EEL 4924 Electrical Engineering Design
(Senior Design)

Final Design Report
May 2011

Project Title:
The Electric Split

Team Name:
ESplit

Team Members:

Shahin Pourkaviani *Shahin9@ufl.edu*

Vivek Tumrukota *vz685@ufl.edu*

ABSTRACT

The Electric Split came to fruition after having to deal with lazy roommates who couldn't seem to turn off their electronics and would consume energy unnecessarily. For demonstration purposes, we created our own breaker box with three outlets to plug in various devices into. An LCD was used to display information based on how much power was being consumed in Watt-Seconds and a keypad was used to enter various inputs, such as how long power was to be monitored and the total bill to be split. A PIC microcontroller was used to receive and manipulate all the data needed and then send it out to the LCD. At the end of every billing cycle total power consumption by room (given as a percentage of the total) is given and a final bill amount for each roommate is given.

A total of three rooms including a common room were used. We feel the Electric Split is a great and convenient way for roommates to evenly split a utilities bill in a hassle free way to avoid unnecessary arguments and payments.

Table of Contents

List of Figures

I. INTRODUCTION

The Electric Split is an easy and accurate way to monitor a home's individual room's power consumption. This power monitoring system is extremely useful for college style housing. The idea of this system arouse from a personal need for a product that could accurately divide a monthly power bill between roommates. Currently, homes are not comprised of any sort of system that has the capability to monitor power consumption within individual rooms of a house. Power meters that utility companies use to monitor electrical usage only have the capability to monitor kilo-watt hours of a home and not its individual living quarters. College style housing is often comprised of a single home with anywhere from two to four individual living quarters. Power bills are divided up evenly because of the lack of technology available for individuals to monitor their own living quarter's power consumption. The following is a scenario in which the Electric Split would be advantageous:

College style housing- 3 bedroom home, 3 roommates

Roommate #1- Does not spend much time at home and is very cautious of energy consumption

Roommate #2- Is always home playing video games in his room and has a mini fridge and space heater.

Roommate #3- Has a mini fridge, but does not spend much time in his room.

The electric bill for the first three months was averaging around $340. Roommate number one and three were getting tired of paying the same amount for the electric bill every month even though roommate number two was clearly consuming the most electricity. Therefore, the Electric Split was installed.

The following is the home's three month reading after the system was installed:

Month	Roommate #1	Roommate #2	Roommate #3
#1	10%	75%	15%
#2	8%	74%	18%
#3	12%	68%	20%

Figure #1 (Percent Energy Consumption)

Once the Electric Split was installed the three roommates were able to divide the monthly electric bill evenly by using the power monitoring system that tracks power consumption per room in a home.

Power monitoring is not a new idea by any means. Forms of power monitoring have existed for years; however, most of these devices connect to an electrical outlet and can only monitor the power load on one electrical outlet. Therefore, it would be impractical to purchase an electrical outlet power monitor for every receptacle in a home. The development of the electric meter has gained interest in the past few years because of two main advantages over the traditional electromechanical design. Electric power meters improve accuracy and expand user capabilities.

The Electric Split will use a PIC microcontroller to develop a watt-second power meter. A watt-second meter was designed to measure energy consumed over time. For alternating currents, such as those found in a home, average AC power is the product of the rms voltage, rms current, and cosine of the phase angle between the two.

This report will describe both the hardware and software aspects involved in designing the Electric Split. The system was built on a small scale so that it could be easily demonstrated. Converting the working prototype into a usable device for an existing home will be explained. Future improvements to the Electric Split will also be discussed.

II. HARDWARE IMPLEMENTATION

The hardware used in the Electric Split was carefully selected so that the system would function properly. The first task in selecting the appropriate hardware was to choose a device that would accurately provide the necessary data needed to compute power consumption. In order to decide what device was needed to perform this task, it was first decided where the data would be measured from within a home.

2.1 Power Measurement

The Electric Split is unique in that it has the ability to measure the power consumed by each room in a home. In order to effectively measure the power consumed per room, it was determined that the power lines feeding off the circuit box would be the most appropriate spot in a home to extract measurements. In most homes, each bedroom is wired to have its own circuit breaker. Therefore, a device had to be chosen that had the capability to measure power from each room's circuit breaker.

A current transformer was chosen as the device needed to measure power on each power line feeding off the circuit breaker.

A current transformer is a device used to measure electric current. A current transformer outputs a reduced current reading that is proportional to the current reading of the circuit being measured. Current transformers are designed to withstand high current which most measuring instruments cannot withstand. Current transformers are comprised of a primary winding core, a secondary winding, and a magnetic core. The AC current flowing through the device creates a magnetic field, which induces a current within the current transformer. The secondary current outputs a current reading linearly proportional to primary current. The split-core current transformers from CR Magnetics (Figure #2) were chosen to be used for this project. The current transformer was designed to provide a low cost method for monitoring electrical current. The unique hinge and locking snap allowed for attachment without interrupting the current carrying wire. The high secondary turns have the ability to develop signals up to 10Vac across a burden resistor. The transformer has the capability to operate effectively at either 50 or 60Hz, allowing the product to be useful worldwide. Current transformers are also considered transparent measuring devices, meaning, the transformer does not add any additional load to the system being monitored.

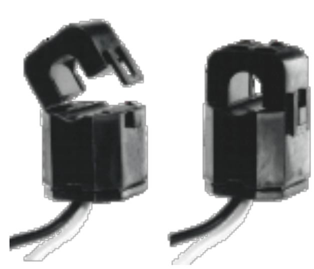

Figure #2 (Current Transformer)

2.2 Microprocessor

The next major component of the Electric Split was the microcontroller. The PIC18F4620 (Figure #3) was chosen because of previous experience and knowledge of working with PIC microcontroller. The PIC's ability to convert analog signals into digital signals at up to 1MHz allows for quick processing of information. The analog data outputted from the current transformers was connected to the analog-to-digital converter in order to be processed.

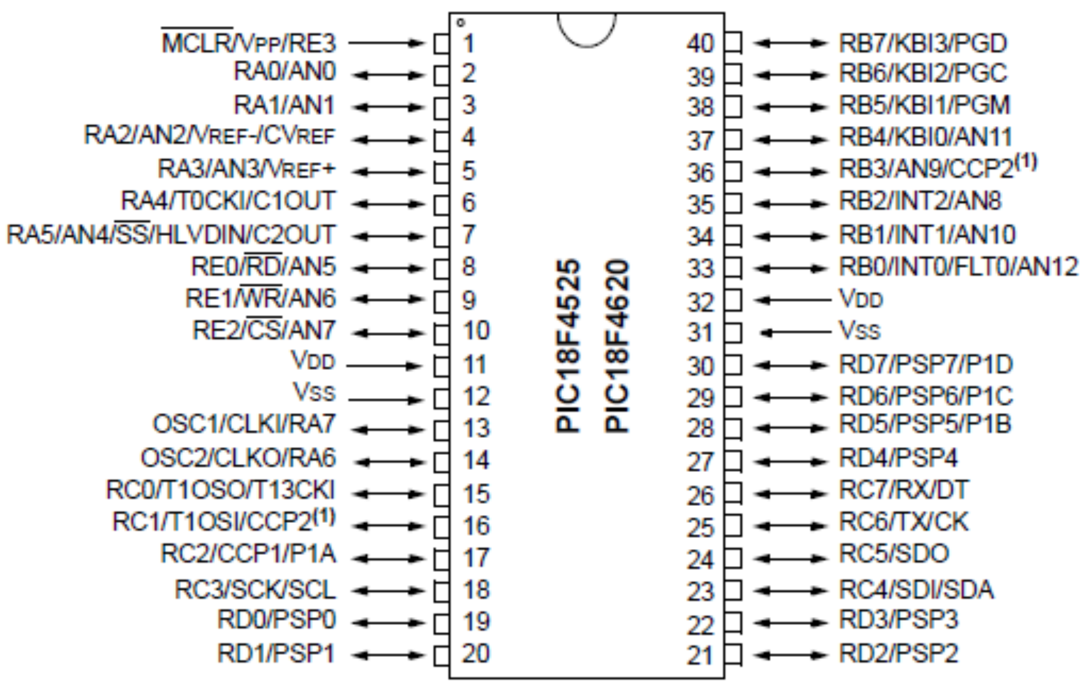

Figure #3 (PIC184620 PinOut)

2.3 User Interface

In order to display the results, a large 160x128 pixel, serial graphical LCD (Figure #4) screen was used. A serial graphical LCD backpack came with the LCD screen which provided a simple serial interface with a full range of controls. Users were also able to input data and communicate with the Electric Split using a basic 12 button keypad (Figure #5).

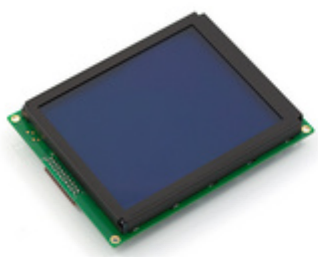

Figure #4 (Serial Graphical LCD)

Figure #5 (12 Button Keypad)

2.4 Hardware Integration

The integration of all the devices mentioned in sections 2.1-2.3 was a major step in successfully building the monitoring system. A block diagram of the process is shown in Figure #6 Setting up the LCD screen was as simple as connecting 6V to the Vin, 0V to GND and a serial TX line from the processor to the RX line on the backpack of the LCD screen. The 12 button keypad was then connected to the PIC through port D of the processor. The next step was to integrate the power measuring device into the system. In order for the current transformer to operate as a measuring device a resistor was placed across the output of the transformer as shown in Figure #7. A 500Ω resistor was used to maximize the accuracy and capability of the transformer. The output of the three current transformers was connected to A0, A1, and A2 pins on the PIC processor. The pins were chosen because of the analog to digital capabilities.

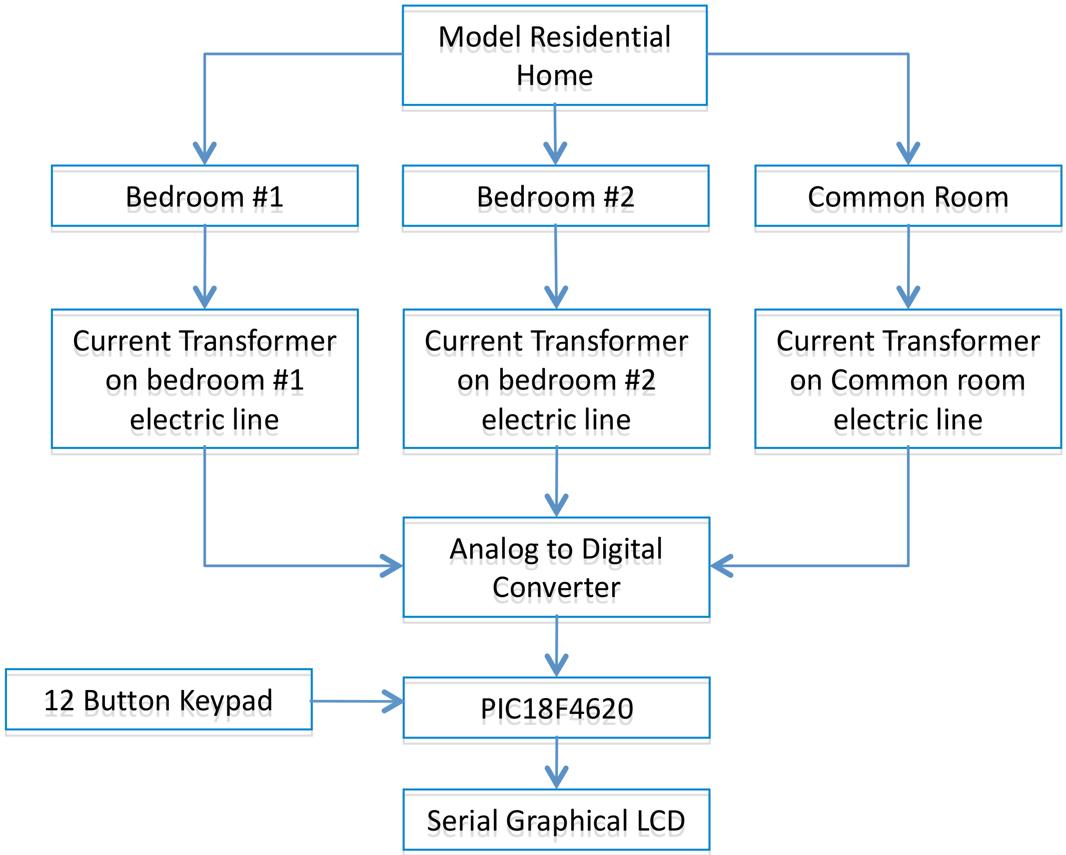

Figure #6 (Hardware Block Diagram)

2.4 Power Supply

To supply energy to the system, a 6V power supply was used. The 6V supply was connected to the LCD screen which requires 6-9V and a voltage regulator which outputted a supply of 5V, necessary to power the PIC microcontroller. The current transformers do not require power, making them an even more desirable measuring tool.

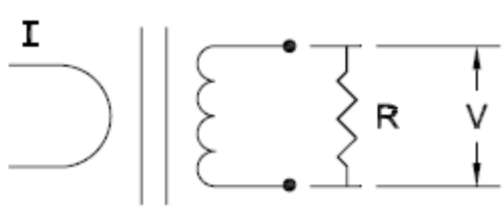

Figure #7 (Current Transformer Circuit)

2.5 Design Layout

The integration of the devices was first tested using a breadboard and then transferred to a PCB once successfully tested. The PCB was designed using Altium, a PC based electronic design software. The schematic of the design is shown in Figure #8, and the layout of the PCB is shown in Figure #9.

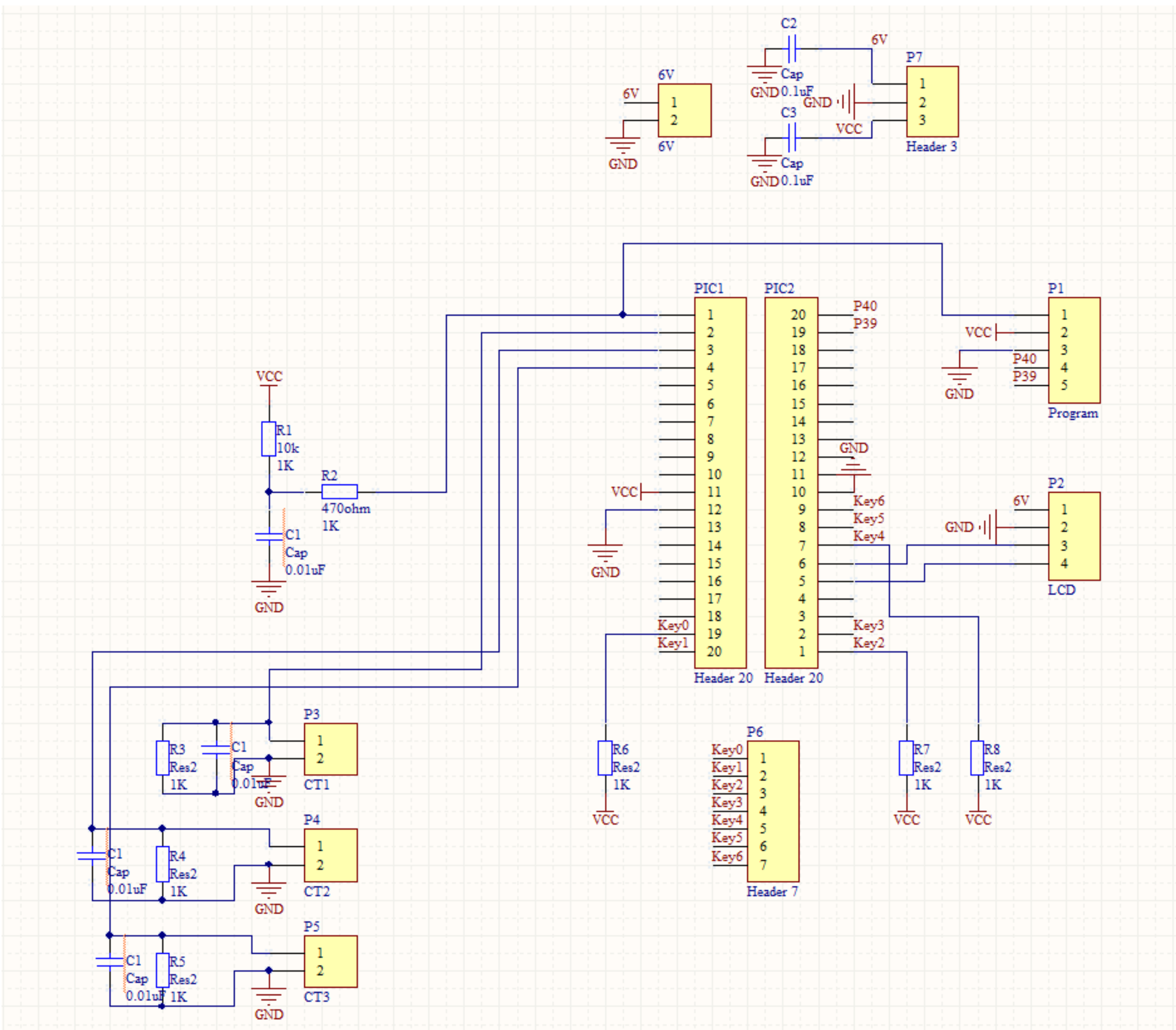

Figure #8 (System Schematic)

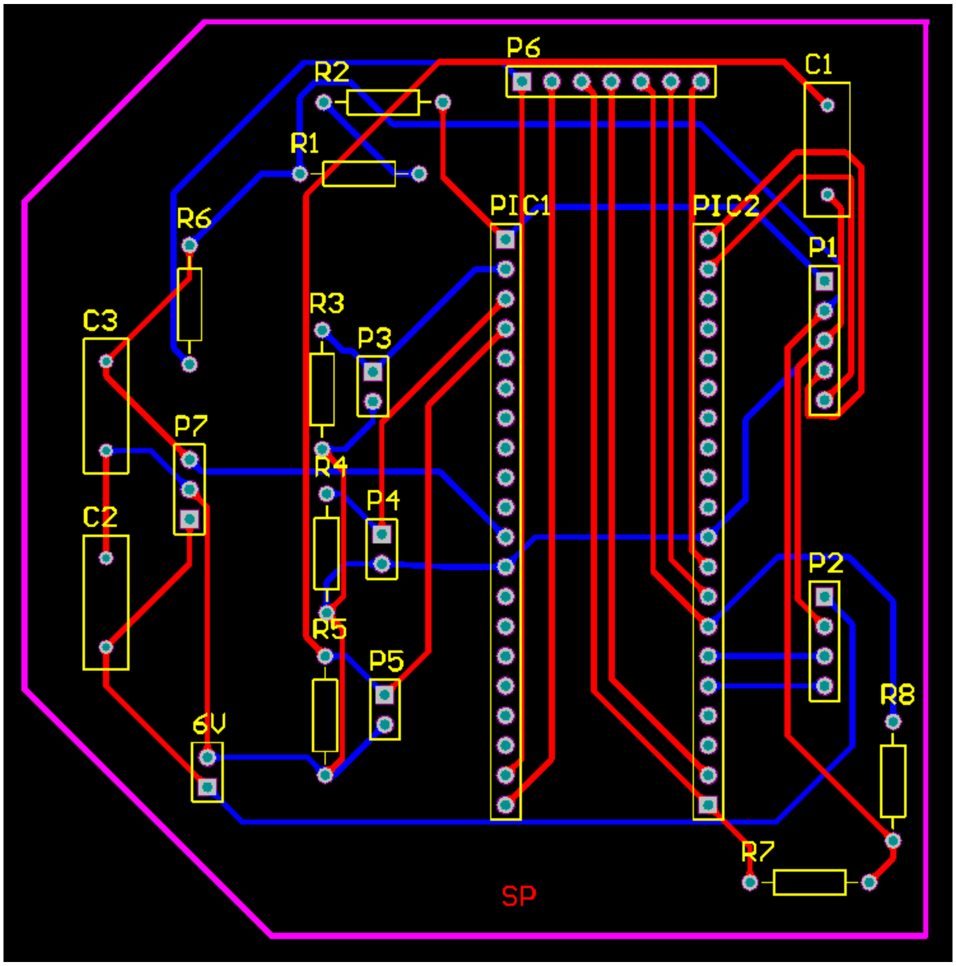

Figure #9 (PCB Design)

III. SOFTWARE DEVELOPMENT

The software needed to develop the system was MPLAB, an integrated toolset for the development of embedded applications on PIC microcontrollers. A basic program was developed to test for functionality of the LCD screen so that it could be used in testing the system as a whole.

3.1 Calibrating Measurements

The current transformers were then tested using a series of different loads. The output of the current transformer was probed so that outputs were displayed on an oscilloscope. The outputs were represented in the form of a sine wave with varying amplitudes depending on load inputs. The outputted maximum amplitude of the sine wave was discovered to be linearly proportional to the input load being tested. A function was derived for this relationship and used to convert maximum amplitude readings into power measurements. The PIC was coded to determine the maximum readings from the current transformers. The maximum value was calculated every 840 milliseconds and converted into an almost instantaneous power reading using the linear function which converted the analog output from the current transformer into a power reading in Watt units. The power was then integrated over time to calculate a kilowatt-second reading.

3.2 User Interface

Once accurate results were calculated through the microcontroller, the user interface was coded and made aesthetically pleasing. A flow chart, as shown in Figure #10, was used as the guideline for developing the user interface. The system was coded to first start by asking the user what the desired cycle time (billing period) was for the application. This time was used to determine how long the system would monitor power readings. Once the cycle time was complete the system then asks the user to enter the bill amount accumulated during the cycle time. The bill amount is then divided accordingly and the LCD screen displays the amount each roommate owes. The user then has the option to continue the previous cycle or continue with a new cycle time.

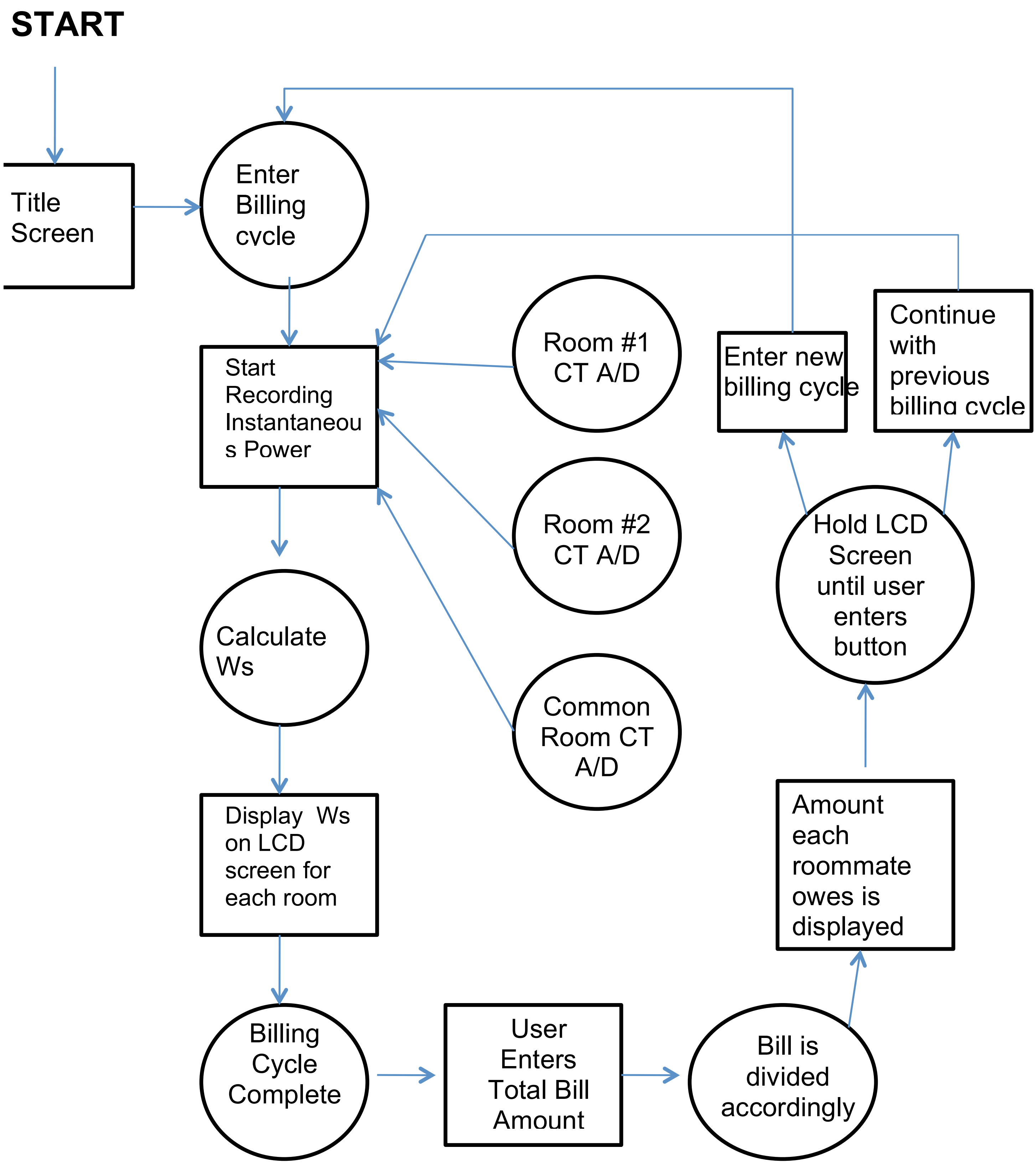

Figure #10 (Flow Chart)

IV. MATERIALS AND TIMELINE

Components	**Cost**
PIC Microcontroller	$7.00
Graphical LCD - Sparkfun	$69.95
Current Transformer (CR3100 Series (.40, 3000)) x3	$44
Demo Equipment	$30.00
12 Button Keypad	$3.95
Total	$155

Figure #11 (Bill of Materials)

Over the course of the project, tasks and duties were split as shown below:

Task	Shahin Pourkaviani	Vivek Tumrukota
Programming Algorithm	20%	80%
PCD Design	50%	50%
LCD	30%	70%
Analog Components	70%	30%
Power Measuring Device	90%	10%
PIC	35%	65%
Model	80%	20%

Figure #12 (Responsibility Table)

A timeline of our project is also given below:

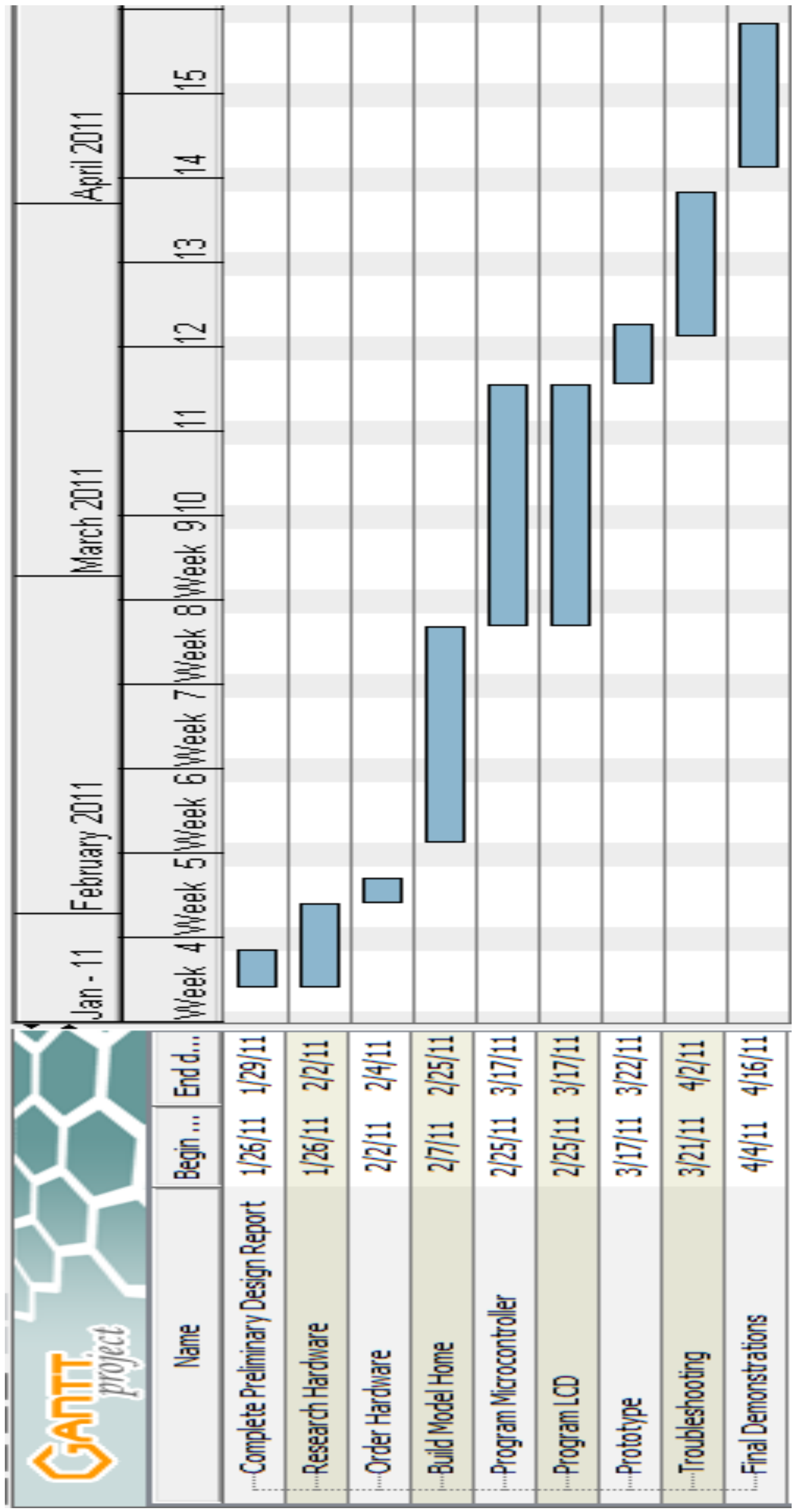

Figure #13 (Gantt Chart)

V. CONCLUSION

The Electric Split serves as a practical device for home tenants who would like the ability to monitor power consumption is each room in a home. The system incorporates a friendly user interface that allows users to monitor instantaneous power consumption and total watt-second readings for each room in a home. Over the course of creating the Electric Split, we encountered a few problems. For one, there was an error with the putrsUSART function in our code. For some reason when this was used to output strings, an extra block character was added to the screen. Many solutions were tried (including hooking up an LSA) but no solution could be found. After doing research online we equated this to the fact that there was no carriage return or new line command for the LCD so it may have something to do with this. Another problem was the fact that the PIC cant output float values in the current version of the compiler. This led to tons of frustration and an eventual 1% error in our calculations (due to integer rounding). In retrospect we wish we had chosen a different controller but this problem came too deep into the project and we had to roll with it. Other than that the project went pretty smoothly. Other functions could have been added if we were to have more time, such as having the ability to display results graphically and store past power consumption data on a USB flash drive, etc.

VI. REFERENCES

1. "How to Measure Your Electrical Use: Electric Meter and Watt-Hour Meter Methods."*Michael Bluejay - Official Home Page*. Web. 15 Mar. 2011. <http://michaelbluejay.com/electricity/measure.html>.
2. "Keypad - 12 Button - SparkFun Electronics." *SparkFun Electronics*. Web. 25 Mar. 2011. <http://www.sparkfun.com/products/8653>.
3. Magnetics, Cr. *CR Magnetics - Current Transformers, Current Transducers, Current Relay, Voltage Transformer, Current Sensing Products*. Web. 1 Mar. 2011. <http://www.crmagnetics.com/>.
4. *Microchip Technology Inc*. Web. 1 Apr. 2011. <http://www.microchip.com/>.
5. "P3 - Kill A Watt." *P3 - Home*. Web. 13 Mar. 2011. <http://www.p3international.com/products/special/P4400/P4400-CE.html>.

APPENDIX A: Code

```
#include <p18f4620.h>
#include <adc.h>
#include <delays.h>
#include <stdio.h>
#include <timers.h>
#include <usart.h>
#include <string.h>
#include <stdlib.h>

#pragma config OSC = INTIO67
                                                    //select internal oscillator
#pragma config WDT = OFF
                                                    //turn off watch dog timer
#pragma config LVP = OFF
                                                    //turn off low voltage program

#define KeypadIO TRISD

                                                    //setup PORTD for Keypad use
#define keypado LATD
#define keypadi PORTD

#define timingio TRISC
#define timing LATCbits.LATC3

int i,j,k,l,peak,in1,time[2],fintime,ws=0,fin,interval;
int wps1=0,wps2=0,wps3=0;
int lcd[10];


void putLCD(char temp[]){
int i;
i = strlen(temp);
for(i=0;i<strlen(temp);i++){
WriteUSART(temp[i]);
}
}
void putsLCD(const rom char* blah){
int i=0;

while(blah[i] != '\0'){
WriteUSART(blah[i]);
i++;
}
```

```
}

void wLCD(char data[]){
do
{  // Transmit a byte
while(BusyUSART());
putcUSART(*data);
} while( *data++  );

}

int maxval (int str[], int len) {
                                                      //function to find max value in array
int i,mxm;
mxm = str[0];
for(i=0;i<len;i++){
if(str[i] > mxm){
mxm = str[i];
}
}
return mxm;
}

int keypadchk(void){
LATD = 0b01111101;
switch(PORTD | ~0x15)
{
case 0xFB:return 1;
case 0xFE:return 2;
case 0xEF:return 3;
}
Delay100TCYx(1);
LATD = 0b00111111;
switch(PORTD | ~0x15)
{
case 0xFB:return 4;
case 0xFE:return 5;
case 0xEF:return 6;
}
Delay100TCYx(1);
PORTD = 0b01011111;
switch(PORTD | ~0x15)
{
case 0xFB:return 7;
case 0xFE:return 8;
case 0xEF:return 9;
}
Delay100TCYx(1);
```

```
PORTD = 0b01110111;
switch(PORTD | ~0x15)
{
case 0xFB:return 10;
case 0xFE:return 0;
case 0xEF:return 11;
}
return 12;
}

void xcord(int pos){

                                        //reset LCD pointer to first position
WriteUSART(0x7C);
Delay100TCYx(25);
WriteUSART(0x18);
Delay100TCYx(25);
WriteUSART(pos);
}

void ycord(int pos){
WriteUSART(0x7c);
Delay100TCYx(25);
WriteUSART(0x19);
Delay100TCYx(25);
WriteUSART(pos);
}

void clrscn(void){
WriteUSART(0x7C);
                                                        //clear screen
Delay100TCYx(50);
WriteUSART(0x00);
Delay100TCYx(50);
}

void drawbox(int x1, int y1, int x2, int y2){
WriteUSART(0x7C);
Delay100TCYx(50);
WriteUSART(0x0F);
Delay100TCYx(50);
WriteUSART(x1);
Delay100TCYx(50);
WriteUSART(y1);
Delay100TCYx(50);
WriteUSART(x2);
Delay100TCYx(50);
WriteUSART(y2);
```

```
Delay100TCYx(50);
}

int timeintv(void){
int i, fin[2], in1;

clrscn();
xcord(0x0A);
ycord(0x5f);
putrsUSART("Please enter a billing    cycle time (mins):");
Delay10KTCYx(5);
for (i=0;i<2;){
in1 = keypadchk();
Delay10KTCYx(10);
if (in1 != 12){
time[i] = in1;
sprintf(fin,"%d",in1);
Delay10KTCYx(30);
WriteUSART(*fin);
Delay10KTCYx(30);
i++;
}
}
Delay10KTCYx(30);
clrscn();
return                                          ((time[0]*10) + time[1]) * 71.43;
}

int watt(void){
overlay int i,j,k,adc1[15],adc2[15],adc3[5];

for(k=0;k<5;k++){
for(j=0;j<15;j++){
for(i=0;i<15;i++){
ADCON0 |= 0x02;

                                                                // set the Go bit
while (ADCON0bits.GO == 1);
                                                // wait until the Go bit is set to 0
adc1[i] = ADRES;
}
adc2[j] = maxval(adc1,15);
}
adc3[k] = maxval(adc2,15);
}
return                                                            maxval(adc3,5);
}
```

```
float thresh (void){
int ser;
float fin;
ser = watt();
fin = ((float)ser - 1.6957) / .1744;
if (fin >= 2){
return fin;
}
else
return 0;
}

void introscn(void){

xcord(0x34);
ycord(0x6E);
putrsUSART("Welcome!");
Delay10KTCYx(0);
xcord(0x10);
ycord(0x50);
putrsUSART(""The Electric Split"");
Delay10KTCYx(0);
Delay10KTCYx(0);
xcord(0x2A);
ycord(0x1f);
putrsUSART("Created By:");
Delay10KTCYx(150);
xcord(0x10);
ycord(0x0f);
putrsUSART("Shahin P. & Vivek T.");
Delay10KTCYx(0);
Delay10KTCYx(0);
Delay10KTCYx(0);
clrscn();
}

void billamt(float r1, float r2, float r3){
int i,in1,in2,bill[3],fbill=0;
long rt,rf1,rf2,rf3;
float rft1,rft2,rft3,r1b,r2b;
overlay int fin[2];
const rom char* li;

xcord(0x20);
ycord(0x6E);
putrsUSART("Cycle Complete!");
Delay10KTCYx(0);
Delay10KTCYx(0);
```

```
xcord(0x00);
ycord(0x40);
putrsUSART("What is the total bill to the nearest dollar?            ($ xxx): $");
Delay10KTCYx(5);
for(i=0;i<3;){
in1 = keypadchk();
if (in1 != 12){
bill[i] = in1;
sprintf(fin,"%d",in1);
Delay10KTCYx(30);
WriteUSART(*fin);
Delay10KTCYx(30);
i++;
}
}
Delay10KTCYx(150);
fbill = ((bill[0]*100)+(bill[1]*10)+bill[2]);
rt = r1+r2+r3;
rf1 = (r1*100l) / rt;
rf2 = (r2*100l) / rt;
rf3 = (r3*100l) / rt;

rft3 = (float)rf3/2;
rft1 = rf1 + (float)rft3;
rft2 = rf2 + (float)rft3;

r1b = ((float)rft1*fbill) / 100;
r2b = ((float)rft2*fbill) / 100;

clrscn();

xcord(0);
ycord(0x7f);
putrsUSART("% of Power By Room:");
Delay10KTCYx(5);
xcord(0x20);
ycord(0x6f);
sprintf(fin,"Room 1:      %d",(int)rf1);
putsUSART(fin);
Delay10KTCYx(5);
xcord(0x20);
ycord(0x60);
sprintf(fin,"Room 2:      %d",(int)rf2);
putsUSART(fin);
Delay10KTCYx(5);
xcord(0x20);
ycord(0x50);
sprintf(fin,"Common Room: %d",(int)rf3);
```

```
putsUSART(fin);
Delay10KTCYx(5);
drawbox(15,25,145,65);
xcord(0x18);
ycord(0x38);
sprintf(fin,"Room 1 Owes: $ %d",(int)r1b);
putsUSART(fin);
Delay10KTCYx(5);
xcord(0x018);
ycord(0x28);
sprintf(fin,"Room 2 Owes: $ %d",(int)r2b);
putsUSART(fin);
Delay10KTCYx(5);
xcord(0x0);
ycord(0x08);
putrsUSART("Press Any Key To Continue");
for(i=0;i<1;){
l = keypadchk();
if(l != 12){
sprintf(fin,"%d",l);
WriteUSART(fin);
i++;
}
}
Delay10KTCYx(5);
}

int cont(void){
int i,j,fin[2];

clrscn();
xcord(0x00);
ycord(0x6f);
putrsUSART("Please Choose from the    following: ");
Delay10KTCYx(10);
xcord(0x0);
ycord(0x4f);
putrsUSART("1 - New Timing Cycle");
Delay10KTCYx(10);
xcord(0x0);
ycord(0x3f);
putrsUSART("2 - Use Same Timing Cycle");
Delay10KTCYx(10);
xcord(0x00);
ycord(0x10);
putrsUSART("Which would you like:");
Delay10KTCYx(10);
for(j=0;j<1;){
```

```
i = keypadchk();
if (i != 12 && i == 1){
sprintf(fin,"%d",i);
WriteUSART(*fin);
Delay10KTCYx(100);
wps1 = 0;
wps2 = 0;
wps3 = 0;
return 1;
}
else if(i != 12 && i == 2){
sprintf(fin,"%d",i);
WriteUSART(*fin);
Delay10KTCYx(100);
wps1 = 0;
wps2 = 0;
wps3 = 0;
return 2;

}
}
}


void main()
{
int over = 1;

Delay10KTCYx(0);
                                        //Delay to not interfere with LCD Baud Rate
Delay10KTCYx(200);
clrscn();
OSCCON |= 0xF7;
BAUDCONbits.BRG16 = 0;
timingio = 0;
KeypadIO = 0x95;
                                                        //Set I/O for Keypad
ADCON1 = 0x0F;

                                                          // PORTA for I/0
ADCON2 = 0x88;

                                          // select internal clock and acquisition time

OpenUSART( USART_TX_INT_ON & USART_RX_INT_OFF & USART_ASYNCH_MODE &

                                                          //SETUP USART
```

```
USART_EIGHT_BIT & USART_CONT_RX & USART_BRGH_HIGH &
USART_ADDEN_OFF,51);

clrscn();
introscn();

while(over == 1){
interval = timeintv();
over = 2;
do{
for(i=0;i<interval;i++){



                                                            //for loop here based on interval
timing = 1;
ADCON0 = 0x01;
fin = thresh();
wps1 = wps1 + (fin * 0.84);
xcord(0);
ycord(0x7f);
putrsUSART("Room 1:");
Delay10KTCYx(10);
xcord(0x20);
ycord(0x71);
sprintf(lcd,"Power:       %d", fin);
putsUSART(lcd);
Delay10KTCYx(10);
xcord(0x20);
ycord(0x68);
sprintf(lcd,"Watt-Seconds: %d",wps1);
putsUSART(lcd);
Delay10KTCYx(10);
timing  = 0;

ADCON0 = 0x05;
fin = thresh();
wps2 = wps2 + (fin * 0.84);
xcord(0);
ycord(0x4f);
putrsUSART("Room 2:");
Delay10KTCYx(10);
xcord(0x20);
ycord(0x41);
sprintf(lcd,"Power:      %d",fin);
putsUSART(lcd);
Delay10KTCYx(10);
sprintf(lcd,"Watt-Second: %d", wps2);
```

```
xcord(0x20);
ycord(0x38);
putsUSART(lcd);
Delay10KTCYx(10);

ADCON0 = 0x09;
fin = thresh();
wps3 = wps3 + (fin * 0.84);
xcord(0);
ycord(0x1f);
putrsUSART("Common Room:");
Delay10KTCYx(10);
xcord(0x20);
ycord(0x11);
sprintf(lcd,"Power:       %d",fin);
putsUSART(lcd);
Delay10KTCYx(10);
sprintf(lcd,"Watt-Second: %d", wps3);
xcord(0x20);
ycord(0x08);
putsUSART(lcd);
Delay10KTCYx(10);
}
clrscn();
billamt(wps1,wps2,wps3);

over = cont();
clrscn();
}while(over == 2);
}
}
```