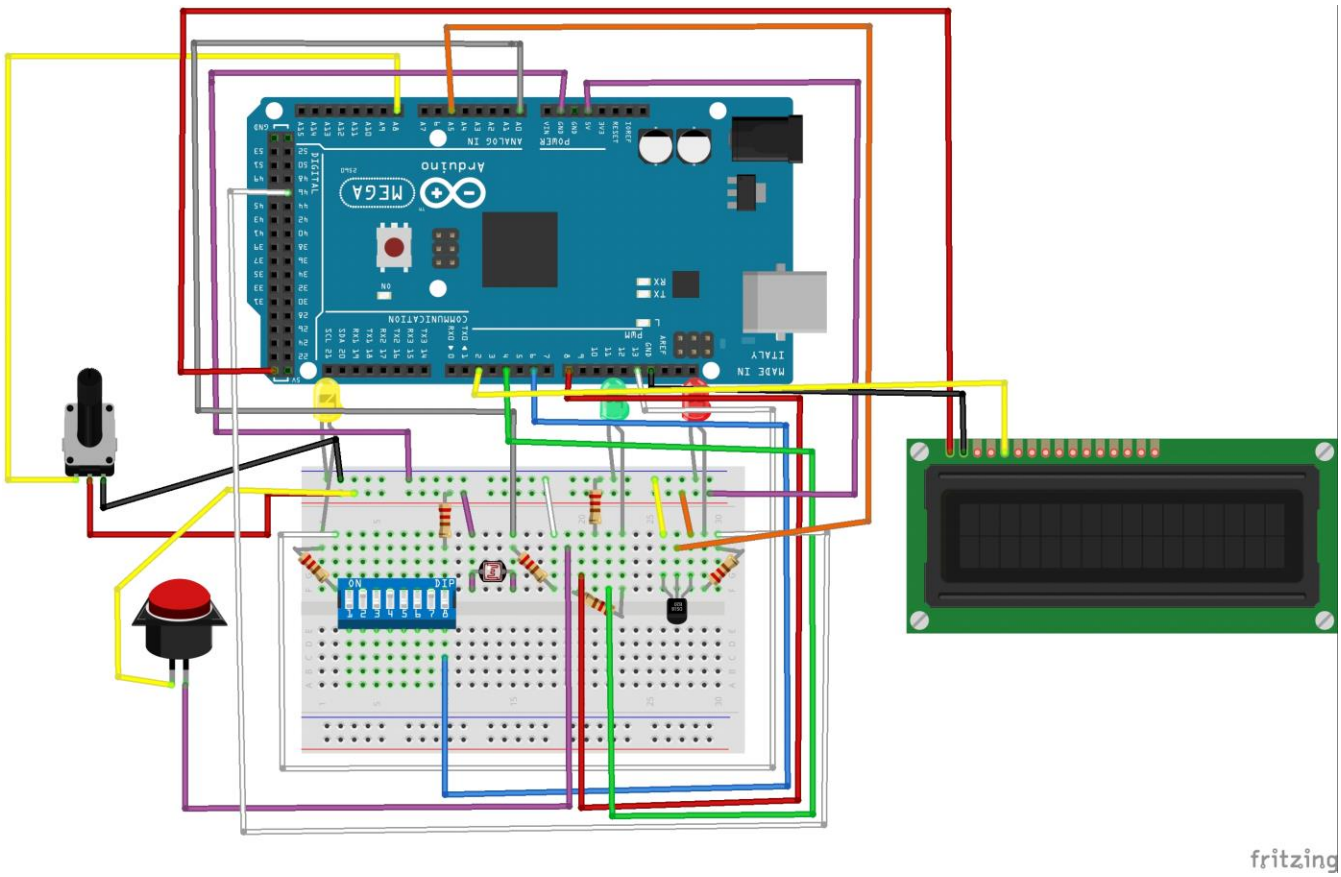
Robert Smith, 100208931

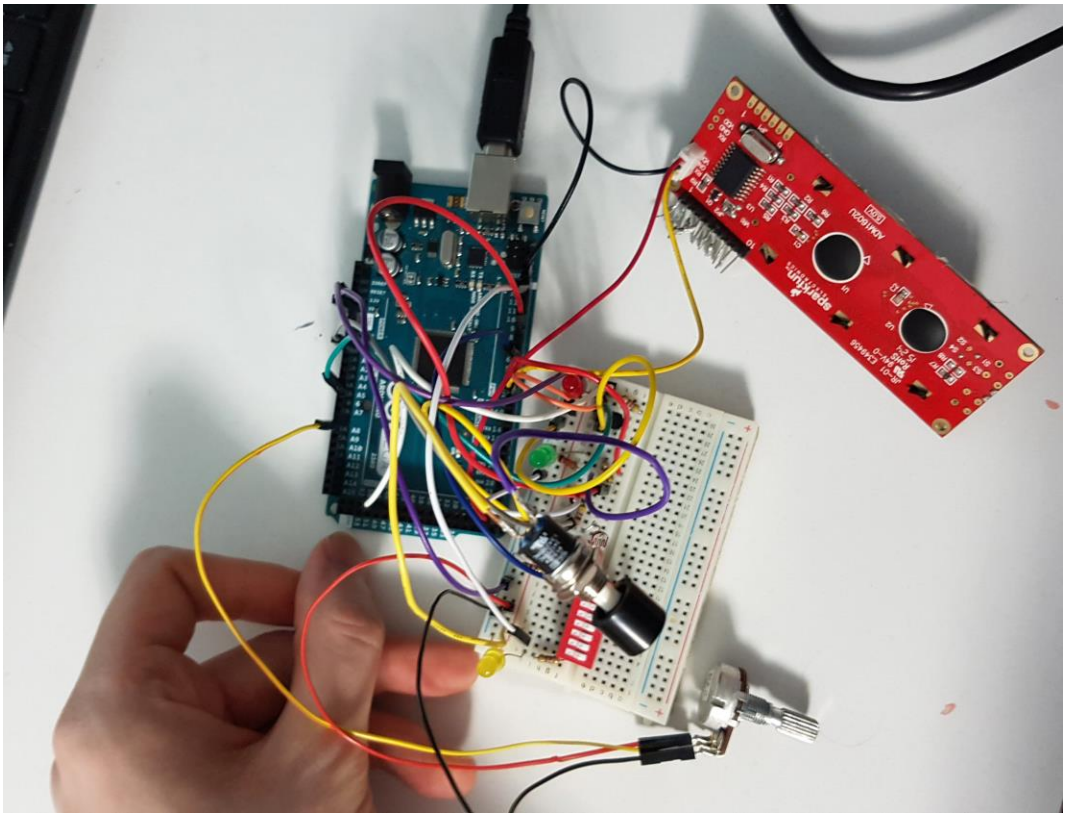
Description of Program's Functionality.

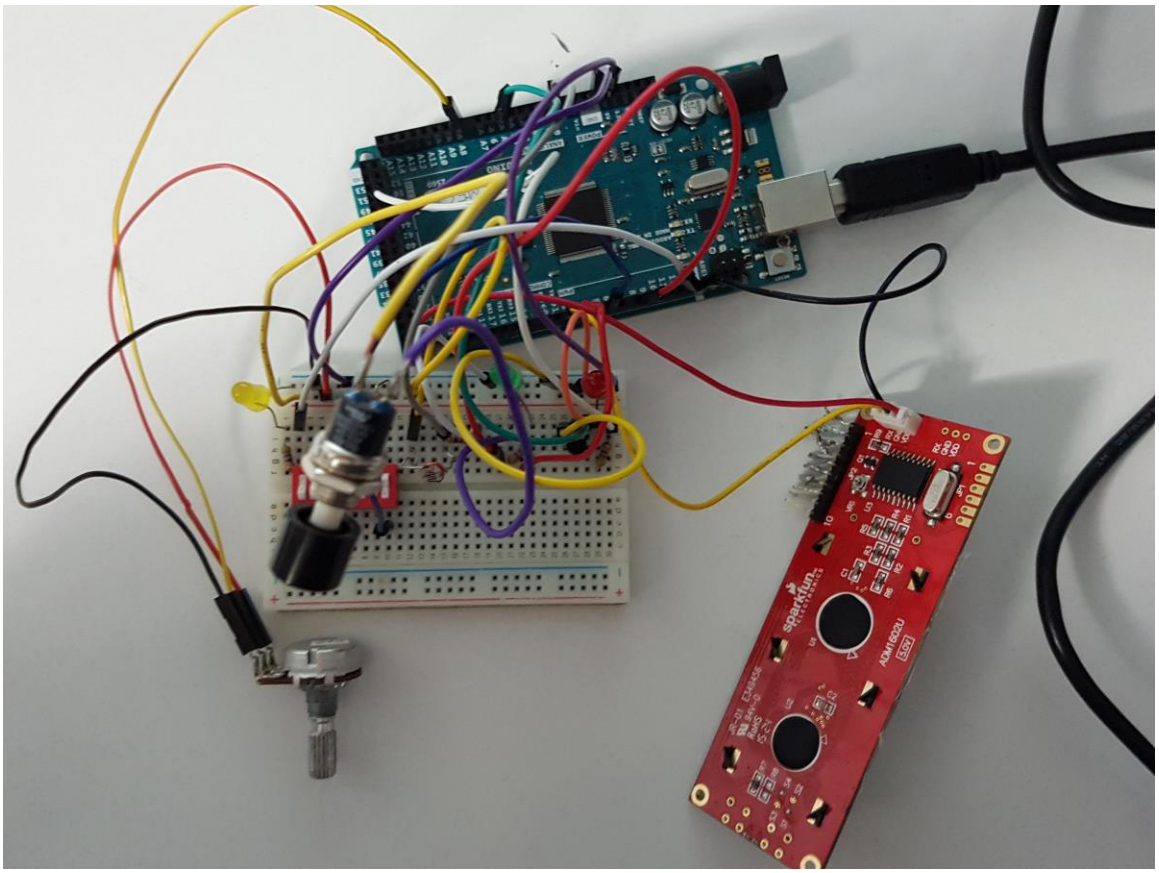
The program collects data from three environmental sensors, the windvane potentiometer, thermoresistor, and photoresistor, that data is averaged and displayed in the serial monitor and LCD screen. The averaged data is used to determine if LEDs should be turned on for the thermo and photoresistors. The windvane potentionmeter uses a more complex approach for determining the wind status and when to light the LED. Each single wind potentiometer data reading is callibrated to a range of 0-100 within the data collection for loop and used to assess the wind status and LED pattern. The area of 40-60 in the middle corresponds to the still zone, and readings above 95 or below 5 indicate a storm. A variable called windadd accumulates the wind readings and is used to compare each reading to the last for the "gusty" status. For controlling the LCD screen, the program uses a push counter variable that increments whenever the pin connected to the pushbutton registers a voltage of HIGH, and then resets when it goes over two. A dipswitch, when pushed, causes a variable (n) in the sensor data collecting for loops to change between 6 and 60 depending on if readings are being averaged across a minute or an hour. Millis, timenow, and a period of 10 seconds are used to ensure that is how long each for loop waits before collecting the next reading.

Fritzing Diagram

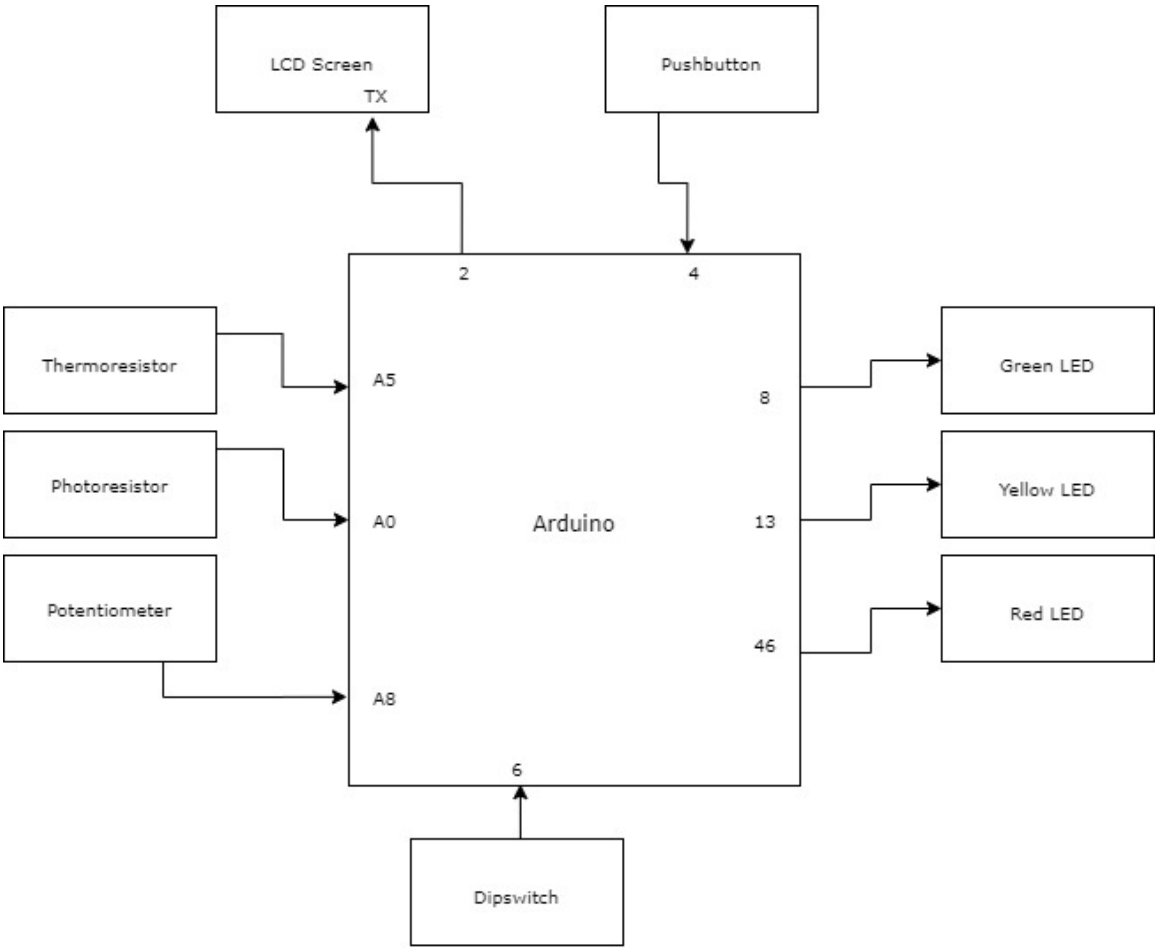


Pictures





Block Diagram


LCD Screen
TX
Pushbutton
2
4
Thermoresistor
A5
8
Green LED
Photoresistor
A0
Arduino
13
Yellow LED
Potentiometer
46
Red LED
A8
6
Dipswitch

CODE

```
#include <SoftwareSerial.h>



//light sensor
int sensorPin = A0; // select the input pin for LDR
int sensorValue = 0; // variable to store the value coming from the sensor
int brightledpin=13;
String lightstatus;

//temperature sensor
int tempsensorpin=A5;
int tempsensorinput;
double temp;
double tempc;
double tempf;
String tempstatus;
int templedpin=8;
```

```
//wind sensor
int windpin=A8;
int windinput=0;
String windstatus;
int windled=46;
void blinkstorm();

//timing variables for millis
int period = 10000;
unsigned long timenow = 0;

//Dipswitch
int dip;
double n;

//pushbutton
int pushpin=4;
int pushcount=0;
boolean check=0;



//Serial Monitor
void printingserial();

//LCD
SoftwareSerial mySerial(3, 2); // pin 2 = TX, pin 3 = RX (unused) //setting up mySerial for digital pin 2
```

```
void LCDwind();

void LCDtemp();

void LCDbright();

void setup() {

Serial.begin(9600); //sets serial port for communication, this one is for serial monitor, had no declaration to send it to any arduino pins earlier so this is just going to serial monitor

mySerial.begin(9600); // using this one for LCD

pinMode(brightledpin,OUTPUT);

pinMode(templedpin,OUTPUT);

//since im using analog in pins , dont actually have to declare them as input with pinmode

//but i did for this one anyway

pinMode(tempsensorpin,INPUT);

pinMode(windpin,INPUT);

pinMode(pushpin,INPUT);

}
```

```
void loop() {



//////////////////PUSHBUTTON CODE//////////////////////////
boolean pushBT=(digitalRead(pushpin));
if (pushBT==1 && check==0){
pushcount++;
check=1;}

if(pushBT==0){
  check=0;
}

if(pushcount>2)
pushcount=0;



//////////////////DIPSWITCH CODE//////////////////////////
dip=digitalRead(6);

if (dip==1)
n=6;
else if (dip==0)
n=60;
```

```
////////////////////////////////////PHOTORESISTOR CODE////////////////////////////////

//COLLECTING THE AVERAGE OVER A MINUTE

double rawsensorvalue=0;

double avgrawsensorvalue=0;

for (int i=0;i<n;i++){

timenow = millis();

rawsensorvalue += analogRead(sensorPin); // read the value from the sensor and add to sum

while(millis() < timenow + period){

  //wait

  }

}

avgrawsensorvalue=rawsensorvalue/n;


//CALIBRATING TO LUX

sensorValue = map(avgrawsensorvalue, 7, 480, 0, 25000); // apply the lux calibration to the sensor reading

sensorValue = constrain(sensorValue, 0, 25000); // in case the sensor value is outside the calibration range

//MADE AN EDIT FROM sensorValue to avgrawsensorvalue, pretty sure it was correct



//CLASSIFYING LIGHT CONDITIONS

if (sensorValue<100)

lightstatus="Dark";

if (sensorValue>=100 && sensorValue<=1000)
```

```
lightstatus="Overcast";
if(sensorValue>=1000 && sensorValue<=10000)
lightstatus="Bright";
if(sensorValue>10000)
lightstatus="Sunny";

//LED CODE
if (sensorValue>10000)
digitalWrite(brightledpin,HIGH);
else
digitalWrite(brightledpin,LOW);



////////////////////////////////WIND POTENTIOMETER CODE//////////////////////////////

//potentiometer will create an anolog input, which will be read as an integer between 0 and 1023
/*When the shaft is turned all the way in one direction,
there are 0 volts going to the pin, and we read 0.
When the shaft is turned all the way in the other direction,
there are 5 volts going to the pin and we read 1023*/

double windtest=0;
double windinput=0;
double avgwind=0;
//still starts as default, if wind goes over 60 or under 40 on ANY single reading in data gathering loop, still=0
/*rest of weather status booleans start at 0, if conditions for one are met and survive until the end of the loop, then
```

```
than boolean will be selected for weather status*/

bool still=1;

bool stormy=0;

bool windy=0;

bool gusty=0;

bool breezy=0;

bool topsidehit=0;

bool bottomsidehit=0;


double winddiff=0;

double windadd=0;

double tempor=0;


for (int i=0;i<6;i++){

timenow = millis();

windtest = analogRead(windpin);//use windtest variable to test individual wind readings within loop

windtest = map(windtest, 6, 1023, 0, 100);// better to callibrate/constrain windtest reading in loop here

windtest = constrain(windtest, 0,100);


tempor=windadd;//store accumulation in another variable

windadd+=windtest; //add this rounds data to accumulation variable

winddiff=abs(windadd-tempor); //winddiff will be difference between this rounds accumulation and last
rounds



if (windtest>60 && windtest<95)

topsidehit=1;
```

```
if (windtest<40 && windtest>5)
bottomsidehit=1;

if (bottomsidehit && topsidehit)
breezy=1;

if ((windtest>60 && windtest<95) || (windtest>5 && windtest<40)){
if (winddiff>10){
   gusty=1;
}
}
else
gusty=0;

if ((windtest>60 && windtest<95) || (windtest>5 && windtest<40)){
if (winddiff<10){
   windy=1;
}
}
else
windy=0;


if (windtest>60 || windtest<40)
still = 0;
if (windtest>95 || windtest<5)
stormy = 1;
```

```
windinput += analogRead(windpin); // use windinput variable to store sums of input from windpin
while(millis() < timenow + period){
  //wait
  }
}



avgwind=windinput/6.0;



avgwind = map(avgwind, 6, 1023, 0, 100); // callibrate potentiometer to a 0-100 scale
avgwind = constrain(avgwind, 0,100); // in case the sensor value is outside the calibration range

if (gusty && !breezy)
windstatus="Gusty";
if (breezy)
windstatus="Breezy";
if (still)
windstatus="Still";
if (stormy)
windstatus="Stormy";
if (windy && !breezy)
windstatus="Windy";

if (stormy)
digitalWrite(windled,HIGH);
```

```
else if (still)
digitalWrite(windled,LOW);
/////PROBLEM AREA, WAS GETTING TRAPPED HERE
else{
blinkstorm();
}


////////////////////////////////THERMORESISTOR CODE//////////////////////////////

double tempsensorinput=0;
for (int i=0;i<n;i++){
timenow=millis();
tempsensorinput+=analogRead(A5); //read the analog sensor and store it
while(millis() < timenow + period){
  //wait
  }
}
tempsensorinput=tempsensorinput/n;
temp=tempsensorinput/1024.0; /*analog voltage input from the thermistor is converted into a
digital value, an integer between 0 and 1023, 0=0V and 1023=5V, therefore divide the integer returned
by 1024 to get the percentage of 5V that was returned*/
temp=temp*5.0; //now multiply by 5V to get the voltage that was returned
tempc=(temp-0.5)*100; //minus 0.5 and multiply by 100 to get temperature in celsius
tempf=((tempc*9.0)/5.0)+32.0;


//classifying temperature conditions
```

```
if (tempc<=0)
tempstatus="Below Freezing";
if (tempc>0 && tempc<=15)
tempstatus="Cool";
if (tempc>15 && tempc<=25)
tempstatus="Warm";
if (tempc>25)
tempstatus="Hot";


//temperature LED code
if (tempc>15)
digitalWrite(templedpin,HIGH);
else
digitalWrite(templedpin,LOW);


////////////////////////////////SERIAL MONITOR CODE////////////////////////////////

printingserial();

//////////////////////////////LCD CODE/////////////////////////////////////////


if (pushcount==0)
LCDtemp();
else if(pushcount==1)
LCDbright();
```

```
else if(pushcount==2)
LCDwind();



}
```

```
//BLINKING STORM LED FUNCTION

void blinkstorm(){
digitalWrite(windled,HIGH);
timenow=millis();
while(millis() < timenow + 200){
  //wait
  }
digitalWrite(windled,LOW);
timenow=millis();
```

```
while(millis() < timenow + 200){
  //wait
}
 digitalWrite(windled,HIGH);
timenow=millis();
while(millis() < timenow + 200){
  //wait
}
 digitalWrite(windled,LOW);
 timenow=millis();
while(millis() < timenow + 200){
  //wait
  }
digitalWrite(windled,HIGH);
timenow=millis();
while(millis() < timenow + 200){
  //wait
  }
digitalWrite(windled,LOW);
timenow=millis();
while(millis() < timenow + 200){
  //wait
}
 digitalWrite(windled,HIGH);
timenow=millis();
while(millis() < timenow + 200){
  //wait
}
```

```
 digitalWrite(windled,LOW);
 timenow=millis();
while(millis() < timenow + 200){
 //wait
 }
 digitalWrite(windled,HIGH);
timenow=millis();
while(millis() < timenow + 200){
 //wait
 }
digitalWrite(windled,LOW);
timenow=millis();
while(millis() < timenow + 200){
 //wait
}
 digitalWrite(windled,HIGH);
timenow=millis();
while(millis() < timenow + 200){
 //wait
}
 digitalWrite(windled,LOW);
 timenow=millis();
while(millis() < timenow + 200){
 //wait
 }
 digitalWrite(windled,HIGH);
timenow=millis();
while(millis() < timenow + 200){
```

```
  //wait
  }
digitalWrite(windled,LOW);
timenow=millis();
while(millis() < timenow + 200){
  //wait
}
 digitalWrite(windled,HIGH);
timenow=millis();
while(millis() < timenow + 200){
  //wait
}
 digitalWrite(windled,LOW);
 timenow=millis();
while(millis() < timenow + 200){
  //wait
  }
}


//SERIAL MONITOR FUNCTION

void printingserial(){
if (dip==0){
timenow = millis();
while(millis()<timenow+600000){
//wait
}
```

```
}
else if (dip==1){
 timenow = millis();
 while(millis()<timenow+60000){
  //wait
 }
}

Serial.println("------------------------------------------------------------");
Serial.println("Mount Lake Resort Date: 31 Jan 2019 Time: 10:23" );
Serial.println("Location: Peak Lake");
Serial.println("------------------------------------------------------------");
Serial.print("Wind:                                          ");
Serial.println(windstatus);
Serial.print("Outside Ambient Light:          ");
Serial.print(sensorValue);
Serial.print(" Lux             ");
Serial.println(lightstatus);
Serial.print("Outside Air Temperature:         "  );
Serial.print(tempc);
Serial.print(" C ");
Serial.print(tempf);
Serial.print(" F        ");
Serial.println(tempstatus);
}



//LCD FUNCTIONS
```

```
void LCDtemp() {
 mySerial.write(254); // move cursor to beginning of first line
 mySerial.write(128);
 mySerial.write("                "); // clear display
 mySerial.write("                ");
 mySerial.write(254); // move cursor to beginning of first line
 mySerial.write(128);
 mySerial.print("Temperature:     ");
 mySerial.print(tempc);
 mySerial.print("C ");
 mySerial.print(tempf);
 mySerial.print("F");
 //do i need to add/can i add while(1); // wait forever
}

void LCDbright() {
 mySerial.write(254); // move cursor to beginning of first line
 mySerial.write(128);
 mySerial.write("                "); // clear display
 mySerial.write("                ");
 mySerial.write(254); // move cursor to beginning of first line
 mySerial.write(128);
 mySerial.print("Light:           ");
 mySerial.print(lightstatus);
 //do i need to add/can i add while(1); // wait forever
}
```

```
void LCDwind() {
 mySerial.write(254); // move cursor to beginning of first line
 mySerial.write(128);
 mySerial.write("                "); // clear display
 mySerial.write("                ");
 mySerial.write(254); // move cursor to beginning of first line
 mySerial.write(128);
 mySerial.print("Wind:           ");
 mySerial.print(windstatus);
 //do i need to add/can i add while(1); // wait forever
}
```