Introductions

In most countries of the world, the flood caused large damage and involved it in significant amounts of loss to individuals and its properties. Even so, we can forecast rain or to track the path of the storm exactly from satellite image, the need to have real-time monitored data is important to make a rational decision on the actions needed to be taken and have good flood response operating system to manage all the movement of the floods.

Every year, it causes lives and damage to infrastructure, agricultural production and serious damage to local economic development. In recent years, high rates of removal, extensive area clearing, and communal forest intrusion, in addition to ineffective protection of cut-over forests, have accelerated the increasing need to address flood risk in the urban and even the rural region. *Due to the significant reduction of forests, peak floods can travel faster to generate higher risk within a short period of lead in populated areas. Problems also include heavy local rainfall, storms may result to typhoon and inadequate drainage, which is causing floods.* Where in this research study is made to track the map of the affected areas, to inform the individual user and they can manage their plan far from possible flood risk. Using this system, the user can view flood movement and status using red color coding of the areas where high flood levels occur caused by heavy rainfall or result of natural disaster.

Many studies have been conducted on research and development of early mobile warning systems communication and information-based technologies. This is mobile communication technologies are well known devices and machines are growing rapidly in the industry and the global world. Previous research was developed using mobile based technologies such as floods monitoring systems and early warning based on SMS

Where this research using the Internet of Things technology device for water level monitoring with the help of cellular communication, along with data analysis, can help monitor floods and provide information that can predict floods, in addition this technology requires significant sources of data processing due to the large amount of incoming data, which can result in delays when real-time situations are measured. To develop an effective flood monitoring system, Internet of Things technology can be used in conjunction with sensors and other technologies, such as machine learning and artificial intelligence techniques. Also, ultrasonic sensors and cellular transmission technologies can be positioned to ensure adequate transmission rates and to prevent data loss by implementing optimized telemetry and lightweight structure data methods to reduce the load of incoming data.

Methods

Message Queuing Telemetry Transport is an open machine-to-machine connection protection designed specifically to implement Internet of Things solutions, low data and bandwidth consumption. Message Queuing Telemetry Transport is ideal for devices that use low power technology and have low data transmission requirements. It is designed to use a Message Queuing Telemetry Transport Protocol to transmit data through a SIM communication module integrated into the main board. The device sends strings of data using AT-Command from the Microcontroller Unit (MCU) to the SIM module to communicate with an external Message Queuing Telemetry Transport broker.

Eclipse Mosquitto ™ is an open source messaging system that interacts with multiple devices using the MQTT protocol, containing a low line of code. It is installed in a server-side environment that manages all subscription and labor topics. The MQTT protocol implements a basic structure, called " topic ", in which messages are sent. The server-side environment is implementing a data capture method that uses a Mosquitto broker to receive messages from the fixed and mobile node. These messages are stored in a NoSQL MongoDB database environment. Due to the combined use of the MQTT protocol and a non-relational database architecture, the environment can be implemented across a variety of different clients.

MongoDB is a database model and is based on JavaScript Object Notation formats of files stored in data collections. Because each document is made up of different structures, any data log has the ability to have different properties. The server implements MongoDB to save readings of a device because different sensors can be connected to the datalogger by using different log structures in its databases.

Because it uses a wireless delivery environment, it can be vulnerable to sniffing technologies or external interference. Considering possible security threats, the system includes security techniques embedded in the delivery protocol, allowing the MQTT protocol to set encrypted server-side credentials for the server per device and user. Using this same method proposes a message-based security token contained in the message header to provide a unique message type contained in code '0000', creating a "spublish" method with encrypted. This security measure for the MQTT protocol considers how to register different devices by assigning them a unique identity through a Universal Resource Identifier. The security methods used by this environment allow it to set encrypted credentials based on Secure Sockets Layer and Transport Layer Security (SSL / TLS) certificates, using the Mosquitto broker's built in pre-shared key encryption (PSK).

Also, Non-relational database-related security procedures, which enable the system to set certificates and client connections x.509 based on the certificate. These dimensions ensure public key infrastructure (PKI) formats and certificates to establish valid database connections, whether they are made directly from devices, data acquisition platforms or otherwise more certified clients. And some security issues regarding MongoDB have already been reported, highlighting the lack of authentication in this database engine during shared mode. To resolve the issue, the environment can set different layers of security, such as SSL / TLS operating system certificates, server access (SSH) and authentication, including protocol.

The designed to combine each sensor according to implementation requirements. And one of these implementations is a streamlined module that senses different variables, such as flow speed, width and water height by using an ultrasonic remote sensor as shown in Figure 1. It can transmit water data through a cellular communication module to monitor variables in a server-side web interface. The fixed node is physically made up of five different devices. It is a 32-bit microcontroller unit, a 3G cellular modem electronic board, a regulated power supply, a solar charge controller and a 12 V 80 Ah battery, as shown in Figure 2. The first prototype of the fixed node contains the right materials and components can be placed in a remote location near a place of environment detecting the water level variations. It was translated into an IP65 enclosure containing an electronic system with exterior designed glands and a battery that allowed the system to run continuously even it is rainfall or without direct sunlight for four days. This means that the backup battery will allow the device when the time is up, or at night, until it receives enough sunlight to recharge. Another is maintaining the battery life energy management system using a 75 W polycrystalline 12 V solar panel that allows the device to recharge before it runs out.

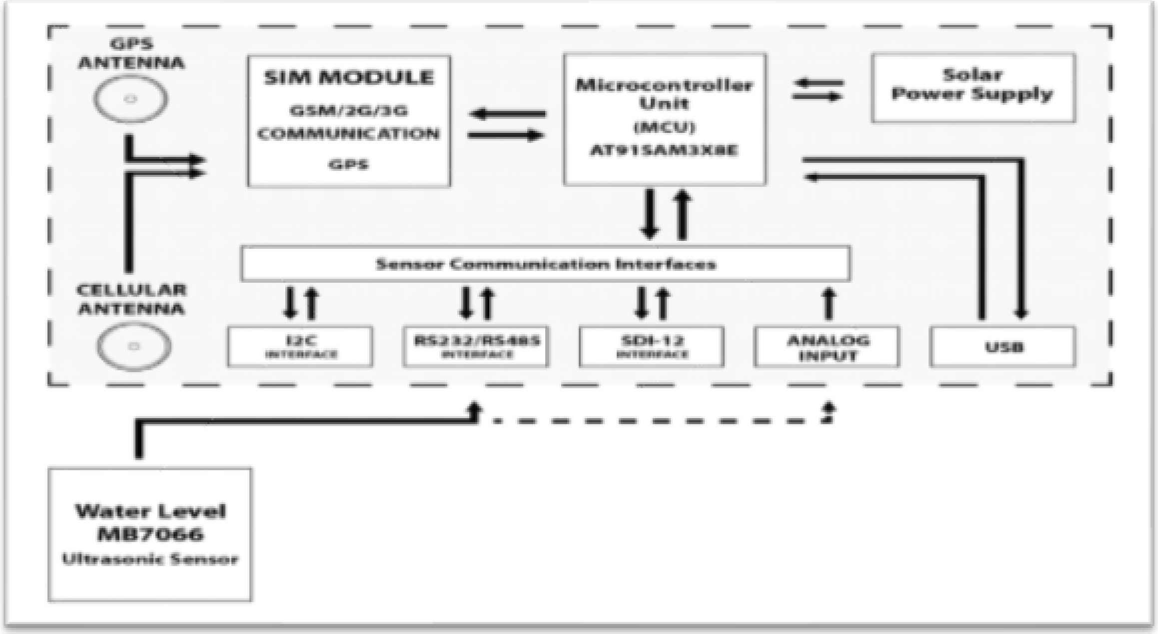

Figure 1: Components of fixed node

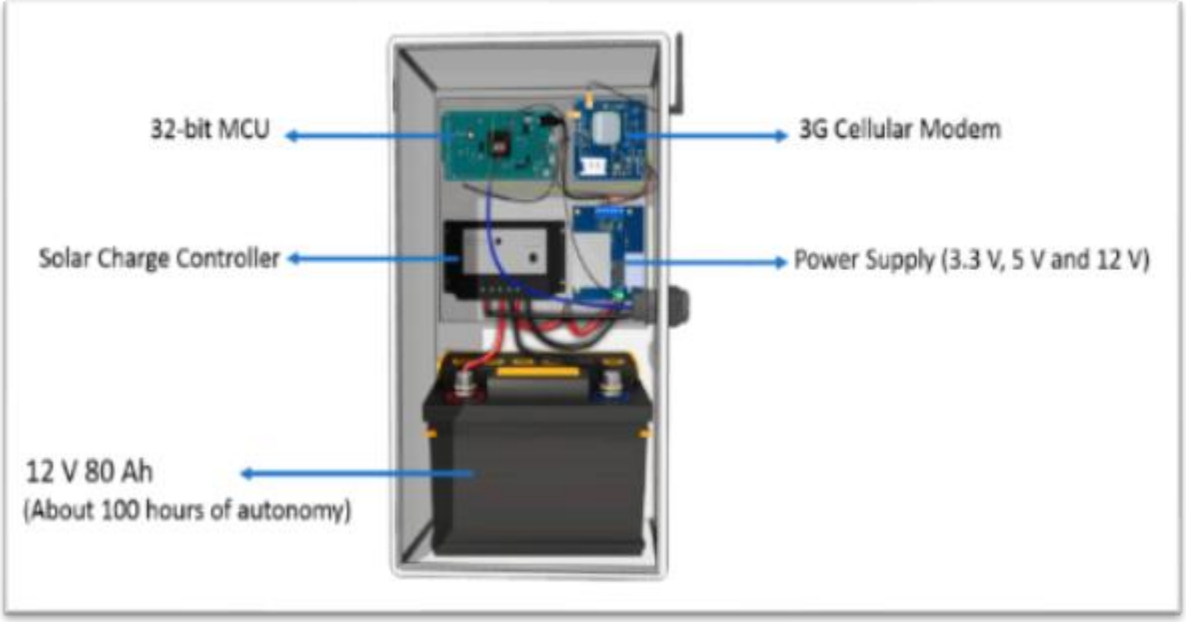

Figure 2: Fixed node 3D model.

MB7066 water level sensor shown in Figure 3 is used to measure the distance between the surface of the water and the location of the sensor, which is processed inside the microcontroller and encoded in the JSON structure, along with GPS node coordinates, the node timestamp. Actual reading and an identity string. This data is then sent to the server via the 3G cellular network and packaged into the NoSQL database to perform additional calculations that can be used in a flood forecasting network.

Figure 3: MaxSonar MB7066 Ultrasonic sensor.

The Node of a mobile implementation that meets the needs of a drifter monitoring device is also designed. This node includes a GPS module that captures location, time and speed variables. It also contains a micro SD card slot to capture scalable data, as shown in Figure 4. The drifter node is sealed inside a waterproof holding magnetic switch inside, enabling the device to initiate log data. This data is stored on the micro SD card and can be analyzed by data acquisition platform. There is no external communication through a cellular or other protocol built at this time.

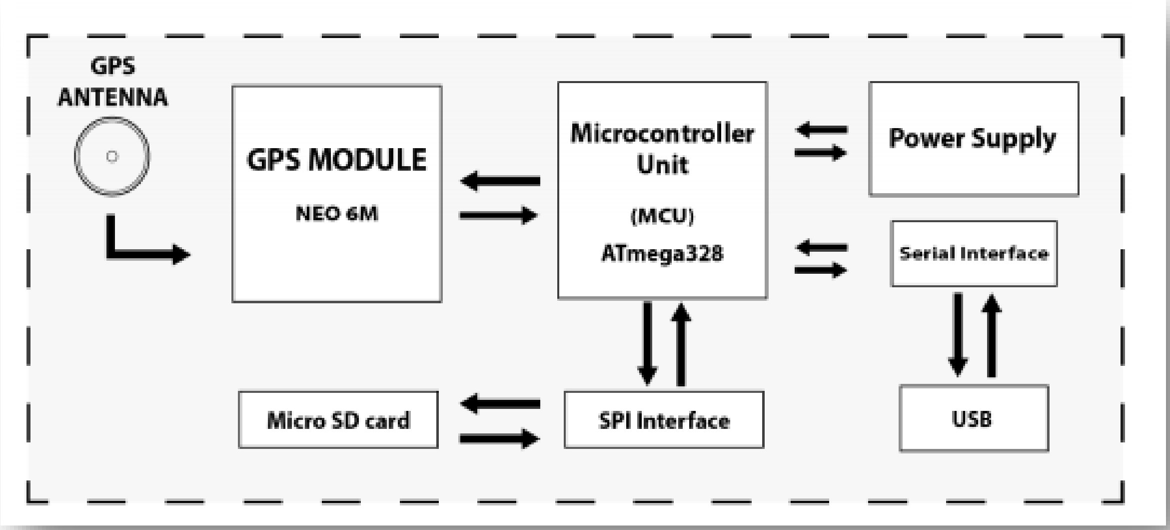

Figure 4: Mobile node components

Hardware components pictured above are shown in Figure 5. The electronics are sealed in a spherical lighweight plastic container that uses a polypropylene rubber like material used frequently in the automatic industry. The container has a diameter of 10cm and is shown in Figure 6. The main board of the device is 8 cm long and 5.1 cm high.

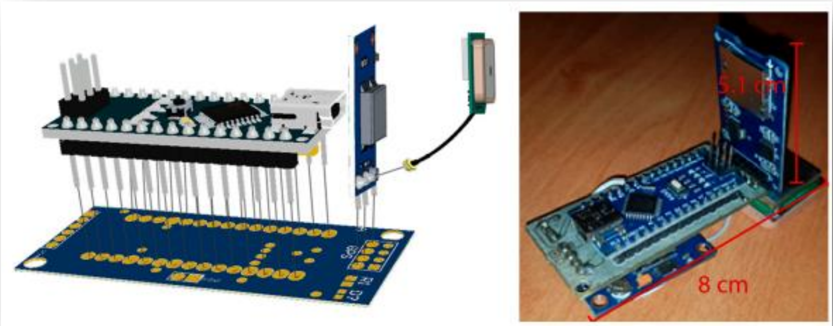

Figure 5: 3D model and physical design Mobile node.

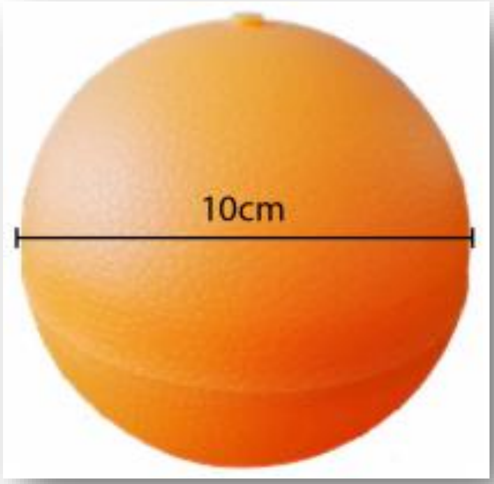

Figure 6: Lightweight plastic container Mobile node.

The acquisition data environment is integrated through a hardware device, an MQTT broker and a NoSQL document-based data structure, as shown in Figure 7. The datalogger get the information from environmental variables and transmits it via a 2G / 3G cellular network, using AT commands and the MQTT protocol, to a Fedora 26 Linux server, with installed MQTT Mosquitto broker. This broker receives all messages published on the "node" topic, while a Node.js background script saves data using a MongoDB document structure. And the information saved to a MongoDB database, it can obtain using an online monitoring web platform that listens to a WebSocket that receives MQTT messages. Because of the open source approach, data acquisition platform is compatible with some devices, software development kits and other Internet of Things projects. Through the use of JSON format, different devices can publish the useful weather data to the system, regardless of the hardware used. Also, devices can subscribe to get useful information on various graphic representations, as illustrated in Figure 8.

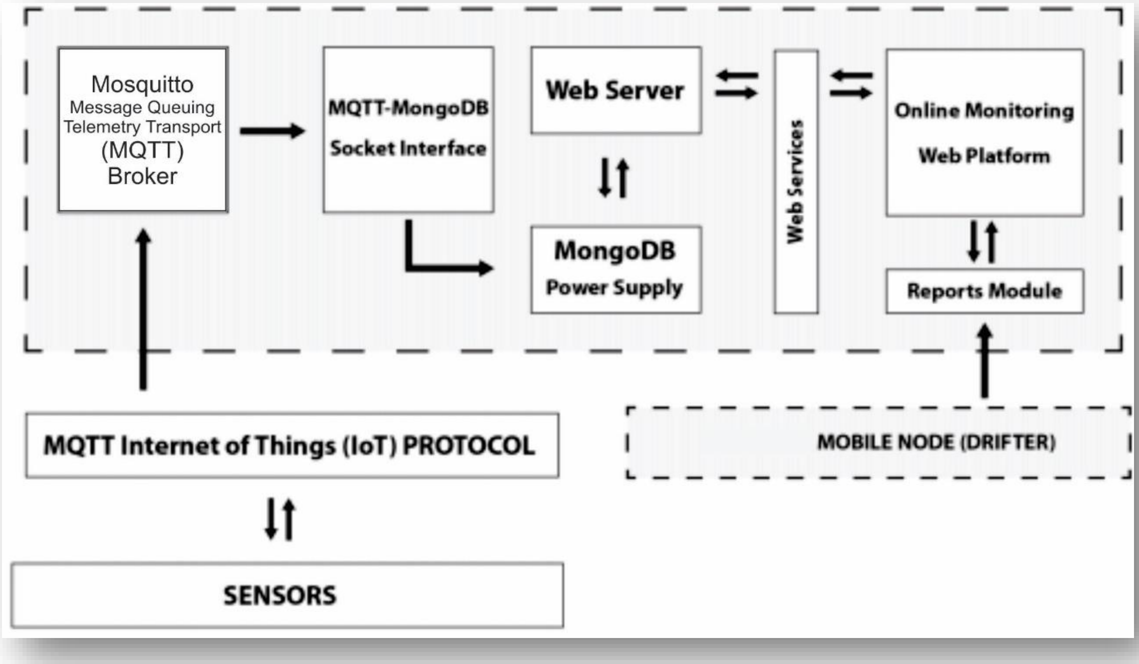

Figure 7: Architecture Data Acquisition.

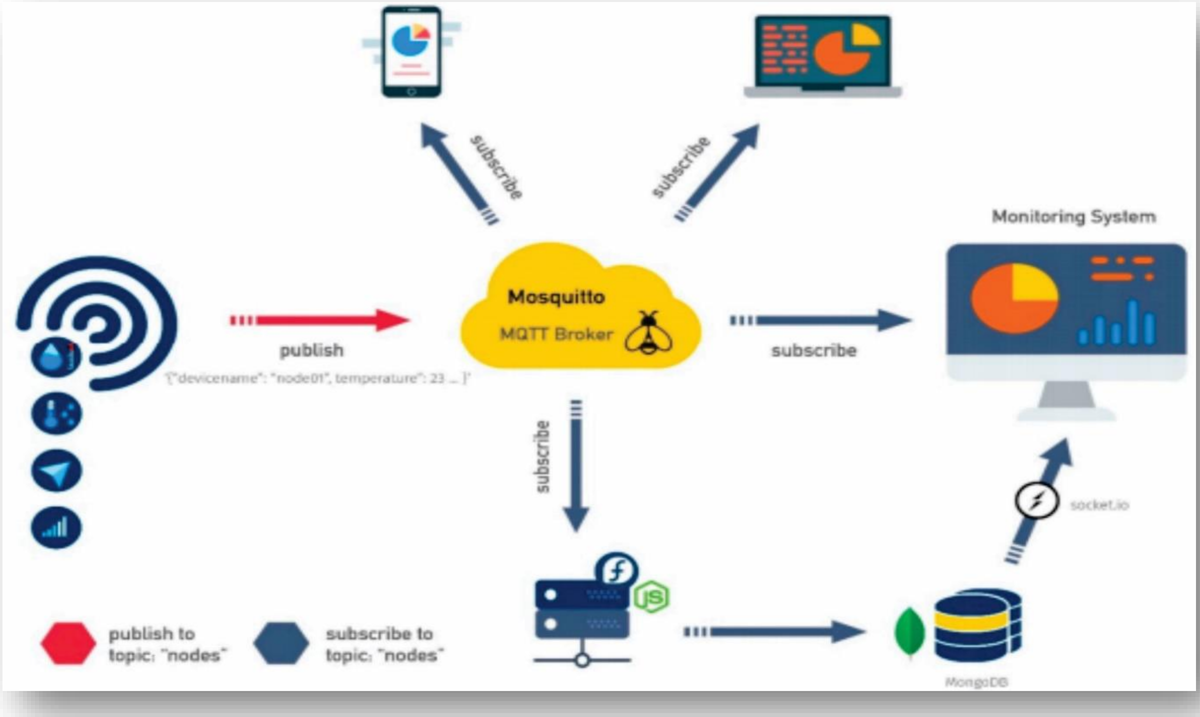

Figure 8: Scheme of Web platform.

Web platform includes a dashboard to display extracted data from all fixed nodes. This dashboard includes the latest notifications received, various related charts that can be described or hidden depending on the sensors connected to the devices, and a map where nodes are located as markers as shown in Figure 9.

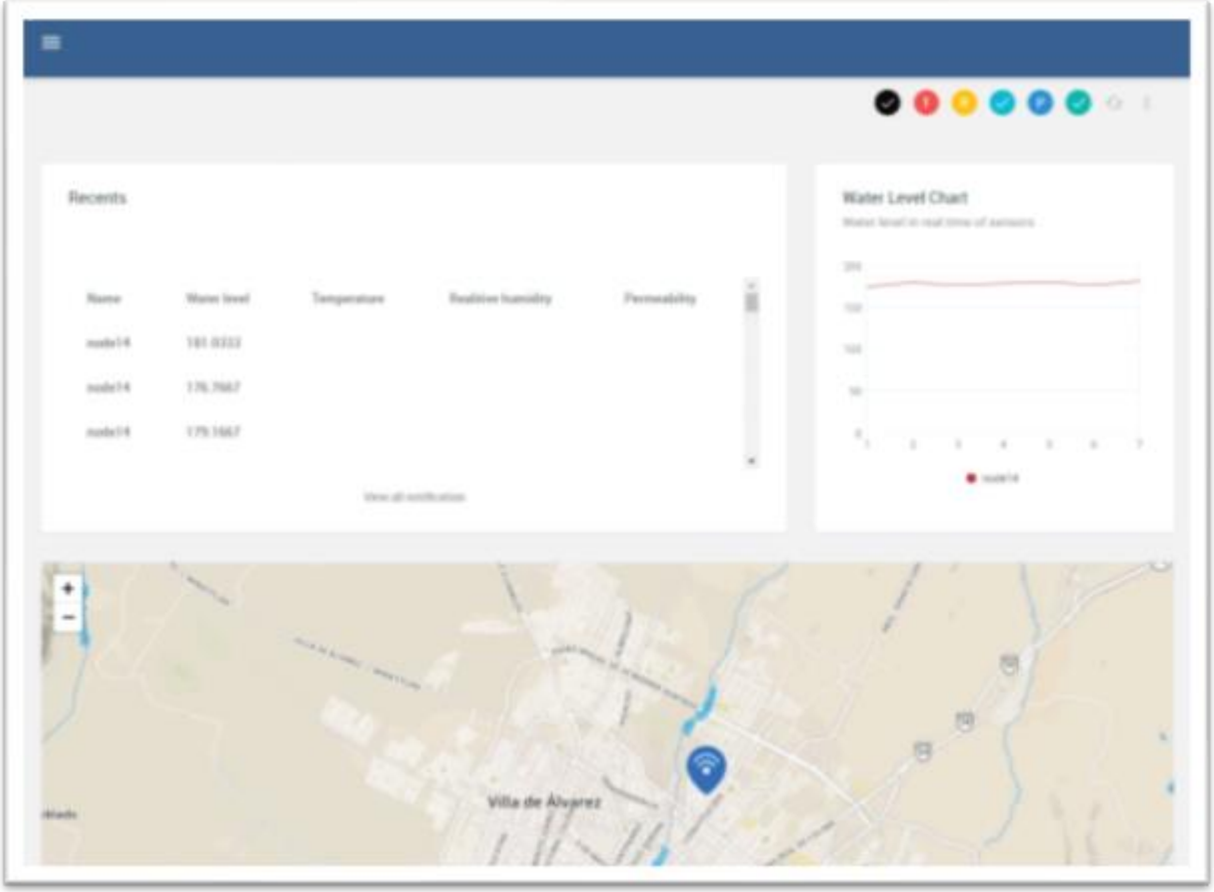

Figure 9: Main view of Cloud platform.

The fixed devices are registered in a dedicated section of the web platform, they have configuration parameters such as node names and sensor features, as shown in Figure 10. Where all registered devices can manage the section shown in Figure 11, which describes the features of registered nodes.

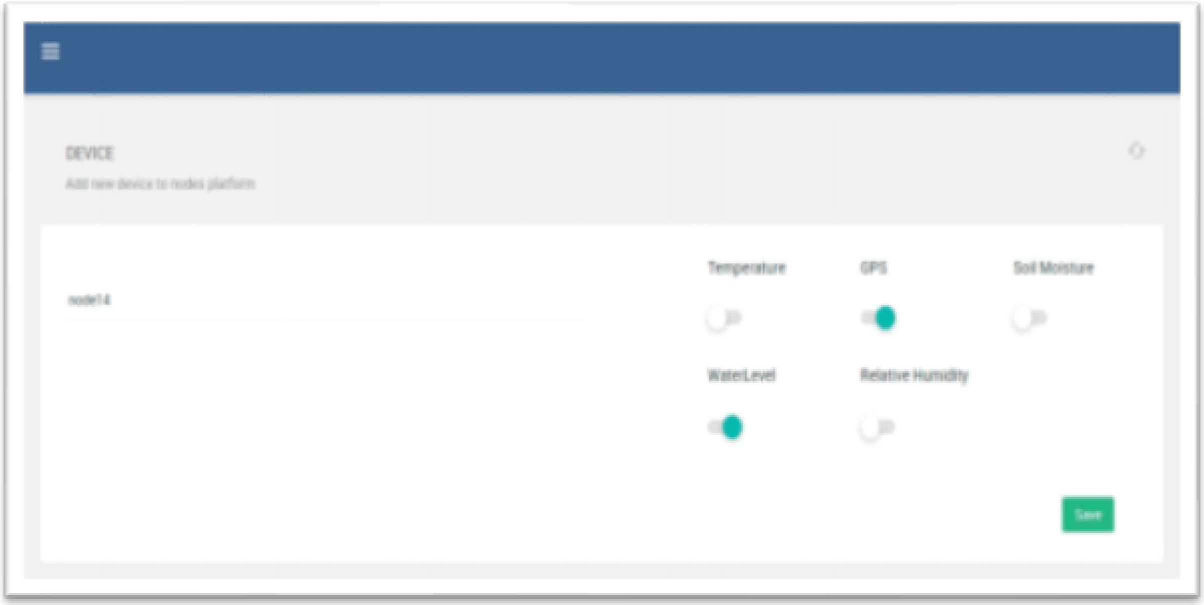

Figure 10: View of the Device registration.

The fixed devices are registered in a dedicated section of the web platform, they have configuration parameters such as node names and sensor features, as shown in Figure 10. Where all registered devices can manage the section shown in Figure 11, which describes the features of registered nodes.

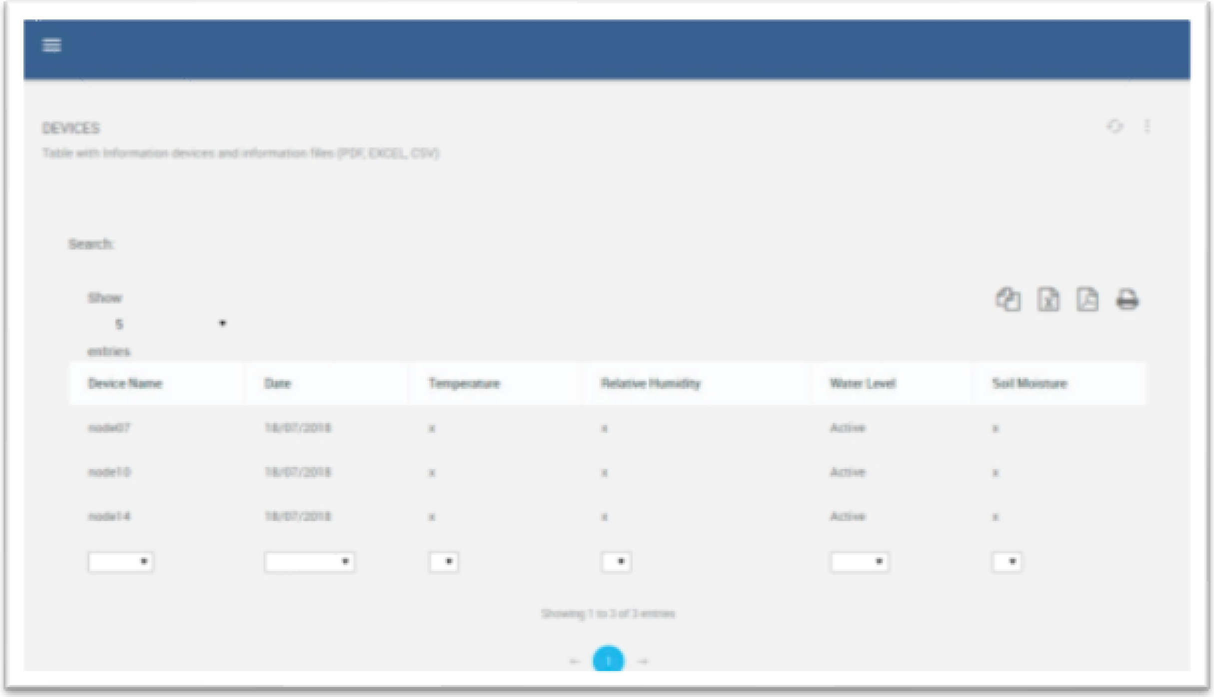

Figure 11: Management view of the Device.

The data obtained from all nodes can be downloaded as reports in several formats such as PDF, EXCEL, directly printed or CSV, and is suitable for further analysis for use in flood control systems. Moreover, the data can be filtered by inbuilt queries such as date, node or variable, as illustrated in Figure 12.

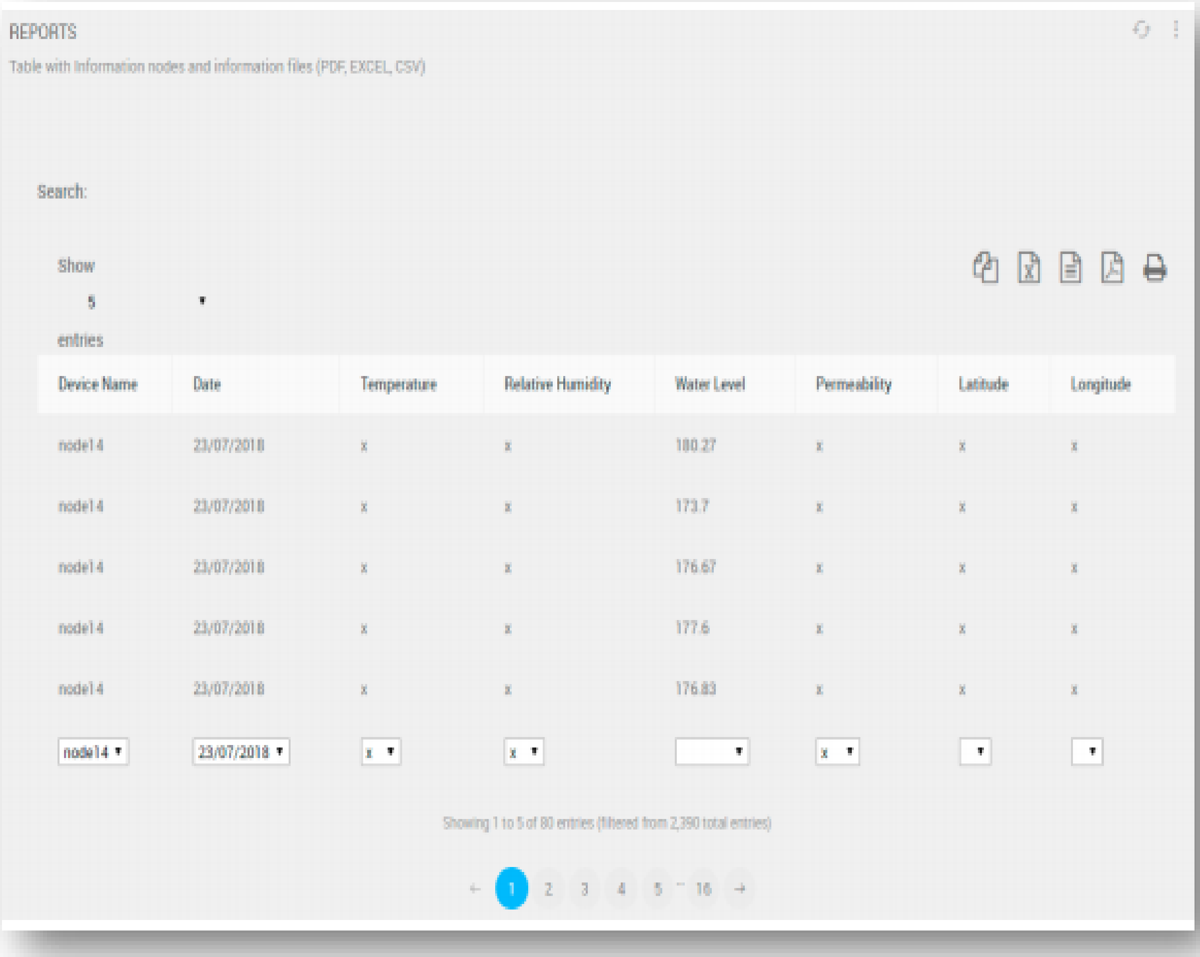

Figure 12: View of Report exportation.

Result

Figures 13 and 15 show that data from three different sources were collected during the measurement period and Figure 15 provides a graph of reports that the percentage of relative humidity is measured in depth (cm) of water source, while Figure 14 showing the temperature variation during measurement and Figure 15 provides a graph that reports the percentage of relative humidity in the period of measurement.

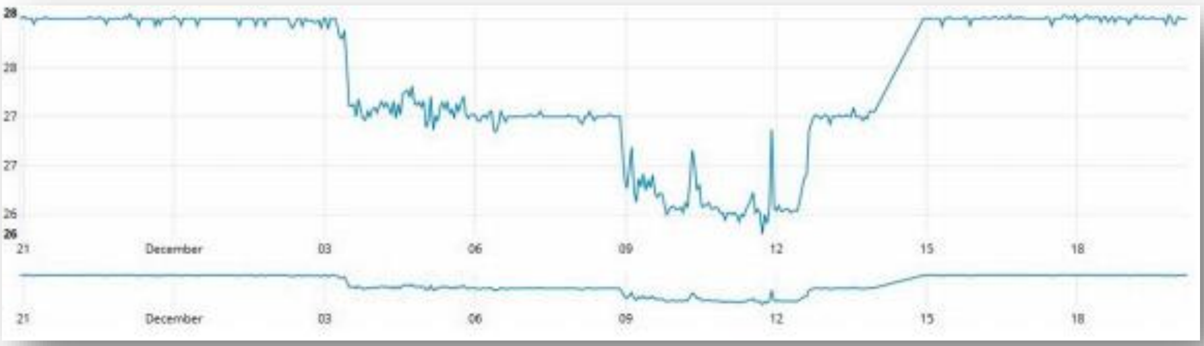

Figure 13: Measurements over time of Water depth.

Figure 14: Measurements over time of Environmental temperature.

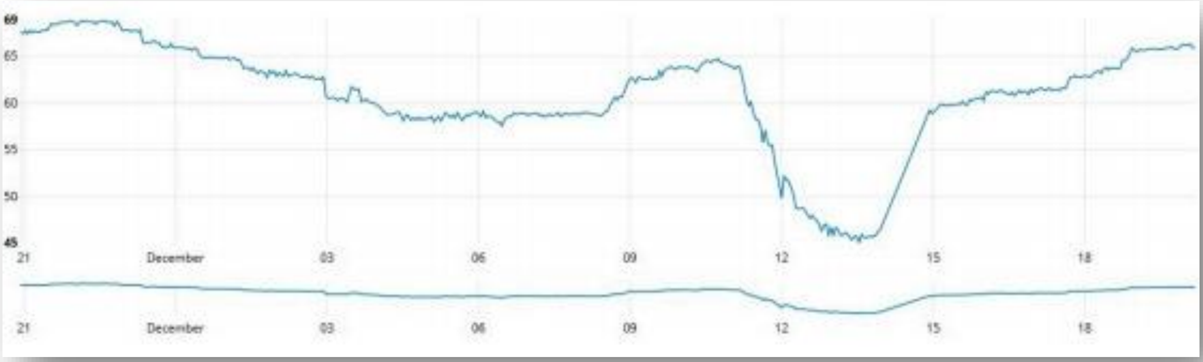

Figure 15: Measurements over time of Relative humidity.

The fixed node is attached to a tree at the selected location by the use of stainless-steel clamps. In this position, two cables were laid out on different parts of the system. First cable was route to a polycrystalline solar panel mounted on a 4 m high pole, and the second is connected to the MB7066 Ultrasonic sensor, which is tightly fixed to the ravine with a steel wire at a height of 6 m away from the IP65 Nema Box. The expansion site can be seen in Figures 16 and 17, where housing can be found containing electronics and other infrastructure components.

Figure 16: Fixed node deployment.

Figure 17: Ultrasonic Sensor and Solar Panel Positioned.

After establishing the fixed node, measurements were taken to assess the operation and performance of the system, the data acquisition and the communication platforms. As seen in Figure 18, data were obtained after 1024 min of normal operation, at which 1024 water levels were recorded, at a rate of one sample per minute. During this time, the server registered a standard deviation of ± 1.75 cm in the water level of a calm stream. Data is transmitted using a 3G cellular connection through the MQTT communication protocol on the data acquisition platform installed on a Fedora Linux Server.

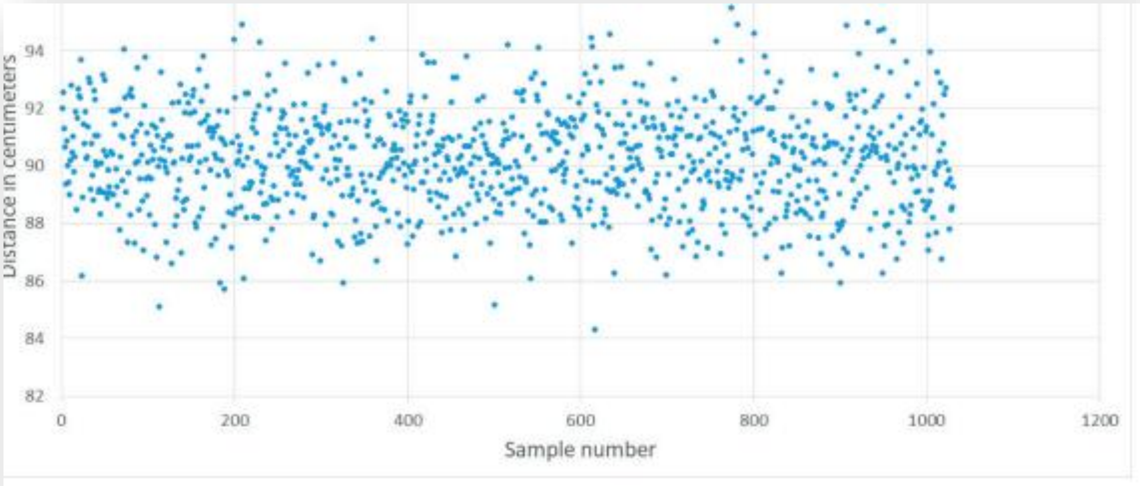

Figure 18: Deployed node Retrieved data.

There is evaluating the trial of developed mobile GPS-based nodes to record their behavior and accuracy while recording GPS positions used in the analysis to determine flow rates were also performed. Data of single downstream GPS drifter track from tests conducted on a relatively shown in Figure 19, where the mobile node was released upstream to record 280 GPS positions during operation. Drift is from right to left in the image with total travel distance of around 194 m. These examples contain data about the path of travel using GPS coordinates. GPS position data was used to calculate the average speed and distance traveled, and to calculate the slope, among other useful data.

Figure 19: GPS sample of retrieval Mobile node.

Drifter test data is obtained on the generated data capture platform, using an input of a JSON file generated data drifting in the device path. Samples of aggregated data analysis charts and reports generated from the JSON data format of drifter data are shown in Figures 20 and 21.

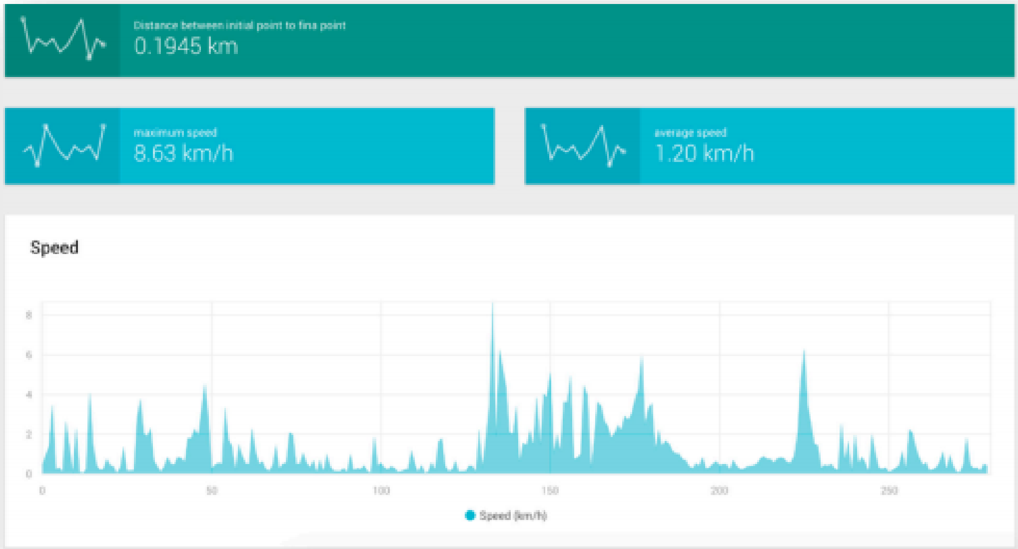

Figure 22: Speed-related data of Mobile node.

Discussion

In developing, testing and implementing The Internet of Things devices and the data acquisition environment, we have identified three important aspects that require further discussion. The first important point is the acquisition rate. As this real-time device is developed, it is important to measure and store variables using as little time as possible to allow time to establish Internet connection communications between the hardware and the web acquisition web platform. It should be low enough to mitigate any significant data loss that could compromise the system's ability to monitor water levels. It is also worth mentioning that the system, as a whole, should ideally consist of multiple fixed nodes to ensure the operation of a network as a source of useful information for forecasting model. Second, the communication signals of 2G / 3G cellular networks may suffer from interference and strong signal variability in remote locations, increasing the likelihood of losing valuable data. One solution to this common problem is to develop mechanisms such as a data pool or data files that store most of this data in device memory so that when an Internet connection is lost, a transmission can be transmitted, data at a more opportune moment. Implementing the same storage device used on mobile nodes, it is possible to maintain data on connection losses. Similarly, it is useful to add a connection of real-time mobile drifter nodes to monitor their location after storm conditions or in the event of a loss of sensor. In this case it can be dangerous to get them if the water levels are at least adequate or the flow is flowing fast. Third is the plans for future improvement include enhancing mobile node capabilities by making them compatible with commercially available weather stations using the SDI-12 protocol, and establishing mixed communications, relationships between drifters and fixed devices to ensure better data transmission on the web-based platform acquisition platform.