**CHAPTER 3**

**Methodology**

**Introduction**

This chapter captures the detailed combination of the best practices, procedures, rules and guidelines observed to ensure completion and proper functioning of the project. Let’s discuss what a microcontroller is.

Microcontrollers are found in almost all modern-day electronic devices. Washing machines, CD/DVD players, digital watches, mobile phones, microwave ovens are devices that function with the aid of microcontrollers. These devices can be said to be similar to a personal computer. While personal computers interact with humans, microcontrollers interact with other machines. Microcontrollers were developed to make processes automated and are widely used in embedded systems.

A microcontroller, from its name, is a micro integrated circuit chip (IC) that is used to control other devices and machines. It is simply a microprocessor with a Random-Access Memory (RAM), a Read Only Memory (ROM) and Input Output ports.

**Development Cycle**

Suitable for a number of reasons, the iterative design model was chosen for the project. The implementation of this model was suitable due to challenges encountered and the model fairly improved the project as better working plans were produced.

The model enables the development of working functionality easily and quickly in the life cycle as it assumes subsequent release of the models adds function to the previous release. The iterative design model can be divided into four major phases. They are requirement management, Design and development, Testing, Implementation

**Requirement management**

This phase involves capturing of requirements, putting them under specific categories and most importantly articulating the project needs in a formal and precise way. Project objectives may change or new one maybe added for testing purposes.

**Design and Development**

This phase involves establishing the overall system architecture by allocating requirements to hardware systems. Improved version of project is usually produced after each iteration.

**Testing**

In this phase, individual components are tested and again tested as an integrated system. Debugging processes are identified for further iteration of the improved version produced. Thus, enabling easier bug fixing in the next iteration.

**Implementation**

From the information gathered during the requirement management and the design phases, the actual work is established and a final outcome is produced in this phase. Programming and debugging are considered as core activities or tasks.

Build 1

Design & Development

Implementation

Testing

Build 2

Requirement

Design & Development

Testing

Implementation

Build 3

Design & Development

Implementation

Testing

Figure 1: Interactive Design Modelling

**Model description of project Kits**

* An Arduino mega Board.
* Breadboard
* (16 \* 2) LCD display.
* The 2 in one Temperature and PH sensor- E-201-C.
* The Turbidity Sensor- SEN0189.
* An ultrasonic Sensor.
* 9V battery (Energizer Battery).

**Arduino Mega Board**

The Arduino MEGA ADK is a microcontroller board based on the ATmega2560. It has a USB host interface to connect with Android based phones, based on the MAX3421e IC. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button.

**Breadboard**

A breadboard is a construction base for prototyping of electronics. A modern solderless breadboard consists of a perforated block of plastic with numerous tin-plated phosphor bronze or nickel silver alloy spring clips under the perforations. The clips are often called tie points or contact points. The number of tie points is often given in the specification of the breadboard. Interconnecting wires and the leads of discrete components (such as capacitors, resistors, and inductors) can be inserted into the remaining free holes to complete the circuit. Where ICs are not used, discrete components and connecting wires may use any of the holes. The edge of the board has male and female notches so boards can be clipped together to form a large breadboard.

**16x2 LCD**

This project used LED and LCD 16x2 as a display system. The LEDs will turn on based on the amount of water level that was sensed by the sensor. The LCD 16x2 has two rows and each row can display 16 characters. First row will display the current aspect of water quality being measured and the second row displays the value of the data

Figure: 16x2 LCD

**The 2 in one Temperature and PH sensor**

The temperature sensor works great with any microcontroller using a single digital pin, can even connect multiple ones to the same pin. Usable with 3.0-5.0V systems. The analog pH meter, specially designed for Arduino controllers and has built-in simple, convenient and practical connection and features. It has an LED which works as the Power Indicator, a BNC connector and PH2.0 sensor interface. To use it, just connect the pH sensor with BNC connector, and plug the PH2.0 interface into the analog input port

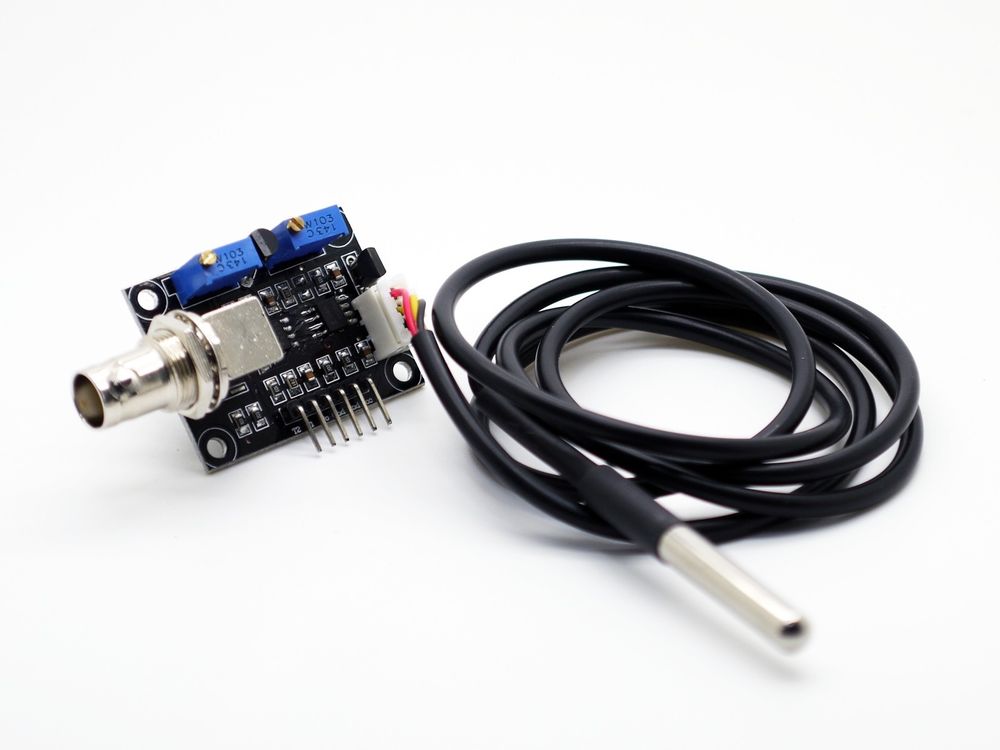


Figure: Temperature and PH sensor from left to right

**Turbidity Sensor**

 turbidity sensor detects water quality by measuring level of turbidity. It is able to detect suspended particles in water by measuring the light transmittance and scattering rate which changes with the amount of total suspended solids (TSS) in water. As the TTS increases, the liquid turbidity level increases. The sensor has both analog and digital signal output modes. You can select the mode since threshold is adjustable in digital signal mode. Turbidity sensors can be used in measurement of water quality in rivers and streams, wastewater and effluent measurements, sediment transport research and laboratory measurements.

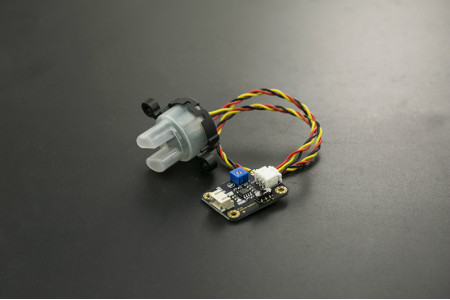
 

Figure: turbidity sensor

**Ultrasonic sensor**

The HC-SR04 Ultrasonic Module has 4 pins, Ground, VCC, Trig and Echo. The Ground and the VCC pins of the module needs to be connected to the Ground and the 5 volts pins on the Arduino Board respectively and the trig and echo pins to any Digital I/O pin on the Arduino Board. It emits an ultrasound at 40 000 Hz which travels through the air and if there is an object or obstacle on its path It will bounce back to the module. Considering the travel time and the speed of the sound you can calculate the distance.

Figure: ultrasonic sensor

**9v Power Supply**

 This project uses a regulated 5V from battery that supplies 9V voltage. 5V power supply connected to main board, sensor board, LCD and relay board.

Figure: 9v Power Supply

**Block Diagram**

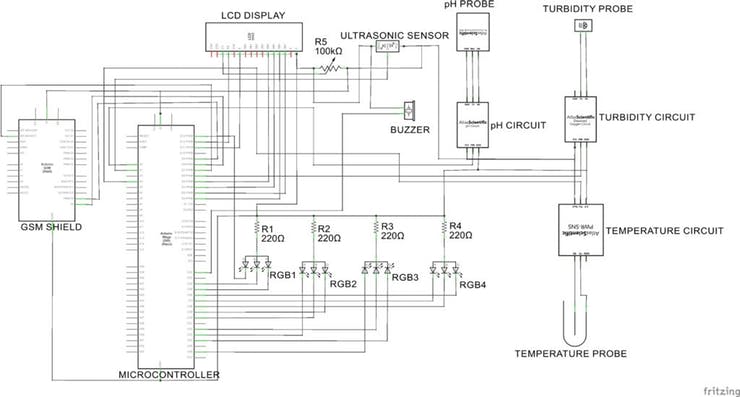


Figure: Block diagram

**Architectural Overview**

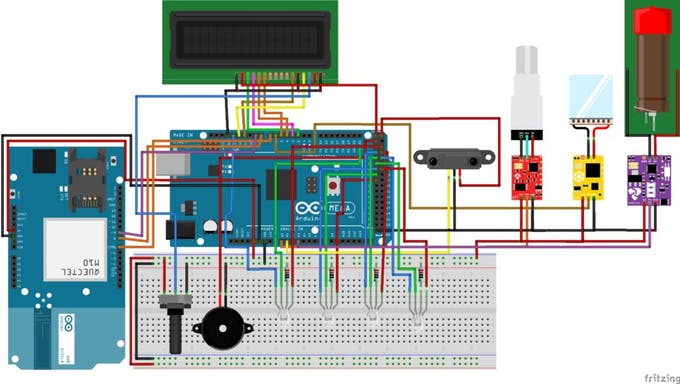


Figure: architectural overview

**Software Implementation**

This was another important part of the project. This aspect captures code written process, in our case, the Arduino IDE was used during the code writing stage for the microcontroller and the necessary hardware components since it is the easiest environment to code with. Again, our web interface was designed using visual studio code was used due to its easy rendering of codes as they are written

**Requirement Gathering and analysis**

During this stage, our main aim is to gain knowledge about the various hardware devices in place. Again, the techniques that would enable us bridge or establish a connection between our Arduino board and the web interface, how to upload our data reading unto our web interface I the require format.

**Functional Requirements**

The project aims to develop a water monitoring system that measures some aspects of water quality. The system should be able to do the following:

* Take correct reading of the set aspects of water quality
* Display the values read
* Upload data values unto our web interface
* Allow users view the data

**Non-functional Requirements**

* User friendly web interface
* Basic interpretation of data over a given period
* Provide information on the importance of the values read
* Web interface should work well on most devices.

**Language Justification**

The choice of language for the project was the C language for a number of reasons. Firstly, the C language was easier to use when programming the microcontroller. Secondly, the team’s comfortable understanding of the language. Again, the availability of assistance when needed. For the web interface, html, CSS and python was used due to availability of tools, assistance and easier understanding of the language’s syntax. Along with python, SQLite was implemented for the database design.

**Design and Development**

This project makes use of the conceptual view of the system and real development system. Many program or codes are gradually improved upon to ensure the proper functioning of hardware components. Versions of the web interface are developed, the easiest and most user friendly is selected as agreed by the team. Sequence diagram along with use case diagrams were considered for the sake of the project.

**System Modeling- Unified Modeling Diagram**

Unified Modeling Language is a standard for visualizing, constructing, specifying and documenting the artifacts of software systems. It was created by the Object Management Group (OMG). In January,1997, UML 1.0 specification draft was proposed on OMG. It was started to captured the behavior of the non-software system and complex software and has now become an OMG standard (Tutorials point, 2017).

UML is a standard notation for the modelling of real-world objects as a first step in developing an object-oriented design methodology. Its notation is derived from and unifies the notations of three object-oriented design and analysis methodologies. UML is actually a combination of several notations: object-oriented design, object modeling technique and object-oriented software engineering. The Unified Modeling Language uses the strengths of these approaches to present a more consistent methodology that is easier to use. UML represents best practices for building and documenting the facets of software and business system modeling. UML is a common language for business analysis, software architects and developers used it to describe, specify, design and document existing or new business process, structure and behavior of artifacts of software systems. UML is a standard modeling language not a software development process. It provides guidance as to the order of a team’s activities, specifies what artifacts should be developed, directs the task of individual developers and the team as a whole offers criterion for monitoring and measuring a project’s products and activities through the use of UML models such as Use Case diagrams, Class Diagrams, Sequence Diagrams and Activity Diagrams

**Sequence Diagram**

The sequence diagram is used to define a sequence of events that are necessary to produce an outcome. Sequence diagrams are the most popular UML artifact for dynamic modelling, which focuses on identifying the behavior within your system. Other dynamic modelling techniques include activity diagramming, communication diagramming, timing diagramming and interaction overview diagramming. Sequence diagrams along with class diagrams and physical data model are in my opinion the most important design level models for modern business application development

Sequence diagrams are necessary in software development processes due to the following:

• They help detect logic, interface and architectural problems very early since an overview of what the system will do in details is provided before the implementation of the components.

• They help describe the expected behavior of the system and form a foundation for the development of system architectures with robust interface

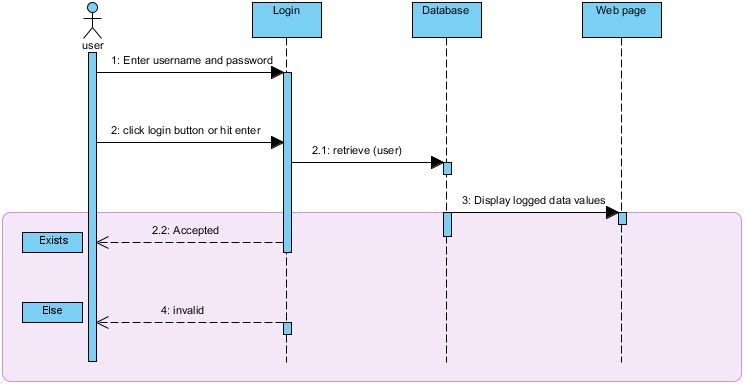


Figure: sequence diagram

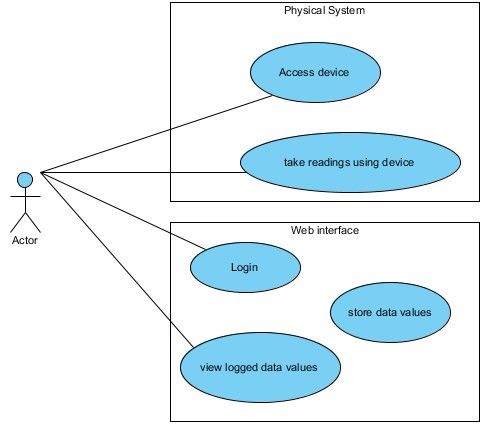
The sequence diagram enables us to understand what happens when the user logs into the web interface. The system checks for the correct credentials in the database before granting access to the data logged.

**Use case diagram**

Use Case diagram is a graphical depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to clarify, identify and organize system requirements. Use Case diagrams are employed in UML a standard notation for the modeling of a real-world objects and systems. System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference or performing a consumer service-oriented task

Use cases also focus on user goals, but the emphasis here is on a user-system interaction rather than the user's task itself. Although their focus is specifically on the interaction between the user (called an "actor'') and a software system, the stress is still very much on the user's perspective, not the system's. The term "scenario" is also used in the context of use cases. In this context, it represents one path through the use case, i.e. one particular set of conditions. This meaning is consistent with the definition given above in that they both represent one specific example of behavior.

User

Figure: Use case diagram

From the figure above, we have the user, the physical system and the web interface and the use cases or the blocks of operations which are actions that the user, the physical system and the web interface can perform along with lines that indicate what each can do.

From the diagram, the system has the following functionalities:

* Read current water: the system checks the water level in the tank.
* Monitoring the system: the system includes a 16x2 LCD that displays the readings visually. The readings include water level values, temperature, PH and turbidity information.
* Storage: the system is able to store the values taken by recording them to a file and also uploading them to a database
* Notification: the system includes a buzzer that sounds when readings are below a set value for each water quality aspect measured. The system includes a web interface that also displays readings recorded over a period of time.
* The user can view the data recorded, interpret it using external tools and retrieve the data file through the web interface.

**Testing and Evaluation**

The system was tested thoroughly throughout the development phase since the iterative design model requires that the debugging process is identified for further iteration. Alpha and Beta testing were use in relation to this system. Alpha testing was done after the design and development phase during which lots of errors were encountered. Through the Beta testing, errors and bugs that were experienced were solved using each step of the iteration process. The web interface was tested using different browser to ensure proper functioning.

**Conclusion**

In this chapter, the iterative design model was selected as methods to be observed to ensure completion and proper functioning of the project. The model assumes that some working functionality can be developed quickly and early in the project life cycle.

The project introduced some forms of system models as UML diagrams, sequence diagrams and Use Cases that were derived during the design process to describe the detail structure, relationships and interactions between users and objects in the system and how system processes are performed

References

1.<https://www.wikiforu.com/2012/10/what-is-microcontroller-how-it-works.html?m=1> , october 15,2012

2. <https://www.arduino.cc/en/Main/ArduinoBoardMegaADK?from=Main.ArduinoBoardADK>,

3. <https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/>