A PROJECT REPORT ON

“PRIORITY BASED SCHEDULING OF HOME APPLIANCES USING IOT”

Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of Engineering

In

Electrical and Electronics Engineering

By

KARTHIK M	**(17GAEE7026)**
MANJUNATH M	**(17GAEE7032)**
RAVINANDAN K	**(17GAEE7051)**
VINAYAK ASKI	**(16GAEE7076)**

Under the guidance of

Dr. MADHUSUDANA J

Associate Professor

Department of Electrical Engineering

University Visvesvaraya College of Engineering, K. R. Circle, Bengaluru – 560001

2020-2021

DEPARTMENT OF ELECTRICAL ENGINEERING

UNIVERSITY VISVESVARAYA COLLEGE OF ENGINEERING

K. R. CIRCLE, BENGALURU-560001

CERTIFICATE

This is to certify that the Project work entitled "PRIORITY BASED SCHEDULING OF HOME APPLIANCES USING IOT" has been successfully completed by KARTHIK M(17GAEE7026), MANJUNATH M (17GAEE7032), RAVINANDAN K (17GAEE7051), VINAYAK ASKI (16GAEE7076) bonafide students of University Visvesvaraya College of Engineering, K. R. Circle, Bengaluru, in partial fulfillment of the degree of Bachelor of Engineering in "Electrical and Electronics Engineering" of Bangalore University during the academic year 2020-21.

Signature of the Guide

Dr. MADHUSUDANA J

Associate Professor
Department of Electrical Engineering UVCE

K R Circle Bengaluru-560001

Signature of the Chairman

Dr. T. S. PRASANNA

Professor and Chairman
Department of Electrical Engineering, UVCE

K R Circle Bengaluru-560001

DECLARATION

This is to declare that this report has been written by us. No part of this report is plagiarized from other sources. All the information included from the other sources has been duly acknowledged. We aver that if any part of the report is found to be plagiarized, we shall take full responsibility for it.

KARTHIK M (17GAEE7026)

MANJUNATH M (17GAEE7032)

RAVINANDAN K (17GAEE7051)

VINAYAK ASKI (16GAEE7076)

CONTENTS

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task is incomplete without the mention of people who made it possible, whose constant guidance and encouragement crown all efforts and success.

We would like to express our profound gratitude to **Dr. T.S PRASANNA** , Professor and Chairman, Department of Electrical Engineering, UVCE, who has been a constant source of inspiration to our work.

We express our immense gratitude to **Dr. MADHUSUDANA J** , Associate Professor and Guide, Department of Electrical Engineering, UVCE, for his skillful guidance, constant supervision, timely suggestion and constructive criticism in successful completion of our project.

We also thank technical staff, office staff, our parents and friends for their help, encouragement and support.

ABSTRACT

Due to lack of situational awareness, automated analysis and the manual operating of switches, today's electric power grid is outdated and ill-suited for satisfying the demand for electricity, which is rapidly increasing in the past few decades. Besides, the global climate change and the greenhouse gas emissions on the Earth caused by the electricity generating stations, the ever growing population, equipment failures, energy storage problems, the capacity limitations of electricity generation, non-uniform consumption, decrease in fossil fuels, and resilience problems put more stress on the existing power grid. Consequently, many methods to reduce the demand side energy requirement has been developed. Demand side management is one such technique. Demand side management refers to a group of actions designed to mane and optimize a site's energy consumption and to cut costs. DSM has been considered as a means for reducing peak demand so that utilities can delay building further capacity. In this project we have implemented DSM for reducing the peak demand by Priority based scheduling of the home appliances using Arduino Uno Microcontroller.

LIST OF FIGURES

1. INTRODUCTION

1.1 DEFINITION (Demand side management)

Demand Side Management (DSM) is the modification of consumer demand for energy through various methods such as financial incentives and behavioral change through education.

Usually, the goal of demand-side management is to encourage the consumer to use less energy during peak hours, or to move the time of energy use to off-peak times such as nighttime and weekends. Peak demand management does not necessarily decrease total energy consumption, but could be expected to reduce the need for investments in networks and/or power plants for meeting peak demands. An example is the use of energy storage units to store energy during off-peak hours and discharge them during peak hours.

A new application for DSM is to aid grid operators in balancing constant generation from wind and solar units, particularly when the timing and magnitude of energy demand does not overlaps with the renewable generation. Generators brought on line during peak demand periods are often fossil fuel units. Minimizing their use reduces emissions of carbon dioxide and other pollutants.

The term Demand Side Management refers to a group of actions designed to manage and optimize a site's energy consumption and to cut costs, from grid charges to general system charges, including taxes.

The aim of Demand Side Management is to modify the overall consumption picture - consumption time profile, contractual supply parameters (contractual power and grid connection parameters) - in order to achieve savings in electricity charges.

As a result of the high penetration of renewable sources and the decentralization of production, grid managers in many countries are encountering increased instability on the grid and consequent disruptions to services. To limit these impacts and ensure a balance between energy consumption and the amount of power being fed into the grid, managers can now utilize generation and consumption systems that offer so-called "grid services" in return for payment, thus increasing the costs for the electrical system.

The first step in Demand Side Management is to carry out an in-depth analysis of on-site consumption: this clarifies the peculiarities of each individual site and whether consumption habits can be optimized without resorting to additional instruments. Whenever a change in habits is not feasible or simply not sufficient to achieve the desired cost reductions, the on-site (behind the meter) installation of the following solutions can be evaluated.

DSM TECHNIQUES

1.2.1 Peak Clipping

Fig 1.a

Peak Clipping is a form of waveform distortion, in Electric load management, the Peak Clipping is the process of implementing measures to reduce peak power demands on a system by direct control of customer loads by signals directed to customer appliances which is relatively simple and highly effective. Peak clipping technologies are those which cause a reduction in coincident demand at the time of system peak. Typically it is implemented by using direct load control (DLC) of appliances or devices by consumer action or by use of automated controls or communications.

1.2.2 Valley Filling

Fig 1.b

Valley filling technologies are those which increase the demand for electricity during off-peak daily or seasonal periods considered "valley" or low periods of demand. Typical technologies employed to fill valleys are electric

vehicles, battery energy storage as well as new space heating, cooling or domestic water heating integrated with storage or designed so as not to operate during on-peak periods.

1.2.3 Load Shifting Techniques

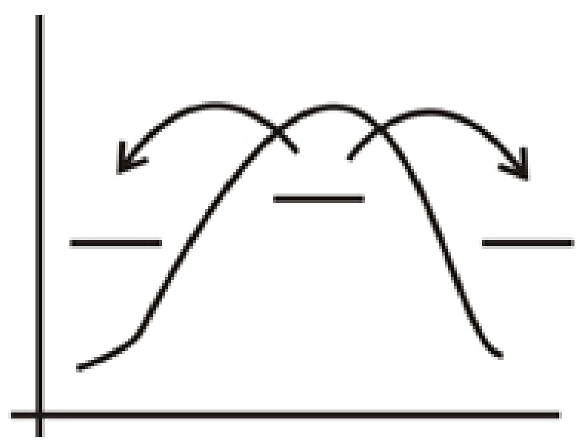
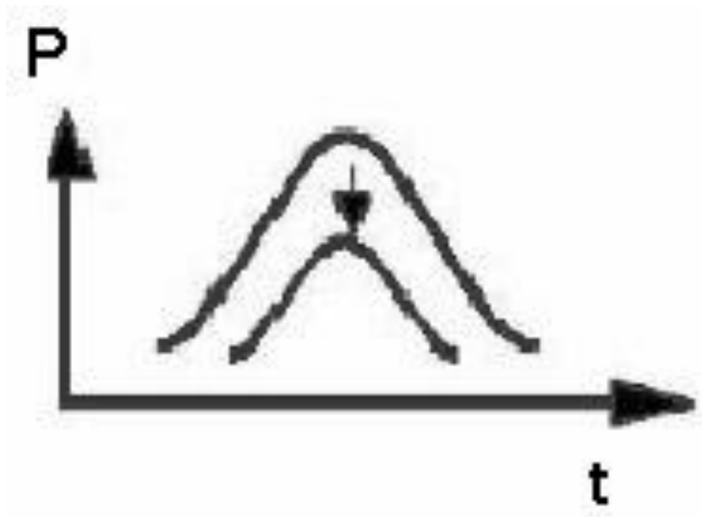


Fig 1.c

Load shifting technologies are those which facilitate moving or shifting existing loads to off-peak periods. Technologies involved often use process control to modify industrial operations or use electric energy storage or thermal energy storage for space heating, cooling or domestic water heating.

1.2.4 Energy Conservation



1.d

Energy efficient technologies are those which reduce overall energy needs while maintaining or improving the quality of energy services. Energy efficient technologies are high efficiency appliances or devices or involve the use of advanced building envelopes, fenestration, controls or ventilation.

2 .SYSTEM MODEL

In this project, the linear problem of scheduling appliances according to user prescribed priority is considered. A clear description of the appliance scheduling problem is provided, which is mathematically formulated as a linear programming optimization problem.

We make the following assumptions as part of the model:

1 . We consider one electricity source provider and neglect any aggregated sources like small scale solar panels etc.
2 .n is the number of appliances to be scheduled.
3 . Priority Pi for appliance is given by user in the form of an array. The priority of an appliance may change over time.
4 . For priorities from 1 to P, 1 is the lowest priority and P is considered the highest priority that any appliance can have.

Our main concern is to provide power to appliances given higher priority by user, so that within the defined power limit, we maximize the power consumption.
We need to find a solution for the following linear programming optimization problem, which is the basis of our problem.

max [summation p(i)]
such that [summation of x(i) <max_power_limit]

Equation (1) shows the optimization problem as a maximization of the sum of priorities given to an appliance as a vector Prio. The value of this indicator priority strictly lies in the range 1 to n, where n is the number of appliance. The next equation (2) is the constraint which says that the sum of the power required by the scheduled appliances should be strictly less than the defined power limit.

For choosing an appropriate power limit many factors should be considered like energy ratings of all the appliances in use, usage pattern, load factor, utilization factor, etc. Conducting a Preliminary Energy Audit would be required for selecting a suitable power limit. This power limit may change during different times of day depending on the peak hours in that region.

3. COMPONENTS

3.1. Arduino Uno

Fig 2.a

The Arduino Uno is a microcontroller board based on the ATmega328. It has 20 digital input/output pins (of which 6 can be used as PWM outputs and 6 can be used as analog inputs), a 16 MHz resonator, a USB connection, a power jack, an in-circuit system programming (ICSP) header, and a reset button.

3.2. 4 channel Relay module

Fig 2.b

A relay is a module used todrive a component, which uses large amount of current,using a small amount of current. It works at 5V, 10A.Relays can work either as switches or as amplifiers, to amplify the small amount of current generated by thesensors.

3.3 ACS712 CURRENT SENSOR

Fig 2.c

ACS712 is a current sensor that can operate on both AC and DC. The output of this current sensor is analog, so to read it, we can directly measure the output voltage using voltmeter or measure it by using a microcontroller like Arduino through Analog Read pin or ADC pin .

3.4 Light Detecting Resistor (LDR)

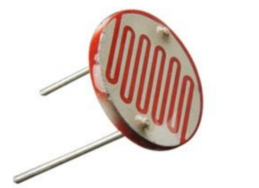


Fig 2.d

The LDR is aphotosensitive device used to measure light intensity.When in dark, the resistance is very high, but the resistance falls heavily when exposed to light.

3.5 16x2 LCD Display

Fig 2.e

The 16×2 LCD display is a very basic module commonly used in DIYs and circuits. The 16×2 translates o a display 16 characters per line in 2 such lines. In this LCD each character is displayed in a 5×7 pixel matrix.

3.6 HC-05 Bluetooth module

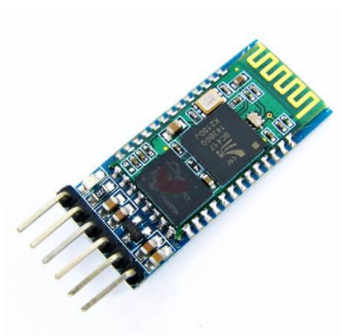


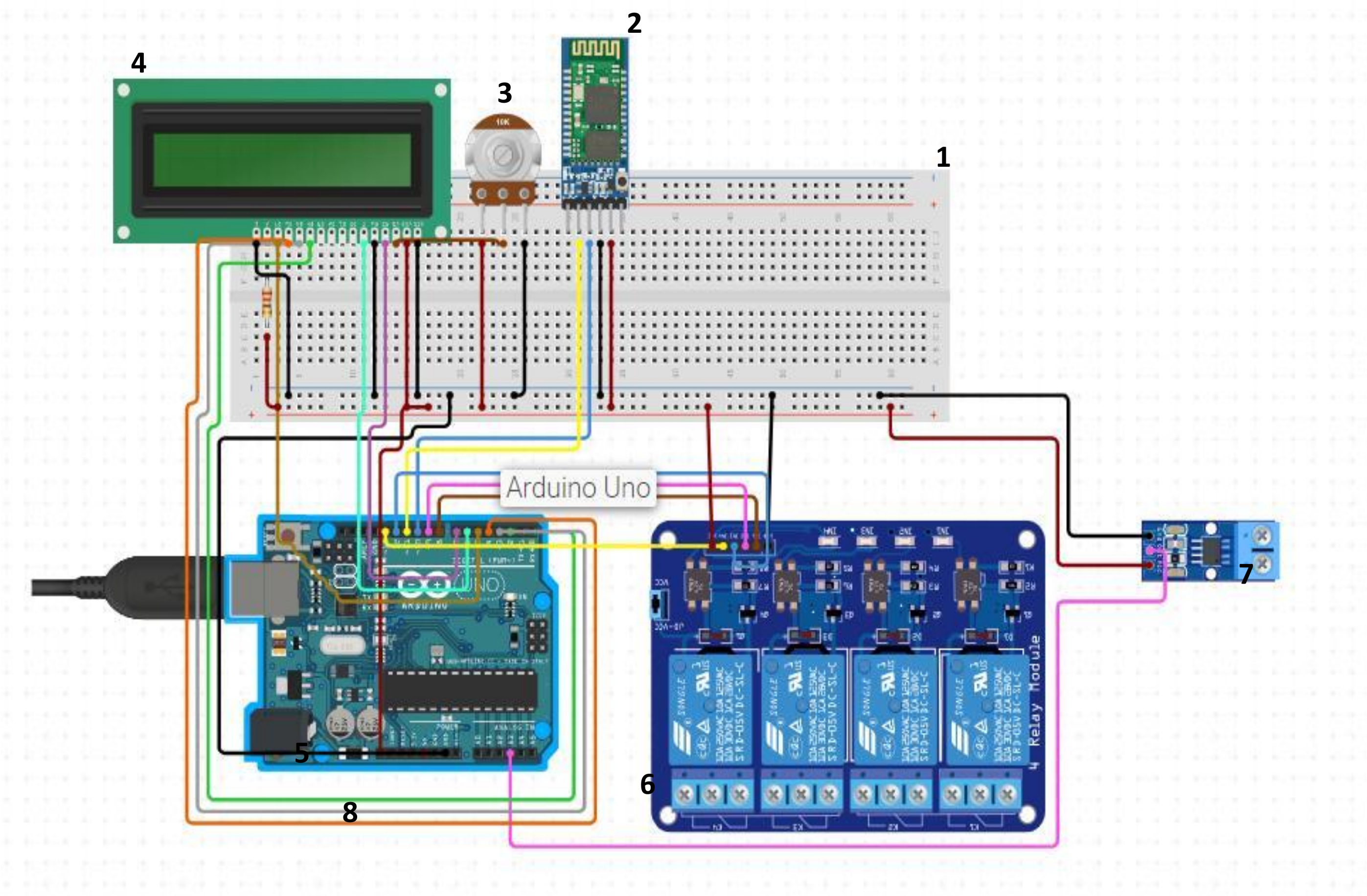
Fig 2.f

HC-05 is a Bluetooth module which can communicate in two way. Which means, It is full-duplex. We can use it with most micro controllers. It can be easily integrated into Arduino projects for connecting Arduino to a bluetooth enabled mobile phone.

3.7 Miscellaneous

Jumper wires, Breadboard, Potentiometer, Etc.

4. CIRCUIT DIAGRAM



Fig(3). MAIN CIRCUIT DIAGRAM

1. BREAD BOARD
2. HC-05 Bluetooth module
3. 10K POTENTIOMETER
4. 16x2 LCD Display
5.ARDUINO UNO
6. 4 channel Relay module
7. ACS712 CURRENT SENSOR
8. JUMPER WIRE

5. ALGORITHM

In this section, we describe the algorithm for scheduling of electrical home appliances based on their priority. We are provided with a number of appliances with variable priorities. The rating of the appliances may be provided directly or obtained through current sensors in real-time. The main objective here is that we need to turn ON appliances such as to maximize the power consumption in the defined limit. We can achieve this priority based switching with multiple approaches. Here we have designed, experimented and tested 2 such algorithms which are-

1. Naive approach(Qualitative priority)
2. Tabular optimization approach(Quantitative priority)

NAIVE APPROACH

For this approach, we will fix the priority of the outputs in advance, i.e, output relay 1 will have the highest priority and the priority decreases up to the last output.

5.1 ALGORITHM

1. We will loop through the output pins and turn ON as long as the total power is under the predefined limit.
2. The loop() function in Arduino will ensure that the switching is continuous even when some appliances are turned OFF by the user.
3. For the simulation purpose, we have concentrated mainly on the algorithm and thus have limited the circuit by excluding current sensors and connectivity modules.
4. The information about the status of an appliance and the real-time power consumption will be displayed.

```
void loop ()
{

  lcd.setCursor (0, 1);
  int i = 0;
  while (isum <= maxlim && i < 4)
    {
      lcd.clear ();
      lcd.print ("--DSM Running!--");
      lcd.setCursor (0, 1);
      if ((isum + OutPins[i]) <= maxlim)
    {
      digitalWrite (in[i], HIGH);
      isum += OutPins[i];
      lcd.print (i + 1);
      lcd.print (" Turned ON");
    }
      else
    {
      lcd.print (i + 1);
      lcd.print (" Turned OFF");
    }
      i += 1;
      delay (3000);
    }
  lcd.clear ();
  lcd.print ("------END------");
  delay (10000);
}
```

Pros:

- Easily implemented and straight forward approach.
- Linear time complexity and no extra space complexities, hence, faster real-time operation.

Cons:

- The priority of each output relay is fixed, therefore, to change the priority, the plugs have to manually swapped.
- Naive approach will not ensure maximum power utilization within the limit.

TABULAR OPTIMIZATION APPROACH

5.2 ALGORITHM

This approach is a modified form of 0-1 knapsack problem. The knapsack problem is a problem in combinatorial optimization: Given a set of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible. The problem often arises in resource allocation where the decision makers have to choose from a set of non-divisible projects or tasks under a fixed budget or time constraint, respectively.

- Our problem is similar to the knapsack problem in that we have to effectively allocate power by maximizing the summation of priority(Thus Quantitative) and also considering the power limit constraint.
- The maximum power limit is defined.
- The power ratings of all the appliances is either measured manually or provided in advance is stored in Power[] array.
- The priority ranging from 1 to n (number of appliances) is provided in Priority[] array.
- sum_of_power is declared and initialized to 0 which stores the value of the summation of the energy.
- An (n+1)x(n+1) 2 Dimensional array matrix dp[][] is created and all the elements are filled according to the optimization solution. The state of dp[i][j] will denote maximum value of 'j-sum_of_power'
- So if we consider power_limit we can fill it in all columns which have sum_of_power> Power[i]. Now two possibilities can take place:
    1. Fill Power[i] in the given column.
    2. Do not fill Power[i] in the given column.

Now we have to take a maximum of these two possibilities to fill the current state i.e, either turn this application ON or turn it OFF (if turned ON).

5.2.1 ALGORITHM FOR OPTIMIZED APPROACH

```
#include <stdio.h>
#include<iostream>
using namespace std;

int priorityDsm (int budget, int power[], int priority[], int n)
{
int dp[n + 1][budget + 1];
int zeroOne[n + 1][budget + 1], i, j;
int result[n];
for (int i = 0; i <= n; i++)
{
    for (int j = 0; j <= budget; j++)
    {dp[i][j] = 0;
    zeroOne[i][j] = 0;
    }
}
for (i = 0; i <= n; i++)
{
    zeroOne[i][0] = 1;
}
for (int i = 1; i <= n; i++)
{
    for (int j = 1; j <= budget; j++)
    {
        if (power[i - 1] > j)
        {
            dp[i][j] = dp[i - 1][j];
            zeroOne[i][j] = zeroOne[i - 1][j];
            }
        else
        {
            dp[i][j] = max (priority[i - 1] + dp[i - 1][j - power[i - 1]],dp[i -
 1][j]);
            zeroOne[i][j] = zeroOne[i - 1][j] || zeroOne[i - 1][j - power[i -
 1]];
        }
    }
}

for (i = 0; i < n; i++)
{
    result[i] = 0;
}
int r = n;
```

```
int c = budget;
int aAscii=65;
while (r > 0 && c > 0)
{
    if (dp[r][c] == dp[r - 1][c])
    {
        r--;
    }
    else
    {
        result[r - 1] = 1;
        c = c - power[r - 1];
        r--;
    }
}
cout<<endl<<endl<<endl;
cout<<"----------------Appliances that are currently running-----------“<<endl;
for (i = 0; i < n; i++)
{
    if (result[i])
    {
        cout << "Load " << (char)(aAscii+i) <<" of power rating "<<power[i]<< "W
is running...!!" << endl;
    }
}
return 0;
}

int main ()
{
    int budget,n;
    cout<<"Enter the number of appliances  : ";
    cin>>n;
    cout<<"Enter the maximum power limit in watts  : ";
    cin>>budget;
    int power[n];
    int priority[n];
    int temporaryPriorityArray[n];
    int prior;
    int individualPower;
    int asciiOfA=65;
    int ans[n];
    for(int i=0;i<n;i++){

        cout<<"Enter the power rating of Load "<<(char)asciiOfA<<" in watts  : ";

        cin>>individualPower;
```

```
      if (individualPower<=0 || individualPower>=budget){
             cout<<"Enter a proper power rating of the machine, it must be between
 1W and " <<budget <<", try again by rerunning the program";
             return 0;
         }
         else{
             power[i]=individualPower;
         }
         cout<<"Enter the priority of Load : "<<(char)asciiOfA;
         cin>>prior;
         if(prior>=1 && prior<=n){temporaryPriorityArray[i]=prior;}
         else{cout<<"Enter a proper priority.....!!! It must be between 1 and "<<n
;return 0;}
         asciiOfA++;

     }
     for(int i=0;i<n;i++){
         priority[i]=n-temporaryPriorityArray[i]+1;
     }

     int answer = priorityDsm (budget, power, priority, n);

     cout<<"\nAppliances are chosen by maximizing their priority such that total p
ower is below the limit.\n";
     cout<<"Project done by :\n";
     cout<<"1.Karthik M \n2.Manjunath M \n3.Ravinandan K \n4.Vinayak Aski \n Guide
d by : Prof. Madhusudan J \n Thank you.";
}
```

5.2.2 Complexity analysis of the algorithm:

- Time complexity: O(n * Power in Watts).

Where n is the number of appliances connected to the relay.

- Auxiliary space: O((n+1) *(Power in Watts +1))
- For creating the extra 2D array for tabulation.[dp[][] Tabulation result here]

Pros:

- Ensures maximum power utilization within the limit therefore quantitative in nature.
- Best approach if we also consider the time limit of operation for each appliance.

Cons:

- Maximizes the summation of priority. Therefore if the sum of priorities of allthe appliance except the first priority appliance is higher, then even the appliance with the maximum priority will not be turned ON.
- Due to extra tabulation, the time and space complexity is higher. Therefore slow in operation.

6 .SIMULATION TEST RESULTS

6.1 TEST 1

```
Enter the number of appliances  : 5
Enter the power budget in watts  : 2000
Enter the power rating of 1th appliance in watts  : 900
Enter the priority of 1th appliance, should be between 1 and 5, 1 being the lowest prior appliance and n being the highest prior appliance : 5
Enter the power rating of 2th appliance in watts  : 1200
Enter the priority of 2th appliance, should be between 1 and 5, 1 being the lowest prior appliance and n being the highest prior appliance : 2
Enter the power rating of 3th appliance in watts  : 400
Enter the priority of 3th appliance, should be between 1 and 5, 1 being the lowest prior appliance and n being the highest prior appliance : 3
Enter the power rating of 4th appliance in watts  : 1600
Enter the priority of 4th appliance, should be between 1 and 5, 1 being the lowest prior appliance and n being the highest prior appliance : 1
Enter the power rating of 5th appliance in watts  : 600
Enter the priority of 5th appliance, should be between 1 and 5, 1 being the lowest prior appliance and n being the highest prior appliance : 4


<------Appliances that are currently running------>
Appliance.1 of power rating 900W is running...!!
Appliance.3 of power rating 400W is running...!!
Appliance.5 of power rating 600W is running...!!
```

Test results 1

In the above test result , there are 5 appliances of power rating 900W ,1200W , 400W, 1600W, 600W are there with the priority of 5 , 2 , 3 , 1 , 4 respectively (5 being the highest prior appliance and 1 being the lowest prior appliance).

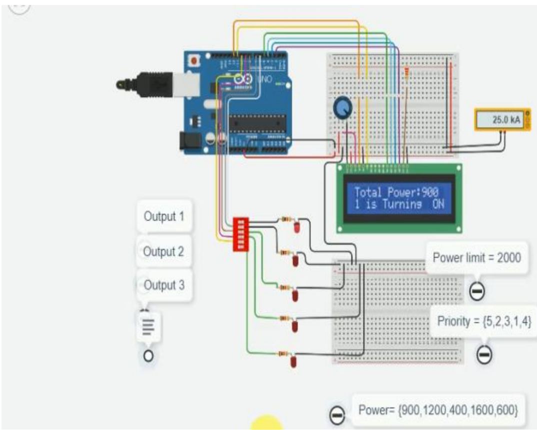
The Budget given is 2000W, which should not be exceeded at any situations , out of 5 appliances , the main intention of this project is to select those appliances such that the power consumption should not exceed the budget defined and the most prior appliance is given chance at the first place , since the budget is 2000 and we have to make best use of that we used 0-1 knapsack algorithm out of which we got maximized the power consumption below the specified budget.

Here the algorithm have chosen appliance 1 , 3 , 5 out of 5 appliance which are at highest priority and the sum of power consumption of these appliances lies within the given budget

The optimized algorithm automatically selects the appliances such that overall power consumption.

In the Algorithm , the result[ ] array is a binary array which will have the data about whether that particular appliance at that index should be on/off .

6.1.1 SIMULATION RESULT



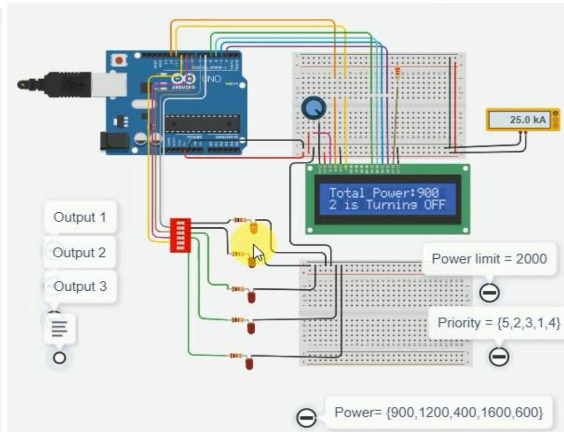
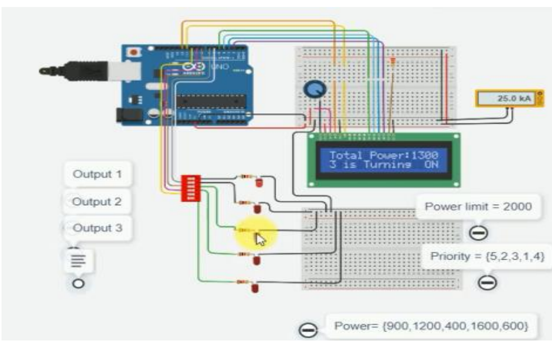


Fig (3.a) Appliance 1 Turning On

Fig (3.b) Appliance 2 Turning Off



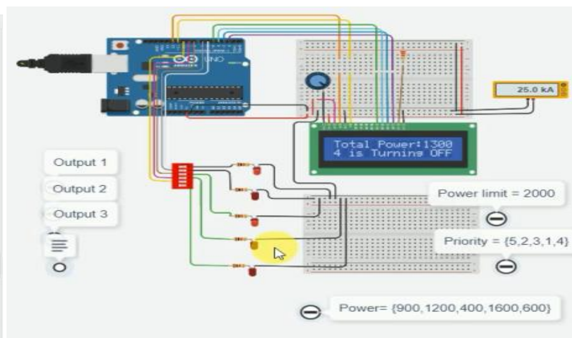


Fig (3.a) Appliance 3 Turning On

Fig (3.b) Appliance 4 Turning Off

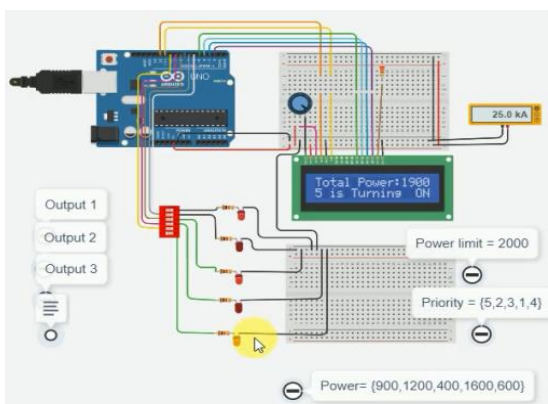


Fig (3.e) Appliance 5 Turning On

6.2 TEST 2

```
Enter the number of appliances  : 5
Enter the power budget in watts  : 5000
Enter the power rating of 1th appliance in watts  : 1800
Enter the priority of 1th appliance, should be between 1 and 5, 1 being the lowest prior appliance and n being the highest prior appliance : 2
Enter the power rating of 2th appliance in watts  : 2500
Enter the priority of 2th appliance, should be between 1 and 5, 1 being the lowest prior appliance and n being the highest prior appliance : 1
Enter the power rating of 3th appliance in watts  : 700
Enter the priority of 3th appliance, should be between 1 and 5, 1 being the lowest prior appliance and n being the highest prior appliance : 4
Enter the power rating of 4th appliance in watts  : 1200
Enter the priority of 4th appliance, should be between 1 and 5, 1 being the lowest prior appliance and n being the highest prior appliance : 3
Enter the power rating of 5th appliance in watts  : 400
Enter the priority of 5th appliance, should be between 1 and 5, 1 being the lowest prior appliance and n being the highest prior appliance : 5


<------Appliances that are currently running------>
Appliance.1 of power rating 1800W is running...!!
Appliance.3 of power rating 700W is running...!!
Appliance.4 of power rating 1200W is running...!!
Appliance.5 of power rating 400W is running...!!


...Program finished with exit code 0
Press ENTER to exit console.
```

Test results 2

In the above test result , there are 5 appliances of power rating 1800W ,2500W , 700W, 1200W , 400W are there with the priority of 2 , 1 , 4 , 3 , 5 respectively (5 being the highest prior appliance and 1 being the lowest prior appliance).

The Budget given is 5000W, which should not be exceeded at any situations , out of 5 appliances , the main intention of this project is to select those appliances such that the power consumption should not exceed the budget defined and the most prior appliance is given chance at the first place , since the budget is 5000W and we have to make best use of that we used 0-1 knapsack algorithm out of which we got maximized the power consumption below the specified budget.

Here the algorithm have chosen appliance 1 , 3 ,4 , 5 out of 5 appliance which are at highest priority and the sum of power consumption of these appliances lies within the given budget

The optimized algorithm automatically selects the appliances such that overall power consumption.

In the Algorithm , the result[ ] array is a binary array which will have the data about whether that particular appliance at that index should be on/off .

6.3 TEST 3

```
Enter the number of appliances  : 3
Enter the power budget in watts  : 1000
Enter the power rating of 1th appliance in watts  : 700
Enter the priority of 1th appliance, should be between 1 and 3, 1 being the lowest prior appliance and n being the highest prior appliance : 2
Enter the power rating of 2th appliance in watts  : 500
Enter the priority of 2th appliance, should be between 1 and 3, 1 being the lowest prior appliance and n being the highest prior appliance : 3
Enter the power rating of 3th appliance in watts  : 400
Enter the priority of 3th appliance, should be between 1 and 3, 1 being the lowest prior appliance and n being the highest prior appliance : 1


<------Appliances that are currently running------>
Appliance.2 of power rating 500W is running...!!
Appliance.3 of power rating 400W is running...!!


...Program finished with exit code 0
Press ENTER to exit console.
```

Test results 3

In the above test result , there are 3 appliances of power rating 700W ,500W , 400W are there with the priority of 2 , 3 , 1 respectively (3 being the highest prior appliance and 1 being the lowest prior appliance).

The Budget given is 1000W, which should not be exceeded at any situations , out of 5 appliances , the main intention of this project is to select those appliances such that the power consumption should not exceed the budget defined and the most prior appliance is given chance at the first place , since the budget is 1000W and we have to make best use of that we used 0-1 knapsack algorithm out of which we got maximized the power consumption below the specified budget.

Here the algorithm have chosen appliance 2 , 3 out of 5 appliance which are at highest priority and the sum of power consumption of these appliances lies within the given budget

The optimized algorithm automatically selects the appliances such that overall power consumption.

In the Algorithm , the result[ ] array is a binary array which will have the data about whether that particular appliance at that index should be on/off .

6.4 TEST 4

```
Enter the number of appliances  : 4
Enter the power budget in watts  : 3000
Enter the power rating of 1th appliance in watts  : 900
Enter the priority of 1th appliance, should be between 1 and 4, 1 being the lowest prior appliance and n being the highest prior appliance : 3
Enter the power rating of 2th appliance in watts  : 1800
Enter the priority of 2th appliance, should be between 1 and 4, 1 being the lowest prior appliance and n being the highest prior appliance : 2
Enter the power rating of 3th appliance in watts  : 500
Enter the priority of 3th appliance, should be between 1 and 4, 1 being the lowest prior appliance and n being the highest prior appliance : 4
Enter the power rating of 4th appliance in watts  : 300
Enter the priority of 4th appliance, should be between 1 and 4, 1 being the lowest prior appliance and n being the highest prior appliance : 1


<------Appliances that are currently running------>
Appliance.1 of power rating 900W is running...!!
Appliance.3 of power rating 500W is running...!!
Appliance.4 of power rating 300W is running...!!


...Program finished with exit code 0
Press ENTER to exit console.
```

Test results 4

In the above test result , there are 4 appliances of power rating 900W ,1800W , 500W, 300W are there with the priority of 2 , 3 , 4, 1 respectively (4 being the highest prior appliance and 1 being the lowest prior appliance).

The Budget given is 3000W, which should not be exceeded at any situations , out of 5 appliances , the main intention of this project is to select those appliances such that the power consumption should not exceed the budget defined and the most prior appliance is given chance at the first place , since the budget is 3000W and we have to make best use of that we used 0-1 knapsack algorithm out of which we got maximized the power consumption below the specified budget.

Here the algorithm have chosen appliance 1 , 3 , 4 out of 5 appliance which are at highest priority and the sum of power consumption of these appliances lies within the given budget

The optimized algorithm automatically selects the appliances such that overall power consumption.

In the Algorithm , the result[ ] array is a binary array which will have the data about whether that particular appliance at that index should be on/off .

6.5 TEST 5

```
Enter the number of appliances  : 4
Enter the power budget in watts  : 800
Enter the power rating of 1th appliance in watts  : 200
Enter the priority of 1th appliance, should be between 1 and 4, 1 being the lowest prior appliance and n being the highest prior appliance : 1
Enter the power rating of 2th appliance in watts  : 500
Enter the priority of 2th appliance, should be between 1 and 4, 1 being the lowest prior appliance and n being the highest prior appliance : 2
Enter the power rating of 3th appliance in watts  : 70
Enter the priority of 3th appliance, should be between 1 and 4, 1 being the lowest prior appliance and n being the highest prior appliance : 4
Enter the power rating of 4th appliance in watts  : 300
Enter the priority of 4th appliance, should be between 1 and 4, 1 being the lowest prior appliance and n being the highest prior appliance : 3


<------Appliances that are currently running------>
Appliance.1 of power rating 200W is running...!!
Appliance.3 of power rating 70W is running...!!
Appliance.4 of power rating 300W is running...!!


...Program finished with exit code 0
Press ENTER to exit console.
```

Test results 5

In the above test result , there are 4 appliances of power rating 200W ,500W , 70W, 300W are there with the priority of 1 , 2 , 4 , 3 respectively (4 being the highest prior appliance and 1 being the lowest prior appliance).

The Budget given is 800W , which should not be exceeded at any situations , out of 4 appliances , the main intention of this project is to select those appliances such that the power consumption should not exceed the budget defined and the most prior appliance is given chance at the first place , since the budget is 800W and we have to make best use of that we used 0-1 knapsack algorithm out of which we got maximized the power consumption below the specified budget.

Here the algorithm have chosen appliance 1 , 3 , 4 out of 4 appliance which are at highest priority and the sum of power consumption of these appliances lies within the given budget

The optimized algorithm automatically selects the appliances such that overall power consumption.

In the Algorithm , the result[ ] array is a binary array which will have the data about whether that particular appliance at that index should be on/off .

7.CODE REPOSITORY AND OTHER LINKS

7.1 Scan these to access code on GitHub :

7.1.1Click here(GitHub)

7.1.2 Click Here(TinkerCad)

7.2 Scan the following Code to access Simulation Video

7.2.1Click Here(Youtube)

SCOPE

Demand side management has been traditionally recognized as are of major intervention to achieve reduction in energy demands while ensuring continuous development, DSM has became an integral part of almost all the actual and state missions in promotion of energy efficiency. DSM interactions will reduce the peak electricity demand which is the main aim of our project.

We can use this device in home automation to enable remote monitoring and management of appliances and systems, such as lighting and heating. We can connect this device to the cloud to share real-time consumer's usage data and can automate actions based on the home owner's preferences.

By integrating smart lighting systems with this device, it can be made to detect when occupants are in the room and adjust lighting as needed. Integrating smart thermostats will allow the users to schedule, monitor and remotely control home temperatures. When the residents are away from the home, pet care can be automated with connected feeders. Home plants and lawns can be watered by the way of connected timers.

The implementation of AMI is widely seen as the first step in the digitalization of the electric grid and control systems. This device will enable two-way communication between the consumers and utility companies.

CONCLUSION

The trend of energy consumption all over the world is increasing rapidly. Since the supply of fossil fuels for the generation is limited and is bound to deplete, Demand side power management becomes crucial.Energy production by many sources, especially wind, PV, and other renewable sources, is generally variable, which affects the balance between demand and supply. It is not sufficient to merely assume supplementary sources whenever there is a shortfall, but rather, it is desirable that a scheduling approach for appliances be used that allows the time-varying nature of such sources. The wind energy conversion system generate its peak energy during midnight and solar panels generate its peak energy at noon, both of which are off peak hours.

Thus, to address this problem we are taking into consideration the changing
power budget from such sources as well as the needs of the system consuming power. We give a priority-based scheduling algorithm for home appliances
using a Linear Programming formulation. According to our problem formulation and algorithm, the appliances with highest priority and satisfying the maximum power limit constraint get required power, with other appliances scheduled
later, with priority ties also being handled.

To further extend this work, scheduled appliances can be further categorized as interruptible and non-interruptible. To handle each and every socket of a house, the algorithm can be enhanced. Time of the day tariff can be implemented with the tariff varying in real-time depending on the over demand in the distribution system. The whole system can be made to operate on cloud platform thus decentralizing the distribution and enabling two-way communication between the consumer and the power utility which is conceptually a smart meter.

REFERENCES

1.www.intechopen.com/books/smart-metering-technology-and-services-inspirations-for-energy-utilities/advanced-metering-infrastructure-based-on-smart-meters-in-smart-grid

2. "DSM | Bureau of Energy Efficiency", beeindia.goc.in.

3. "Smart meter - Wikipedia".

4. Konstantinos Kotsalos, "On the development of a framework for the advanced monitoring of LV grids"

5. "What is Demand Side Management", corporate.enelx.com.

6. S. Rao, "A foundation of demand side resource management in distributed systems," Trans. on Comput. Sci., vol. VIII, pp. 114–126, Oct.
2010, doi:10.1007/978-3-642-16236-7 8.

7. L. C. N. Li and S. H. Low, "Optimal demand response based on utility maximization in power networks," in IEEE Power & Energy Society General Meeting, Detroit, MI, July 2011, doi:10.1109/PES.2011.6039082.

8 . Knapsack Algorithm, "GeeksForGeeks" ,0-1 Knapsack Problem