PEOPLE DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
YAHIA FARES UNIVERSITY OF MEDEA



Faculty of **T**echnology

Department of **E**lectrical **E**ngineering

Master Thesis

Sector: Electronics

Specialty: Electronics of Embedded Systems

DESIGN AND REALIZATION OF A SMART WATER METER

Presented by:

Mr. BENGHERBIA Hocine

Ms. MANSEUR Serine

Proposed and Directed by:

Dr. KARA Reda

Mr. ELMOHRI Mohamed Abdessalam

Promotion 2022/2023

Acknowledgement

We would like to thank our dear teacher and supervisor Dr. Kara as well as our co-supervisor Dr.El mohri for supporting us throughout this tough year and kept motivating us and pushing us to give it all we got and to give the best we could. Thank you to the jury members for giving us the honor of judging our work .We also sincerely thank all the Electronics engineering department teachers. Finally, we thank our families , and friends

Dedication

We dedicate this project thesis to the Almighty God above. Finishing this manuscript is a great opportunity to express our whole-hearted thanks and many praises unto him.

We dedicate this project thesis to our families and many friends. A special feeling of gratitude to our loving parents, who tirelessly encouraged and supported us , and to all our siblings who have never left our sides.

Last but not least we dedicate this whole work to ourselves, it was a long hard year full of stress but at the end we made it

RESUME

ملخص

كان قياس استهلاك المياه في كل منزل أمرًا صعبًا للغاية ، وذلك بإرسال رجل لجمع معلومات حول كل منزل وإبلاغ الشركة بها ، وبعد سنوات قليلة تم اختراع عداد المياه الميكانيكي ، وحل مشكلة المسافة ، فهو يقيس معدل التدفق من خلال دوران المكره ولكن مرة أخرى لم يعد عداد المياه الميكانيكي في حالة مثالية لأنه غير قادر على تسجيل أي بيانات بدقة أو الاحتفاظ بها. حيث تعمل التكنولوجيا على تحسين هذا الجانب ولم يتم نسيانه ولهذا تم العثور على فكرة العمل على عداد مياه إلكتروني ، وهذا لاحقًا يحدد معدل التدفق باستخدام مصدر الطاقة ، حيث يمر الماء عبر الأنبوب يولد نبضات من خلال دوران العجلة ومن ثم نحدد التردد الذي يؤدي إلى حساب معدل التدفق وبالتالي الحجم ، بمساعدة STM32 MCU(لاستهلاك منخفض للطاقة) توصيلها بوحدة LoRa(لمسافة أكبر) التي ترسل البيانات التي تم الحصول عليها إلى بوابة ، بعد توصيلها بخادم MQTT ، سيتم تكديس البيانات التي تم جمعها في قاعدة البيانات التي ستظهر على الواجهة.

كلمات مفتاحية :وحدة متحكم دقيق(MCU),بعيد المدى (LoRa),النقل عن بعد لخدمة وضع الرسائل في قائمة انتظار (MQTT)

Résumé

Mesurer la consommation d'eau dans chaque maison était très difficile et c'est en envoyant un homme rassembler les informations de chaque maison et les rapporter à l'entreprise, quelques années plus tard, le compteur d'eau mécanique a été inventé, il a résolu le problème de distance, il mesure le débit à travers le rotation de la turbine mais encore Le compteur d'eau mécanique n'est plus dans un état optimal car il est incapable d'enregistrer ou de conserver avec précision les données. , au fur et à mesure que la technologie s'améliore, ce côté n'a pas été oublié et c'est pourquoi l'idée de travailler sur un compteur d'eau électronique a été trouvée. la roue et à partir de là on détermine la fréquence qui conduit à calculer le débit donc le volume , à l'aide d'un microcontrôleur STM32 (pour une faible consommation électrique) en le connectant à un module LoRa (pour une plus grande distance) qui envoie les données obtenues à une passerelle, après l'avoir connectée à un serveur MQTT, les données collectées seront empilées dans la base de données qui sera apparue sur une interface.

Mots clés : Unité de microcontrôleur (MCU),Longue portée (LoRa),Transport de télémétrie Message Queuing (MQTT)

Abstract

Measuring water consumption in each house was very difficult and that's by sending a man to gather around each house's information and report it to the company , few years later the mechanical water meter was invented it solved the distance problem , it measures the flow rate through the impeller rotation but again The mechanical water meter is no longer in optimal condition as it is unable to accurately record or retain any data. , as the technology improves this side was not forgotten and that is why an idea of working on an electronic water meter was found .This later determines the flow rate using power supply , as the water passes through the pipe it generates pulses through the rotation of the wheel and from that we determain the frequency which leads to calculating the flow rate thus the volume , with the help of STM32 MCU (for low power consumption) connecting it to a LoRa module (for a bigger distance) which sends the data obtained to a gateway , after connecting it to an MQTT server , the data collected will be stacked in the database that will be appeared on an interface.

Key Words:Microcontroller Unit (MCU),Long Range (LoRa),Message Queuing Telemetry Transport (MQTT)

Table of contents

ABREVIATIONS: .....
Figures list: .....
General Introduction 1
CHAPTER 1 : Water – Meters 3
Introduction: 3
I.1 Water-meter : 3
I.2 The Evolution of Water- Metering : 4
I.2.1 Manual Water- Meter Reading : 4
I.2.2 The Postcard Method of Water- Meter Reading : 4
I.2.3 Walk-By Water-Metering : 4
I.2.4 AMR and Drive-By Metering : 4
I.2.5 Advanced Metering Infrastructure (AMI) : 5
I.3 Benefits of using Water-Meter : 5
I.3.1 Water-meters Flaws : 6
I.4 Ways of reading a Water-Meter : 6
I.5 Types of Water-Meters : 8
I.5.1 Positive Displacement Water- Meters : 8
I.5.2 Velocity water meters : 9
I.5.3 Jet water meters : 10
I.5.4 Electromagnetic Water-Meters : 11
I.5.5 Ultrasonic Water-Meter : 12
I.5.6 Turbine Water-Meters : 13
CHAPTER 2 : ARCHITECTURE OF A WATER METE 14
Introduction: 14
II.1 Water flow meter : 14
II.1.1 Water flow sensor Features : 14
II.1.2 Water flow meter : 14
II.1.2.1 YF-S201 Specifications : 15
II.1.3 Concept of Water flow meter : 15
II.1.4 Water generator : 16
II.1.5 Water flow detector Description: 17
II.2 Definition of LoRa: 17
II.2.1 LoRa’s potential: 17

II.2.2 LoRa ‘s Features : 18
II.2.3 LoRa Modulation : 18
II.2.4 Definition of NRF : 19
II.2.4.1 NRF’s Features : 19
II.2.4.2 Difference between LoRa and NRF : 20
II.2.5 LoRa Gateway : 20
II.2.6 STM32 : 20
II.2.6.1 STM32F103C8T6 : 21
II.2.6.2 STM32 PINOUT : 23
II.2.6.3 STM32 Factor : 24
II.2.6.4 STM32 Applications : 24
II.2.6.5 STM32CubeIDE : 24
II.2.7 Esp8266 : 27
II.2.7.1 WEMOSD1 MINI ESP8266 WIFI DEVELOPMENT BOARD: 27
II.2.7.2 Esp8266 Fonctions : 27
II.2.7.3 Esp8266 Applications : 28
II.2.8 Getting to know MQTT : 28
II.2.8.1 Concept of MQTT : 28
II.2.8.2 Concept of HTTP: 29
II.2.8.3 Understanding the difference between MQTT , HTTP : 29
II.2.8.4 An MQTT Broker : 30
II.2.8.5 Benefits of using MQTT : 30
II.2.8.6MQTT in Iot: 30
II.2.9 Definition of HiveMQ : 31
II.2.10 Definition of Node-Red: 32
II.2.10.1 Node-Red Features : 33
II.2.10.2 Benefits of using Node-Red: 33
II.2.10.3 Definition of IoT: 33
II.2.10.4 Understanding the bond between Node-Red and IoT: 34
II.2.11 Database: 35
II.2.12 Structured Query Language (SQL): 36
CHAPITRE 3 : EXPERIMENTS AND TESTS 37
Introduction : 37
III.1 Water-Meter Functional Diagram : 37

III.2 Consumption measurement: 38
III.3 Volume Code : 39
III.4LoRa configuration : 40
III.5 Sleep Mode: 41
III.6 Power control circuit: 42
III.7 Water flow meter deactivation circuit: 42
III.8 Battery's power voltage circuit : 43
III.9 Water flow detector circuit : 43
III.10 MCU STM32 and LoRa module : 44
III.11 Valve control circuit: 44
III.12 Full System Function: 44
III.13 System power usage calculation : 47
III.14LoRa Gateway: 47
Conclusion : 51
GENERAL CONCLUSION : 52

ABREVIATIONS:

LoRa: Long Range area

MCU: Micro controller Unit

MQTT: Message Queuing Telemetry Transport

WFM: Water flow meter

WFD: Water flow detector

NSS: Negative Slave Select

MISO: Master In/ Slave Out

MOSI: Master Out/Slave In

SCK: Serial Clock

CWFM: Control water flow meter

CWFD: Control water flow detector

CCR: Capture/Compare Register

TIM : timer

PCB : Printed Circuit Board

Figures list:

FIGURE I.1: WATER METER ---- 3
FIGURE I.2: ANALOG WATER-METER ---- 7
FIGURE I.3: DIGITAL WATER-METER ---- 8
FIGURE I.4: POSITIVE DISPLACEMENT WATER-METER ---- 9
FIGURE I.5: JET WATER-METER ---- 10
FIGURE I.6: ELECTROMAGNETIC WATER-METER ---- 11
FIGURE I.7: ULTRASONIC WATER-METER ---- 12
FIGURE I.8: TURBINE WATER-METER ---- 13
FIGURE II.9: YF-S201 ---- 15
FIGURE II.10: ARCHITECTURE OF WATER FLOW SENSOR ---- 16
FIGURE II.11: F50 -5V GENERATO ---- 16
FIGURE II.12: WATER FLOW DETECTOR ---- 17
FIGURE II.13: LORA MODULE ---- 18
FIGURE II.14: CHIRP MODULATION ---- 19
FIGURE II.15: NRF MODULE ---- 19
FIGURE II.16: STM32 ---- 21
FIGURE II.17: STM32 MICROCONTROLLER ---- 23
FIGURE II.18: STM32 PINOUT ---- 24
FIGURE II.19: STM32CUBEIDE PROJECT ---- 25
FIGURE II.20: STM32CUBEIDE ---- 25
FIGURE II.21: STM32CUBEIDE ---- 26
FIGURE II.22: STM32CUBEIDE ---- 26
FIGURE II.23 : ESP8266 WI-FI MODULE ---- 27
FIGURE II.24: HTTP PROTOCOL ---- 29
FIGURE II.25: HIVEMQ ---- 32
FIGURE II.26: NODE-RED WORKSPACE ---- 32
FIGURE II.27: IOT LAYERS ---- 34
FIGURE II.28: GOOGLE TRENDS FOR NODE-RED ---- 35
FIGURE III.29: WATER-METER SYSTEM ---- 37
FIGURE III.30: BLOCK DIAGRAM OF WATER-METER ---- 38
FIGURE III.31: FLOW RATE ORGANIGRAM ---- 39
FIGURE III.32: POWER CIRCUIT ---- 42
FIGURE III.33: WATER FLOW METER DEACTIVATION CIRCUIT ---- 42
FIGURE III.34: BATTERY'S POWER VOLTAGE CIRCUIT ---- 43
FIGURE III.35: WATER FLOW DETECTOR CIRCUIT ---- 43
FIGURE III.36: STM32 AND LORA CIRCUIT ---- 44
FIGURE III.37: FULL SYSTEM ORGANIGRAM ---- 46
FIGURE III.38: ENERGY CONSUMPTION GRAPH ---- 47
FIGURE III.39: GATEWAY AND LORA FUNCTIONING CIRCUIT ---- 48
FIGURE III.40: NODE-RED ---- 49
FIGURE III.41: VALUES RECEIVED ---- 50
FIGURE III.42: UI DASHBOARD ---- 51

General Introduction

One of the most essential things in life is water due to its importance in our ordinary chores such as cleaning , showering…. , but providing water to the households has been an issue in our country ever since 1962 aka after independence ,there was no water policy Algiers was supplied with water from Mitidja and Sahel underground aquifers. The millennium water resource was sufficient to meet the needs of the population. The lack of a water policy induced wasteful behaviours. The use of water meters wasn't systematic and water was ceded at symbolic prices. This situation wasn't long before producing adverse effects on water resources. During this period, there was no hydraulic strategy for surface water mobilization. It should also be noted that the urban planning instruments did not take into account the water supply issues.[1]

However providing water supply was being solved ever since till this day but sadly the increment of country population meant more water usage with no supervision and that's where technology stepped in by offering intelligent water meters that collect data that has been used during the day so every individual of each house can be informed of the amount of water that has been used.[1]

Water meters are devices used to measure the amount of water consumed by residential, commercial, and industrial buildings. They play a crucial role in managing and monitoring water usage, allowing for accurate billing, leak detection, and water conservation efforts.[1]

Water meters operate on the principle of volumetric measurement, measuring the flow of water passing through them. They are typically installed at the point of entry of a building or within individual units to measure water consumption. The most common types of water meters include mechanical, electromagnetic, and ultrasonic meters.[1]

Mechanical water meters consist of a chamber with a rotor or a piston that rotates or moves as water flows through it. The rotation or movement is directly proportional to the volume of water passing through the meter, allowing for accurate measurement. These meters often have a dial or a digital display that shows the water consumption in cubic meters, gallons, or other standard units.[1]

Electromagnetic water meters use electromagnetic induction to measure the flow of water. They have a pipe section with two magnetic coils that create a magnetic field across the water flow. As water passes through this field, an electric voltage is induced, which is proportional to the velocity of the water. By measuring this voltage, the flow rate and total consumption can be determined.[1]

Ultrasonic water meters use ultrasonic waves to measure water flow. They have two sensors, one acting as a transmitter and the other as a receiver. The transmitter emits ultrasonic waves, which travel through the water. The receiver detects the waves after they have traveled across the pipe, measuring the time difference between the transmitted and received signals. Based on the time difference, the meter calculates the flow rate and total consumption.[1]

Water meters are typically calibrated to ensure accurate measurement, and they may require periodic maintenance and verification to maintain their precision. They are often connected to a

remote reading system or data logger, enabling automated meter reading and efficient data management for billing purposes.[1]

In addition to accurate billing, water meters also play a crucial role in identifying leaks or abnormal water usage patterns. By monitoring water consumption regularly, unusual spikes or sustained high consumption can be detected, indicating a possible leak or wastage. This allows for timely repairs and conservation efforts, contributing to sustainable water management.[1]

Overall, water meters are essential devices in managing water resources efficiently, promoting fair billing practices, and encouraging water conservation. They provide valuable data for utilities, building owners, and consumers, helping to monitor and control water usage effectively.[1]

CHAPTER 1 : Water – Meters

Introduction:

Water metering is an intelligent way to understand water usage in a household or an apartment society. Given the social, economic and environmental impact of water scarcity, it is best to start conserving the things that people usually take for granted, water being on top of that, Over the last 25 years, there has been a significant decline in rainfall. Consequently, the quantities of water flowing into the catchment's basins of the dam have decreased and the basins are filled only partially [2][2]owing to inefficient water management practices and leakages. Checking the wastage of water and our own pattern of water usage, is the easiest way to conserve water. Here, we look at what you gain when you go in for water metering, types of water metering…

Water meters have been used for almost two centuries, but their evolution has sped up rapidly in recent years. In the past, water company employees went house-to-house to read water meter data. Now, most water companies are equipped with automatic meter-reading technology. Reading your water meter is a good practice to ensure that readings that calculate your bills are accurate and to give you knowledge about your water consumption.

I.1 Water-meter :

A water meter is a device that measures the volume of water used in a building supplied with municipal water (figure 1). In a residential context, a water meter can be read to determine that water bills are accurate and ensure no leaks are present in a home. Water meters measure the volume of water in either gallons or centrum cubic feet (CCF). The volume of water read in a meter reading can be subtracted from the previous reading to calculate total water usage between readings. Water metering can educate you on your regular water usage, ensure you are being charged a fair amount, and alert you to any irregularities in water usage. [3]



Figure I.1: Water Meter

I.2 The Evolution of Water- Metering :

Water metering is a relatively new field. The modern water meter was invented in the mid-19th century and has only seen widespread use for a little over 100 years. Water metering technology has improved significantly since then, and continues to do so. Let's take a look at the evolution of water metering, and the benefits each improvement in technology has realized. [4]

I.2.1 Manual Water- Meter Reading :

If you are of a certain age you probably remember the meter man walking through your neighborhood once every two or three months. He would walk up to each house and read the water meter, jotting down the numbers on a notepad. This information would then have to be compiled and analyzed back at the municipal offices.While the adoption of computer technology vastly improved the efficiency of this process, the drawbacks to this method are obvious: it is costly and time-consuming, with a very real margin of human error. [4]

I.2.2 The Postcard Method of Water- Meter Reading :

In areas where it was not practical to send a meter reader individually to each house (for instance, in some of the sparsely populated water districts in the western U.S., which may encompass thousands of square miles), an honour system was devised. Once or twice a year, a postcard would be supplied to each water customer, who would then be expected to read their own meter and send the results in.[4]

I.2.3 Walk-By Water-Metering :

The first big technological improvement to manual reading came with the invention of a hand-held device which allowed the meter reader to simply touch the device to the meter to download the information, rather than have to open and read each meter individually. This not only saved significant time and labor costs, it also improved the accuracy of readings.[4]

I.2.4 AMR and Drive-By Metering :

With the invention of AMR (Advanced Meter Reading) technology, meter readers no longer had to walk down the street. Information from each meter is wirelessly transmitted every few seconds, and is accessible for a short distance, enabling the meter reader to simply drive by and collect the data with a transceiver in the vehicle. The information is then downloaded into CIS software and from there to billing software, which generates the invoices.At this point in time, AMR is still the most-sold water metering technology on the market. However, it does have some significant drawbacks.

Because the data transmission from an AMR unit is short-range, it still requires vehicle deployment. This adds to the cost and environmental impact of metering and limits the frequency with which it is practical to take readings. Also, it is very common to miss a few readings per route. At the end of the day, it is then necessary for the meter reader to go back and manually collect information from those meters that did not report.[4]

I.2.5 Advanced Metering Infrastructure (AMI) :

The newest water metering technology eliminates the physical visit to the metered neighborhoods. With the long range data transmission capability of AMI technology, data can now be collected directly and automatically over a distance of ½ to 5 miles away. This has revolutionary implications:[4]

- AMI enables far more frequent meter readings, such as hourly or even at 15 minute intervals. This gives a much more granular picture of water consumption and can help a water utility rapidly locate and address leaks that in the past would have gone unnoticed for weeks or months.[4]
- AMI allows for improved customer communication to address water usage issues. Its smart software can be programmed to automaticallyflag anomalous usage, and contact the customer through email or text.[4]

I.3 Benefits of using Water-Meter :

- **Incentive to conserve water:** Using a water meter acts as an incentive for you to conserve water. It is the first step towards discipline and monitoring your water use pattern.[5]

- **Volumetric pricing:** Why pay more for water use when you aren't using it? If you install water meters, you will pay only as much as you use.[5]

- **Lower energy consumption:** Directly or indirectly, you are misusing energy when you waste water. Water metering ensures that you minimize energy consumption, when you use only as much water as you need.[5]

- **Regular supply:** If water consumption is checked and reasons related to wastage identified and corrected, housing societies will be better equipped to ensure that the water supply is adequate and equitably distributed to all households. Water supply in unoccupied flats can be temporarily cut, as well.[5]

- **Detect water wastage:** It is important to monitor water wastage through leakages, dripping taps or any other reason. Some of you may be using water in moderation but end up paying (or wasting) more owing to structural defects in the source of

  water at home or at the distribution level. Water meters will help you analyse and take action against such problems.[5]

- **Estimating subsidies:** Authorities often subsidise water charges for those who cannot afford to pay for it. However, it is important to check water wastage in such scenarios, especially because the state authorities often have to bear a huge cut in their revenue when they offer subsidies. Water metering is an efficient way to understand usage and set targets when rolling out subsidies.[5]

I.3.1 Water-meters Flaws :

- Water meters rely on consistent water pressure to be functional, and this is not a given in developing countries.[6]

- Customers would have to pay for all water used (according to the meter reading), even if some of that water was from a leak in their home[6]

- Families with higher-than-average water use will likely end up spending more money on their water if they use a volume-based water meter compared to paying a monthly flat fee[6]

- It is more difficult for customers to prepare for a fluctuating water bill, and customers could end up underestimating their water use and money owed.[6]

I.4 Ways of reading a Water-Meter :

No matter the type of water meter in your home, reading the meter is typically simple and quick. Water meters use one of two types of displays: analog and digital. Regardless of which display your meter uses, you can keep track of your water usage the same way. Water meters work much like the odometer of a car. They record the total volume of water used over the life of the meter. Because of this, you should log the meter's reading, alongside the date, each time you check your meter.Before you can read your water meter, you must first locate it. Water meters can be found where your main water valve is located. In most homes, this will be in one of two places: under a plate where your yard meets the street or in your basement. When you find your meter, take note of the unit of measurement that it uses.[3]

- An **analog water meter** (figure 2) The face of your water meter, often covered by a plastic, flip lid, tracks your water with either an analog dial or digital reader. An analog dial meter will have one or more moving hands that move clockwise around the meter. If your meter has an analog dial, your water usage will be recorded in either gallons or cubic feet, as indicated on the bottom of the face of your water meter. As the hand moves from one number to the next, 1 gallon or 1 cubic foot of water as passed through your meter. So when the moving hand makes a complete rotation of the face of the water meter, 10 gallons or 10 cubic feet of water has passed through your water meter. To read a water meter in gallons, take note of the number at the bottom of your water meter. The number to the far right is a "standing 0", represented by the number of gallons used as indicated by the hand , Some analog water meters read water usage in cubic feet, instead of gallons, and incorporate a decimal point, reading like an odometer in a car. The conversion between cubic feet and gallons is **1 cubic foot = 7.48 gallons** of water; to calculate your water usage in gallons on a meter that reads in cubic feet, multiply the number on the bottom of the face of your water meter by 7.48. Take note of the decimal point; on these odometer water meters, the decimal point may be written as a period or a comma.[7]



Figure I.2: Analog water-meter

- A **digital water meter** (figure 3) contains the same information as an analog meter, but it displays this data on a digital screen. To activate a digital meter, you may need to expose it to some form of light, such as a flashlight. These meters cycle through readings for the total

number of gallons used and the active flow rate of water in the home. if the active flow rate on a digital meter is increasingthere is no water turned on in the home, then there is likely a leak present.[3]



Figure I.3: Digital Water-Meter

I.5 Types of Water-Meters :

There are two major types of water meters used to measure and monitor water flow: positive displacement and velocity. Both types of meters contain multiple subtypes that use different methods to achieve the same result. These subtypes are each beneficial with their own specific viscosities, contamination levels, and flow rates.[3]

I.5.1 Positive Displacement Water- Meters :

Positive displacement (PD) water meters (figure 4) are a type of mechanical water meter that are often used in low-volume residential and commercial applications. PD water meters utilize rotation of a chamber to trap small quantities of water as it passes through. The meter counts the number of times the chamber rotates to determine the volume of water flowing past. The speed and volume of incoming water determine how fast the chamber rotates, so these factors do not cause inaccuracies in the final count. Positive displacement meters are not typically used with water outside of residential applications. They are used with high-viscosity liquids. These applications include chemical injection, fuel measurement, hydraulic testing, and many others.[3]



Figure I.4: Positive Displacement Water-Meter

- **Advantage of Positive Displacement water meters :**
  - Accurate and precise readings regardless of flow variation
  - Low maintenance
  - Long lifespan
  - Easy to read
  - Can handle high-viscosity liquids
  - Most types do not need a power source

- **Disadvantage of Positive Displacement water meters :**
  - Can block flow if water is too contaminated
  - Not useable in high-flow applications
  - Can cause fluctuations in flow

I.5.2 Velocity water meters :

Velocity water meters, also known as flow meters, calculate volume by measuring the speed at which water flows through the meter. Several subtypes of velocity meters measure the flow of water in different ways, and they all use the determined speed to calculate the volume of water that has passed through. The four types of velocity meters are jet, electromagnetic, ultrasonic, and turbine.[3]

I.5.3 Jet water meters :

Jet water meters(figure 5) measure water flow by calculating the rotational speed of an impeller. Of the types of velocity water meters, jet meters are the most common. These come in single-jet and multi-jet configurations. Single-jet water meters have one port that water flows through to impact an impeller. This contact spins the impeller with a force that allows the flow of water to be calculated. Multi-jet water meters, on the other hand, contain multiple ports for water to flow through. This allows the impeller to be impacted at multiple points. The added ports grant multi-jet pumps longer lifespans and help the accuracy of readings when water flow is low.[3]

- **Advantage of Jet water meters :**
  - Inexpensiveness of single-jet configuration
  - Long lifespans of multi-jet meters
  - Stability of multi-jet meters
- **Disadvantage of Jet water meters :**
  - Clogging risk in single-jet configuration
  - Not effective with contaminated water



Figure I.5: Jet Water-Meter

I.5.4 Electromagnetic Water-Meters :

Electromagnetic water meters (figure 6) use magnetic fields to measure water flow. These meters utilize the Law of Electromagnetic Induction, which states that liquids produce voltage when flowing through a magnetic field. The faster the fluid flows through the magnetic field, the higher the voltage. As a result, electromagnetic water meters can calculate the volume of water flow by measuring the speed at which water flows through the meter's magnetic field.[3]

- **Advantage of Electromagnetic Water-Meters :**
  - Consume abit of electricity
  - Can handle very low flow rates
  - Low flow obstruction
- **Disadvantage of Electromagnetic Water-Meters :**
  - Liquids used must be substantially electrically conductive
  - More expensive than many types of meters



Figure I.6: Electromagnetic Water-Meter

I.5.5 Ultrasonic Water-Meter :

Ultrasonic water meters (figure 7) utilize sound waves to calculate water flow. These meters use the Doppler Effect to determine how fast water flows through the device. The Doppler Effect is the phenomenon that occurs when vehicles rush toward and past a still observer. As the vehicle moves closer to the observer and rushes past, the pitch of the vehicle's sounds appears to change. This principle also applies to liquids. The meter's interior acts as the observer as water flows toward and past it. Ultrasonic water meters can measure the sound waves throughout this process and determine the water's speed. By discovering the speed, ultrasonic meters can calculate the volume of water that flows through them. These meters are commonly used to monitor wastewater and other dirty water applications.[3]

- **Advantage of Ultrasonic Water-Meter :**
  - Need little maintenance
  - Accurate and precise
  - Do not contain moving parts

- **Disadvantage of Ultrasonic Water-Meter :**
  - More expensive than most meter types
  - Small errors in installation can lead to large inaccuracy in measurements
  - Can be influenced by outside sounds (pumps, bubbles...)



Figure I.7: Ultrasonic Water-Meter

I.5.6 Turbine Water-Meters :

Turbine water meters (figure 8) use the rotational energy generated by water's interaction with a turbine to measure flow rate. The speed of the turbine directly correlates with the speed of the incoming water. This speed can then be converted to calculate the volume of water passing through the meter. Turbine water meters are similar to jet meters because they use rotational energy to calculate flow rate, but turbine meters do not contain the ports that jet meters utilize.[3]

- **Advantage of Turbine Water-Meter :**
  - Can handle high and low flow rates
  - Good accuracy/price ratio
  - Easy installation and maintenance
  - Minimal effect on water pressure

- **Disadvantage of Turbine Water-Meter :**
  - Cannot handle fluctuations in viscosity
  - Do not handle contaminants well
  - Accuracy affected by bubbles in liquids



Figure I.8: Turbine Water-Meter

CHAPTER 2 : ARCHITECTURE OF A WATER METE

Introduction:

Any person in every household would want his house to have the best equipment to make the daily activities easier. This desire for convenience and efficiency also applies to choosing a water meter for your home. But how do you know what to choose in the first place? The answer to this seemingly simple question requires careful consideration and thoughtful decision-making. It begins by understanding the intricacies of a water meter, delving into its architecture, and exploring its capabilities. By navigating through the layers of a water meter's design and functionality, you can uncover the key factors that define a good water meter. From accuracy and reliability to smart features and compatibility with existing systems, there are numerous aspects to evaluate. It's important to weigh the specific needs and requirements of your household, such as water consumption patterns, budget constraints, and future scalability. By considering these factors and conducting thorough research, you can make an informed decision and choose the best water meter that suits your household's unique needs. In this chapter, we will dive deeper into the world of water meters, exploring their architecture, functionalities, and factors to consider, enabling you to make a well-informed choice for your home.

II.1 Water flow meter :

Water flow sensor consists of a copper body, a water rotor, and a hall-effect sensor. When the water flows through the rotor, the later rolls(its speed changes with different rate of flow)and the hall-effect sensor outputs the corresponding pulse signal. This one is suitable to detect flow in water dispenser or coffee machine**.** More importantly, the life of the copper one is longer than that of plastic body.[8]

II.1.1 Water flow sensor Features :

- Compact, Easy to Install
- High Sealing Performance
- High Quality Hall Effect Sensor

II.1.2 Water flow meter :

We use a water flow sensor to measure the water flow rate. The water flow rate is the volume of fluid that passes per unit time. People often use water flow sensor for automatic water heater control, DIY coffee machines, water vending machines, etc. There are a variety of flow sensors of different principles, but for makers using Arduino or Raspberry Pi, the most common flow sensor is based on a Hall device. For example, the most classic water flow sensor YF-S402 and YF-S201(figure 9) rely on Hall sensors.[8]



Figure II.9: YF-S201

II.1.2.1 YF-S201 Specifications :

- Mini. Wokring Voltage: DC 4.5V
- Max. Working Current: 15mA (DC 5V)
- Working Voltage: DC 5V~24V
- Flow Rate Range: 1~30L/min
- Load Capacity: ≤10mA (DC 5V)
- Operating Humidity: 35%～90%RH
- Storage Temperature: -25～+ 80℃
- Storage Humidity: 25%～95%RH

II.1.3 Concept of Water flow meter :

The main components are the Hall Effect sensor, turbine wheel, and magnet (figure 10). The water flows in through the inlet and out through the outlet. The water current drove the wheel to turn, and the magnet on the wheel turned with it. Magnetic field rotation triggers the Hall sensor, which outputs high and low level square waves (pulse).[8]

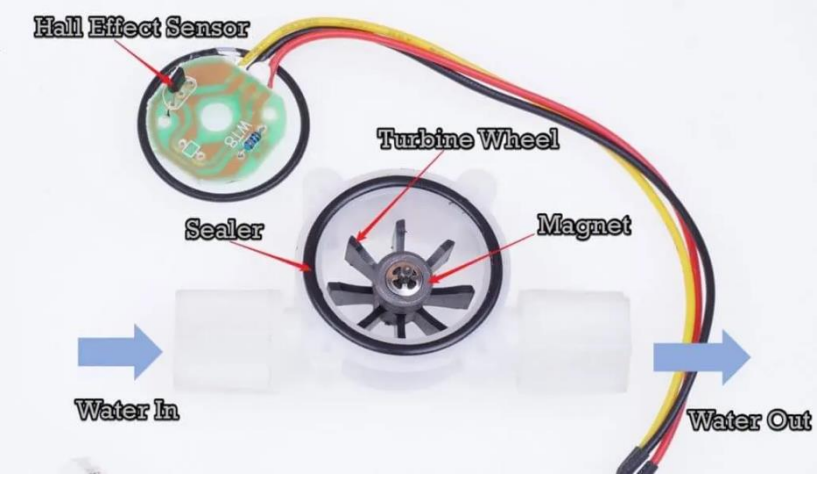


Figure II.10: Architecture of water flow sensor

II.1.4 Water generator :

Micro hydro generator : the potential energy conversion of the water into electrical energy mechanical equipment (figure 11) .[9]

There are 3 kinds of generators:

- **F50-80V** DC output , can directly charge 3-12V 10W LED lights, small battery, the maximum charge current 220mA.
- **F50-12V** DC voltage regulator output, can give the 12V radio power, charging, power supply to the LED.
- **F50-5V** regulated output, can charge a phone.



Figure 11: F50 -5V Generato

II.1.5 Water flow detector Description:

Water Flow Switch (figure 12) adopts a Magnetic Reed Switch, G3/4 male threaded ports each end. Equipped with filter water entry, it is mainly used for water heater, boiler, water dispensers, steam equipment, water system, the campus card sales, selling self-help water systems, water treatment equipment, cooling systems, mechanical equipment, instrumentation.....[10]



Figure II.12: water flow detector

II.2 Definition of LoRa:

Lora (long rage): LoRa is a wireless modulation technique derived from **Chirp Spread Spectrum (CSS)** technology. It encodes information on radio waves using chirp pulses - similar to the way dolphins and bats communicate! LoRa modulated transmission is robust against disturbances and can be received across great distances.[11]

II.2.1 LoRa's potential:

LoRa is ideal for applications that transmit small chunks of data with low bit rates. Data can be transmitted at a longer range compared to technologies like Wi-Fi, Bluetooth or ZigBee. These features make LoRa well suited for sensors and actuators that operate in low power mode. It can be operated on the license free **sub-gigahertz** bands, for example, 915 MHz, 868 MHz, and 433 MHz (433 MHz in our case), It also can be operated on **2.4 GHz** to achieve higher data rates compared to sub-gigahertz bands, at the cost of range. These frequencies fall into ISM bands that are reserved

internationally for industrial, scientific, and medical purposes.[11]

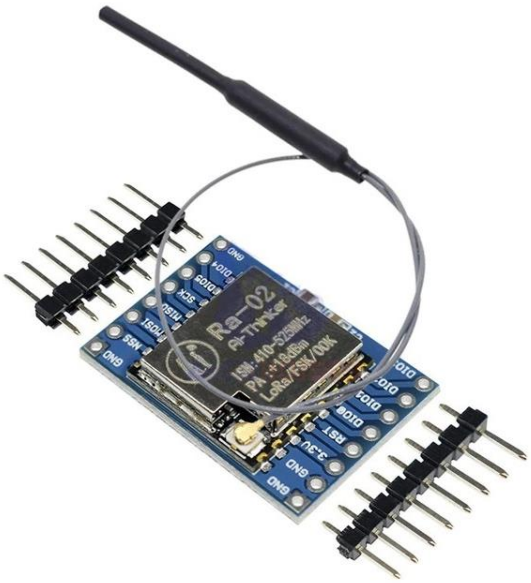


Figure II.13: LoRa Module

II.2.2 LoRa ‘s Features :

- LoRa Max range : 15-20 km.[12]
- Low Power: Battery life up to 5 years.[12]
- Operating voltage : 1.8-3.7v (DC). [12]
- Deep sleep consumption [12]

II.2.3 LoRa Modulation :

Modulation is converting digital data into analog signals that can be transmitted wirelessly. In LoRa modulation, a special technique called chirp spread spectrum (CSS) is used, where the frequency of the transmitted signal changes over time in a specific pattern called a chirp.This chirping effect allows the signal to have a long duration, which helps it overcome interference and multipath effects. In CSS modulation, data is encoded in the frequency modulation of the chirp signal.

The CSS modulation in LoRa uses a wide bandwidth and a slow chirp rate, which results in long symbols and increased resilience to interference. This modulation scheme enables LoRa devices to achieve a long communication range.[13]



Figure 14: Chirp Modulation

II.2.4 Definition of NRF :

NRF : a wireless transceiver module (figure 13) that is manufactured by Nordic Semiconductor ASA and is an equally powerful and versatile device as the SX1278. This module operates in 2.4GHz and has a special protocol named *Enhanced Shock Burst (ESB)* to enable high power transmission and reception at lower power consumption. [14]

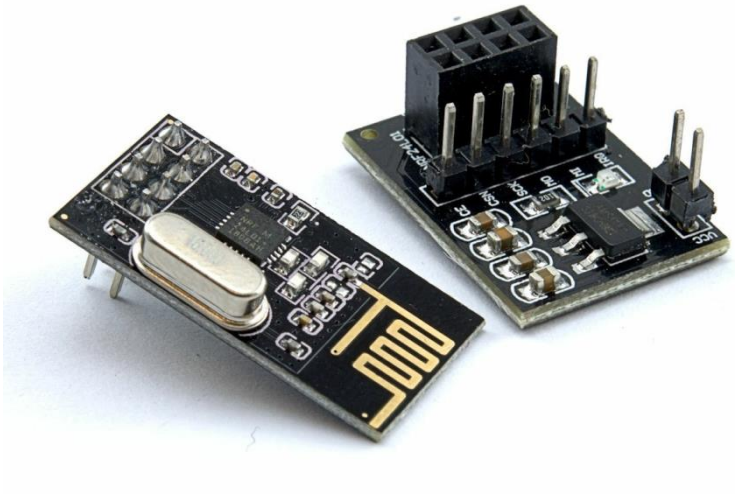


Figure II.15: NRF Module

II.2.4.1 NRF's Features :

- Max range : 800m
- Operating voltage : 3.3v(DC)
- Operating current : 12mA , 900Na power down mode,26uA standby mode .
- 250Kbps-2Mbps transmission rate (configurable)

II.2.4.2 Difference between LoRa and NRF :

Considering all the sides if we talking about the bandwidth, NRF will be a good option to use since it has a good bandwidth (2mbps vs 37.5kbps) , when we look at the range LoRa would defeat the NRF with its long range (10km vs 800m) . As well as we want a bi-directional communication, NRF does not offer that, only if we use other methods as TDM (time division multiplexing) .[14]

II.2.5 LoRa Gateway :

A **gateway** typically refers to the physical box or encasement housing the hardware and application software that performs essential tasks to connect IoT devices to the cloud. IoT devices use a gateway as a central hub to drop sensed knowledge and connect that data to external networks. A gateway is much like a Wi-Fi router. It has a LoRa concentrator, allowing it to receive RF signals sent out by LoRaWAN devices, which get converted to a signal compatible with a server, such as Wi-Fi, to send data to the cloud. It is a jumping-off point as all the devices on the network have to be able to transmit their data back to the gateway. This often means large scale LoRaWAN systems run on more than one gateway. The maximum number of sensors or devices per gateway depends on several factors such as data rate and data packet size but hundreds or even low thousands per gateway is achievable. [15]

II.2.6 STM32 :

STM32 series of MCUS which created by the STMicroelectronics consists of ARM Cortex-M thirty two-bit processor core. It is a very common type of microcontroller used in numerous types of devices. It also provides facilities to linking with other types of microcontrollers.[16]



Figure II.16: STM32

II.2.6.1 STM32F103C8T6 :

The STM32F103C8T6 is likely to remain the lowest-cost way for students and hobbyists alike to explore the ARM Cortex-M3 platform for quite some time. The device is readily available and is extremely capable. Finally, the form factor of the small PCB allows header strips to be soldered to the edges and plugged into a breadboard. Breadboards are the most convenient way to perform a wide array of experiments. The MCU on a blue PCB is affectionately known as the “Blue Pill,” inspired by the movie The Matrix. There are some older PCBs that were red in color and were referred to as the “Red Pill.”

- STM32 (STMicroelectronics platform)
- F1 (device family)
- 03 (subdivision of the device family)
- C8T6 (physical manifestation affecting amount of SRAM, flash memory, and so on)

As the platform name implies, these devices are based upon a 32-bit path and are considerably more powerful than 8-bit devices as a result, The F103 is one branch (F1 + 03) of the STM32 platform. This subdivision decides the CPU and peripheral capabilities of the device, Finally, the C8T6 suffix further defines the capabilities of the device, like the memory capacity and clock speeds.

The peripheral support of the STM32F103 is simply amazing . Peripherals included consist of:

- 4 x 16-bit GPIO Ports (most are 5-volt tolerant)
- 3 x USART (Universal Synchronous/Asynchronous Receiver/ Transmitter)
- 2 x I2C controllers
- 2 x SPI controllers
- 2 x ADC (Analog Digital Converter)
- 2 x DMA (Direct Memory Address controllers)
- 4 x timers
- watch dog timers
- 1 x USB controller
- 1 x CAN controller
- 1 x CRC generator
- 20K static RAM
- 64K (or 128K) FLASH memory
- ARM Cortex M3 CPU, max 72 MHz clock

There are some restrictions, however. For example, the USB and CAN controllers cannot operate at the same time. Other peripherals may conflict over the I/O pins used. Most pin conflicts are managed through the AFIO (Alternate Function Input Output) configuration, allowing different pins to be used for a peripheral function. In the peripheral configuration, several separate clocks can be individually enabled to tailor power usage. The advanced capability of this MCU makes it suitable for study. What you learn about the STM32F103 family can be leveraged later in more advanced offerings like the STM32F407 . The flash memory is officially listed at 64K bytes, but you may find that it supports 128K.



Figure II.17: STM32 Microcontroller

II.2.6.2 STM32 PINOUT :

- **CN6 :** It is a power supply pinout with a 3.3 volts [16]
- **RESET PIN :** it is used to reset the microcontroller.[16]
- **+3.3 volts pin :** it is an output pinout that provides the power of 3.3V for the microcontroller [16]
- **+5 volt :** It is an output pinout with a 5V power
- **GND :** it is the ground pinout
- **CN5 :** it is a combination of pinout D8 to D15 and these are digital pinouts used for SPI.
- **D13, D12,D11 and D10 Pinouts :** These pinouts operate as a serial clock, master in slave out, master out slave in, and SC pinout for SPI (Serial Peripheral Interface) transmission.
- **CN9 :** These are digital pinouts and comprises of D0 to D7 pins
- **D0 and D1 :** These two pinouts operate as receiver and transmitter for USART (Universal Synchronous Asynchronous Receiver Transmitter) transmission.
- **CN7 :** These are digital port C pinouts and comprises of PC0, PC1, PC2, PC3, PC10, PC11, PC12, PC13, PC14, PC15.
- **PD2 :** It is port D input-output pinouts

- **PA0, PA1, PA4, PA13, PA14, PA15 pinouts :** These are pinouts of port A.
- **PB7, PB8, and PB9 :** These are pinouts of port B.
- **PH0 and PH1 :** These are pinouts of port H.

.

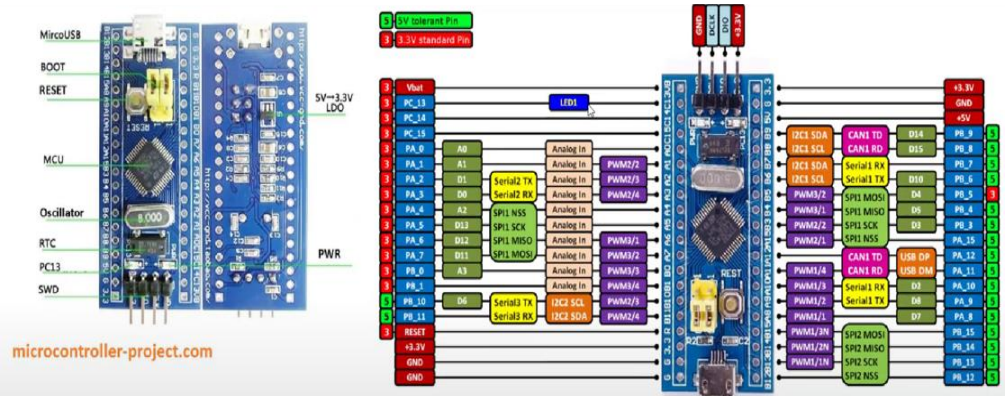


Figure II.18: STM32 PINOUT

II.2.6.3 STM32 Factor :

- Very low cost[16]
- Availability (can be found on sites such as Ali express , EBay...)[16]
- advanced capability[16]

II.2.6.4 STM32 Applications :

- It is used in less power using handheld devices.[16]
- It is used in different types of robots.[16]
- It uses an automation system.[16]

II.2.6.5 STM32CubeIDE :

STM32CubeIDE is an all-in-one multi-OS development tool, which is part of the STM32Cube software ecosystem , it is an advanced C/C++ development platform with peripheral configuration, code generation, code compilation, and debug features for STM32 microcontrollers and microprocessorssee figure (17, 18, 19, 20). It is based on the Eclipse®/CDT™ framework and GCC tool chain for the development, and GDB for the debugging , and allows the integration of the hundreds of existing plugins that complete the features of the Eclipse® IDE. STM32CubeIDE includes build and stack analyzers that provide the user with useful information about project status and memory requirements , also includes standard and advanced debugging features including

views of CPU core registers, memories, and peripheral registers, as well as live variable watch, Serial Wire Viewer interface, or fault analyzer.[17]

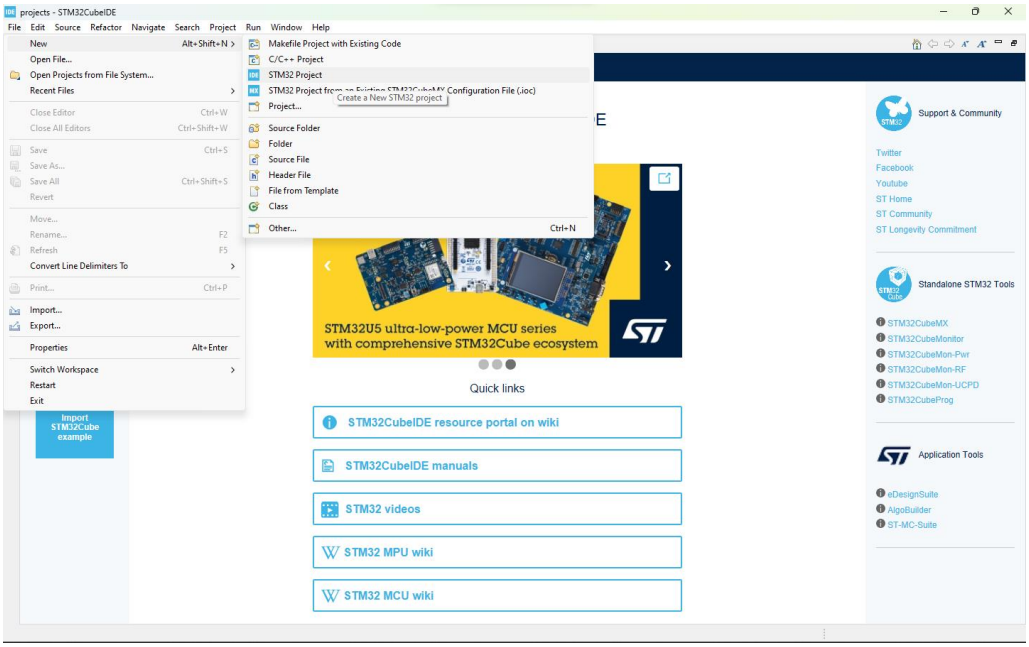


Figure II.19: STM32CubeIDE Project

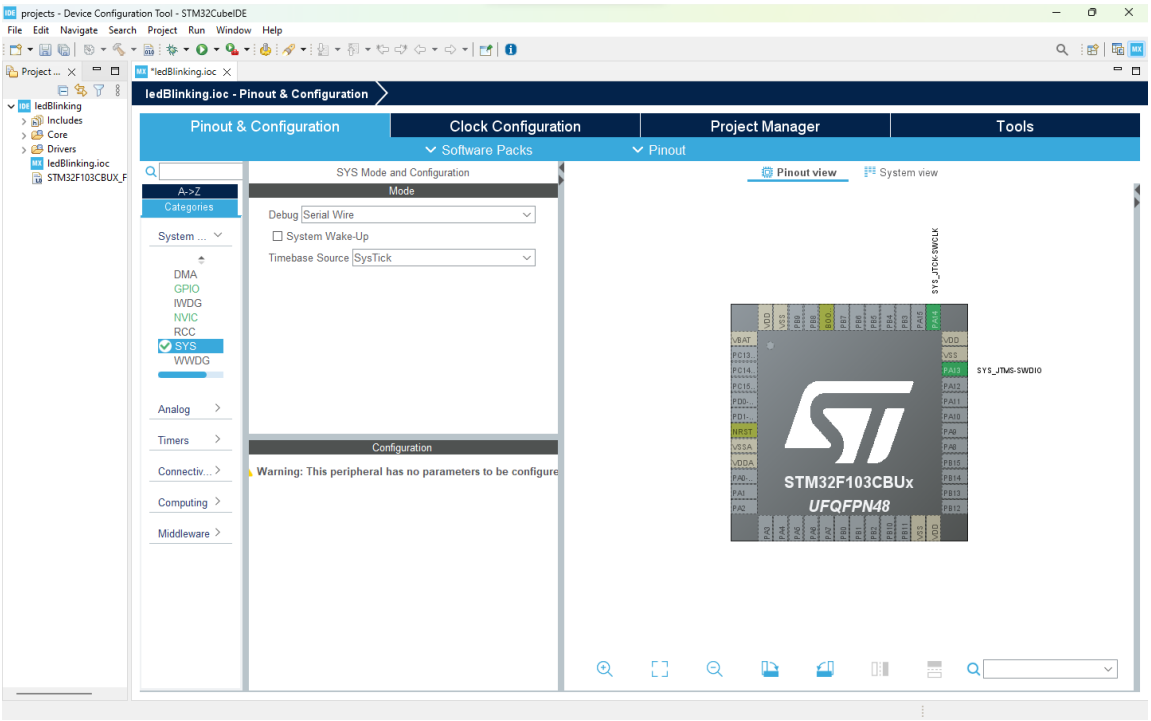


Figure II.20: STM32CubeIDE

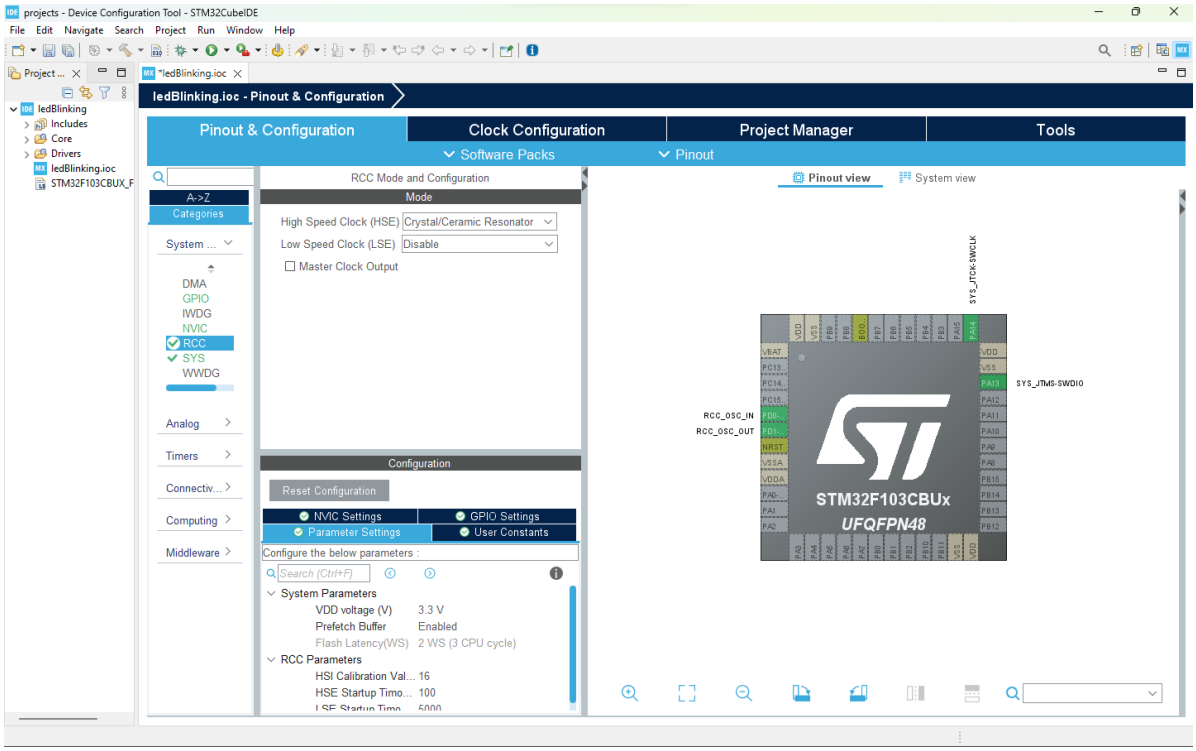


Figure II.21: STM32CubeIDE

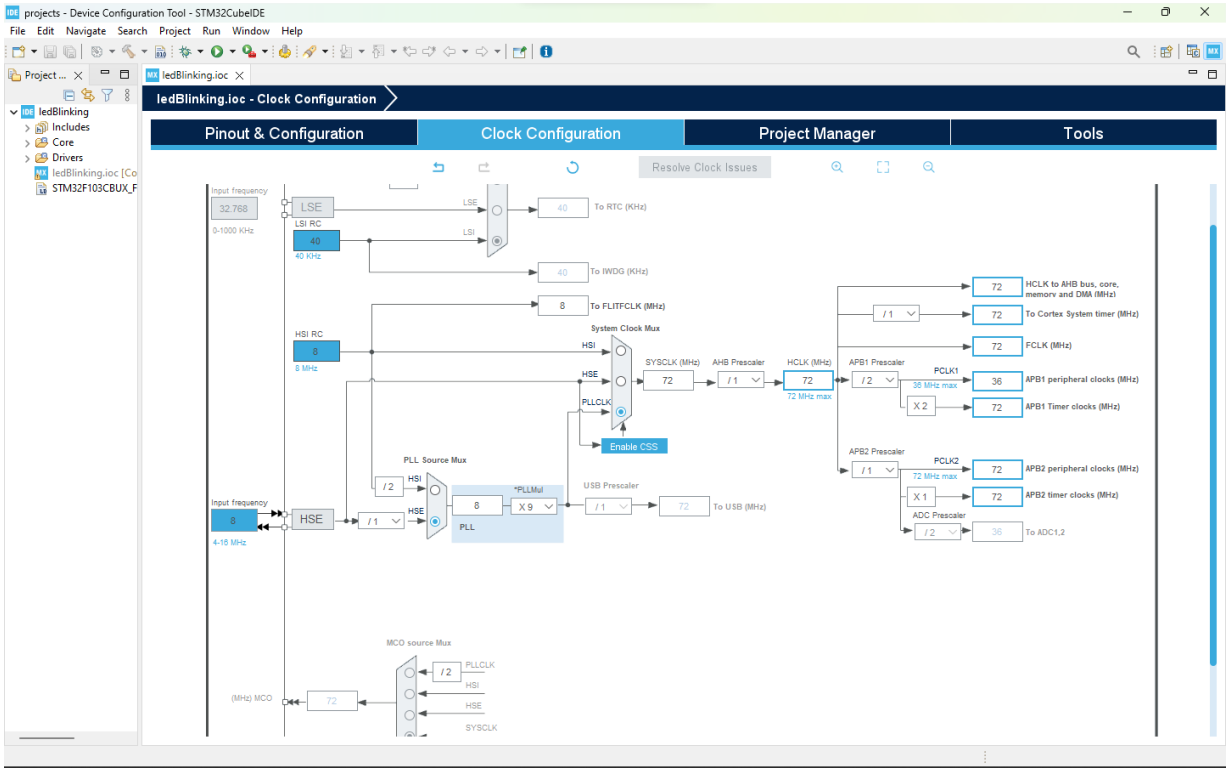


Figure II.22: STM32CubeIDE

II.2.7 Esp8266 :

The ESP8266 is a microcontroller integrated circuit (figure 14) with a system on a chip (SOC) Wi-Fi microchip for Internet of Things (IoT) applications produced by Espressif Systems. Given its low cost, small size and adaptability with embedded devices, it is now used extensively across IoT devices.[18]

II.2.7.1 WEMOSD1 MINI ESP8266 WIFI DEVELOPMENT BOARD:

The WeMos D1 mini is a miniature wireless 802.11 (Wifi) microcontroller development board. It turns the very popular ESP8266 wireless microcontroller into a fullyfledged development board. Programming the D1 mini is as simple as programming any other Arduino based microcontroller as the module includes a built-in micro USB interface allowing the module to be programmed directly from the Arduino IDE with no additional hardware.[19]

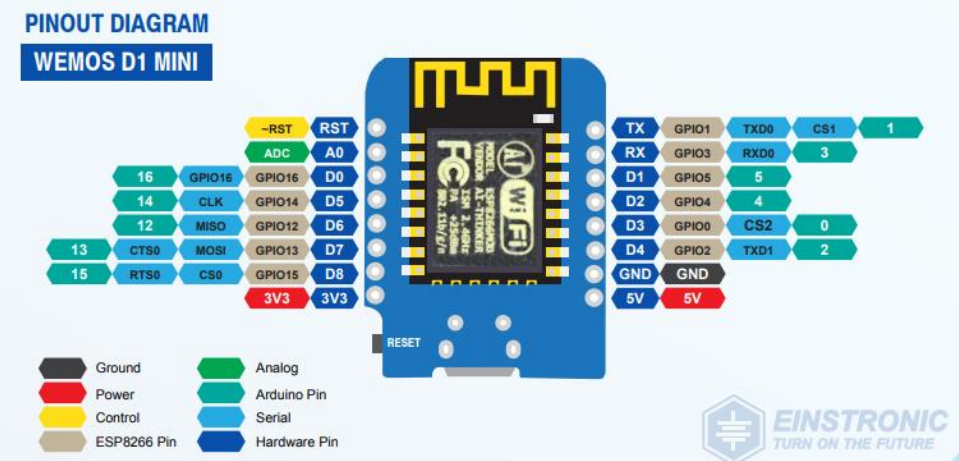


Figure 23 : Esp8266 Wi-Fi Module

II.2.7.2 Esp8266 Fonctions :

ESP8266 has many applications when it comes to the IoT . Here are just some of the functions the chip is used for [18]:

- **Networking:** The module's Wi-Fi antenna enables embedded devices to connect to routers and transmit data[18]
- **Data Processing:** Includes processing basic inputs from analog and digital sensors for far more complex calculations with an RTOS or Non-OS SDK[18]
- **P2P Connectivity:** Create direct communication between ESPs and other devices .[18]
- **Web Server:** Access pages written in HTML or development languages.[18]

II.2.7.3 Esp8266 Applications :

The ESP8266 modules are commonly found in the following IoT devices:

- **Smart security :** devices, including surveillance cameras and smart locks[18]
- **Smart energy :** devices, including HVACs and thermostats [18]
- **Smart industrial :** devices, including Programmable Logic Controllers (PLCs) [18]
- **Smart medical :** devices, including wearable health monitors[18]

II.2.8 Getting to know MQTT :

MQTT (MQ Telemetry Transport) is a lightweight open messaging protocol that provides resource-constrained network clients with a simple way to distribute telemetry information in low-bandwidth environments. The protocol, which employs a publish/subscribe communication pattern is used for machine-to-machine (M2M) communication. Created as a low-overhead protocol to accommodate bandwidth and CPU limitations, MQTT was designed to run in an embedded environment where it could provide a reliable, effective path for communication, suitable for connecting devices with a small code footprint[20]

II.2.8.1 Concept of MQTT :

Aimed at maximizing the available bandwidth, MQTT's publish/subscribe (pub/sub) communication model is an alternative to traditional client-server architecture that communicates directly with an endpoint. By contrast, in the pub/sub model the client that sends a message (the publisher) is decoupled from the client or clients that receive the messages (or the subscribers). Because neither the publishers nor the subscribers contact each other directly, third parties "the brokers " take care of the connections between them. MQTT clients include publishers and subscribers, terms that refer to whether the client is publishing messages or subscribed to receive messages. These two functions can be implemented in the same MQTT client. When a device (or client) wants to send data to a server (or broker) it is called a *publish*. When the operation is reversed, it is called a *subscribe*. Under the pub/sub model, multiple clients can connect to a broker and subscribe to topics in which they are interested. If the connection from a subscribing client to a broker is broken, then the broker will buffer messages and push them out to the subscriber when it is back online. If the connection from the publishing client to the broker is disconnected without notice, then the broker can close the connection and send subscribers a cached message with instructions from the publisher.[20]

II.2.8.2 Concept of HTTP:

HTTP (Hypertext Transfer Protocol) was invented as a component of the World Wide Web to transfer documents. It is most familiar to us as one of the enabling technologies that allows web browsers to work. Servers contain resources that are identified by the **URLs** that have the basic familiar form[21]:

https://www.hivemq.com/mqtt-essentials/

to which HTTP clients can make requests: GET, PUT, DELETE and POST, to name the most common. In a normal web environment, the simple model is a web browser retrieving web pages from a server with the GET method [21]:

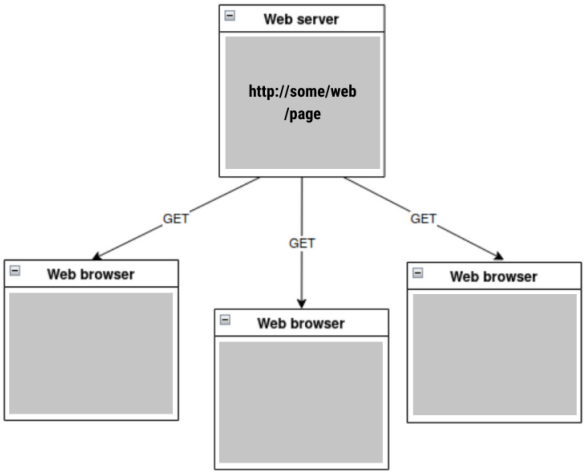


Figure II.24: HTTP Protocol

II.2.8.3 Understanding the difference between MQTT , HTTP :

MQTT was designed for the Internet of Things (although it wasn't called that at the time) whereas HTTP was created to make documents available across the internet. They both run over TCP connections, and are both client-server in architecture, but MQTT allows messages to pass in both directions between clients and servers whereas HTTP servers only respond to requests from clients. Once an MQTT connection is established, any number of messages can be sent through it in both directions, data from sensor to back-end, and commands the other way. HTTP deals with requests one at a time, with overhead such as authentication being carried out each time. HTTP does have

some ability to pool TCP connections, but the overhead for each message remains. MQTT message payloads are binary, whereas HTTP are textual so that binary data needs to be base64 encoded. [21]

II.2.8.4 An MQTT Broker :

An MQTT broker acts as a go-between the clients who are sending messages and the subscribers who are receiving those messages. In a post office analogy, the broker is the post office itself. All messages have to go through the broker before they can be delivered to the subscriber. Brokers may have to handle millions of concurrently connected MQTT clients, so when choosing an MQTT broker, enterprises should rate them based on their scalability, integration, monitoring and failure-resistance capabilities.[20]

II.2.8.5 Benefits of using MQTT :

The lightweight properties and minimum overhead of the MQTT protocol architecture help ensure smooth data transfer with low bandwidth and reduce the load on the CPU and RAM. Among MQTT's benefits[20] :

- Efficient data transmission and quick to implement[20]
- Low network usage, due to minimized data packets [20]
- Successful implementation of remote sensing and control [20]
- Fast, efficient message delivery [20]
- Low power usage, which is good for the connected devices[20]
- Optimizes network bandwidth.[20]

II.2.8.6MQTT in Iot:

Because MQTT clients are small, they require minimal resources and thus can be used on small microcontrollers, according to MQTT.org. To optimize network bandwidth. Plus, MQTT "can scale to connect with millions of IoT devices," according to the organization.As a result, MQTT is one of the most used protocols in IoT , for utilities industries to efficiently transmit data between their services and their customers and devices.[20]

Here are some examples of MQTT use in IoT instructor[20]

- **Smart metering:** The MQTT protocol can be used to transmit data with guaranteed message delivery to provide accurate meter readings in real time.[20]

- **Gathering ambient sensor data :** Sensors used in remote environments are often low-power devices, so MQTT is a good fit for IoT sensor build outs with lower-priority data transmission needs.[20]
- **Machine health data :** which provides a pub/sub messaging platform, gives the example of a wind turbine requiring "guaranteed delivery of machine health data to local teams even before that information hits a data center.[20]
- **Billing systems :** MQTT helps eliminate duplicate or lost message packets in billing or invoicing.[20]

II.2.9 Definition of HiveMQ :

HiveMQ is an MQTT broker and a client-based messaging platform (figure which uses MQTT protocol for fast, reliable and efficient bi-directional data transfer to and from IoT devices. It provides its own client library, but it can be used with any MQTT compliant client library. It can be deployed on a private, hybrid or public cloud. You can integrate HiveMQ with existing enterprise systems thanks to its open API and a flexible extension framework. HiveMQ also offers an open-source tool called MQTT CLI, which provides a command line interface to interact with MQTT brokers. This tool can be used in a shell mode, allowing you to use multiple MQTT clients simultaneously.[22]

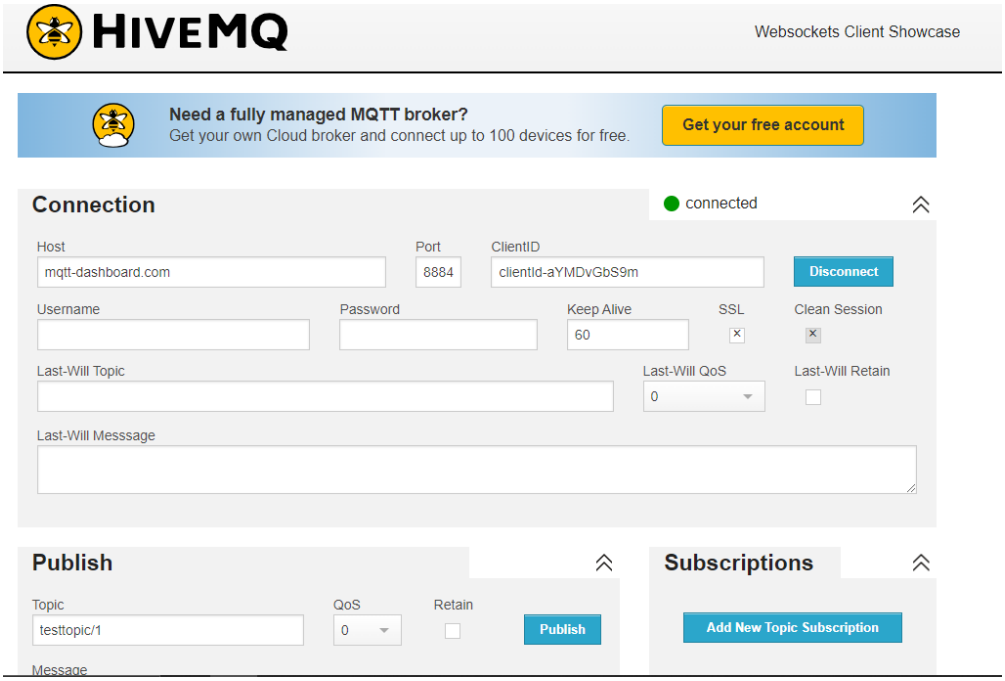


Figure II.25: HiveMQ

II.2.10 Definition of Node-Red:

Node-RED is a **virtual environment** that combines hardware devices, APIs and online services in new and interesting ways. It provides a browser-based editor (figure 24) that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click.[23]

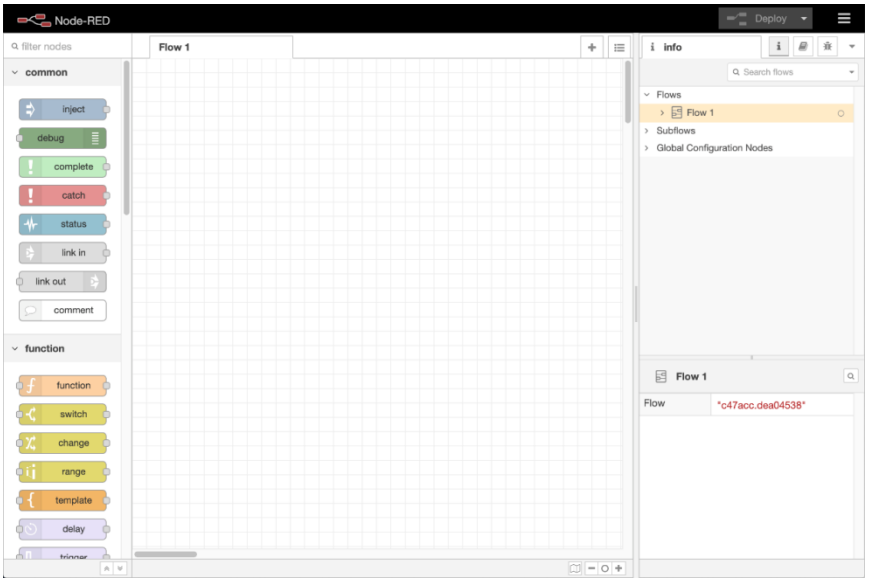


Figure II.26: Node-Red WorkSpace

II.2.10.1 Node-Red Features :

- Browser-based UI.[24]
- Works with Node.js and is lightweight.[24]
- Encapsulates function and can be used as a node (meaning functions are locked in an abstract capsule) .[24]
- You can create and add your own nodes.[24]
- Easy access to IBM Cloud services.[24]

II.2.10.2 Benefits of using Node-Red:

- **Simplification: When** programming with Node-RED, you'll notice its simplicity. As the name no-code/low-code indicates, coding is eliminated and programming is intuitively completed with a minimal number of operations needing to be used.[24] [24]
- **Efficiency:** The FBP typified by Node-RED can be completed with almost only GUI operations. Node-RED flow editor takes care of building the application execution environment, library synchronization, the **integrated development environment** (**IDE**), and editor preparation so that you can concentrate on development.[24]
- **Open source:** Node-RED is an open-source piece of software. Therefore, it can be used flexibly under the Apache2 license. Some are developing their own services based on Node-RED, while others are changing to their own UI and deploying it as built-in. As we mentioned previously, we have also established a platform where we can publish our own developed node so that anyone can use it.[24]
- **High quality:** High quality is the true value of flow-based and visual programming. Each node provided as a component is a complete module that has been unit tested. As a result, app authors can focus on checking the operation at the join level without worrying about the contents of node. This is a big factor that eliminates human error at the single level and ensures high quality.[24]

II.2.10.3 Definition of IoT:

The Internet of Things (IoT) describes the network of physical objects, that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet. These devices range from ordinary household objects to sophisticated industrial tools.[25]

II.2.10.4 Understanding the bond between Node-Red and IoT:

In other words, it can be said that Node-Red is suitable for building IoT-related services, such as data control on devices, and linking edge devices and cloud services. Originally, the development concept of Node-RED was for IoT.Now, let's look at the basic structure of IoT .[24][24]

It can be said that IoT is basically composed of six layers, as shown in the following diagram [24]:

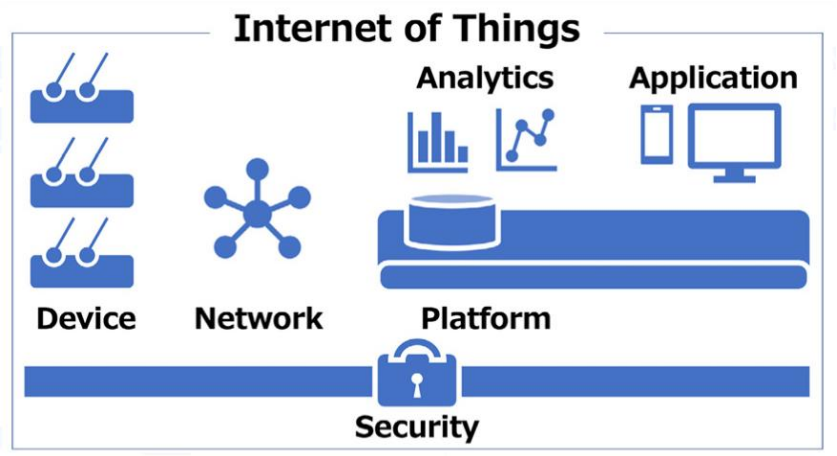


Figure II.27: IoT Layers

By taking a close look :

- **Device:** is a so-called edge device. IoT has various sensors and handles the data that's acquired from them. Since it doesn't make sense to have the data only on the edge device, we need to send that data through the Gateway to the network.[24]
- **Network:** it required to send the data that's been obtained from the device to a server on the internet. It usually refers to the internet. In addition to the internet, there is also a P2P connection via Bluetooth or serial.[24]
- **Platform:** The party that receives and uses the data is the platform. We may also have a database for activating and authenticating things, managing communications, and persisting received data.[24]
- **Analytics:** This is a layer that analyzes the received data. Broadly speaking, it may be classified as an application. This is the part that prepares the data so that it can be processed into a meaningful form.[24]

- **Application:** An application provides a specific service based on data analysis results. It can be a web or mobile application, or a hardware-specific embedded application. It can be said to be the layer that's used by the end user of the IoT solution.[24]

After we took a look on what does IoT covers , we can say that Node-Red is suitable for it , you can understand why FBP (Flow Based Programs) tools that have been developed for IoT survive when used with Node-RED. In particular, the following three points should be taken into account [24]:

- Since it can be run on edge devices (pre-installed on specific versions of Raspberry Pi OS), it is ideal for data handling at the device layer.
- Since it can be run on the cloud (provided as a default service in IBM Cloud), it is easy to link with storage and analysis middleware.
- Since MQTT and HTTP protocols can be covered, it is very easy to exchange data between the edge device and the server processing cloud.

As said , Node-RED, which largely covers the elements required for IoT, is now used for a wide range of applications, such as web services and chart display, as well as programming for IoT. Also, as of June 2020, if you look at Google Trends for Node-RED, you can see that the number of users is gradually increasing. As such, Node-RED is a very attractive FBP tool [24]:



Figure II.28: Google Trends for Node-Red

II.2.11 Database:

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management

system (DBMS).Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database.Data within the most common types of databases in operation today is typically modeled in rows and columns in a series of tables to make processing and data querying efficient. The data can then be easily accessed, managed, modified, updated, controlled, and organized. Most databases use structured query language (SQL) for writing and querying data.[26]

II.2.12 Structured Query Language (SQL):

SQL is a programming language used by nearly all relational databases to query, manipulate, and define data, and to provide access control. SQL was first developed at IBM in the 1970s with Oracle as a major contributor, which led to implementation of the SQL ANSI standard, SQL has spurred many extensions from companies such as IBM, Oracle, and Microsoft. Although SQL is still widely used today, new programming languages are beginning to appear.[26]

CHAPITRE 3 : EXPERIMENTS AND TESTS

Introduction :

In the quest for efficient monitoring and management of water consumption, the development of advanced water metering systems has become increasingly crucial. This project aims to design a state-of-the-art water metering solution that leverages the capabilities of the STM32 microcontroller unit (MCU). By integrating the STM32 MCU with a LoRa module for data transmission and an ESP8266 Wi-Fi module for MQTT server connectivity, to achieve seamless data collection and storage in a database.

III.1 Water-Meter Functional Diagram :

Wehave implemented an STM32 microcontroller in conjunction with a water flow meter and a water flow detector to accurately measure and monitor various parameters related to water flow. The STM32, known for its robust processing capabilities, acts as the central controller for the system. It receives data from the water flow meter and flow detector, allowing it to calculate essential parameters such as frequency, flow rate, and volume.The water flow meter precisely measures the rate at which water is flowing through the system, while the flow detector helps in detecting any anomalies or irregularities in the flow pattern. By combining the data from both sensors, the STM32 microcontroller provides real-time information about the flow characteristics of the water.

To transmit the data wirelessly, we have incorporated a LoRa (Long Range) module. LoRa technology offers a significant advantage in terms of long-range communication while consuming minimal power. This enables the system to send the calculated frequency, flow rate, and the volume over considerable gateway (esp8266).

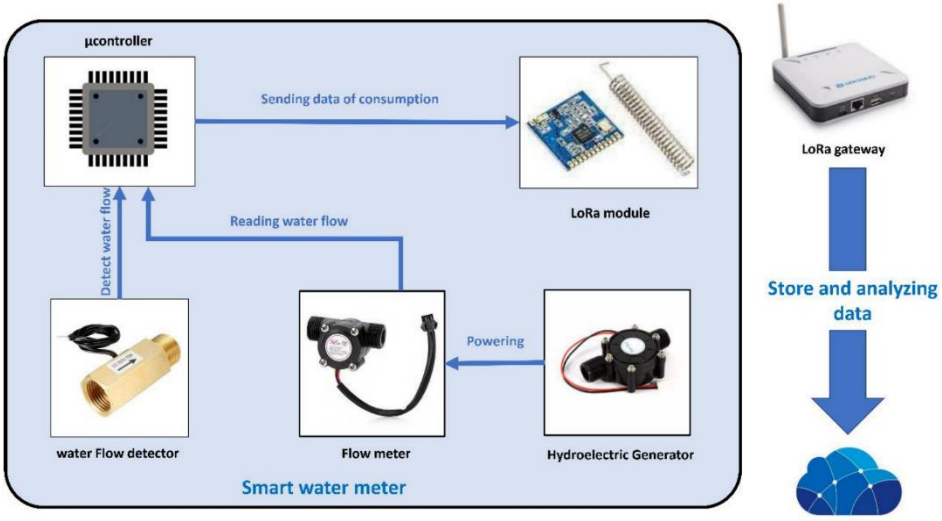


Figure III.29: Water-Meter System

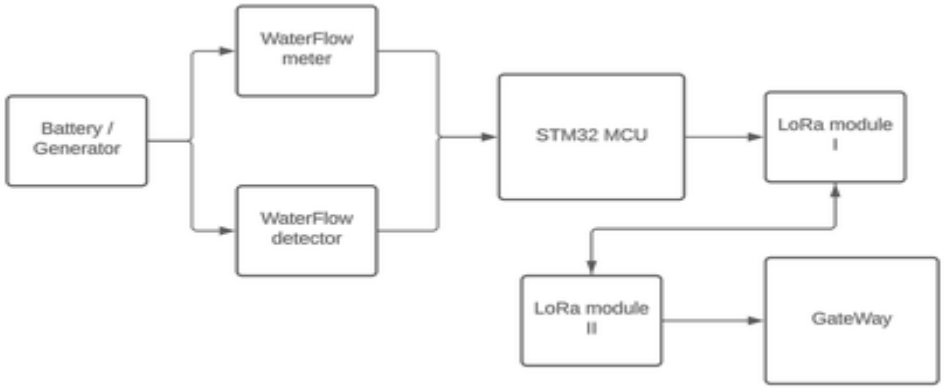


Figure III.30: Block Diagram of water-meter

III.2 Consumption measurement:

HAL_TIM_IC_CaptureCallback is a call back function which is used in conjunction with the STM32 HAL library for handling input capture interrupts on a timer peripheral , for further information to understand we got :

The function begins with the condition if (state == IDLE). It checks the value of a variable state to determine if it is in the IDLE state. Inside the first if block, the code reads the capture value (CCR1) of timer TIM2 into a variable T1.

It also sets a variable TIM2_OVC to zero and updates the state variable to DONE. If the previous condition is not met, the function checks the else if (state == DONE) condition. This means that if the state is not IDLE, it must be DONE.

Inside the else if block, the code reads the capture value (CCR1) of timer TM2 into a variable T2. It then calculates the number of ticks elapsed by subtracting T1 from T2 and adds the overflow counter value TM2_OVC multiplied by 65536. The result is stored in a variable called ticks.The code calculates the frequency by dividing the system clock frequency (F_CLK) by the ticks value. The frequency is stored in a variable called F. while the flow rate (rate of flow) is calculated by multiplying the frequency by 7.5 and dividing the result by 60.
Inside the if (frequency != 0) block, there are some commented out lines that appear to be transmitting messages over UART (Universal Asynchronous Receiver-Transmitter) or printing them to the console. These lines are currently disabled as they are commented out using // at the beginning of each line. Finally, the code sets the state variable back to IDLE.

In summary, this code is part of a larger program that handles input capture interrupts from timer TM2. It captures the timer values, calculates the frequency and flow rate.

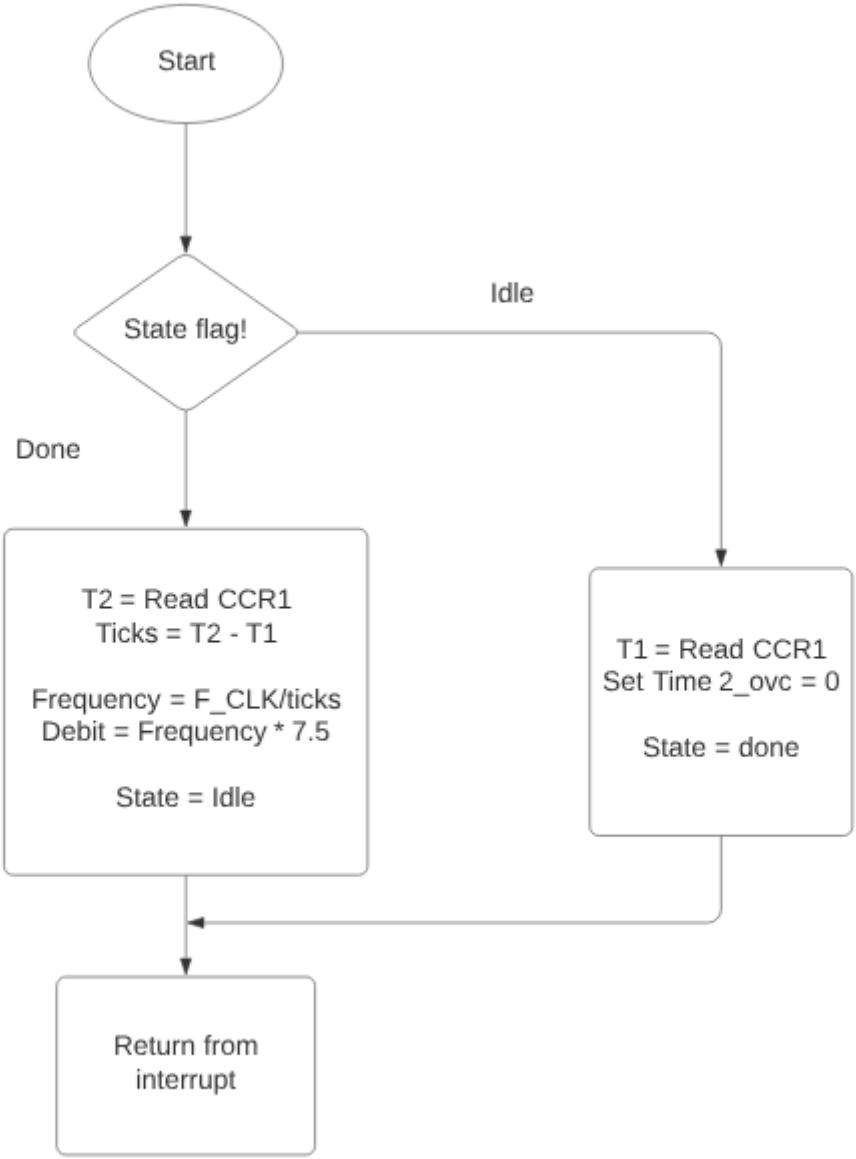


Figure III.31: Flow rate organigram

III.3 Volume Code :

The code is inside an if statement condition, checking if the difference between milliseconds and t1 is greater than 60. This suggests that the code block will execute when at least 60 milliseconds have passed since t1.

The code then checks if frequency is not equal to 0. If frequency is not zero, it performs the following operations:

Assigns the value of debit to d1.

Updates the volume variable by adding d1 multiplied by 0.000001 and 1000 to it.

Resets d1 and frequency to 0.

Updates t1 with the value of milliseconds.

If frequency is 0 then print the volume.

Finally, there is another if block that checks if frequency is not equal to 0, it prints the value of the flow rate.Overall, this code calculates and updates the values related to volume and flow rate based on the frequency measurement

III.4LoRa configuration :

we initialized and configured a LoRa module using a custom LoRa library. Here is a breakdown of what each part of the code does:

myLoRa = newLoRa();: This line creates a new instance of the LoRa structure or object, which holds the configuration parameters for the LoRa module.

The following lines assign values to various properties of the myLoRa object, specifying the GPIO ports and pins for different control signals, as well as the SPI interface (hSPIx) to communicate with the LoRa module. These values are specific to the hardware being used.

The next few lines set default values for LoRa communication parameters such as frequency, spreading factor, bandwidth, CRC rate, power level, overcurrent protection, and preamble length. These parameters determine the behavior and performance of the LoRa communication.

if(LoRa_init(&myLoRa)==LORA_OK): This line calls the LoRa_init function from the LoRa library and passes the address of the myLoRa object as a parameter. The function initializes the LoRa module based on the provided configuration.

If the initialization process returns LORA_OK, indicating that it was successful, the code sets LoRa_stat to 1. This variable can be used later in the code to check if the LoRa module initialization was successful.

III.5 Sleep Mode:

Let's go through the code and explain its functionality:

The code begins with an if statement checking if the standby flag (PWR_FLAG_SB) is set using the __HAL_PWR_GET_FLAG function. If the flag is set, it indicates that the microcontroller woke up from the standby mode.

Inside the if block, the standby flag is cleared using the __HAL_PWR_CLEAR_FLAG function.

A string "Wakeup from the STANDBY MODE\n\n" is defined, and it is transmitted via UART using the HAL_UART_Transmit function to communicate the wake-up event.

The code enters a loop that blinks an LED (GPIOC Pin 13) for a certain number of iterations (20 in this case) with a delay of 200 milliseconds between toggling the LED state.

The wake-up pin (PWR_WAKEUP_PIN1) is disabled using the HAL_PWR_DisableWakeUpPin function to prevent further wake-up events from that pin.

The standby flag (PWR_FLAG_WU) is cleared using the __HAL_PWR_CLEAR_FLAG function.

Another string "About to enter the STANDBY MODE\n\n" is defined and transmitted via UART to indicate that the system is about to enter the standby mode.

A while loop checks if GPIOA Pin 0 is not pressed (low state). Inside the loop, the following operations are performed:

Another string "STANDBY MODE is ON\n\n" is transmitted via UART to indicate that the system has entered the standby mode.

Finally, the microcontroller enters the standby mode using the HAL_PWR_EnterSTANDBYMode function.

This code handle the wake-up from the standby mode, perform certain operations, and enter the standby mode again based on certain conditions. The specific functionality and purpose of the code depend on the larger context and the overall program structure.

III.6 Power control circuit:

This circuit controls if our system will be working with the generator or the battery

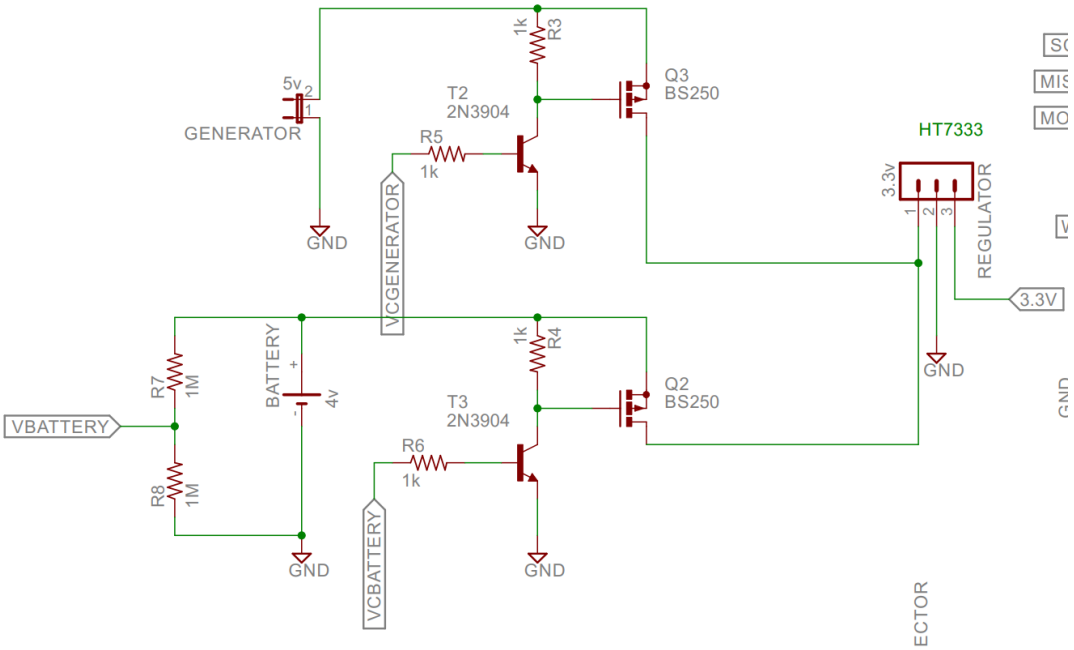


Figure III.32: Power circuit

III.7 Water flow meter deactivation circuit:

This circuit controls the activation of the water flow meter circuit.

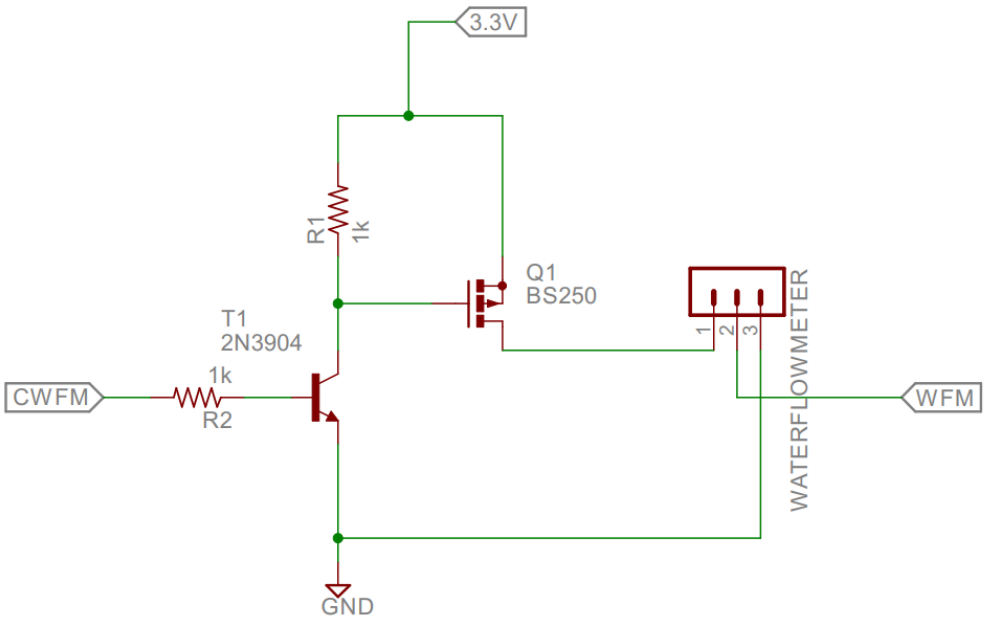


Figure III.33: Water flow meter deactivation circuit

III.8 Battery's power voltage circuit :

This circuit used to calculate the power voltage of the battery

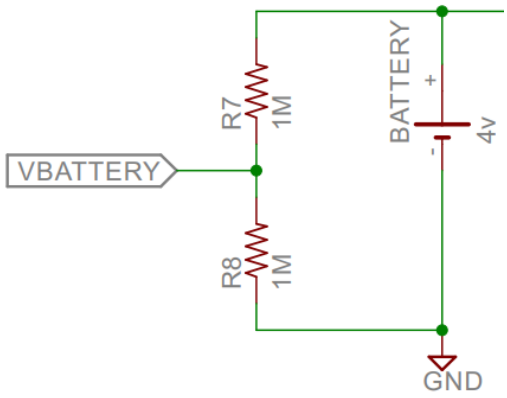


Figure III.34: Battery's power voltage circuit

III.9 Water flow detector circuit :

This circuit used to detect the water if available or not

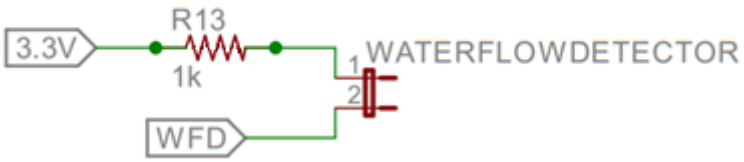


Figure III.35: Water flow detector circuit

III.10 MCU STM32 and LoRa module :

The connection between the microcontroller and the LoRa module and the other connections

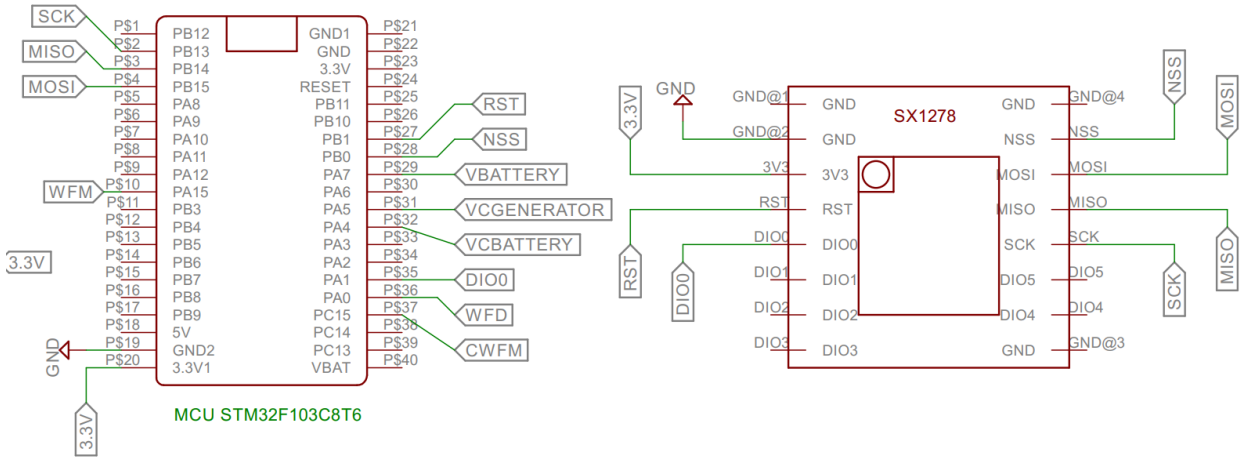


Figure III.36: STM32 and LoRa circuit

III.11 Valve control circuit:

This circuit used to control the valve

III.12 Full System Function:

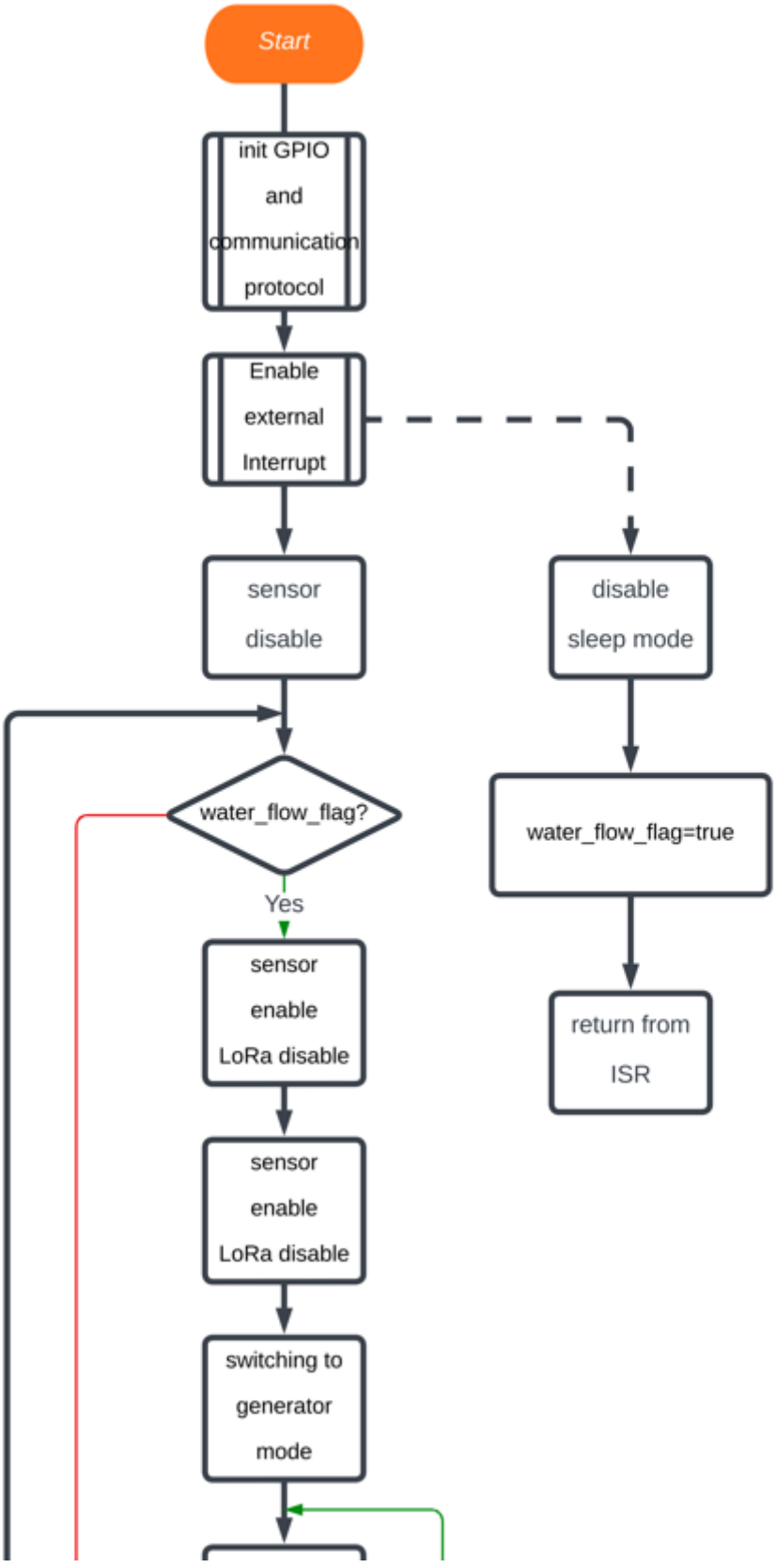
This system is based on three main functions; two of them work at the same time with the loop function.

The first one used to calculate the frequency and the flow rate using a timer T2 from the MCU, the other one that use another timer T3 to calculate the volume using the data from the flow rate that we got from the first function.

By using an interruption function we managed to make the system to enter the deep sleep mode to use less energy, as well as the rest of the devices.

Using a condition of the water's presence we can go back to wake up mode and start calculating the flow rate, volume, and the battery state of our circuit.

Before entering the sleep mode again we send our data to the gateway and receive data as well if available. The main idea of receiving a data is to control our valve so we can open and close our water system anytime needed. When the water flow detector is off, the system will go back to sleep mode waiting for the next interruption to pop.


Start
init GPIO and communication protocol
Enable external Interrupt
sensor disable
disable sleep mode
water_flow_flag?
water_flow_flag=true
Yes
sensor enable LoRa disable
return from ISR
sensor enable LoRa disable
switching to generator mode

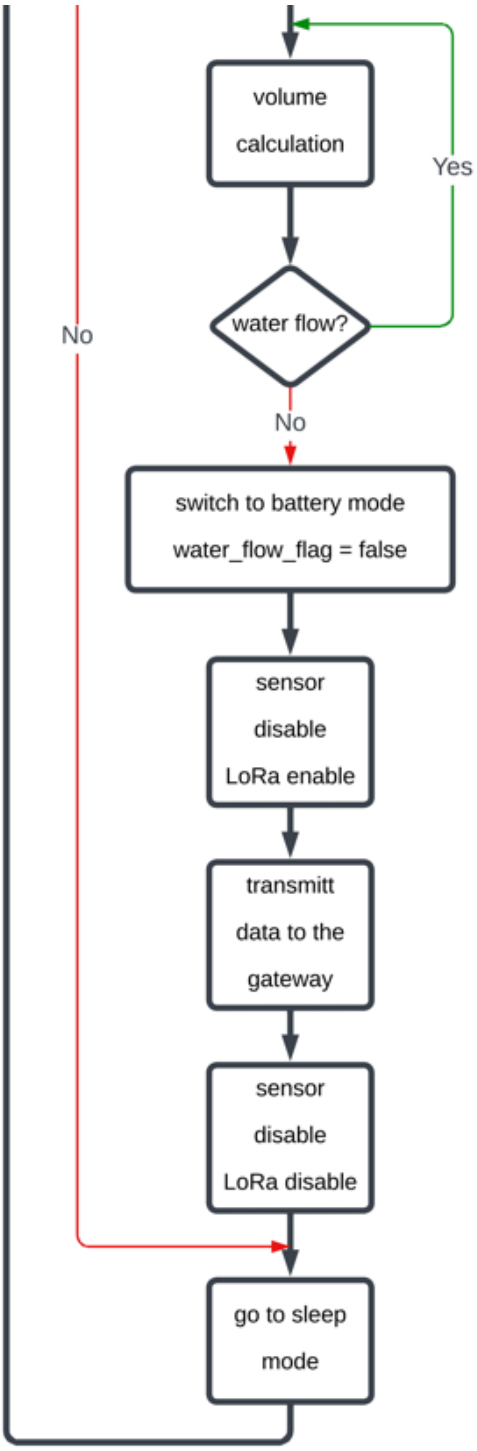


Figure III.37: Full system organigram

III.13 System power usage calculation :

This system is powered by a battery of 5000mAh, by calculating our system power usage, the whole system use about 50mA at the wake up mode.

After entering to the deep sleep mode, the system will start consuming between 3 and 5mA.

Using the deep sleep mode was a reason to extend our battery life

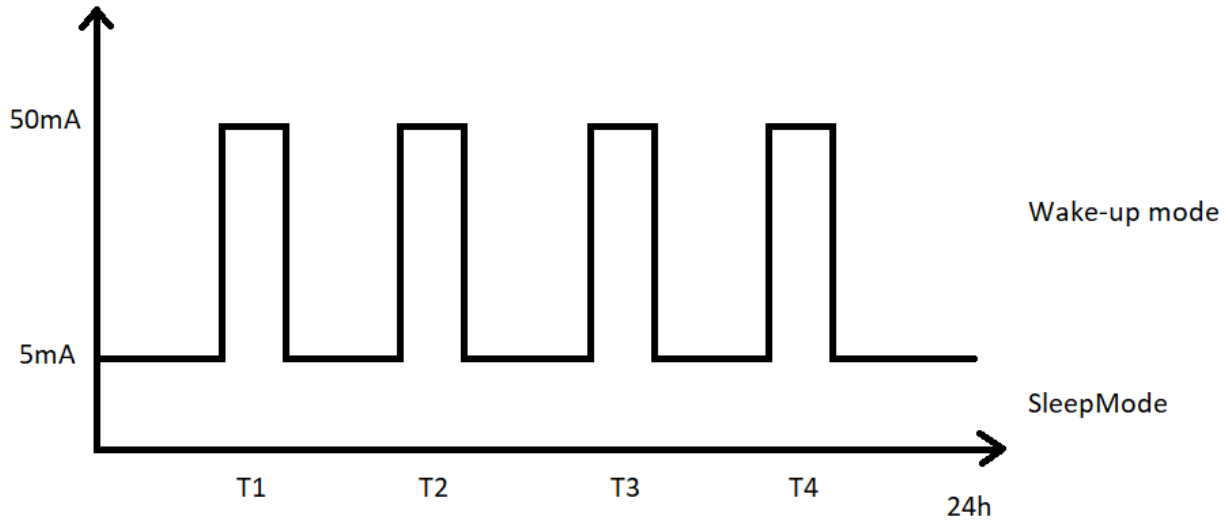


Figure III.38: Energy Consumption Graph

After calculating the power system consummation, our system life extended from 4 days to 41 days.

By including the time that the system gets powered by a generator, we can improve the power system more by adding a rechargeable battery.

III.14LoRa Gateway:

LoRa (Long Range) technology enables long-range wireless communication, making it ideal for IoT applications.Building a LoRa gateway using the ESP8266 Mini and LoRa SX1278 module. After Integrating LoRa communication with an MQTT broker server for seamless data transmission.

- **Hardware Setup:**

  - ESP8266 Mini Setup:Configure the ESP8266 Mini as a Wi-Fi access point or connect it to an existing network for internet connectivity.

- LoRa SX1278 Module Setup:Establish a connection between the ESP8266 and the LoRa SX1278 module using the SPI interface.Ensure proper wiring and connections between the ESP8266 and the LoRa module.

➢ **Software Development:**

- Firmware Programming: Utilizing Arduino IDE or a similar environment to program the ESP8266 Mini.Include necessary libraries for LoRa and MQTT communication.
- LoRa Configuration: Configuring the SX1278 module with appropriate settings, such as frequency, spreading factor, bandwidth, and coding rate.Use the LoRa library to enable LoRa communication between the ESP8266 Mini and other LoRa devices.
- MQTT Integration:Establish a connection to the MQTT broker server by providing the server's IP address, port number, and credentials.Utilize the MQTT library to publish and subscribe to MQTT topics.Define MQTT topics for sending and receiving data between the LoRa gateway and other MQTT clients.

➢ **Functionality:**

In the main loop of the program:

1. Read LoRa data received by the SX1278 module.
2. Publish the data as MQTT messages to the relevant topics.
3. Subscribe to specific MQTT topics to receive commands or data from other MQTT clients.
4. Transmit the received data via the LoRa module.

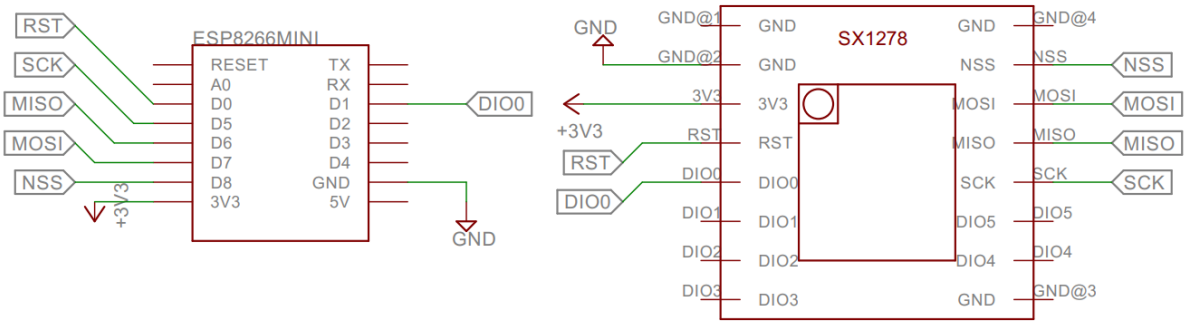


Figure III.39: Gateway and LoRa functioning circuit

➢ **Software :**

What is going on in Node-Red is that after building our own workspace by dragging the needed nodes such as (MQTT in/out , INJECT node, Function node, SQLITE node, and Debug) as showen below (figure 26):

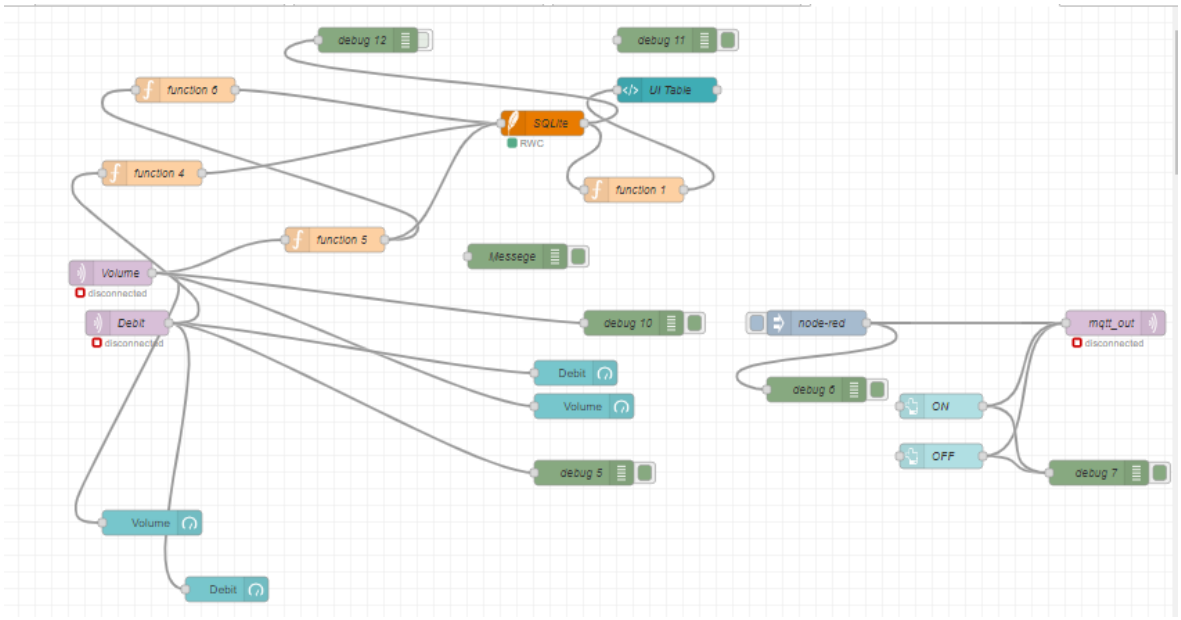


Figure III.40: Node-Red

So basically after creating our database in SQLITE , we had to build a similar node in order to stack our incoming data from the MQTT in node .We dragged two MQTT in and defined each one of them by Volume and flow rate (debit) and we edited the proprieties based on the subscribed topic coming from the MQTT Broker and of course both topics are different .

After connecting successfully to the broker we started getting our data that contains values of the Flow rate and the Volume with the help of the debug node we will get to see the values obtained like this (figure 27):

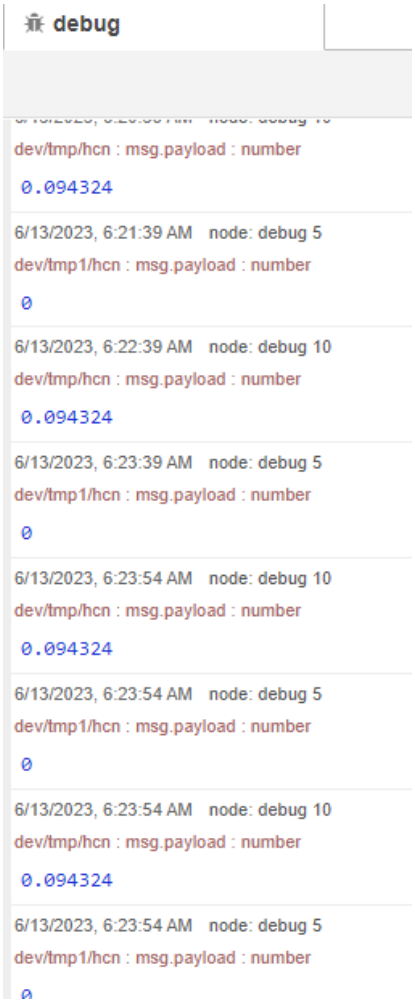


Figure III.41: Values received

Coming to the function node : we got 3 function nodes however the function 4 defines the flow rate coming from the broker , time and the state of our solenoid valve to stack it in our database table with leaving the volume's column empty , and that's where the function 5 gets to to work where it updates the volume's column with the right value coming from the broker , when everything is set in the right place the function 1 stacks each colomn in the database tablethat is in SQLITE node with the right order (Debit, Volume, Time, State) , then converts the time from TimeStamp to the current time and stacks everything we collected in our database Table that will be presented in our UI throught Table UI node .

The results will be represented on a user interface dashboard (figure 28)that is made of two gauges each one defines the Flow rate and Volume , a small table that shows the changed values of the flow rate , volume , time and the state depending on the consumption of these variables .

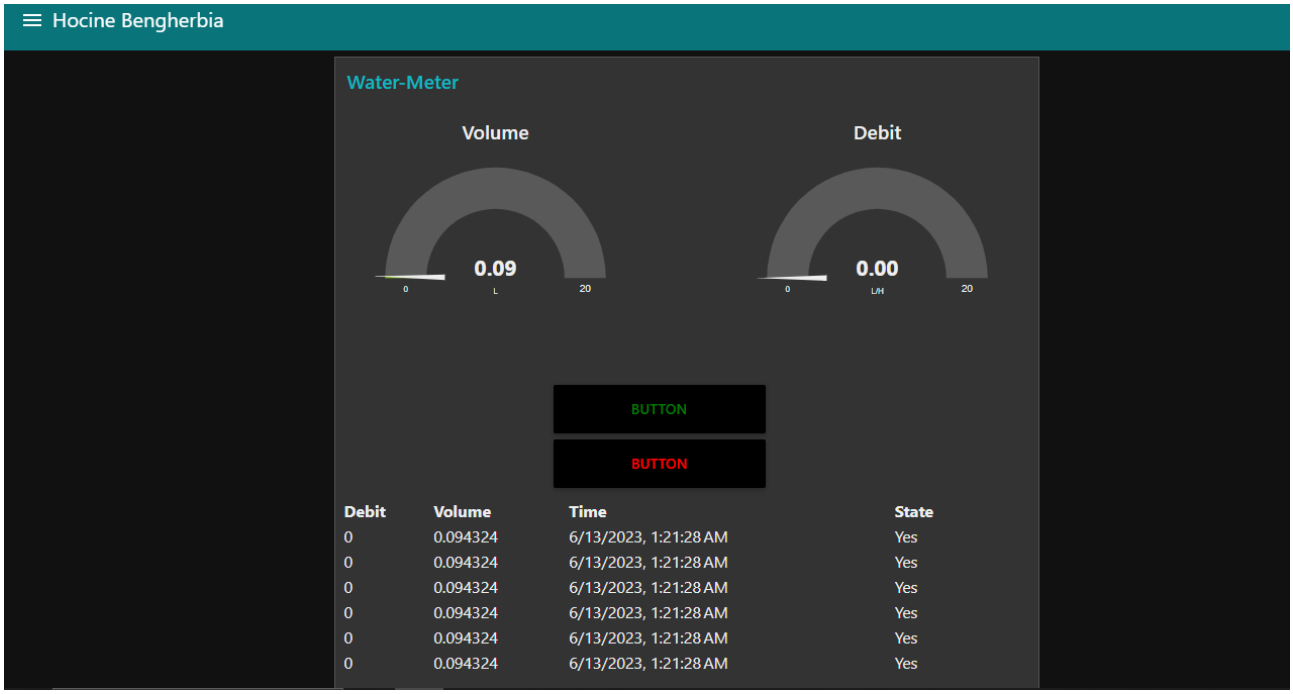


Figure III.42: UI Dashboard

Conclusion :

The successful development of a water metering system that utilizes the STM32 MCU, LoRa module, ESP8266 Wi-Fi module, and MQTT server has yielded promising results in achieving accurate flow rate and volume calculations .The integration of the STM32 MCU, known for its reliability and performance, has played a pivotal role in data processing and control within the water metering system. It has provided the necessary computational power and functionality to ensure accurate calculations and real-time data management, and utilization of the LoRa module has enabled long-range wireless communication, allowing seamless transmission of water metering data to the ESP8266 Wi-Fi module. The LoRa technology's low power consumption characteristics have contributed to the system's overall energy efficiency . The ESP8266 Wi-Fi module, acting as a bridge between the LoRa network and the MQTT server, has facilitated the smooth transfer of collected data to the MQTT server for further processing and storage. This integration ensures secure and reliable data storage in a database, enabling easy access and analysis of the water consumption information.

GENERAL CONCLUSION :

Our project involved creating a water flow meter using an STM32 microcontroller (MCU) and a LoRa SX1278 module to transmit data to a gateway. The STM32 MCU goes into a deep sleep mode when no water is detected, consuming 3 - 5 mA, and wakes up to a wake-up mode that consumes 50 mA.

This project aims to efficiently monitor water flow by utilizing low-power modes in the STM32 MCU. This approach helps conserve energy and extend the battery life of the system. The LoRa SX1278 module provides long-range wireless communication, allowing the water flow data to be transmitted to a gateway.

Overall, This project demonstrates an effective solution for water flow monitoring with low power consumption. By incorporating sleep modes and a power-efficient MCU, we have optimized energy usage while maintaining reliable data transmission through the LoRa module.

By utilizing LoRa technology, this project enables long-range communication between the water flow meter and the gateway, making it suitable for applications where distance is a factor. The integration with MQTT allows seamless data transfer to your server, facilitating real-time monitoring and analysis of the water flow data.

Long-range communication: LoRa enables communication over extended distances, ideal for IoT applications that require wide coverage.

MQTT integration: Integrating with an MQTT broker server provides a standardized and scalable approach for data transmission between devices.

Flexibility: The ESP8266 Mini's Wi-Fi capability allows seamless integration with existing networks and cloud services.

Creating a LoRa gateway with ESP8266 Mini and LoRa SX1278 provides a powerful combination of long-range wireless communication and MQTT integration.

This setup opens up possibilities for IoT applications where devices can communicate over a wide area while leveraging the benefits of MQTT-based systems.

The integration of LoRa and MQTT enables seamless data transmission and facilitates connectivity with other IoT devices and applications.

At last, This project successfully combines multiple components to create an end-to-end solution for water flow monitoring, leveraging the STM32 MCU, LoRa communication, and MQTT integration for efficient data collection and transmission.

Bibliography

[1] M. Naimi-Ait-Aoudia and E. Berezowska-Azzag, “Household water consumption in Algiers facing population growth,” 2014.

[2] M. Habi and O. Harrouz, “Domestic water conservation practices in Tlemcen City (Algeria),” *Appl. Water Sci.*, vol. 5, no. 2, pp. 161–169, Jun. 2015, doi: 10.1007/s13201-014-0175-9.

[3] “What Is a Water Meter and How Do I Read One?,” *Fresh Water Systems*. https://www.freshwatersystems.com/blogs/blog/what-is-a-water-meter-and-how-do-i-read-one (accessed Jun. 12, 2023).

[4] E. Orenstein, “The Evolution of Water Metering - Impact on Water Use & Utility Revenue.” https://blog.bermad.com/waterworks/the-evolution-of-water-metering (accessed Jun. 12, 2023).

[5] S. S. Mammen, “A quick guide on why you should use water metering,” *Housing News*, Aug. 05, 2020. https://housing.com/news/water-metering-advantages/ (accessed Jun. 12, 2023).

[6] B. Campbell, “What is Water Metering?,” *Wastewater Digest*, May 18, 2022. https://www.wwdmag.com/editorial-topical/what-is-articles/article/11004163/what-is-water-metering (accessed Jun. 13, 2023).

[7] admin, “How to Read your Water Meter,” *Flume*, Aug. 02, 2019. https://flumewater.com/how-to-read-your-water-meter/ (accessed Jun. 12, 2023).

[8] “Water Flow Sensor | Seeed Studio Wiki,” Feb. 03, 2022. https://wiki.seeedstudio.com/Water-Flow-Sensor/ (accessed Jun. 12, 2023).

[9] “80V 10W DC Micro Hydro Generator Tap Water Flow Hydraulic DIY (OEM),” *Hellas Digital*. https://www.hellasdigital.gr/electronics/sensors/hydro-generator/80v-10w-dc-micro-hydro-generator-tap-water-flow-hydraulic-diy-oem/ (accessed Jun. 13, 2023).

[10] “G3/4 Hall Effect Liquid Water Flow Sensor Switch Flowmeter Meter 2.5-30L/min | Walmart Canada,” *Walmart.ca*. https://www.walmart.ca/en/ip/G3-4-Hall-Effect-Liquid-Water-Flow-Sensor-Switch-Flowmeter-Meter-2-5-30L-min/PRD3ZGSHI5W23DW (accessed Jun. 13, 2023).

[11] “What are LoRa and LoRaWAN?,” *The Things Network*. https://www.thethingsnetwork.org/docs/lorawan/what-is-lorawan/ (accessed Jun. 13, 2023).

[12] “LoRa - Key Features and Advantages,” *mutelcor*. https://mutelcor.com/lora/ (accessed Jun. 13, 2023).

[13] A. Jain, “How does LoRa modulation enable long-range communication?,” *Engineers Garage*. https://www.engineersgarage.com/how-does-lora-modulation-enable-long-range-communication/ (accessed Jun. 13, 2023).

[14] “NRF24 VS LoRa For Wireless Communication for IoT Devices - JFrog Connect,” *JFrog*, Aug. 15, 2021. https://jfrog.com/connect/post/nrf24-vs-lora-for-wireless-communication-between-iot-devices/ (accessed Jun. 13, 2023).

[15] “What is a LoRaWAN Gateway?,” *Wyld Networks*. https://wyldnetworks.com/blog/what-is-a-lorawan-gateway (accessed Jun. 13, 2023).

[16] Henry, “Introduction to STM32 Microcontroller,” *The Engineering Knowledge*, Jan. 29, 2021. https://www.theengineeringknowledge.com/introduction-to-stm32-microcontroller/ (accessed Jun. 13, 2023).

[17] “STM32CubeIDE - Integrated Development Environment for STM32 - STMicroelectronics.” https://www.st.com/en/development-tools/stm32cubeide.html (accessed Jun. 13, 2023).

[18] I. Hübschmann, “ESP8266 for IoT: A Complete Guide,” *Nabto*, Aug. 10, 2020. https://www.nabto.com/esp8266-for-iot-complete-guide/ (accessed Jun. 13, 2023).

[19] “WeMos - D1 Mini NodeMcu Lua WIFI ESP8266 Development Board,” *Maker Store PTY LTD*. https://www.makerstore.com.au/product/elec-wemos-d1/ (accessed Jun. 14, 2023).
[20] “What is MQTT and How Does it Work?,” *IoT Agenda*. https://www.techtarget.com/iotagenda/definition/MQTT-MQ-Telemetry-Transport (accessed Jun. 13, 2023).
[21] I. Craggs, “MQTT Vs. HTTP: Which protocol is the best for IoT or IIoT?” https://www.hivemq.com/article/mqtt-vs-http-protocols-in-iot-iiot/ (accessed Jun. 13, 2023).
[22] “HiveMQ.” https://www.thethingsindustries.com/docs/integrations/mqtt/mqtt-clients/hivemq/ (accessed Jun. 13, 2023).
[23] “Node-RED.” https://nodered.org/ (accessed Jun. 13, 2023).
[24] “Node-RED and IoT | Practical Node-RED Programming,” *Packt*. https://subscription.packtpub.com/book/web-development/9781800201590/2/ch02lvl1sec06/node-red-and-iot (accessed Jun. 13, 2023).
[25] “What is the Internet of Things (IoT)?” https://www.oracle.com/internet-of-things/what-is-iot/ (accessed Jun. 13, 2023).
[26] “What is a database?” https://www.oracle.com/database/what-is-database/ (accessed Jun. 13, 2023).