```
    /*
MP8511 UV Sensor Read Example
The MP8511 UV Sensor outputs an analog signal in relation to the amount of UV light
it detects.
This sensor detects 280-390nm light most effectively. This is categorized as part of
the UVB (burning rays)
spectrum and most of the UVA (tanning rays) spectrum.
There's lots of good UV radiation reading out there:
*/
//Hardware pin definitions
int UVOUT = A2; //Output from the sensor
int REF_3V3 = A3; //3.3V power on the Bluno beetle board
int X1= 4;
int X2 = 3;
int X3= 2;
int buttonPin= A0;
int BUZZER = A1;
int buttonState = 0;
int lastAreaCode= 0;
int greenTimer=0;
int yellowTimer=0;
int orangeTimer=0;
int redTimer=0;
int purpleTimer=0;
int buzzerTimer=0;
int valueTimer=0;//TIMER USED TO CHECK IF ONE MINUTE HAS PASSED
int checkInterval=1;//INTERVAL IN WHICH THE VALUE GETS CHECKED
int updateInterval=12;//INTERVAL IN WHICH THE LED GETS UPDATED
float lastMinuteValues[12];//ARRAY TO STORE THE VALUES
void setup()
{
 Serial.begin(9600);
 pinMode(UVOUT, INPUT);
 pinMode(REF_3V3, INPUT);
 pinMode(BUZZER, OUTPUT);
 pinMode(buttonPin, INPUT);
 Serial.println("MP8511 example");
  int uvLevel = averageAnalogRead(UVOUT);
 int refLevel = averageAnalogRead(REF_3V3);
 //Use the 3.3V power pin as a reference to get a very accurate output value from
sensor
 float outputVoltage = 1.10 / refLevel * uvLevel;
 float uvIntensity = mapfloat(outputVoltage, 0.99, 2.9, 0.0, 15.0);
turnOnLedOnce(uvIntensity);
}
```

```
void turnOnLed(float uvIntensity){

//just for testing the LED
//uvIntensity=14;
 // Serial.println(uvIntensity);



 if(uvIntensity>0&&uvIntensity<3){
    if(lastAreaCode!=1){
      pinMode(X1, INPUT);
      pinMode(X2, OUTPUT);
      pinMode(X3, OUTPUT); //Three-State
      digitalWrite(X2, HIGH);
      digitalWrite(X3, LOW);
      tone(BUZZER, 2000);
      Serial.println("Buzzer turned on");
      Serial.println("Green led turned on");
      Serial.println(lastAreaCode);
      lastAreaCode=1;
      greenTimer=2;
      buzzerTimer=1;
     }
  }
 if(uvIntensity>=3&&uvIntensity<6){
    if(lastAreaCode!=2) {
      pinMode(X1, OUTPUT);
      pinMode(X2, OUTPUT);
      pinMode(X3, INPUT); //Three State
      digitalWrite(X1, HIGH);
      digitalWrite(X2, LOW);
      tone(BUZZER, 2000);
      Serial.println("Buzzer turned on");
      Serial.println("Yellow led turned on");
      Serial.println(lastAreaCode);
      lastAreaCode=2;
      yellowTimer=2;
      buzzerTimer=1;
     }
  }
 if(uvIntensity>=6&&uvIntensity<8){
   if(lastAreaCode!=3) {
      pinMode(X1, OUTPUT);
      pinMode(X2, OUTPUT);
      pinMode(X3, INPUT); //Three-State
```

```
     digitalWrite(X1, LOW);
     digitalWrite(X2, HIGH);
     tone(BUZZER, 2000);
     Serial.println("Orange led turned on");
     Serial.println("Buzzer turned on");
     Serial.println(lastAreaCode);
     lastAreaCode=3;
     orangeTimer=2;
     buzzerTimer=1;
    }
  }
 if(uvIntensity>=8&&uvIntensity<11){
   if(lastAreaCode!=4) {
      pinMode(X1, OUTPUT);
      pinMode(X2, INPUT);
      pinMode(X3, OUTPUT); //Three-State
      digitalWrite(X1, HIGH);
      digitalWrite(X3, LOW);
     tone(BUZZER, 2000);
     Serial.println("Red led turned on");
     Serial.println("Buzzer turned on");
     Serial.println(lastAreaCode);
     lastAreaCode=4;
     redTimer=2;
     buzzerTimer=1;
    }
  }
 if(uvIntensity>=11&&uvIntensity<30){
   if(lastAreaCode!=5) {
     pinMode(X1, INPUT);
     pinMode(X2, OUTPUT); //Three State
     pinMode(X3, OUTPUT);
     digitalWrite(X2, LOW);
     digitalWrite(X3, HIGH);
     tone(BUZZER, 2000);
     Serial.println("Purple led turned on");
     Serial.println("Buzzer turned on");
     Serial.println(lastAreaCode);
     lastAreaCode=5;
     purpleTimer=2;
     buzzerTimer=1;
    }
  }

}
//reset button
```

```
void turnOnLedOnce(float uvIntensity){
 if(uvIntensity<3){
    pinMode(X1, INPUT);
     pinMode(X2, OUTPUT);
     pinMode(X3, OUTPUT); //Three-State
     digitalWrite(X2, HIGH);
     digitalWrite(X3, LOW);
   tone(BUZZER, 2000);
   Serial.println("Buzzer turned on");
   Serial.println("Green led turned on");
   delay(2000);
   pinMode(X1, INPUT);
   pinMode(X2, INPUT);
   pinMode(X3, INPUT);
   noTone(BUZZER);
  }
 if(uvIntensity>=3&&uvIntensity<6){
    pinMode(X1, OUTPUT);
     pinMode(X2, OUTPUT);
     pinMode(X3, INPUT); //Three State
     digitalWrite(X1, LOW);
     digitalWrite(X2, HIGH);
   tone(BUZZER, 2000);
   Serial.println("Buzzer turned on");
   Serial.println("yellow led turned on");
   delay(2000);
   pinMode(X1, INPUT);
   pinMode(X2, INPUT);
   pinMode(X3, INPUT);
   noTone(BUZZER);
  }
 if(uvIntensity>=6&&uvIntensity<8){
   pinMode(X1, OUTPUT);
     pinMode(X2, OUTPUT);
     pinMode(X3, INPUT); //Three-State
     digitalWrite(X1, HIGH);
     digitalWrite(X2, LOW);
   Serial.println("orange led turned on");
   delay(2000);
   pinMode(X1, INPUT);
   pinMode(X2, INPUT);
   pinMode(X3, INPUT);
   noTone(BUZZER);
  }
 if(uvIntensity>=8&&uvIntensity<11){
   pinMode(X1, INPUT);
```

```
      pinMode(X2, OUTPUT);
      pinMode(X3, OUTPUT); //Three-State
      digitalWrite(X2, HIGH);
      digitalWrite(X3, LOW);
   tone(BUZZER, 2000);
   Serial.println("Buzzer turned on");
   Serial.println("red led turned on");
   delay(2000);
   pinMode(X1, INPUT);
   pinMode(X2, INPUT);
   pinMode(X3, INPUT);
   noTone(BUZZER);
   }
 if(uvIntensity>=11&&uvIntensity<30){
   pinMode(X1, INPUT);
     pinMode(X2, OUTPUT); //Three State
     pinMode(X3, OUTPUT);
     digitalWrite(X1, LOW);
     digitalWrite(X3, HIGH);
   tone(BUZZER, 2000);
   Serial.println("Buzzer turned on");
   Serial.println("purple led turned on");
   delay(2000);
   pinMode(X1, INPUT);
   pinMode(X2, INPUT);
   pinMode(X3, INPUT);
   noTone(BUZZER);
 }

  }


void checkTimers(){
 if(greenTimer==1){
   pinMode(X1, INPUT);
   pinMode(X2, INPUT);
   pinMode(X3, INPUT);
   Serial.println("Green LED turned off");
 }
 {
 if(yellowTimer==1){
   pinMode(X1, INPUT);
   pinMode(X2, INPUT);
   pinMode(X3, INPUT);
   Serial.println("Yellow LED turned off");
 }}
```

```
 {if(orangeTimer==1){
   pinMode(X1, INPUT);
   pinMode(X2, INPUT);
    pinMode(X3, INPUT);
   Serial.println("Orange LED turned off");
 }}
 {if(redTimer==1){
   pinMode(X1, INPUT);
   pinMode(X2, INPUT);
   pinMode(X3, INPUT);
   Serial.println("Red LED turned off");
 }}
 {if(purpleTimer==1){
   pinMode(X1, INPUT);
   pinMode(X2, INPUT);
   pinMode(X3, INPUT);
   Serial.println("Purple LED turned off");
 }}
 if(buzzerTimer==1){
   noTone(BUZZER);
   Serial.println("Buzzer turned off");
 }
 greenTimer=greenTimer-1;
 yellowTimer=yellowTimer-1;
 orangeTimer=orangeTimer-1;
 redTimer=redTimer-1;
 purpleTimer=purpleTimer-1;
 buzzerTimer=buzzerTimer-1;
}

void addValueToArray(float uvIntensity){
  // METHOD TO ADD A NEW VALUE TO THE ARRAY
lastMinuteValues[(valueTimer/checkInterval)]=uvIntensity;
}

floatcalculateHighestValue(){
  float value = 0;
  int i;
 for(i=0;i<12;i++){
   if(lastMinuteValues[i]>value){
     value = lastMinuteValues[i];
     Serial.println("New highest value found");
    }
  }
 return value;
```

```
}

//Takes an average of readings on a given pin
//Returns the average
int averageAnalogRead(int pinToRead)
{
  byte numberOfReadings = 8;
  unsigned int runningValue = 0;
  for(int x = 0 ; x < numberOfReadings ; x++)
   runningValue =runningValue+analogRead(pinToRead);
  runningValue = runningValue/numberOfReadings;
  return(runningValue);
}

//The Arduino Map function but for floats
float mapfloat(float x, float in_min, float in_max, float out_min, float out_max)
{
  return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
}


void loop()
{
 int uvLevel = averageAnalogRead(UVOUT);
 int refLevel = averageAnalogRead(REF_3V3);
 //Use the 3.3V power pin as a reference to get a very accurate output value from
sensor
 float outputVoltage = 1.10 / refLevel * uvLevel;
 float uvIntensity = mapfloat(outputVoltage, 0.99, 2.9, 0.0, 15.0);
 Serial.print("MP8511 output: ");
Serial.print(uvLevel);
 Serial.print(" MP8511 voltage: ");
Serial.print(outputVoltage);
 Serial.print(" UV Intensity (mW/cm^2): ");
Serial.print(uvIntensity);
Serial.println();
 addValueToArray(uvIntensity);//ADD CURRENT VALUE TO ARRAY
 Serial.print("Timer: ");
Serial.println(valueTimer);
 buttonState = digitalRead(buttonPin);
 if(buttonState == HIGH){
  void software_Reset();
// Restarts program from beginning but
// does not reset the peripherals and registers
 {asm volatile ("  jmp 0");  }
```

```
 Serial.println("buttonpressed");}
if(valueTimer==(updateInterval-checkInterval)){
    Serial.println("1 minute passed");
    float highestValue=calculateHighestValue();//THIS METHOD CALCULATES THE HIGHEST
OF THE 12 VALUES
    Serial.println("Highest value was: ");
    Serial.println(highestValue);
    turnOnLed(highestValue);//THIS METHODS TURNS ON THE LED
    valueTimer=0;//RESET THE TIMER FOR THE NEXT MINUTE

 }
 else{
   valueTimer=valueTimer+checkInterval;//UPDATE TIMER

 }
delay(checkInterval*1000);
 checkTimers();
}
```