

```

#include <OneWire.h>
OneWire ds(8);
const int numReadingstempensor = 10;
int readingstempensor[numReadingstempensor];
int readIndextempensor = 0;
int totaltempensor = 0;
int averagetempensor = 0;
const int rainsensorMin = 0;
const int rainsensorMax = 1024;
const int numReadingsrainsensor = 5;
int readingsrainsensor[numReadingsrainsensor];
int readIndexrainsensor = 0;
int totalrainsensor = 0;
int averagerainsensor = 0;
int motorPin1 = 3;
int motorPin2 = 4;
int pushsensorclosedPin = 5;
int pushsensoropenPin = 6;
int pushsensorclosedState = 0;
int pushsensoropenState = 0;
int wintercoverledPin = 7;

void setup() {
  Serial.begin(9600);
  pinMode(motorPin1, OUTPUT);
  pinMode(motorPin2, OUTPUT);
  pinMode(wintercoverledPin, OUTPUT);
  pinMode(pushsensorclosedPin, INPUT);
  pinMode(pushsensoropenPin, INPUT);
  for (int thisReading = 0; thisReading < numReadingsrainsensor; thisReading++) {
    readingsrainsensor[thisReading] = 0;
  }
  for (int thisReading = 0; thisReading < numReadingstempensor; thisReading++) {
    readingstempensor[thisReading] = 0;
  }
}

void loop() {
  //Temperature sensor
  byte data[8];
  ds.reset();
  ds.write(0xCC);
  ds.write(0x44);
  delay(500);
  ds.reset();
  ds.write(0xCC);
  ds.write(0xBE);
  data[0] = ds.read();

```

```

data[1] = ds.read();
int Temp = (data[1]<<8)+data[0]; Temp = Temp>>4;

totaltempsensor = totaltempsensor - readingstempensor[readIndextempensor];
readingstempensor[readIndextempensor] = Temp;
totaltempsensor = totaltempsensor + readingstempensor[readIndextempensor];
readIndextempensor = readIndextempensor + 1;
if (readIndextempensor >= numReadingstempensor) {
    readIndextempensor = 0;
}
averagetempensor = totaltempsensor / numReadingstempensor;

Serial.print("Status temperature sensor = ");
Serial.print(averagetempensor);
Serial.println("C");

//Rain sensor
totalrainsensor = totalrainsensor - readingsrainsensor[readIndexrainsensor];
readingsrainsensor[readIndexrainsensor] = analogRead(A0);
totalrainsensor = totalrainsensor + readingsrainsensor[readIndexrainsensor];
readIndexrainsensor = readIndexrainsensor + 1;
if (readIndexrainsensor >= numReadingsrainsensor) {
    readIndexrainsensor = 0;
}
averagerainsensor = totalrainsensor / numReadingsrainsensor;

int rainsensorReading = averagerainsensor;
int range = map(rainsensorReading, rainsensorMin, rainsensorMax, 0, 2);
switch (range) {
    case 0:
        Serial.print("Status rain sensor = ");
        Serial.print(averagerainsensor);
        Serial.println(" (Raining)");
        break;
    case 1:
        Serial.print("Status rain sensor = ");
        Serial.print(averagerainsensor);
        Serial.println(" (Not Raining)");
        break;
}

//Motor
pushsensorclosedState = digitalRead(pushsensorclosedPin);
pushsensoropenState = digitalRead(pushsensoropenPin);
if (pushsensorclosedState == HIGH && range == 0 && Temp > 6) {
    Serial.println("Summer cover closing because of rain");
    digitalWrite(motorPin1,HIGH);
}

```

```

    digitalWrite(motorPin2,LOW);
  }
  else if (pushsensorclosedState == HIGH && Temp > 31 && !range == 0) {
    Serial.println("Summer cover closing because too hot");
    digitalWrite(motorPin1,HIGH);
    digitalWrite(motorPin2,LOW);
  }
  else if (pushsensoropenState == LOW && Temp < 30 && range == 1 && Temp > 6) {
    Serial.println("Summer cover opening because of no rain and not hot enough");
    digitalWrite(motorPin1,LOW);
    digitalWrite(motorPin2,HIGH);
  }
  else {
    Serial.println("Summer cover open or closed");
    digitalWrite(motorPin1,LOW);
    digitalWrite(motorPin2,LOW);
  }

  //Winter cover LED
  if (Temp < 5) {
    Serial.println("Winter cover needed");
    Serial.println( );
    digitalWrite(wintercoverledPin, HIGH);
  }
  else {
    Serial.println("Winter cover not needed");
    Serial.println( );
    digitalWrite(wintercoverledPin, LOW);
  }
  //Delay
  delay(500);
}

```