**SMART WATER FOUNTAIN**

SOLAR WATER FOUNTAIN

ABOUT

Solar water fountains enhanced with IOT devices combine sustainable energy sources with smart technology to create efficient and interactive water features.



OBJECTIVES

* Our aim is to utilise solar power to operate the fountain,reducing reliance on grid electricity and promoting renewable energy sources.
* Our goal is to
* Implement water recirculation system is to reduce water consumption and waste
* Enable customize water patterns like lighting effects and operation through IOT controls
* Utilize IOT sensors to maintain water quality

IOT SENSOR SETUP

**Water level sensor**

* Monitors the water level in the fountain basin
* This sensor helps prevent the pump from running dry
* Ensure it operates where there’s enough water

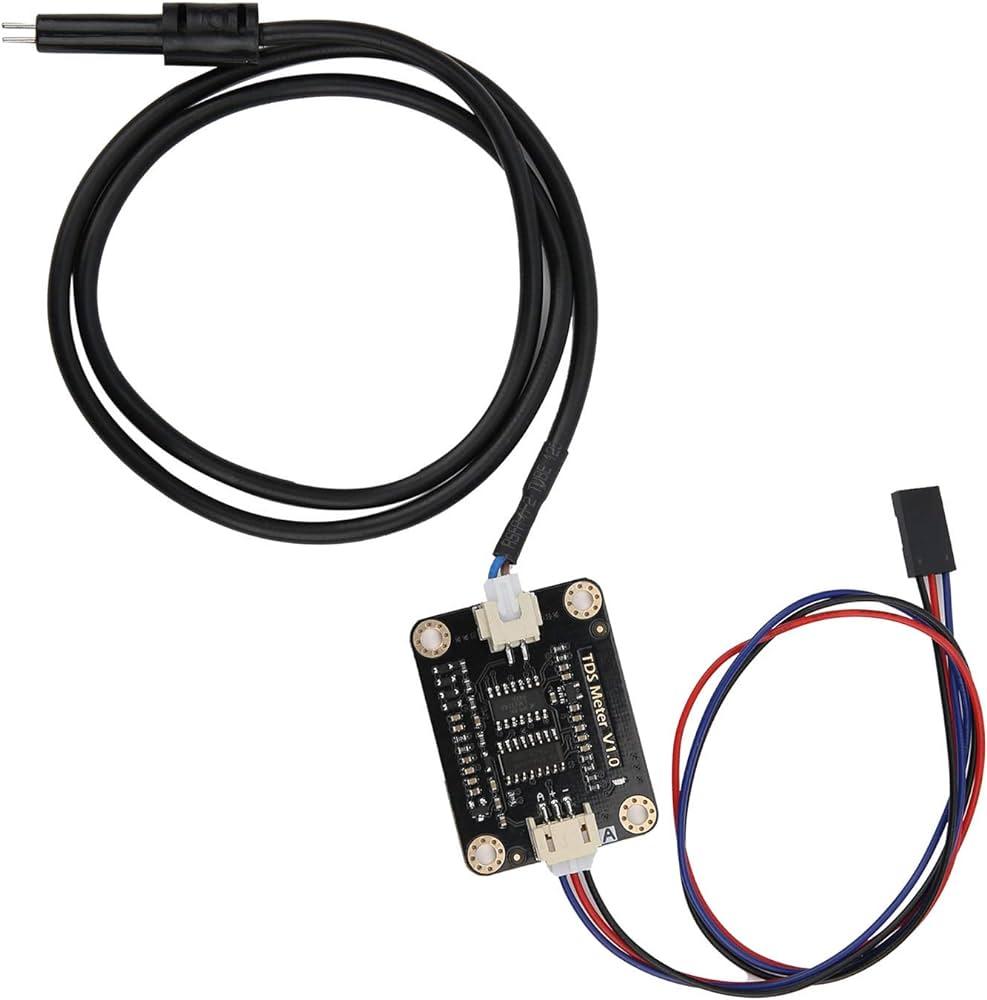


**Temperature sensor**

Monitors the ambient temperature ,which can affect the operation of the fountain and help adjust settings for energy efficiency.

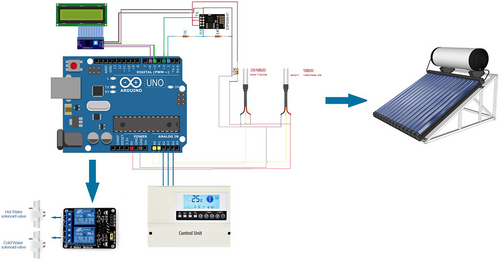
**Conductivity sensor**

Maintain water quality,preserve the health of aquatic life,and ensure the safety and aesthetics of the fountain.



**PH sensor**

* Measure the acidity or alkalinity of the water by assessing the concentration of hydrogen ions(H+) in the water
* Ph sensors require periodic calibration using standard ph solutions to ensure accurate measurements.

**Schematic diagram(prototype)**

**Mobile app**

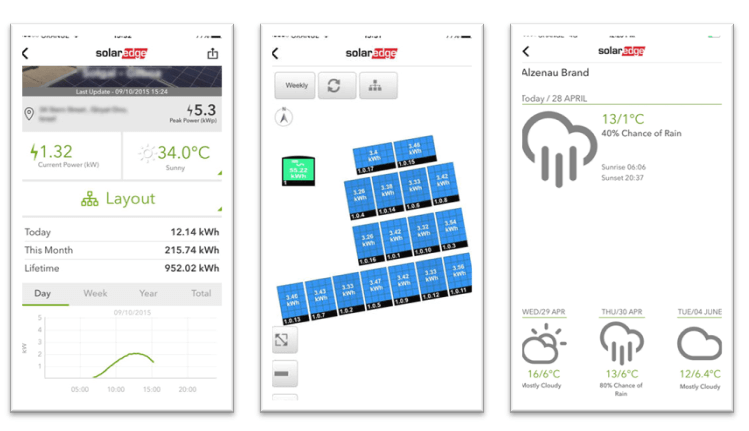
**Pros:**

* Set up automated alerts for any system issues
* Provides you full visibility of the solar system’s functionality
* Helps detect excessive power usage

#### **Cons:**

* No scope to control the charging behavior

#### **Features:**

* Current system energy measurement
* Energy production monitoring in months, days, as well as lifetime
* Power graph display
* Site list and image
* Weather data
* Customized user preference configuration

**Code implementation**

import time

import RPi.GPIO as GPIO # Assuming you're using a Raspberry Pi

# Define GPIO pins for sensors and actuators

WATER\_LEVEL\_SENSOR\_PIN = 17

PUMP\_PIN = 18

# Initialize GPIO

GPIO.setmode(GPIO.BCM)

GPIO.setup(WATER\_LEVEL\_SENSOR\_PIN, [GPIO.IN](http://gpio.in/))

GPIO.setup(PUMP\_PIN, GPIO.OUT)

def check\_water\_level():

# Function to check water level using the sensor

# Implement logic to read the sensor and return water level status

pass

def control\_pump(status):

# Function to control the pump

# Turn pump on (status = True) or off (status = False)

pass

if \_\_name\_\_ == "\_\_main\_\_":

try:

while True:

water\_level = check\_water\_level()

if water\_level:

print("Water level is low. Turning on the pump.")

control\_pump(True)

else:

print("Water level is sufficient.")

control\_pump(False)

time.sleep(10) # Adjust the delay based on your requirements

except KeyboardInterrupt:

GPIO.cleanup() # Cleanup GPIO on exit