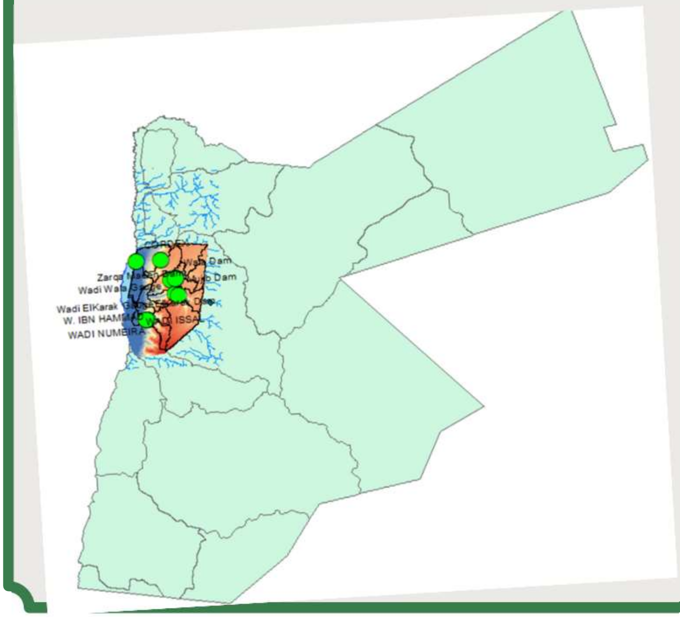
UJ-DAP Project 2023



"محاكاة الفيضانات الوميضية لحوض البحر الميت
ضمن توقعات سيناريوهات التغير المناخي لبناء
المرونة للمجتمعات المتضررة "

Simulate Flash Flood in the Dead Sea Basin under the GCMs/RCMs Projections: Building Resiliency for Flood Vulnerable Communities

Project team

Dr. Fayha Al-Shibli

Hydro-climate modelling

Assoc Prof. Mohammad Ashraf Ottom

data analysis

Prof. Ross Thompson
Water Planning and Governance consultation

eWater

Community
Women sector

Prof. Radwan Al-Weshah
Water Engineering

Prof. Hani Saoub
Vulnerable communities' consultation

Land use

Climate change

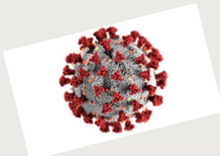
Resources cut off the water supply to the Dead Sea

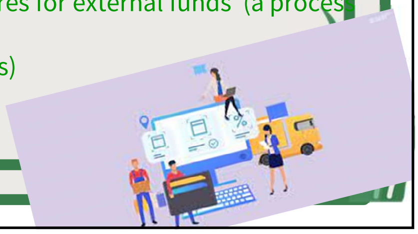
Dead Sea Stressors

Nature's Plan

Project Limitations

- Data shortage
- The ambiguity of responsibilities
- Budget limitations
- Seasonality (planting)
- CORONA
- Purchase requisition process: university regulations and procedures for external funds (a process implemented to streamline the ordering of goods through tenders)



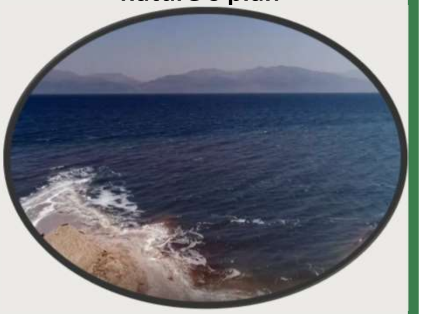


History to present an overview

Water from the sea of Galilee flows south down an ancient river

Jordan River provides a constant supply of water to the Dead Sea

Powerful force altered nature's plan



A water carrier was built to divert millions of gallons of water instead of the dead sea went to Mediterranean coastal cities and Negev.

The main source of water supply was cut off to the dead sea.

Over 96%

Of water flowing from Jordan river to dead sea is no longer reach the dead sea.

Mining the Sea

Natural Dead Sea Resources

Potash

Magnesium Chloride

Bromine

"Sinkholes abound in the northern basin The potash industry thrives in the southern basin"

Artificial evaporation bonds to dredge the minerals from the sea

Divide the dead sea into two strikingly basins

Since shallow ponds need to be replenished

The potash industry pumps out water from north to south (ponds)

Geologic evidence shows that the lake bottom witnessed continuous dark deposits through winter (due to floods)

and white deposits through summer (due to earthquakes cause deposits from seashells)

Land use maps

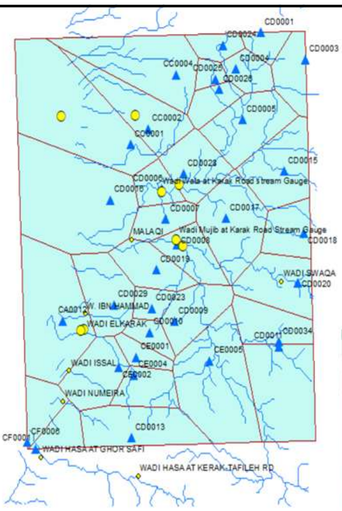
Thiessen polygons areal rainfall

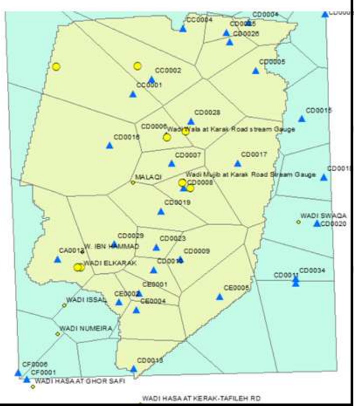
Added nodes

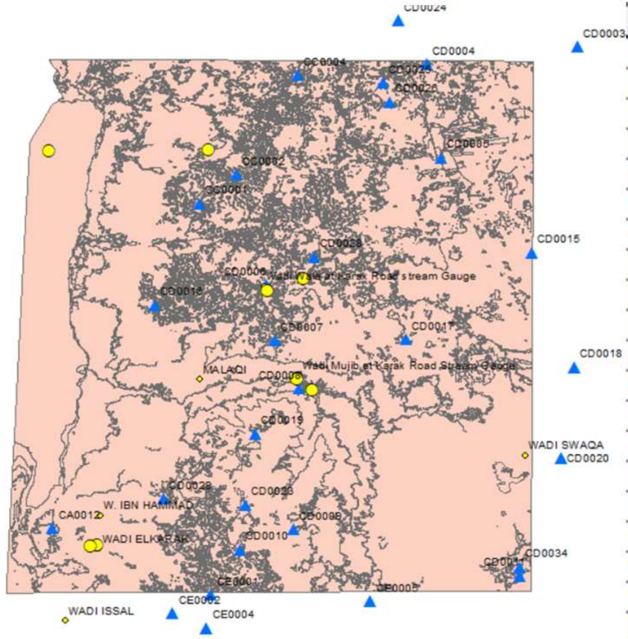
Calibration

Validation

Downloading Landsat Images for study area for the periods: 1972,1982,1992,2002,2012 and 2021. Supervised Image Classification, ARCmap and Geomatica software







AGG_USLB	Sha	Sh	Shape_Area
Artificial Waterbody	135	11	3875757.5127
Bare soil	826	70	1589593751.2005
Basaltic Plain	460	39	2210731.7901
Built-up	288	24	108394858.7468
Closed Trees	738	63	1974455.0441
Extraction Site	125	10	9072874.9776
Grasslands	488	41	274193154.8708
Herbaceous Crop Irrigated	115	98	65142576.0822
Herbaceous Crop Rainfed	607	51	675111886.9840
Irrigated orchards	110	94	36416071.3967
Natural Waterbody and Fresh	257	21	27800.0
Open Trees	127	10	2665585.4396
Rainfed orchards	156	13	44327867.4861
Saline Soil	156	13	20805685.5787
Saline Waterbody	158	13	264489499.5911
Sandy Areas	631	53	4825646.6406
Undifferentiated Bare Rock	172	14	255332006.357
Wady	464	39	10218930.8250
Wetlands	860	73	1384573.4068
Woody Vegetation	893	76	22580145.9807

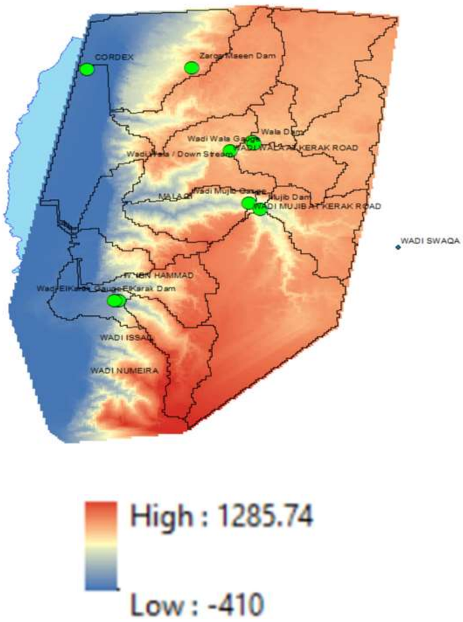


Table 4: Monthly CORDEX scenario runs (EUR-44 domain) used in the study

No.	GCM	RCM
1.	CORDEX Average	-
2.	CORDEX Ensemble	-
3.	CCCma-CanESM2 r1i1p1	SMHI-RCA4
4.	CNRM-CERFACS-CNRM-CM5 r1i1p1	SMHI-RCA4
5.	CNRM-CERFACS-CNRM-CM5 r1i1p1	CNRM-ALADIN53
6.	CSIRO-QCCCE-CSIRO-Mk3-6-0 r1i1p1	SMHI-RCA4
7.	ICHEC-EC-EARTH r12i1p1	SMHI-RCA4
8.	ICHEC-EC-EARTH r1i1p1	KNMI-RACMO22E
9.	ICHEC-EC-EARTH r3i1p1	DMI-HIRHAM5
10.	IPSL-IPSL-CM5A-MR r1i1p1	SMHI-RCA4
11.	MIROC-MIROC5 r1i1p1	SMHI-RCA4
12.	MOHC-HadGEM2-ES r1i1p1	SMHI-RCA4
13.	MOHC-HadGEM2-ES r1i1p1	KNMI-RACMO22E
14.	MPI-M-MPI-ESM-LR r1i1p1	SMHI-RCA4
15.	MPI-M-MPI-ESM-LR r1i1p1	MPI-CSC-REMO2009
16.	MPI-M-MPI-ESM-LR r2i1p1	MPI-CSC-REMO2009
17.	NCC-NorESM1-M r1i1p1	SMHI-RCA4
18.	NOAA-GFDL-GFDL-ESM2M r1i1p1	SMHI-RCA4

HYDROLOGICAL MODEL

SIMHYD

Conceptual rainfall-runoff model, where its parameters are relatively sensitive to base flow index, pervious percent, and the capacity of soil moisture store.

Coefficients

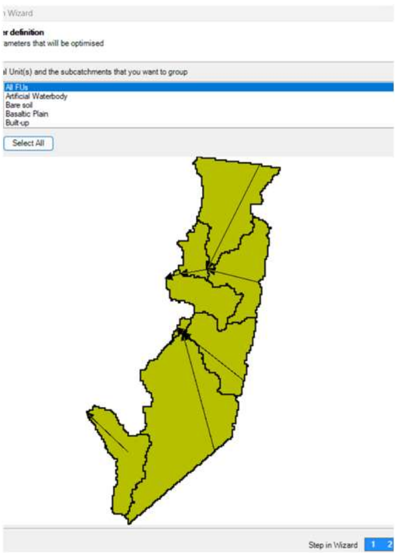
SIMHYD-Impervious Threshold (mm)	1
SIMHYD-RISC (mm)	1.5
SIMHYD-Perv. Fraction (None)	0.9
SIMHYD-SMSC (mm)	320
SIMHYD-Infiltration shape (None)	3
SIMHYD-Infiltration Coeff. (None)	200
SIMHYD-Interflow Coeff. (None)	0.1
SIMHYD-Recharge coefficient (None)	0.2
SIMHYD-Baseflow coeff. (None)	0.3

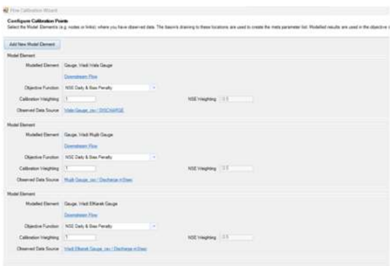
3 stores:

groundwater, soil moisture and interception loss.

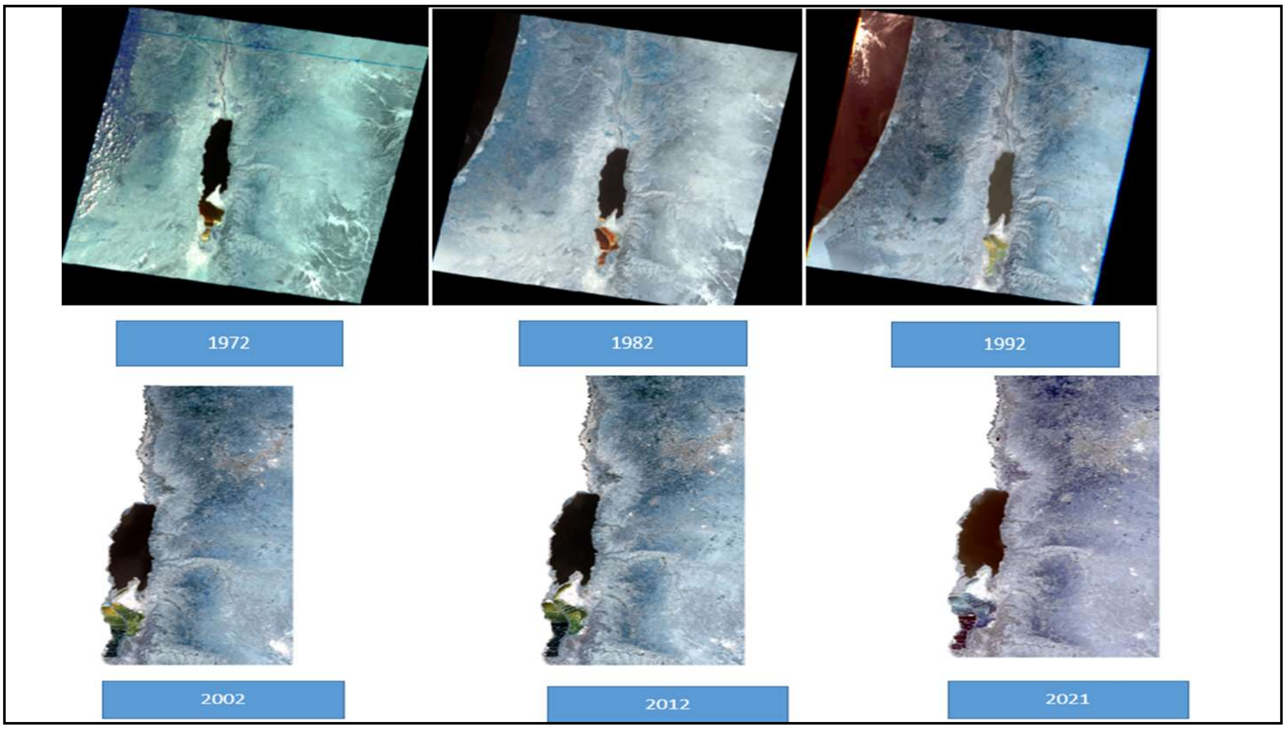
INPUTS

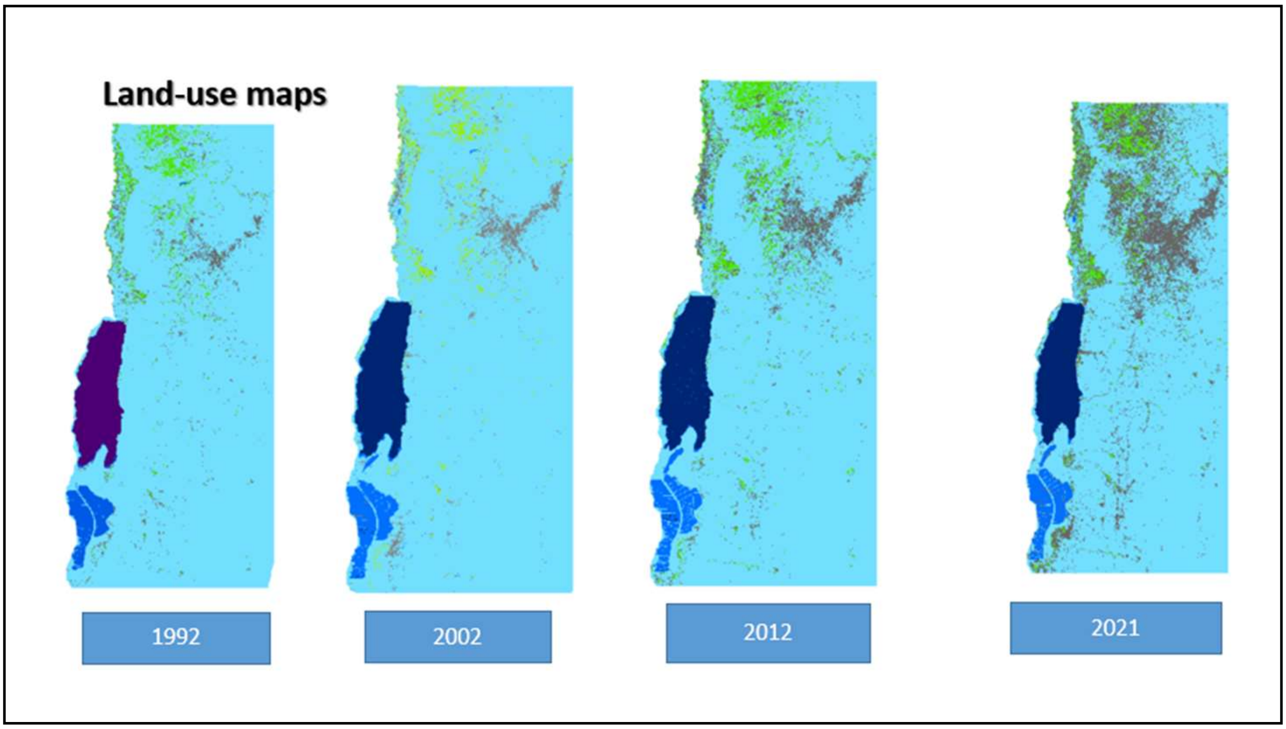
Daily rainfall, areal potential evapotranspiration.

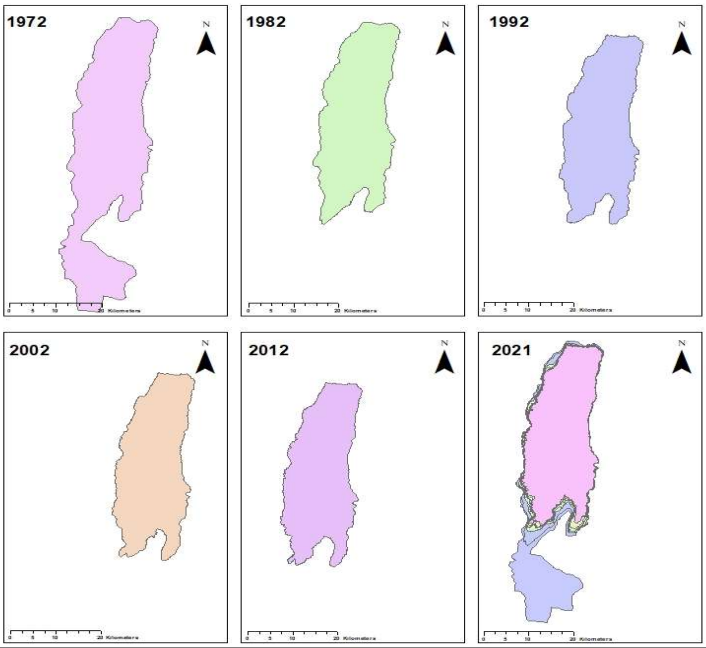




Calibration Wizard


1972
1982
1992
2002
2012
2021


Land-use maps
1992
2002
2012
2021



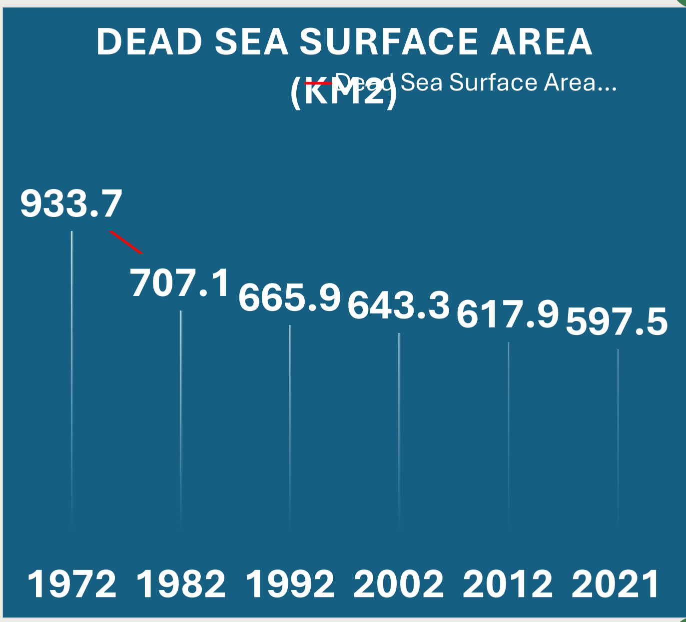
DeadSea Area Change
Between 1972-2021

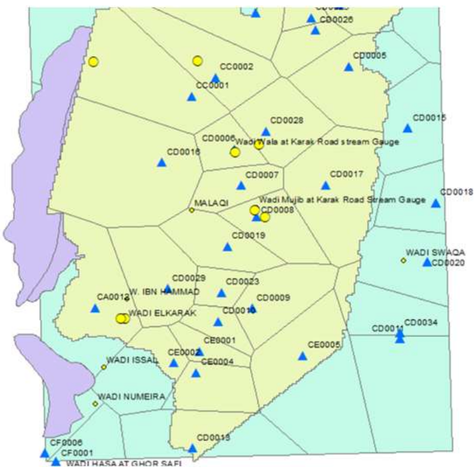
Year	DeadSea Area (Sq. KM)
1972	933.7
1982	707.1
1992	665.9
2002	643.3
2012	617.9
2021	597.5

The Dead Sea is Shrinking

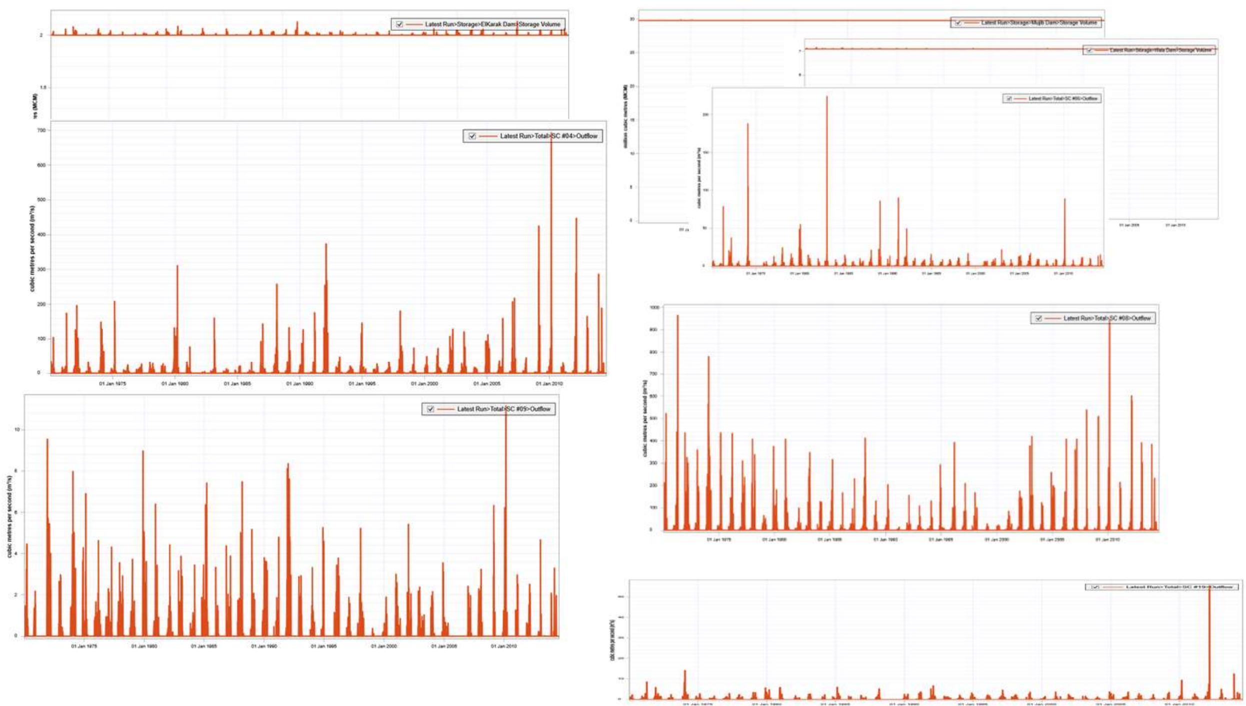
Write Your Topic or Idea

Elaborate on what you want to discuss.



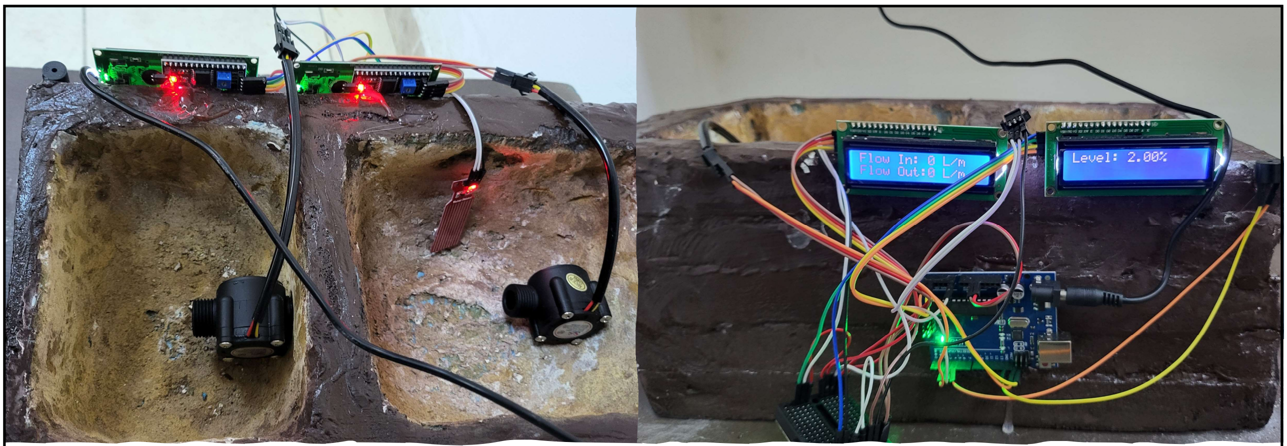


Runs	deadSea-subcatch250 (1)	deadSea-subcatch250 (1)	deadSea-subcatch250 (1)	deadSea-subcatch250 (1)	deadSea-subcatc
Variable	Flow	Flow	Flow	Flow	Flow
Statistic	90th Percentile	90th Percentile	Mean	Maximum	Maximum
Units	m^3.s^-1.m^-2	ML/d	ML/d	ML/d	m^3.s^-1.m^-2
SC #01	8.84E-11	0.000661	0.00421	5.93	7
SC #02	8.84E-11	0.0239	0.152	215	7
SC #03	7.02E-10	0.245	0.452	334	9
SC #04	4.51E-09	1.31	1.68	693	2
SC #05	8.84E-11	0.000182	0.00116	1.63	7
SC #06	1.12E-09	0.175	0.287	224	1
SC #07	3.40E-09	0.818	1.02	352	1
SC #08	9.26E-09	2.6	4.17	965	3
SC #19	4.33E-10	0.0298	0.063	55.5	8
SC #09	1.01E-10	0.0182	0.0657	11.2	6
SC #10	1.07E-10	0.0167	0.126	121	7
SC #11	1.75E-11	0.00329	0.0628	28.2	1
SC #17	0	0	0.000662	0.671	7
SC #12	1.81E-09	1.04	1.39	1.05E+03	1
SC #13	0	0	0.00461	5.4	5
SC #14	5.28E-09	0.732	1.17	314	2
SC #15	5.62E-12	0.000508	0.013	5.67	6
SC #16	9.39E-11	0.0114	0.0583	15.2	1
SC #18	7.18E-09	1.17	2.36	666	4

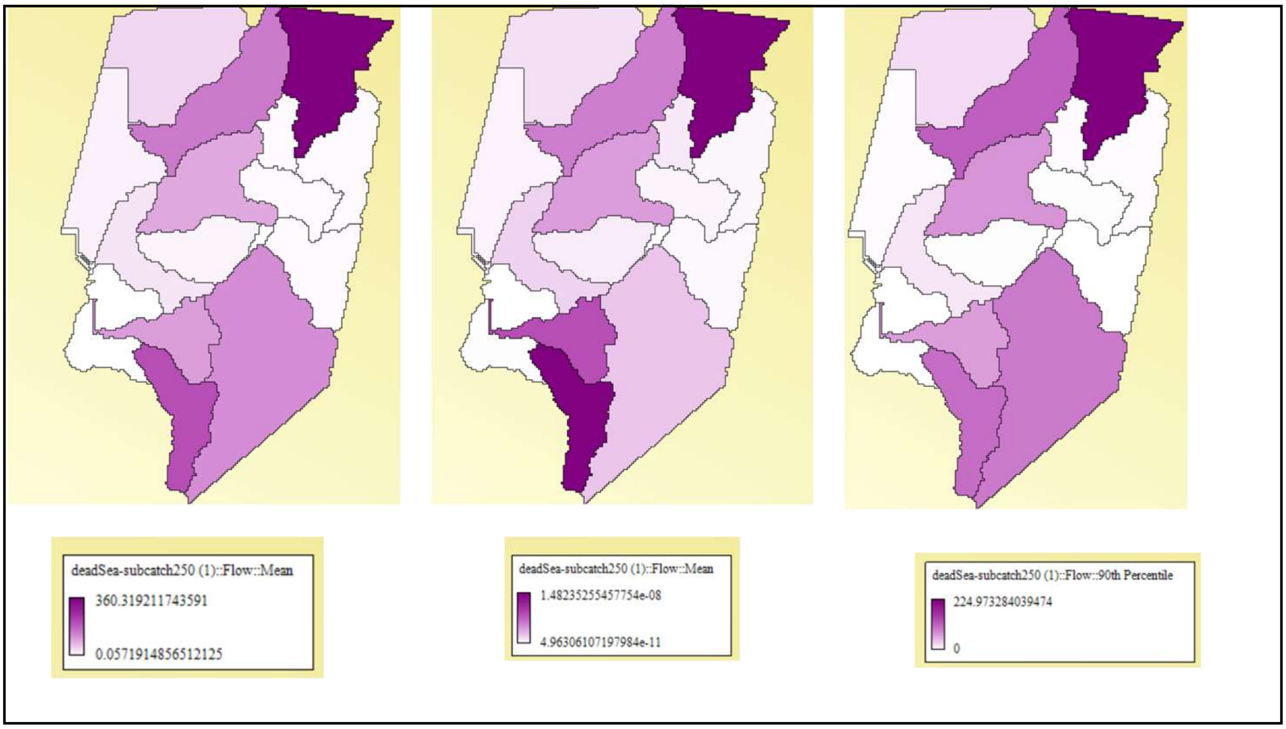


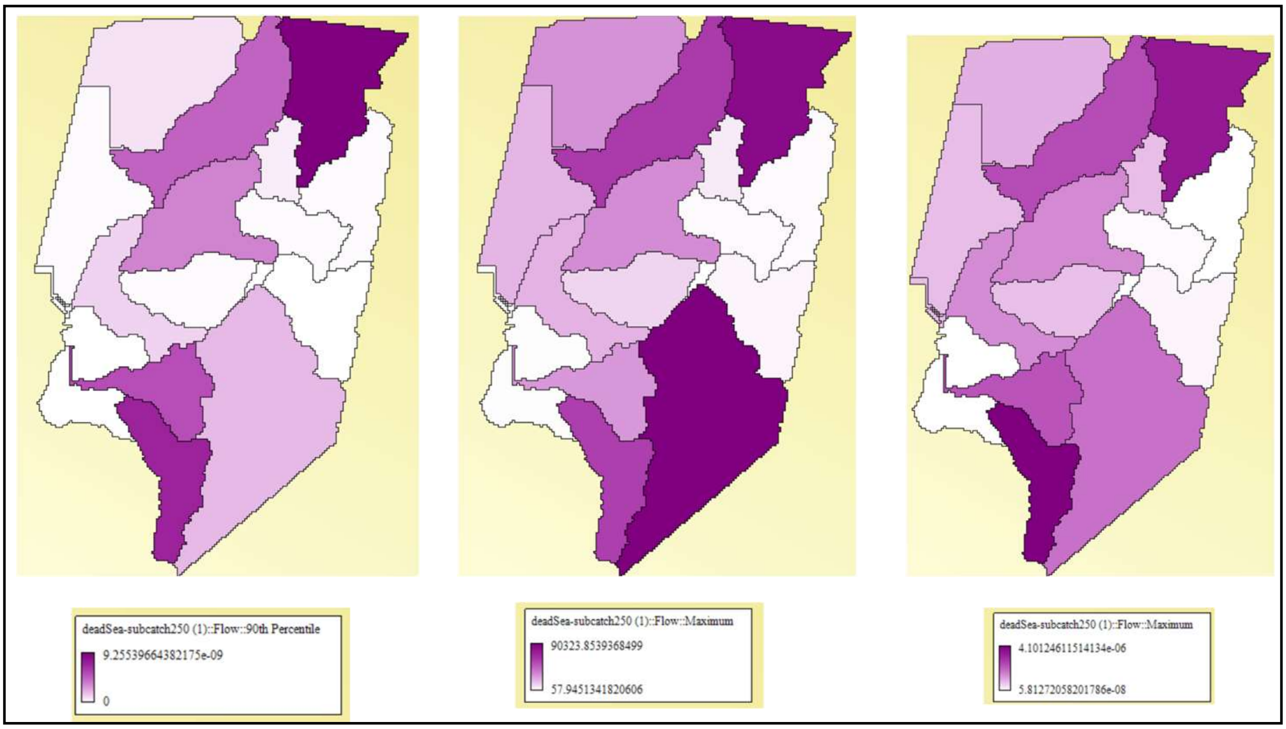
02

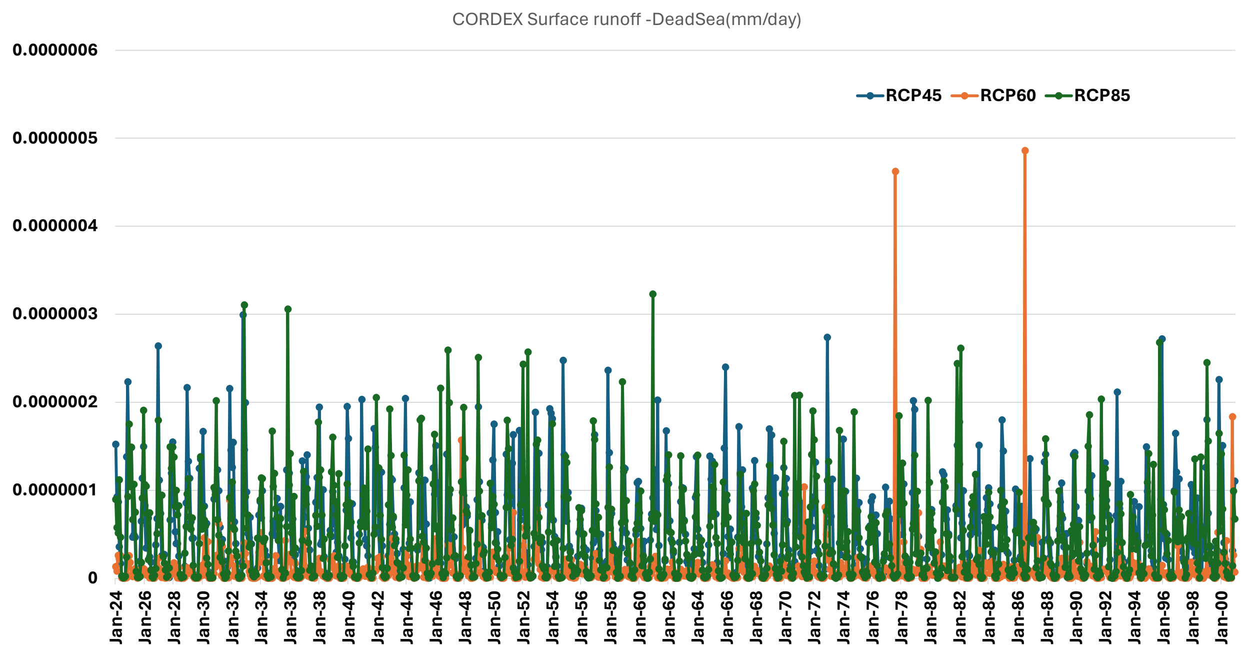
Dams Results



Dam sensors and IoT Flow Units and Water Level Alert System


deadSea-subcatch250 (1)::Flow::Mean
360.319211743591
0.0571914856512125
deadSea-subcatch250 (1)::Flow::Mean
1.48235255457754e-08
4.96306107197984e-11
deadSea-subcatch250 (1)::Flow::90th Percentile
224.973284039474
0


deadSea-subcatch250 (1)::Flow::90th Percentile
9.25539664382175e-09
0
deadSea-subcatch250 (1)::Flow::Maximum
90323.8539368499
57.9451341820606
deadSea-subcatch250 (1)::Flow::Maximum
4.10124611514134e-06
5.81272058201786e-08


CORDEX Surface runoff -DeadSea(mm/day)
0.0000006
0.0000005
0.0000004
0.0000003
0.0000002
0.0000001
0
RCP45
RCP60
RCP85
Jan-24
Jan-26
Jan-28
Jan-30
Jan-32
Jan-34
Jan-36
Jan-38
Jan-40
Jan-42
Jan-44
Jan-46
Jan-48
Jan-50
Jan-52
Jan-54
Jan-56
Jan-58
Jan-60
Jan-62
Jan-64
Jan-66
Jan-68
Jan-70
Jan-72
Jan-74
Jan-76
Jan-78
Jan-80
Jan-82
Jan-84
Jan-86
Jan-88
Jan-90
Jan-92
Jan-94
Jan-96
Jan-98
Jan-00

03

IoT and ML

ML & IoT

Arduino Uno - R3

Arduino Uno is a microcontroller board
has 14 digital input/output pins
6 analog inputs
16 MHz ceramic resonator
a power jack,
ICSP header and a reset button.

Rain Water Level Detection Sensor Module

This sensor module can estimate the water level through a series of exposed parallel conductor line marks measuring the water droplets and capacity of dams.

Easily convert water to analog signal, the output analog signal can be read directly on the Arduino board to achieve the effect of water level alarm.

The surface has gone through plating processing to enhance the electrical conductivity and corrosion resistance.

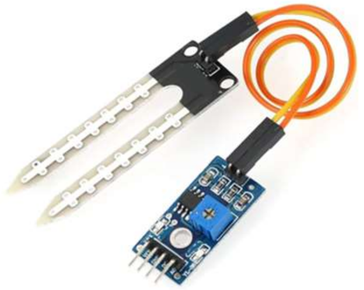
Water Flow Sensor

Flowmeter

ESP32 ESP-WROOM-32 NodeMCU Wi-Fi + Bluetooth Development Board CP2102 (38 Pins)

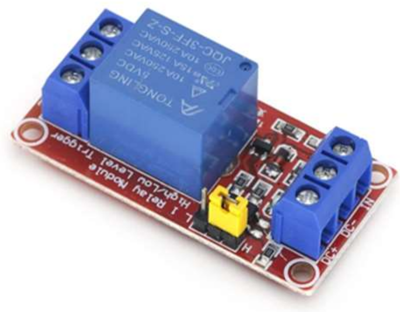
ESP32 can perform as a complete standalone system or as a slave device to a host MCU, reducing communication stack overhead on the main application processor. ESP with other systems to provide Wi-Fi and Bluetooth functionality through interface.

Soil Humidity Sensor Module



The Soil Moisture Sensor Module determines the amount of soil moisture by measuring the resistance between two metallic probes.

Opto-coupler relay 1 channel red

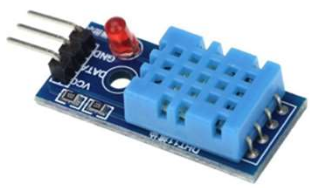


Relays are used wherever it is necessary to control a high-power or high-voltage circuit with a low-power circuit, especially when galvanic isolation is desirable.

Aquarium Pump 12V Model RS-385

Aquarium pumps influence agitation by displacing water with air bubbles.

DHT11 Digital Temperature and Humidity Sensor Module



DHT11 module is a Temperature and humidity sensing module, which uses Digital Signal Acquisition, which converts the Temperature and Humidity to a digital Reading, which can be easily read by a Microcontroller.

LM2596 DC-DC Step Down Module / dc-dc

The LM2596 Step-Down Regulator Module is widely used in various electronic applications and is capable of handling input voltages ranging from 4.5V to 40V.

To Power ESP32

04

Codes

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Define the pins for the sensors
const int sensorPin1 = 2; // Sensor 1 connected to pin 2
const int sensorPin2 = 3; // Sensor 2 connected to pin 3
const int Buzzer = 4; // Buzzer connected to pin 4
const int waterLevelPin = A0; // Water level sensor connected to analog pin A0

// Variables for sensor 1
volatile int flowRate1;
unsigned int flowMilliLitres1;
unsigned long totalMilliLitres1;
unsigned long oldTime1;

// Variables for sensor 2
volatile int flowRate2;
unsigned int flowMilliLitres2;
unsigned long totalMilliLitres2;
unsigned long oldTime2;

// Variables for water level sensor
int waterLevelValue;
float waterLevelPercentage;

// LCD configuration
LiquidCrystal_I2C lcd(0x27, 16, 2); // Address 0x27 for a 16x2 LCD
LiquidCrystal_I2C lcd2(0x26, 16, 2); // Address 0x26 for another 16x2 LCD
```

```
void setup() {
  // Initialize serial communication
  Serial.begin(9600);

  // Initialize LCD
  lcd.init();
  lcd.backlight();

  lcd2.init();
  lcd2.backlight();

  // Set the sensor pins as inputs
  pinMode(sensorPin1, INPUT);
  pinMode(sensorPin2, INPUT);
  pinMode(Buzzer, OUTPUT);
  digitalWrite(Buzzer, LOW);

  // Attach interrupts to the sensor pins
  attachInterrupt(digitalPinToInterrupt(sensorPin1), pulseCounter1, FALLING);
  attachInterrupt(digitalPinToInterrupt(sensorPin2), pulseCounter2, FALLING);

  // Initialize variables for sensor 1
  flowRate1 = 0;
  flowMilliLitres1 = 0;
  totalMilliLitres1 = 0;
  oldTime1 = 0;

  // Initialize variables for sensor 2
  flowRate2 = 0;
  flowMilliLitres2 = 0;
  totalMilliLitres2 = 0;
  oldTime2 = 0;
```

```
  // Display initial message on LCD
  lcd.setCursor(0, 0);
  lcd.print("Flow In:");
  lcd.setCursor(0, 1);
  lcd.print("Flow Out:");

  lcd2.setCursor(0, 0);
  lcd2.print("Level:");

}

void loop() {
  // Sensor 1 calculations
  unsigned long currentTime1 = millis();
  unsigned long elapsedTime1 = currentTime1 - oldTime1;

  if (elapsedTime1 >= 1000) {
    detachInterrupt(digitalPinToInterrupt(sensorPin1));
    flowRate1 = (1000 / (elapsedTime1 / 1000.0)) * (float) flowMilliLitres1 / 1000.0;
    flowMilliLitres1 = 0;
    oldTime1 = currentTime1;

    Serial.print("Flow rate In: ");
    Serial.print(flowRate1);
    Serial.println("L/m");

    lcd.setCursor(9, 0);
    lcd.print("      "); // Clear previous value
    lcd.setCursor(9, 0);
    lcd.print(flowRate1);
    lcd.print(" L/m");

    attachInterrupt(digitalPinToInterrupt(sensorPin1), pulseCounter1, FALLING);
  }
```

```
  // Sensor 2 calculations
  unsigned long currentTime2 = millis();
  unsigned long elapsedTime2 = currentTime2 - oldTime2;

  if (elapsedTime2 >= 1000) {
    detachInterrupt(digitalPinToInterrupt(sensorPin2));
    flowRate2 = (1000 / (elapsedTime2 / 1000.0)) * (float) flowMilliLitres2 / 1000.0;
    flowMilliLitres2 = 0;
    oldTime2 = currentTime2;

    Serial.print("Flow rate Out: ");
    Serial.print(flowRate2);
    Serial.println("L/min");

    lcd.setCursor(9, 1);
    lcd.print("      "); // Clear previous value
    lcd.setCursor(9, 1);
    lcd.print(flowRate2);
    lcd.print(" L/m");


    attachInterrupt(digitalPinToInterrupt(sensorPin2), pulseCounter2, FALLING);
  }
```

```
  // Convert analog value to percentage (mapping range from 0-1023 to 0-100)
  waterLevelPercentage = map(waterLevelValue, 5, 625, 0, 100);


  if (waterLevelPercentage >= 95) {
  digitalWrite(Buzzer, HIGH);
  }
  else {
      digitalWrite(Buzzer, LOW);

  }

  // Display water level percentage on LCD 2
  lcd2.setCursor(7, 0);
  lcd2.print("      "); // Clear previous value
  lcd2.setCursor(7, 0);
  lcd2.print(waterLevelPercentage);
  lcd2.print("%");

  Serial.print("Water Level:");
  Serial.print(waterLevelValue);
  Serial.println("%");

  delay(1000); // Adjust the delay as needed for updating the LCDs

}

void pulseCounter1() {
  flowMilliLitres1++;
  totalMilliLitres1++;
}

void pulseCounter2() {
  flowMilliLitres2++;
  totalMilliLitres2++;
}
```

Thank you!