

Department: BIO MEDICAL ENGINEERING

Year: III rd. YEAR

TOPIC: SMART WATER MANAGEMENT

Team members

1.Gowri Sankar. S

2.Kaushik. E

3.Jegan. S

4.Hari Krishnan.K

5.Gowtham. V

By.

Kaushik .E

Introduction

Smart water management systems can provide a more resilient and efficient water supply system, reducing costs and improving sustainability. High-technology solutions for the water sector include digital meters and sensors, supervisory control and data acquisition (SCADA) systems, and geographic information systems (GIS).

This explainer is adapted from proceedings of a workshop conducted by the Asian Development Bank (ADB) in Tashkent, Uzbekistan for the water sector. The workshop introduced smart systems and focused on remote monitoring of water networks using smart meters and other instruments.

Hardware components:

(a)Sensors Used With Boltduino/Arduino

* 5V Relay
* I2C LCD
* Boltduino
* 9V Battery
* Bolt Wifi Module
* IRF540 MOSFET
* Water Flow Sensor
* Ultrasonic Sensor X 2
* 1N4007 Rectifier Diode
* 12V DC Solenoid Valve
* Water Lifting Submersible Pump
* 4-way Capacitive Touch Switch Module
* 3-6 V Mini Micro Submersible Water Pump
* LM35 IC (Temperature sensor)

(b)Sensors Used With Boltduino/Arduino:

* Nodemcu
* Piezo Buzzer
* IR Sensor X 2
* DC Motors X 2
* 12V DC Adapter
* TCS3200 Color Sensor
* Capacitive Touch Sensor
* ESP8266 Motor Driver Shield
* Analog Multiplexer IC – CD4051

Software apps and online services:

* Arduino IDE
* Bootstrap Studio
* Spyder (Anaconda)
* Twilio
* Canva
* Hostinger
* Integromat
* Mega Creator
* Pichon (Icons8)

Smart water Management coding:

import RPi.GPIO as GPIO

import time

# Set GPIO mode to BCM

GPIO.setmode(GPIO.BCM)

# Define GPIO pins for the sensor

TRIG = 23

ECHO = 24

# Set up the GPIO pins

GPIO.setup(TRIG, GPIO.OUT)

GPIO.setup(ECHO, GPIO.IN)

def get\_distance():

# Trigger the sensor

GPIO.output(TRIG, True)

time.sleep(0.00001)

GPIO.output(TRIG, False)

start\_time = time.time()

end\_time = time.time()

# Wait for the ECHO pin to go high

while GPIO.input(ECHO) == 0:

start\_time = time.time()

# Wait for the ECHO pin to go low

while GPIO.input(ECHO) == 1:

end\_time = time.time()

# Calculate the duration of the pulse

pulse\_duration = end\_time - start\_time

# Calculate distance in centimeters

distance = pulse\_duration \* 17150

return distance

try:

while True:

distance = get\_distance()

print("Distance: {:.2f} cm".format(distance))

time.sleep(1) # Delay for 1 second

except KeyboardInterrupt:

GPIO.cleanup()