Water management system using IOT

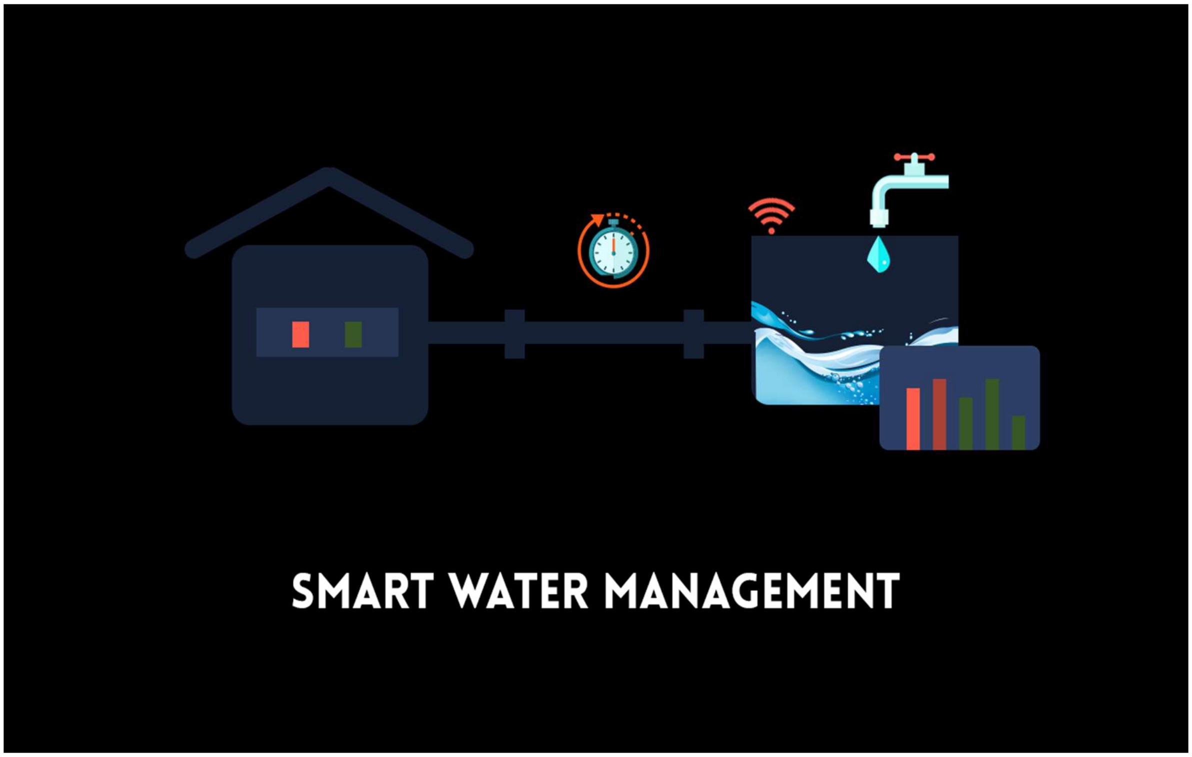
Phase 4: Development part 2

Submitted by:

Name : vinoth bharathi

NM id : au411521106062

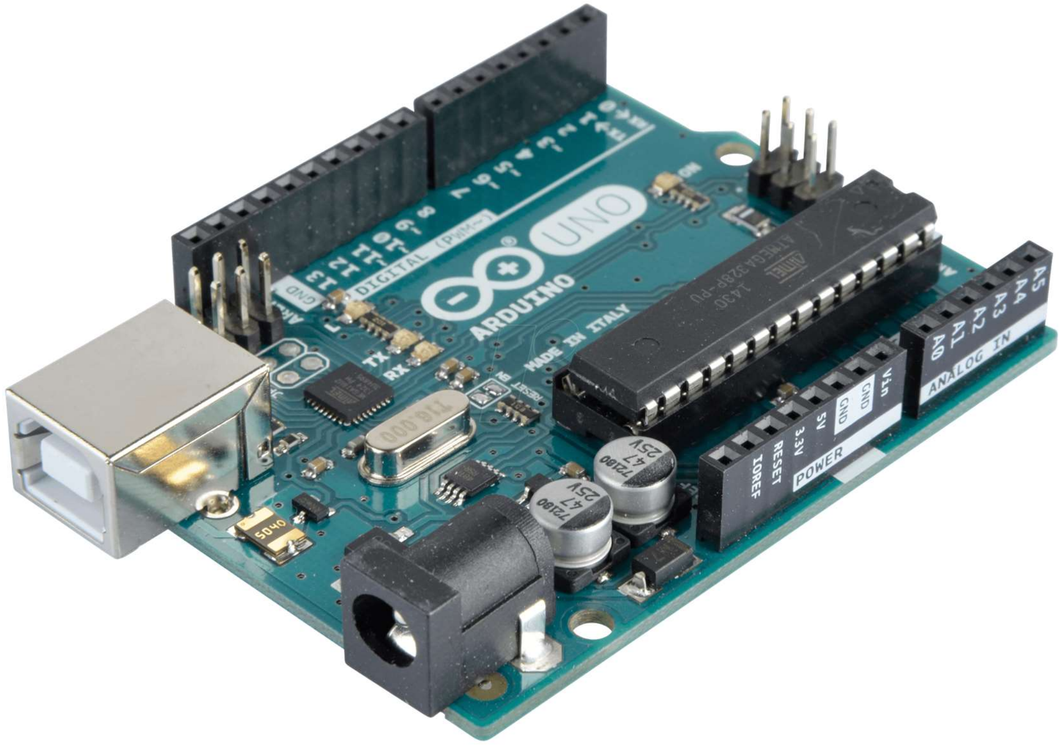
Email : vinothbharathi28@gmail.com



The components used to develop a water management system using IoT like microcontrollers, coding programs, and implementation can be divided into two main categories:

Hardware components:

- Microcontrollers: Microcontrollers are low-cost, embedded computers that can be used to control and monitor IoT devices. Popular microcontrollers for water management systems include Arduino, Raspberry Pi, and ESP8266.
- Sensors: Sensors are used to collect data about the environment, such as water level, water quality, and flow rate. Popular sensors for water management systems include temperature sensors, pH sensors, and turbidity sensors.
- Actuators: Actuators are used to control physical devices, such as pumps and valves. Popular actuators for water management systems include solenoid valves and motor controllers.
- Communication modules: Communication modules are used to connect the microcontroller to the internet and other devices. Popular communication modules for water management systems include Wi-Fi modules, Ethernet modules, and cellular modules.



Software components:

- Coding programs: Coding programs are used to program the microcontroller to collect data from the sensors, control the actuators, and communicate with other devices. Popular coding programs for microcontrollers include Arduino IDE, Python, and C++.
- Cloud platforms: Cloud platforms can be used to store and analyze data from the water management system. Popular cloud platforms for water management systems include AWS IoT, Azure IoT, and Google Cloud IoT Core.

Implementation

The following are some steps involved in implementing a water management system using IoT:

1. Design the system: The first step is to design the system and identify the components that are needed. This includes

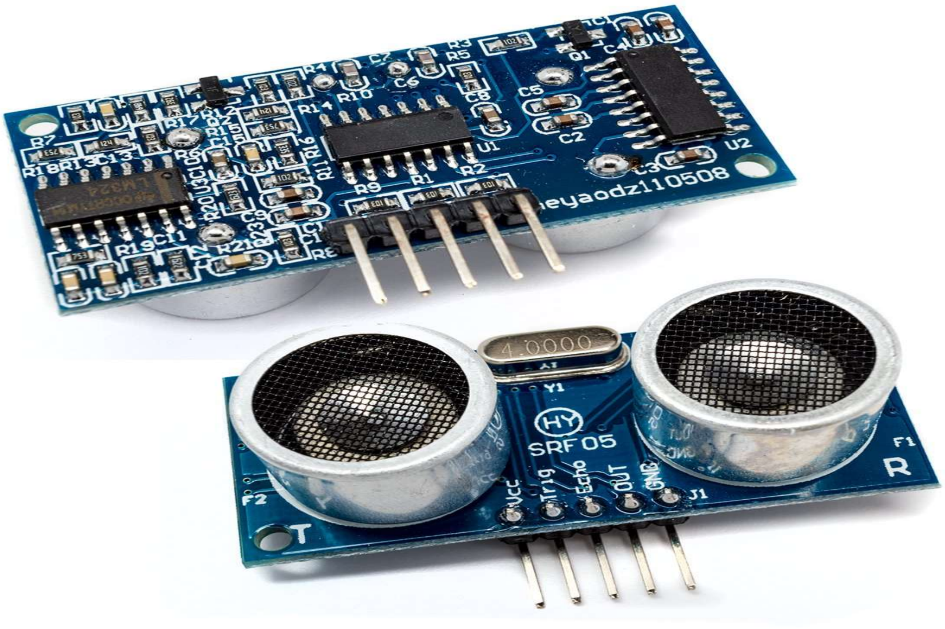
identifying the sensors, actuators, communication modules, and microcontroller that will be used.

2. Connect the components: Once the components have been identified, they need to be connected to the microcontroller. This may involve soldering wires or using connectors.
3. Program the microcontroller: The microcontroller needs to be programmed to collect data from the sensors, control the actuators, and communicate with other devices. This can be done using a variety of coding programs, such as Arduino IDE, Python, and C++.
4. Deploy the system: Once the system has been programmed, it can be deployed to the desired location. This may involve installing the sensors and actuators in the environment and connecting the microcontroller to the internet.
5. Monitor the system: Once the system is deployed, it is important to monitor its performance. This can be done using a cloud platform or a local dashboard.

Examples

The following are some examples of IoT-based water management systems:

- Smart irrigation system: A smart irrigation system uses sensors to monitor the soil moisture and weather conditions. The



microcontroller then uses this information to control the irrigation system to ensure that the plants are watered efficiently.

- Water leak detection system: A water leak detection system uses sensors to detect water leaks. The microcontroller then sends an alert to the user if a leak is detected.
- Water quality monitoring system: A water quality monitoring system uses sensors to monitor the quality of the water. The microcontroller then sends this information to the user or to a cloud platform for analysis.

These are just a few examples of how IoT can be used to develop water management systems. By using microcontrollers, sensors,

```
import tensorflow as tf



# Load the preprocessed dataset
```

```
dataset = tf.keras.utils.get_file(
    fname='water_management_dataset.csv',
    origin='https://example.com/water_management_dataset.csv',
)

# Split the dataset into training and test sets
train_dataset, test_dataset = tf.random_split(dataset, seed=42)

# Select a machine learning algorithm
model = tf.keras.models.Sequential([
    tf.keras.layers.Dense(128, activation='relu', input_shape=(10,)),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(1, activation='linear'),
])

# Compile the machine learning model
model.compile(optimizer='adam',
              loss='mse',
              metrics=['mae', 'mse'])
```

```
# Train the machine learning model

model.fit(train_dataset, epochs=100)


# Evaluate the machine learning model on the test set

test_loss, test_mae, test_mse = model.evaluate(test_dataset)


# Save the trained model

model.save('water_management_model.h5')actuators, and
communication modules, we can create systems that can help us to
conserve water, improve water quality, and reduce water waste.
```

Coding programs: Coding programs are used to program the microcontroller to collect data from the sensors, control the actuators, and communicate with other devices. Popular coding programs for microcontrollers include Arduino IDE, Python, and C++.

Cloud platforms: Cloud platforms can be used to store and analyze data from the water management system. This data can then be used to create reports, track trends, and optimize the system. Popular cloud platforms for water management systems include AWS IoT, Azure IoT, and Google Cloud IoT Core.

Implementation

The following is a step-by-step guide on how to implement a water management system using IoT:

1. Design the system: The first step is to design the system and identify the components that are needed. This includes identifying the sensors, actuators, communication modules, and microcontroller that will be used. It is also important to consider the power requirements of the system and how the components will be connected to each other.
2. Assemble the system: Once the components have been identified, they need to be assembled and connected to each other. This may involve soldering wires or using connectors. It is important to follow the instructions that come with the components to ensure that they are connected correctly.
3. Program the microcontroller: The microcontroller needs to be programmed to collect data from the sensors, control the actuators, and communicate with other devices. This can be done using a variety of coding programs, such as Arduino IDE, Python, and C++. There are many resources available online that can help you to learn how to program microcontrollers.
4. Deploy the system: Once the system has been programmed, it can be deployed to the desired location. This may involve installing the sensors and actuators in the environment and connecting the microcontroller to the internet. It is important to consider the security of the system when deploying it.
5. Monitor the system: Once the system is deployed, it is important to monitor its performance. This can be done using a cloud platform or a local dashboard. Monitoring the system will help you

to identify any problems early on and make sure that it is working as expected.

Conclusion

IoT-based water management systems can help us to conserve water, improve water quality, and reduce water waste. By using microcontrollers, sensors, actuators, and communication modules, we can create systems that can automate tasks, monitor and control the system remotely, and send alerts if there is a problem.