AUTOMATED IRRIGATION SYSTEM WITH SOLAR PANEL

FINAL REPORT

WIN SEM 2022-23

ECS1002

(Raspberry phi using python)

Submitted by

P.Dhanush*(20BCI7319)*

Guided By

SUCHARITHA JACKSON

VIT AP-UNIVERSITY

AMARAVATI

ANDHRA PRADESH, INDIA

2022-2023

1.Introduction:

Irrigation is the artificial application of water for the success of crop production in the field. Agriculture is one of the most important sectors in India; around 60% of the total population still depends on agriculture for their livelihood. The sector is facing major perils due to various factors; one of them is lack of proper irrigation systems in the fields In a tropical monsoon country such as India, where the rainfall is scant and unreliable, and irrigation becomes an important agricultural input. Also, it is important to create a scientific and technically advanced irrigation method so that large tracts of land are not affected by salinity, alkalinity and water logging-the ill effects of outdated practices. Currently, there are many irrigation models which are coming up in this field, but their adoption rate is very less.As most of the systems involve high investment and are complex in nature.Hence, there is a need to create a low-cost, feasible system which can address the above issues and will benefit the small and medium scale farmer's.

2. Background Study:

According to the survey conducted by the Bureau of Electrical Energy in India in 2011 there are around 18 million agricultural pump sets and around 0.5 million new connections per year is installed with average capacity 5HP. Total annual consumption in agriculture sector is 131.96 billion KWh (19% of total electricity consumption).As cited in paper [1] solar powered smart irrigation technique is the future for the farmers and a solution for energy crisis. So for the proposed solar powered system we are using techniques analyzed in paper [2] and [4] and modified. Sine PWM technique has been used for inverter operation for minimum harmonics as given in paper [3] which further increases the efficiency of the system.The rating of the system was calculated corresponding to the pump specifications referring to paper [5].

3.Problem Definition:

The agriculture sector in India is going through a major crisis. One of the major reason for this crisis, is lack of technological advancement in the farming sector, only around 34% of the cultivated land is reasonably irrigated. This project mainly aims to address the issues by creating a model which can efficiently irrigate the crops, therebyreducing energy, labour and water consumption.

4.Objective:

This project mainly aims to address the issues by creating a model which can efficiently irrigate the crops, therebyreducing energy, labour and water consumption.And also it aims to implement and environment friendly solution for irrigating fields.

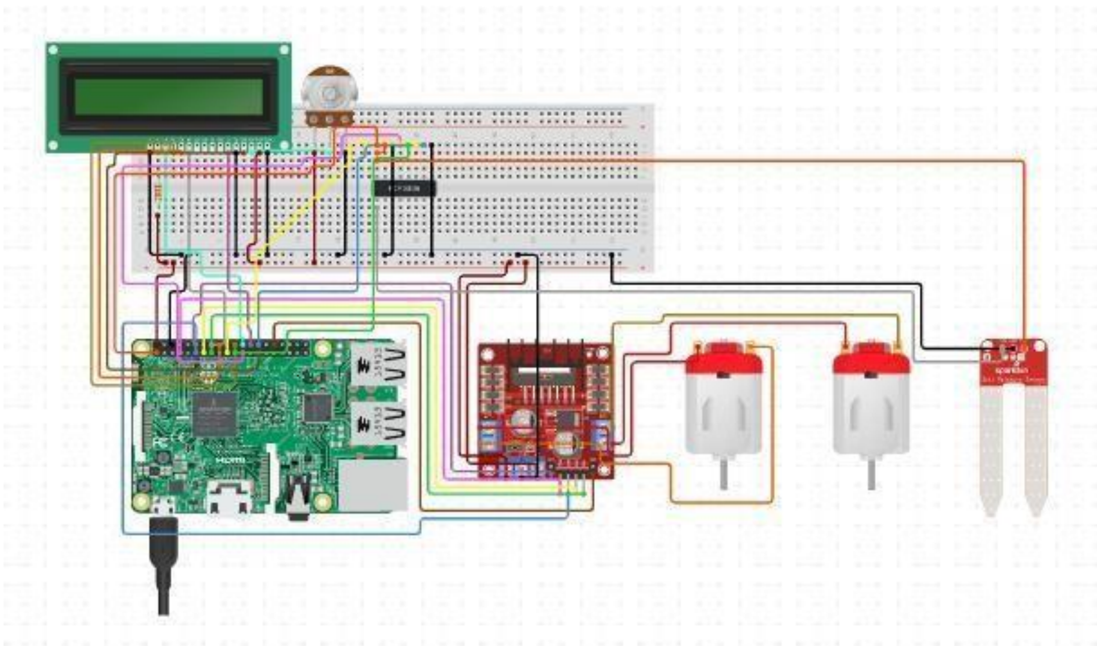
5. Methodology:

5.1.System Requirements:

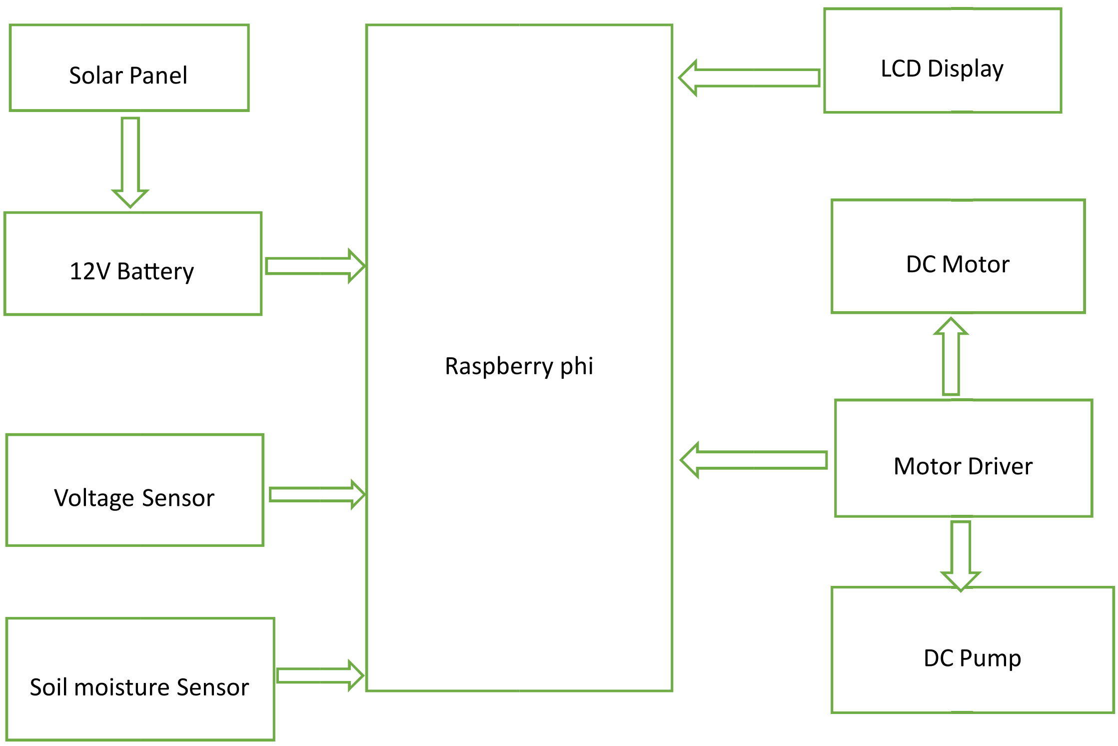
- Rasberry phi board
- Battery

- Soil Moisture Sensor
- Voltage Sensor
- Solar Panel
- DC Motor
- DC Pump
- Motor Driver ● LCD Display

5.2.Circuit Diagram:



5.3.Block Diagram:



5.4.Procedure:

Solar cell will generate energy from solar panel .These solar panel convert energy into electricity. These energies will get combine and pass to charge controller. It will control the amount of charge generated by solar panel and the energy is then stored in battery which is of 12V . Battery Level, water pump and soil moisture sensors which are connected to Raspberry phi.Wastage of water must be monitored in agricultural field by using automatic plant irrigation system.

Photo voltaic cell generates power from solar energy. Hence alternative form of electricity is introduced in irrigation. Small belt can be connected to DC motor and fan that fixed in the outlet of pump and generate electricity.

The whole system is controlled by microcontroller which monitors the sensors. The soil state will be identified by the sensors which in return microcontroller passes the command to relay driver IC. If the soil is in wet state the motor is turned off and vice versa. The microcontroller receives the signal from the sensors through the output of the op-amp, the software controls the signal which is stored in ROM of the microcontroller. The condition of the pump i.e., ON/OFF is displayed on a 16X2 LCD which is connected with the microcontroller.

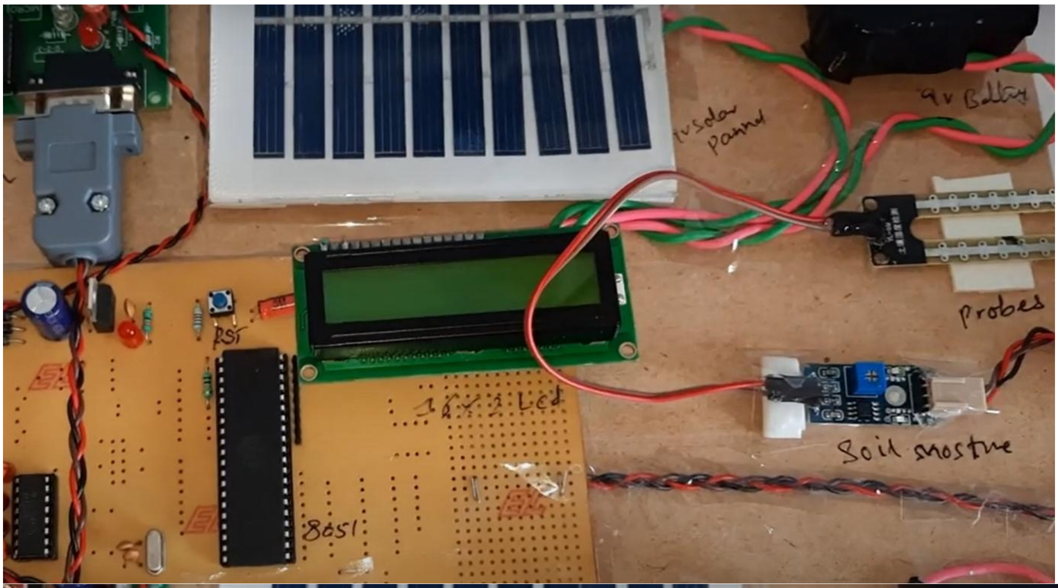
5.5.Advantages Over Other Traditional Methods:

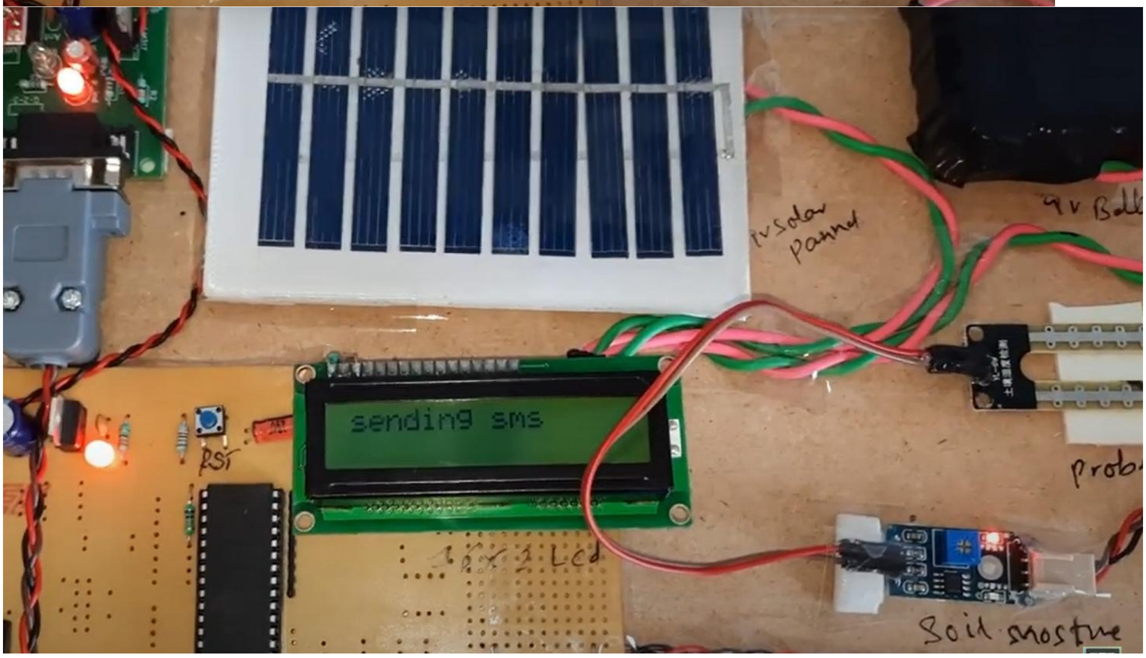
This framework can run the water siphon for a couple of hours without the sun.

- The put away energy can be utilized for different purposes.
- Soil dampness sensor assists with controlling the water siphon so the wastage of water can be diminished.
- Sensors utilized have high affectability and are not difficult to deal with.
- Low support and low force utilization.
- Can be utilized for various plant species by rolling out minor improvements in the encompassing ecological boundaries.
- Can be effortlessly changed for improving the arrangement and adding new highlights.
- A low- cost system, providing maximum mechanization

6. Results and Discussion:

While using DC motor we can achieve electricity with the wind and flow of water. And also solar panel used for the primary source of electricity. And soil moister level is updated in cloud. It can be enhance the electricity to make use in future purpose.





7. Conclusion and Future Scope:

7.1. Conclusion:

By implementing the proposed system there are various benefits for the government and the farmers. For the government a solution for energy crisis is proposed. By using the automatic irrigation system it optimizes the usage of water by reducing wastage and reduce the human intervention for farmers.The excess energy produced using solar panels can also be given to the grid with small modifications in the system circuit, which can be a source of the revenue of the farmer, thus encouraging farming in India and same time giving a solution for energy crisis. Proposed system is easy to imple ment and environment friendly solution for irrigating fields. The system was found to be successful when implemented for bore holes as they pump over the wholeday. Solar pumps also offer clean solutions with no danger of borehole contamination. The system requires minimal maintenance and attention as they are self starting. To further enhance the daily pumping rates tracking arrays can be implemented.

7.2. Future Scope:

The goal of this task is satisfied however there is some update that should be possible to make this water system framework more viable.

- GSM can be added for sending SMS to the versatile if happens any issue.
- Ambient temperature, light power, and dampness can be estimated.
- Weather update can be sent through SMS
- A sun oriented board can be utilized as a programmed sun beam global positioning framework.
- As the proposed framework is programmed, a ultrasonic sensor can be utilized to maintain a strategic distance from an impediments for wonderful activity.

8. Reference

Garg, H.P. 1987. Advances in solar energy technology, Volume 3. Reidel Publishing, Boston, MA.

[2] Halcrow, S.W. and Partners. 1981. Small-scale solar powered irrigation pumping systems: technical and economic review. UNDP Project GLO/78/004.Intermediate Technology Power, London, UK. A. Harmim et al., “Mathematical modeling of a box-type solar cooker employing an asymmetric compound parabolic concentrator,” Solar Energy, vol.86, pp. 1673–1682, 2012.

[3] K. K. Tse, M. T. Ho, H. S.-H. Chung, and S. Y. Hui, “A novel maximum power point tracker for PV panels using switching frequency modulation,” IEEE Trans. Power Electron., vol. 17, no. 6, pp. 980–989, Nov.2002. [4] Haley, M, and M. D. Dukes. 2007. Evaluation of sensor-based residential irrigation water application. ASABE 2007 Annual International Meeting, Minneapolis, Minnesota, 2007. ASABE Paper No. 072251.

[5] Prakash Persada, Nadine Sangsterb, Edward Cumberbatchc, AneilRamkhalawand andAatmaMaharajh, "Investigating the Feasibility of Solar Powered Irrigation for Food Crop Production: A Caroni Case," ISSN 1000 7924 The Journal of the Association of Professional Engineers of Trinidad and Tobago, Vol.40, No.2, pp.61-65, October/November 2011.

9.Appendix:

```
from machine import Pin, ADC, Timer
import time, os

#constants
PIN_PUMP_1 = 15
PIN_PUMP_2 = 14
PIN_LED_R_1 = 2
PIN_LED_R_2 = 3
PIN_LED_R_3 = 4
PIN_LED_R_4 = 5
PIN_LED_G_1 = 18
PIN_LED_G_2 = 19
PIN_LED_G_3 = 20
PIN_LED_G_4 = 21

PIN_BUTTON_1 = 6
PIN_BUTTON_2 = 17
```

```
PIN_WATER_LEVEL_SENSOR_1 = 0
PIN_WATER_LEVEL_SENSOR_2 = 1

WATERING_CYCLE_TIMES_ARRAY = [2, 4, 6, 8] #time in minutes

#global variables
selectionActive = True
dailyWateringCycles = 3
wateringCycleTime = 1 #choose from WATERING_CYCLE_TIMES
selectionTimer = Timer() lastWatering = time.time()

pump1 = Pin(PIN_PUMP_1, Pin.OUT)
pump2 = Pin(PIN_PUMP_2, Pin.OUT)

ledR1 = Pin(PIN_LED_R_1, Pin.OUT)
ledR2 = Pin(PIN_LED_R_2, Pin.OUT)
ledR3 = Pin(PIN_LED_R_3, Pin.OUT)
ledR4 = Pin(PIN_LED_R_4, Pin.OUT)
ledG1 = Pin(PIN_LED_G_1, Pin.OUT)
ledG2 = Pin(PIN_LED_G_2, Pin.OUT)
ledG3 = Pin(PIN_LED_G_3, Pin.OUT)
ledG4 = Pin(PIN_LED_G_4, Pin.OUT)

button1 = Pin(PIN_BUTTON_1, Pin.IN, Pin.PULL_DOWN)
button2 = Pin(PIN_BUTTON_2, Pin.IN, Pin.PULL_DOWN) wl1 =
Pin(PIN_WATER_LEVEL_SENSOR_1, Pin.IN, Pin.PULL_UP) wl2 =
Pin(PIN_WATER_LEVEL_SENSOR_2, Pin.IN, Pin.PULL_UP)


def checkKeys(): global selectionActive
    if(button1.value() or button2.value()):
    selectionActive = True
    selectionTimer.init(mode=Timer.ONE_SHOT
    , period=10000,
callback=disableSelection)

def disableSelection(t):
    global selectionActive
    selectionActive = False
    ledR1.value(0)
    ledR2.value(0)
    ledR3.value(0)
    ledR4.value(0)
    ledG1.value(0)
    ledG2.value(0)
    ledG3.value(0)
    ledG4.value(0)
    saveSettings()
    selectionTimer.init(mode=Timer.PERIODIC, freq=0.1,
callback=signOfLife)
```

```
def saveSettings():
    #saving settings try:
    os.remove('t.txt')
    except OSError: print("t.txt
        doesn't exists")
    try:
        os.remove('c.txt'
        )
    except OSError: print("t.txt
        doesn't exists")

    f = open('t.txt', 'w+')
    f.write(str(wateringCycleTime))
    f.close() f =
    open('c.txt', 'w+')
    f.write(str(dailyWateringCycles))
    f.close()

def loadSettings(): global wateringCycleTime,
    dailyWateringCycles try:
        f = open('t.txt') wateringCycleTime =
        int(f.read()) print('wateringCycleTime',
        wateringCycleTime) f.close()
    except OSError: print("t.txt
        doesn't exists")
    try:
        f = open('c.txt')
        dailyWateringCycles = int(f.read())
        print('dailyWateringCycles', dailyWateringCycles)
        f.close()
    except OSError: print("c.txt
        doesn't exists")
def displayWateringTimes():
    if(selectionActive):
        ledR4.value(1) if wateringCycleTime >= 0 else
ledR4.value(0) ledR3.value(1) if wateringCycleTime >=
        1 else
ledR3.value(0) ledR2.value(1) if wateringCycleTime >=
        2 else
ledR2.value(0) ledR1.value(1) if wateringCycleTime >=
        3 else
ledR1.value(0)


def displayWateringCycles():
    if(selectionActive):
        ledG1.value(1) if dailyWateringCycles >= 0 else
ledG1.value(0) ledG2.value(1) if dailyWateringCycles >=
        1 else
ledG2.value(0) ledG3.value(1) if dailyWateringCycles >=
        2 else
```

```
ledG3.value(0) ledG4.value(1) if dailyWateringCycles >=
        3 else
ledG4.value(0)


def checkAndModifySettings(): global
    wateringCycleTime, dailyWateringCycles
    if(button1.value()): time.sleep_ms(50)
    if(button1.value()):
            wateringCycleTime = (wateringCycleTime+1) if
wateringCycleTime < 3 else 0 displayWateringTimes()
time.sleep_ms(200)

    if(button2.value()):
        time.sleep_ms(50)
        if(button2.value()):
            dailyWateringCycles = (dailyWateringCycles+1) if
dailyWateringCycles < 3 else 0 displayWateringCycles()
time.sleep_ms(200)



def checkWatering(): global
lastWatering if((time.time() -
lastWatering) >
(86400/(dailyWateringCycles + 1))):
        wateringStart = time.time()
        if(checkWaterLevel()):
        pump1.value(1)
            while(time.time() - wateringStart <
(WATERING_CYCLE_TIMES_ARRAY[wateringCycleTime]*60)):
ledG1.value(1) time.sleep_ms(200) ledG1.value(0)
time.sleep_ms(200)
            pump1.value(0)
        lastWatering = time.time()


def signOfLife(t):
    ledR1.value(1)
    time.sleep_ms(10)
    ledR1.value(0)
    print('sign of life')

def checkWaterLevel(): return 0 if
    wl1.value() else 1


def checkWaterAndAlarm():
    if(checkWaterLevel() == 0):
        ledR4.value(1)
        time.sleep_ms(500)
```

```
        ledR4.value(0)
        time.sleep_ms(500)


#loading the settings from flash before entering the loop
loadSettings()
#disable LEDs after 10 seconds
selectionTimer.init(mode=Timer.ONE_SHOT, period=10000,
callback=disableSelection)

while(1):
    checkKeys()
    checkWatering()
    if(selectionActive):
        checkAndModifySettings()
        displayWateringTimes()
        displayWateringCycles()
    else:
    checkWaterAndAlarm()
    time.sleep_ms(50)
```