to locate the groundwater interface among different types of ground material layers. While this groundwater measuring technique has some advantages over other traditional measuring methods, it has some key limitations.

The process of getting accurate depth data using a GPR demonstrated by Mahmoudzadeh et al. (2012) does require some manual labor and estimations, which can impair the results. To be able to scan a large ground area, the GPR system needs to be moved. This was achieved by the researchers by pulling the GPR on a wheeled cart behind a manned motor vehicle. Another challenge using this system is varying signal attenuation in different types of soil which must be estimated using other additional technologies, for example, frequency domain reflectometry or groundwater level data from existing wells. Since the dielectric constant varies between soil types, using only one constant when processing the data can impair the accuracy of the measurement. Mahmoudzadeh et al.

(2012) stated that the groundwater table may be impossible to find using a GPR in cases where a layer of soil with a high dielectric constant lies above the groundwater table. High signal attenuation was also observed by Salih et al. (2022) and they concluded that layers of clay can cause high signal reflectivity limiting accuracy and ground penetration range. They also saw errors in groundwater table estimations caused by saline groundwater and capillary action.

After data conversion and processing, Mahmoudzadeh et al. (2012) observed a measurement error of 4 cm compared to the calculated depth of groundwater. Other researchers have reported less accurate depth estimations. Bentley and Trenholm (2002) state that “Theoretical analysis and field experiments indicate that, under favorable circumstances, the elevation of shallow water tables can be estimated with an accuracy on the order of 0.20m.” Additionally, Salih et al. (2022) state that when using a radar frequency of 200-250 MHz the maximal achievable penetration depth is 10-15 meters while keeping adequate accuracy as good as 5 cm. Lower frequencies can penetrate deeper into the ground but at the cost of reduced accuracy since frequency and accuracy are related.

4 Prototype Design and Implementation

This chapter presents a prototype as a proof-of-concept solution for an autonomous hydrostatic pressure-based water level sensor. The primary objectives for this prototype are to test the individual components working together as a complete system, evaluate the performance of the sensor and system as well as energy efficiency, and finally discuss the functionality of the system and firmware.

The first tested prototype is based on a submersible water pressure sensor with an included sensor wire. This type of water pressure sensor minimizes the need for waterproofing of the rest of the system since only the sensor and part of its wire will be submerged and submitted to higher water pressure. The other major component of the system is the Microcontroller Unit (MCU). Firmware running on the MCU, in combination with the integrated hardware features of the MCU, controls the logging process and wireless communication. Other necessary hardware components needed for this design include an external Analog-to-Digital Converter (ADC) module, a boost circuit, and a battery. The boost circuit increases the relatively low 5 V USB to the 24 V required by the sensor to operate optimally. Figure 5 shows a block diagram illustration of the connections and functionality between these components and Figure 6 shows the system circuit diagram. All connections and components, except for the pressure sensor itself, are housed inside a plastic box with a cable passthrough for the incoming sensor cable (see Figure 7).

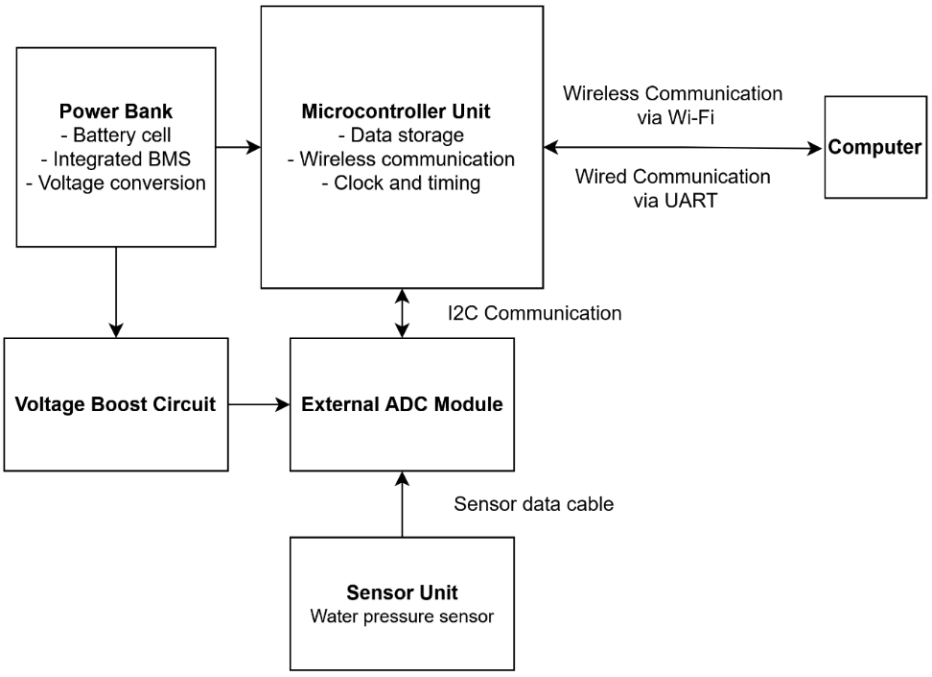

Figure 5. Block diagram of prototype 1.

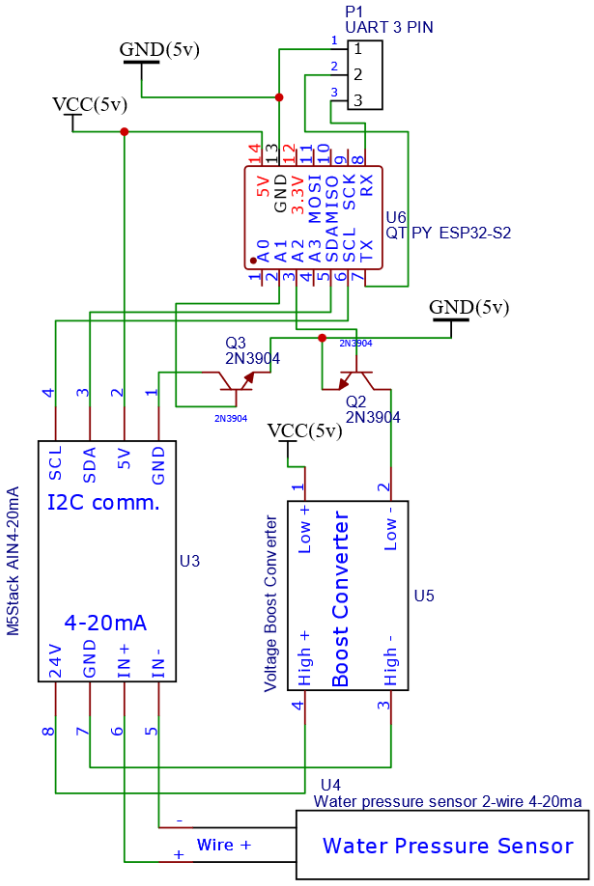

Figure 6. Circuit diagram of prototype.

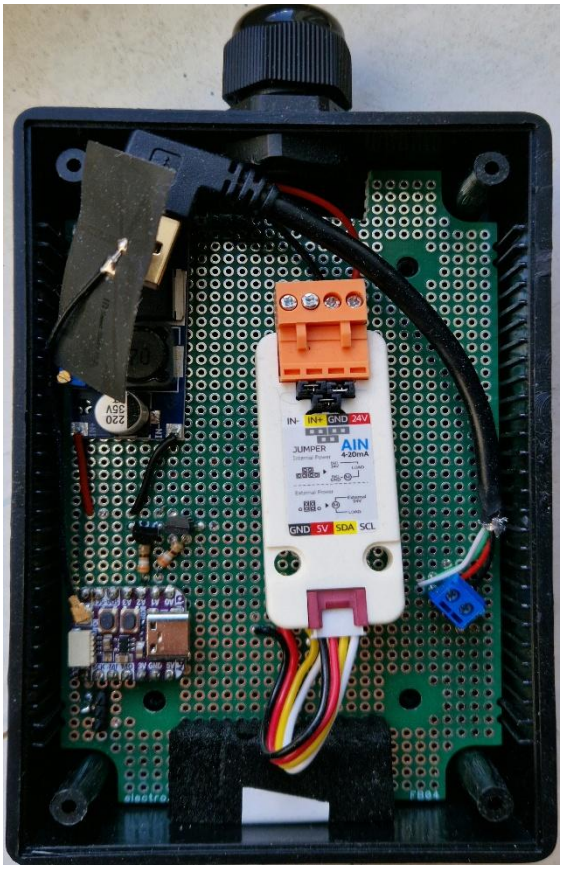

Figure 7. Image of prototype.

4.1 ESP32-S2 microcontroller unit

Choosing the correct MCU is crucial since it will be the core controller responsible for managing and interacting with all other components of the system. The MCU is the only component of the system that cannot be completely powered down, making energy efficiency a key factor in minimizing the overall system energy demands and prolonging battery life. The ESP32-S2FN4R2 microcontroller, referred to as ESP32-S2 for simplicity, from the ESP32 microcontroller family made by Espressif, proved to be suitable due to its high availability, low price, and included features. It is a low-cost, low-power system on a chip (SoC) with a single 32-bit Central Processing Unit (CPU). Figure 8 shows the SoC architecture and its features including which features are supported in which power mode. The ESP32 family of microcontrollers varies in its support for wireless

communication. As shown in Figure 8, the ESP32-S2 specifically includes support for wireless communication in the 2.4 GHz frequency spectrum using Wi-Fi. The microcontroller also supports the proprietary wireless communication protocol ESP-NOW (Espressif Systems, 2024).

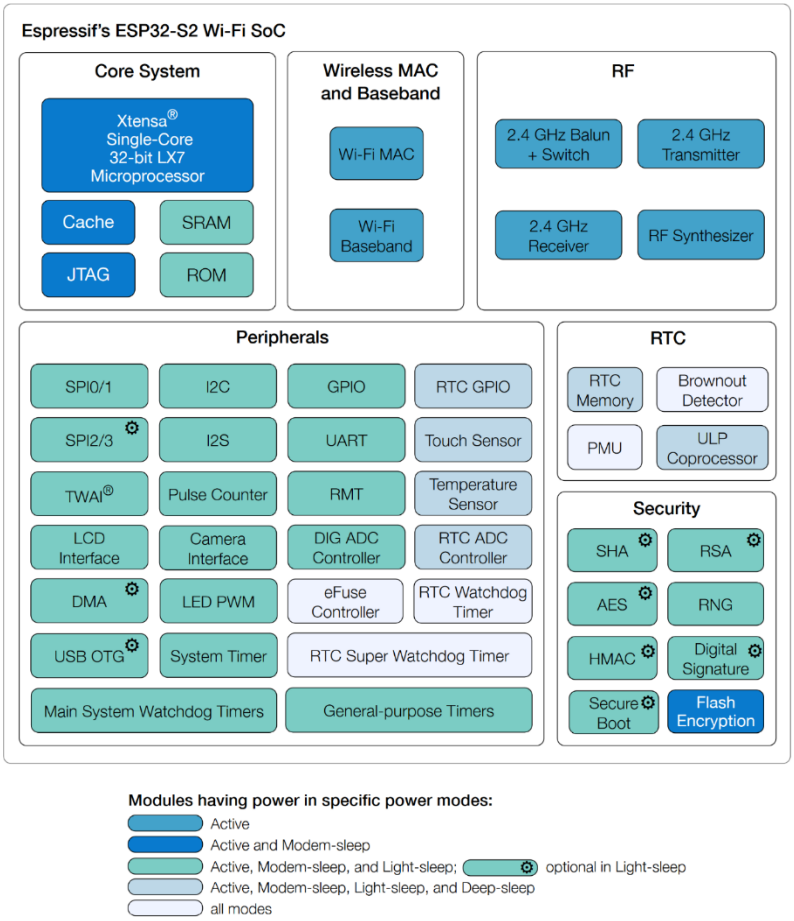

Figure 8. Block Diagram of ESP32-S2 (Espressif Systems, 2024).

The ESP32-S2 has a big-little CPU configuration using three cores. One of them is a faster general-purpose CPU supporting all features of the MCU, while the other two are scaled-down Ultra Low Power (ULP) coprocessors. Although it has two ULP coprocessors, only one of them can be used simultaneously. One of them is a ULP finite state machine (FSM) type which can be programmed using assembly code. The other coprocessor is a more advanced RISC-V core with support for the open-source RV32IMC instruction set

architecture. The RISC-V coprocessor can be programmed using the C language. These coprocessors are designed to perform simpler tasks, for example, continuous sensor monitoring, while the main CPU is turned off in a low-power mode. Using the ULP co-processor instead of repeatedly waking up the main CPU to monitor a sensor or to perform a simple task decreases the overall chip energy consumption while still being able to execute code (Espressif Systems, 2024).

The ESP32-S2 has native hardware support for several wired communication protocols including Universal Asynchronous Receiver Transmitter (UART), Inter-Integrated Circuit (I2C), and Serial Peripheral Interface (SPI). These communication protocols are useful for debugging during development and runtime while also providing MCU-to-MCU communication and communication between MCU and peripheral devices, for example sensors. The UART port serves as the main wired communication between the MCU and the user. Although it provides a wired connection for general communication between two devices, its main purpose in this prototype is to provide debugging of the firmware and system as well as providing low-level user control through a command line interface. This simultaneous two-way communication functionality, where both the MCU and an external computer can send and receive data at the same time, is made possible by the full-duplex capability of the UART protocol. The integrated SPI is by default only used for interfacing with non-volatile external SPI flash memory chips or external Random Access Memory (RAM) chips but can also be used to interface with other external hardware components and sensors (Espressif Systems, 2024).

Two successive approximation register ADCs are integrated into the ESP32-S2 and enabled on analog GPIO pins of the MCU. These can be used to convert an analog voltage, for example from an analog sensor to a digital binary value which can be further processed by the MCU. These ADCs are, however, not suitable for precise measurements. Makerfabs (2022), reports an average error of 37 mV and 45 mV for the models ESP32-S3 and ESP32 respectively, while the ESP32-C3 has an average error of 200 mV. After using Espressif's official inbuilt calibration for the ESP32, Makerfabs observed an

improvement in average error of 18 mV in the voltage range 0,1 V to 3,2 V but it still struggles with the lowest and highest voltage ranges 0 V to 0,1 V and 3,2 V to 3,3 V. Due to the relatively bad performance of the integrated ADC, it was decided to use an external ADC module with support for digital communication over I2C. This enables the MCU to communicate and request data from the ADC. Additionally, an external ADC is not affected by the varying temperature of the microcontroller compared to the integrated ADC.

Previously mentioned communication and interfacing methods are configured in the firmware, including which GPIO or RTC GPIO pins the interface will use through the GPIO multiplexer. With some limitations, most GPIO pins can be routed through the multiplexer to allow them to perform various functions. This feature simplifies hardware development since the GPIO pins can be assigned different functionalities and are not constrained to one function only (see Figure 9). While GPIO pins can only be used when the main CPU is active in a high-power mode, GPIO pins with RTC support can be configured to work in low-power modes. The prototype utilizes this feature when in deep sleep mode to provide a means of communication between the MCU and the user through a pushbutton. These GPIO RTC pins can additionally be controlled by the ULP coprocessor while the main CPU is not active to, for example, communicate over UART and I2C with other devices (Espressif Systems, 2025).

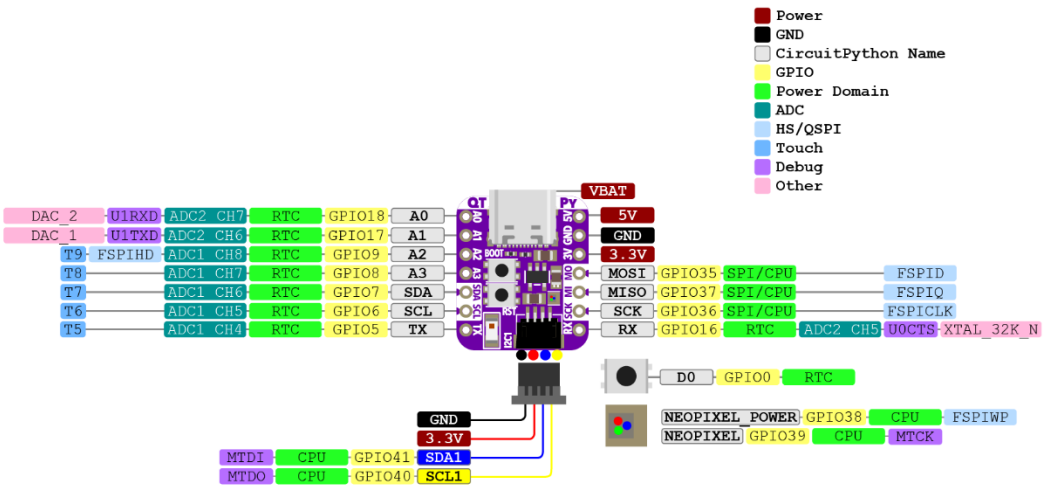

Figure 9. Pinout of ESP32-S2 QT PY microcontroller from Adafruit (adapted from AdaFruit Learning System, n.d.).

The ESP32-S2 has several built-in watchdog timers used for different purposes. Watchdogs can give priority to a task, reset a task, or reset the CPU and reboot the entire system if needed. These functionalities make systems more reliable by being able to restart themself without manual intervention in case a problem arises. It is especially useful in remote applications that are difficult or impossible to manually reset, for example in a deep borehole. The supported watchdog timers by the ESP32-S2 include the interrupt watchdog timer, task watchdog timer, and XTAL32K Watchdog Timer. The interrupt watchdog timer monitors Interrupt Service Routines (ISR). Whenever an interruption occurs, its attached ISR will be executed. The interrupt watchdog timer steps in and gives priority to the ISR if it is blocked by other tasks or ISRs with lower priority. The task watchdog timer is used to monitor tasks. It makes sure the system does not freeze indefinitely in case a task hangs, not yielding to other tasks after a specified time, or gets stuck in a loop for example. The XTAL32K Watchdog timer can be used if the ESP32-S2 internal RTC_SLOW_CLK is configured to use an external 32 kHz clock signal as its source. The watchdog monitors the incoming clock signal and can create an interruption or switch the clock source for the RTC_SLOW_CLK to an internal clock if it detects a problem with the external clock signal (Espressif Systems, n.d.).

The ESP32-S2 chip has a Power Management Unit (PMU) controlled by firmware and it allows the MCU to enter different power modes depending on the use case. The power modes activate or deactivate modules and their integrated hardware features enabling the chip to use the most optimal power mode for a specific task. Table 2 shows the different sleep modes and what the energy usage is depending on what component is powered on. Depending on the preferred system operation, the ESP32-S2 should spend most of the time in deep sleep using either of the three modes where the ULP coprocessor is powered down. When the main CPU is woken up by the ULP coprocessor, internal timer, or external interruption, it will boot and start executing code from the start.

Table 2. Current Consumption in Low-Power Modes (Espressif Systems, 2024).

Mode	Description		Typ (uA)
Light-sleep	VDD_SPI and Wi-Fi are powered down, and all GPIOs are high impedance		750
Deep-sleep	The ULP co-processor is powered on	ULP-FSM	170
		ULP-RISC-V	190
	ULP sensor-monitor pattern		22
	RTC timer + RTC memory		25
	RTC timer only		20
Power off	CHIP_PU is set to low level, the chip is powered off		1

4.2 Timekeeping

System time of the ESP32-S2 can be set through firmware at compile time and can in a later stage be synchronized from an external source, for example, Simple Network Time Protocol if the MCU has a network connection or manually over the wired UART communication port. Accurate timestamps of when the groundwater levels are measured included in the data logs are important and help when analyzing the data. For example, correlations between fluctuations in the water level and pumping activities can be made

if the level drops consistently at a certain time of day when an electric pump is expected to run. When logging data periodically over a long time, it is also important to consider the time deviation.

A correlation between an increasing water level and precipitation or pumping activities and a falling water level may be possible to accurately observe from the data in a short time frame after equipment installation or after a time synchronization event. However, the same correlation may be difficult to observe if the timestamp error increases significantly, although the time of pumping stays consistent. For example, if the timestamp error increases by ten seconds over a period of 24 hours, the cumulative time error would amount to nearly one hour after one year. Synchronizing the time at regular intervals can minimize this problem. How frequently the time needs to be synchronized depends on the maximal allowable error and the speed at which the time deviates. One source for the timestamp error is the clock source it is relying on. The greater its frequency deviation is, the greater the error will be. A possible alternative to regular synchronization of time, which can be difficult, expensive, or impractical, is to correct the timestamps while post-processing the data. If the frequency deviation is known and stays consistent over time, and the exact logging start and end times are known, all timestamps can be shifted forwards or backward the amount of time they are expected to have changed. Later in this section will be discussed the result of a timestamp error test.

The task of keeping system time is performed by hardware timers integrated into the MCU. The ESP32-S2 version used in the prototype has two available hardware timers for this purpose, the RTC timer and the high-resolution timer. One or both timers can be configured to keep system time depending on the system design and which power modes are used. The RTC timer can be configured to run in all power modes and does only reset in the event of a power loss reset, while the high-resolution timer is only available while in an active power mode. The high-resolution timer uses APB_CLK as its clock source and provides a resolution of 1 µs and less than ±10 ppm frequency deviation with

a typical frequency of 80 MHz. To maximize battery life, the MCU is configured to use deep sleep power mode while waiting dormant for the next logging. This makes the high-resolution timer unsuitable to be used for keeping system time (Espressif Systems, n.d.)

The RTC clock domain is split into the RTC slow clock and the RTC fast clock (see Figure 10). The RTC timer used for system time relies on the RTC slow clock, which remains active in low-power modes, unlike the RTC fast clock, which is deactivated. The RTC timer has a resolution of 6.6667 µs and its accuracy depends on the clock source of the RTC slow clock. It is configurable to use one of three clock sources. The options are an adjustable frequency from an internal RC oscillator which by default is 90 kHz, the internal 8 MHz oscillator clock signal divided by 256 to achieve a lower frequency of 31,25 kHz, and an external 32 kHz signal originating from a crystal or other clock circuit. In Figure 10, these are called RTC_CLK, RTC8M_D256_CLK, and XTAL32K_CLK respectively (Espressif Systems, 2024).

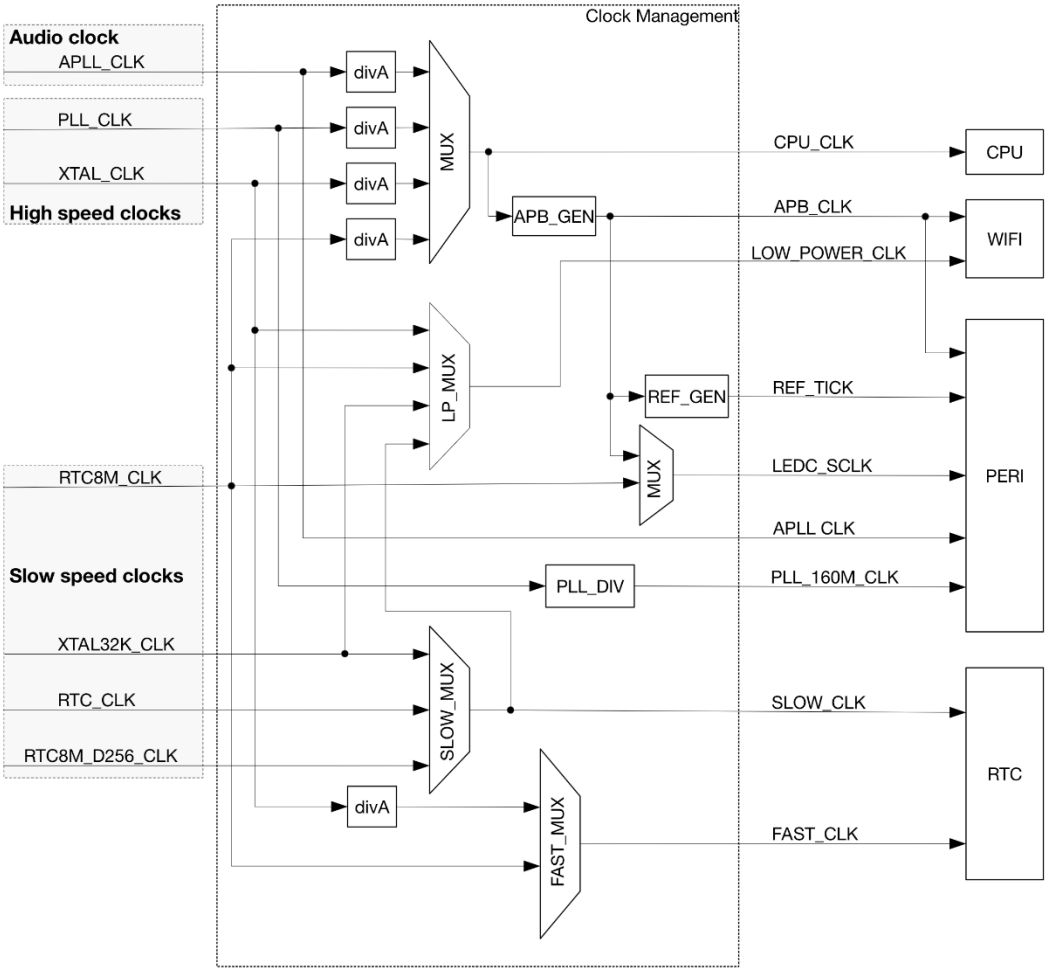

Figure 10. System Clock (Espressif Systems, 2020).

According to the documentation for the ESP32-S2, the internal 90 kHz RC oscillator has the lowest power consumption, but its accuracy is affected by changes in temperature during deep sleep. The datasheet claims the 8 MHz oscillator to be more stable than the 90 kHz oscillator, but no parts-per-million (ppm) accuracy measurement is mentioned. The 8 MHz oscillator additionally results in an increase of deep sleep current by 5 µA. An external crystal can provide better frequency stability compared to the 8 MHz and the 90 kHz RC oscillators but at the expense of an increased current consumption in deep sleep of 1 µA. Using an external crystal or clock circuit requires supported GPIO pins for interfacing to not be used for other functions. An external clock circuit does not increase the current consumption of the ESP32-S2, albeit the circuit itself does use some amount

of energy which will be part of the total energy consumption of the system (Espressif Systems, n.d.).

Minimizing energy consumption is a priority, especially during deep sleep which is the power state the MCU will use most of the time, therefore, the 90 kHz RC oscillator would be the optimal clock source. The problem with using this clock source is, however, its accuracy. Due to the limited availability of data on the accuracy and the effect changes in temperature have on the clock frequency deviation, it is difficult to determine if using it would have a significant negative impact on the system time. Two tests were performed to evaluate and get a better insight into the potential of using this 90 kHz signal. The second test also includes frequency analysis for the 8 MHz signal. Due to the lack of an appropriate oscilloscope, the tests relied on a practical testing method and inbuilt frequency calibration functions.

4.2.1 Analysis of internal 90 kHz RC clock deviation

The 90 kHz clock signal is analyzed in the first test with default clock configurations. The default timer configuration includes both the RTC timer and the high-precision timer in combination with the internal 90 kHz RC oscillator as the clock source. Although both timers are used by default, only the RTC timer is active during deep sleep mode. Both timers are enabled while awake. The test was performed by using the ULP processor to wake up the main CPU at a specified interval of one second over a recorded arbitrary period. Later the recorded wakeup count was compared to the recorded elapsed time. No significant ambient temperature changes were observed during the test period.

Figure 11 shows the timing process flow of the main CPU on the left side and the ULP process on the right. After being woken up by the start timer, the ULP coprocessor disables and then re-enables the sleep timer which will cause it to restart counting. After re-enabling the sleep timer, the ULP coprocessor tries to wake up the main CPU by sending an interrupt to the RTC controller. The interruption is expected to be successful since the

main CPU is in deep sleep mode and not blocked by another process. After sending the wakeup interrupt, the ULP coprocessor goes back to sleep.

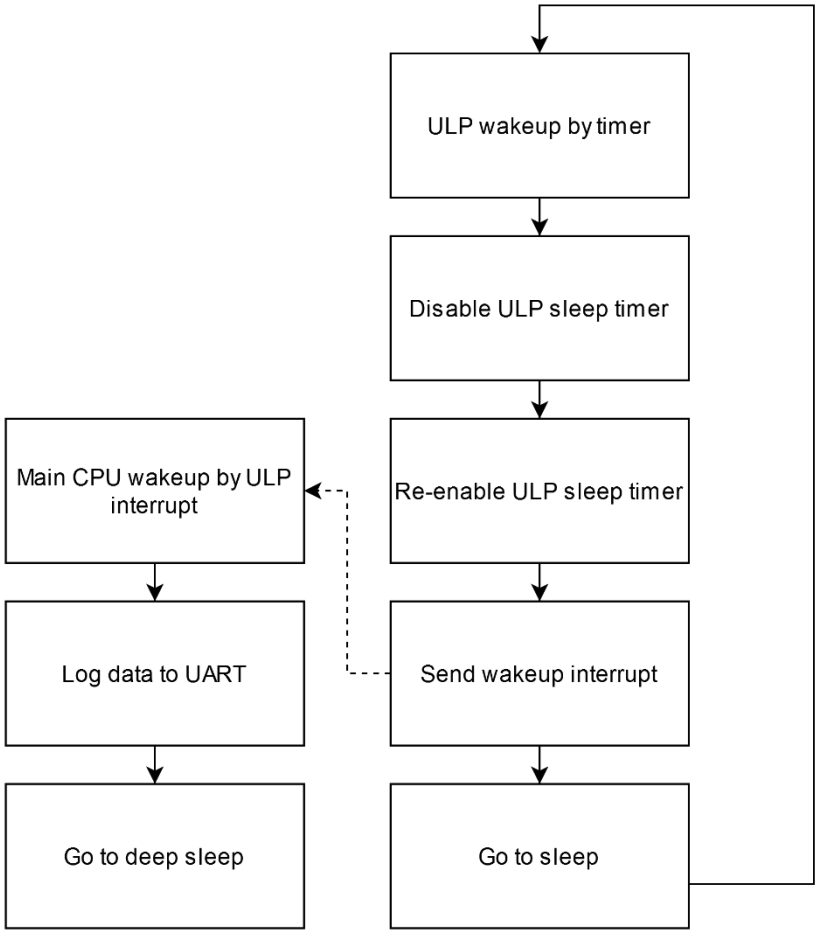

Figure 11. Flowchart visualizing program process flow during test of timing interval.

The ULP coprocessor was chosen to manage the timing interval because it can run independently of the main CPU once started, and the time deviation caused by its code execution is relatively short and known.

The other methods supported by the firmware framework, used during development, and the hardware for creating a wakeup interruption proved to be insufficient to accurately and consistently wake up the main CPU with a set interval. The main CPU can be woken up from several sources including a timer set by the main CPU, ULP interrupt, and external interrupt through one of the GPIO pins. Using the timer creates a problem since

it is restarted only once the main CPU goes to deep sleep. The main CPU needs time after waking up to execute code for reading, processing, and writing the data logs to memory. The time these processes take to execute would need to be subtracted from the timing interval before entering a deep sleep state creating unnecessary complexity and uncertainty because the code execution time was observed to vary between logs.

After running the test for an arbitrary time of 26250 seconds, the system was observed to have logged 26229 times. It can be concluded based on the test result that the system time error was 21 seconds over a test period of close to seven hours. Since the system should log once every second, there should be the same number of logs as seconds. There are two main possible sources of this error. First is the ULP assembly code execution time, and second is the frequency instability of the RC oscillator, which was used as the clock source for the ULP wakeup timer.

Algorithm 1 includes a code snippet of the assembly instructions executed by the ULP FSM coprocessor after its wake-up timer overflows. After the wakeup timer overflows, the ULP coprocessor needs 2 clock cycles to wake up and then it waits for 16 clock cycles until the 8 MHz clock is stable according to the documentation for the ESP32-S2. The ULP coprocessor starts at the memory address where the entry point is stored. The first instruction is executed two times. It first disables and then re-enables the ULP wakeup timer by writing a 0-bit and then a 1-bit to the RTC register controlling the ULP timer. This write instruction does not have documentation for execution clock cycles but is expected to need the same as REG_WR, which needs 8 cycles to execute, and fetching the next instruction needs 4 cycles. The remaining instructions will not have an impact on the time interval deviation since the timer has already been restarted, however, the documentation is unclear if the HALT instruction restarts the timer if it has been restarted earlier. If this is the case, an additional 4 to fetch WAKE, 2 cycles to execute, 4 cycles to fetch HALT, and 2 clock cycles to execute. Another 2 cycles are needed after HALT for the ULP to go to sleep according to the documentation (Espressif Systems, n.d.). This results in either 42 or 56 clock cycles. Since the ULP is clocked from the 8 MHz clock signal

calculated to have an actual frequency of approximately 8,8 MHz, each clock cycle execution time is approximately 113,6 ns. Scaled up to the complete test runtime of 26250 seconds, the time error is approximately 125 ms or 167 ms.

Two conclusions can be made from this test. The ULP coprocessor can in some applications be used as a reliable timer for waking up the main CPU for further sampling and data processing, and the RC oscillator frequency deviation is most likely the main cause of the error in time. The frequencies of the RC oscillator and the 8 MHz oscillator are analyzed in the following subchapter.

```
#include "soc/rtc_cntl_reg.h"
#include "soc/rtc_io_reg.h"
#include "soc/soc_ulp.h"

.bss
.text
.global entry

entry:
// Disable the ULP sleep timer

     WRITE_RTC_REG(RTC_CNTL_STATE0_REG,
     RTC_CNTL_ULP_CP_SLP_TIMER_EN_S,
     RTC_CNTL_ULP_CP_SLP_TIMER_EN_S, 0)

// Re-enable the ULP sleep timer, starting the countdown
immediately

     WRITE_RTC_REG(RTC_CNTL_STATE0_REG,
     RTC_CNTL_ULP_CP_SLP_TIMER_EN_S,
     RTC_CNTL_ULP_CP_SLP_TIMER_EN_S, 1)

     WAKE
     HALT
```

Algorithm 1. Assembly instructions executed by the ULP FSM coprocessor after being woken up by its timer.

4.2.2 Analysis of internal clock stability

In this test, the deviation of clock frequency for the 90 kHz and the 8 MHz internal oscillators of the ESP32-S2 are analyzed. As was mentioned in the previous chapter, the 8 MHz oscillator has better frequency stability than the 90 kHz oscillator, but no accuracy measurement was provided in the datasheet. This chapter will compare the accuracy of both internal oscillators and their potential to be used for keeping accurate system time.

Both clock frequencies were logged once every second with a sample size of 9000. The firmware framework used, called ESP-IDF, has inbuilt methods for calibrating clock frequencies against the external 40 MHz crystal called XTAL. These methods were used to get a frequency reading with an accuracy within the XTAL frequency error range of ±10 ppm (Espressif Systems, n.d.). The temperature was additionally recorded in the same process in case it affects the stability of the frequencies. The temperature sensor used is the internal sensor of the ESP32-S2 which measures the internal temperature of the microcontroller chip. It has a claimed accuracy error of less than 1 °C in the range -10 °C ~ 80 °C (Espressif Systems, n.d.).

By analyzing the RTC_SLOW_CLK clock signal plotted in Figure 12, we can see small fluctuations in the frequency with soft transitions except for the last samples where the frequency shows a sudden increase. The last 250 samples in the range increase in frequency to an average of 91022 compared to the rest of the samples that have an average of 90934 Hz. All samples in the range have an average frequency of 90936 Hz (see Table 3). After a decrease in temperature at the beginning of the test from 29 °C to 27 °C, the temperature does not change significantly during the rest of the test although the 90 kHz signal is not stable. This means there are additional factors not measured in the test that influence frequency stability and not only the temperature.

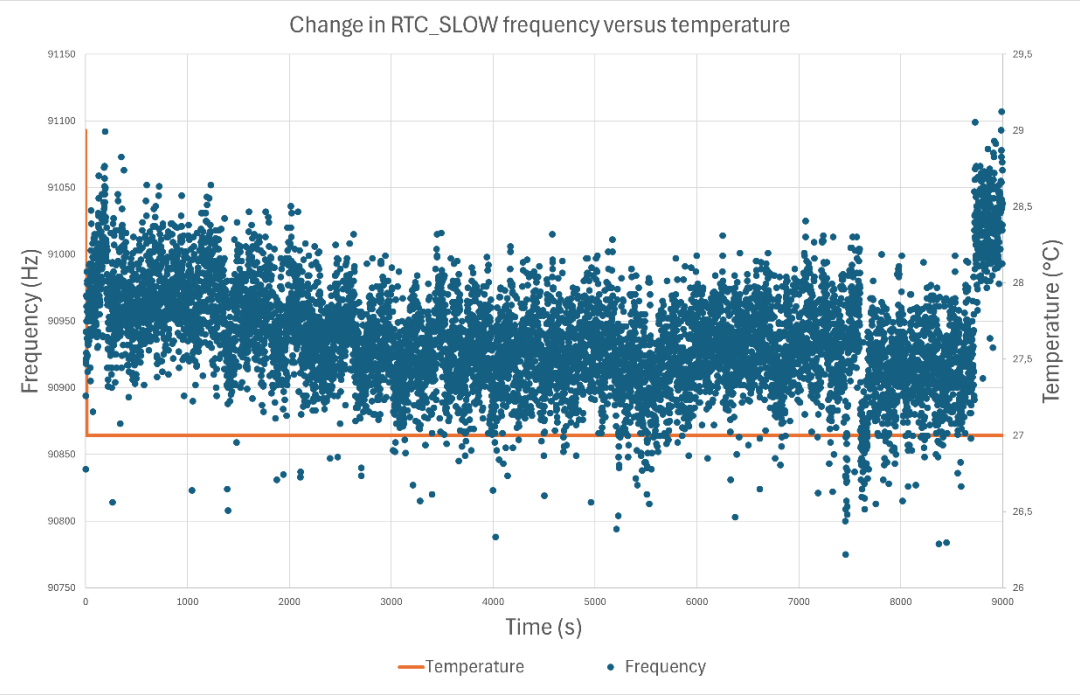

Figure 12. Log series of the internal 90 kHz RC oscillator circuit frequency in blue and temperature change in °C in orange over a sampling period of 9000 seconds.

Table 3. Calculated variables for RTC_SLOW_CLK frequency deviation test log.

Category	Hz
Max	91107
Min	90775
Difference Max/Min	332
Peak diff. from average	170,253806
Median	90935
Average	90936,74619

The 8 MHz oscillator circuit frequency plotted in Figure 13 shows a different behavior compared to the 90 kHz signal in the previous figure. From the plotted clock signal in Figure 13 two distinct frequency groups can be observed. The higher group has an average calculated frequency of 8839342 Hz while the lower has 8751709 Hz, a difference of 87633 Hz. The higher frequency group consists of 8354 samples, corresponding to 92,8 %

of all samples, while the remaining fall into the lower. Although there are two distinct groups of frequencies, the overall frequency is more stable than that of the 90 kHz clock. The average frequency for all samples is 8833061 Hz (see Table 4). The temperature change is the same as in the previous graph, with an initial decrease after which it stays at a stable level for the remaining log period. A correlation between changes in temperature and frequency cannot be made from this data and would require further testing with controlled variations in temperature.

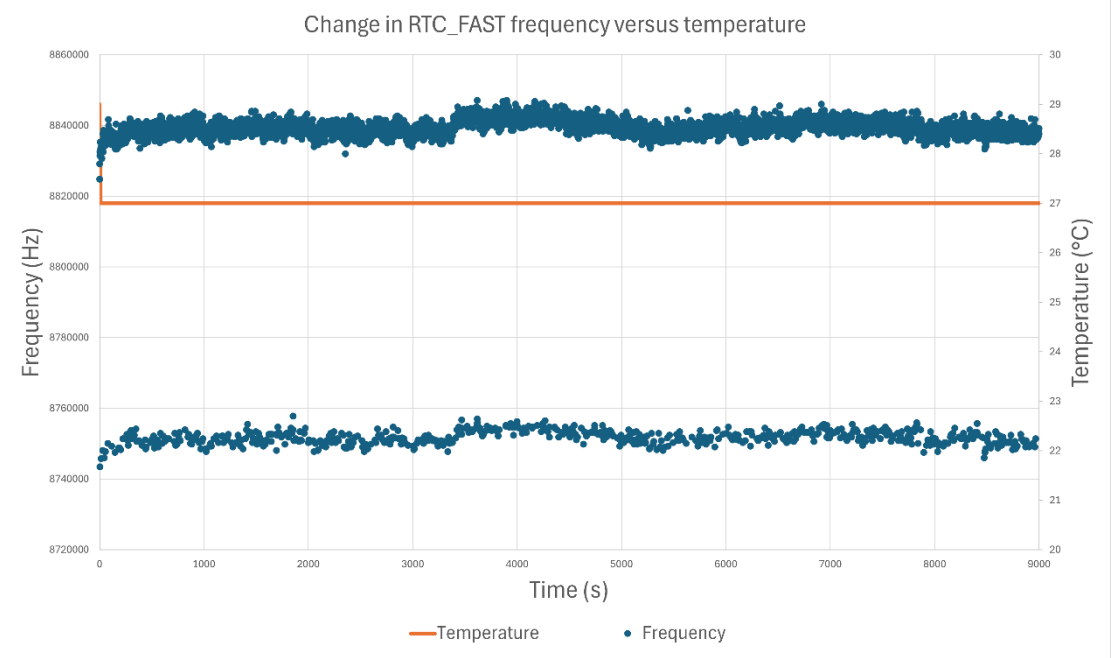

Figure 13. Log series of the internal 8 MHz oscillator circuit frequency in blue and temperature changes in °C in orange over a sampling period of 9000 seconds.

Table 4. Calculated variables for RTC_FAST_CLK frequency deviation test log.

Category	Hz
Max	8847104
Min	8743424
Difference Max/Min	103680
Peak diff. from average	89637
Median	8839168
Average	8833061

Previously mentioned parts-per-million (ppm), also called fractional frequency deviation (FFD), is a measurement used to describe "the degree to which the frequency of a clock can deviate from a nominal (or reference) frequency". This measurement gives an insight into how stable a frequency is and can be calculated using the following formula

$$FFD = \frac{df}{fc} * 10^6\ ppm, \quad (5)$$

where df is the peak variation in frequency from a center frequency represented by *fc* (adapted from Ajmeri et al., 2022).

A ppm error for both the 90 kHz and the 8 MHz signals can be calculated using this formula. The ppm error for the 90 kHz signal can be calculated by using the average frequency as the center frequency and peak difference from the average as *df* in the formula, both values found in Table 3, we get a value of approximately 1872 ±ppm. Using the same formula for the 8 MHz signal with its respective values from Table 4, we get an approximate error of 10147 ±ppm.

These ppm values can be converted to time deviation over elapsed time to get a better understanding of how much the signal drifts over time. The following formula describes how to convert ppm to units of time

$$FFD\ (in\ ppm) = \frac{X\ \mu s}{sec}, \tag{6}$$

where the ppm value multiplied by a number of seconds results in time deviation in microseconds over the same time period (Ajmeri et al., 2022).

By using the previously mentioned conversion relation formula to convert the 90 kHz and 8 MHz ppm errors to time it would result in ±1872 µs and ±10147 µs drift per second respectively. Scaling up to 24 hours results in a time deviation of ±2,7 minutes and ±14 minutes and over a year ±16,4 hours and ±88,9 hours respectively. These test results do not comply with the statement in the documentation that the 8 MHz signal would have better stability compared to the slower 90 kHz signal. This discrepancy likely originates from the two frequency groups of the higher frequency signal. To test the impact the lower frequency group has on the error, it is filtered out and the same calculations are performed on the higher frequency group. This results in a positive ppm value of +878 and a negative of -1641. The positive ppm is a significant improvement by close to halving the error, but the negative is not much better than that of the 90 kHz signal.

The TD-Diver can be used as a benchmark to have something to compare against since it is already in use and accepted by LMI. It claims to have a maximum time deviation of ±5 minutes over a period of one year according to its datasheet. The specified operating temperature range is from 0 °C to 50 °C (Van Essen Instruments, 2023). Comparing the results against the TD-Diver it is clear the time deviation of the internal clock signals tested are multiple times too high and far outside the acceptable range. This makes both unsuitable for keeping system time over extended periods if the system is in deep sleep.

4.3 Sensor side of prototype

The sensing side of the prototype is comprised of a gauge-type water pressure sensor encapsulated in a watertight stainless-steel tube with an attached cable (see Figure 14), Analog to Digital Converter (ADC) module, and a voltage boost circuit. The sensor was selected based on its affordable price, availability, and ease of use since it comes in a premade watertight package simplifying prototype development. The negative side of the sensor is its lack of advanced specifications, guarantees, and its unknown longtime reliability.

As was mentioned in previous chapters, changes in atmospheric pressure can have a significant impact on the accuracy of water pressure sensors and need to be compensated for. The developed prototype cannot measure changes in atmospheric pressure, but the sensor itself compensates for changes in atmospheric pressure by design. This means that the sensor can compensate for changes in atmospheric pressure, but it cannot compensate for variations in the groundwater level resulting from changes in atmospheric pressure. Accurate atmospheric pressure data from an additional sensor as well as a correlation between groundwater level and atmospheric pressure is needed for this compensation. The compensation enabled by the sensor design is made possible due to being a gauge-type sensor with its reference pressure equalized to atmospheric pressure through a vent tube. This hollow tube is housed inside the attached cable and runs all the way down to the sensor housing. In addition to the vent tube, the attached cable also houses a two-wire design for power and data transmission. A gauge pressure sensor measures the pressure above a reference pressure, which in this case is the atmospheric pressure.

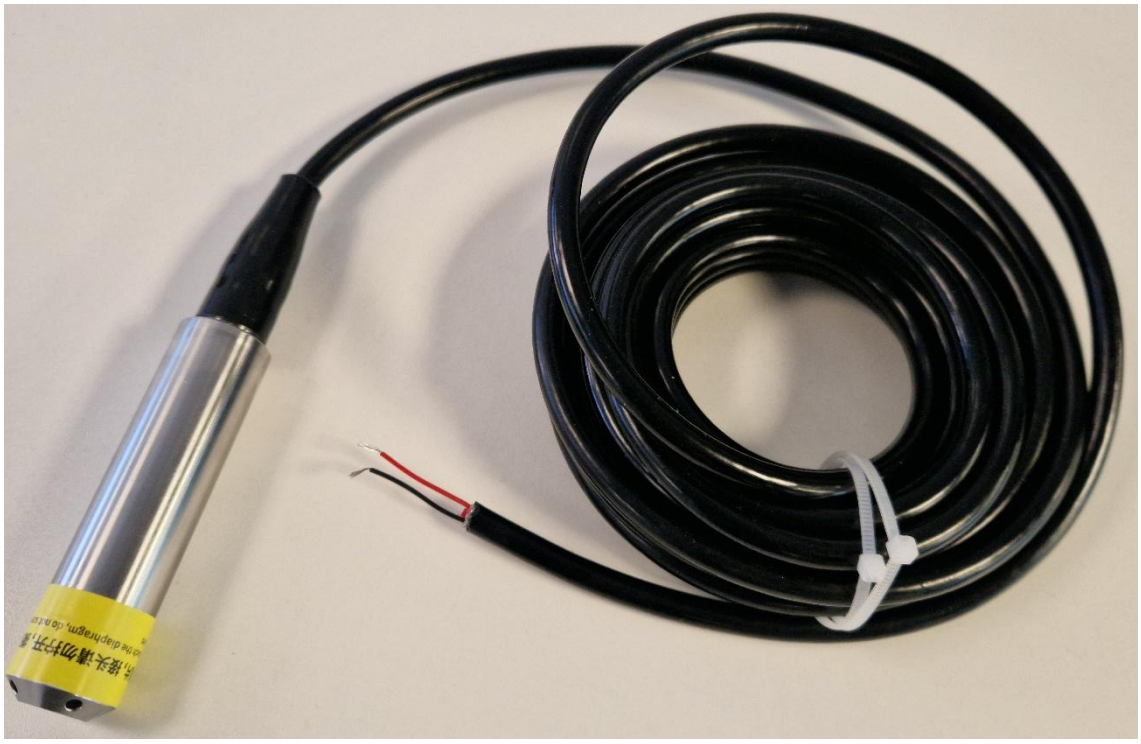

Figure 14. Waterproof submersible 4-20 mA water pressure sensor with attached data cable and vent tube.

This type of sensor uses a current loop design and communicates over the two wires by changing the current flow. First, 24 VDC needs to be applied to the two wires to power up the sensor. Then the sensor changes the flow of current through the wire relative to the pressure the sensor experiences. The current flow varies in amplitude between 4 mA and 20 mA when the sensor operates correctly within its design pressure range. An amplitude of 4 mA represents the lowest measurable pressure which for this sensor is atmospheric pressure, and 20 mA represents the highest measurable pressure of the sensor. This sensor is designed to measure a maximum water depth of 5 meters which means the sensor can measure a maximum pressure corresponding to the pressure at a depth of 5 meters above atmospheric pressure.

The external ADC module AIN4-20mA Unit from M5STACK was used to read the varying currents produced by the sensor. It is designed for 4-20 mA sensors and can measure a current between 0 and 25 mA. This enables the system to detect if the sensor is faulty through the current reading. If the reading is in the buffer zone outside the valid output

range, the sensor is not working correctly. The ADC module has integrated I2C communication through which the MCU can request ADC values from the module. The module has an inbuilt galvanic isolation chip to separate the higher voltage side from the sensitive electronics on the low voltage side. This design makes it less likely to break the electronics in case of a malfunction or wrong connection. The module additionally features a 16-bit ADC converter resulting in a step distance of 0,119 mm in the range of 0 to 5 meters in the range of 4 to 20 mA (M5-docs, n.d.).

4.4 Battery configuration

A common rechargeable lithium polymer USB power bank with a rated capacity of 16,7 Wh was used as the main power source for the system. Maximizing space and energy efficiency while minimizing self-discharge and weight were not of the highest priority in this prototype, which is why it was decided to use this battery implementation. It simplifies the hardware design stage since the power bank is removable, can be recharged with a common 5 V USB power source, and has an integrated Battery Management System (BMS) and voltage converter. This removes the need for implementation of additional hardware components, although the exact specifications of the BMS and voltage converter are not published by the manufacturer which makes efficiency, energy consumption, and usable output energy calculations less accurate.

Some USB-power banks, including the one used in the prototype according to its manufacturer, have implemented an automatic switch-off feature to conserve energy when not in use. This feature is designed to switch off the power bank when it detects a fall in output current below a specified threshold. Although a useful feature if the power bank is used to recharge the battery of another device, it can become a problem if used to power systems that use small amounts of current. This problem, however, was not observed during the testing of the prototype.

A BMS is necessary to use in lithium-based battery-operated systems to maximize battery protection, lifespan, and overall system safety. This is done through the BMS by managing the integrated lithium polymer cell from over- and undervoltage, high currents, and temperature. Failing to do this may result in irreversible damage to the battery, fire, or in the worst case an explosion (Suganya et al., 2024).

The runtime using the power bank was tested by charging the battery fully and running the system until it shuts itself off. The logging interval of one minute consists of being awake for 5 seconds reading the sensor and writing data and in deep sleep for 55 seconds. While in deep sleep the energy consumption is minimal with a best case of 20 µA according to its official documentation, however, the MCU development circuit used in the prototype has a documented deep sleep current of approximately 70 µA (Adafruit Industries, n.d.). With this current consumption only the RTC timer is powered on which is needed for waking up the ULP FSM core. On the contrary, energy consumption increases significantly during the logging process. This increase results from powering up the MCU, boost circuit, ADC module, and sensor. Of all the hardware components, only the MCU has specified in its documentation its current consumption while awake, which makes it difficult to calculate the current needed while logging which leads to estimations.

After letting the system run independently until the battery was empty it managed to run for approximately 5 days. Based on the available documentation and battery test results, the following values can be estimated. The system uses 0,005 Wh/day while in deep sleep and 17,57 Wh/day while awake. This is an increase of 3500 times. With the theoretically lowest deep sleep consumption of the MCU, the factor would increase to close to 9000 times.

4.5 Local data storage

How data is stored is an important factor to consider. Losing data can lead to the loss of important groundwater information recorded over long periods. All data that needs to be stored over extended periods without running the risk of losing it needs to be stored in non-volatile memory in case the power source is interrupted since this memory retains its stored data without power. RAM or other volatile memory can only be used for short and temporary periods to avoid the loss of large amounts of important data. Situations when volatile memory can be used are for example during the logging process. Data requested from connected sensors can be stored in RAM while the MCU does necessary processing before writing it to non-volatile memory for long-term storage. In this case, only the last log would be lost in case of power failure.

Although non-volatile memory is preferable, it is not without its problems. In the event of a sudden loss of power, the MCU is shut down in an uncontrolled fashion without turning off its flash memory chip the correct way. The MCU can be in any power state and stage of code execution during a potential loss of power. If the MCU is executing code for writing to the flash chip when a shutdown occurs, the risk of corrupting parts of the memory or suffering a complete loss of data is possible. The risk of corruption can be decreased with a multi-memory chip design, where only one of two or more memory chips are powered on at any one time and to which the MCU writes data. Additionally, at least one backup memory is needed in case of hardware failure. This can be for example an SPI Flash chip or other suitable external non-volatile flash memory. The ESP32-S2 model used in the prototype has 4 megabytes of integrated non-volatile flash storage that serves as the main storage of the system and is partitioned into smaller sections needed for different functionalities.

Table 5 describes how the flash memory in the prototype is configured including the offset and size of each partition. The otadata partition is needed for wireless over-the-air (OTA) functionality to be able to do firmware updates remotely in future development. In this partition is stored the part of the firmware used to manage the update. The

partitions app0 and app1 are used to store the actual firmware running on the MCU. These are needed for OTA updates to have a backup of the previous working firmware in case the update causes problems. The next partition holds a filesystem called SPIFFS, SPI Flash File System, that is used to store different files needed for the functionalities of the MCU for example the log file and configuration file. The panic handler makes use of the core dump partition to dump the state of the firmware when a serious error occurs that crashes the MCU. The nvs partition is used to store among other things Wi-Fi data if it is enabled (Espressif Systems, n.d.).

Table 5. Partition table for ESP32-S2 inbuilt flash memory used in the prototype.

Name	Type	SubType	Offset	Size	Flags
otadata	data	ota	0x9000	0x2000	
app0	app	ota_1	0x10000	0x140000	
app1	app	ota_2	0x150000	0x140000	
spiffs	data	spiffs	0x290000	0x14D000	
coredump	data	coredump	0x3DD000	0x10000	
nvs	data	nvs	0x3ED000	0x13000	

SPIFFS is a file system designed to be used in embedded systems with several useful features including support for wear leveling to make the flash memory wear evenly and last longer, checking the filesystem to find and repair inconsistencies, and it has a garbage collector that can scan for empty memory (Espressif Systems, n.d.). Its lack of support for sub-directories does not have an impact on how the system stores files since a flat file structure is enough for a couple of files. The documentation mentions that write operations can vary in time between writes. This problem needs to be taken into consideration if the logging interval is shorter than the time it takes to write. Time taken for mounting the files system, opening the log file, and then writing data to that file were observed to be consistent over the test period, but closing the file had a varying time

with the longest being 167 ms. Overall, the whole process always stayed below 500 ms which would result in a possible safe logging frequency of 2 Hz.

The log file is a text file that stores all sensor logs. The logs are stored in a comma-separated values structure (see Figure 15) with a header line describing the order of the values including log index, ADC sample value, time and date of the log, and the internal temperature of the MCU. The battery column is only a placeholder since measuring the actual battery charge level is not supported in the current prototype. Each saved log uses between 30 and 50 bytes of memory depending on how many characters are saved in each measurement data point. By using 50 bytes per log a memory size of 1 MB can store 20000 logs if all memory can be used without overhead.

```
#Index,Sample,TimeDate,Battery,Temp
0,27692,2024-11-18T22:53:36,34,27.20
1,28113,2024-11-18T22:54:36,34,24.20
2,27898,2024-11-18T22:55:36,34,23.20
3,28202,2024-11-18T22:56:36,34,22.20
4,27993,2024-11-18T22:57:36,34,22.20
5,28074,2024-11-18T22:58:36,34,21.20
6,27959,2024-11-18T22:59:36,34,21.20
7,27873,2024-11-18T23:00:36,34,21.20
8,28125,2024-11-18T23:01:36,34,21.20
9,27831,2024-11-18T23:02:35,34,20.20
10,27803,2024-11-18T23:03:35,34,21.20
```

Figure 15. Extract from log file in csv format.

4.6 Wireless communication and user interface

The wireless capabilities of the MCU provide remote access for retrieving data, visualizing live data, and system configuration through a graphical user interface. All these features are supported by a WebSocket server running on the MCU. The MCU can be configured to run in station mode or access point mode. A computer or other Wi-Fi-enabled

device can connect directly to the wireless network created by the MCU if it is using access point mode. In station mode the system needs to be configured with the SSID and password of the local wireless network it should connect to. This can, for example, be a portable Wi-Fi network created by a smartphone. The web server running on the system in communication mode can then be accessed over the local network.

The web server serves two different HTML webpages with CSS for styling and JavaScript for interactivity. One of them gives a live view of sensor readings and the other is a file browser that can be used in two different ways. A user can connect to the web server and use the graphical user interface for the file browser (see Figure 16) to download the log file for further analysis and the configuration file that is used to configure system behavior. After updating the config file, the user can upload it through the user interface. A request handler in the web server saves the updated configurations and initiates a system restart for the system to use the updated variables. The web server additionally supports direct access to the log file. A user can retrieve the log file over a local wireless network by using the local IP address of the web server followed by ”/logfile.txt”. After the web server receives this HTTP request, a handler will respond with the log file by sending it in chunks until the complete file has been sent.

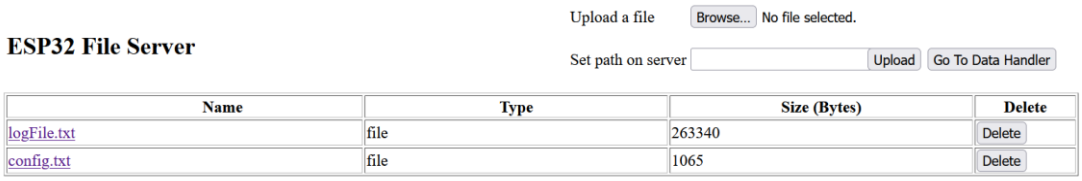

Figure 16. File browser running on the integrated web server on the system.

The overall system energy consumption increases significantly with peaks of 160 mA to 310 mA while transmitting data depending on power level according to its datasheet. This is much higher than in deep sleep. This sets higher requirements for energy storage demanding an increase in battery size compared to only data acquisition using a sensor and a microcontroller. Energy consumption can be kept to a minimum by decreasing the

interval of transmitting wireless data to a lower frequency than that of data logging. For example, a water pressure measurement is logged once every minute, but data is sent once an hour. Signal attenuation and penetration are other problems with wireless communication. Wireless communication is challenging from inside a deep well which would require the communication module or antenna to be installed on ground level making it visible and susceptible to theft.

4.7 Firmware and startup process flow

The development environment used is Visual Studio Code with the plugin PlatformIO to automate hardware configurations and firmware flashing. All firmware running on the system is C code and uses the framework esp-idf. This framework has support for all necessary features of the MCU including ULP FSM and ULP RISC-V programming.

The system can operate in two modes, communication mode and low-power autonomous logging mode. While in communication mode, wired UART communication over the debug port is enabled making it possible to send commands to debug problems as well as configure and read data. This mode additionally enables the wireless communication feature by starting the WebSocket server and making it available over a wireless network, either in station mode or access point mode. The MCU always starts code execution with a function called “app_main()” if it is programmed with the esp-idf framework. This function is always executed after the MCU is powered up for the first time or after waking up from deep sleep. The function starts by checking the source of the wakeup and based on that, determines if the MCU should start the logging process and then go back to deep sleep, or if it will go into communication mode (see Figure 17). An external interruption can be used to put the MCU into communication mode after the system has performed an initial system configuration. The logging process and communication modes are described in more detail in the following chapters.

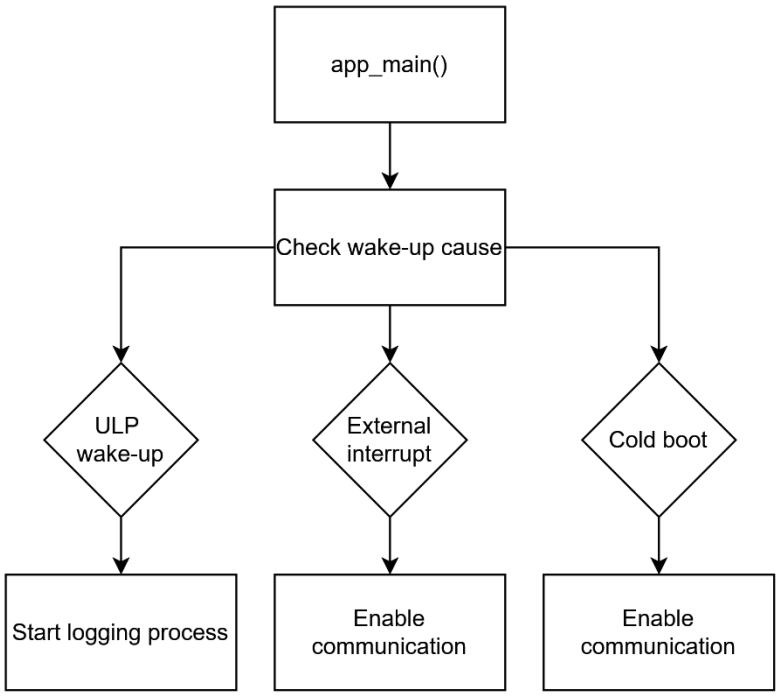

Figure 17. Firmware flowchart at system startup.

4.8 Communication mode

After the system powers up from a cold boot or deep sleep, and the startup source check procedure has finished and the system goes into communication mode, a setup function will run that will configure the MCU and start necessary tasks and processes. The steps inside the setup function including the order they are executed are described in Figure 18.

The first step inside the setup function is to create binary semaphores. They are initialized immediately inside the setup function or by another function later if necessary. They are needed at a later stage in the process when multiple tasks may use the same resources. The semaphores make sure a shared resource can only be accessed by one task at a time by acting as a guard. A race condition, deadlock, or possible data corruption may happen if multiple tasks try to access and modify the same resource at the same time. When a task wants to access a shared resource, it first needs to get the binary

semaphore if it is available. The semaphore may not be available in case another thread is accessing the shared resources in which case the task will wait its turn. In some cases, it is important that a resource is not inaccessible by a task for too long if the task is time sensitive. This may lead to a system reboot by the task watchdog timer. The next step in the setup function is to initialize the file system to be able to create files, write data logs to the log file and read the user modified config file. If the system is unable to parse the config file or if it does not exist, a new config file template with default values is automatically created and added to the file system.

At this stage, necessary tasks, or threads, are created using FreeRTOS which is a Real-Time Operating System designed for low-power embedded microcontroller systems (FreeRTOS, n.d.). Multiple tasks are necessary while the system is in communication mode because multiple processes need to be executed concurrently. The one high level CPU core of the MCU can only process one task at a time, forcing it to share its processing time by switching between running tasks. The FreeRTOS thread scheduler chooses which thread will get processing time next based on its state and priority level.

The important tasks consist of a logging task, a system reboot task, a deep sleep handler task, and a UART communication task. An independent task for logging is necessary while the system is in communication mode to be able to log data while ensuring it does not monopolize processor time preventing other critical functions from working. The logging process is explained in more detail in the following chapter. The system reboot task is used to initiate a system reboot once it is notified by a handler in the WebSocket server responsible for detecting when a user uploads a new configuration file. The UART communication task handles all UART communication between the MCU and the user over the debug port, and the last task is responsible for configuring necessary GPIO pins and enabling ULP wakeup before putting the system into deep sleep.

Following the task creation step is to establish a Wi-Fi connection according to its configuration set in the config file and upon a successful connection, start the WebSocket

server. An inactivity timer is additionally started in this step that is used to put the system into deep sleep if it does not detect a user interacting with the system for a predetermined timespan. The next step in the setup process is to update the MCU system time by synchronizing it using Simple Network Time Protocol, SNTP. This protocol is a scaled-down version of the more well-known Network Time Protocol, NTP, but designed for systems with low-power hardware.

The last step in the setup procedure is to do the necessary configurations to run the ULP. The first step in the case of the ESP32-S2 is to choose which one of the two low-power coprocessors to use since it has two, the ULP FSM and the ULP RISC-V, and only one of them can be configured to run at a time. This is done by setting the bit in the register corresponding to which ULP to use. The next step is to load the ULP assembly instructions embedded in the flash chip as binary code to RTC slow memory which is powered on during deep sleep. The timer responsible for waking up the ULP at a certain interval is set by choosing one of 5 timer registers and giving it a value in microseconds and lastly the ULP is started by giving it an entry point from where it will start running the code.

The last step in the setup function is to install the Interrupt Service Routine, ISR, which is a small piece of code that runs every time the ULP gives a wakeup interruption while the system is awake and in communication mode. If the system is in deep sleep, a wake-up interrupt simply wakes up the system. The ISR gives a binary semaphore which the logging task is waiting for. The ISR checks if the logging task has been unblocked, and if successful, yields for the logging task priority to give it priority to be processed next. This is important to minimize the time delay between ULP interrupt and reading the sensors.

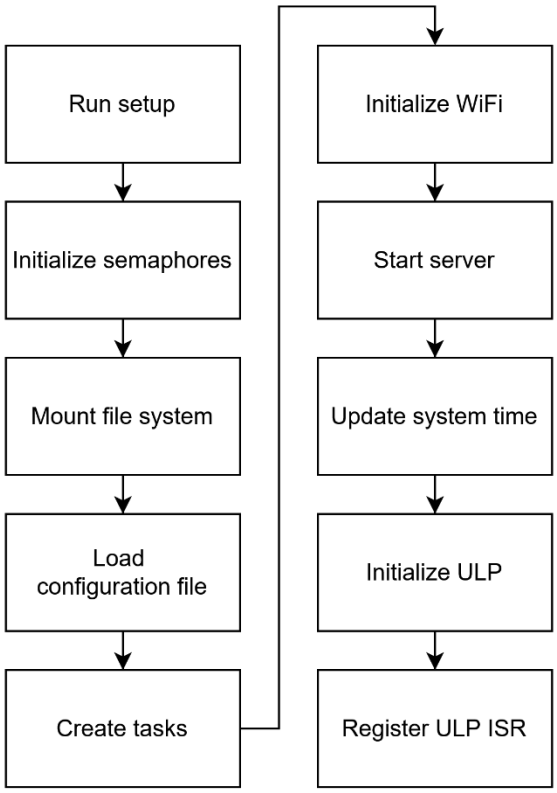

Figure 18. Program flowchart in communication mode.

4.9 Logging mode

Logging mode is similar to communication mode, but it does not configure anything in the system and stays in deep sleep for most of the time. The assembly code run by the ULP, which handles the timing interval, is the same as described in Algorithm 1, and it follows the same program flow as shown in Figure 11. The main difference between logging mode and communication mode is that after the MCU wakes up from deep sleep and performs a wakeup source check, it determines it was caused by the ULP and begins the logging process.

After the wakeup cause check has been performed, the logging process begins which includes the steps described in Figure 19. The first step is to record the system time to minimize delays in the timing. Next, the initialization of the I2C bus starts and the ADC

module and boost circuit are powered on through two NPN transistors controlled by GPIO pins on the MCU. The maximal power the ESP32-S2 is capable of outputting through its GPIO pins is limited making it necessary to feed power to the ADC module and the boost circuit directly from the USB power bank through transistors. The ADC module has a fast power up time, but the boost circuit needs a time delay to stabilize its 24 VDC output. It was determined after testing by continuously reading the sensor value measuring a constant water pressure to see when it becomes stable that a time delay of two seconds is sufficient. At this stage the MCU requests an ADC value from the ADC module over I2C and powers down the sensor side after it has received the data.

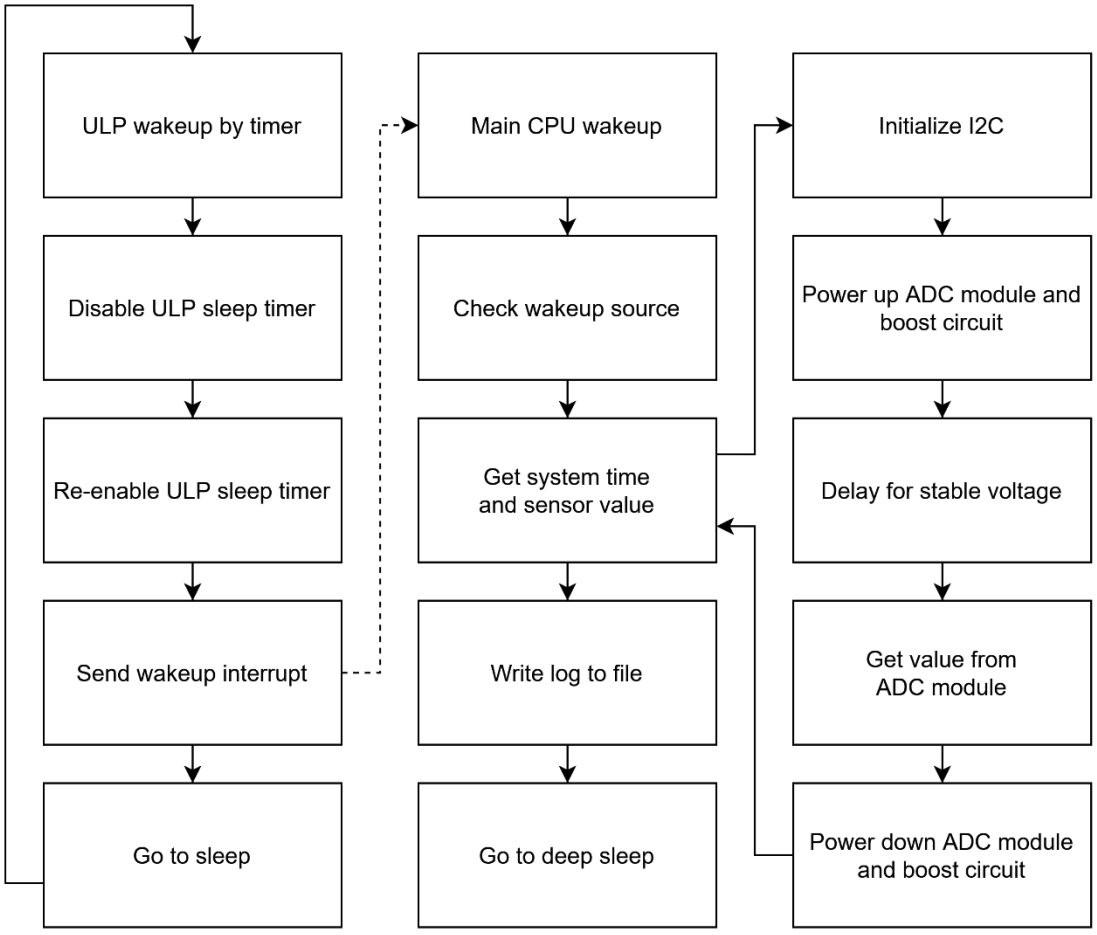

Figure 19. Program flowchart in logging mode.

5 Evaluation and Testing

This chapter discusses and presents a comprehensive evaluation of the prototype beginning with a cost breakdown for each of the components used in the prototype. The performance of the sensor is then evaluated on multiple points including long-term stability with compensation for estimated water evaporation and accuracy over a range of pressures. Additionally, noise in the measurement is analyzed and the noise suppression effectiveness of some common filters is tested and analyzed. Finally, other system functions are evaluated including wakeup time, ADC oscillations, and the potential problem of electromagnetic interference in the intended application.

5.1 Prototype cost overview

This chapter presents a breakdown of component costs associated with the sensor prototype. Only the cost of the hardware components of the prototype are considered excluding costs related to development and software. Table 6 provides an overview of the system cost on a component level in SEK without sales tax. The relatively high cost for the ESP32-S2 MCU is due to being a development board that includes additional features such as an addressable color LED, an easy-to-use I2C connector, and support for battery cell connection straight to the printed circuit board (PCB). The price for a standalone ESP32-S2 microcontroller or module is lower. The ADC module is optional but makes development faster and removes the need to use the internal ADC of the ESP32-S2 which lacks accuracy. An alternative approach would be to use another microcontroller with a higher precision integrated ADC or to use an external calibrated ADC chip with sufficient accuracy.

The cost for the sensor can vary significantly depending on its specification, calibration, and quality guarantees. The sensor used in the prototype was selected for its balance between price and ease of use and was widely available at the time of purchase. The last row in Table 6 includes the price of the plastic enclosure, protoboard, and other

inexpensive miscellaneous components such as transistors, wires, resistors, and solder connections. In total, the estimated cost of the prototype amounts to approximately 700 SEK, or 600 SEK if only the electronics are calculated.

Table 6. Prototype 1 component price breakdown.

Microcontroller	143,2 SEK
ADC module	70,20 SEK
Boost circuit	39,2 SEK
4-20 mA sensor	320 SEK
Power bank	40 SEK
Container and miscellaneous	100 SEK

5.2 Sensor stability analysis

Sensor stability performance was tested over a period of approximately 80 hours without moving the sensor or changing the water level. The test was performed indoors in a temperature controlled and ventilated room by continuously logging the water level inside a pipe autonomously once every minute for the test period. The temperature was set to 22 °C. The two main factors affecting the water level and the stability of the sensor readings are the sensor side of the system and uncontrolled water evaporation. The sensor side includes the sensor itself, the boost circuit, and the ADC module. Figure 20 shows the logged level in blue and its calculated trendline in red. The yellow and green lines represent the estimated water evaporation with 60% and 30% relative humidity respectively. How the evaporation estimation is calculated is described below. The water level started at 283,4 cm based on the calculated average for the first half hour and decreased to an average of 282,1 cm for the last half hour. This is a decrease of 1,3 cm over the sampling period. The average for the first half hour was used as the starting point for the evaporation estimation.

As can be seen from Figure 20, the red trendline for the sensor values decreases faster than both evaporation estimation lines. The 60% relative humidity estimation decreases by 0,4208 cm and the 30% relative humidity decreases by 0,7368 cm over a period of 80 hours. This difference can be a result of several factors. The evaporation rate may be inaccurate due to not having precise measurements for the air velocity and relative humidity leading to estimations, although they are based on realistic values. Another potential source of the inaccuracy may come from the sensor side of the system. The sensor itself may not be accurate over time which can cause a drift in the sensor value, and the output of the boost circuit may vary over time leading to a drift in voltage, which in turn causes a drift of the sensor measurement. Additionally, the ADC module may cause a deviation although it was not observed to have a major deviation in the development phase.

Further testing of the system and its components is required to be able to assess the long-term stability. Conducting further tests within an environmentally controlled test space helps reduce the effect unknown factors have on stability. For example, knowing the precise water evaporation rate removes it as a potentially significant source of error. Moreover, an improved voltage source with higher precision and stability can minimize its potential impact on the variations in measurements. Certain remaining unknown factors are the potential impact of the pipe and the decreasing water level on measurement accuracy, the effect the sensor itself has on the level, and if small changes in atmospheric pressure can have a significant enough effect on the sensor accuracy since, as mentioned in previous chapters, it has inbuilt compensation.

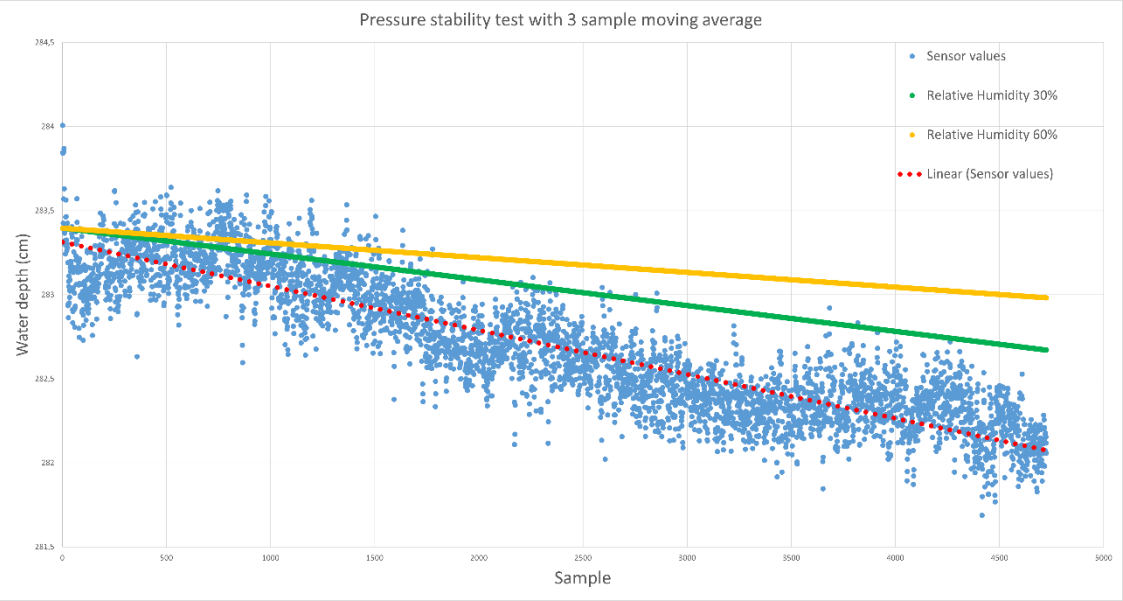

Figure 20. Sensor stability over approximately 80 hours with a 3-sample moving average, trend-line, and water evaporation estimation with 30% and 60% relative humidity.

The decrease in water level resulting from water evaporation was not measured but can be estimated using the equations presented in the article by Shah (2021). This article considered methods, models, and data regarding water evaporation estimation under several different conditions and types of bodies of water. According to Shah (2021), if the layer of air immediately above the water surface gets saturated and does not circulate, evaporation slows down significantly. The rate of evaporation increases with an increase in air velocity. The author presents two different ways that the immediate air layer can circulate, natural convection and forced convection. The latter can occur from wind or from ventilation systems found indoors. Of all the tested models the most accurate model presented achieved a 16,1% mean absolute deviation (Shah, 2021).

The following equation recommended by Shah (2021) can be used to estimate water evaporation in cases where forced convection is higher than natural convection and there is no disturbance in the water surface.

$$E_{fc,minm} = 0{,}00005 * (p_w - p_\infty) * \left(\frac{u}{0{,}12}\right)^{0{,}8}, \quad (7)$$

where $E_{fc,minm}$ is the water evaporation from forced convection measured in kg/m^{2}h. The saturated air density above the water is represented by p_w, and p_∞ unsaturated air density further away from the water measured in Pascal. Airspeed in m/s is unitless and represented by *u* (Shah, 2021).

A velocity *u* of 0,12 should be used in cases where the velocity is less than 0,12 m/s. This leads to a simplified version of the equation

$$E_{fc,minm} = 0{,}00005 * (p_w - p_\infty), \quad (8)$$

where the last parenthesis is removed. This equation additionally gives the minimum evaporation in case of forced convection (Shah, 2021).

Air velocity caused by ventilation in the test space was not recorded but can be estimated using the ANSI standard 55-2017 (Pande et al., 2020) for indoor ventilation. According to the standard, ventilated residential indoor spaces should have a minimum air velocity of 0,1 m/s and not exceed 0,2 m/s when the indoor temperature is 23 °C or below. Based on the description for equation 8, the air velocity is likely low enough to not have a significant effect and will be disregarded to simplify the evaporation estimation (Olesen et al., 2002).

The saturated pressure, p_w, can be calculated or taken from a lookup table. According to Oklahoma State University (2001), the pressure at 22 °C is 19,8 mmHg, or 2639,783 Pascal, which is the unit required for the equation. The partial vapor pressure can be calculated using the following equation

$$p_\infty = RH * p_w, \quad (9)$$

where p_∞ is the partial pressure, p_w the saturated pressure, and RH the relative humidity of the surrounding unsaturated air (Lonberg, n.d.). Since relative humidity was not recorded in the test space, a reasonable lower and upper bound of 30% and 60% respectively were chosen based on the recommended indoor relative humidity to avoid mold (IUOE National Training Fund, 2018).

Using equation 9 with the saturated pressure and the upper and lower bounds for the relative humidity, we get a partial pressure of 1583,869 Pa and 791,9349 Pa respectively. At this stage the water evaporation can be estimated using equation 7 with the calculated values. This results in a lower evaporation rate of 0.0527957 kg/m^{2}h and upper of 0.092392405 kg/m^{2}h.

The inner diameter of the test pipe of 70 mm gives a surface area of 0,003848 m^2. Scaling this evaporation rate to the water surface area in the test results in a rate of 0,000203157 kg/h and 0,000355526 kg/h. Converting this weight to volume using 0.997770 g/cm^3 as the weight of water at 22 °C (Density of Water, n.d.) results in a volume of 0,20270395989 cm^3/h and 0,35473317702 cm^3/h. Using the cylinder volume equation, v=pi*r2*h, a change in height resulting from evaporation can be calculated. This results in a height change of 0,00526 cm/h and 0,00921 cm/h, or 0,4208 cm and 0,7368 cm over a period of 80 hours.

5.3 Evaluation of sensor accuracy over range

This chapter evaluates the accuracy of the sensor in a range of pressures. The test was performed by gradually lowering the sensor down a water-filled pipe starting from no pressure and then incrementally lowering the sensor down in steps, each step increasing the depth by 10 centimeters, until reaching the maximum pressure at a depth of 200 cm equal to 20 steps. Measurement marks were added onto the sensor cable indicating each step of 10 centimeters to provide a guide. All ADC values were post-processed on an external computer, although the MCU is capable of some data processing. The ADC

depth values from the sensor were converted to centimeters and then calibrated so that the sensor readings at a known depth of 20 cm correspond to the 20 cm mark, with all other values adjusted accordingly.

The ADC module was configured to output the measured mA from the sensor as raw ADC values to get the highest possible resolution. If configured to output mA, the resolution is limited to two decimal places, corresponding to a distance resolution of 3,125 mm after conversion. However, by using raw ADC values, the resolution improves to 0,119212245 mm although the overall accuracy of the sensor and ADC module remains lower than the maximal achievable resolution.

The raw ADC values were first processed to remove values outside of the interesting range corresponding to 4-20 mA. The remaining range of valid ADC values were then converted to distance by dividing the range of the sensor, 5 meters, by the number of ADC steps in the range. This process is visualized in Figure 21. Lastly, all data points were calibrated by a correction factor to have a calibrated reference point. The reference depth used was 20 cm and the correction factor is the calculated difference in distance between the average for all data points at the 20 cm depth mark compared to 20 cm.

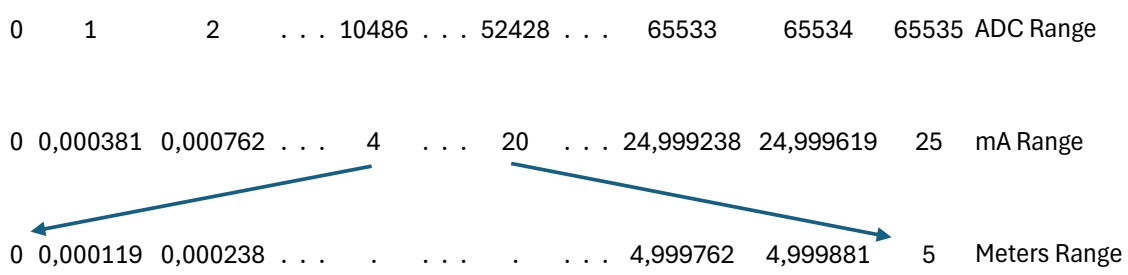

Figure 21. Conversion process from 16-bit ADC range, via sensor mA range, to distance in meters.

The graph in Figure 22 shows the relation between the expected depth versus the measured depth. The expected depth points are marked in blue and the measured depth in

orange. The points representing the measured depth are the calculated average depth value for all measured points for each step. From Figure 22 can be seen that the measured depth points follow a clear linear progression with a calculated average increase of 12,43 cm compared to the expected 10 cm with each step. The first step and last step are not included in the calculations since they are less accurate.

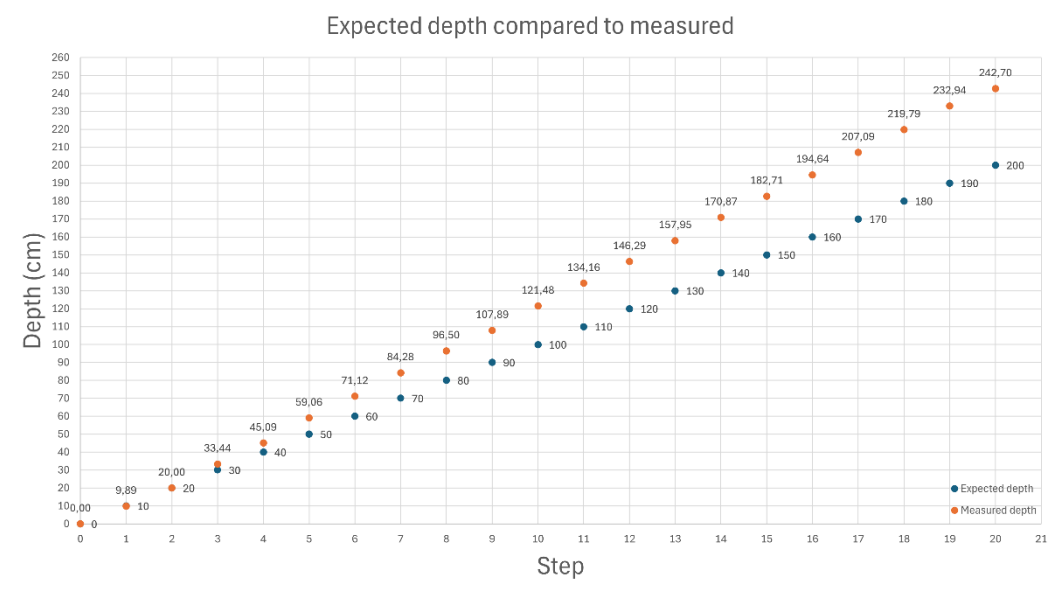

Figure 22. Expected depth compared to the average depth at each step measured by the sensor.

A best-fit line for the measured data points can be calculated using least squares linear regression. This method works by finding a best-fit line for the data points by minimizing the sum of the squared distances between the measured samples and the predicted best-fitting line. This technique is suitable for data sets that are linear with few outliers since they can distort the prediction reducing its accuracy (GeeksforGeeks, 2025). As this regression method is sensitive to noise and outliers, a moving average, windowed median filtering, or other suitable data processing method can be applied to the data to decrease the effect outliers on them, however, the depth correction presented below proved good results without additional data processing. After testing with and without a moving average, an insignificant difference of less than 2 mm was seen between the results. This may be a result of the large number of data points which make the outliers have a lesser impact.

The measured values can be adjusted using the straight-line equation with modified variables

$$y = a_{slope}x + b_{offset}, \tag{10}$$

where *y* is the adjusted measurement value, a_{slope} is the slope correction factor, *x* is the original measurement, and b_{offset} is the offset correction factor. The variable a_{slope} is the slope of the best fit line for the measured values divided by the slope of the expected values. The correction factors can be derived from the following equation where the slope and offset values for the measured data series are m' and c'.

$$mx + c = a_{slope}(m'x + c') + b_{offset} \tag{11}$$

$$a_{slope} = \frac{m}{m'} \qquad b_{offset} = c - ac', \tag{12}$$

These equations give an a_{slope} value of 0,805 and b_{offset} of 2,784. By applying these adjustment values to the measured data, a clear improvement is achieved by going from a max deviation of 42,935 cm and an average of 11,559 cm before correction to a max deviation of 1,124 cm, min deviation of 0,018 cm, and average deviation of 0,221 cm from the expected values after correction. Additionally, the calculated best-fit line has an R^2 value of 0,9999. An R^2 value of 1 represents a perfect fit. The original measurements versus the adjusted values are graphed in Figure 23. The adjusted values in blue show a clear improvement over the original measurements in orange. The adjusted values closely follow the expected increase in depth of 10 cm for each step.

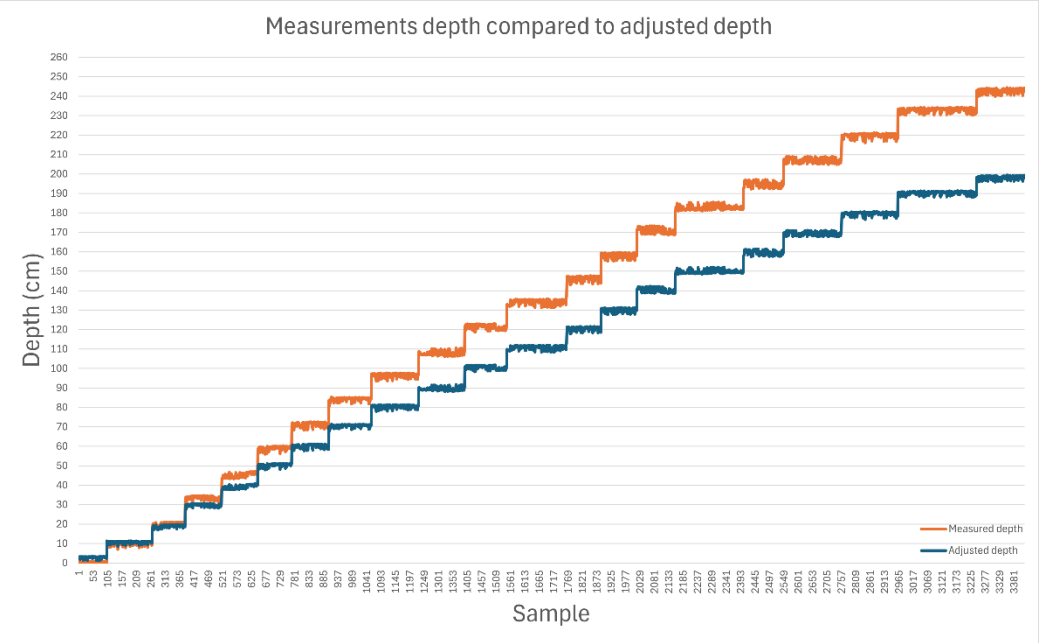

Figure 23. Measured depth compared to adjusted depth.

Although the measurements increase linearly in the tested pressure range, it is unknown if the measurements follow a linear increase over the entire five-meter span. The sensor would need to be tested several times over its entire pressure range to verify its linearity and repeatability. The are several potential factors causing the deviation of measured depth relative to expected depth, although due to the short test duration, the negligible effect of water evaporation can be disregarded. Depth markings were manually measured and added to the sensor cable to indicate each step of 10 cm. The markings may contribute to a part of the deviation, but only by a few millimeters. Disturbance of the sensor is another unlikely source. The sensor was disturbed only when lowered from step to step, leaving plenty of time for any possible water turbulence to settle, eliminating this as a potential source of error. The last possible major factor causing deviation is the hardware itself.

5.4 Analysis of depth measurement noise

This chapter analyses the measurement points and different filtering methods. The points used in the analysis are from step 15 of the test in the previous chapter. Step 15 has an average depth of 182,74 cm before adjustments. The first and last five samples in the step are removed since they might be influenced by moving the sensor between steps. This chapter only analyses the measurements of step 15 which are expected to show a steady water level. Analyses of all expected level behaviors, including stable level, increasing, and decreasing changes at different speeds, are needed to get the full understanding of the accuracy, noise, and best filtering techniques to reduce variations and noise in the sensor measurements. Other filtering techniques may give better results in the event of a changing water level compared to the ones used in this analysis.

Figure 24 and Figure 25 show the original unfiltered measurements as blue points. From these figures can be seen that the sensor measurements include variations or noise since the measured level was stable. This noise gives the measurements a variance of 1,39 and a difference of 4,67 cm (see Table 7) from the largest to the smallest measurement while the average depth is 182,74 cm for all depth measurements. Additionally, the variations in the measurements do not follow a pattern and seem to be random.

Different techniques exist to denoise a noisy sensor signal to be able to extract useful information. The techniques presented below to filter the signal are three simpler filters and a more advanced standard Kalman filter. The three simpler filters are a Moving Average with a window size of three samples, a sliding five sample Windowed Median in combination with a Weighted Moving Average, and a Trimmed Average. The second filter works by filtering the data twice, first with a sliding median, and then with a weighted average using the weights 3, 2, and 1. Most weight is put on the most recent sample. The trimmed average was configured to remove 20 % of the outliers, in this case 2 out of 5 samples, and then calculate the average of the remaining.

The more advanced standard Kalman filter, also known as Linear Least Mean Squares Estimator, is based on the filter design proposed by Ma'arif et al. (2019) for reducing the effect of noise in a signal when there is no known noise frequency. Frequency-based filters, for example, Low Pass, High Pass, and Band Pass Filters are not suitable in this case since they are more effective at filtering out specific ranges of frequencies. The computations performed by the standard Kalman filter have low hardware requirements making this type of filter potential to be used locally on the MCU. The standard Kalman filter can have generally good performance according to the authors, but other versions of the Kalman filter exist which can have better performance in specific applications.

The working principle of the Kalman filter is to use known samples to predict future samples. The filter design has two constants, the variance constant Q, and the measurement constant R. Changing the value of these constants has an effect on how the filter performs. There are no predefined best values for these constants, instead, Ma'arif et al. (2019) suggest they need to be adjusted based on the filter performance and expected outcome. Increasing the ratio between Q and R will increase the noise dampening effect, while too great a ratio can lead to dampening of real signal changes. The author recommends a balanced ratio of 100. The filter implementation in Python is shown in Appendix 1.

The Kalman filter was tested with three different ratios of Q and R to see how well the filter performs. The constant value pairs are displayed in Table 8. The filter needs an initial estimate of the depth and initial uncertainty to work. The estimate used was the average for the first three samples and the uncertainty was set to one as suggested in the paper.

The original unfiltered results and the outcome of the first three filters are graphed in Figure 24 and their statistical values are in Table 7. From this graph can be seen that the original measurements have a wide spread with a 4,67 cm difference between the highest and lowest measurements and a variance of 1,39 which is a measurement of the

spread of data calculated as the average for all squared differences between the measurements and the average for all points in the series. The lower the variance is from the original, the better the noise is suppressed. All the tested filters produce a lower variance compared to the original with varying success. The trimmed average filter performed the best out of the simpler filter designs with the lowest variance, however from Figure 24 can be seen that the filter is not aggressive enough to not be affected by the noise leading to a distance variation of 2,91 cm between the highest and lowest measurements. The simpler filters were tested in different configurations and in general performed similarly. The most significant performance boost gained by the filters in terms of smoothing the signal was achieved by increasing the window size to use more samples. This would, however, lead to an increase in the samples needed for the filter to work as expected, and more computation is needed making it less suitable for onboard processing. A low computation filter performing well without the need for multiple samples is preferable. Additionally, filter designs relying on past data may have an unfavorable delay in response to real changes in data. Median-based filters proved to be better at filtering outliers compared to average-based filters during the analysis, although both types of filters were able to reduce the largest spikes.

The more advanced Kalman filter showed promising results while having an easily adjustable noise-dampening effect and not needing multiple samples due to its internal design of continuously updating itself based on past samples. Figure 25 plots the Kalman filter response with the three tested ratios of Q and R compared to the original measurements. The lowest ratio of 1 produces a small improvement in dampening the noise compared to the original while increasing the ratio to 100 produces a clear improvement in noise reduction. The largest ratio of 10000 removes almost all noise in the signal, reducing the difference between the largest and the smallest measurement to 0,55 cm. Although the largest tested ratio gives the best noise reduction, it may not give the most optimal filter performance in all cases, especially when the real water level changes significantly in a short amount of time. Additional testing is needed to cover all expected behaviors, especially with measurements from a changing water level to find an optimal

type of filter that has a balance between responding to a changing water level, degree of smoothing, samples needed, and calculation demands if implemented directly on the microcontroller.

Table 7. Statistics for original measurements versus filtered measurements.

	Unfiltered Measurements	3 Sample Moving Average	5 Sample WM - WMA 3-2-1	Trimmed Average 5-2
MIN	180,98	181,45	181,49	181,51
MAX	185,65	184,72	184,44	184,42
MAX - MIN	4,67	3,27	2,94	2,91
AVERAGE	182,74	182,75	182,64	182,75
MEDIAN	182,44	182,59	182,42	182,59
VARIANCE	1,39	0,67	0,58	0,51

Table 8. Statistics for original measurements versus filtered using Standard Kalman with three variations of Q and R values.

	Unfiltered Measurements	Kalman Q1=1 R1=1	Kalman Q2=0,1 R2=10	Kalman Q3=0,01 R3=100
MIN	180,98	181,24	181,89	182,61
MAX	185,65	185,24	184,07	183,15
MAX - MIN	4,67	4,00	2,18	0,55
AVERAGE	182,74	182,75	182,77	182,89
MEDIAN	182,44	182,52	182,64	182,88
VARIANCE	1,39	0,79	0,27	0,02

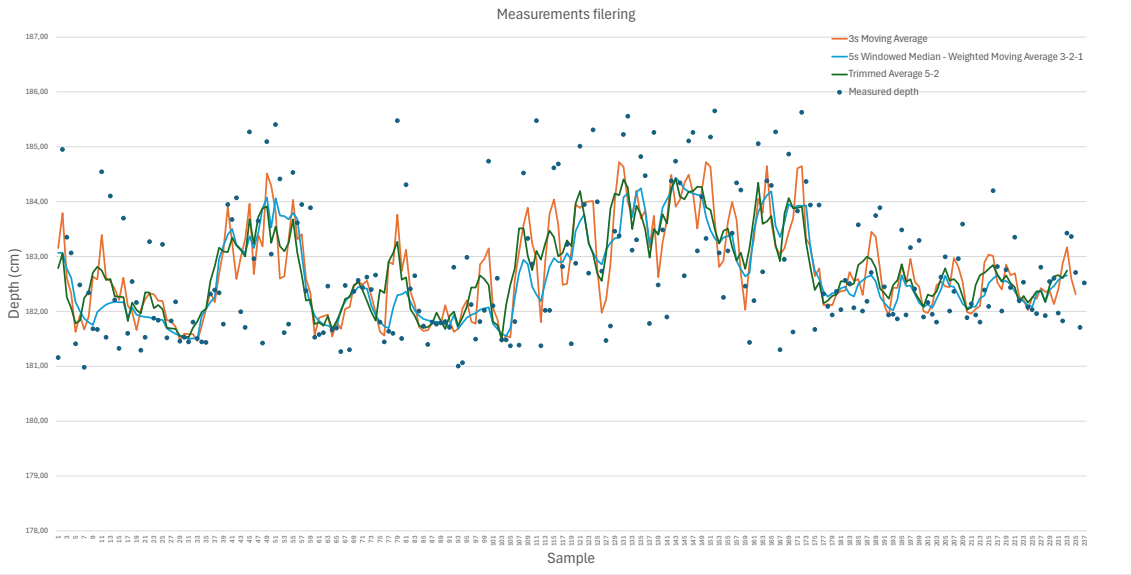

Figure 24. Measurements filtering using three different filtering techniques.

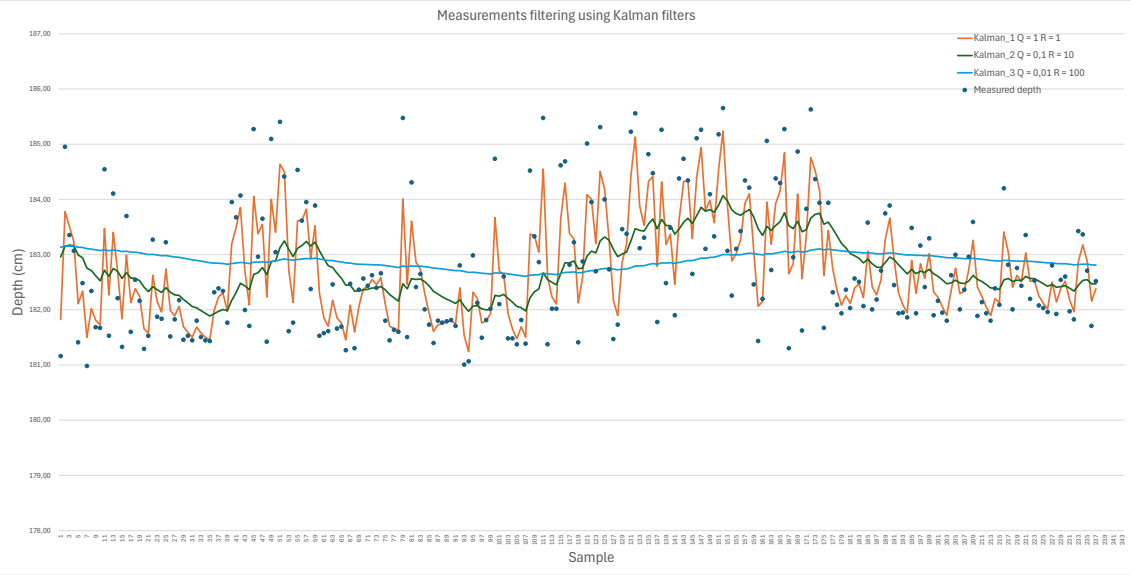

Figure 25. Measurements filtering using standard Kalman filter evaluated with different Q and R values.

5.5 Wakeup latency, oscillations, and EMI

Another downside of the system is the time needed until the sensor can provide an accurate mA signal that can be read by the ADC module after waking up from deep sleep.

This causes an increase in overall energy consumption since, as previously mentioned, energy usage is significantly higher while the system is awake. To maximize battery operation time, the logging process needs to be as short as possible making the system stay in deep sleep for more of the time. The time it takes for the MCU to start up from deep sleep is negligible relative to the overall time delay for the system. This time delay is caused by a combination of the ADC module, the boost circuit, and the sensor. It is unknown how long the startup time is for each component, but it takes approximately two seconds after system startup until the MCU reads a stabilized ADC value from the module. This was observed in a test where the ADC module was polled by the MCU continuously with a high frequency starting immediately after powering on the system. Additionally, the refresh rate of how often the ADC module measures the mA signal is also a potential delaying factor. This makes it mandatory for the system to wait a minimum of two seconds every time logging process after the system wakes up from deep sleep. The delay is minimal if the boost circuit, sensor, and ADC module are powered on continuously, but this would require a significant increase in battery capacity to get a usable battery run time.

Upon closer inspection of the data collected from the previous test, the following problem was discovered. When the MCU polled the ADC module with max frequency beginning immediately after powering on the electronic components, the ADC value was observed to continuously oscillate within a small but still significant error margin after the ADC value reached its expected stable value. The graph in Figure 26 shows the ADC values from two different tests and their absolute error to series average from increasing sample average. The second test sampling was performed immediately after the first test. The water level stayed unchanged during the tests. From the graph can be seen that the variations are not changing at the same rate nor at the same time. This means if the MCU requests the ADC value from the module after two seconds from powering up, it may be the highest value, while another time the value after two seconds may be the lowest. Calculated from the test the highest value recorded was 28167 and lowest 27708. By

using the conversion process described in Figure 21, this worst-case scenario would result in a variation of 5,47 cm.

The variation can be decreased by averaging multiple samples. The absolute error from an increasing number of averaged samples to series average for both tests (see Figure 26) show that by increasing the number of samples in the average, the absolute error decreases. Series 3 starts with a 2,5 cm absolute error but quickly drops below 1 cm and stays at or below 0,27 cm error from the 18th sample onward. Series 5 starts with a smaller absolute error of 0,95 cm and stays at or below 0,37 cm 13th sample onwards. This method would improve accuracy but require several samples decreasing the logging frequency and making the system stay awake for a longer time decreasing battery life even further. Implementing a more advanced algorithm to find the average value could make the process faster with fewer samples needed, but it would still need more than one sample and lead to more calculations on the MCU using more energy. The low sampling frequency of the ADC module is another factor that makes this process slower. The datasheet for the module does not specify at which frequency it refreshes, but it is approximately 2 Hz based on testing. The averaging process would potentially be shorter with a higher refresh rate. This, however, requires changing out the current ADC module and replacing it with another design, either with another external module better suited for this application, or using the internal ADC of the MCU.

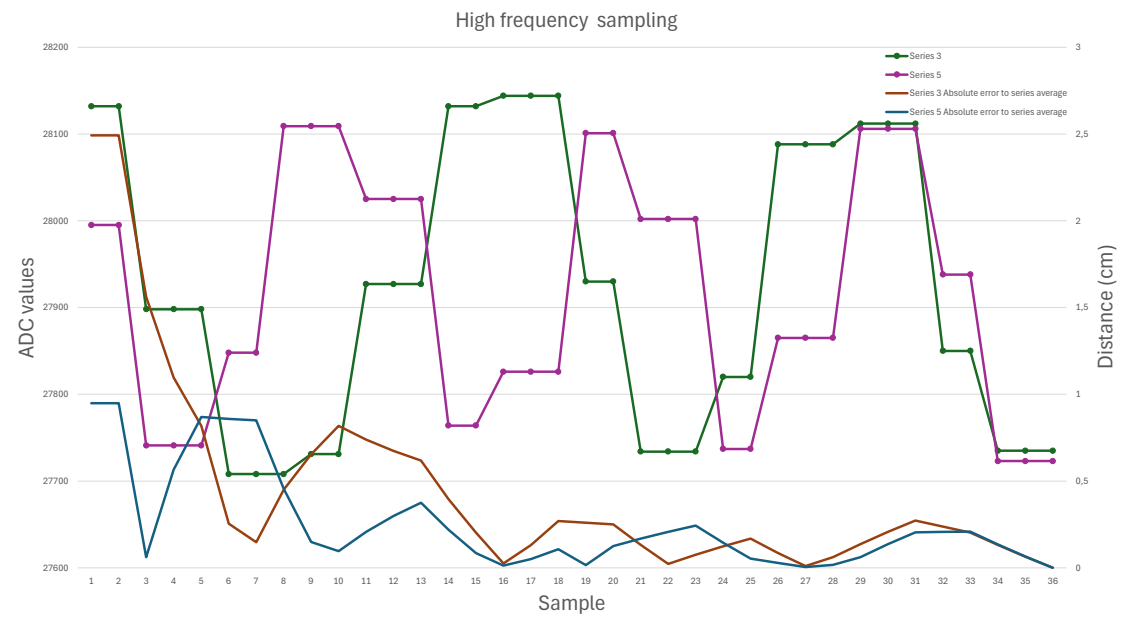

Figure 26. Two different series of the MCU polling the ADC module with maximum frequency on the left y-axis. The absolute error to series average in centimeter distance on the right y-axis.

Another potential issue is radiation from electromagnetic fields which can interfere with the system. In the intended application for this prototype, electrically powered pumping equipment is inside the borehole close to the data logger, sensor, and included cable. The pump and its power supply are a potential source of this radiation. Some or all parts of the system may be susceptible to this radiation and the potential problems can range from operational problems while the pump is running to damage to the sensitive electronics. Electromagnetic shielding can reduce the problems, but additional testing is needed to determine the limitations of the current system design and if shielding is necessary and possible to add.

6 Concept for Improved Design

In this chapter, an improved design concept is presented addressing issues found with the prototype including suggesting improvements. The three major weaknesses discovered in the prototype are sensor performance, time deviation, and energy consumption. These factors severely affect the usability and reliability of the system. The time deviation can be solved to within an acceptable range with a more accurate clock signal from an external source. The ESP32-S2 can then use this clock signal for its internal timers. Changing out the sensor for a different one with higher accuracy and lower energy consumption improves precision and battery life. The suggested sensor additionally makes some of the other components redundant. By removing these components, their cost can be removed from the overall system cost and the prototype can be made smaller in size with lower energy consumption.

6.1 Improving timekeeping accuracy

Testing of the inbuilt clock sources of the ESP32-S2 available during deep sleep showed that they have a frequency that is unstable and causes the system clock to deviate outside the error margin for the application. An accurate clock signal is necessary to keep the logging period consistent and to know when each measurement was taken. The system clock can be synchronized manually or using SNTP, but due to the unstable frequency of the clock signal it would need to be synchronized frequently requiring wireless connectivity and processing time. Additionally, no compensation for environmental variations is present in the MCU.

This problem can be solved by using an external clock signal with higher accuracy and less deviation. One affordable example compatible with the ESP32-S2 and the design of the system is the DS3231 Real Time Clock (RTC) (see Figure 27). It can improve the time accuracy of the system to within an acceptable range and provide time and date keeping. The DS3231 RTC clock has several advantages over the inbuilt clocks of the ESP32-S2. It

is based on a crystal oscillator, compared to the RC oscillator circuit found in the ESP32-S2, with compensation for variations in frequency due to temperature changes. The temperature range in a borehole is unknown but is expected to change more at a shallow depth closer to the surface. A wide temperature range will, however, unlikely be a problem since the DS3231 is designed to operate in temperatures between – 40 °C to 85 °C. The DS3231 chip claims an accuracy of ±3.5ppm within this temperature range and improves to ±2ppm in the smaller range of 0°C to +40 °C. Converting the worse accuracy to time deviation over a year would result in ±1,84 minutes falling inside the acceptable deviation range. Additionally, the RTC clock has an inbuilt feature to compensate for frequency variations from aging through a user-configurable offset.

The DS3231 module design shown in Figure 27 includes a charging circuit for its inbuilt battery backup system using an onboard coin-type battery cell. The module can run from external power, its own battery, or external power with its own battery as backup in case of power loss. Powering the clock chip using the external power that also powers the MCU does not require additional components for voltage conversion since they can run on the same voltage. The current needed with the lowest current draw configuration is less than 3 µA. This makes the chip suitable for an application where battery lifetime is important. The clock chip supports I2C making integrating communication to the MCU simple with the current prototype. It can be connected to the existing I2C bus the MCU already uses for communication with the ADC module. The 32 kHz clock signal from the DS3231 can be connected to the GPIO supporting external clock signals called XTAL_32K_P. The MCU can then be configured in firmware to use this clock signal while in deep sleep and for system time.

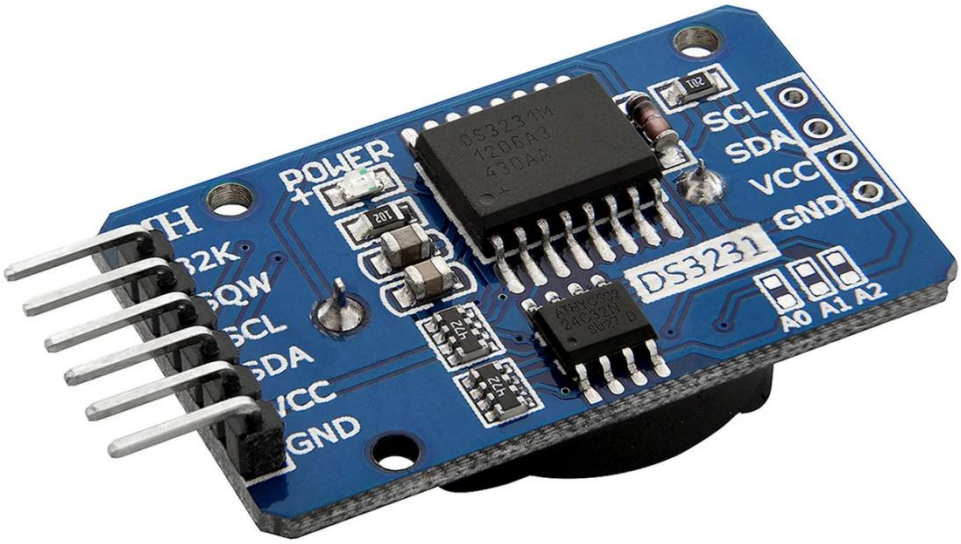

Figure 27. DS3231 Real Time Clock module on development board (adapted from AZ-Delivery, n.d.).

6.2 Pressure sensor optimization

The tested 4-20 mA pressure sensor is usable, but it has limitations. A boost circuit is needed to up the 5 V USB to 24 V for the sensor and reading its analog mA signal requires additional complexity with the implementation of analog to digital conversion. The boost circuit also adds additional power consumption to the system and the time delay until a stable depth reading is possible after waking up from deep sleep is too long. Despite the sensor being usable in some cases it does not fully meet the expectations and requirements set for the application. These factors make it less than ideal for long-term deployment in the field.

The tested sensor is a gauge-based sensor that needs a ventilation tube to the surface to equalize the reference pressure with atmospheric pressure. An alternative approach is to use an absolute-based pressure sensor, which does not require a ventilation tube. Instead, the pressure data would be adjusted afterward with atmospheric pressure data from a sensor located on the surface. This approach would require an additional sensitive pressure sensor and data post-processing, but it would eliminate the need for

ventilation, and if integrated into the data logger, it would eliminate the need for a data cable.

Selecting the optimal sensor for this application is crucial. Testing of the current sensor has provided a better understanding of what criteria the sensor should meet. An improved pressure sensor must be suitable for clean water environments, but some resistance to harsher liquids is preferable. The sensor is expected to only be in contact with non-contaminated groundwater, but it may contain minerals and potentially salt from seawater intrusion. The 5-meter maximum depth of the tested sensor severely limits its application. There exist sensors of the same type capable of measuring higher pressures, but they get more expensive especially as they get more accurate. A sensor with a higher maximum pressure of 5 bar or more would make it a better fit for deployment in a broader variety of boreholes compared to the 0,5 bar of the tested sensor. A pressure of 5 bar corresponds to a water depth of 50 meters, but 100 meters would make the sensor more versatile and able to cover a broader range of applications. Another key consideration is the sensor accuracy and resolution. A specific distance resolution is not defined but it should have a resolution that is sufficient for an accuracy of 10 cm over a distance of 100 meters, which is an acceptable precision.

The size of the sensor is not specified but should be compact enough to fit inside a narrow borehole. One major area where the tested sensor and its required components fall short is their startup time. A shorter time between waking up and providing a stable measurement would make the system stay in a low-power mode for more of the time and decrease overall power consumption. Another factor to consider in the area of power saving is the operating voltage of the sensor. The tested sensor needs 24 V to operate effectively. A sensor using the same voltage as the rest of the system would eliminate the hardware needed for voltage conversion and the efficiency loss coming from it. The last point not to forget is the price. It needs to be cheap enough to make it affordable to implement.

A potential replacement sensor is the MS5837-30BA from TE connectivity. It provides significant improvements over the tested sensor according to its datasheet (TE Connectivity Sensors, 2019). It is designed for high-pressure liquid applications, with a max measurable pressure of 30 bar, making it suitable for deployment in boreholes with water depths reaching 300 meters. Unlike the gauge-based sensor in the prototype, the suggested replacement sensor is an absolute pressure sensor, meaning it measures pressure relative to a vacuum. This eliminates the need for a ventilation tube connection to the surface but makes it mandatory to use an atmospheric pressure sensor for adjusting.

Like the DS3231 RTC clock chip, the MS5837-30BA pressure sensor has a wide safe operating temperature range of -20 °C to +85 °C exceeding the expected temperature changes deep inside a borehole. Its inbuilt 24-bit ADC provides a 0,2 cm resolution over the 30 bar pressure range if used in water. It has a claimed typical accuracy error of less than ±50 mbar between 0 °C and 45 °C in the pressure range of 0 to 6 bar, however, the typical error between 25 °C to 45 °C in the pressure range of 0 to 10 bar is less than 10 mbar. The accuracy error gets worse as the temperature increases or decreases outside of the mentioned ranges as well as variations in supply voltage. All sensors come factory-calibrated with six coefficients from two temperatures and pressures. This data is stored and accessible from onboard memory using a connected microcontroller and can then be used in a manufacturer provided algorithm by the microcontroller to adjust the measurements from the sensor.

In terms of energy efficiency, the MS5837-30BA sensor severely cuts energy consumption and in turn, can provide a substantial improvement in battery life. The sensor claims a typical power consumption of 0,01 µA while in standby mode in addition to having the same supply voltage requirements as the MCU. This further decreases the overall energy consumption compared to the tested prototype by not needing voltage conversion. The datasheet does not explicitly mention startup time from standby, but it does mention conversion time. The claimed typical conversion time ranges from 0,54 ms to 16,44 ms, depending on the configured oversampling rate. The highest current draw of the sensor

is seen during conversion. The typical peak current according to the datasheet is 1,25 mA. The significantly lower power requirements and shorter time needed to be awake of the MS5837-30BA compared to the tested 4-20 mA sensor makes it a better choice for long-term battery operation.

The MS5837-30BA sensor provides numerous advantages, however, a major downside of this sensor is its ease of use. Compared to the 4-20 mA sensor which comes packaged in a waterproof housing and ready to be connected to the logger, the proposed improved sensor needs to be mounted to a PCB. Designing a data logger around this sensor sets higher requirements on design and requires considerably more hardware development in terms of PCB design and housing. The sensor is challenging to solder due to its compact size of 3,3 mm by 3,3 mm by 2,75 mm and strict solder procedure requirements although the small size makes it possible to be implemented in most designs. The MCU can communicate with the sensor over I2C and can be connected to the same I2C bus as the DS3231 clock chip. Getting the best accuracy of the sensor requires tight control and monitoring over the supply voltage since variations within the safe operating voltage range can alter the measurement accuracy.

Although the sensor seems to be suitable for this application based on its datasheet, thorough practical tests with varying pressures, temperatures, and voltages are essential to evaluate its performance. This includes performance points such as measuring noise, linearity over the complete pressure range, repeatability, pressure reading drift over time, and pressure, and temperature hysteresis. Collecting additional calibrated points across multiple pressures, temperatures, and voltage levels can potentially help to further improve accuracy. By closely monitoring the supply voltage and sampling the inbuilt temperature sensor of the sensor package, variations caused by these two factors can be adjusted for during postprocessing or, depending on the calculation complexity, onboard by the MCU.

6.3 Firmware-based energy optimization

Another way to gain energy efficiency is through firmware improvements to better utilize the features of the ESP32-S2. The other components of the prototype are not possible to optimize with firmware changes while powered on. All other components except for the power bank are powered off during deep sleep and otherwise when not needed. The possible efficiency improvements while awake have a minimal effect since the system is designed to stay in deep sleep for most of the time. This leads to a small window resulting in minimal efficiency gains. The possible power saving and management features supported by the ESP32-S2, the ESP-IDF framework, and RTOS not implemented in the prototype but worth exploring include Dynamic Frequency Scaling (DFS), automatic light sleep, the use of a deep-sleep wake stub, and GPIO isolation.

DFS manages the APB clock frequency from which the main CPU core is clocked. The RTOS runs by default at a tick frequency of 1000 Hz and, according to the documentation (Espressif Systems, n.d.), DFS is able to change the APB frequency multiple times every tick. By lowering the APB frequency whenever the main CPU is idle, the power consumption can be reduced. Automatic light sleep can be used in some cases to put the MCU in a lower power mode to reduce power consumption. The usage of this functionality depends on the firmware design of the application and how the hardware is utilized.

A deep sleep wake stub is the first piece of code that runs after the MCU wakes up from deep sleep before the SPI flash has been initiated. Compared to light sleep or other higher-level sleep modes, waking up from deep sleep takes more time because the memory has been powered down, and the MCU needs to access the firmware from flash memory to boot. Storing the wake stub code is limited to RTC fast memory and can only be used if the RTC fast memory is powered on during deep sleep. Data accessed by the wake stub can be stored in either RTC fast memory or RTC slow memory. Functions implemented in ROM memory are also accessible by the wake stub. After the wake stub has finished executing, it can decide if the boot process continues or if the MCU will go back to deep sleep. The advantage of the wake stub is its short execution delay after

waking up from deep sleep, however, it has several disadvantages and limitations in its functionality and capability compared to the main code in terms of maximum code size, hardware access, and memory access. This feature needs to be further explored to determine its usefulness in this application (Espressif Systems, n.d.).

Additional energy saving features while in deep sleep are limited, since depending on the deep sleep mode, most components are already powered down. One way to make sure externally connected components do not cause an increase in current flow through the internal pull-up and pull-down resistors of the MCU, is by using RTC GPIO pin isolation. The ESP-IDF framework has an included isolation function called rtc_gpio_isolate() that takes a GPIO pin number as parameter. This function can be run before the MCU is set to deep sleep to stop current leaking (Espressif Systems, n.d.).

Redesigning the logging process by better utilizing the RISC-V coprocessor can potentially improve energy efficiency. The tested prototype uses the ULP coprocessor only as a timer and completely relies on the main CPU to perform the rest of the logging process. This means that the main CPU needs to start every time to log data. This process can be changed by offloading a part of the process to the RISC-V coprocessor. It is capable of I2C communication with a sensor, although it has some limitations compared to the main CPU. It cannot perform all steps of the logging process since, according to its documentation (Espressif Systems, n.d.), it does not have access to non-volatile flash memory needed for long term storage of the logs in case of power failure. One approach is to use the coprocessor in a sensor monitor pattern where it reads the sensor at a specified interval. It then compares the pressure reading with earlier readings and, if it has changed more than a threshold, it wakes up the main CPU. This approach keeps the main CPU in deep sleep when the pressure does not change and only wakes it up when the water level is moving and should be logged. The main CPU needs to boot once to configure the coprocessor and move its code to RTC slow memory and then only wake up once the coprocessor detects a change in pressure.

6.4 Proposed system layout

In the current prototype design, the sensor is submerged, while the data logger remains above the water surface. Because the datalogger has wireless capability, it must be located somewhere near the top of the borehole with sufficient wireless signal strength, since reliable wireless communication from inside the well is challenging. The improved logger design concept integrates the logger hardware and sensor into one package capable of withstanding high pressure while submerged. This design would require manual retrieval of the logger from inside the borehole to offload pressure data.

To avoid removing the sensor from inside the borehole, a shielded data cable could be used to provide a data connection between the submersed logger and the surface. This would make data offloading easier but still take time and manual work. A solution to this would be to use a module with long-distance wireless communication capabilities connected to the wire and installed in a location with access to wireless networks. Wireless communication sets higher demands on energy storage requiring a bigger battery or external power from, for example, a photovoltaic solar module. If no wireless long-distance terrestrial networks are available, alternative methods need to be explored.

7 Future Development

After testing the developed prototype, it became evident that further improvements are necessary for it to meet performance requirements. As was proposed in the previous chapter, the sensor and its accompanying hardware would be integrated into a submergible package capable of withstanding high water pressure. The high deviating clock sources of the MCU could be made redundant with an improved external temperature-compensated real-time clock chip. In terms of performance, the shortcomings of the tested sensor would be significantly improved by integrating the proposed alternative sensor while also lowering the overall energy consumption of the system. Additionally, the power bank used as the power source would need to be changed for another design with necessary features including the ability to be powered off by the MCU in a safe manner.

To further enhance the safety and usability of the system, several other features should be considered. Expanding the integrated non-volatile flash memory would enable higher sampling frequencies before needing to be offloaded while adding a backup flash memory chip would ensure no data loss in case of chip failure or data corruption. The ULP Risc-V coprocessor of the ESP32-S2 can provide additional energy savings by performing lower complexity tasks while leaving the main CPU core to handle more demanding tasks like user interactions and wireless communication. Implementing over-the-air firmware updates would significantly improve maintainability and flexibility by allowing the firmware to be updated remotely.

7.1 Importance of accurate environmental variables

To calculate a water depth, or rather the height of the water column above a submerged pressure sensor, Equation 1 presented in the chapter Hydrostatic pressure can be used. From this equation can be concluded that the gravity acting on the water and its density are the two factors that can create an error, assuming the error resulting from the sensor

is excluded. This chapter will present and discuss the importance of measuring gravity, water temperature, salinity, and atmospheric pressure to minimize the error in accuracy resulting from changes in environmental variables.

The gravitational force changes depending on several factors. Gravity effects resulting from other celestial bodies are not considered including the effect from tides and aquifer deformation due to tides (Water Level Monitoring, n.d.). The factors presented are referenced to sea level. The uneven shape of the Earth gives it an uneven mass distribution. Since the mass of the Earth creates its gravity, where the mass is located, and its density determines one factor of the overall gravity. This factor creates a higher gravity on the equator compared to the north and south poles. The centrifugal effect from the Earth rotating on its axis is another factor to consider. This effect wants to push away an object located on the equator leading to a low effect on the poles and a higher effect on the equator. The last factor is the distance between the center of the Earth and an object on the surface. Gravity decreases with distance following the inverse square law. This creates a weakening effect on gravity on the equator since a point on the equator is farther away from the center of the Earth compared to a pole. Combining all factors gives an equation called the Geodetic Reference System. This equation, shown below, can be used to estimate the gravity of a location if its latitude angle is known in radians (University of Alberta, 2008).

$$g(\theta) = 9.78031846(1 + 0.0053024 * \sin^2(\theta) - 0.0000058 * \sin^2(2\theta)). \quad (13)$$

The effect gravity has on the water level reading at sea level can be calculated by substituting this equation for all angles between 0° and 90° into Equation 1 while keeping the pressure and density constant. Figure 28 shows the simulation results of using a fixed pressure of 10 bar, equivalent to approximately 100 meters of water, and a density of 1000 kgm^{-3} corresponding to fresh water at 5°C. The calculated difference in water level between the equator and the pole is 0,5391 meters. Since the water height depends on

the pressure divided by the product of density and gravity, a lower water density decreases the impact of gravity, and a higher density increases the impact of gravity. While this simulation shows that gravity has an influence, the variation is relatively small, and gravity is not expected to change significantly over time making it less necessary to continuously measure (University of Alberta, 2008).

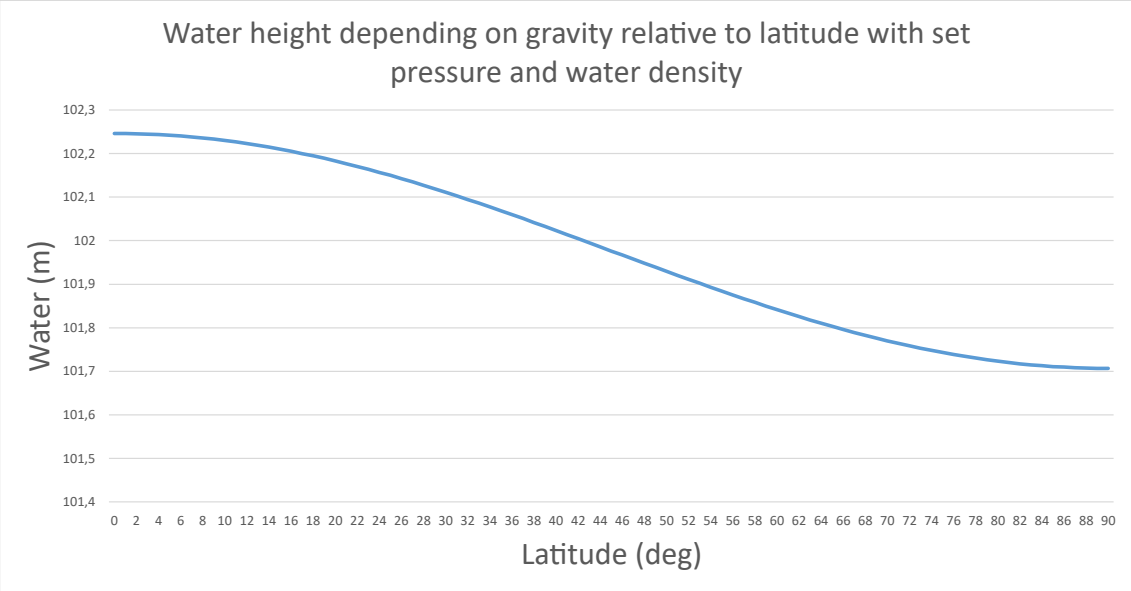

Figure 28. Water height above a submerged pressure sensor depending on gravity relative to latitude with a set pressure of 10 bar and water density of 1000 kgm^{-3}.

The second variable in the equation that can vary is the density of the water defined as mass per unit of volume. Water density is affected by changes in temperature and is generally denser at lower temperatures (Water Density, 2019). Another factor is foreign substances dissolved in the water. For example, seawater has a higher density than freshwater at the same temperature due to its salt content (National Oceanic and Atmospheric Administration, 2023). In the case of aquifers, groundwater can become contaminated from naturally occurring salt and other water-soluble minerals found in the subsurface geology close to a borehole. Additionally, seawater intrusion is another potential source of salt in groundwater if the aquifer lies near a coastline.

Two scenarios were simulated to evaluate the effect temperature and salt have on the density of water. First, the effect of temperature is tested in fresh water in the range of 0 to 30 °C. The simulation used a gravity of 9,81 m/s^2 and a constant pressure of 10 bar corresponding to a depth of approximately 100 meters. As shown in Figure 29, the water height corresponding to 10 bar of pressure decreases from 0 to 4°C, after which it starts to steadily increase. The change in water height in the tested temperature range between the lowest and highest is approximately 0,47 meters. The second test simulates a change in water density based on dissolved salt content in the range of 0 ‰ to 40 ‰. The simulation uses the same gravity and pressure values and a constant water temperature of 5 °C. From the graph in Figure 30 can be concluded that the change in water height for a specific pressure is linearly decreasing as the salt content increases. The difference in height between fresh water and a salt content of 35 ‰, typical for salty ocean water (Salinity and Conductivity, 2018), is approximately 2,747 meters.

These simulations show the importance of measuring both the temperature and salt concentration to decrease measurement errors from unknown factors. The temperature of the water inside the well can be measured with the integrated temperature sensor of the proposed improved pressure sensor, while the salt level needs an additional electrical conductivity meter integrated into the submersible sensor package. An important factor to consider is changes in salt concentration and temperature at different depths (Density, Temperature, and Salinity, n.d.). Layers with different densities can form in the water if it is stagnant. More dense water with a higher salt concentration sinks while warmer less salty water rises. Only using measurements from the sensor depth to calculate density can lead to wrong results and worse accuracy.

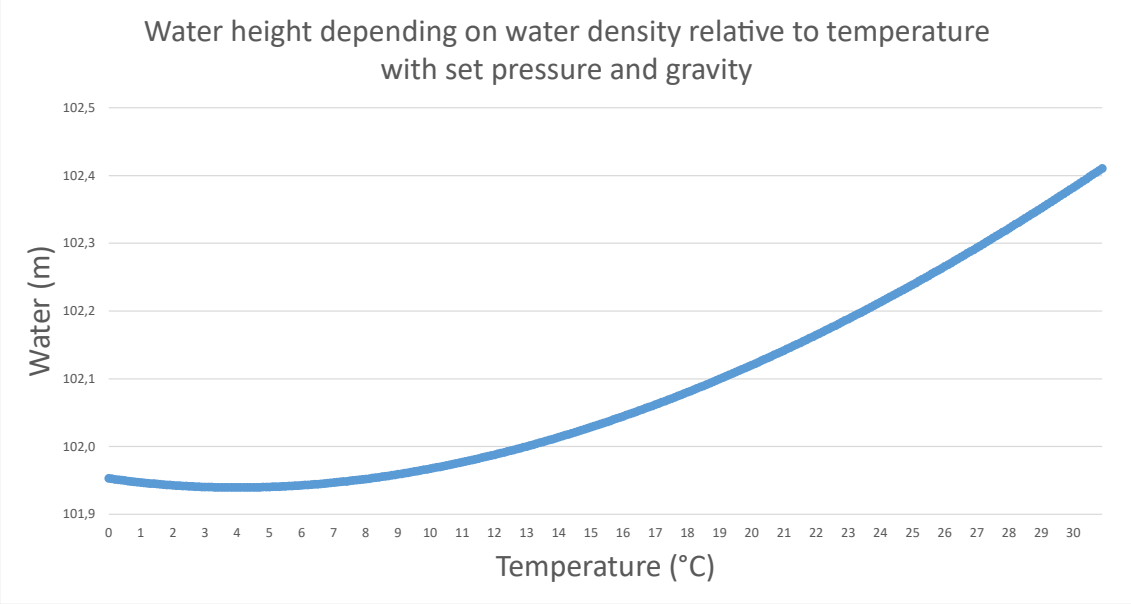

Figure 29. Water height above a pressure sensor submerged in freshwater depending on water density relative to temperature with a set pressure of 10 bar and gravity of 9,81 m/s^2. Water density relative to temperature adapted from Density of Water (n.d.).

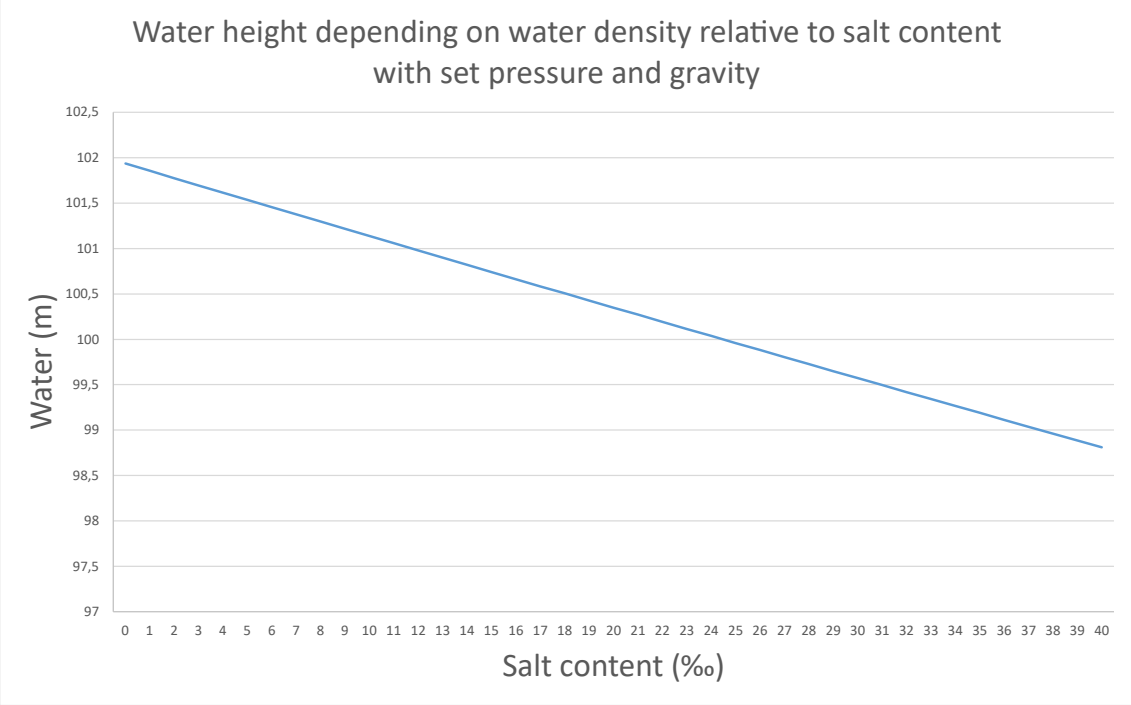

Figure 30. Water height above a submerged pressure sensor depending on water density relative to salt content at a pressure of 10 bar and gravity of 9,81 m/s^2. Water density relative to salt concentration adapted from Cospheric (n.d.).

Changes in atmospheric pressure are also important to consider. The prototype used a gauge-type pressure sensor with a vent tube enabling it to automatically compensate for changes in atmospheric pressure. On the other hand, the proposed upgraded sensor is an absolute-type pressure sensor making it mandatory to use an atmospheric pressure sensor. Regardless of the method used, it is necessary to know changes in atmospheric pressure since it can have a direct effect on the groundwater level according to Rasmussen and Crawford (1997). Generally, an increase in atmospheric pressure results in a decrease in groundwater level, however, the extent of the effect depends on several factors. The type of aquifer, its hydraulic permeability, and the type of unsaturated ground material between the aquifer and the surface can cause the groundwater level to respond differently. Additionally, if groundwater is measured through a borehole, its skin effect can cause the hydraulic permeability around the borehole to be lower than the surrounding aquifer producing a delayed response. To determine to what degree groundwater is affected, a dedicated barometric pressure sensor with suitable accuracy should

be integrated into the communications module proposed in the system layout. If the borehole opening is sealed and the module is located outside the borehole, additional testing is needed to assess how the seal affects air pressure transmission down to the water level.

7.2 Opportunity for machine learning integration

Machine learning (ML) models have traditionally been run on relatively powerful hardware to perform tasks such as object recognition from images or finding keywords in speech. This technology is now being adapted to run on microcontrollers with limited hardware and power constraints commonly known as TinyML (Beningo, 2022). Neural networks on microcontrollers have already seen some use cases such as object recognition in doorbell cameras and predicting the state of health of a system. The proposed improved system hardware and layout can benefit from integrating machine learning on three levels. First is the submerged level consisting of the pressure sensor and its accompanying logging hardware, the surface level with its communications module, and the offsite server level. This chapter will focus on the submerged and surface levels and discuss which benefits TinyML can provide.

The submerged datalogger has a limited battery capacity giving it high constraints on energy usage. A major priority is to decrease energy consumption to have the longest battery operation time possible. Using a machine learning algorithm is unfavorable since it requires additional awake time and code execution. An ML model running on the submerged level would need to be fast and lightweight requiring minimal calculations to be able to provide an overall lower energy consumption compared to using another approach, for example, simple logic. One potential use case where ML could be beneficial is adaptive frequency. In the case of a borehole equipped with a solar-powered pump, activity is expected to be higher during the day compared to nighttime. An ML model could in this case dynamically adjust the sampling interval by sampling more frequently during times of high activity than low activity or vice versa depending on the use case.

This could potentially reduce the overall energy usage, however, a simple If-Else logic algorithm can provide some of the same functionality with lower complexity and overhead.

The main objective for the surface-level communications module is to have a wireless link to an offsite server. It is therefore expected to be powered by a bigger battery than the submerged level since wireless communications require more energy. It can potentially be powered by external solar power if located in a suitable place. This makes the communications module less energy-constrained and able to do more data processing. Wireless communication in remote regions in Africa can be low bandwidth, expensive, or unreliable, and if satellite communication is used, data transmission may be limited to a timeframe when the satellite is in view. If wireless communication is limited, sending large amounts of data should be avoided and only sending important data for example, system health status or unexpected groundwater level behavior. Pereira et al. (2023) demonstrated promising results by running a two-parameter Weibull distribution function, an unsupervised statistical model, on a Raspberry Pi microcontroller to detect anomalies in time series accelerometer data generated by fan vibrations at different speeds. The algorithm uses parameters calculated from computer simulations and parameters that are continuously in real-time learning changes in the environment. This approach could potentially be adapted for detecting groundwater and system health anomalies by training on data from normal operations.

8 Conclusion

The goal of this thesis was to explore methods and commercially available groundwater monitoring solutions suitable for Läkarmissionen International's applications in Africa. An appropriate groundwater measuring system must be affordable and autonomous over long periods and capable of measuring at short intervals. Additionally, it needs to be capable of handling obstructions found in active boreholes while having a small form factor to fit inside a borehole. Based on the research, sound, and water pressure-based measuring methods were identified as suitable among the researched methods for active boreholes.

A prototype groundwater level logger was developed as a proof-of-concept for an affordable alternative to commercially available solutions. The prototype used an affordable water pressure sensor and other off-the-shelf components for fast development. Firmware was developed for a microcontroller to handle the logging process, user communication, and other features. After testing the prototype and analyzing its performance, it was determined that it had potential but was lacking in some areas.

To make the system comply with the requirements set by LMI, the following improvements were proposed. The water depth measurement accuracy and maximum measurable pressure can be significantly improved with an alternative pressure sensor that, according to its datasheet, does comply with the requirements in terms of pressure range and accuracy. It has additionally lower energy consumption and faster readout time than the developed prototype resulting in potentially a significant reduction in overall system energy consumption. The frequency of the available clock sources on the microcontroller proved to deviate outside the acceptable range requiring frequent synchronization from an outside source. This time deviation can be significantly reduced and kept within an acceptable error range by using a more accurate clock signal from an external clock chip, for example, the DS3231 RTC chip. The high energy consumption of the prototype resulting in a short Battery life can be extended partly through an improved battery solution

with a higher capacity cell and a custom BMS, and partly through a lower energy consumption provided by the hardware improvements.

Additionally, a new alternative system layout was proposed with a submerged level and a surface level consisting of a communications module close to the borehole. This module would house the hardware for long-distance wireless communication and a wired connection to the submerged sensor. It would additionally house a barometric pressure sensor that is necessary for compensating for atmospheric pressure changes when using the proposed absolute-type pressure sensor. While the proposed system layout would require a prototype redesign, the suggested improvements would result in a capable groundwater level measuring solution suitable for LMI at an affordable price compared to commercial solutions. These improvements, however, have not been tested in practice and would need to be tested in a real-world scenario to know their performance.

References

Acconeer. (2025, April 28). *System overview*. Retrieved August 12, 2024, from https://docs.acconeer.com/en/latest/a111/radar_data_and_control/system_overview.html#fig-transmit-signal-length

Adafruit Industries. (n.d.). *Adafruit QT Py ESP32-S2 WiFi Dev Board with STEMMA QT*. Retrieved January 17, 2025, from https://www.adafruit.com/product/5325#technical-details

AdaFruit Learning System. (n.d.). Retrieved January 17, 2025, from https://learn.adafruit.com/assets/107493

AFRICAN DEVELOPMENT BANK GROUP, Boucher, C., & United Nations. (2005). *The Rural Water Supply and Sanitation Initiative*.

Ajmeri, S., Hagarty, D., & Tanwar, A. (2022, January 8). *Clocks, time error, and noise > Clocks | Cisco Press*. Retrieved October 5, 2024, from https://www.ciscopress.com/articles/article.asp?p=3128857

Avnet Abacus. (n.d.). *Absolute vs Gauge vs Differential Pressure Sensors | The Design Engineer's Guide | Avnet Abacus*. Retrieved January 9, 2025, from https://my.avnet.com/abacus/solutions/technologies/sensors/pressure-sensors/measurement-types/

AZ-Delivery. (n.d.). *Real Time Clock RTC DS3231 I2C real-time clock compatible with Arduino and Raspberry Pi*. Retrieved March 4, 2025, from https://www.az-delivery.de/en/products/ds3231-real-time-clock

Behera, B., Borole, U. P., Gawade, T. C., Khan, J., Barshilia, H. C., & Chowdhury, P. (2022). Design and development of submersible hydrostatic level sensor using a GMR sensor. *Measurement*, *206*, 112310. https://doi.org/10.1016/j.measurement.2022.112310

Beningo, J. (2022, August 18). *How to Run a "Hello World" Machine Learning Model on STM32 Microcontrollers*. DigiKey. Retrieved April 10, 2025, from https://www.digikey.com/en/articles/how-to-run-a-hello-world-machine-learning-model-stm32

Bentley, L. R., & Trenholm, N. M. (2002). The Accuracy of Water Table Elevation Estimates Determined from Ground Penetrating Radar Data. *Journal of Environmental and Engineering Geophysics*, *7*(1), 37–53. https://doi.org/10.4133/jeeg7.1.37

British Geological Survey. (2011). *Download digital groundwater maps of Africa*. Retrieved June 24, 2024, from https://www2.bgs.ac.uk/groundwater/international/africanGroundwater/mapsDownload.html

Catsamas, S., Shi, B., Wang, M., Xiao, J., Kolotelo, P., & McCarthy, D. (2023). A Low-Cost Radar-Based IoT sensor for noncontact measurements of water surface velocity and depth. *Sensors*, *23*(14), 6314. https://doi.org/10.3390/s23146314

Chen, X., Xu, Z., & Shang, L. (2023). Satellite Internet of Things: challenges, solutions, and development trends. *Frontiers of Information Technology & Electronic Engineering*, *24*(7), 935–944. https://doi.org/10.1631/fitee.2200648

Climatic Research Unit. (n.d.). *Climate Change Knowledge Portal*. World Bank Group. Retrieved June 24, 2024, from https://climateknowledgeportal.worldbank.org/country/chad/climate-data-historical

Cospheric. (n.d.). *Density*. Cospheric - PRECISION SPHERICAL PARTICLES GLOBALLY. Retrieved April 9, 2025, from https://www.cospheric.com/density_specific_gravity_of_water_and_other_liquids.htm

Cunningham, W. L., & Schalk, C. W. (2011). Groundwater technical procedures of the U.S. Geological Survey. *Techniques and Methods*, GWPD 13. https://doi.org/10.3133/tm1a1

Density of Water. (n.d.). Retrieved April 7, 2025, from https://www.csus.edu/indiv/m/mackj/chem1a/docs/h2oden.pdf

Density, Temperature, and salinity. (n.d.). Exploring Our Fluid Earth. Retrieved April 3, 2025, from https://manoa.hawaii.edu/exploringourfluidearth/physical/density-effects/density-temperature-and-salinity

Eno Scientific. (2024). *Well Watch 670*. Retrieved August 8, 2024, from https://enoscientific.com/well-watch-670/

Espressif Systems. (n.d.). *System API*. ESP-IDF Programming Guide v5.4.1 Documentation. Retrieved October 12, 2024, from https://docs.espressif.com/projects/esp-idf/en/stable/esp32s2/api-reference/system/index.html

Espressif Systems. (2020). *ESP32-S2 Technical Reference Manual* (Preliminary V0.7). https://www.mouser.com/pdfDocs/esp32-s2_technical_reference_manual_en-2.pdf

Espressif Systems. (2024). *ESP32-S2 Series datasheet*. https://www.espressif.com/sites/default/files/documentation/esp32-s2_datasheet_en.pdf

Espressif Systems. (2025). *ESP32-S2 Technical Reference Manual* (Version 1.2). https://www.espressif.com/sites/default/files/documentation/esp32-s2_technical_reference_manual_en.pdf

Farr, J. L., GUMIREMHETE, R., DAVIES, J., ROBINS, N. S., Wellfield Consulting Services (Pty) Ltd, & British Geological Survey. (2005). GROUNDWATER DEPENDENCE AND DROUGHT WITHIN THE SOUTHERN AFRICAN DEVELOPMENT COMMUNITY. In *World Bank GEF Programme*. World Bank.

FreeRTOS. (n.d.). *FreeRTOS*. Retrieved February 16, 2025, from https://www.freertos.org/

Fulton, A., Dudley, T., Staton, K., UC Cooperative Extension, & California Department of Water Resources. (n.d.). GROUNDWATER LEVEL MONITORING: WHAT IS IT? HOW IS IT DONE? WHY DO IT? *GROUNDWATER LEVEL MONITORING*.

Garber, M. S., & Koopman, F. C. (1968). *METHODS OF MEASURING WATER LEVELS IN DEEP WELLS* (Vol. 8). UNITED STATES GOVERNMENT PRINTING OFFICE.

García-López, S., Vélez-Nicolás, M., Zarandona-Palacio, P., Curcio, A. C., Ruiz-Ortiz, V., & Barbero, L. (2022). UAV-borne LiDAR revolutionizing groundwater level mapping. *The Science of the Total Environment*, *859*, 160272. https://doi.org/10.1016/j.scitotenv.2022.160272

GeeksforGeeks. (2025, May 3). *Least Square Method | Definition graph and formula*. Retrieved May 4, 2025, from https://www.geeksforgeeks.org/least-square-method/

Hasan, K., Tom, N., & Yuce, M. R. (2023). Navigating Battery Choices in IoT: An extensive survey of technologies and their applications. *Batteries*, *9*(12), 580. https://doi.org/10.3390/batteries9120580

Instrumentation Tools. (2019, November 13). *Guided Wave Radar – Principle, Limitations, Installation*. Inst Tools. Retrieved August 12, 2024, from https://instrumentationtools.com/guided-wave-radar/

Iovescu, C., & Rao, S. (2020). *The fundamentals of millimeter wave radar sensors*. https://www.ti.com/lit/wp/spyy005a/spyy005a.pdf?ts=1711441433330&ref_url=https%253A%252F%252Fduckduckgo.com%252F

IUOE National Training Fund. (2018). *Protecting Ourselves from Mold: awareness training for operating engineers*. https://www.thelippygroup.com

Keeland, B. D., Dowd, J. F., Hardegree, W. S., USGS, National Wetlands Research Center, Department of Geology, University of Georgia, & U.S. Environmental Protection Agency. (1997). Use of inexpensive pressure transducers for measuring water levels in wells. In *Wetlands Ecology and Management* (Vol. 5, pp. 121–129). Kluwer Academic Publishers.

Klements, M. (2020, August 17). *How long can an arduino run on batteries? I tested 6 of the most common boards*. The DIY Life. Retrieved December 4, 2024, from https://www.the-diy-life.com/how-long-can-an-arduino-run-on-batteries-i-tested-6-of-the-most-common-boards/

Kłonowski, M., Brytan, P., Krzonkalla, A., Ładocha, A., Maćko, A., Midttømme, K., Nermoen, A., Pratkowiecka, W., Sergiel, W., Thomas, P., Wyrwalska, U., Zawistowski, K., & Żuk, P. (2024). Temperature Logging in Selected Geological Boreholes: DTS vs. Manual Profiling. Preliminary Results of the OptiSGE Project. In *EAGE GeoTech 2024* (pp. 1–5). https://doi.org/10.3997/2214-4609.202430035

LM International. (2024). Retrieved July 22, 2024, from https://www.lminternational.org/

Lonberg, F. (n.d.). Class 37. In *Class 37* (pp. 1–6). https://faculty.uml.edu/flonberg/95.103/Class37_09.pdf

m5-docs. (n.d.). Retrieved August 17, 2024, from https://docs.m5stack.com/en/unit/AIN4-20mA%20Unit

Ma'arif, A., Iswanto, I., Nuryono, A. A., & Alfian, R. I. (2019). Kalman filter for noise reducer on sensor readings. *Signal and Image Processing Letters*, *1*(2), 11–22. https://doi.org/10.31763/simple.v1i2.2

Mahler, M., Schmidberger, F., & Fuchs, J. (2022). Evaluation of a sound tube supported acoustic Sensor behavior in an agriculture machinery context. In *NAFEMS DACH Regionalkonferenz*. NAFEMS DACH Regionalkonferenz. https://www.researchgate.net/publication/374921331

Mahmoudzadeh, Francés, A., Lubczynski, M., & Lambot, S. (2012). Using ground penetrating radar to investigate the water table depth in weathered granites — Sardon case study, Spain. *Journal of Applied Geophysics*, *79*, 17–26. https://doi.org/10.1016/j.jappgeo.2011.12.009

Makerfabs. (2022). *Cautions in using ESP32 ADC - MakerFabs*. Quick PCB Assembly, IOT Solution, Open Hardware - Makerfabs. Retrieved September 18, 2024, from https://www.makerfabs.cc/article/cautions-in-using-esp32-adc.html

Massa, D. P. (n.d.-a). *Avoiding False Echoes from Fixed Structures When Using Ultrasonic Sensors for Water Level Measurement*. https://www.massa.com/wp-content/uploads/2018/06/Massa-Whitepaper-2-DPM-160512.pdf

Massa, D. P. (n.d.-b). Understanding how frequency, beam patterns of transducers, and reflection characteristics of targets affect the performance of ultrasonic sensors. In *Massa Products Corporation*.

MATLAB. (2023, April 18). *Pulse waveform basics: Visualizing radar performance with the ambiguity function* [Video]. YouTube. https://www.youtube.com/watch?v=UCir6mOW2OU

National Academies. (2022, May 24). *Can we measure water underground from space?* National Academies. Retrieved May 29, 2024, from https://www.nationalacademies.org/based-on-science/can-we-measure-water-underground-from-space

National Oceanic and Atmospheric Administration. (2023, March 28). *Sea water*. Retrieved April 28, 2025, from https://www.noaa.gov/jetstream/ocean/sea-water

Oklahoma State University. (2001, May 13). *Vapor pressure of water*. Retrieved October 8, 2024, from https://intro.chem.okstate.edu/1515SP01/Database/VPWater.html

Olesen, B. W., Parsons, K. C., Wirsbo-VELTA GmbH, & Loughborough University. (2002). Introduction to thermal comfort standards and to the proposed new version of EN ISO 7730. In *Energy and Buildings* (Vol. 34, pp. 537–548). Elsevier Science B.V. https://doi.org/10.1016/S0378-7788(02)00004-X

Pande, A., *, Int-Hout, D., III, Paliaga, G., Heinzerling, D., *, Khalil, E., *, Schoen, L. J., *, Alspach, P. F., Khovalyg, D., Simmonds, P., Arens, E. A., Koupriyanov, M., Smith, A. R., Bean, R., Li, B., Mora, R., Williams, J. G., *, Offermann, F. J., Crawley, D. B., Katipamula, S., . . . American National Standards Institute. (2020). ANSI/ASHRAE Addendum d to ANSI/ASHRAE Standard 55-2017. In *Thermal Environmental Conditions for Human Occupancy*. https://www.ashrae.org/file%20library/technical%20resources/standards%20and%20guidelines/standards%20addenda/55_2017_d_20200731.pdf

Paul, J. D., Buytaert, W., & Sah, N. (2020). A technical evaluation of LIDAR-Based measurement of river water levels. *Water Resources Research*, *56*(4). https://doi.org/10.1029/2019wr026810

Pereira, E. S., Marcondes, L. S., & Silva, J. M. (2023). On-Device tiny machine learning for anomaly detection based on the extreme values theory. *IEEE Micro*, *43*(6), 58–65. https://doi.org/10.1109/mm.2023.3316918

Purivigraipong, S., Jantarang, S., & Mahanakorn University of Technology. (2020). Satellite-Enabled IoT, Low-Earth Net Centric and its applications for Thailand. In *ENGINEERING TRANSACTIONS* (Vols. 23–1, pp. 67–68) [Journal-article].

Qiu, Z., Lu, Y., & Qiu, Z. (2022). Review of ultrasonic ranging methods and their current challenges. *Micromachines*, *13*(4), 520. https://doi.org/10.3390/mi13040520

Rasmussen, T. C., & Crawford, L. A. (1997). Identifying and removing barometric pressure effects in confined and unconfined aquifers. *Ground Water*, *35*(3), 502–511. https://doi.org/10.1111/j.1745-6584.1997.tb00111.x

RayMing. (2023, October 16). *How does a water pressure sensor work - RAYPCB*. RayPCB. Retrieved January 22, 2025, from https://www.raypcb.com/water-pressure-sensor/

Robins, N. S., Fergusson, J., British Geological Survey, & Springer. (2014). Groundwater scarcity and conflict – managing hotspots. In *Earth Perspectives*. http://www.earth-perspectives.com/1/1/6

Salih, S. A., Dhahir, M. N., & Ahmed, M. M. (2022). Detection of groundwater table by using ground penetrating radar in two selected Sites/Northern Iraq. *IOP Conference Series Earth and Environmental Science*, *1120*(1), 012026. https://doi.org/10.1088/1755-1315/1120/1/012026

Salinity and conductivity. (2018, December 14). VIRTUE. Retrieved March 21, 2025, from https://virtue.gmbl.se/sv/svenskt-innehall/salinity-and-conductivity

Schindel, D., Bashford, A., & Hutchins, D. (1997). Focussing of ultrasonic waves in air using a micromachined Fresnel zone-plate. *Ultrasonics*, *35*(4), 275–285. https://doi.org/10.1016/s0041-624x(97)00011-5

Sercu, J. (2023, March 7). *Effortless groundwater level monitoring*. Crodeon. Retrieved December 28, 2024, from https://www.crodeon.com/blogs/news/groundwater-level-monitoring

Shah, M. M. (2021). Evaluation of methods for prediction of evaporation from water pools. *Journal of Building Physics*, *45*(5), 629–648. https://doi.org/10.1177/17442591211034193

Smoot, B. J. (2021, April 1). *The Basics of ultrasonic sensors | Same Sky*. Same Sky. Retrieved July 26, 2024, from https://www.sameskydevices.com/blog/the-basics-of-ultrasonic-sensors

Solinst Canada Ltd. (2024). *Sonic water level meter*. Retrieved July 20, 2024, from https://www.solinst.com/products/level-measurement-devices/104-sonic-water-level-meter/104.php

Stumm, F., & Como, M. (2017). Delineation of Salt Water Intrusion through Use of Electromagnetic-Induction Logging: A Case Study in Southern Manhattan Island, New York. *Water*, *9*(9), 631. https://doi.org/10.3390/w9090631

Suganya, R., Joseph, L. L., & Kollem, S. (2024). Understanding Lithium-Ion Battery Management Systems in Electric Vehicles: Environmental and Health Impacts, Comparative Study, and Future Trends: A review. *Results in Engineering*, 103047. https://doi.org/10.1016/j.rineng.2024.103047

Taylor, C. J., & Alley, W. M. (2001). *Ground-Water-Level monitoring and the importance of Long-Term Water-Level data*. U.S. Geological Survey.

TE Connectivity Sensors. (2019). *MS5837-30BA26*. https://www.mouser.fr/datasheet/2/418/6/ENG_DS_MS5837_30BA_C2-1130109.pdf

Teguh, R., & Usup, H. (2021). Realtime monitoring for groundwater level and local climate based on universal communication system. *Computer Science and Information Technologies*, *2*(2), 67–76. https://doi.org/10.11591/csit.v2i2.p67-76

Toa, M., & Whitehead, A. (2021). Ultrasonic Sensing Basics. In *Application Note*. https://www.ti.com/lit/an/slaa907d/slaa907d.pdf?ts=1710319010247#:~:text=Ultrasonic%20sensors%20can%20measure%20distance,centimeters%20up%20to%20several%20meters.

Trimmer, W. L. (2000). *Measuring well water levels*. https://extension.oregonstate.edu/sites/extd8/files/documents/ec1368.pdf

Understanding Pressure Sensor Specifications. (n.d.). AV Sensors. Retrieved February 7, 2025, from https://www.avsensors.com/understanding-pressure-sensor-specifications

University of Alberta. (2008, September). Variation of gravity with latitude and elevation. In *Geophysics*. https://sites.ualberta.ca/~unsworth/UA-classes/210/notes210/B/210B3-2008.pdf

U.S. Geological Survey. (n.d.). *GWPD 13—Measuring water levels by use of an air line* [Report].

Van Essen Instruments. (2023). *TD-Diver & Baro-Diver® DI8XX Series*. https://www.vanessen.com/wp-content/uploads/2024/01/TD-Diver-DI8xx-ProductManual-en-2024.pdf

Van Essen Instruments. (2024, December 11). *Van Essen - Water Level Data Logging & Monitoring Products*. Retrieved July 4, 2024, from https://www.vanessen.com/

Wang, D., Lin, P., Chen, Z., Fei, C., Qiu, Z., Chen, Q., Sun, X., Wu, Y., & Sun, L. (2021). Evolvable Acoustic Field Generated by a Transducer with 3D-Printed Fresnel Lens. *Micromachines*, *12*(11), 1315. https://doi.org/10.3390/mi12111315

Water density. (2019, October 22). USGS. Retrieved February 24, 2025, from https://www.usgs.gov/special-topics/water-science-school/science/water-density

Water level monitoring. (n.d.). Upper Loup Natural Resources District. Retrieved August 1, 2024, from https://www.upperloupnrd.org/water-level-monitoring

Water Resources Mission Area. (2019, March 2). *Saltwater intrusion*. USGS. Retrieved May 29, 2024, from https://www.usgs.gov/mission-areas/water-resources/science/saltwater-intrusion#overview

Wilopo, W., Rachman, M. H., & Putra, D. P. E. (2020). Assessment of groundwater resources potential using geoelectrical method and slug test in Tegal District, Central Java Province, Indonesia. *E3S Web of Conferences*, *200*, 02003. https://doi.org/10.1051/e3sconf/202020002003

Zheng, H., Wu, H., Wang, Y., Shen, X., Fang, Z., Huang, D., Dash, J. N., Htein, L., Cheng, X., Tam, H., Ding, X., & Lu, C. (2024). High-sensitivity distributed optical fiber sensor for simultaneous hydrostatic pressure and temperature measurement based on birefringent frequency-scanning ϕ-OTDR. *Optics & Laser Technology*, *175*, 110756. https://doi.org/10.1016/j.optlastec.2024.110756

Appendices

Appendix 1. Kalman filter implemented in python

```
# Modified version of Kalman filter implemented in python
# based on the design and code example
# presented by Ma'arif et al. (2019)

import numpy as np
import matplotlib.pyplot as plt

# Variance and measurement constants
Q = 0.1
R = 10

# Initial estimate
x_prev = (measurements[0]
          + measurements[1] + measurements[2]) / 3
# Initial uncertainty
p_prev = 1

# Arrays to store results
filtered_estimates = []
for m in measurements:
    # Prediction
    x_update = x_prev
    p_update = p_prev + Q

    # Update
    K = p_update / (p_update + R)
    x_prev = x_update + K * (m - x_update)
    p_prev = (1 - K) * p_update

    # Results list
    filtered_estimates.append(x_prev)
```