PROJECT OBJECTIVES:

1. Create Smart Water management that can dispense clean and safe drinking water.

2. Implement a monitoring system to track water usage and quality in real-time.

3. Minimize water wastage by optimizing management operation.

4. Ensure continuous availability of water to the public.

5. Implement user-friendly features for easy access and use.

Understanding the Problem:

The problem statement revolves around designing and implementing Smart Water management. These fountains aim to address several key issues.

Clean and Safe Drinking Water:

The primary purpose of these management is to provide access to clean and safe drinking water to the public. Ensuring water quality is paramount to prevent health risks.

Water Conservation:

In addition to providing clean water, these management must be equipped with technology to minimize water wastage. This involves tracking water usage and implementing efficient dispensing mechanisms.

Continuous Availability:

Smart Water management should be available to the public at all times. This requires a monitoring system to detect issues such as low water levels or malfunctions.

Proposed Solution:

To solve the problem, we propose the following solution. 1.Smart Dispensing Mechanism:

Implement a sensor-based system that dispenses water when a user approaches. Utilize ultraviolet (UV) or other suitable technology to ensure water quality. Incorporate a touchless user interface for easy and hygienic access.

2. Real-time Monitoring:

Install water quality sensors to monitor water quality continuously. Use flow sensors to track water usage in real-time. Implement a central monitoring system that collects data from all managements.

3. Smart Control System:

Optimize management operation based on real-time data to reduce wastage. Send alerts to maintenance teams when issues are detected.

4. Water Source and Storage:

Connect the fountains to a clean and reliable water source (e.g., municipal water supply). Include water storage tanks with automatic refilling mechanisms to ensure uninterrupted service during water supply interruptions.

5. User-friendly Features:

Design a user interface that is intuitive and accessible to people of all ages and abilities. Provide information about water quality and source for user confidence. Ensure the fountains are ADA-compliant for accessibility.

6. Maintenance and Support:

Establish a maintenance schedule for regular checks and cleaning. Provide user support through a dedicated helpline or mobile app for reporting issues.

7. Sustainability:

Implement energy-efficient components, such as LED lighting and low-power sensors. Promote the use of reusable water bottles to reduce plastic waste.

Project Implementation Steps:

1. Requirements Gathering:

Gather detailed requirements from stakeholders, including local authorities, public health officials, and potential users.

2. Design:

Develop a detailed design plan that includes the management’s physical structure, the placement of sensors, water source connections, and user interface.

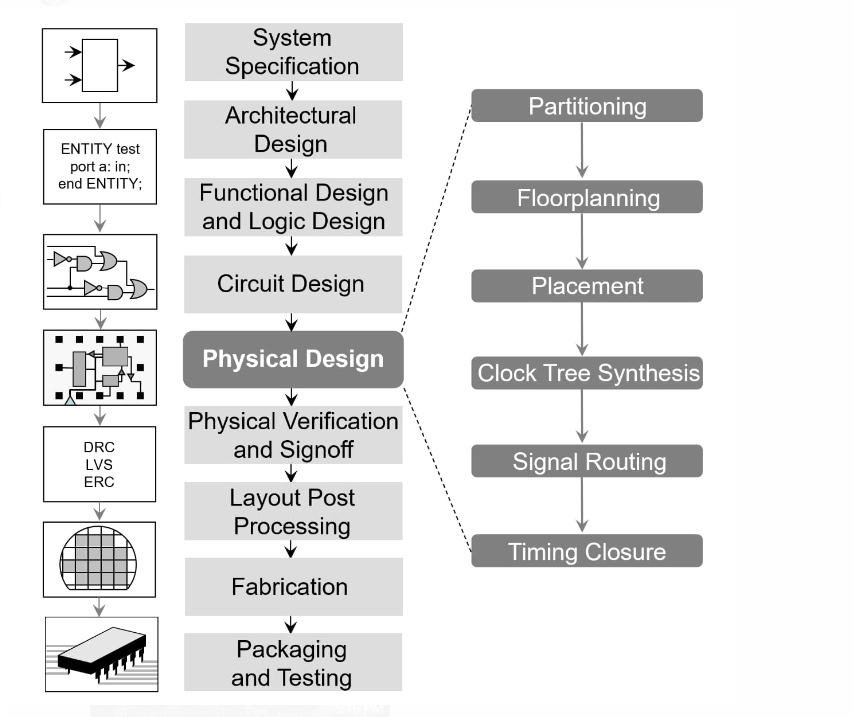
3. Prototyping:

Build a prototype management to test the smart dispensing mechanism and monitoring system.

3. Full-scale Implementation:

Deploy Smart Water management at selected locations.

Physical design



Architectural design:

A physical architecture is an arrangement of physical elements (system elements and physical interfaces) which provides the design solution for a product, service, or enterprise, and is intended to satisfy logical architecture elements and system requirements.

System Design Specifications:

System Design Specifications Provide a physical design that will meet the specifications outlined in the systems requirement document. See the Week 8 Announcement for an example of the Physical Design. It is a diagram that shows the people, processes, and data that your proposed system will impact.

Signoff in physical design:

Signoff is the final stage in ASIC design where we close the design against all the design rule checks in order the make sure our design will work properly after fabrication and it is flawless. The order of design closer is also important we have to follow a certain order in design closer.

Functional design:

The functional design (FD) phase of a development project focuses on the actions of a new or revised product, program, service, or process. The functional design specification (FDS) identifies what its design object is to do and is more concerned with what is to be done and less with how it happens.

Logic design:

Logic design, basic organization of the circuitry of a digital computer. All digital computers are based on a two-valued logic system—1/0, on/off, yes/no (see binary code).

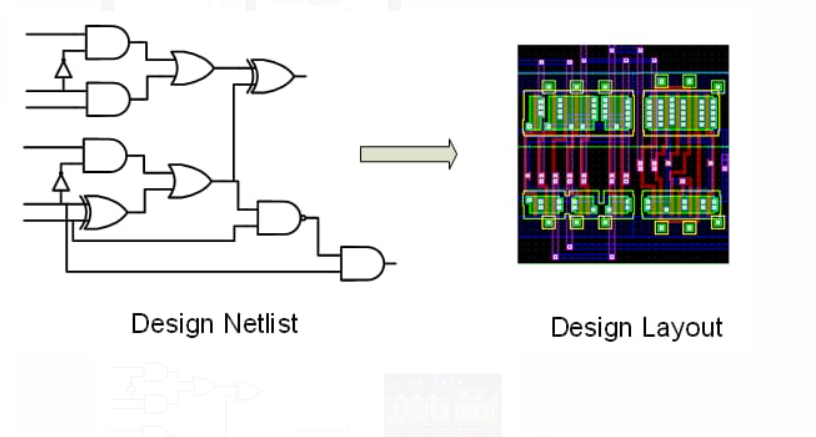
Fabrication:

Fabrication is using processes to create component parts that can be used to make a product or structure, as well as the process of constructing an item from standardised parts.

Packaging and Testing:

The packaging test is a study that allows knowing the strengths and weaknesses of a product's packaging and the messages it conveys. Thanks to this technique, it's possible to obtain consumer insights that allow knowing if the presentation of a product will have the desired success in the consumer.

Circuit design:



Phython code

GlowScript 2.8 Vpython

def xrange (v0, h1,m1):

#this function takes the initial velocity (as a vector) and the i

#initial y-value. It returns the horizontal distance.

G1=vector(0,-9.8,1)

T1=0

dt=0.001

#initial position

R1=vector(0,h1,0)

P1=m1\*v0

while r1.y1>0:

F1=m1\*g1

P1=p1+F1\*dt

R1=r1+p1\*dt/m1

T1=t1+dt

return(r1.x1)

theta1=0

dtheta1=0.05

vlaunch=5

mstart1=1

tgraph=graph(xtitle="Launch Angle [Degrees]",ytitle="Range [m1]")

f1=gcurve(color=color.blue)

While theta1<pi/2:

vtemp1=vlaunch\*vector(cos(theta1),sin(theta1),0)

x1=xrange(vtemp1,ystart1,mstart1)

f1.plot(theta1\*180/pi,x1)

theta1=theta1+dtheta1

Sensor Unit:

This block contains the four sensors. The data acquired from the sensors will be transmitted to the control unit. Control unit will then have some logic designed to send corresponding signals to control other blocks of the water management. At the same time, the display screen on the water fountain will display the readings along with the determined water quality level and remaining water quantity. For the PH-value sensor, temperature sensor and conductivity sensor, values will be retrieved and calculated to determine the overall water quality level. When poor water quality is determined, the water replacement procedures will take place. The weight sensor readings will be used to determine the amount of fresh water left in the water tank.

Temperature Sensor:

A water-proof temperature sensor is going to be used. Part number from sparkfun is: DS18B20 [6]. This temperature sensor is compatible with a relatively wide range of power supply from 3.0V to 5.5V. The measured temperature ranges from -55 to +125 celsius degrees.

Between -10 to + 85 degrees, the accuracy is up to +-0.5 degrees. This sensor can fulfill all requirements needed for this project.

PH-sensor:

PH value is a valued indicator of water quality. This PH-sensor[7] works with 5V voltage, which is also compatible with the temperature sensor. It can 6measure the PH value from 0 to 14 with an accuracy of +- 0.1 at the temperature of 25 degrees.

Conductivity sensor:

Conductivity sensor is also part of the water quality assessment. The input voltage is from 3.0 to 5.0V. The error is small, +-5%F.S. The measurement value ranges from 0 to 20 ms/cm which is enough for water quality monitoring.

Liquid Level Sensor:

This sensor [9] is responsible for reflecting how much freshwater is left in the water tank. When the water level is low, fresh water will be pumped to the water tank to ensure the water management keeps running with freshwater. This sensor is 0.5 Watts. For water level from 0 to 9 inches, the corresponding sensor outputs readings from 0 to 1.6. From that, the quantity of freshwater left can be determined

1. Continue building the project by developing the water management status platform.
2. Use web development technologies (e.g., HTML, CSS, JavaScript) to create a platform that displays real-time water management status.
3. Design the platform to receive and display real-time water management data, including water flow rate and malfunction alerts.

HTML: Create the structure of your web page.

<!DOCTYPE html>

<html>

<head>

<title>Smart Water Management</title>

</head>

<body>

<header>

<h1>Welcome to Smart Water Management</h1>

</header>

<nav>

<!-- Navigation links can be added here -->

</nav>

<main>

<section id="dashboard">

<!-- Real-time monitoring and data visualization components go here -->

</section>

<section id="alerts">

<!-- Display alerts and notifications here -->

</section>

<section id="analytics">

<!-- Present data analysis and insights here -->

</section>

</main>

<footer>

<p>&copy; 2023 Smart Water Management</p>

</footer>

</body>

</html>

CSS:

/\* Example CSS for a water management dashboard \*/

body {

font-family: Arial, sans-serif;

background-color: #f0f0f0;

}

.header {

background-color: #3498db;

color: #fff;

padding: 10px;

text-align: center;

}

.container {

width: 80%;

margin: 0 auto;

padding: 20px;

}

.card {

background-color: #fff;

border: 1px solid #ccc;

border-radius: 5px;

padding: 10px;

margin: 10px;

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);

}

/\* Add more styles as needed for your specific application \*/

JAVASCRIPT:

// Simulated water level sensor data

const waterLevelSensor = {

currentLevel: 0, // Initialize to a safe level

};

// Function to check water level and send alerts

function checkWaterLevel() {

if (waterLevelSensor.currentLevel > 80) {

// Water level is too high, send an alert

console.log("Water level is too high! Alerting authorities...");

// You can implement alerting mechanisms here (e.g., sending emails or notifications).

}

}

// Simulate changing water levels (for demonstration purposes)

function simulateWaterLevelChange() {

setInterval(() => {

## waterLevelSensor.currentLevel = Math.random() \* 100; // Random level (0-100)

checkWaterLevel();

}, 3000); // Simulate every 3 seconds

}

// Start simulating water level changes

simulateWaterLevelChange();

Conclusion:

Smart water management is crucial for addressing the growing global water challenges. It involves the use of technology and data-driven solutions to optimize water usage, reduce wastage, and improve the overall efficiency of water systems. This approach can lead to significant benefits, including conservation of water resources, cost savings, and environmental sustainability. To conclude, smart water management is essential for a more sustainable and water-secure future.

SUBMITTED BY

SASIKUMAR J