Project Report

On

**Automated Iron Box**

**Under the guidance of:**

## Ms.C Aishwarya and Mr. Gautam Amiya

**Submitted by:**

Vybhav Pai(171CO252)

Shrinidhi Anil Varna(171CO145)

Rakshith S(171CO233)



BACHELORS OF TECHNOLOGY

NITK-CSE DEPARTMENT (JULY-NOVEMBER 2018)

**Abstract**

The objective of the project is to simulate the working of an iron box. Taking various factors like power supply,temperature range for a specific cloth material,iron plate material and its rate of heating we have calculated the amount of time the iron box should be ON and the amount of time it should be OFF. The settings can be changed at the end of each cycle(from current temperature to maximum temperature and then to its minimum temperature) of a particular cloth setting.

The code has been modularized to different settings of the iron box and the previous state is always recorded for future use. Each module is returning its current state of setting to the main module.

**Important terminologies** :

1.Cycle : the total time duration in which the temperature of the iron box changes from its current value to the maximum value of a particular setting and then cool down to its minimum value.

2.ON : when the iron box draws power and heats the iron plate.

3.OFF : when the iron box does not draw power, to cool down OR the power supply is cut.

Once the iron box has been completely used for a required purpose , a waveform of the cycles that the iron box went through while being in a particular state is printed. An option has been kept where waveform at any instant of time can be printed ,if required .

* Table containing temperature range for a different cloth materials:

|  |  |  |
| --- | --- | --- |
| **Cloth** **type** | **Minimum Temperature(in degree C)** | **Maximum Temperature(in degree C)** |
| Nylon | 131 | 139 |
| Sequential Fabric | 139 | 144 |
| Wool | 144 | 152 |
| Cotton | 200 | 208 |
| Linen | 226 | 234 |

* *Different formulae and constants used to calculate the per degree rise and per degree fall time*

Time taken to reach highest temperature(heating time),t

**t =** seconds.

**t** = time taken to heat ∆T degree celsius (in seconds),

**m** = mass of iron plate (in kilograms)

**S** = specific heat of the aluminium plate (in KJ/kg/K)

**∆T** = temperature difference (in degree celsius)

**P** = power supply (in MW)

* *Setting shift time*

The setting can be shifted only at the end of each cycle ,when transiting from one setting to another. The different possibilities are as follows :

With reference to the *minimum temperature of the previous state* if:

(i) the *maximum temperature of next state is lesser than it*

- then cool from previous minimum temperature to the current maximum temperature and from there on cool to its minimum temperature

(ii) the *minimum temperature of next state is greater than it*

- then heat from previous minimum to the current minimum temperature and from there on heat up to its maximum temperature and from here on cool down to its minimum temperature

(iii) the *minimum temperature of next state is lesser than it*

- then heat it from the previous minimum temperature up to the current maximum temperature and cool down to the current minimum temperature

* By experimentation , it was found that the heating rate is 2.5 times the cooling rate .

**Note** : If the setting is not changed then the iron box takes 6s to heat in its ON state and 15s to cool in its OFF state.

**Code**

#include<stdio.h>

#include<stdlib.h>

char wave[10000000]; //character array to store ON and OFF signals during the process

int k=0;

int t[6][6]; //two dimensional array used to store time difference between setting transitions (heating and cooling)

int nylon(int state)

{

printf("Nylon:\n");

int i,cur = 1;

if(t[state][cur] >= 0)

{

for(i=1;i<=t[state][cur];i++,k++)

wave[k] = '\_';

for(i=1;i<=15;i++,k++)

wave[k] = '\_';

return cur;

}

if(t[state][cur] < 0)

{

for(i=t[state][cur];i<=0;i++,k++)

wave[k] = '\_';

for(i=1;i<=15;i++,k++)

wave[k] = '\_';

return cur;

}

}

int seq(int state)

{

printf("Sequential fabric:\n");

int i,cur = 2;

if(t[state][cur] >= 0)

{

for(i=1;i<=t[state][cur];i++,k++)

wave[k] = '-';

for(i=1;i<=15;i++,k++)

wave[k] = '\_';

return cur;

}

if(t[state][cur] < 0)

{

for(i=t[state][cur];i<=0;i++,k++)

wave[k] = '\_';

for(i=1;i<=15;i++,k++)

wave[k] = '\_';

return cur;

}

}

int wool(int state)

{

printf("Wool:\n");

int i,cur = 3;

if(t[state][cur] >= 0)

{

for(i=1;i<=t[state][cur];i++,k++)

wave[k] = '-';

for(i=1;i<=15;i++,k++)

wave[k] = '\_';

return cur;

}

if(t[state][cur] < 0)

{

for(i=t[state][cur];i<=0;i++,k++)

wave[k] = '\_';

for(i=1;i<=15;i++,k++)

wave[k] = '\_';

return cur;

}

}

int linen(int state)

{

printf("Linen:\n");

int i,cur = 4;

if(t[state][cur] >= 0)

{

for(i=1;i<=t[state][cur];i++,k++)

wave[k] = '-';

for(i=1;i<=15;i++,k++)

wave[k] = '\_';

return cur;

}

if(t[state][cur] < 0)

{

for(i=t[state][cur];i<=0;i++,k++)

wave[k] = '\_';

for(i=1;i<=15;i++,k++)

wave[k] = '\_';

return cur;

}

}

int cotton(int state)

{

printf("Cotton\n");

int i,cur = 5; // current state in which it is at the moment

if(t[state][cur] >= 0)

{

for(i=1;i<=t[state][cur];i++,k++)

wave[k] = '-';

for(i=1;i<=15;i++,k++)

wave[k] = '\_';

return cur;

}

if(t[state][cur] < 0)

{

for(i=t[state][cur];i<=0;i++,k++)

wave[k] = '\_';

for(i=1;i<=15;i++,k++)

wave[k] = '\_';

return cur;

}

}

void display()

{

printf("Waveform:-\n");

printf("%s",wave);

printf("\n\n\n\n");

}

void main()

{

int ch,i,prev; //below are the calculated time intervals for different setting transitions in the form t[previous\_setting][current\_setting]

t[0][0] = 0 , t[1][1] = 6,t[2][2] = 6,t[3][3] = 6,

t[4][4] = 6, t[5][5] = 6, t[1][2] = 12,t[1][3] = 15,

t[1][4] = 57,t[1][5] = 76,t[2][3] = 10,t[2][4] = 51,

t[2][5] = 70,t[3][4] = 47,t[3][5] = 67,t[4][5] = 25,

t[2][1] = 0,t[3][1] = -10,t[4][1] = -113,t[5][1] = -162,

t[3][2] = 2,t[4][2] = -99,t[5][2] = -147,t[4][3] = -87,

t[5][3] = -137,t[5][4] = -33;

printf("\n\t\t\t\tState: ON\n\n");

printf("Options:\n1.Nylon\n2.Sequential fabrics\n3.Wool\n4.Linen\n5.Cotton\n6.Display waveform\n7.OFF\n");

scanf("%d",&ch); // ch is choice of cloth material

i=0;

for(;;)

{

if(i==0)

prev = ch;

if(i++!=0)

{

printf("\nShift setting to :\n");

scanf("%d",&ch);

}

switch(ch)

{

case 1:

{

prev = nylon(prev); // here we are storing the previous state of the iron box so that we can feed this information to the next state

printf("Your current state is: 1.Nylon\n");

break;

}

case 2:

{

prev = seq(prev);

printf("Your current state is: 2.Sequential fabrics\n");

break;

}

case 3:

{

prev = wool(prev);

printf("Your current state is: 3.Wool\n");

break;

}

case 4:

{

prev = linen(prev);

printf("Your current state is: 5.Linen\n");

break;

}

case 5:

{

prev = cotton(prev);

printf("Your current state is: 6.Cotton\n");

break;

}

case 6:

{

display(); /*\*to display the waveform generated during the process at any posedge of a wave signal\**/

break;

}

case 7:

{

printf("\nState: OFF\n");

display();

exit(0);

break;

}

}

}

}

**Conclusion**

Here an iron box has been automated using various parameters as mentioned earlier and has successfully implemented using C code.

One limitation that can be noted is that the transition of states is observed at the end of each cycle and not at any instant of time which doesn’t completely simulate the smooth functioning of an iron box. Thus limiting the main idea to a discrete one.

**References**

1) [www.thespruce.com](http://www.thespruce.com/)

2) [www.wikipedia.org](http://www.wikipedia.org/)

3) [www.Sciencing.com](http://www.Sciencing.com/)