# Sunscreen Reminder Hat

Created by Becky Stern

https://learn.adafruit.com/sunscreen-reminder-hat

Last updated on 2024-06-03 01:33:05 PM EDT

# Table of Contents

# Overview

photo by William Ward (https://adafru.it/dJK)

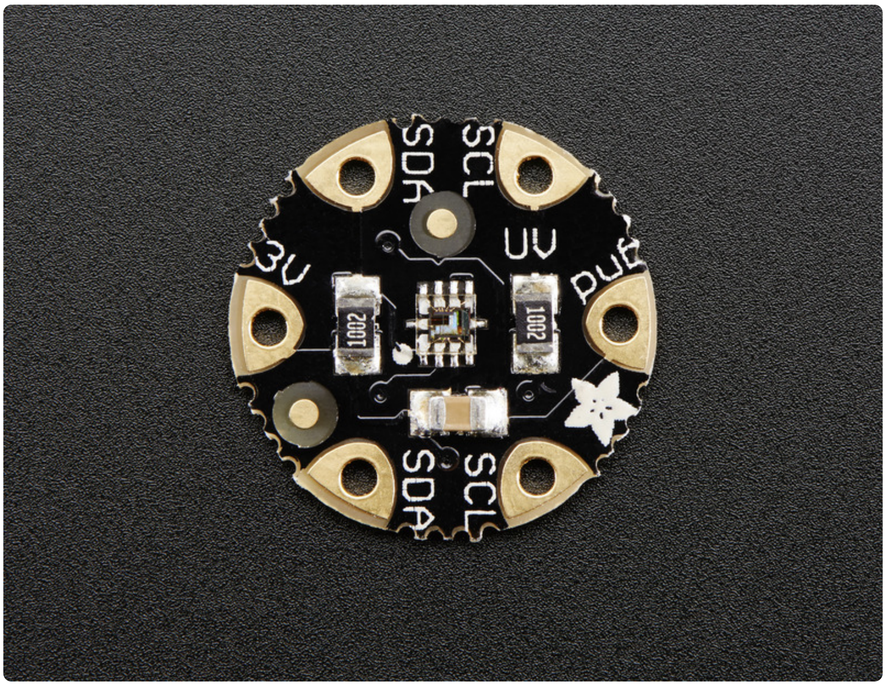
Help remember to reapply your sunblock by building a reminder right into your hat! This sewable circuit uses a FLORA sewable microcontroller and its compatible UV index sensor to sense the sun and play a tune on a piezo buzzer when its time to refresh your SPF.

This is a good beginner project because it doesn't have very many parts and requires no soldering. For this project you will need:

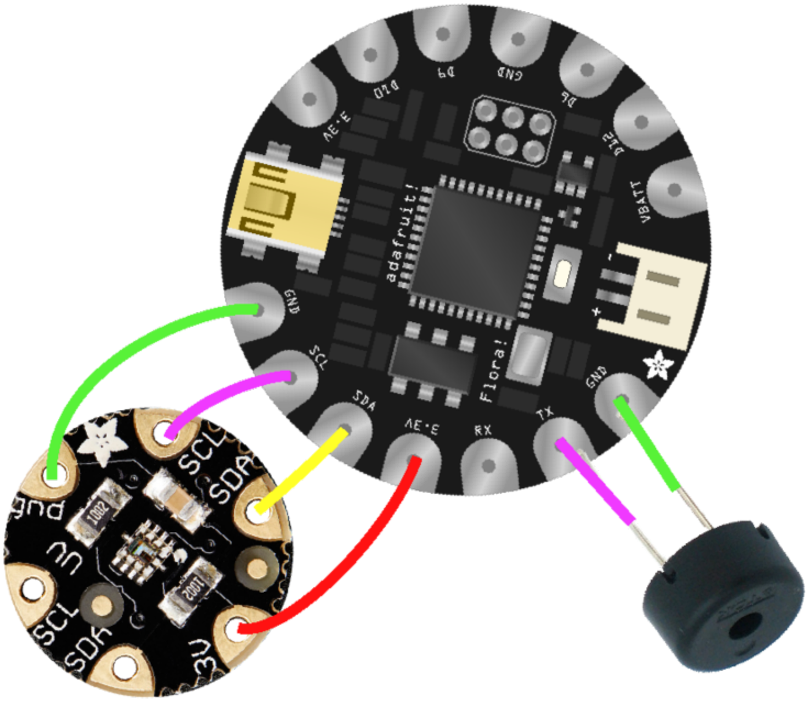
- a hat on which to build your circuit
- FLORA main board (http://adafru.it/659) (The UV sensor does not work with Gemma or Trinket, so a Flora is required!)
- FLORA UV index sensor (http://adafru.it/1981)
- piezo buzzer (http://adafru.it/160)
- 500mAh lipoly battery (http://adafru.it/1578)
- 2-ply conductive thread (http://adafru.it/640)
- sewing needle (http://adafru.it/615) and plain thread
- scissors
- pliers
- clear nail polish

Before you begin, please check out the following prerequisite guides:

- Getting Started with FLORA (https://adafru.it/dJN)
- Wearable Piezo Tones with Flora (https://adafru.it/dJO)
- Adafruit SI1145 Breakout Board - UV / IR / Visible Sensor (https://adafru.it/dJP)



# Circuit Diagram



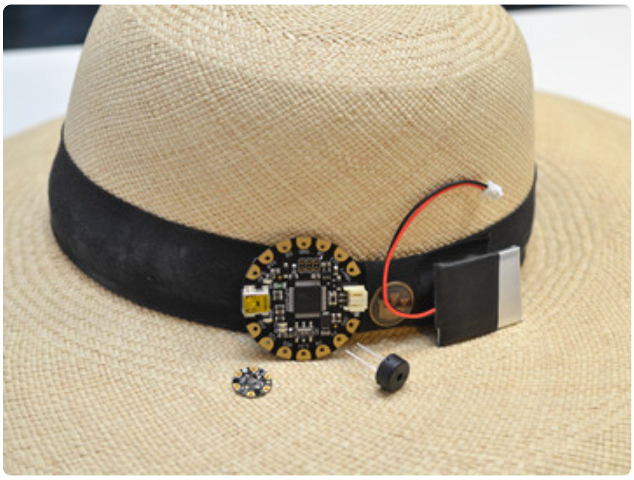
The piezo connects to FLORA GND and FLORA TX (digital pin 1). It does not matter which leg goes to which pin (not polar).

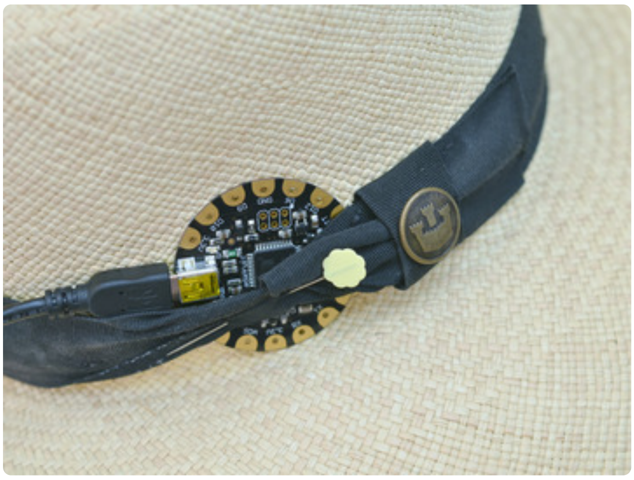
The UV index sensor connects as follows:

- FLORA 3.3v -> sensor 3V
- FLORA SDA -> sensor SDA
- FLORA SCL -> sensor SCL
- FLORA GND -> sensor gnd

You can use any battery pack 3-9V, though we recommend a 500mAh rechargeable lipoly battery (http://adafru.it/1578) for a good balance of long battery life and portability.

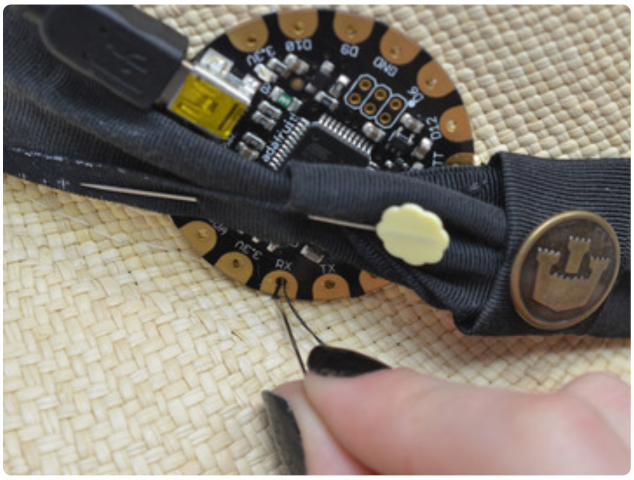
# Stitch FLORA and piezo

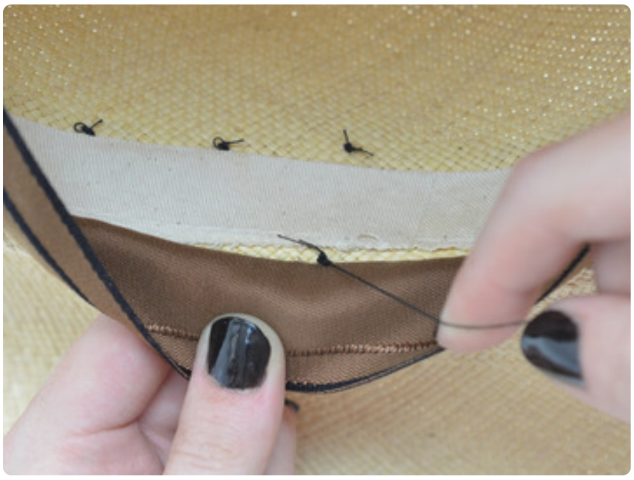




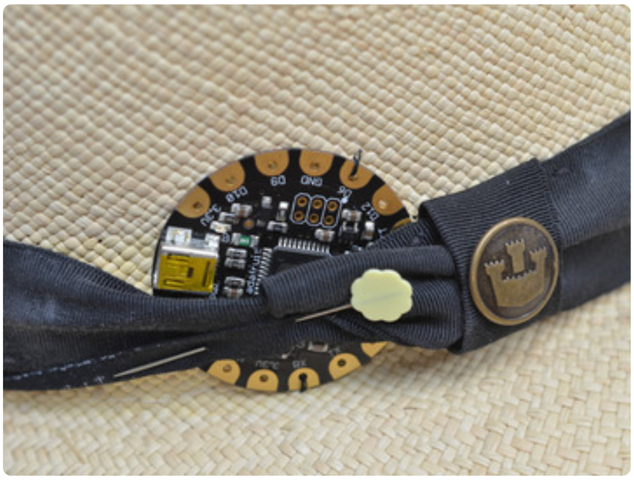
First find a good spot on your hat for the circuit. You want the sensor to face the sky, and the piezo to be near your ear.

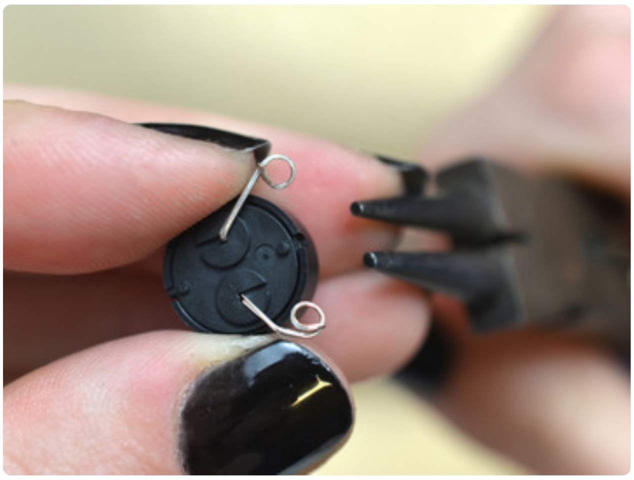
2-ply thread it easy to stitch with but does have increased resistance over long distances, so we recommend putting FLORA as close as possible to the other components-- behind the ribbon on this sunhat is perfect. Pin it back or temporarily remove it so you can stitch your FLORA. Make sure you can access the USB port!

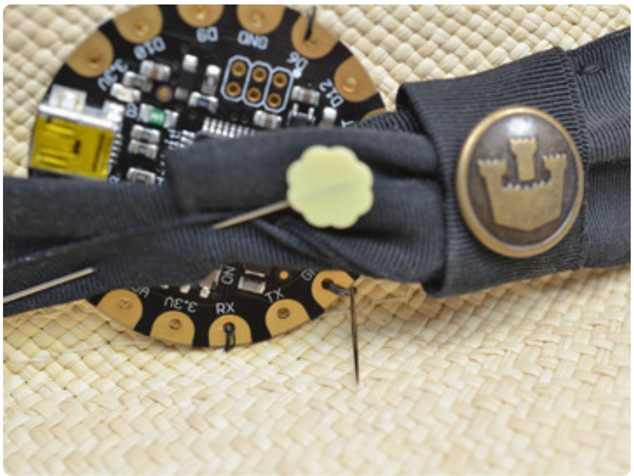




First use plain thread to tack FLORA to your hat. Use the unused RX pin on one side, and any of the pads on the other side so it doesn't shift around. Hide knots behind the grosgrain ribbon on the inside of the hat.



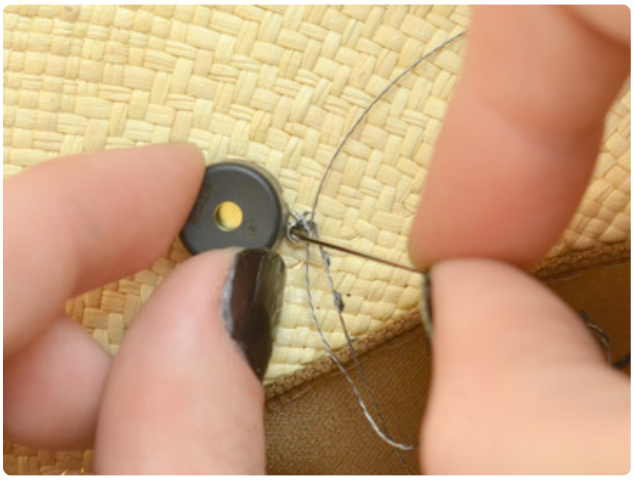


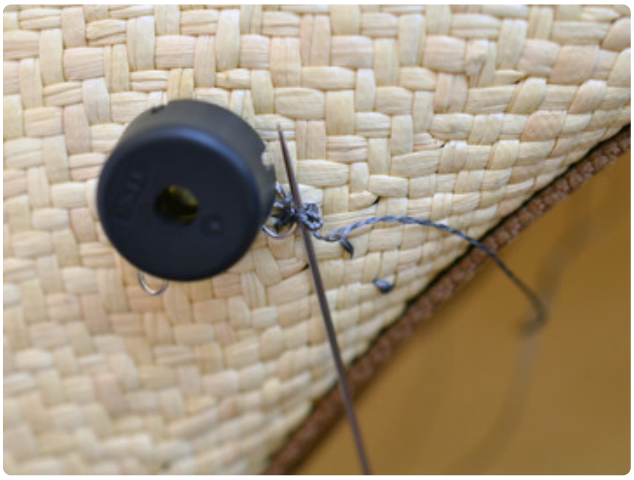


Twirl the leads of the piezo so they're easier to stitch around with conductive thread.

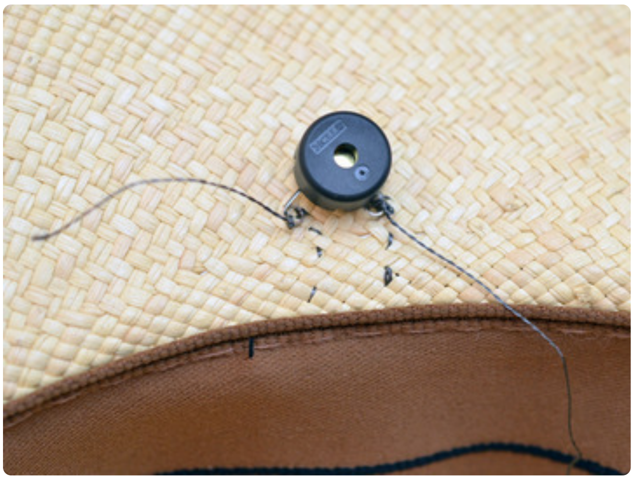
Thread your needle with an arm's length strand of conductive thread and begin stitching around FLORA's GND pad closest to the battery port (see circuit diagram).

Leave a ~4" tail while sewing, then stitch a few times around the GND pad. Then, at the inside of the hat, tie the working thread and the tail in a double knot. Do not trim tail yet.





Use a running stitch to make your way to the location of the piezo (should be facing down near your ear), and stitch many times around one leg of the piezo. Tie a knot and cut the working thread, leaving a tail.



Repeat to attach the FLORA TX pin to the other leg of the piezo. Pull thread tails taut and apply the tiniest little bit of clear nail polish to the knots. Less is more! Try to avoid getting nail polish in or on the piezo connections, just the knot.

Tug on the knots and allow to dry, then check by tugging again that the nail polish is holding the knot tight before trimming the thread tails short.

You may wish to also stitch the piezo to the hat with plain thread to prevent undo stress on the legs.

Test your piezo by loadign up the mario code from the Wearable Piezo Tones guide (https://adafru.it/dJQ).

Look for:

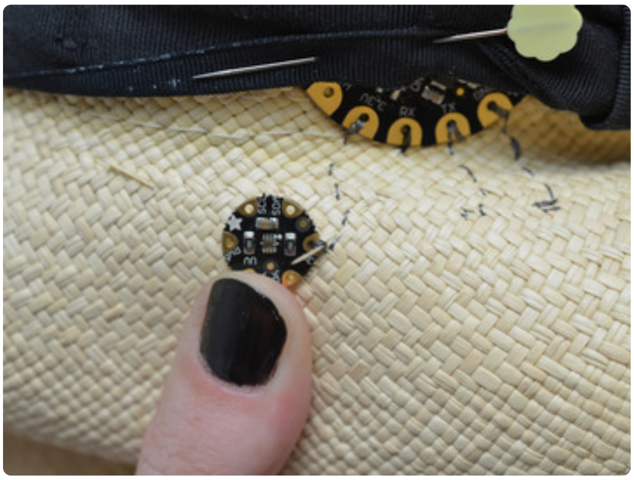
```
int speaker = 9;
```

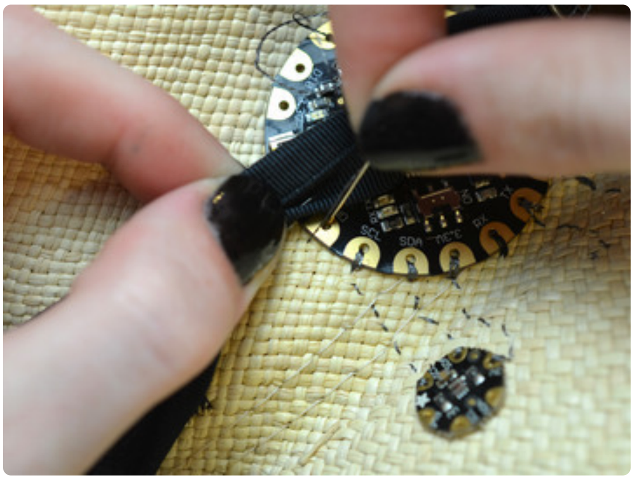
and change it to read:

```
int speaker = 1;
```

Your piezo should play the mario theme. If it doesn't, go back and check your connections!

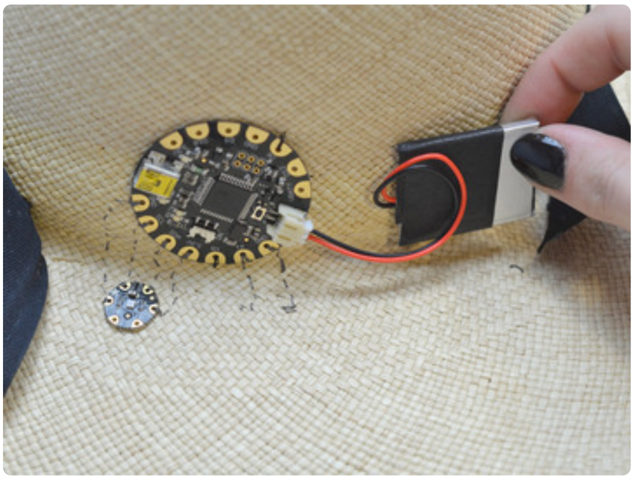
# Add UV index sensor

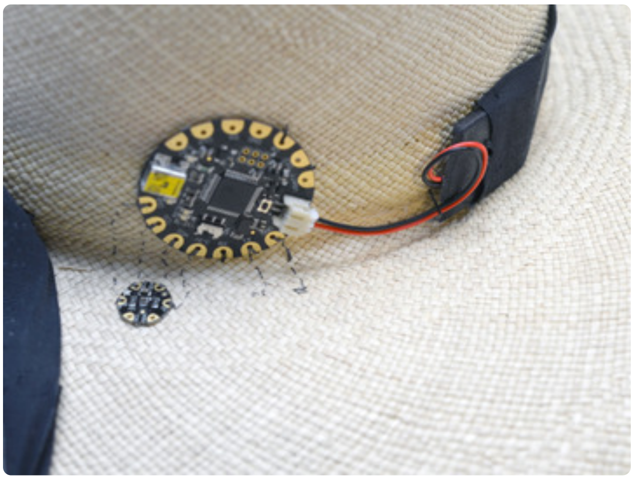




Following the circuit diagram, stitch conductive traces between all four connections to the UV index sensor (3V, SDA, SCL, and GND). Use the same technique as with the piezo, leaving tails on your knots at each end of all four threads.

Apply nail polish to all eight knots, wiping the brush on the edge of the bottle opening first to remove all but the smallest amount. Remember, less is more! Just saturate the knot, not the connection to the FLORA or sensor board. Allow to dry, then tug test before snipping the thread tails short.





Follow the instructions on the UV index guide for testing your sensor-- upload the demo code and verify with the serial monitor that the sensor is returning readings.

Reattach the hat's ribbon using plain thread, securing the battery in the process.

## Code

Upload the following code to your FLORA, ensuring you have the UV index sensor library (https://adafru.it/ddZ) installed properly (you should if you tested your sensor in the last step). This code is suited well for testing your complete circuit indoors, near a window. The UV index threshold is set very low (0.05, achievable by pointing it towards an open window), with a reminder interval of ten seconds. You can also open up serial monitor while it's running to check out the readings and timer.

```
/***************************************************
  Sunscreen reminder hat using the Si1145 UV/IR/Visible Light Sensor
  visit https://learn.adafruit.com/sunscreen-reminder-hat for full tutorial

  contains Super Mario Bros theme composed for piezo by Tiago Gala: http://
therandombit.wordpress.com/2011/11/21/arduino-piezo-speaker-super-mario/

  Written by Becky Stern for Adafruit Industries.
  BSD license, all text above must be included in any redistribution
 ****************************************************/

#include &lt;Wire.h&gt;
#include "Adafruit_SI1145.h"

Adafruit_SI1145 uv = Adafruit_SI1145();

//tones for reminder tune
#define toneC    1911
#define toneC1    1804
#define toneD    1703
#define toneEb    1607
#define toneE    1517
#define toneF    1432
#define toneF1    1352
#define toneG    1276
#define toneAb    1204
#define toneA    1136
#define toneBb    1073
#define toneB    1012
#define tonec       955
#define tonec1      902
#define toned       851
#define toneeb      803
#define tonee       758
#define tonef       716
#define tonef1      676
#define toneg       638
#define toneab      602
#define tonea       568
#define tonebb      536
#define toneb       506
#define tonep       0
int speaker = 1; // piezo wired to FLORA TX


uint32_t sinceTime;
uint32_t markerTime;
boolean shouldChime = false;
float UVindex;
float UVthreshold = 0.05;
uint32_t reapplyInterval = 10000; // 15 minutes = 900000 milliseconds. One hour = 
3600000 milliseconds. Two hours = 7200000 milliseconds

void setup() {
  pinMode(speaker, OUTPUT);
  Serial.begin(9600);
  Serial.println("Adafruit SI1145 test");
  if (! uv.begin()) {
    Serial.println("Didn't find Si1145");
    while (1);
  }
  Serial.println("OK!");
}
```

```
void loop() {
  Serial.println("===================");
  UVindex = uv.readUV();
  UVindex /= 100.0; // the index is multiplied by 100 so to get the integer index,
divide by 100!
  Serial.print("UV: ");  Serial.println(UVindex);
  Serial.print("Seconds until next alert: ");  Serial.println((reapplyInterval-
sinceTime)/1000);
  delay(1000);

  sinceTime = millis()-markerTime;

  if (UVindex &gt; UVthreshold &amp;&amp; shouldChime){ //only chime when we are
currently outside
    chime();
    Serial.println("===================");
    Serial.println("CHIME");
    shouldChime = false;
    resetTimer();
  }
  if (sinceTime &gt; reapplyInterval) { //check to see if we've exceeded the time
limit
    shouldChime = true;
    resetTimer;
  }

}

void resetTimer(){
  markerTime = millis();
  sinceTime = 0;
}

void chime(){
int melody[] = {tonec, toneG, toneE, toneA, toneB, toneBb, toneA, toneG, tonee,
toneg, tonea, tonef, toneg, tonee, tonec, toned, toneB};
int rhythm[] = {18, 18, 18, 12, 12, 6, 12, 8, 8, 8, 12, 6, 12, 12, 6, 6, 6};
long vel = 20000;

for (int i=0; i&lt;17; i++) {
    int note = melody[i];
    int tempo = rhythm[i];

    long tvalue = tempo * vel;

    //tocar(note, tvalue);
    long tempo_progress = 0;
  while (tempo_progress &lt; tvalue) {
    digitalWrite(speaker, HIGH);
    delayMicroseconds(note / 2);

    digitalWrite(speaker, LOW);
    delayMicroseconds(note/2);
    tempo_progress += note;
  }

    delayMicroseconds(1000);
  }

}
```

If everything's working properly, modify these two lines:

```
float UVthreshold = 0.05;
long reapplyInterval = 10000; // 15 minutes = 900000 milliseconds. One hour =
3600000 milliseconds. Two hours = 7200000 milliseconds
```

to reflect the UV index you'd like to sound the alarm, and how often you'd like a reminder.

# Wear it!

photo by William Ward (https://adafru.it/dJK)

This circuit is not waterproof, so don't get it wet! Leave it on your towel when you're going for a dip, hopefully within earshot. Watch out for humidity from your wet hair.

This project is a good starting point for your dream features like GPS, NeoPixel UV status indicators, and more. Modify the alert song, and show us your hat on our weekly Show and Tell on Google+ ()!