```
int setCurrentPin = A1;
int setInversityPin = A2;
// voltage input
int inputVol = A3;
int relay = 7;
// summation of samples
double sum=0;
// store samples
double arrval[100];
// number of samples
int N = 0;
double rms;
double val;
// Table of inversity
double alpha[4] = {0.02, 1, 2 ,1};
double beta[4] = {0.14, 13.5, 80, 120};
// 0 = normal, 1 = very inverse, 2 = extremely inverse, 3 = long time
inverse
int inversity;
// calculate rms current
double current;
// a temporary variable to indicate the current state of overcurrent
int emergency;
// Initial setup function
void setup(){
 // Begin Serial communication to display in serial monitor
 Serial.begin(9600);

 // Assign zero to arrval
 for(int i=0;i<100;i++){
 arrval[i] = 0;
 }

 pinMode(relay, OUTPUT);
 digitalWrite(relay, HIGH);
 // To avoid inrush current
 delay(3000);
39
 emergency = 0;
}
void loop(){
 // Final stage of overcurrent, keep the circuit open
 if(emergency==3){
 N = 0;
 return;
 }
 val = analogRead(inputVol);

 //Get Set Current and measure of inversity in a range of 0 to 10A,
0.009775 = 10/1023
 double setCurrent = analogRead(setCurrentPin)*0.009775;

 int tmp = analogRead(setInversityPin);
 if(tmp>767){
 inversity = 3;
 }else if(tmp>511){
 inversity = 2;
 }else if(tmp>255){
```

```
inversity = 1;
}else{
inversity = 0;
}

//Take samples
if(val>=2 && val<6){
while(true){
val = analogRead(inputVol);
if(val<=2 || N>80){
break;
}
arrval[N] = val;
N++;
}
}

//Calculate rms value of current and Top
if(N>10){
sum = 0;
int i = 0;
while(i<N){
// digital to analog conversion
val = arrval[i]*5/1024;
sum = sum + val*val;
i++;
}
sum = sum/N;
```

```
 rms = sqrt(sum);
 //Serial.println(rms);
 rms = (rms*100)/100.0;

 // collect data of source current from ameter and the rms value of
the above line
 // 2nd order regression
 current = 0.19907*rms*rms+1.3293*rms-0.0218624+.1;
 Serial.print("N= ");
 Serial.println(N);
 Serial.print("I = ");
 Serial.println(current);
 Serial.print("Set Current: ");
 Serial.println(setCurrent);
 Serial.print("Inversity: ");
 Serial.println(inversity);
 if(current>setCurrent && emergency==0){
 emergency = 1;
 N = 0;
 delay(1000);
 return;
 }
 if(current>setCurrent && emergency==1){
 emergency = 2;
 double Top = beta[inversity]/(pow(current/setCurrent,
alpha[inversity])-1);
 Serial.print("Top: ");
 Serial.println(Top);
 delay(Top*1000);
 delay(1000);
 }else if(current>setCurrent && emergency==2){
 emergency = 3;
 Serial.println("Disconnecting....");
 digitalWrite(relay, LOW);
 }else{
 emergency = 0;
 }
 }
 N = 0;
}
```