AUTOMATIC LIGHTING SYSTEM IDE PROGRAMME

Floating City

```
int val;

int lightPin = 1;

float limit = 18;

const int LED = 7;


void setup()

{

Serial.begin(9600);

pinMode(7, OUTPUT);

}

void loop()

{

val = analogRead(lightPin);

float steps = val * (255.0 / 5000);

Serial.print("val = ");

Serial.print(val);

Serial.println();

Serial.print("light = ");

Serial.print(steps);

float light = steps +2;

Serial.println(steps +2);




delay(1000);

if (light> limit){
```

```
  digitalWrite(7, HIGH);
}
else {
  digitalWrite(7, LOW);
}
}
```

AUTOMATIC COOLING SYSTEM IDE PROGRAMME

Floating City

```
int val;
int tempPin = 0;
float limit = 25.0;


void setup()
{
Serial.begin(9600);
pinMode(6, OUTPUT);
}
void loop()
{
val = analogRead(tempPin);
float steps = val * (255.0 / 5000);
Serial.print("val = ");
Serial.print(val);
Serial.print("steps = ");
Serial.print(steps);
Serial.print(", c = ");
float temp = steps *10;
Serial.println(steps * 10);
delay(1000);
if (temp < limit){
  digitalWrite(6, HIGH);
}
```

```
else {
  digitalWrite(6, LOW);
}
}
```

AUTOMATIC LIGHTING –COOLING SYSTEM IDE PROGRAMME

Floating City

(LM35 sensor big fluctuation has been controlled by average temperature)

```
int temp;

int tempPin = 0;

float templimit = 23.00; // temperature limit turns fan on

int val2;

int lightPin = 1;

float lightlimit2 = 10;

const int LED = 7;


const int numReadings = 30;     //size of temp array change as required larger number gives a smoother output


int readings[numReadings];      // the readings from the analog input

int readIndex = 0;              // the index of the current reading

int total = 0;                  // the running total

int average = 0;                // the average


void setup()

{

Serial.begin(9600);

pinMode(6, OUTPUT); // Temp

pinMode(7, OUTPUT); // Light

pinMode(13, OUTPUT); // Pump

for (int thisReading = 0; thisReading < numReadings; thisReading++) {

   readings[thisReading] = 0;
```

```
  }
}
void loop()
{
{
 // subtract the last reading:
  total = total - readings[readIndex];
  // read from the sensor:
  analogRead(tempPin);
  delay (10);
  readings[readIndex] = analogRead(tempPin);
  // add the reading to the total:
  total = total + readings[readIndex];
  // advance to the next position in the array:
  readIndex = readIndex + 1;

  // if we're at the end of the array...
  if (readIndex >= numReadings) {
    // ...wrap around to the beginning:
    readIndex = 0;
  }
 // calculate the average:
  average = total / numReadings;
  // send it to the computer as ASCII digits
  Serial.println(average);
  delay(100);        // delay in between reads for stability
```

```
//temp = analogRead(tempPin);
temp = average; // reading from the smoothing array

//float steps = (temp * 5) * 0.1; //convert mV voltage to temp C
//float steps = ((temp / 1024) * 5000) / 10; // convert raw data to C this one works
float steps = (temp * 0.48828125);  //convert mV voltage to temp C this one works

Serial.print("temp = ");
Serial.print(temp);
Serial.print("steps = ");
Serial.print(steps);
Serial.print(" , c = ");
//float temp = steps *10;
Serial.println(steps * 10);
delay(1000);
if (steps >templimit){
  digitalWrite(6, HIGH);
}
else {
  digitalWrite(6, LOW);
}
}
{
  analogRead(lightPin); //take a reading
  delay (10);          // put in a delay to allow for stabalisation of the voltage Debounce
val2 = analogRead(lightPin);  //then take a second reading
```

```
float steps = val2 * (255.0 / 5000);
Serial.print("val2 = ");
Serial.print(val2);
Serial.println( );
Serial.print("light = ");
Serial.print(steps);
float light = steps +2;
Serial.println(steps +2);



delay(1000);
if (light< lightlimit2){
  digitalWrite(7, HIGH);
}
else {digitalWrite(7, LOW);
}
}
}
```